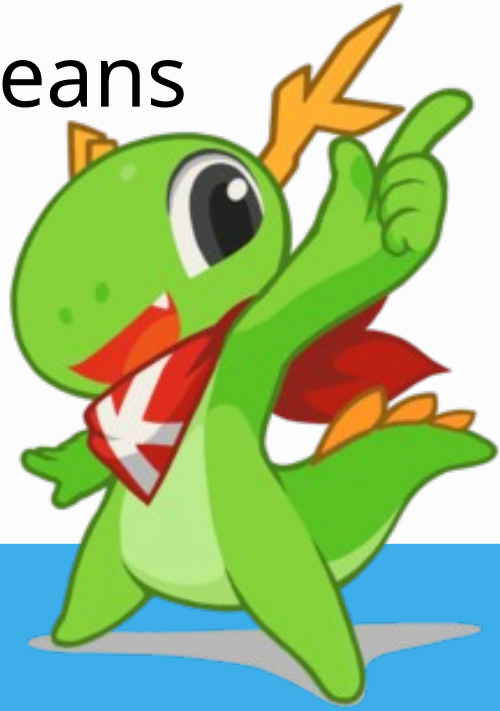
What packaging format changes means for wider KDE

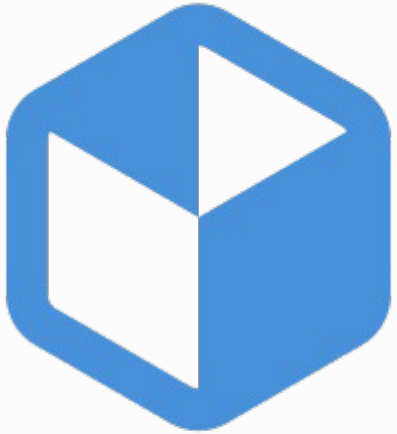


Sept 2024, Akademy

David Edmundson

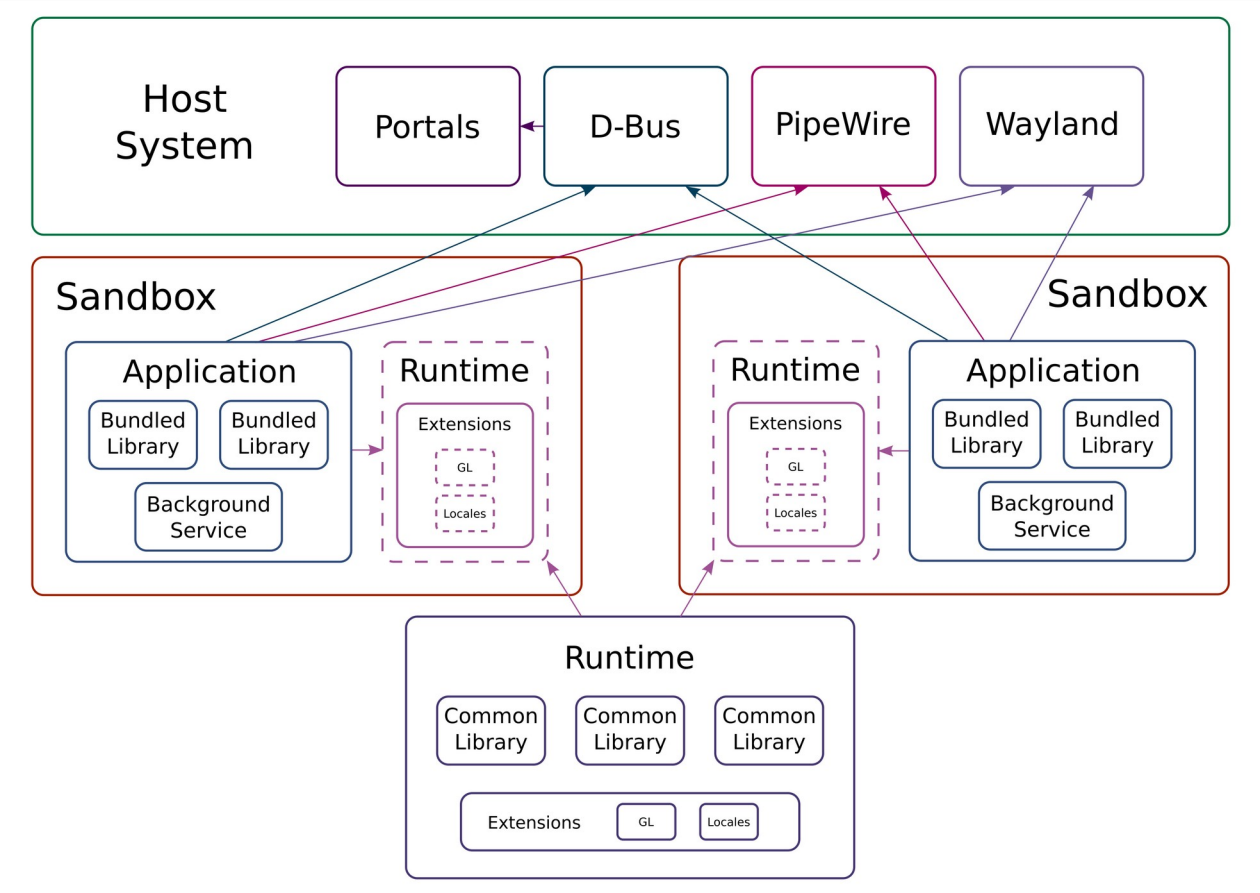
davidedmundson@kde.org
@d_ed

What packaging formats am I rambling about



Flatpak

New packaging



They’re amazing!

My moment of enlightenment: Development

My moment of enlightenment: Security

My moment of enlightenment:
We don’t have a choice!

Immutable Linuxes are coming up everywhere

fedora SILVERBLUE

ubuntu core

MicroOS

bazzite 3.0

Vanilla

What is Immutable Linux

- Read only root file system
- Atomically update the whole system at once
- Apps can only be installed via container tech

FACT!

Most Linux users will use an immutable distro within 5 years

What we need to deliver

- Our apps need to be available in Flatpaks

- The base outside apps (Plasma) needs to be small
- The base outside apps (Plasma) needs to be robust and secure

- The app SDK needs to work for developers
- Working integration between apps and Plasma

What is the state in KDE?

Making it more than an afterthought

Security Overview

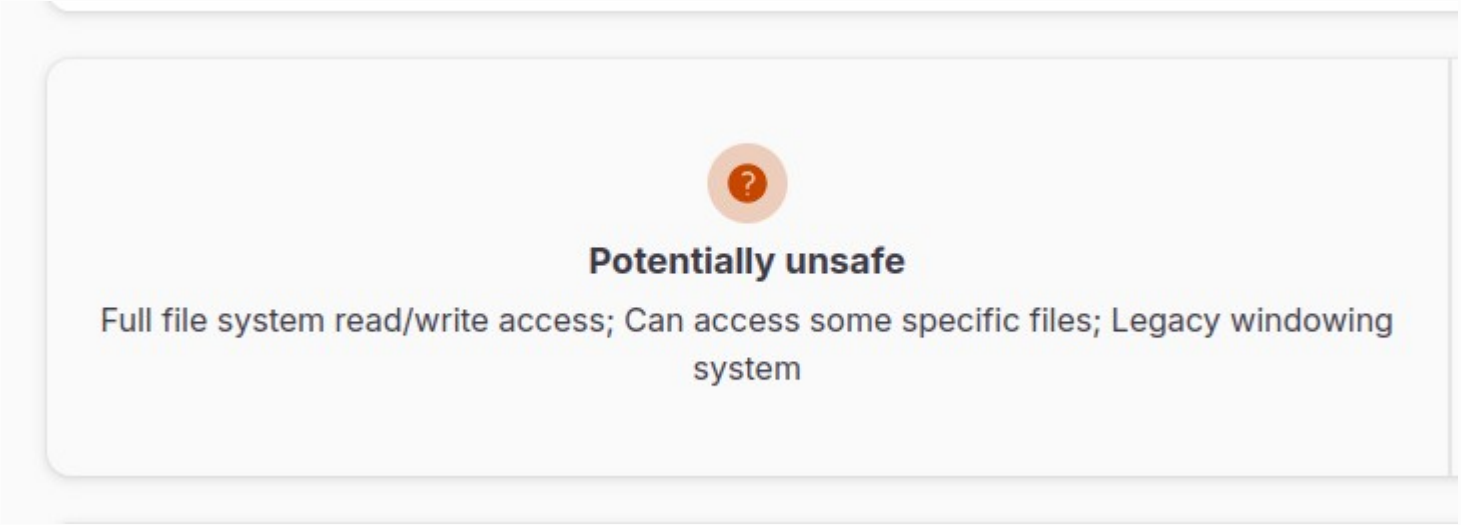
- By default:
  - No access to home directories or root file system
  - No access to devices
  - DBus traffic goes through a proxy, messages go in, but not out
  - The host only sees a handful of installed files*:
    - Single .desktop file in $XDG_DATA_DIRS/applications
    - Dbus service activation file
    - Icon

Sandboxed Apps: Runtime permissions

- Additional tasks can be obtained via the portal.
  - Open a URL in another app
  - Open/Save a specific file
  - Start a screencast
  - Etc.

Static permissions

- Somewhat a fallback
- Flathub gives scary warnings



s

FACT!

The permissions model and warnings will get stricter over time

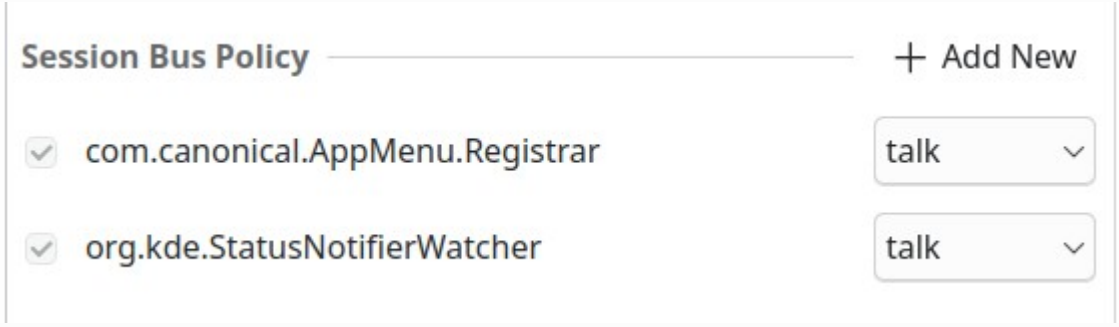
Changes in Frameworks

Examples:

Case Study: System Tray Icons

System Tray Icons

- Out the box, KstatusNotifierItem or QsystemTrayIcon fail!
- Can be fixed with a static permission



- --talk-name=org.kde.StatusNotifierWatcher
- No documentation on KStatusNotifierItem or QSystemTrayIcon mention this

Alternative fixes:

- Rename org.kde.StatusNotifierWatcher to org.freedesktop.portal.kde.StatusNotifierWatcher

- Make it a proper portal API....by talking to people

Case Study: KConfig

KConfig

- Base systems + packages on Linux distros typically only get upgraded
- With these formats the option to upgrade and downgrades is surfaced a lot more user-facing
- We do not handle this well!

Kconfig - kdeglobals

- Kconfig has a mechanism to mix application configuration and system wide configuration completely transparently loading a second config file “kdeglobals”
- In Flatpak, if we want to access global configuration it needs a completely separate explicit path…
- Making is transparent is the worst thing we can do

Case Study: KWallet

Wallets

The way it’s meant to work:

- A Portal API gives the application *A SINGLE* key which has been decprypted from the host’s password store
- This key can then be used by the app to open decrypt/encrypt it’s own passwords on disk

- Our APIs do not do any of this
- There cannot be a global password manager*

Case Study: KIO

KIO

- Does KIO workers work in a Flatpak?
- Sort of – we bundle our plugins

- Password sharing doesn’t work
- File change notification doesn’t work

KIO – Compared to GVFS

- Network file access work without the app having network access
- Apps can access* the GVFS daemons on the host and use them
- Can expose the equivalent to kio-fuse to “legacy” apps

Overall SDK Design

Overall SDK Design

- KF5 layered things based on dependencies
- We need to thing about intended usage
- If it doesn’t work in a Flatpak, it shouldn’t be in an SDK for Flatpak apps

Example – Does KDBusAddons make sense in contained apps?

Example:

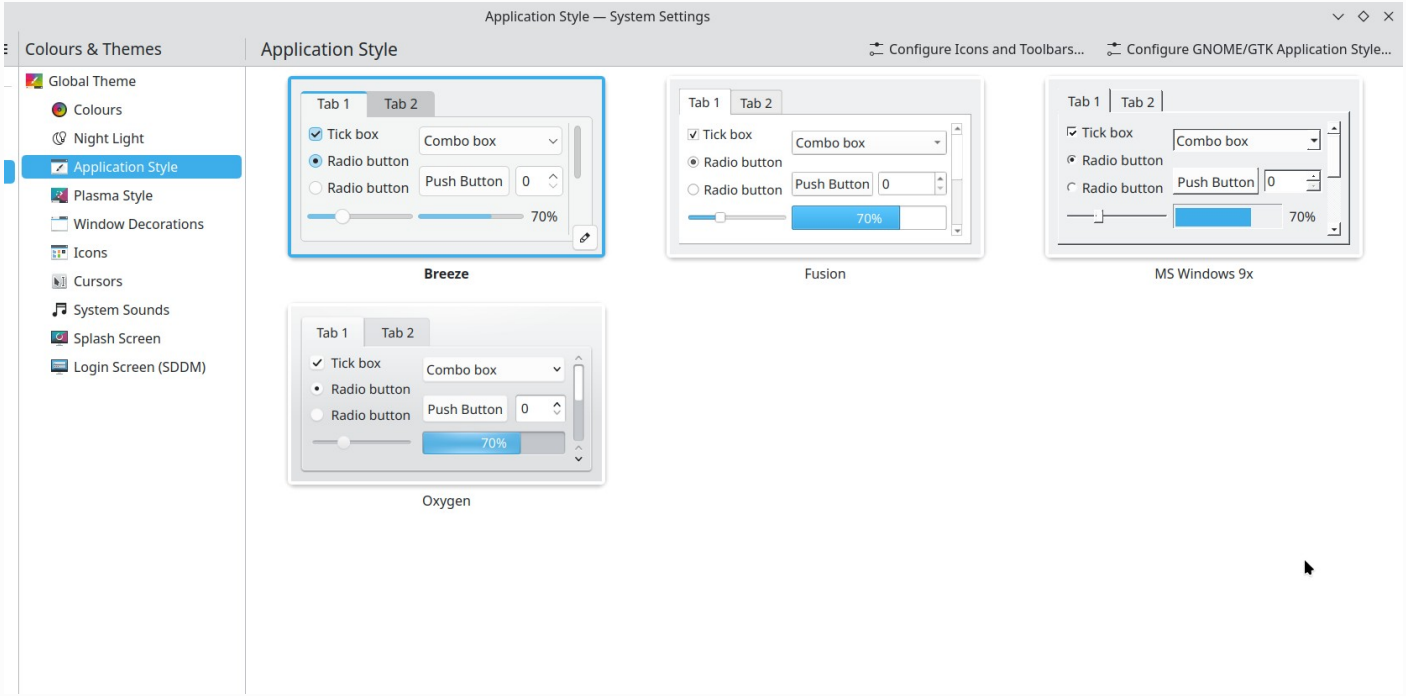
- KDBusAddons::KDBusService – YES!
- KDBusAddons::KDedModule – NO!
- KDBusAddons::KupdateLaunchEnvJob – NO!

Thinking about an SDK

- Does KAuth belong in an app SDK?
- Should Baloo be in the SDK?
- Does KPurpose work properly?
- Which other libraries should be included, and which others are niche enough to be built by just the application

Better Flaptak’s integration within Plasma

Styles


Application Style — System Settings
Colours & Themes
Application Style
Configure Icons and Toolbars...
Configure GNOME/GTK Application Style...
Global Theme
Colours
Night Light
Application Style
Plasma Style
Window Decorations
Icons
Cursors
System Sounds
Splash Screen
Login Screen (SDDM)
Tab 1
Tab 2
Tick box
Radio button
Radio button
Combo box
Push Button
0
70%
Breeze
Fusion
MS Windows 9x
Oxygen

- It works if you have (manually!) installed the matching style plugin on every KDE runtime version
- Even if it did work, it undermines the sandboxing

We need to:

- Move away from plugins to configure apps

Plasma Integration QPT

- Every configuration key we use in plasma-integration / breeze needs to have a stability promise
- Should this repo be in lock-step with Plasma releases?

Can we use container tech to Improve Plasma?

Case Study: KRunner

Krunner: A near success story!

- Used to be a plugin system, apps and third parties wrote plugins for krunner

- Ported to be external processes via Dbus and a metadata file
- Krunner then calls into the app asking for results

Krunner: The bad

- Users started shipping scripted content on GHNS, great!

  Over 50 new plugins!

- Then those scripts started calling apt-get, pacman etc. to make them work properly...

- Result is a mess.

Krunner: Why the right way doesn’t work

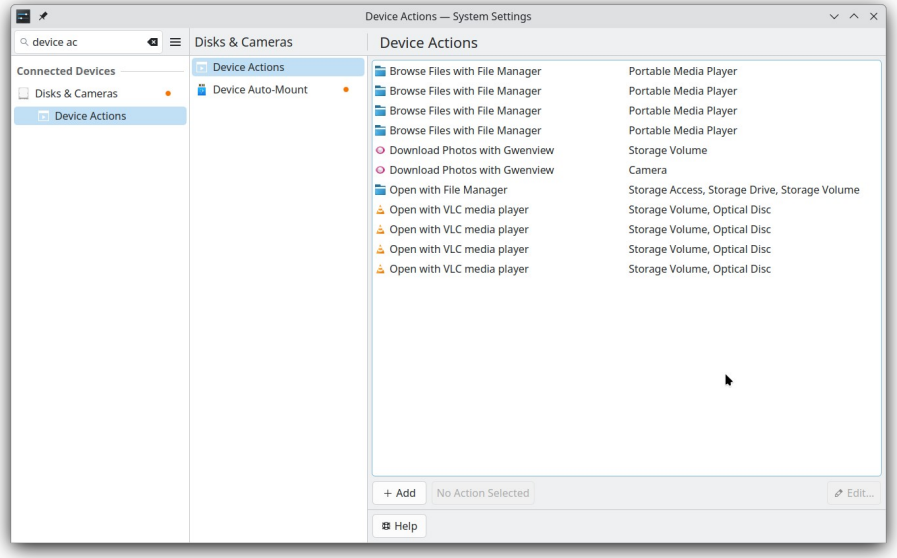
- We want the runners to be in flatpaks themselves
- Doesn’t work installed from flatpak because we rely on the desktop file exsting in /usr/share/krunner/dbusplugins!
- Krunner can’t see unexported files

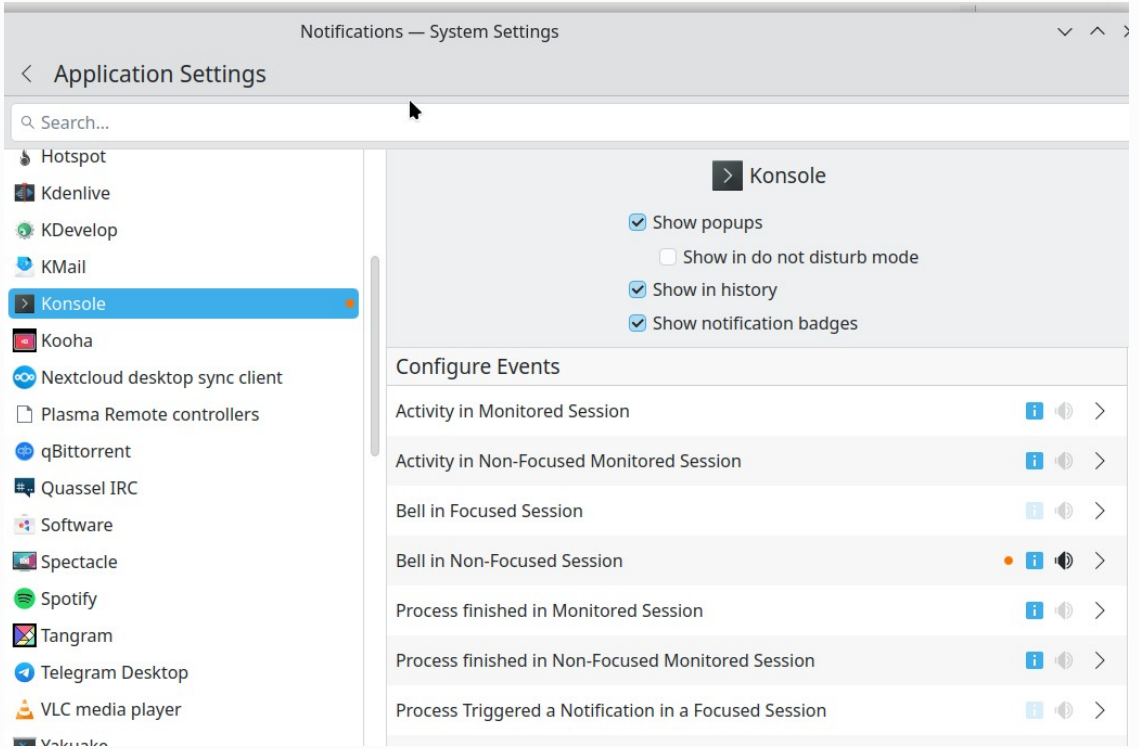
Krunner: Fixes!

- Fixing is easy – use the main application .desktop file
- (WIP branch + demo posted)

- Gnome has a similar problem with their “gnome-search”, which has an identical design.
- They patched flatpak to also export this file!

Other app extension points





Case Study: File Thumbnails

- Currently shipped as plugins that have to match the application version
- New apps in Flatpak cannot install new thumbnailers

- They run on the host system completely un-isolated

- There is work by Akseli that addresses both of these

What do we support from containers next?

- System Setting Modules
- Applets
- Wallpapers

Questions?