COLLABORA

Flatpak and your distribution

Simon McVittie

smcv@{collabora.com,debian.org}

2018-02-04

FOSDEM 18

“Introduction to Flatpak” — tl;dr edition

- A sandboxed app framework for desktop Linux
  - GUI, desktop apps, as in `/usr/share/applications`, AppStream, app stores
  - Sandboxed to mitigate badness
  - *Portals* provide user-controlled interfaces to the outside world, to keep the app useful
  - Using the same Linux kernel entry points as OS/server/daemon containers
  - Apps run on *runtimes*

Runtimes — tl;dr edition

- Give app authors a stable platform to work on
  - Update when they're ready
- A library stack with known contents and versions
  - Updated with security/micro releases
  - Not updated with incompatible changes
  - A support lifetime
- Hey, that sounds familiar

Runtimes — Distros devroom edition

- A small Linux distribution, without all the parts that app containers don't need
  - No kernel, init, boot process
  - No sysadmin things
  - No dev tools (in the version users see)
  - No `apt/dpkg/rpm/...`

What can Flatpak do for you?

Fig.1. Flatpack ready to be deployed

Photo: 51% Studios, 2012. CC-BY-SA-2.0

New software on an old distribution

- For some distributions, long term stability is a virtue
  - Debian, Ubuntu LTS, SUSE Linux Enterprise, Red Hat Enterprise Linux
- Users want predictability
  - Except for the subset that they want to be newer
- Upstreams need to choose what they target
  - New libraries: hard to install on older base
  - Old libraries: app can't benefit from a newer base

Backports

- Rebuild selected software from a newer branch for an older release
- But then you have to choose: do you want this library to be predictable, or up to date?
  - AbiWord uses GTK+, but so does all of GNOME
  - GNOME uses GLib, but so does thermald

Flatpak for backports

- Build each app against the stack its maintainer recommends
  - GNOME Builder needs latest shinies? Choose a fast-moving runtime
  - Inkscape less so? Choose a stable runtime
- Apps upgrade at their own pace (within reason)
- There's some duplication if apps choose differently
  - But perhaps less than you might think

Old software on a new distribution

- Some interesting software depends on obsolete libraries
  - GNOME 2 stack
  - SDL 1
  - Qt 3 or 4
- Distributions don't have enough resources to maintain those libraries forever
  - They were usually deprecated for a reason!

Steam games on a new distribution

- Valve aim to avoid “dependency hell” for Linux Steam games
- Steam Runtime: Ubuntu 12.04 library stack with selected backports
  - No graphics stack: use the host's
- All Linux Steam games run in the Steam Runtime to get a somewhat predictable environment
  - Currently a long `LD_LIBRARY_PATH`

Flatpak for forward-ports?

- Collabora are experimenting with running Steam games in containers
  - `flatdeb`: Runtimes from Debian/Ubuntu packages (in this case the Steam Runtime)
  - `libcapsule`: Decoupling graphics library dependencies from game dependencies
  - Future goal: Run old games in the old Steam Runtime and newer games in a less-old Steam Runtime

Problems with forward-ports

- Putting libraries in a runtime doesn't make them less unmaintained
- Who's going to fix security vulnerabilities?
  - Ubuntu 12.04 is EOL; LTS isn't **that** long
- Mitigation: sandboxing helps to protect you
- Still better than a static binary
- Still better than `wget old-library_2012_i386.deb`

Software not yet in your distro, on your distro

- We can't package **everything**
  - ... although Debian tries
- The perfect is the enemy of the good?

At our best, “maintainers matter”

- Sometimes we do better than upstreams
  - Security updates
  - Avoiding regressions
  - Being open source
  - Not infringing copyright
- Sometimes a “gatekeeper” role is good: protecting our users from defective software

At our worst, maintainers get in the way

- Sometimes we're not adding any value to what upstream do
- Sometimes a “gatekeeper” role is harmful: restricting our users' choices for no good reason

Software outside your distro's scope, on your distro

- I said Debian tries to package everything, but that was a lie
- Some distributions don't want to touch proprietary software
  - Some users need proprietary software anyway
- We can't debug it, fix it, or know that it has your best interests in mind, so we should protect the rest of the system from it
  - Sandboxing!

Software outside your distro's scope, on your distro

- Not just about freedom
- Some software is obscure (but maybe you need it)
- Some software is awful (but maybe you need it)
- Even if your distro won't package it, **someone** can (maybe you)
  - But maybe they'll get it wrong
  - Sandboxing mitigates accidents too

Flathub — a reference app-store for Flatpak

- Reference runtimes: freedesktop.org, GNOME, KDE
- Upstream software, provided by upstreams
- Proprietary software, packaged by its author or the community
  - Tagged with license information, so you can avoid it if you want to
  - Downloaded on-demand if redistribution isn't allowed (*extra data* scripts)

What can you do for Flatpak?

Fig.2. A curated distribution of flatpack ready for download

Photo: themightycondorman, 2012. CC-BY-2.0

Ship Flatpak (and friends)

- All this won't do your users much good if they can't install Flatpak
- Portals won't work very well if your users can't install them
- Follow upstream stable branches, or backport the latest to older distribution branches
- Other components are picking up Flatpak/portal support: dbus-daemon (WIP), dconf (future)

Runtimes — this is Unix, I know this

- A stable library stack
- Versions chosen to work together
- Security updates applied
- Destabilising changes avoided
- Distributions have had a lot of practice at this

License compliance

- Distributions have had a lot of practice at this too
- `flatpak-builder` knows how to bundle corresponding source code
- Distributions can push back to upstreams, so everyone wins

Apps — curation and QA

- At our best, we're domain experts on leaf packages
  - We fix their bugs even if upstreams don't
  - We fix their security vulnerabilities even if upstreams don't
- Distributions: a ready-made source of high-quality Flatpak apps?
  - Fedora think so!

Fixing apps

- Distributions fix apps where upstreams won't or can't
- Don't keep it to yourself
  - Send bug fixes upstream
  - Send AppStream upstream (try saying that 10 times fast)
- If your upstream is uncooperative, try coordinating with other downstreams

Where shouldn't I use Flatpak?

Fig.3. Flatpack is unsuitable for this use-case

Photo: Kinshuk Sunil, 2009. CC-BY-2.0

The OS layer

- Flatpak doesn't try to package OS-level services
  - Kernel, initrd and boot
  - init and daemons
  - Sysadmin and privileged tools
- Alternatives include:
  - Traditional package managers (dpkg, rpm)
  - libostree

The desktop platform

- Flatpak doesn't try to package non-app software
  - Desktop shell (GNOME Shell, Plasma)
  - Compositor, window manager, other chrome (but not Chrome)
  - Per-user services: dbus, dconf, portals, Flatpak
- Use the same mechanisms as the OS layer

Unix CLI tools

- Flatpak isn't really intended for CLI tools
  - coreutils, sed, tar, rsync, ssh
  - gcc, make, strace
- Alternatives include:
  - Put it in the OS layer
  - Put it in a Platform runtime
  - Put it in an SDK runtime

Servers

- Servers aren't generally apps
  - Web, mail, databases, continuous integration, ...
- Alternatives include:
  - Put them in the OS layer
  - Use a more service-oriented container technology: Docker, LXC, systemd-nspawn
  - Put them in the cloud and hope they're somebody else's problem :-)

COLLABORA

Flatpak and your distribution

FOSDEM 18

col.la/fosdem18flatpak

Questions?

We're hiring: col.la/careers