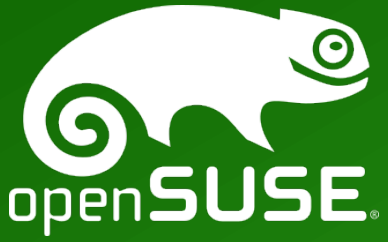


What could go wrong? Me! I was...

Containerised Applications are the way

Richard Brown

MicroOS Release Engineer

rbrown@opensuse.org

@sysrich@fosstodon.org

About Myself

Richard Brown

MicroOS Release Engineer
rbrown@opensuse.org

openSUSE contributor since it began
SUSE employee since 2013

Passionate advocate of rolling releases, creator of **openSUSE MicroOS Desktop**

Former Customer (Systems Manager), QA Engineer, openSUSE Board Member & openSUSE Chairperson

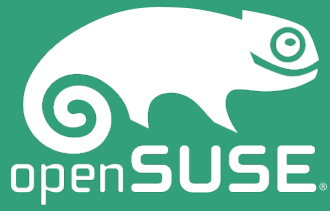
Currently Linux Distribution Engineer in SUSE's Early Adoption Team release managing two rolling distributions

openSUSE MicroOS – Single Purpose Self Administering OS
openSUSE Tumbleweed – "Traditional" General Purpose OS

Consulting on various SUSE Linux Enterprise Micro (SLE Micro) customer projects

Photographer in spare time

A long time ago. . .
in a room not very far away



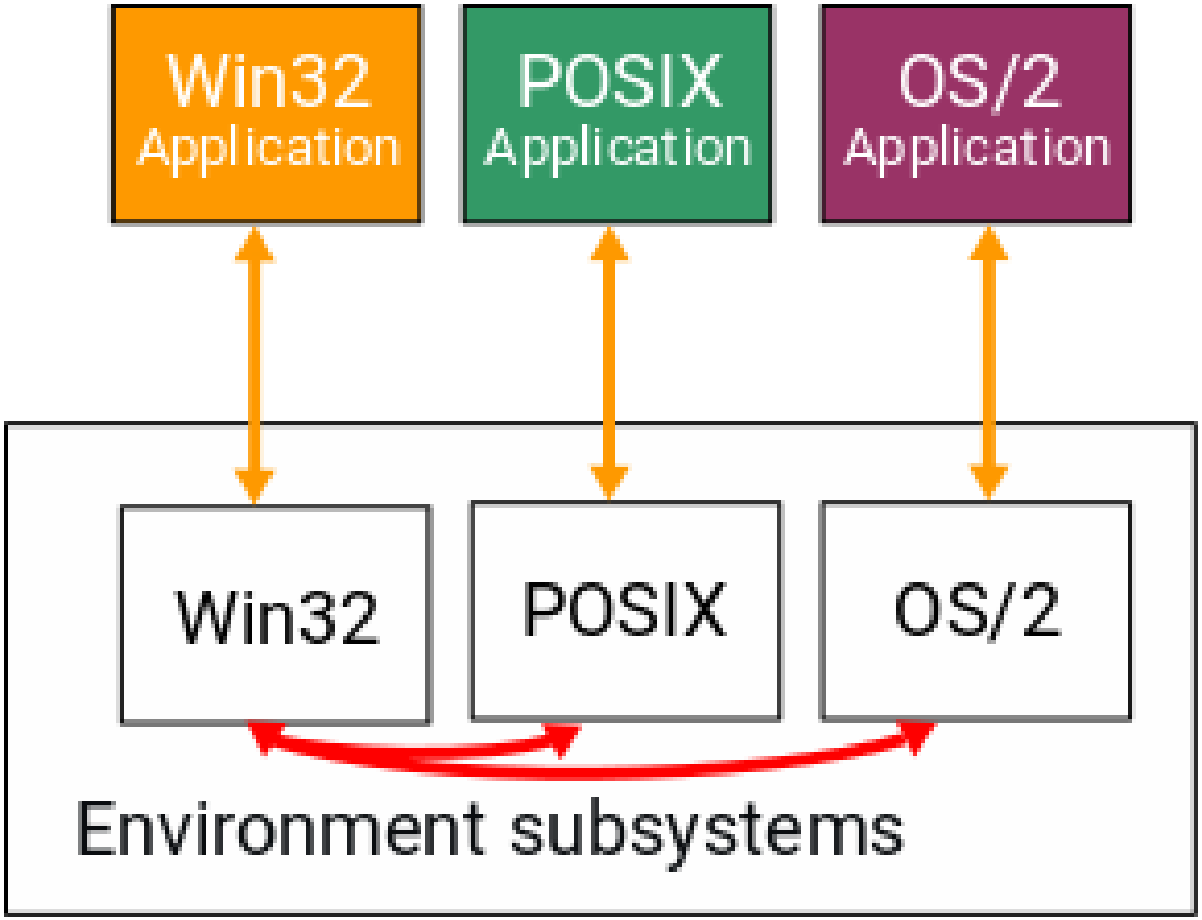
Resurrecting dinosaurs, what can possibly go wrong?

How Containerised Apps could eat our users.

Richard Brown
openSUSE Chairman
rbrown@opensuse.org

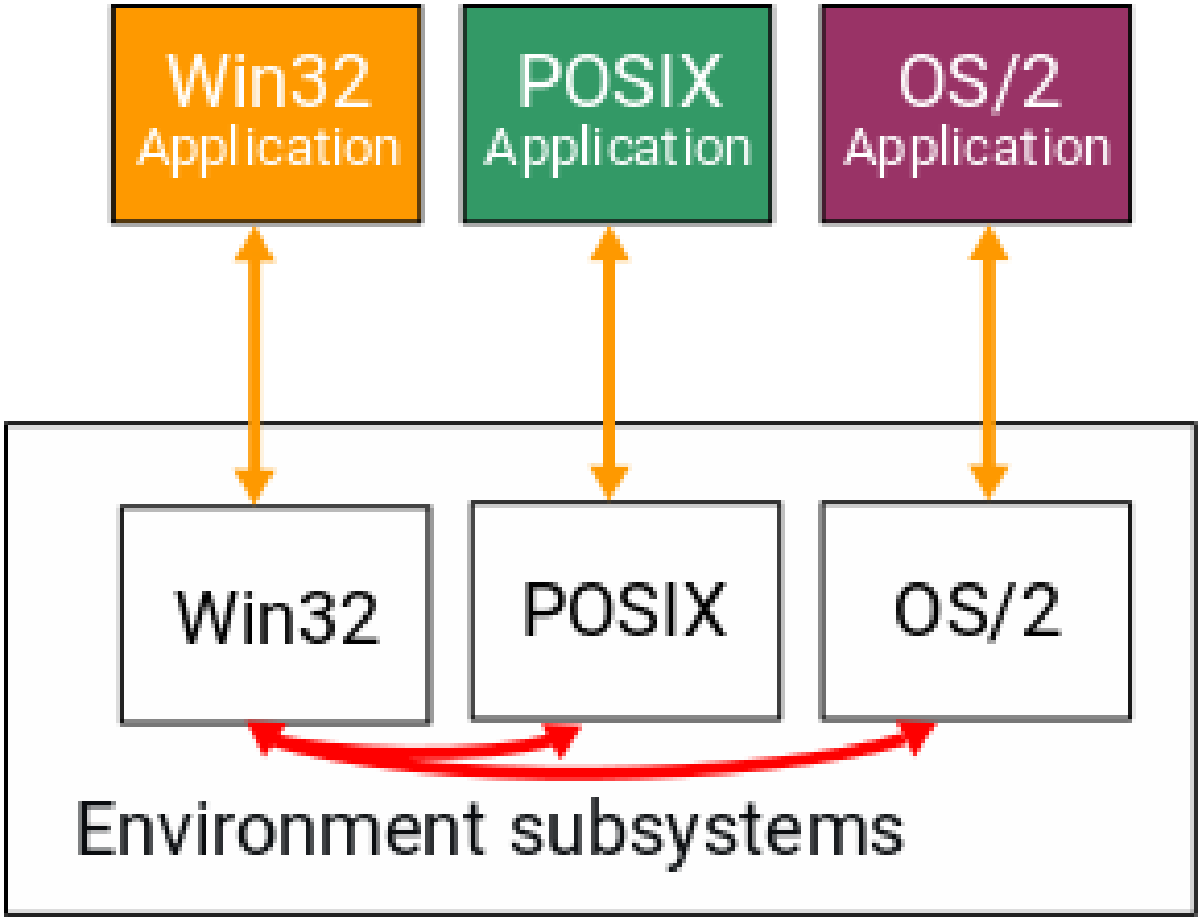
“Those who cannot remember the past are condemned to repeat it”

- George Santayana

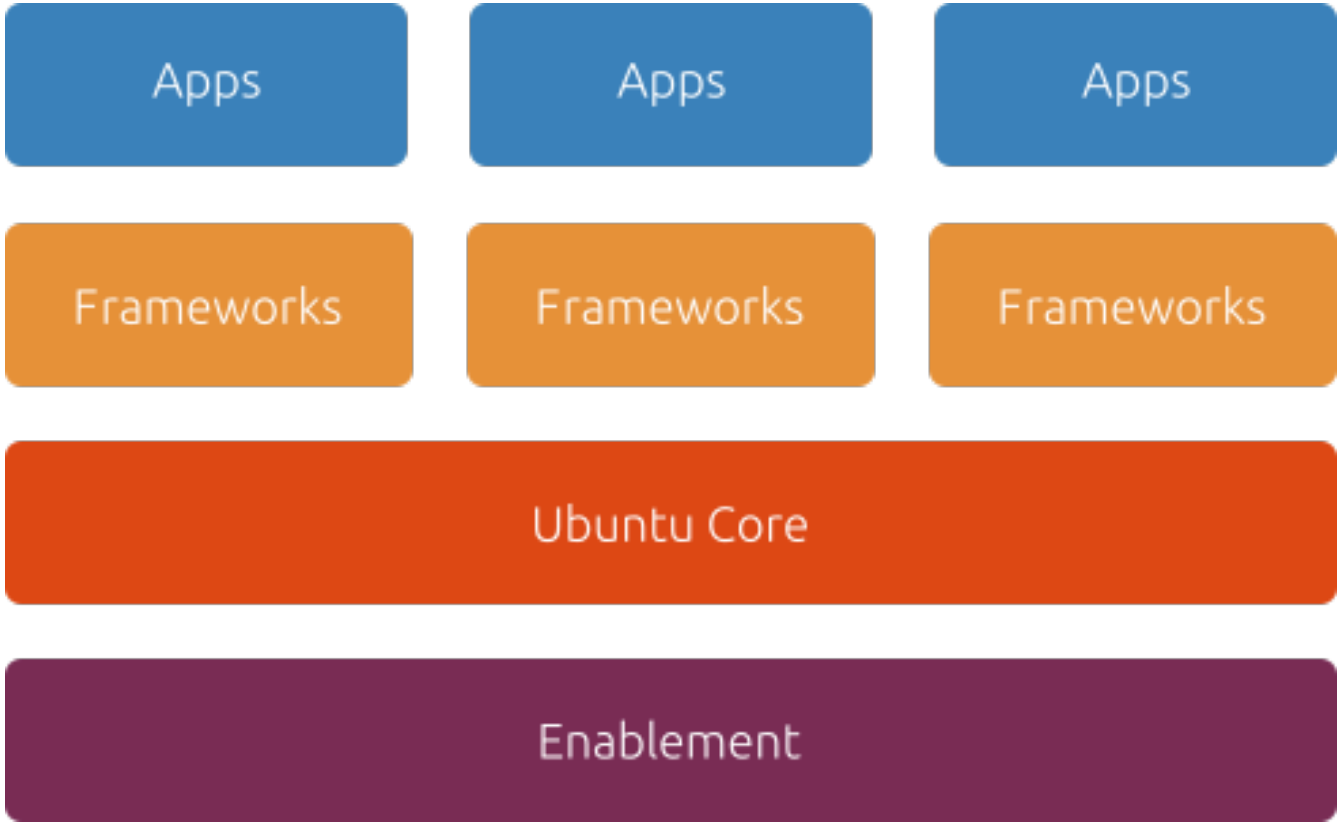


Problem Solved? Right?

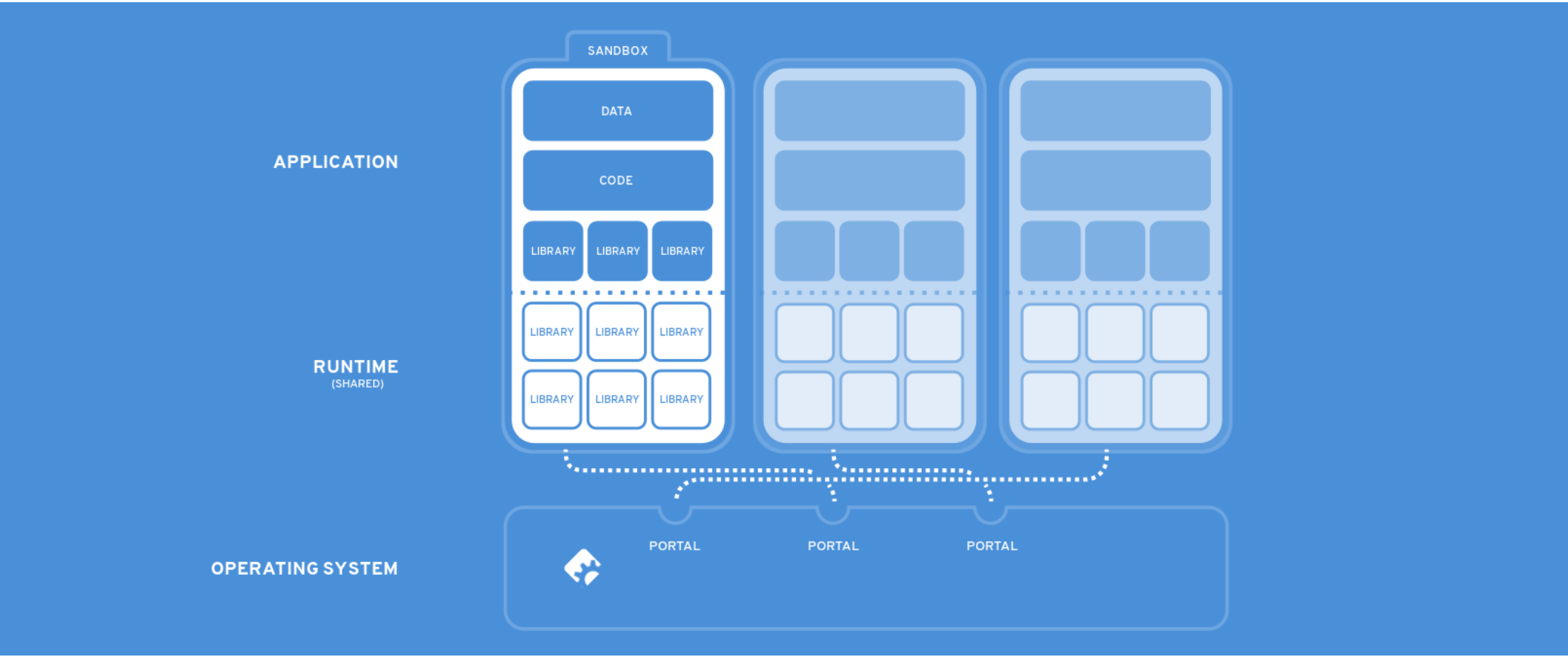
- Security nightmare
  - Security relevant DLLs lurking in countless application folders
- Maintenance nightmare
  - How are we going to update our app? Oh we'll ship an updater!
- Legal nightmare
  - Can we legally redistribute all the DLLs we need to?
- Storage vendor dream
  - More disk consumption, everyone buying bigger disks!



Compatibility & Portability



Compatibility & Portability



History Repeating?

- Security nightmare?
  - Security relevant libs lurking in countless application bundles
- Maintenance nightmare?
  - How are we going to update our app and every single lib?
- Legal nightmare?
  - Can we legally redistribute all the libs we need to?
- Storage vendor dream
  - More disk consumption, everyone buying bigger disks!

“With Great Power…”

“… Comes Great Responsibilities”

- AppImage/FlatPak/Snappy are tools that enable App Developers to directly distribute software without the ‘need’ for Distributions
- Therefore, they must adopt the responsibilities which come with being a distributor of software

AppImage

WHAT PEOPLE THINK

Richard Brown

"I now love AppImage. I really love AppImage."

OBS now builds AppImages

- OBS built AppImages make use of OBS's strengths in
  - Auditing
  - Update & Dependency Tracking
  - License Compliance
  - Build Hosting
- All without impeding AppImages strengths in getting the software in the hands of users

Dear Snappy & Flatpak

- You are falling behind
- AppImage has a smoother build story, a stronger compliance story, and a more straight forward user experience
- Most importantly: They ENGAGE and WORK WITH OTHERS
- Be more like AppImage
- openSUSE / OBS / openQA and more are all here to help

Problems Remain

- Dependency Hell still on the Horizon
  - Assumptions are still being made about what a base system must provide containerised apps
  - Let’s all get together, distros & new app formats, and discuss & design standards/common practice
  - A common understanding of what distros provide will make life easier for App developers, users, and distributions

Problems Remain

- Security / Sandboxing / App Isolation is still a mess
  - Snap requires not-yet-upstreamed AppArmor patches
  - Flatpak – bubblewrap, too desktop orientated?
  - AppImage – firejail/nothing
- Let’s clear this up – AppArmor all the way?

AppImages


AppImage
Documentation
Source Code
Wiki
Community
Star
7,724
Linux apps that run anywhere
"As a user, I want to download an application from the original author, and run it on my Linux desktop system just like I would do with a Windows or Mac application."
"As an application author, I want to provide packages for Linux desktop systems, without the need to get it 'into' a distribution and without having to build for gazillions of different distributions."

AppImages

They don't

AppImages

They don't

You can't

AppImages

They don't

You can't

The only distro you need to get 'into' is the one you now have to build

AppImages – The Unportable App Format

I get some errors related to something called “FUSE”

AppImages require a Linux technology called *Filesystem in Userspace* (or short *FUSE*). The majority of systems ships with a working FUSE setup. However, sometimes, it doesn't quite work. This section explains a few solutions that fix the most frequently reported problems.

AppImages – The Unportable App Format

I have issues with Electron-based AppImages and their sandboxing

AppImages based on Electron require the kernel to be configured in a certain way to allow for its sandboxing to work as intended (specifically, the kernel needs to be allowed to provide "unprivileged namespaces"). Many distributions come with this configured out of the box (like Ubuntu for instance), but some do not (for example Debian).

AppImages – Avoid Distros by Building Your Own

not required.

2. **Gather suitable binaries of all dependencies** that are not part of the base operating systems you are targeting. For example, if you are targeting Ubuntu, Fedora, and openSUSE, then you need to gather all libraries and other dependencies that your app requires to run that are not part of Ubuntu, Fedora, and openSUSE.
3. **Create a working AppDir** from your binaries. A working AppImage runs your app when you execute its AppRun file

AppImages – Build Your Own Distro & make it OLD

Binaries compiled on old enough base system

The ingredients used in your AppImage should not be built on a more recent base system than the oldest base system your AppImage is intended to run on.

AppImages – Install like a Mac

How to run an AppImage

It's quite simple to run AppImages. All you have to do is download them, make them executable and run them. This can either be done using the GUI or via the command line.

Using the GUI

1. Open your file manager and browse to the location of the AppImage
2. Right-click on the AppImage and click the 'Properties' entry
3. Switch to the Permissions tab and
4. Click the 'Allow executing file as program' checkbox if you are using a Nautilus-based file manager (Files, Nemo, Caja), or click the 'Is executable' checkbox if you are using Dolphin, or change the 'Execute' drop down list to 'Anyone' if you are using PCManFM
5. Close the dialog
6. Double-click on the AppImage file to run

AppImages – Install like a Mac, except not

Install apps

On your Mac, do any of the following:

- *For apps downloaded from the internet:* In the Downloads folder, double-click the disk image or package file (looks like an open box). If the provided installer doesn't open automatically, open it, then follow the onscreen instructions.

  Note: If you get a warning dialog about installing an app from an unidentified developer, see Open a Mac app from an unidentified developer.

- *For apps on a disc:* Insert the disc into the optical drive on your Mac or connected to your Mac.

To reinstall apps you downloaded from the App Store, see Install purchases from the App Store.

In Short

AppImages fail in everything they set out to do

They also fail to do anything to address dependency, licensing, security, and maintenance issues that any responsible distributor must consider

I argue that AppImages make many of the recurring issues with FOSS App Delivery even worse

Please do not use them

WHAT PEOPLE THINK

Richard Brown

"I now love AppImage. I really love AppImage."

Snap

Snap

Despite my reservations, 2017 was actually promising

- Active collaboration between Canonical and other distributions trying to get snapd fully supported
- Able to run your own snap store
- “Upstream First” mentality
- Snap shared many technical benefits of Flatpak, plus a strong non-graphical App story

Snap confinement

snapd does not support confinement on most non-Ubuntu distributions

Snapd STILL requires out-of-tree apparmor patches for strict confinement

snapd

diddledani Aug '20

Ubuntu is currently the *only* distribution, as far as I can tell, that supports strict confinement out-of-the-box. This is due to the fact that snapd relies on out-of-tree apparmor patches that are only applied in Ubuntu distributed kernels. Specifically, this patch 31 is required to get snapd to support strict confinement, which has still not been upstreamed. Why, after 3 years of this being known (Snapd vs upstream kernel vs apparmor 8) are we still in this situation?

From reading about the upstreaming process for this particular patch it seems that it was not upstreamed on purpose due to it re-enabling legacy features. In that case then why are we relying on functionality that is not ever going to be upstreamed and claiming that we are fully functional on many distributions when that claim can only ever apply to Ubuntu?

6

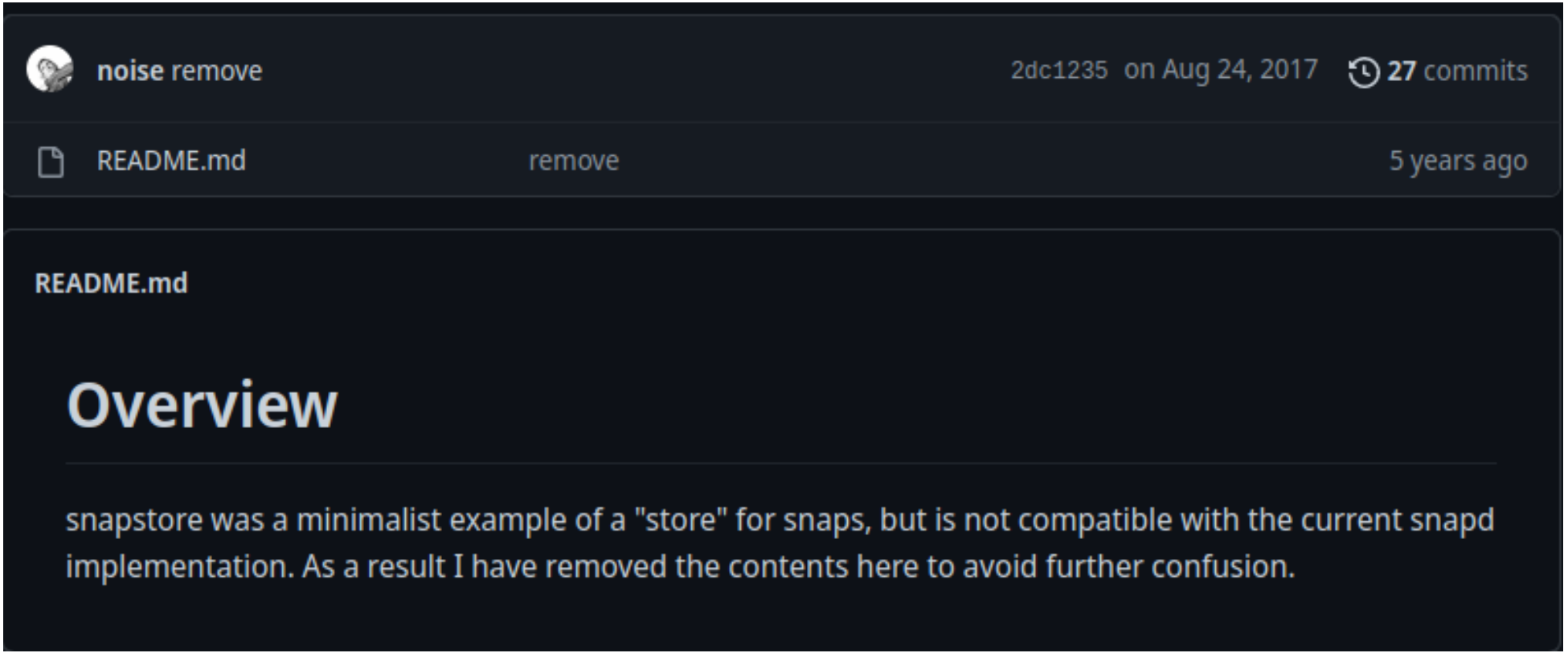
created	last reply	13	1.6k	8	33	9
Aug '20	25d	replies	views	users	likes	links

alexmurray policy-reviewers Jan 3

The changes to vendor apparmor into snapd have been merged but subsequently reverted twice now - unfortunately the complexity involved here is larger than initially realised - so that is still ongoing but I hope to have another stab at it again in near future.

As far as getting the various AppArmor kernel features merged upstream, that work is still ongoing as well. With any luck both should hopefully happen within the first few months of this year 🤞

Snap store


noise remove
2dc1235 on Aug 24, 2017
27 commits
README.md
remove
5 years ago
README.md
Overview
snapstore was a minimalist example of a "store" for snaps, but is not compatible with the current snapd implementation. As a result I have removed the contents here to avoid further confusion.

Snap store

Brand stores

For larger projects and ISVs, it is often a requirement to publish snaps using a brand account to a brand-specific store.

A brand store allows vendors running Ubuntu Core and snap-based devices to control exactly what snaps are available, and when. It can inherit selected packages from other snap stores, and host a set of snaps specific to a brand and device models, and be either open to all developers or a specific list.

SmartStart

$45,000

Let our Linux experts help you ensure the success of your application packaging on Ubuntu.

Enjoy snap store services, device fees and support for one year for up to 1,000 devices; deployment services (excluding proxy deployments); and either the packaging of up to three application snaps (subject to approval) or equivalent consulting hours for clients that wish to do the packaging themselves with help from our top-notch Linux experts.

Snap creation service

$25,000

Get three critical apps containerised as snaps by Canonical experts, using best practices and focused on your business and technical needs.

Get access to a workshop to review requirements dependencies, outline optimal containerisation strategy, snap containerisation and publishing on your own App Store.

IoT App Store setup

From $5,000

Get your own dedicated app store set up, including role-based access controls, complete control of application versions, over-the-air (OTA) updates, and controlled rollouts.

Choose from either our hands-off SaaS edition or our fully air-gapped on-prem edition to suit your business needs.

IoT app store fees not included.

Rapid Prototyping Services

From $10,000

Get a rapid Proof-Of-Concept (POC) image based on the Ubuntu kernel, which includes necessary drivers to initialise and run your hardware and chosen peripherals.

Includes a workshop to review requirements, image customisation and generation, as well as the initial deployment of the image on one of your devices.

Snap on openSUSE

snapd is the ONLY package I am aware of to ever fail MULTIPLE security audits

Has NEVER been included in any official SUSE/openSUSE distributions as a result

Last security bug closed in December 2019 due to NORESPONSE

In 2023

- Canonical's "Upstream First" effort appears to have stalled
- No apparent effort to make snapd available on other distributions
- No open source delivery option for snap Apps
- Not a viable alternative to Flatpak unless you only use Ubuntu, trust Canonical, and like to give them money to distribute your stuff

Flatpak

One more thing

Rolling Releases for Everyone?

- To get Applications in the hands of users fast, what model beats a rolling distribution?
- Users can be guaranteed an integrated “built together” experience
- Security/Maintenance burdens less broadly distributed, fewer points of failure, Devs don’t need to be security engineers
- “It just works” can be reached with good tools – OBS & openQA

openSUSE MicroOS

openSUSE MicroOS is both predictable & immutable. It cannot be altered during runtime.

MicroOS is reliable with automated updates and automated recovery from faulty updates.

MicroOS is small with only what is needed to run it's "one job". Applications/Services are expected to be Containerised or Sandboxed.

MicroOS Desktop

openSUSE MicroOS Desktop is MicroOS where the ”one job” is running as a Desktop.

MicroOS Desktop provides only a minimal base system with a Desktop Environment and Basic Configuration Tools ONLY.

All Applications, Browsers, etc are provided by FlatPaks from FlatHub.

MicroOS Desktop

openSUSE MicroOS Desktop is MicroOS where the ”one job” is running as a Desktop.

MicroOS Desktop provides only a minimal base system with a Desktop Environment and Basic Configuration Tools ONLY.

All Applications, Browsers, etc are provided by FlatPaks from FlatHub.


peacock

What happened?

Invited to GUADEC 2017

- I gave an even meaner, more-flatpak-hating version of the FOSDEM talk
- Flatpak devs listened
- They challenged some of my views
- They accepted others
- Flatpak changed

Responsibility is the key

Regardless of the toolset used, Containerised Apps require App Developers/Packagers to think MORE (not less) like a Distribution builder

- Dependencies are their problem now
- Licenses are their problem now
- Maintenance is their problem now
- Security is their problem now

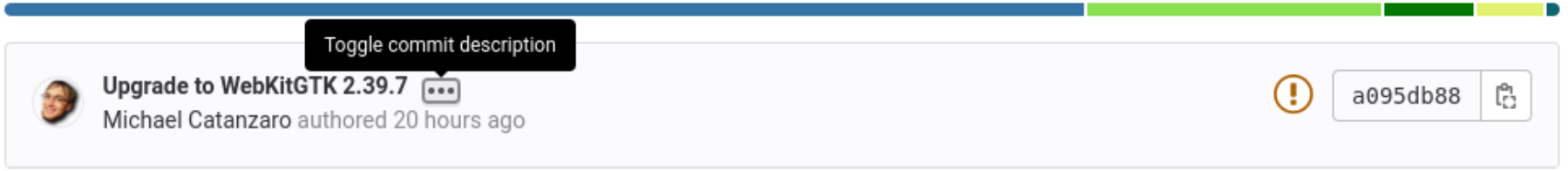
Flatpak dependencies

Flatpaks has always handled App dependencies well with the concept of ‘runtimes’

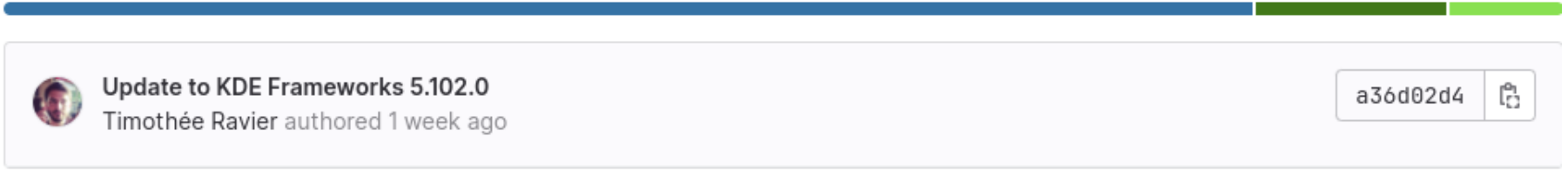
- Effectively “mini-distributions” surrounding key Linux App Ecosystems
  - GNOME
  - KDE
  - elementary
  - Freedesktop (aka everything else)

Runtimes

The GNOME Build Metadata repository is where the GNOME release team manages build metadata for building the GNOME software stack.

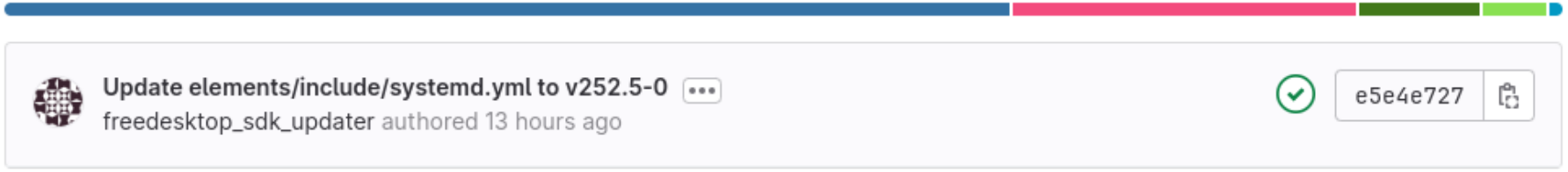


Files to generate a repository with org.kde.Platform and Sdk



A minimal Linux runtime

pipeline passed website online docker pulls 1.8k Follow



Flatpak client dependencies

Flatpak client is relatively easy to integrate on any Linux distribution with no egregious requirements

- bubblewrap
- ostree
- xdg-dbus-proxy
- xdg-desktop-portal

Each relatively self-contained with few/no install-time dependencies

Flathub licensing

All Flatpaks on Flathub must either

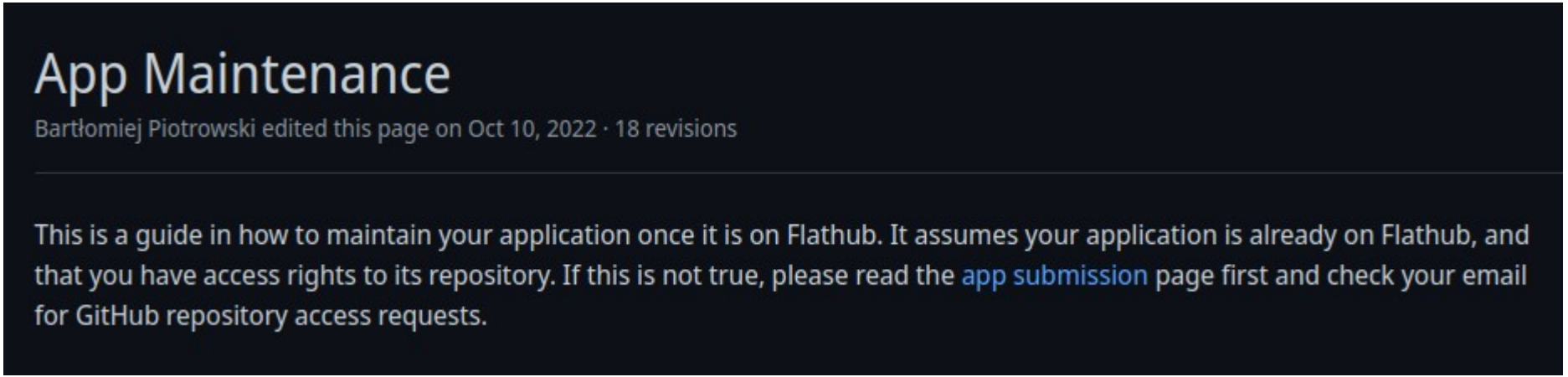
- Be openly licensed in a way that allows legal redistribution

or

- Download non-redistributable data at install time, which is scanned, monitored, and checked by the Flathub team

As good as/better than most distributions

Flathub maintenance



App Maintenance

Bartłomiej Piotrowski edited this page on Oct 10, 2022 · 18 revisions

This is a guide in how to maintain your application once it is on Flathub. It assumes your application is already on Flathub, and that you have access rights to its repository. If this is not true, please read the app submission page first and check your email for GitHub repository access requests.

- Flatpak specific patches strongly discouraged
- Robust built-test-publish workflow

As good as/better than most distributions

Flatpak security

As of 2023 Flatpak remains the ONLY Desktop containerised app platform that is sandboxed by default everywhere

Portals have proven themselves to be a secure-enough and expandable-enough solution

Flatpak CVE’s are rare and fixed quickly (last in Feb 2022, Medium CVSSv2/v3 score)

MicroOS Desktop – The Story So Far

Adopted Flatpaks from Flathub by default since earliest experiments in November 2017

Considered implementing own Flatpaks and/or Fedora-like filtered view

Opted for trusting Flathub first and waiting for problems to arise

It’s now 2023

We’re still waiting

Final Thoughts

- Flatpaks are ready for primetime
- Desktop Linux distributions don’t need to package everything any more
- Narrow the scope of your distributions, save yourself time, energy, infrastructure and everything else
- **Contribute to Flatpak and encourage more App developers to publish on Flathub**


openSUSE®