

Why and How I Switched to Flatpak for App Distribution and Development in Sandbox

Jiří Janoušek

[@fenryxo](#)

CC-BY-SA 3.0

Part One

What's Flatpak? What is its mission?

What's Flatpak?

“ Flatpak is a next-generation technology for building and distributing desktop applications on Linux. <https://flatpak.org> ”

- Applications as in an app store (Android, iOS)
- Sandbox (gradually improved)
 - You trust the developer enough to run the app
 - But not to let it read passwords and private keys
- Shorter distance between developers and users
 - Apps directly from developers

What **Isn't Flatpak** For?

- Platform services (GNOME Shell, dconf, screen locker)
- System services (HTTP and database servers)
- Server apps (OwnCloud)
- Root access (administrator tools, firewall, btrfs snapshots)
- CLI tools (coreutils, gcc, Python)

Use **dnf/apt, Docker, ...** for that.

Flatpak Ecosystem

- Flatpak - build, install & run flatpaks
 - OSTree - "git for operating system"
 - Bubblewrap - sandbox
- Flatpak Builder - build flatpaks from recipes
- Portals - bridges between host & sandbox
- Frontends - search for and install flatpaks
 - GNOME Software, KDE Discover, ...
- Flathub, Winepack, ... - app repositories

Problem: Software **From Distributions**

- Great for base system, questionable for apps
- Developer has no control over the result
 - Are **bug fixes** backported?
 - Are **all features** enabled?
 - Modifications may introduce **new bugs**
- **Version often frozen** until new distribution release
 - Feedback on **old versions**
 - Already fixed **bugs reported again**

Solution: Software **From Developers**

- **Launchpad PPAs (Personal Package Archives)** popular in **Ubuntu**
 - Developer in **control of quality**
 - **Feedback on the latest version**
 - **Users get latest bug fixes**
 - **Better engagement** between users and developers
- **Flatpak** goes further
 - **Cross-distribution** app delivery method
 - **Sandbox** to improve security

Problem: **Fragmentation** of Linux

- Independent software vendors cannot target **standard Linux**
 - Various distributions, versions of libraries, package formats
 - Bug fixes not always backported
- Common approach: Target old distribution (RHEL, Ubuntu LTS)
 - Disconnected from platform development
 - No incentive to report and fix platform bugs

Solution: **Apps Isolated** From Host

- Apps running in **sandbox**, shielded from host libraries
- Runtime with the **latest stable** library stack
- Benefits from platform development: **new features and bug fixes**
- **Forward compatibility**: no need to rebuild flatpaks for new distro
- **Control over dependencies** (+ basic ones included in *runtimes*)
- Develop and test your app in **the same environment** it will run in

Part Two

Runtimes, Sandbox, Portals, Development

Runtime, Platform, SDK

- **Runtime:** Basic libraries and unix tools
 - Shared to save disk space and memory
 - Quality & security assurance (crypto, codecs, ...)
- **Mounted** as `/usr` in sandbox
- **Platform/SDK:** runtime to run/build flatpaks
 - SDK contains C/C++ header files, compilers, ...
- **Freedesktop SDK** (essential libs) - the base
 - **GNOME SDK** (GTK+, ...), **KDE SDK** (Qt, ...)

App and Bundled Dependencies

- Installed at `/app`
- `./configure --prefix /app ...` or hexa-edit `/usr` → `/app` (last resort)
- Add whatever is missing in runtime
- Add patches
- Override libraries from runtime
- Add application itself

Extensions of Runtimes/Apps

- Mounted at specific locations under `/usr` or `/app`
 - GTK+ and icon themes
 - OpenGL drivers (Nvidia)
 - SDK extensions (Go, Rust, Java, Mono)
 - Debugging symbols
 - Localization
 - Plugins
 - ...

Flatpak **Sandbox**

- Bubblewrap - chroot on steroids
- Private tmpfs: `/`, `/dev/shm`, `/tmp`
- Private namespaces: user, network, process
- Unnecessary system calls disabled
- Read-only mounts: runtime at `/usr`, app at `/app`
- User data: read/write at `~/ .var/app/${APPID}`
- Host files: `/etc/resolv.conf`, icon themes, ...

Permissions to Weaken Sandbox

- **Explicit permissions** must be set when sandbox is created
 - Shown when app is installed - similar to old Android style
- **Filesystem access:** host, \$HOME, individual paths
- **Devices:** DRI, Bluetooth, ...; "all devices"
- **Sockets:** Xorg, Wayland, DBus (filtered), PulseAudio
- **Network access,** development API, ...
- **Long-term goal:** get rid of dangerous permissions

Long Term Goal: **Portals**

- **Trusted Dbus services** on host, called from sandbox
- **Actions confirmed by user** - similar to new Android style
 - Opening/saving files from/on host
 - Opening URIs
 - Printing
 - Taking screenshots
 - ...
- Usage outside Flatpak - **screen sharing** (Wayland); **Snap packages**

Flatpak Builder

- Build flatpaks from recipes ("manifest", JSON/YAML)
- **Source:** directory; archive; git, bzd, svn; patch files
- **Build system:** configure&make, autotools, cmake, meson; custom
- Sources downloaded and verified (checksums)
- Building with limited filesystem and no network access
- CCache supported
- Repeatable build paths `/run/build/...`
- **Incremental builds** - each module cached

eu.tiliado.Hello.yaml (1/3): metadata and bundled dependencies

```
app-id: eu.tiliado.Hello
branch: stable
runtime: org.freedesktop.Platform
runtime-version: "18.08"
sdk: org.freedesktop.Sdk

modules:
  - name: gnome-common
    sources:
      - type: git
        url: "https://g.g.o/GNOME/gnome-common.git"
  - name: adwaita-icon-theme,
    sources:
      - type: git
        url: "https://g.g.o/GNOME/adwaita-icon-theme.git"
  ...
```

eu.tiliado.Hello.yaml (2/3): app itself

```
...
modules:
  ...
  - name: hello
    buildsystem: meson
    config-opts:
      - "--libdir=lib"
    builddir: true
    sources:
      - type: git
        url: "file:///home/fenryxo/Projects/hello"

command: hello
...
```

eu.tiliado.Hello.yaml (3/3): permissions

...

finish-args:

- # Need to download pictures of cats
- "--share=network"
- # Xorg + X shared memory
- "--share=ipc"
- "--socket=x11"
- # Wayland
- "--socket=wayland"
- # DConf
- "--filesystem=xdg-run/dconf"
- "--filesystem=~/.config/dconf:ro"
- "--talk-name=ca.desrt.dconf"
- "--env=DCONF_USER_CONFIG_DIR=.config/dconf"

Base Application

- Useful for apps with common dependencies, e.g. **Electron apps**
1. Build base app with the dependencies
 2. Build child app:
 - Base app is copied to `/app`
 - Add other dependencies
 - Add the app itself
 - Clean up
- Build-only optimization, child app does not depend on base app

eu.tiliado.HelloElectron.yaml

```
app-id: eu.tiliado.HelloElectron
branch: stable
runtime: org.freedesktop.Platform
runtime-version: "1.6"
sdk: org.freedesktop.Sdk

base: io.atom.electron.BaseApp
base-version: stable
base-extensions: [ "io.atom.electron.BaseApp.Debug" ]

command: ...
modules: ...
```

Develop **Inside Sandbox**

- Develop in the same environment users will your run app in
 - Easier reproducibility of bugs
- **GNOME Builder** can build apps in sandbox and create bundles
- **KDevelop** probably too
- **Bash session in the sandbox**

```
flatpak run --devel --command=bash --filesystem=~/projects ...
```

hello

Branch master

Build Profile eu.tiliado.Hello.json

Runtime org.freedesktop.Platform **18.08**

Build status Installing adwaita-icon-theme

Last build Errors 0

Build result Building Warnings 0

Build

Rebuild

Clean

Export Bundle

org.freedesktop.Platform **18.08** ✓

org.freedesktop.Sdk **18.08**

Host operating system

Press Ctrl+. to search

eu.tiliado.Hello.json 49:18

```
47     "name": "adwaita-icon-theme",
48     "sources": [
49
50     ]
51 }
52 }
```

theme.git, ref refs/heads/master
remote: Enumerating objects: 3946, done.
remote: Counting objects: 100% (3946/3946), done.
remote: Compressing objects: 100% (3602/3602), done.
remote: Total 3946 (delta 432), reused 3705 (delta 288)
Receiving objects: 100% (3946/3946), 36.69 MiB | 3.70 MiB/s, done.
Resolving deltas: 100% (432/432), done.
From https://gitlab.gnome.org/GNOME/adwaita-icon-theme
* [new branch] master -> master
Stopping at module hello

```
flatpak-builder --arch=x86_64 --ccache --force-clean --disable-  
updates --disable-download --state-dir /home/fenryxo/.var/app/  
/org.gnome.Builder/cache/gnome-builder/flatpak-builder --stop-  
at=hello /home/fenryxo/.var/app/org.gnome.Builder/cache/gnome-  
builder/projects/hello/flatpak/staging/x86_64-master /home/fen-  
ryxo/dev/projects/linux-days-2018-flatpak/hello/eu.tiliado.Hel-  
lo.json  
Initializing build dir  
Committing stage init to cache  
Starting build of eu.tiliado.Hello
```

(flatpak-builder:103): flatpak-builder-WARNING **: 19:52:55.28

Open Pages

- eu.tiliado.Hello.yaml
- eu.tiliado.Hello.json

Project Tree

- hello
 - build-aux
 - data
 - po
 - src
 - .buildconfig
 - COPYING
 - eu.tiliado.Hello.json
 - eu.tiliado.Hello.yaml
 - meson.build

Press Ctrl+. to search

eu.tiliado.Hello.json 49:18

```
47     "name": "adwaita-icon-theme",
48     "sources": [
49
```

Open Pages

- eu.tiliado.Hello.yaml
- eu.tiliado.Hello.json

Project Tree

- hello
 - build-aux
 - data
 - po
 - src
 - .buildconfig
 - COPYING
 - eu.tiliado.Hello.json
 - eu.tiliado.Hello.yaml
 - meson.build

```
of config/user-dirs.dirs --tmpfs /tmp/.X11-unix --bind /tmp/.X11-u
nix
y -
/us
lin
p/e
**
ili
.fr
Gtk-Message: 19:56:13.654: Failed to load module "pk-gtk-module"
Gtk-Message: 19:56:13.654: Failed to load module "canberra-gtk-mo
dule"
Gtk-Message: 19:56:13.654: Failed to load module "pk-gtk-module"
Gtk-Message: 19:56:13.654: Failed to load module "canberra-gtk-mo
dule"
(hello:2): Gtk-DEBUG: 19:56:13.654: Connecting to session manager
(hello:2): Gtk-DEBUG: 19:56:13.655: Failed to get the GNOME sessi
on proxy: The name org.gnome.SessionManager is not owned
(hello:2): Gtk-DEBUG: 19:56:13.656: Failed to get the Xfce sessio
n proxy: The name org.xfce.SessionManager is not owned
```

Hello, Jiří!

Part Three

How Flatpak helped with my project

Nuvola Apps Project

- Integration of web-based music streaming services (Spotify, ...) with Linux desktop to improve user experience
- Might embrace Progressive Web Apps in future
- Homepage: <https://nuvola.tiliado.eu>
- Source: <https://github.com/tiliado>
- Technologies:
 - GTK+ (GUI), Chromium Embedded Framework (web engine)
 - Vala (core), Python (various tools), JavaScript (integration scripts)



Trash



- All Windows
- Add to Favorites
- Show Details
- Quit





- Pause
- Previous song
- Next song
- Rating: 0 stars
- Rating: 1 star
- Rating: 2 stars
- Rating: 3 stars
- Rating: 4 stars
- Rating: 5 stars
- Nuvola Player**
- Unlock from Launcher



- Keep in Dock

- Pause
- Previous song
- Next song
- Rating: 0 stars
- Rating: 1 star
- Rating: 2 stars
- Rating: 3 stars
- Rating: 4 stars
- Rating: 5 stars
- Quit






- Activate main window
- Pause
- Previous song
- Next song
- Rating: 0 stars
- Rating: 1 star
- Rating: 2 stars
- Rating: 3 stars
- Rating: 4 stars
- Rating: 5 stars
- Quit



- Activate main window
- Play
- Previous song
- Next song
- Rating: 0 stars
- Rating: 1 star
- Rating: 2 stars
- Rating: 3 stars
- Rating: 4 stars
- Rating: 5 stars
- Quit

Nuvola Player



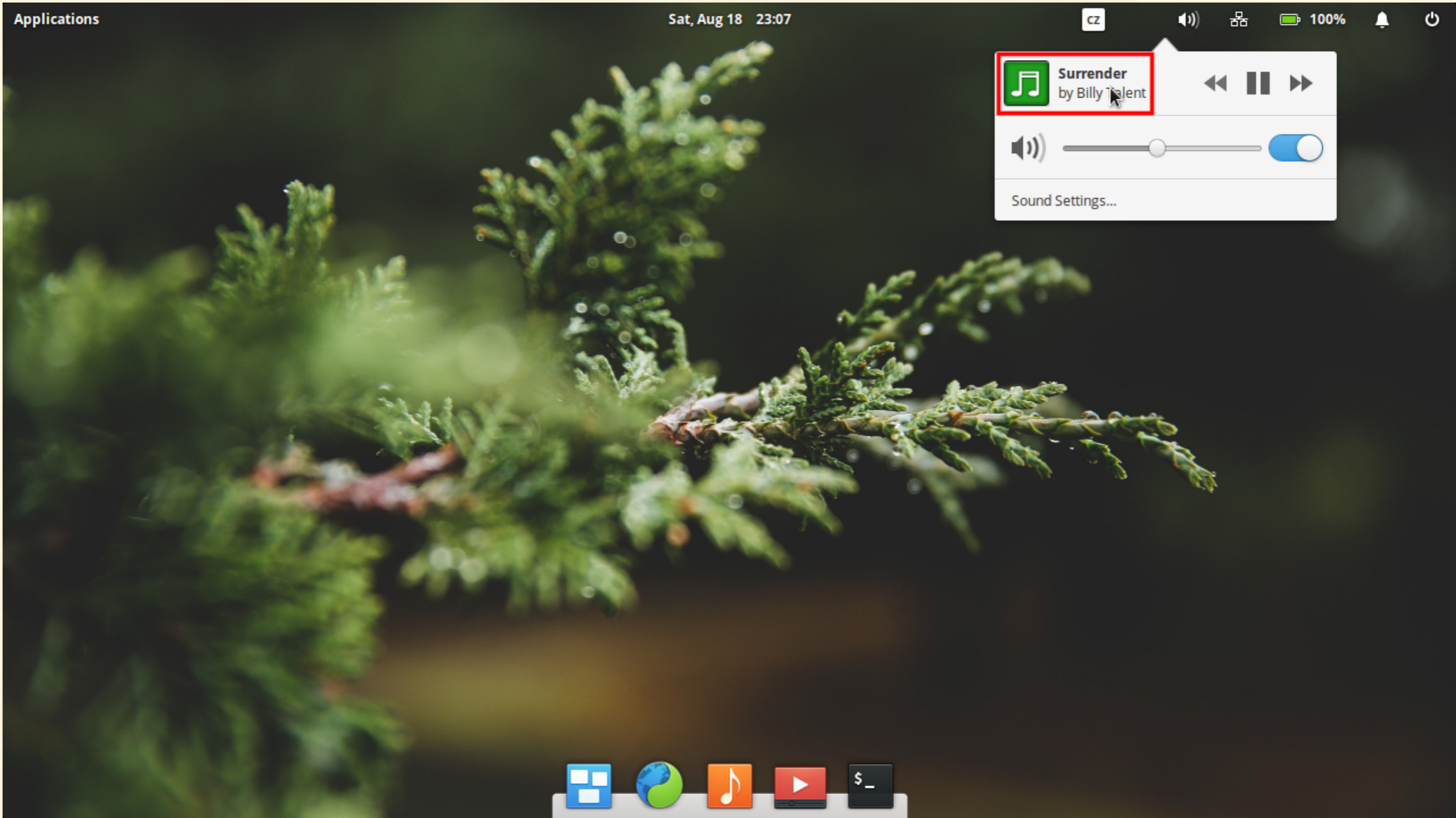
★★★★☆

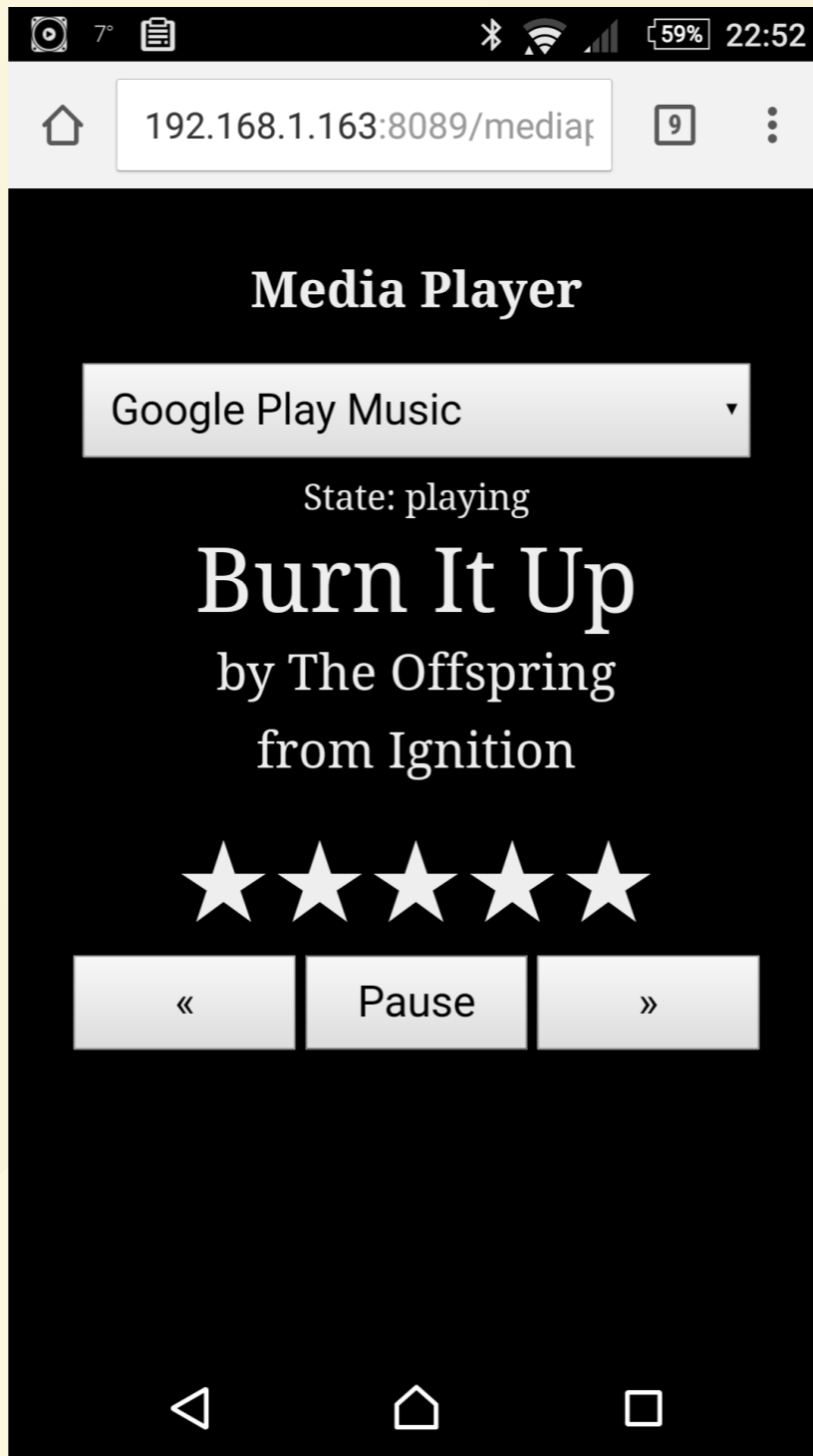
Billy Talent
Surrender
Billy Talent II

⏮ || ⏭

🕒

🔊





**Audio Scrobbling**

Integration with an audio scrobbling service - Last FM.

**Audio Tweaks**

Tweaks for PulseAudio integration.

**Developer's tools**

Enables developer's sidebar.

**Extra Dock Actions**

Adds extra actions to the menu of the application icon in Unity Launcher or elementaryOS dock.

**Lyrics**

Shows lyrics for the current song.

**Media Player DBus Interface**

Implementation of Media Player Remote Interface Specification (MPRIS 2) used by various applets.

**Multimedia keys**

Handles multimedia keys of your keyboard.

**Notifications**

Shows desktop notifications.

**Remote control over HTTP (experimental)**

Remote media player HTTP interface for control over network.

**Tray Icon**

Small icon with menu shown in the notification area.



Maintenance Burden of Packaging

- Nuvola Runtime and Diorite library
 - Stable/development branches, 2 architectures
 - DEBs for Ubuntu - 4 releases (from LTS to LTS)
 - DEBs for Debian - 3 suites (stable, testing, unstable)
 - RPMs for Fedora - 2 releases
 - $2 \times 2 \times (4 + 3 + 2) = 36$ packages
- Web app integration scripts (`metadata.json`, `integrate.js`, icons)
 - DEBs/RPMs built with <https://github.com/jordansissel/fpm>

Incompatibilities of Deps Over Time

- **Vala Compiler:** Vala code → C code → binary
 - Every release fixes bugs and improves VAPIs
 - But sometimes in an incompatible way
 - `#if VALA_0_40 ... #elif ... #else ... #endif` nightmare
- **ABI/API breaks:** webkit2gtk-3.0 × webkit2gtk-4.0, json-glib, GTK
- **With Flatpak:**
 - Switch to new SDK version in *devel* brach
 - Promote to *stable* when ready, no need to hurry

Nuvola × WebKitGTK+

- **WebKitGTK+**: library to embed web rendering engine into GTK+ applications; used by GNOME Web, Devhelp, GNOME Help, ...
- **Rapid** development of web technologies
- Distributions: WebKitGTK **upgraded slowly**
- With Flatpak: **always up-to-date WebKitGTK**
 - Possibility to enable and test experimental features
 - *Media Source Extension* for audio playback without Flash plugin

Nuvola → Chromium

- **WebKitGTK+**
 - *MSE* still not good enough (Google Play Music still didn't work)
 - *Flash* support also buggy ([Bug 172450](#))
 - No support for *Widevine* (needed by Spotify)
- **Chromium Embedded Framework**
 - Not available in distributions as library
 - *MSE* & *Widevine* supported
 - Only Amazon Music still needs Flash

Flatpaks of **Nuvola: Now**

- Hierarchy of base apps for a specific purpose
- eu.tiliado.NuvolaCdk - Core Developer Kit
 - eu.tiliado.NuvolaAdk - App Developer Kit
 - eu.tiliado.NuvolaBase - production base
 - eu.tiliado.Nuvola - Nuvola service
 - eu.tiliado.NuvolaAppSpotify - Spotify
 - eu.tiliado.NuvolaAppDeezer - Deezer
 - eu.tiliado.NuvolaApp...

CDK: Core Developer Kit

- All deps for Nuvola but without Nuvola
- Development of Nuvola in the same sandbox environment the app will run in
- Quick set-up everywhere, especially in VirtualBox

```
$ cd ~/projects/nuvolaruntime
$ flatpak install nuvola eu.tiliado.NuvolaCdk
$ flatpak run -d --filesystem=~/projects eu.tiliado.NuvolaCdk
$ . setup_nuvolacdk.sh
$ rebuild
$ run-app nuvola-app-nuvola-demo-player
```

ADK: App Developer Kit

- CDK as *base app* + Nuvola Runtime + Nuvola SDK
- Nuvola SDK = a few helper scripts in Python/Bash
- Development of web app integration scripts in JavaScript, no need to build Nuvola or learn Vala

```
$ flatpak install nuvola eu.tiliado.NuvolaAdk
$ flatpak run --filesystem=~/projects eu.tiliado.NuvolaAdk
$ nuvolasdk new-project ...
$ cd ./nuvola-app-hello
$ nano metadata.in.json integrate.js
$ nuvolasdk check-project; ./configure; make all
$ nuvolaruntime --debug
```

Nuvola *Service* and Apps

- `eu.tiliado.NuvolaBase` = CDK + *production* build of Nuvola, development tools removed
 - `eu.tiliado.Nuvola` = NuvolaBase cleaned-up further + AppStream metadata
 - **Nuvola Service:** media keys, global keybindings, remote control over HTTP, ...
 - `eu.tiliado.NuvolaAppDeezer`, `NuvolaApp...`
 - NuvolaBase + web integration script
 - `metadata.json`, `integrate.js`, icons, desktop file, AppStream

Flatpaks of Nuvola: Future

- Modernize aging manifests (from the era of Flatpak 0.8)
- **Nuvola CDK: *base-app* → Platform/SDK based on GNOME SDK?**
 - Less copying and hashing → **faster builds?**
 - Development with **GNOME Builder?**
 - **Maintenance:** GNOME SDK gets update → CircleCI job rebuilds Nuvola CDK
 - WIP <https://github.com/tiliado/nuvola-flatpaks>
- **Drop old distros and embrace new Flatpak and Portal features?**

Drawbacks of Flatpak Packaging

- Maintenance of bundled libraries
 - Script to update manifest with the latest versions
 - Version information from <https://release-monitoring.org/api>
- Missing functionality in sandbox - talk to Flatpak devs
- Poor command line experience:
 - `flatpak run --command=nuvolactl eu.tiliado.Nuvola track-info`
 - <https://github.com/flatpak/flatpak/issues/1188>
- **New limiting factor** on host OS - version of Flatpak, Portals, ...

GTK+ Themes

- Available as Flatpak extensions since 0.8.4 (2017-03-10)
- Flatpak command installs the current theme automatically
- GNOME Software since 3.30.0 (2018-09-04)
- Expect that your user might have **older version**
- **Adwaita**: default/fallback theme, may seem too bright
- **Greybird**: fallback theme in Nuvola, less bright (neutral)
- Nuvola lets user change theme



Home

Recent

Home

Desktop

Documents

Downloads

Music

Pictures

Videos

Trash

Other Locations

Desktop

Documents

Downloads

Music

Pictures

Public

snap

Templates

Videos

Open Untitled Document 1 Save

Col 1 INS

Basic Mode

7 8 9 ÷ ↶ C

4 5 6 × ()

1 2 3 - x² √

0 . % + =

DEEZER

Home

Browse

Hear This

My Music

Favorite Tracks

Playlists

Albums

Apps

Appearance Tweaks

Change user interface theme and window decoration preferences.

User interface theme
This option has no effect on the theme of the web app itself. Adwaita

Get more themes
Themes for Flatpak applications can be found on Flathub, the store for Flatpak apps. A restart is required after the installation of a new theme. [Install themes](#)

Prefer a dark variant if the theme provides it
If no dark variant is available, this option has no effect. OFF

Use dark scrollbars for web view
This option applies only on the web app itself. OFF

Use system window decorations (requires restart)
It may cause a visual inconsistency if the system theme differs from the application's theme. OFF

SUBSCRIBE

Inspired

Hardwell, Raiden,

Popular



Home

Recent

Home

Desktop

Documents

Downloads

Music

Pictures

Videos

Trash

Other Locations

Desktop

Documents

Downloads

Music

Pictures

Public

snap

Templates

Videos

Open

Untitled Document 1

Save

Col 1

INS

Basic Mode

7	8	9	÷	←	C
4	5	6	×	()
1	2	3	-	x ²	√
0	.	%	+	=	

DEEZER

Home

Browse

Hear This

My Music

Favorite Tracks

Playlists

Albums

Apps

Appearance Tweaks

Change user interface theme and window decoration preferences.

User interface theme
This option has no effect on the theme of the web app itself. Greybird

Get more themes
Themes for Flatpak applications can be found on Flathub, the store for Flatpak apps. A restart is required after the installation of a new theme. [Install themes](#)

Prefer a dark variant if the theme provides it
If no dark variant is available, this option has no effect.

Use dark scrollbars for web view
This option applies only on the web app itself.

Use system window decorations (requires restart)
It may cause a visual inconsistency if the system theme differs from the application's theme.

SUBSCRIBE

Inspired

Hardwell, Raiden,

Popular



Home

Recent

Home

Desktop

Documents

Downloads

Music

Pictures

Videos

Trash

Other Locations

Desktop

Documents

Downloads

Music

Pictures

Public

snap

Templates

Videos

Open

Untitled Document 1

Save

Col 1

INS

Basic Mode

7 8 9 ÷ ↶ C

4 5 6 × ()

1 2 3 - x² √

0 . % + =

Appearance Tweaks

Change user interface theme and window decoration preferences.

User interface theme
This option has no effect on the theme of the web app itself. Ambiance

Get more themes
Themes for Flatpak applications can be found on Flathub, the store for Flatpak apps. A restart is required after the installation of a new theme. [Install themes](#)

Prefer a dark variant if the theme provides it
If no dark variant is available, this option has no effect. OFF

Use dark scrollbars for web view
This option applies only on the web app itself. OFF

Use system window decorations (requires restart)
It may cause a visual inconsistency if the system theme differs from the application's theme. OFF

DEEZER

Home

Browse

Hear This

My Music

Favorite Tracks

Playlists

Albums

Apps

SUBSCRIBE

Inspired

Hardwell, Raiden,

Popular

Nvidia OpenGL Drivers

- Available as Flatpak extensions since 0.8.4 (2017-03-10)
- Flatpak command installs the driver automatically
- GNOME Software since 3.30.0? (2018-09-04), maybe sooner
- Expect that your user might have older version
 - or freshly released driver is not on Flathub yet
- Nuvola check the presence of the driver
 - Better than to just crash

Portals Not Found

- Portals are only weak dependencies of Flatpak
- **Expect** that they might not be installed and dependent features may fail (silently)
 - Network proxy settings
 - Opening links in user's default web browser
- **Nuvola** checks the presence of Desktop Portal (via Dbus introspection) and refuses to start if it isn't available
 - or if it crashes (in OpenSUSE, already promptly fixed)

Part Four

How about Snapcraft and AppImage?

What's **Snapcraft**?

- Flatpak competitor with key differences
- Broader focus: Desktop apps as well as **server & CLI apps**
- Snaps work out of the box on **Ubuntu**
 - More adventurous on other distributions
 - SELinux, strict confinement availability, `/snap` directory
- You may need **Ubuntu LTS** to build snaps
- You can publish snaps to **Ubuntu Store**
 - Readily available to **large user base**

Snapcraft × Nuvola

- Snapcraft seems to use some **core bits from Ubuntu LTS**
 - Snap created in Ubuntu 17.10 failed to load (C library issue?)
- Nuvola uses the **latest stable GNOME stack**
 - Build it on top of Ubuntu 16.04 LTS core libs?
 - Use GNOME platform snap package?
- How about GTK+ themes, Nvidia drivers, debugging symbols?
- Cannot give up - native to Ubuntu - **large user base**
- Need to try again with Ubuntu 18.04 LTS

What's **AppImage**

- Tool to create cross-distribution **self-executable bundles**
- **Easy installation:** download, `chmod a+x` and run
 - No need to install any tool to manage App Images
- **Portable mode** - apps on USB stick
- **Optional features:**
 - Sandbox - Firejail
 - Desktop integration - `appimaged`, `AppImageLauncher`
 - Updates - `AppImageUpdate`

ApplImage × Nuvola

- Probably not a good fit
- ApplImage creation approach:
 - Build on old systems, run on newer systems
 - If you need newer libs, bundle them
- Nuvola:
 - Uses the latest stable GNOME Platform stack
 - Don't want to build it and maintain it
- How about GTK+ themes, Nvidia drivers, debugging symbols?

Part Five

Recommended articles

Alexander Larsson

- [Flatpak history, Kick-starting the revolution 1.0](#)
- [The flatpak security model - part 1: The basics](#)
- [The flatpak security model - part 2: Who needs sandboxing anyway?](#)
- [The flatpak security model, part 3 - The long game](#)
- [Maintaining a flatpak repository](#)
- [On application sizes and bloat in flatpak](#)
- [The birth of a new runtime](#)

Matthias Clasen

- [Flatpak in detail, part 1](#) - bundling, runtimes, extensions, host
- [Flatpak in detail, part 2](#) - organization on host filesystem
- [Flatpak in detail, part 3](#) - flatpak sandbox from the inside
- [Flatpak, making contribution easy](#)
- [Flatpak - a look behind the portal](#)
- [On Flatpak updates](#)
- [About Flatpak installations](#)
- [On Flatpak dependencies](#)

Patrick Griffis

- [Flatpak now supports themes](#)
- [Flatpaking application plugins](#)
- [Easier Flatpak manifest editing with VSCode](#)
- [Using host Nvidia driver with Flatpak](#)
- [Status of emulators Flatpak'd](#)

Jan Grulich

- [Telegram Desktop Client For Flatpak](#) - related to Qt toolkit
- [Telegram Desktop Client For Flatpak 2](#)
- [Telegram Desktop Client For Flatpak 3](#)
- [KDE Flatpak Portals Introduction](#)
- [Flatpak Support In KDevelop](#)
- [Gnome Integration For Qt Based Applications In Flatpak](#)

Jiří Eischmann

- [Flatpak: New way to distribute linux applications](#) (Czech only)
 - June 1st, 2016
 - Jiří Janoušek discovered Flatpak!
- [Why I use Flatpak for 3rd party apps](#)
- [Flathub Experience: Adding an App](#)
- [Telegram Desktop Client for Flatpak](#)
- [Dark Adwaita and HighContrast Themes for Qt](#)



My Opinion

- Flatpak 1.0 & GNOME Software 3.30 provide quite good UX
 - More polishing still needed
 - Try Flatpak, report bugs and suggest improvements
- **Huge benefit:** Stable platform and cross-distribution app delivery
 - I'm not limited by distribution fragmentation.
 - I can focus on the development of my app.
 - I have time for refactoring to fight technological debt.
- **Drawbacks:** There are some. Do they outweigh the benefits?

The End

I'd like to thank to all awesome people involved in the development of all open source technologies I mentioned in this presentation.

And to users of Nuvola Apps.

- Generated with [Marp](#)
- Picture [End of Track](#), Bernhard Renner, CC0 Public Domain