Move Fast and Break Everything

Sam Thursfield

Move Fast and Break Everything

Part 1. Get to know your daemons

Part 2. Learn to control them

https://wiki.gnome.org/Newcomers/ChooseProject

GNOME Newcomers / ChooseProject Home RecentChanges Schedule SamThursfield Settings Logout

Choose a Project

GNOME has got hundreds of projects. To make it easier for you to get started, we have highlighted the applications which are great starting points for making your first contribution.

Polari (#polari)

An easy to use IRC client, written in Javascript

Project complexity: Simple

Code: https://gitlab.gnome.org/GNOME/polari.git

Mentors: Bastian Ilsø (bastianilso), Florian Müllner (fmuellner)

Games (#gnome-games)

Game manager for your retro and Steam games, written in Vala

Project complexity: Medium

Code: https://gitlab.gnome.org/GNOME/gnome-games.git

Mentors: Alexander Mikhaylenko (exalm)

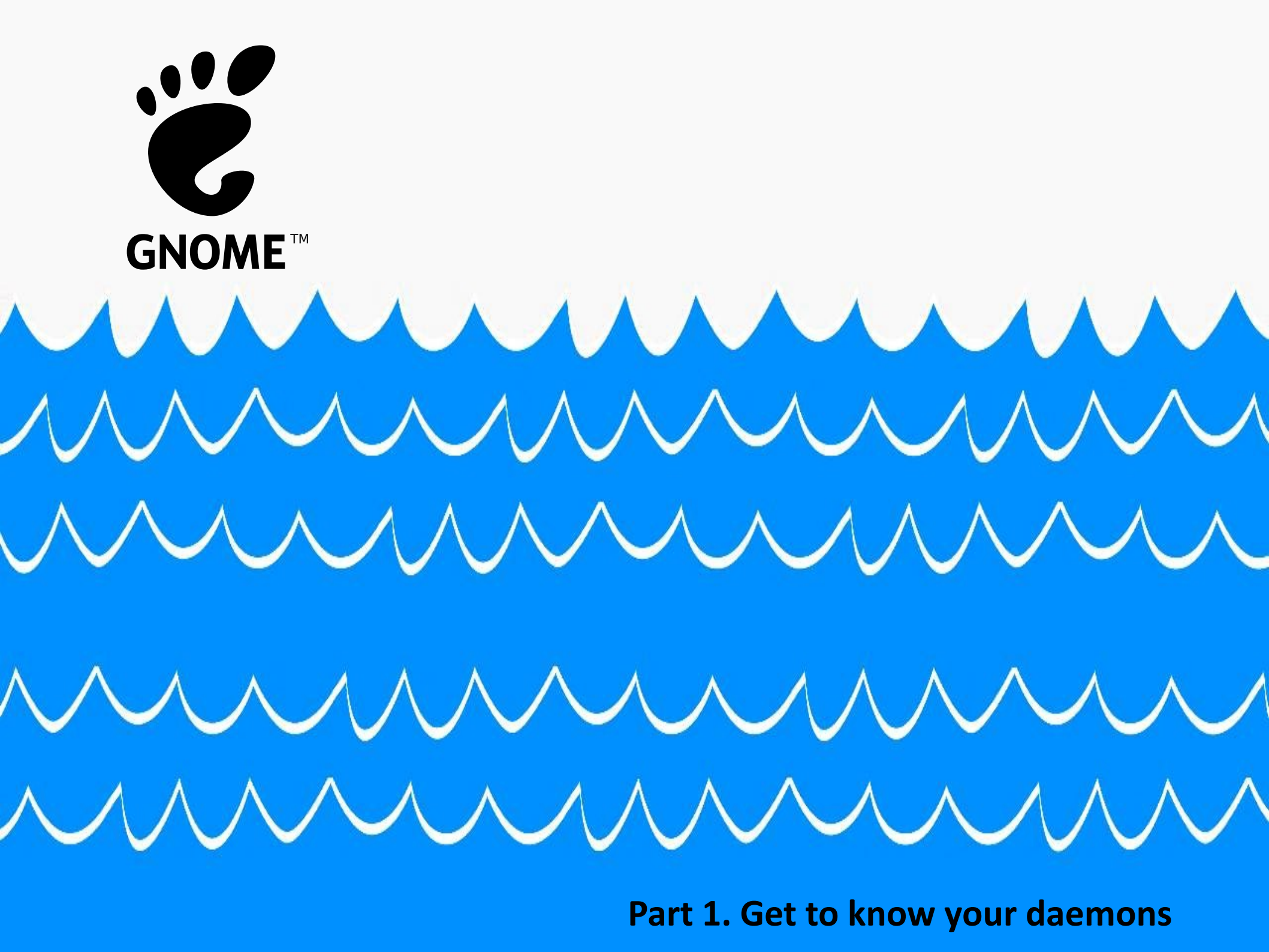
Maps (#gnome-maps)

A simple map application, written in Javascript.

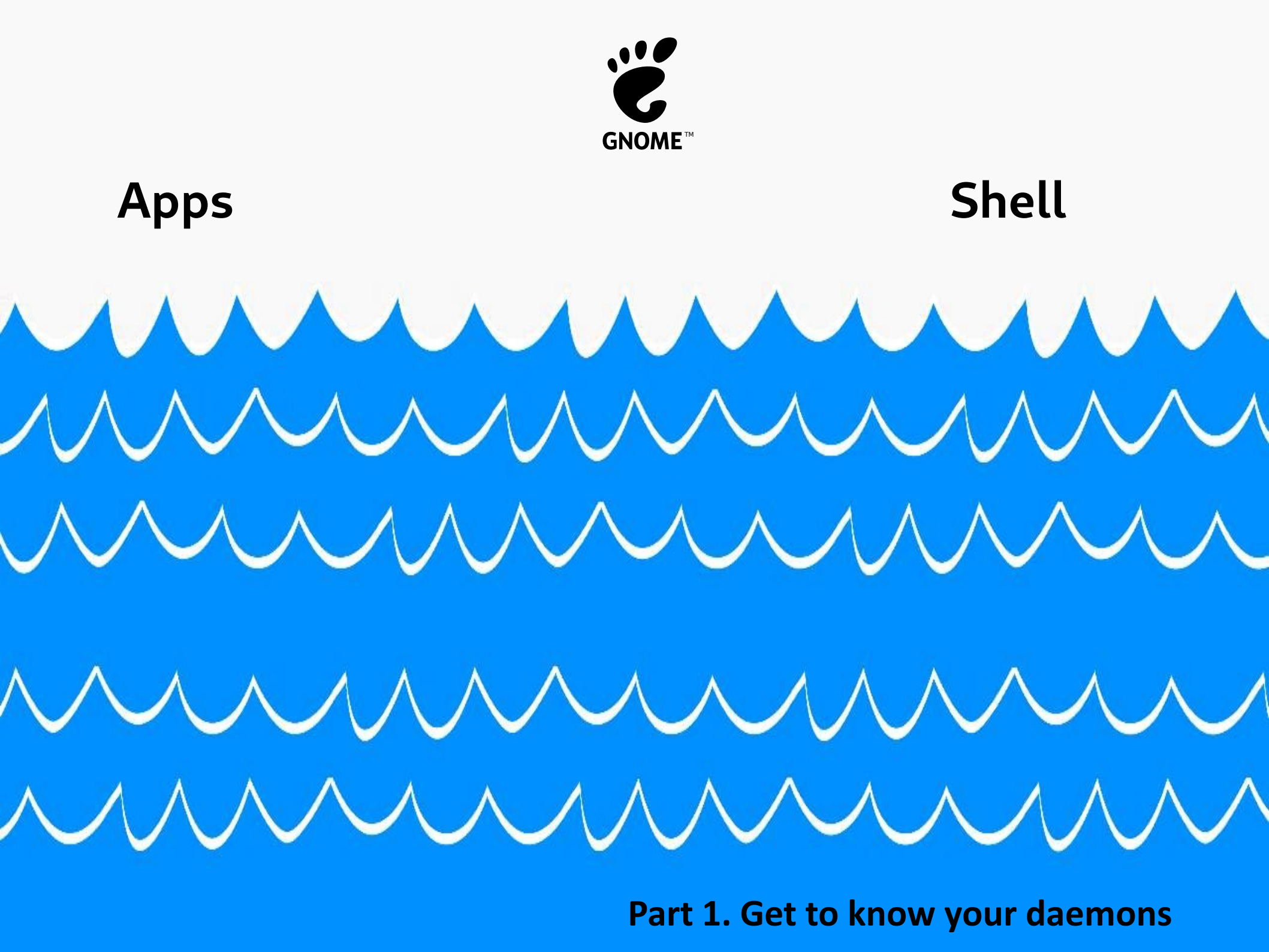
Project complexity: Simple

Code: https://gitlab.gnome.org/GNOME/gnome-maps

Mentors: Jonas Danielsson (jonasdn), Marcus Lundblad (marcus), Amisha Singla (amisha)



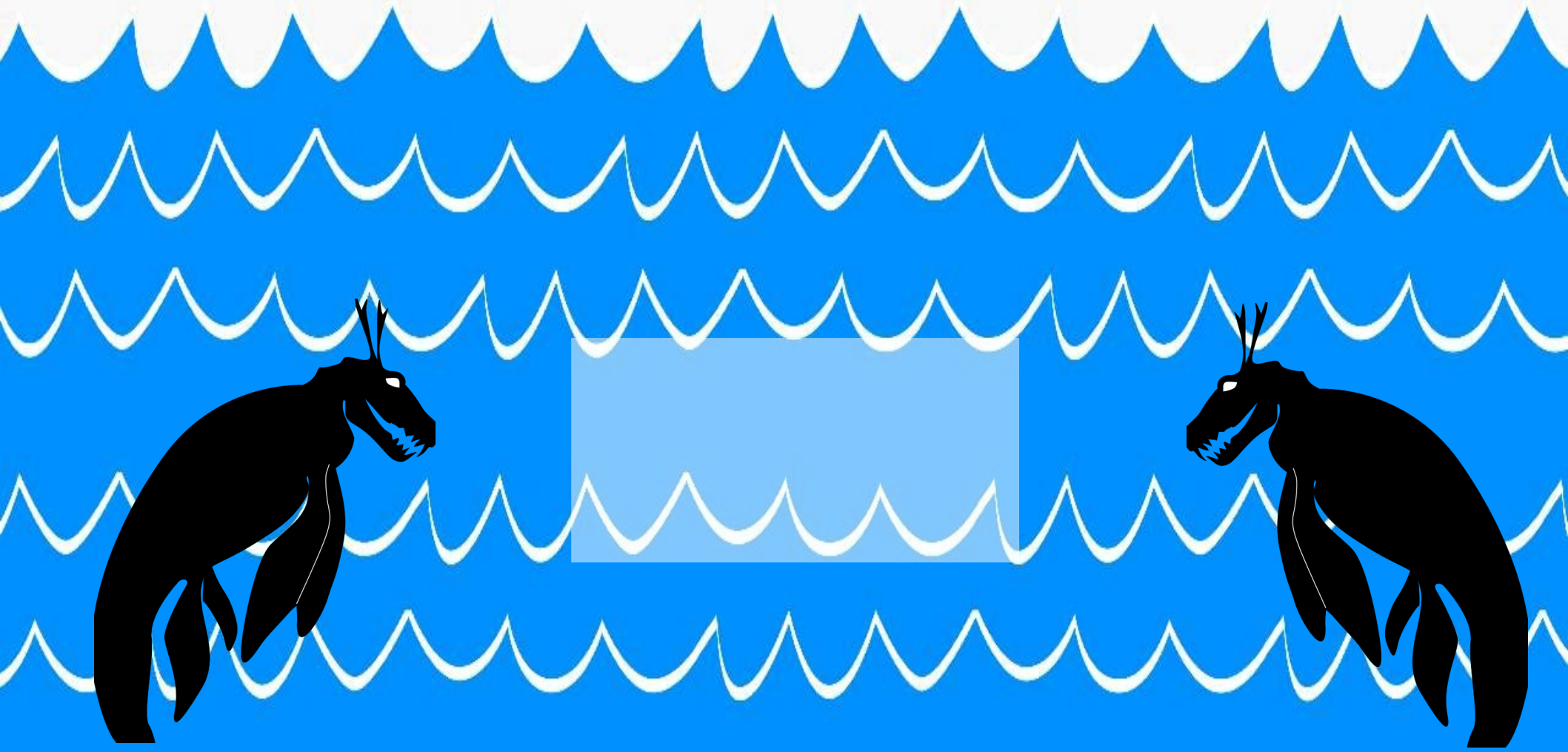
Part 1. Get to know your daemons


GNOME™
Apps
Shell
Part 1. Get to know your daemons

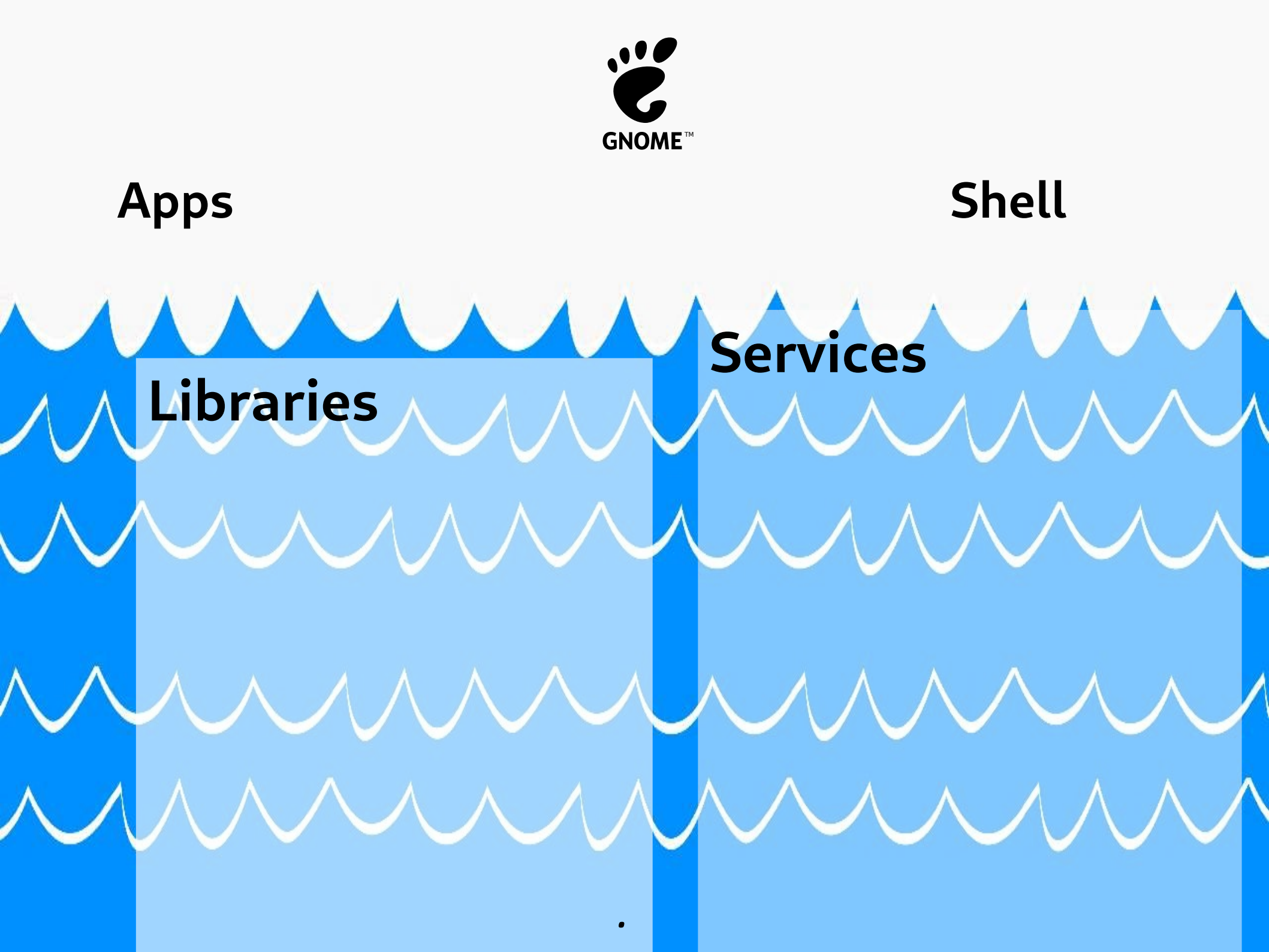
GNOME™

Apps

Shell



Part 1. Get to know your daemons

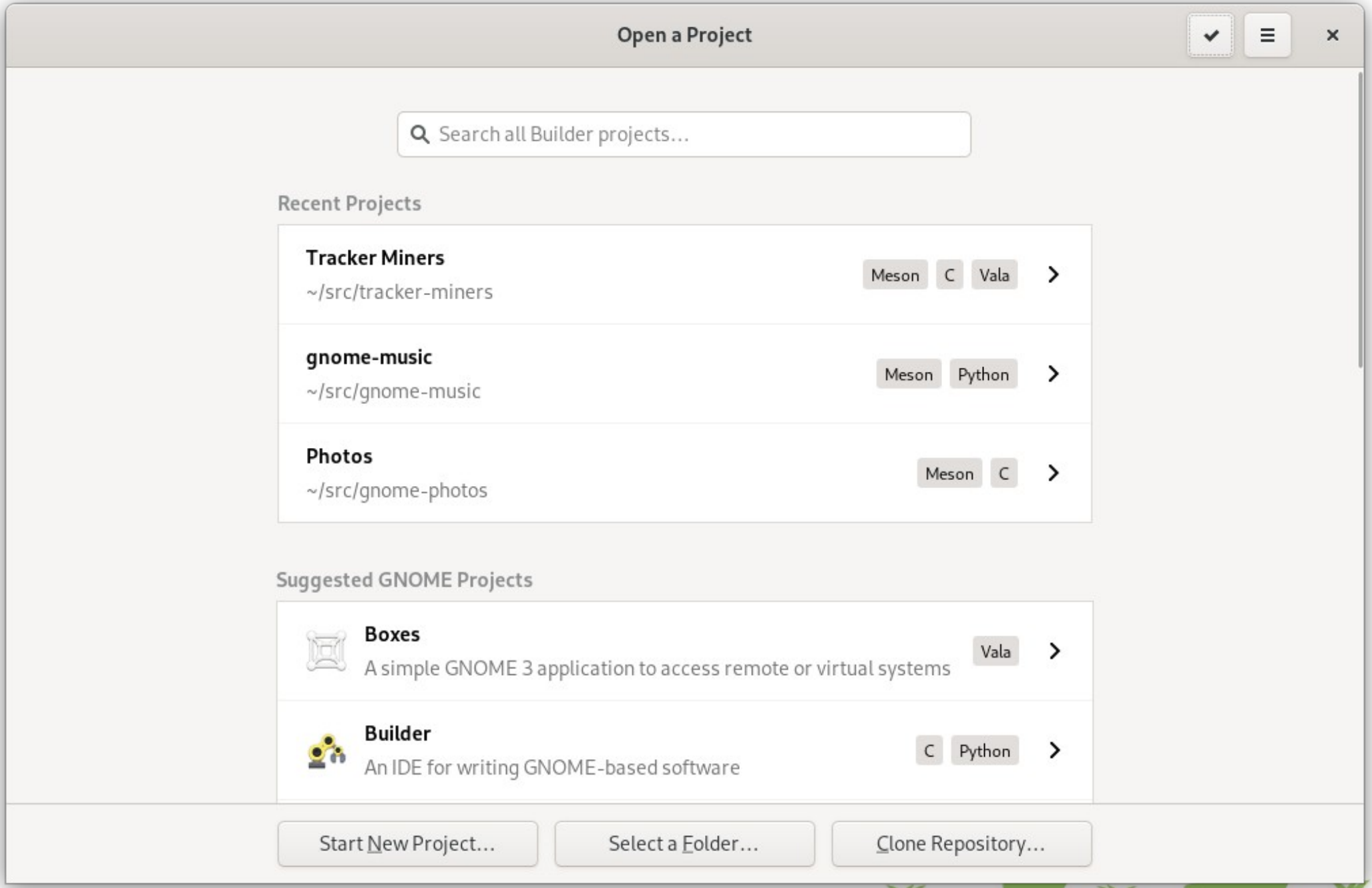

GNOME™
Apps
Shell
Services
Libraries

-
-
-

pstree

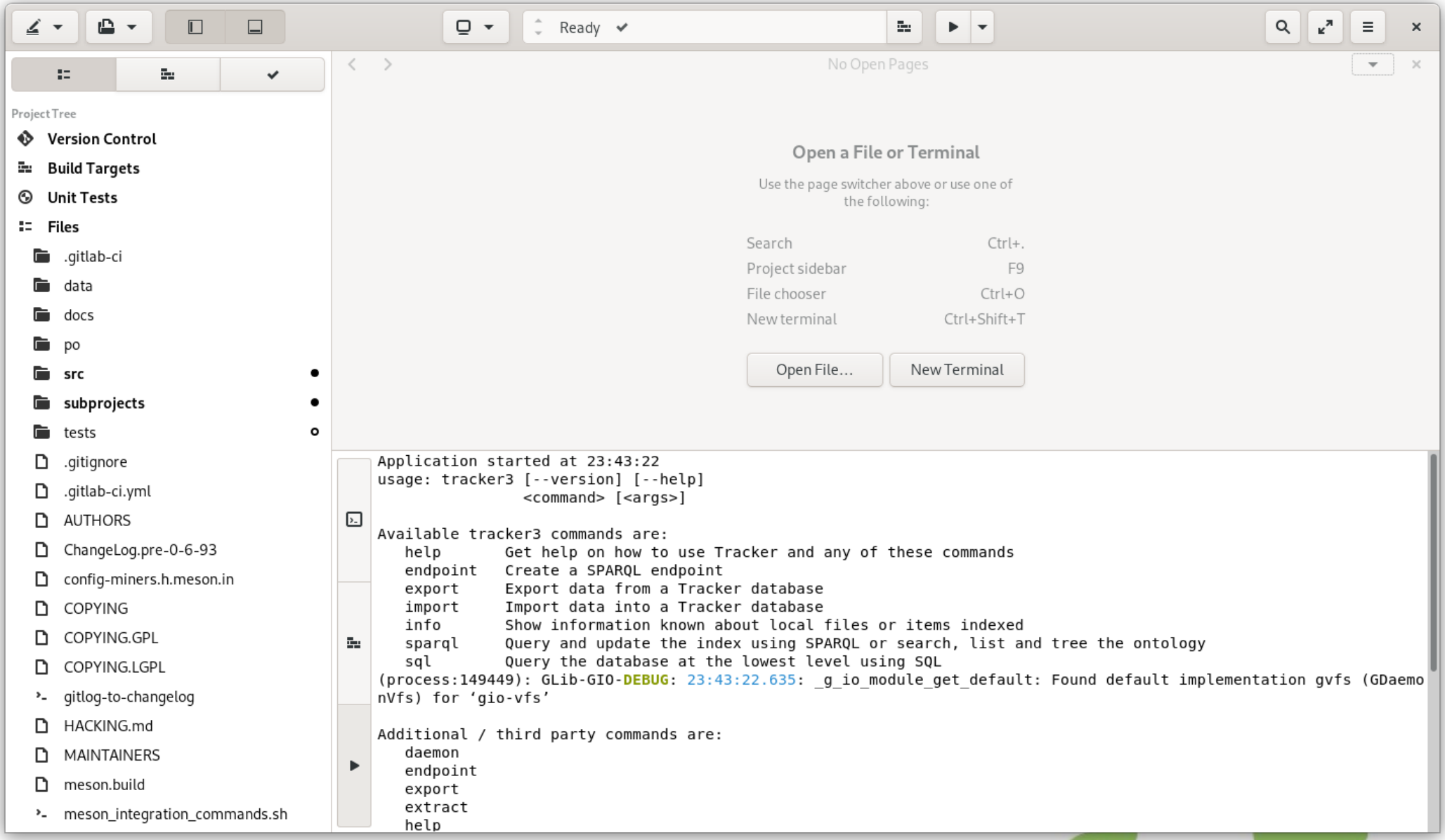
```
systemd─┬─ModemManager───2*[{ModemManager}]
        ├─NetworkManager───2*[{NetworkManager}]
        ├─abrt-dbus───2*[{abrt-dbus}]
        ├─3*[abrt-dump-journ]
        ├─abrtd───2*[{abrtd}]
        ├─accounts-daemon───2*[{accounts-daemon}]
        ├─alsactl
        ├─auditd─┬─sedispatch
        │        └─2*[{auditd}]
        ├─avahi-daemon───avahi-daemon
        ├─colord───2*[{colord}]
        ├─crond
        ├─cupsd
        ├─dbus-broker-lau───dbus-broker
        ├─2*[dnsmasq───dnsmasq]
        ├─earlyoom
        ├─firewalld───{firewalld}
```

Part 2: Control your daemons



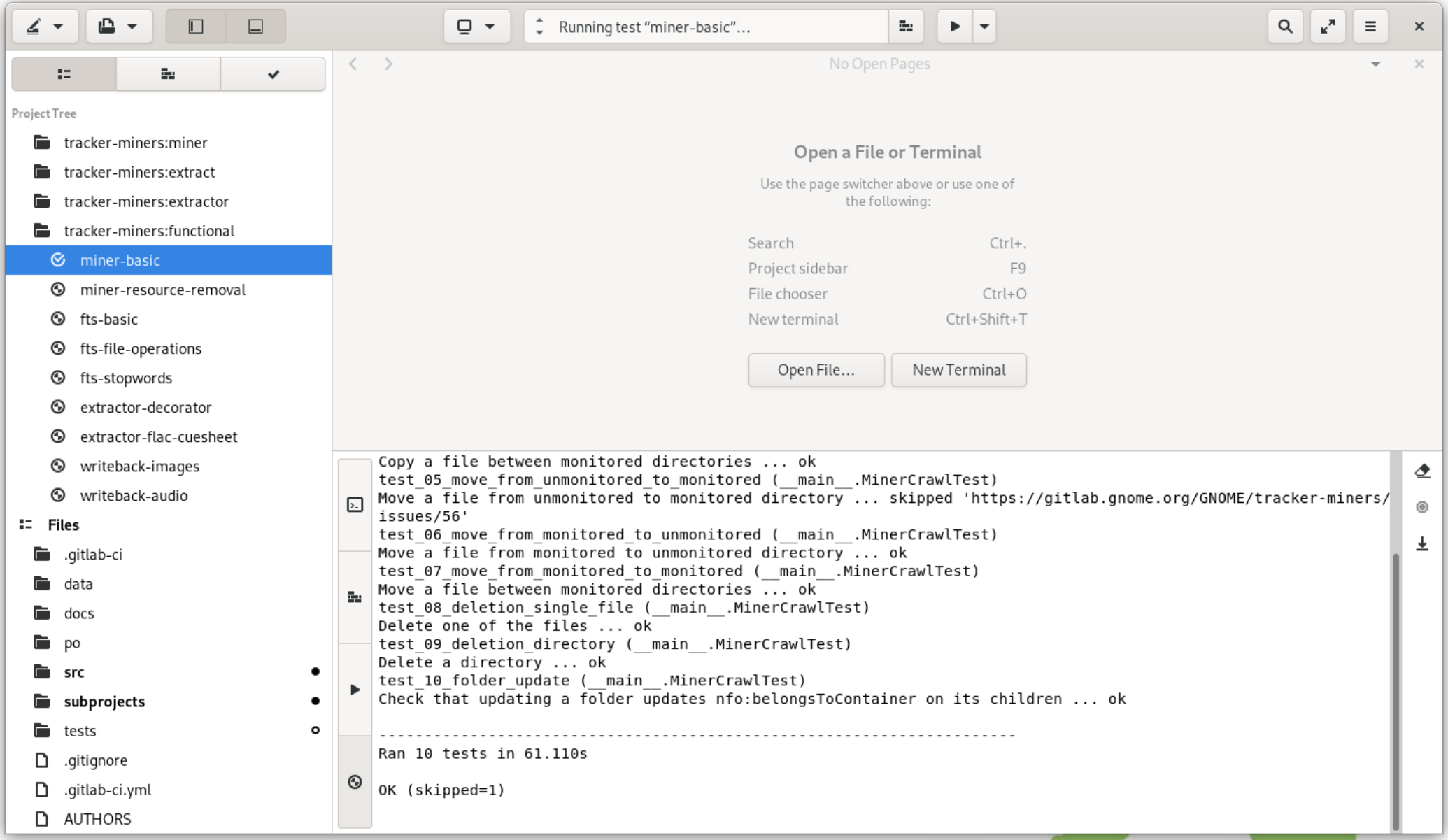
Part 2. Control your daemons

Building and running Tracker with GNOME Builder



Part 2. Control your daemons

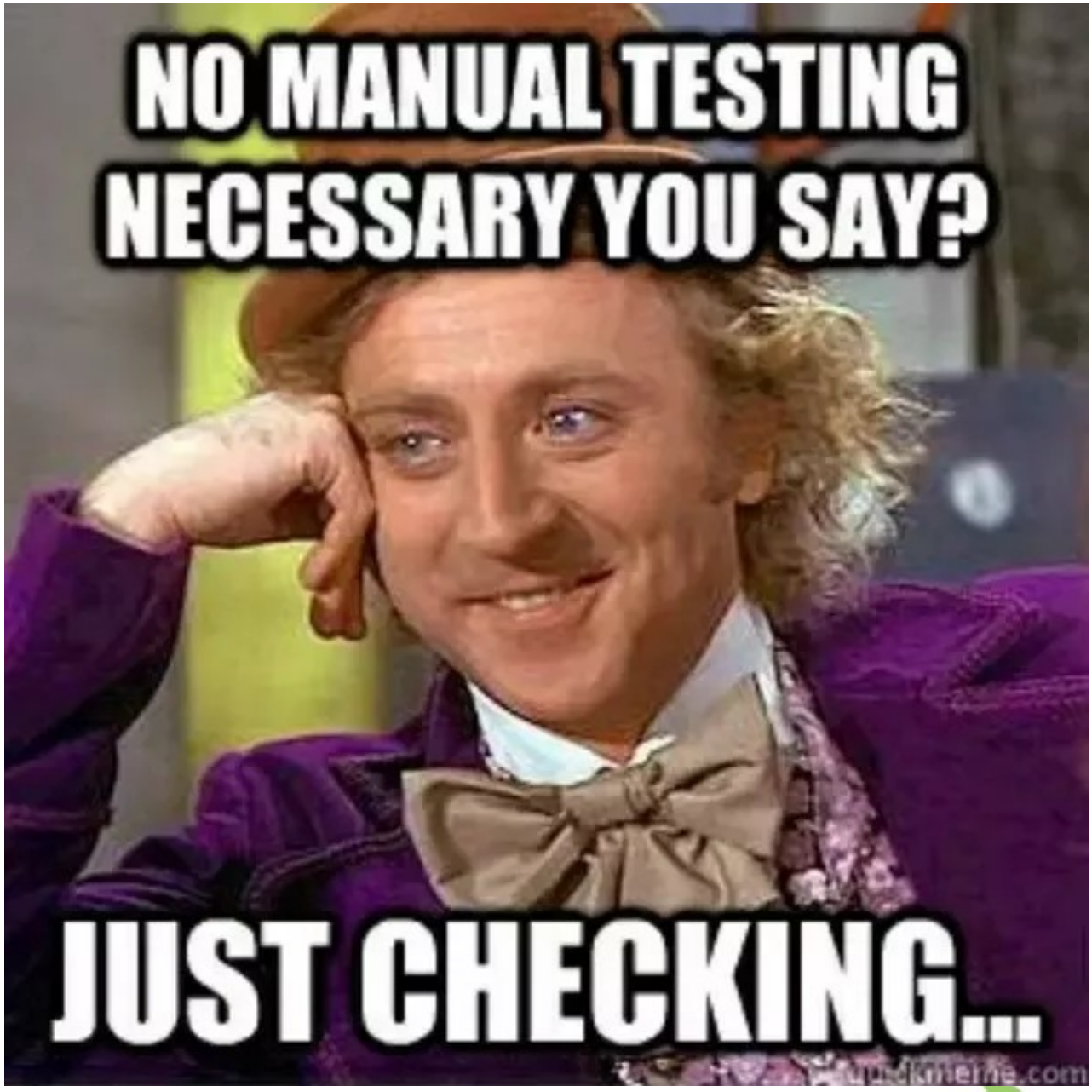
Running an automated test



Part 2. Control your daemons

- `dbus-run-session`
- `umockdev`
-

`tests/functional-tests`


NO MANUAL TESTING
NECESSARY YOU SAY?
JUST CHECKING...

The best way to deploy a test build?

Run from source tree
Install into /usr
Install into /opt
Use distro packaging tools
Use BuildStream to build a VM image

Run from the source tree

- Unlikely to work, but try it!

- Project can provide a helper script.

https://gitlab.gnome.org/GNOME/tracker-miners/-/blob/master/run-uninstalled.in

-
-

-
-

```
export XDG_DATA_DIRS=/opt/tracker3/share:/usr/share
dbus-run-session /opt/tracker3/bin/tracker3 search Foo
```

- jhbuild

-
-
-

-
- `bst shell`
-
-

(see Valentin's talk)

The best way to deploy a test build?

	Fast	Reproducible	No extra codepaths needed	No extra computer or VM needed
Run from source tree	✓	✓		✓
Install into /usr	✓		✓	
Install into /opt	✓			✓
Use distro packaging tools		✓	✓	
Use BuildStream to build a VM image		✓	✓	

In summary…

-
  - *Do you maintain a daemon? Update the README :)*
-
  - *If a service project doesn't have functional testing, look at how to add it!*
-
  - *How can we make sure it's frictionless?*