The Future of Linux Application Distribution

OSTree & Flatpak

Mario Sánchez Prada <mario@endlessm.com>

Samsung Research UK. Staines, 2017 March 16th

Who am I?

- Computer Science Engineer by the *University of Coruña*
- Open Source developer, *GNOME* Foundation member
- Previously worked at *Igalia* and *Samsung Research UK*
- Currently at *Endless Computers*, working in the *Desktop* team focused on the development of the core platform
- Your neighbour in Staines-upon-Thames since 2013

OSTree

What is OSTree?

- Git-like system for complete & bootable filesystems
- Disk efficient: de-duplication via SHA256SUM hashes, check outs files from the object store via hard links
- Network efficient: static deltas, summary file
- Reliable updates & rollback: atomicity, no inconsistencies
- Safe: GPG verification for commits and summary file
- Introspectable library and command line tools

Multiple use cases: OS deployment, efficient OTA updates, continuous integration & QA, bundled applications...

Atomic & incremental upgrades

- Git-like fetching via HTTP: simple setup
- Incremental downloads of objects, using pre-generated static deltas when available
- Automatic verification of fecthed objects and deltas
- Automatic creation of new deployments (+ 3-way merge)
- Atomic swapping of boot configurations via symlinks

```
$ ls -l /ostree/
total 12K
lrwxrwxrwx 1 root root    8 Oct  4 16:55 boot.0 -> boot.0.1
drwxr-xr-x 3 root root 4.0K Oct  4 16:55 boot.0.1
drwxr-xr-x 3 root root 4.0K Oct  4 16:55 deploy
drwxr-xr-x 7 root root 4.0K Mar 12 12:59 repo
```

Some internal details

Anatomy of an OSTree repository:

- Types of repositories: bare, bare-user, archive-z2
- Objects (commits, dirtree, dirmeta, content) + refs
- The summary file

OSTree deployments:

- Multiple deployments per OS, parallel installable
- Shared stateful data among deployments (/etc, /var)

Comparison with other systems

OSTree vs APT/RPM

- Deploying full filesystem VS partial ones
- Truly atomic VS potentially broken intermediate stages
- No dependencies hell, no postinst/postrm hooks...

OSTree vs image replication (flashing)

- Predictable like flashing, but much more efficient
- Only 2 persistent directories supported: /etc & /var
- Works on top of any filesystems supporting hard links
- Supports installing different versions of the OS in parallel

Who is using OSTree?

- Atomic project (Fedora, CentOS)
- GNOME Continuous
- Qt OTA updates
- Automotive Grade Linux (AGL)
- Endless OS
- Flatpak

Flatpak

What is Flatpak?

- A new way of distributing applications in Linux
- Sits on top of OSTree and *bubblewrap* (chroot on steroids)
- Allows having both user and system-wide installations
- Cross-platform by design: **runtimes** and **applications**
- Reliable and secure: GPG signatures, sanboxing, portals...
- Open Source project. Started by Red Hat, contributions from Endless, Collabora, Codethink, Intel, Kinvolk, Solus...

Similar in some ways to Docker, but with the focus on end user applications instead of for containerized system-wide services.

A brief note on bubblewrap

- Allows running sandboxed applications in chroot-like environments as an **unprivileged user**
- Creates a mount namespace with `/` on a `tmpfs`
- Uses `PR_SET_NO_NEW_PRIVS` when cloning the process to limit what the binary can do after dropping privileges
- Implements a subset of the Kernel's user namespaces feature to isolate processes
- More namespaces: `CLONE_NEWUSER`, `CLONE_NEWIPC`, `CLONE_NEWPID`, `CLONE_NEWNET`, `CLONE_NEWUTS`
- Allows passing a list of *seccomp* filters to limit *syscalls*

Bubblewrap example

```
[fedoravm ~]$ bwrap --ro-bind /usr /usr --ro-bind /etc/resolv.conf /etc/resolv.conf \
       --symlink usr/lib /lib --symlink usr/lib64 /lib64 --symlink usr/bin /bin \
       --dir /tmp --proc /proc --dev /dev \
       --unshare-pid --unshare-net \
       --chdir / \
       /bin/sh

sh-4.3$ ls /
bin  dev  etc  lib  lib64  proc  tmp  usr

sh-4.3$ ls /dev/
console  full  null  ptmx  pts random shm  stderr  stdin  stdout  tty  urandom  zero

sh-4.3$ ls -l /etc/
total 4
-rw-r--r-- 1 65534 65534 53 Mar 14 00:46 resolv.conf

sh-4.3$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

sh-4.3$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
1000         1  0.0  0.0  15472   160 ?        S    01:28   0:00 bwrap --ro-bind /usr /usr --
1000         2  0.0  0.1 122136  3608 ?        S    01:28   0:00 /bin/sh
1000         8  0.0  0.1 150020  3544 ?        R+   01:29   0:00 ps aux
```

Anatomy of a Flatpak Runtime

```
$ tree -L 3 /var/lib/flatpak/runtime/org.gnome.Platform/x86_64/3.22/active/
|-- deploy
|-- files
|  |-- bin
|  |  |-- [...]
|  |  |-- basename
|  |  |-- bash
|  |  |-- [...]
|  |-- etc
|  |  |-- [...]
|  |  |-- ca-certificates.conf
|  |  |-- dbus-1
|  |  |-- [...]
|  |-- lib
|  |  |-- [...]
|  |  |-- libglib-2.0.so.0.5000.2
|  |  |-- libGL.so -> libGL.so.1.0.0
|  |  |-- [...]
|  |-- lib64
|  |  `-- ld-linux-x86-64.so.2 -> /usr/lib/ld-linux-x86-64.so.2
|  |-- [...]
|  |-- manifest-base-1.json
|  |-- manifest.json
|  |-- sbin -> bin
|  |-- share
|  |  |-- [...]
|  |  |-- applications
|  |  |-- [...]
|  `-- var
|      |-- cache
|      |-- lib
|      `-- run
`-- metadata
```

Anatomy of a Flatpak Application

```
$ tree -L 3 /var/lib/flatpak/app/org.gnome.Todo/current/active/
/var/lib/flatpak/app/org.gnome.Todo/current/active/
|-- deploy
|-- export
|   `-- share
|       |-- applications
|       |-- dbus-1
|       `-- icons
|-- files
|   |-- bin
|   |   `-- gnome-todo
|   |-- lib
|   |   |-- debug
|   |   |-- evolution-data-server
|   |   |-- girepository-1.0
|   |   |-- gnome-todo
|   |   |-- goa-1.0
|   |   |-- libcamel-1.2.so -> libcamel-1.2.so.59.0.0
|   |   |-- [...]
|   |   `-- systemd
|   |-- manifest.json
|   `-- share
|       |-- appdata
|       |-- applications
|       |-- dbus-1
|       |-- GConf
|       |-- gir-1.0
|       |-- glib-2.0
|       |-- icons
|       |-- locale
|       |-- pixmaps
|       `-- runtime
`-- metadata
```

Putting all together: running a flatpak app

```
/.flatpak-info
/app
    /app/bin
    /app/lib
    /app/share
    [...]
/bin
/dev
    /dev/console
    /dev/full
    /dev/null
    [...]
/etc
[...]
/home/mario
    /home/mario/.config
    /home/mario/.local/share/flatpak
    /home/mario/.var/app/org.gnome.Todo
/lib
/lib64
/local
/proc
    /proc/1
    /proc/1/attr
    [...]
/run
    /run/build
    /run/build-runtime
    /run/host
    /run/systemd
    /run/user/1000
    [...]
    [...]
    /run/user/1000
    /run/user/1000/Xauthority
    /run/user/1000/app
    /run/user/1000/app/org.gnome.Todo
    /run/user/1000/bus
    /run/user/1000/dconf
    /run/user/1000/dconf/user
    /run/user/1000/doc
    /run/user/1000/flatpak-info
    [...]
/sbin
/sys
    /sys/block
    [...]
/tmp
    /tmp/.X11-unix
    /tmp/.X11-unix/X99
    [...]
/usr
    /usr/bin
    /usr/share
    /usr/share/applications
    [...]
/var
    /var/cache
    /var/config
    /var/config/user-dirs.dirs
    [...]
    /var/data
    /var/run
    /var/tmp
```

Platform and SDK Runtimes

Currently **two main standard runtimes** available:

- Freedesktop runtime: containes a set of essential libraries and services: D-Bus, GLib, PulseAudio, X11, Wayland
- GNOME runtime: based on the Freedesktop runtime, adds libraries like GTK+, GStreamer or GVFS on top.

  A KDE runtime is currently under development too:
  `https://github.com/KDE/flatpak-kde-runtime`

Two **types of runtimes**:

- Platform runtime: just the bits needed to **run** apps
- SDK runtime: platform + the necessary tools and files for development purposes (e.g. headers, debug symbols...)

The Sandbox

Limited access to the host system by default:

- No access to processes outside the sandbox (*namespaces*)
- No access to the network, session bus and devices
- Controlled execution of certain *syscalls* (*seccomp* filters)
- Read-only access to the runtime and app (*bind mounts*)
- Read-write access to `$HOME/.var/app/$APPID`
- Controlled access to resources (*cgroups*)
- No access to host services (e.g. X/Wayland, system bus...)

Flatpak's sandbox is very limiting by default, but there are ways of dealing with that to run real-word applications...

Escaping the Sandbox: fine-grained permissions

Easiest way to work with the sandbox is to open "holes" in it:

- Grant access to UNIX domain sockets: X.org, Wayland, PulseAudio, System and Sesssion D-Bus...
- Grant access to specific devices: dri, kvm
- Grant access to see, use and/or own specific D-Bus names
- Share specific subsystems with the host (network, IPC)
- Fine-grained permissions for filesystem access
- Define extensions for runtimes or applications (e.g. l10n)

Combining all this enables makes it possible to run apps in a more controlled way, but it's not very secure.

The manifest file

A Flatpak manifest file (`metadata`):

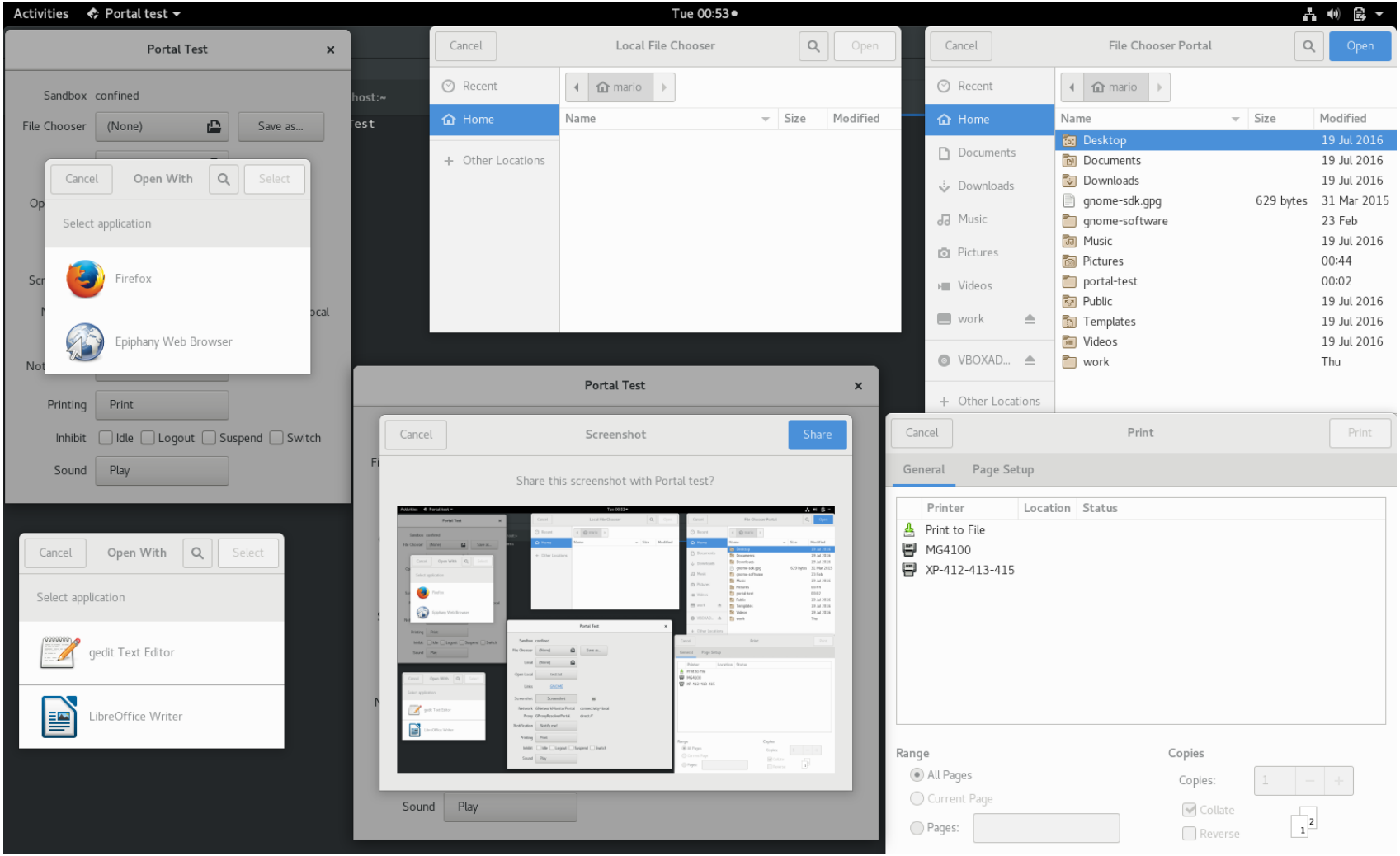
```
[Application]
name=org.gnome.Calculator
runtime=org.gnome.Platform/x86_64/3.20
sdk=org.gnome.Sdk/x86_64/3.20
command=gnome-calculator

[Context]
shared=network;ipc;
sockets=x11;wayland;
filesystems=xdg-run/dconf;~/.config/dconf:ro;

[Session Bus Policy]
ca.desrt.dconf=talk

[Environment]
DCONF_USER_CONFIG_DIR=.config/dconf

[Extension org.gnome.Calculator.Locale]
directory=share/runtime/locale
subdirectories=true

[Extension org.gnome.Calculator.Debug]
directory=lib/debug
```

Escaping the sandbox: Portals

- High-level APIs to allow sandboxed apps request access
- Out-of-process services, run in the host system
- Sandboxed apps communicate via D-Bus
- Different types of portals for different needs:
  - NetworkMonitor, OpenURI, Filechooser, Documents, Printing, Geolocation, Screenshots, Notifications, Proxy...
- Using portals is **safe**:
  - They don't expose sensible information from the host
  - Portal-initiated operations are **interactive** an **cancellable**
- Split in UI-less frontend + desktop-specific backends:
  - Currently backends for GTK+, with KDE work-in-progress.
  - GLib & GTK+ include support for several portals since 3.22

An example of Portals

```
$ flatpak run org.gnome.PortalTest
```



Building a flatpak apps

```
{
  "id" : "org.gnome.Todo",
  "branch" : "stable",
  "runtime" : "org.gnome.Platform",
  "runtime-version" : "3.22",
  "sdk" : "org.gnome.Sdk",
  "build-options" : {
    "cflags" : "-O2 -g",
    "cxxflags" : "-O2 -g",
    "env" : {
      "V" : "1"
    }
  },
  "command" : "gnome-todo",
  "modules" : [
    {
      "name" : "gnome-online-accounts",
      "config-opts" : [
        "--disable-telepathy",
        "--disable-documentation",
        "--disable-backend"
      ],
      "sources" : [
        {
          "url" : "https://download.gnome.org/sources/gnome-online-accounts/3.22/gnome-online
          "sha256" : "aacce93a71bf5e687a45ae0d00f31ea0625ddd8143235d6d8c64c4ec21bbfa33",
          "type" : "archive"
        }
      ]
    },
    [...] ---> More depedencies here
```

Building a flatpak apps (II)

```
    [...]
    {
      "name" : "gnome-todo",
      "sources" : [
        {
          "url" : "https://download.gnome.org/sources/gnome-todo/3.22/gnome-todo-3.22.1.tar.x
          "sha256" : "cb80f64f5edeeac7b221146d2203bd1bebc49d275b7a41e7a5418f409d9c74af",
          "type" : "archive"
        }
      ]
    }
  ],
  "cleanup" : [
    "/include", "/lib/pkgconfig", "/share/pkgconfig", "/share/aclocal", "/man",
    "/share/man", "/share/gtk-doc", "/share/vala", "*.la", "*.a"
  ],
  "finish-args" : [
    "--share=ipc",
    "--socket=x11",
    "--socket=wayland",
    "--share=network",
    "--talk-name=org.gnome.OnlineAccounts",
    "--talk-name=org.gnome.evolution.dataserver.AddressBook9",
    "--talk-name=org.gnome.evolution.dataserver.Calendar7",
    "--talk-name=org.gnome.evolution.dataserver.Sources5",
    "--talk-name=org.gnome.evolution.dataserver.Subprocess.Backend.*",
    "--filesystem=xdg-run/dconf",
    "--filesystem=~/.config/dconf:ro",
    "--talk-name=ca.desrt.dconf",
    "--env=DCONF_USER_CONFIG_DIR=.config/dconf"
  ]
}
```

Application distribution

- Publish your local repository: `build-export`
  - Export your app to an OSTree (archive-z2) repository
  - You could publish this repository now over HTTP
- Sign everything: `build-sign`, `build-update-repo`
  - Important to GPG-sign the commits and the summary file
  - Allows using unencrypted HTTP (faster downloads)
  - Recommended to create a dedicated GPG key
- Push to a production public repository: e.g. `rsync`
  - Simple requirements: static files served over HTTP!
  - Push it to your public server once you're happy
  - Order your commands wisely (avoid race conditions)

Application distribution (II)

- Configure your public repository appropriately:
  - `build-update-repo -title=<title>`
  - `build-update-repo -default-branch=<branch>`
- Provide efficient updates:
  - Enable HTTP keep-alive in the server (lots of files)
  - Use OSTree's static-deltas feature (good for big files)
  - Run `build-update-repo` everytime an app changes
- Generate application metadata for software centers:
  - Generate AppStream data for each application in your repo: `build-update-repo` will put it an `appstream` branch
  - Make sure your apps must export an AppData XML file!

Flatpak filetypes: .flatpakrepo and .flatpakref

Configuring flatpak "repositories": `gnome.flatpakrepo`

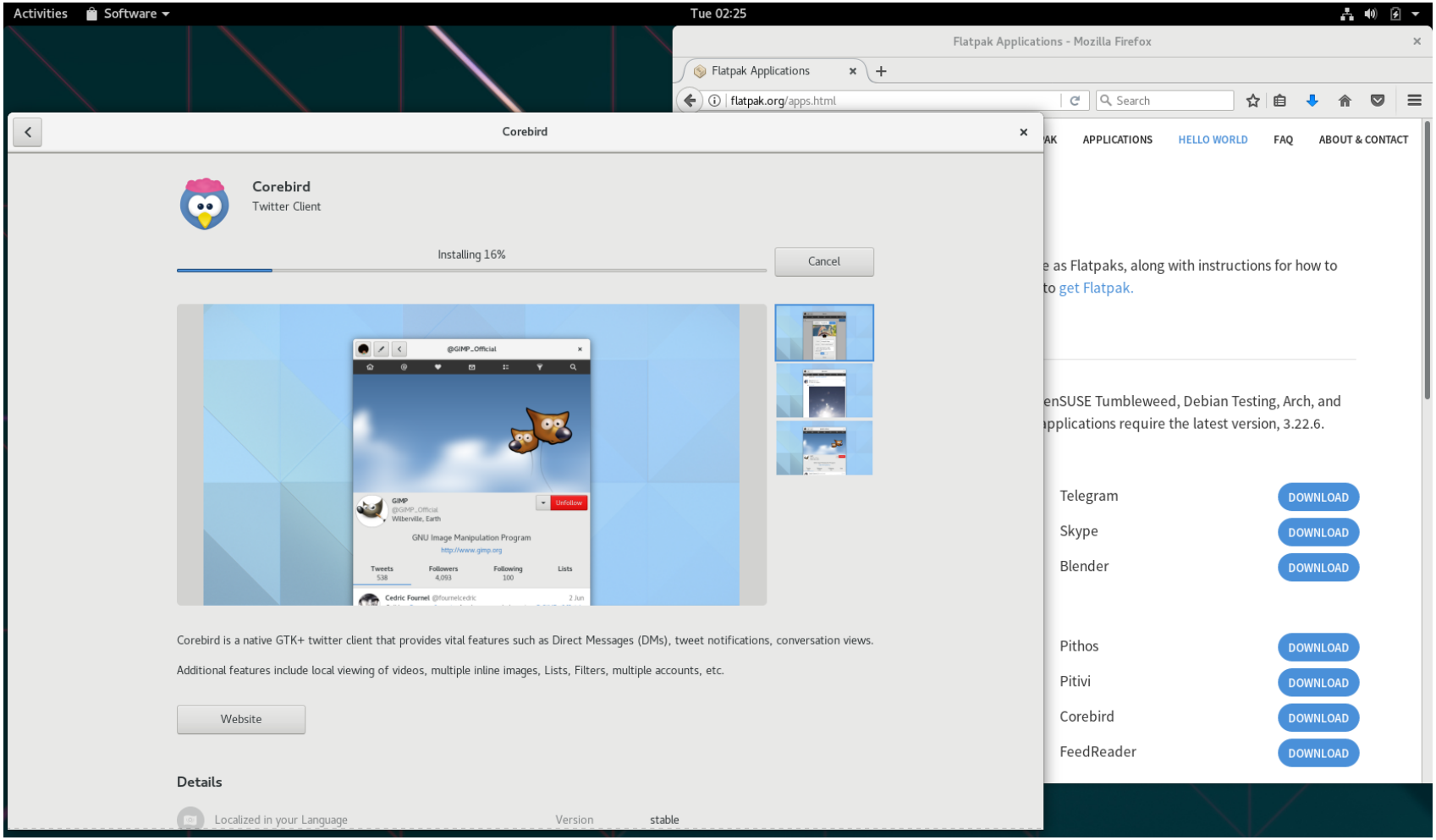
```
[Flatpak Repo]
Title=Gnome Stable Runtimes
Url=http://sdk.gnome.org/repo/
Homepage=https://www.gnome.org/get-involved/
Comment=The standard Gnome runtime used by most gnome apps
Description=GNOME runtimes are released with each major release and contain the main GNOME
platform libraries. At the moment they only receive minor bug fixing and security updates,
but should be considered ABI stable and frozen.
Icon=https://www.gnome.org/wp-content/themes/gnome-grass/images/gnome-logo.png
GPGKey=mQENBFUU[...]15w8jmY=
```

Installing an application: `gnome-recipes.flatpakref`

```
[Flatpak Ref]
Title=GNOME Recipes
Name=org.gnome.Recipes
Branch=master
Url=https://matthiasclasen.github.io/recipes-releases/repo
IsRuntime=False
GPGKey=mQENBFis[...]Kpp5G2YW
RuntimeRepo=https://sdk.gnome.org/gnome.flatpakrepo
Comment=GNOME loves to cook
```

Installing a flatpak application in one click

GNOME Software, flatpak and `.flatpakref` files in action:



Questions?

Thank you!

References:

- *OSTree documentation:* `https://ostree.readthedocs.io/en/latest`
- *Project Atomic:* `https://www.projectatomic.io`
- *GNOME Continuous:* `https://wiki.gnome.org/Projects/GnomeContinuous`
- *Qt OTA updates:* `https://doc.qt.io/QtOTA`
- *Automotive Linux:* `https://automotivelinux.org`
- *Endless OS:* `https://endlessos.com`
- *Bubblewrap:* `https://github.com/projectatomic/bubblewrap`
- *Flatpak documentation:* `https://docs.flatpak.org/en/latest`
- *Flatpak portals:* `https://github.com/flatpak/xdg-desktop-portal`
- *Flatpak portals (GTK):* `https://github.com/flatpak/xdg-desktop-portal-gtk`
- *Alex Larsson's blog:* `https://blogs.gnome.org/alexl`
- *Christian Hergert's talk on Scale15x:* `https://hergert.me/talks/Flatpak-Scale-15x.pdf`