accu 2021
VIRTUAL EVENT

Bloomberg
Engineering

undo

mosaic
CONSULTANTS TO FINANCIAL SERVICES

What Use Is a Confined User Shell?

Alan Griffiths

What use is a confined user shell?

Alan Griffiths
alan.griffiths@canonical.com

What use is a confined user shell?

Confinement

A confined process has limited access to the system

User shell

A shell interacts with the computer on a user's behalf: A way to control other programs

Confinement

Traditionally

- ➢ The programs you run can access everything you can
- ➢ Installation mechanisms use root access

Confinement

Restricts access to your computer to only those things a program needs to work

Confinement

[DEMO] confined command-line

Confinement

Code running on a computer can be divided into “kernel” and “userspace”

Snap confinement

Snap confinement is Canonical’s chosen approach to confining programs for Ubuntu. Snaps use AppArmor “under the hood”.

The rest of this talk covers snaps and AppArmor because I work with this.

Other packaging and confinement technologies I’m aware of:

- Flatpak uses SELinux
- “clicks” which also use AppArmor

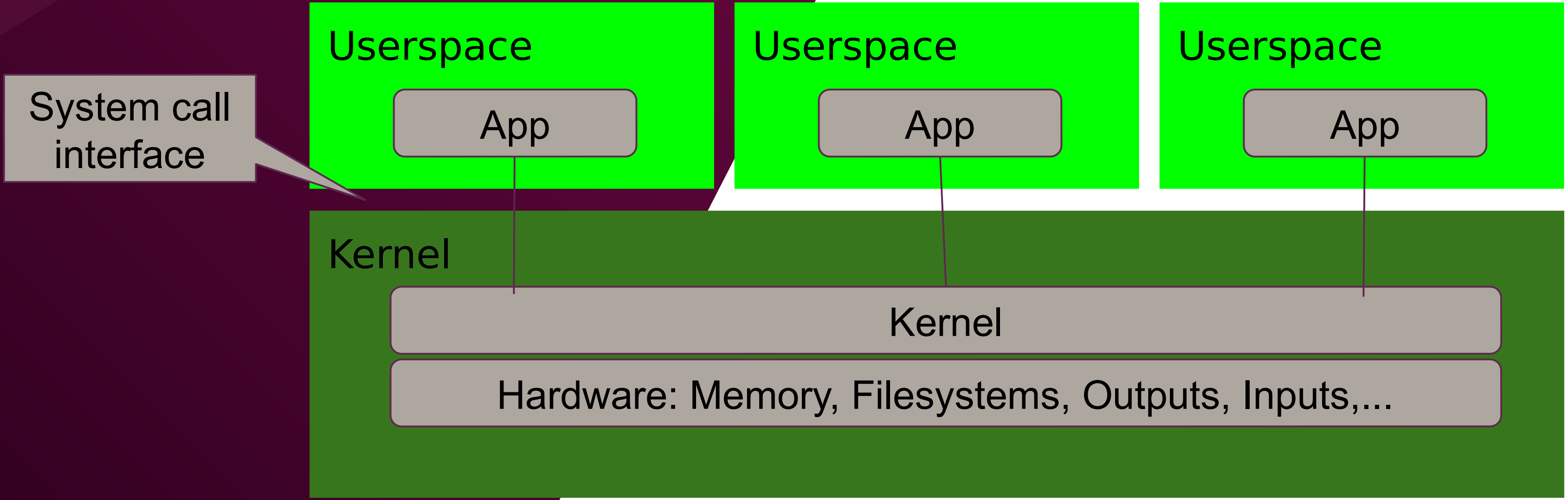
While details differ the principles

Confinement

Code running on a computer can be divided into “kernel” and “userspace”

Userspace

The **userspace** is everything that runs within a normal program



Confinement

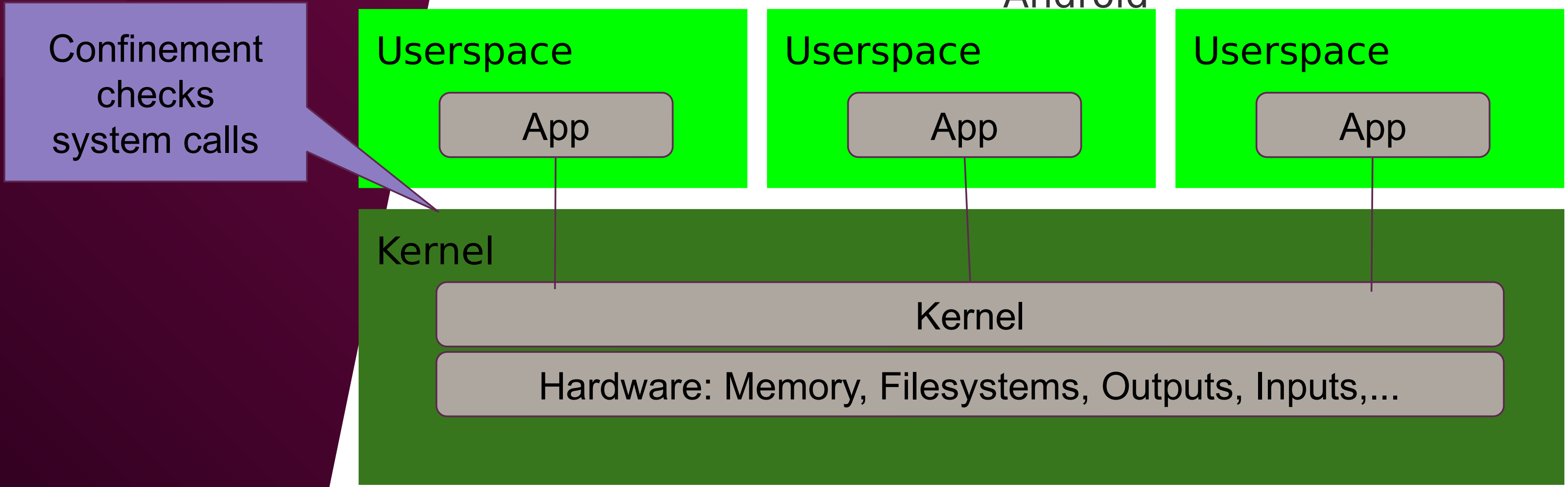
Code running on a computer can be divided into “kernel” and “userspace”

Debian, Ubuntu etc

AppArmor is common in Debian derived distros

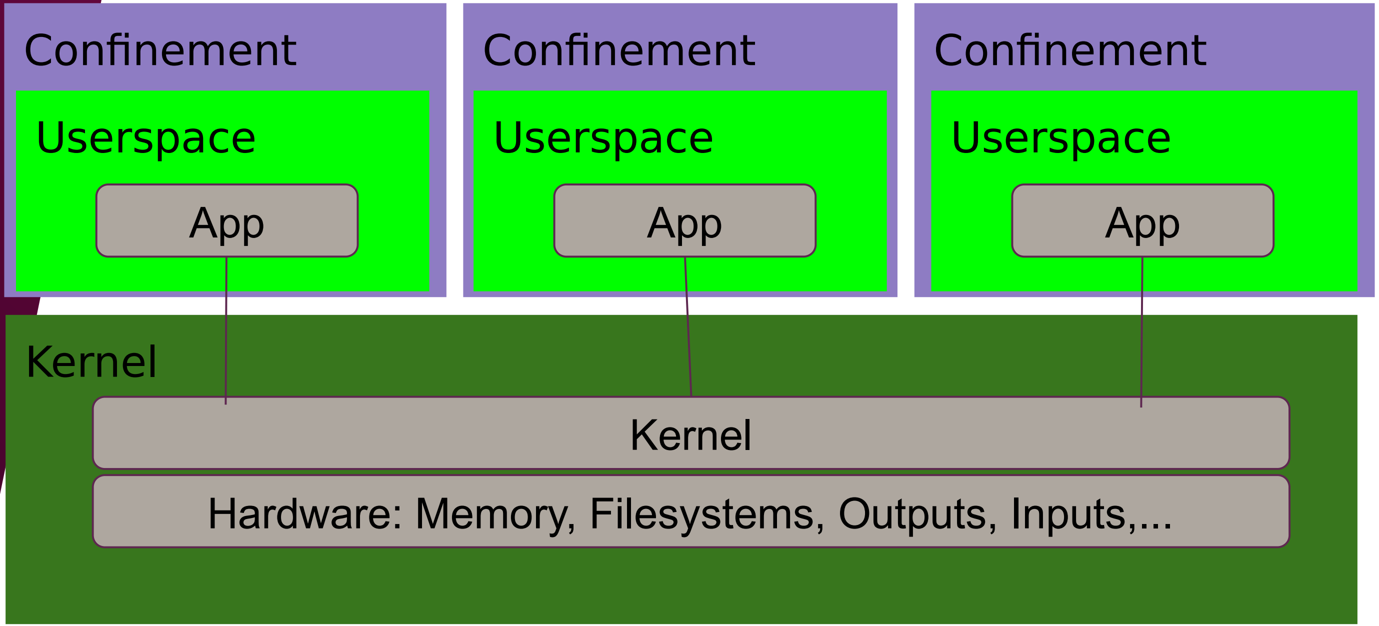
Red Hat, Android, etc

SELinux is common in Red Hat based distros and in Android


Confinement checks system calls
Userspace
App
Userspace
App
Userspace
App
Kernel
Kernel
Hardware: Memory, Filesystems, Outputs, Inputs,...

Confinement

Code running on a computer can be divided into “kernel” and “userspace”



Confinement

AppArmor
Code running on a computer can be divided into “kernel” and “userspace”

AppArmor

AppArmor configuration is based on text files. These contain rules for matching resources on the system and specify the access that is permitted.

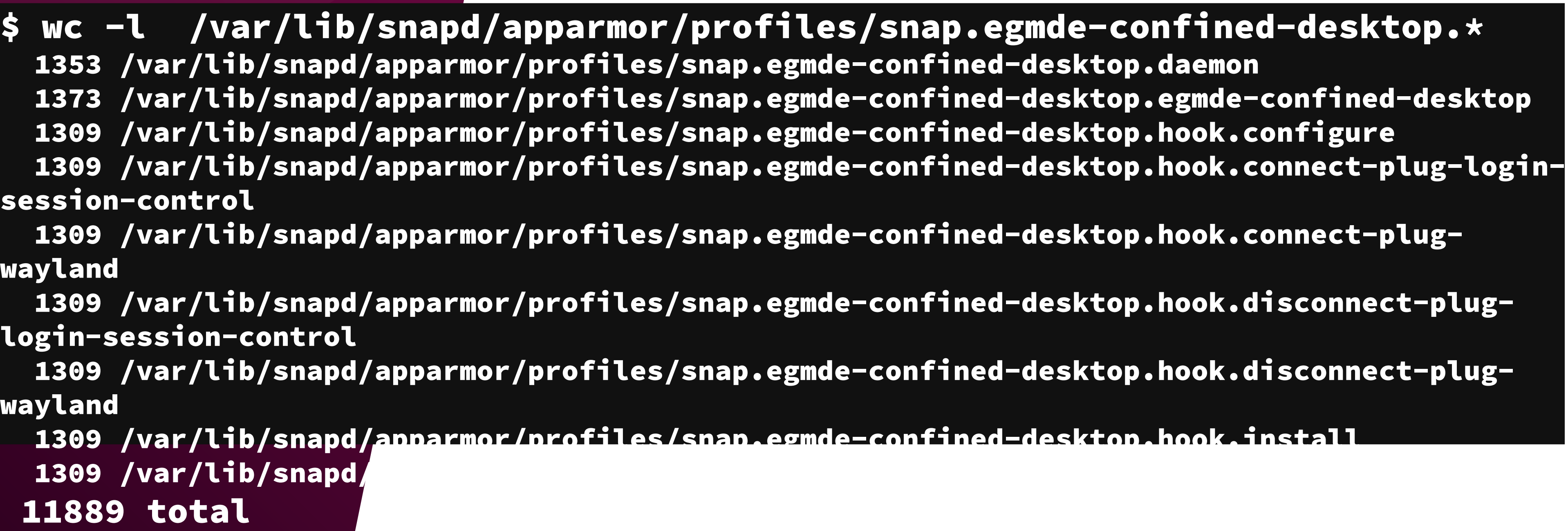
For example the line:

```
    owner /run/user/[0-9]*/wayland-[0-9]* rw,
```

allows read and write access to any files matching the pattern that have the same owner (i.e. user) as the app’s process.

Confinement

AppArmor profiles

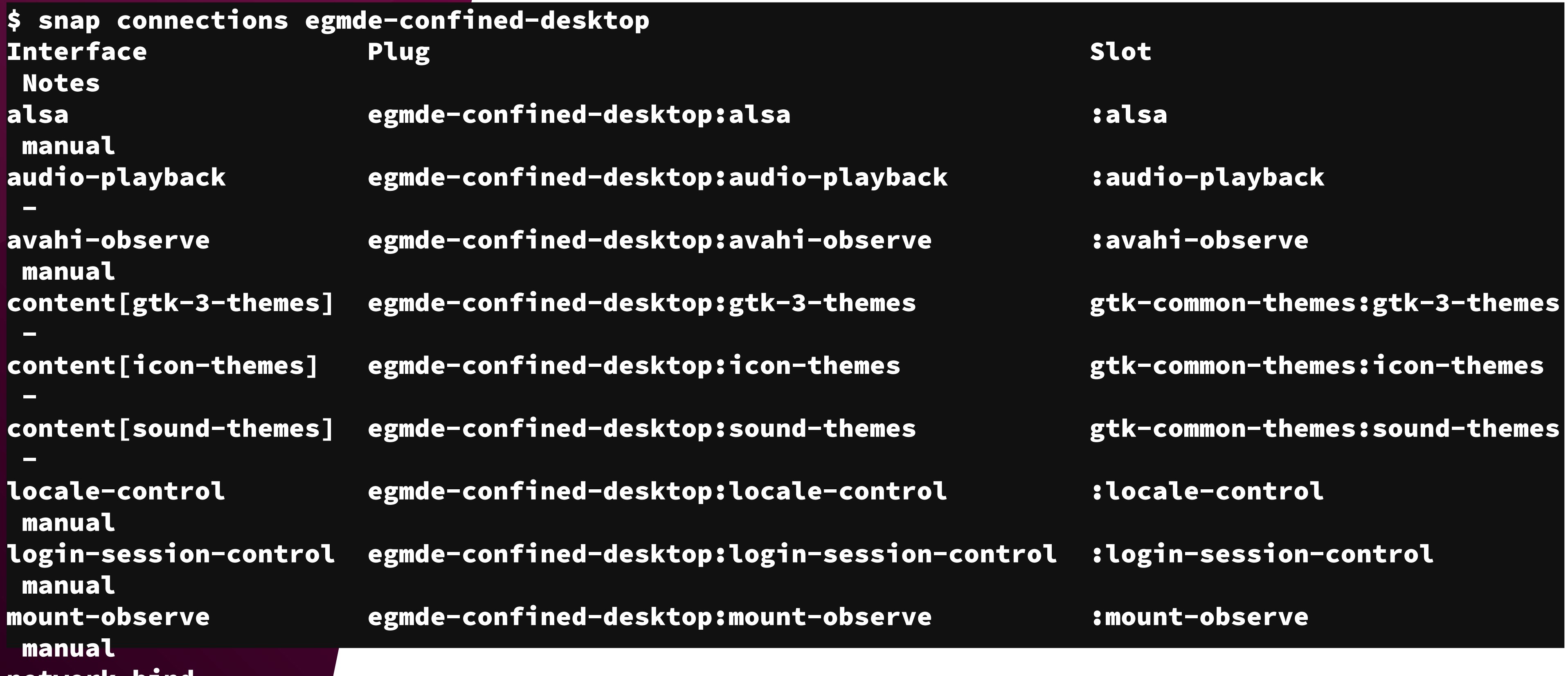


```
$ wc -l  /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.*
  1353 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.daemon
  1373 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.egmde-confined-desktop
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.configure
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.connect-plug-login-
session-control
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.connect-plug-
wayland
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.disconnect-plug-
login-session-control
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.disconnect-plug-
wayland
  1309 /var/lib/snapd/apparmor/profiles/snap.egmde-confined-desktop.hook.install
  1309 /var/lib/snapd/
 11889 total
```

Confinement

Snap interfaces

Snaps make use of lists of AppArmor rules called “interfaces” each of which covers identifiable capabilities. These can be enabled (or disabled) by the end user.



```
$ snap connections egmde-confined-desktop
Interface              Plug                                          Slot
 Notes
alsa                   egmde-confined-desktop:alsa                   :alsa
 manual
audio-playback         egmde-confined-desktop:audio-playback         :audio-playback
 -
avahi-observe          egmde-confined-desktop:avahi-observe          :avahi-observe
 manual
content[gtk-3-themes]  egmde-confined-desktop:gtk-3-themes           gtk-common-themes:gtk-3-themes
 -
content[icon-themes]   egmde-confined-desktop:icon-themes            gtk-common-themes:icon-themes
 -
content[sound-themes]  egmde-confined-desktop:sound-themes           gtk-common-themes:sound-themes
 -
locale-control         egmde-confined-desktop:locale-control         :locale-control
 manual
login-session-control  egmde-confined-desktop:login-session-control  :login-session-control
 manual
mount-observe          egmde-confined-desktop:mount-observe          :mount-observe
 manual
network-bind
```

What use is a confined user shell?

Confinement

A confined process has limited access to the system

User shell

A shell interacts with the computer on a user’s behalf: A way to control other programs

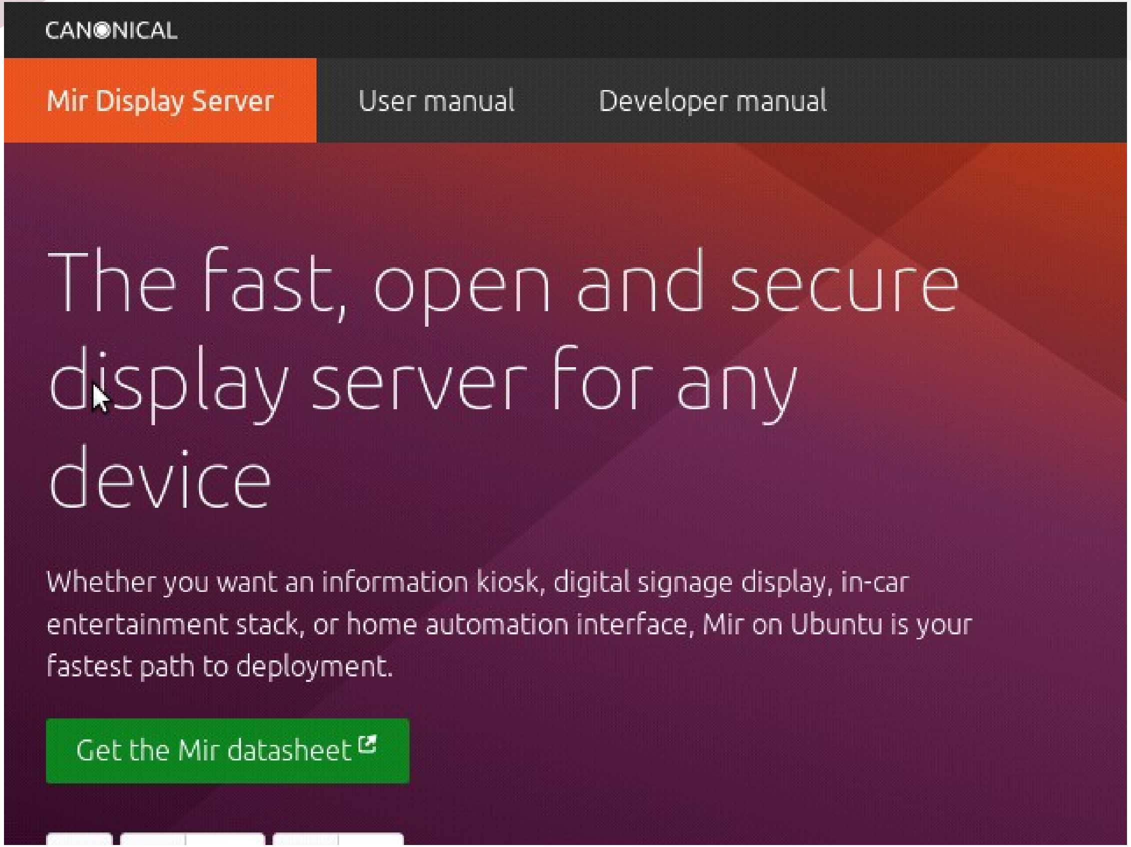
Graphical shell

Mir-kiosk is a simple embedded shell based on Mir

[DEMO] confined graphical shell

Graphical shell

Mir-kiosk is a simple embedded shell based on Mir

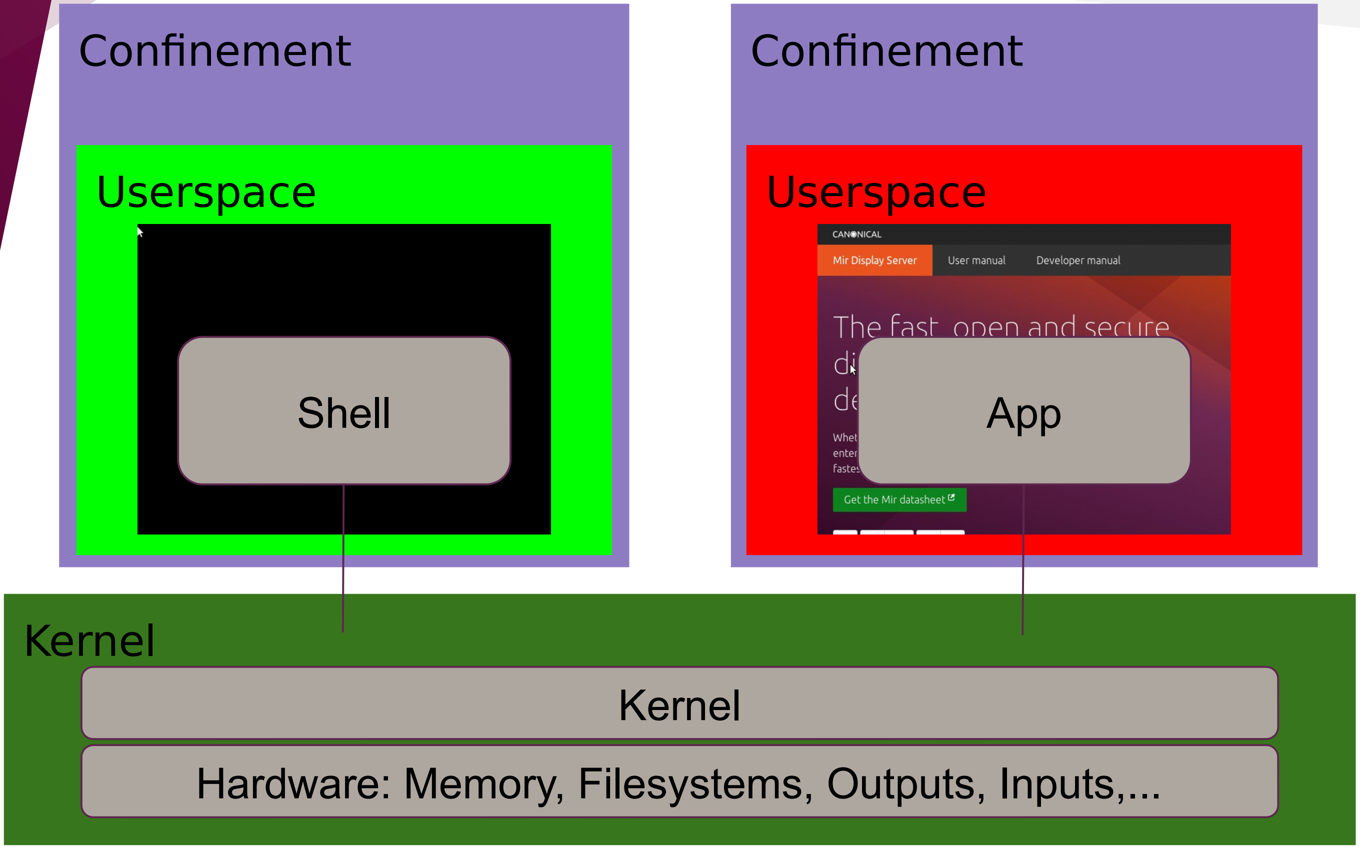


The fast, open and secure display server for any device

Whether you want an information kiosk, digital signage display, in-car entertainment stack, or home automation interface, Mir on Ubuntu is your fastest path to deployment.

Graphical shell

Confinement
Shell and App are confined separately



Shell and apps are different

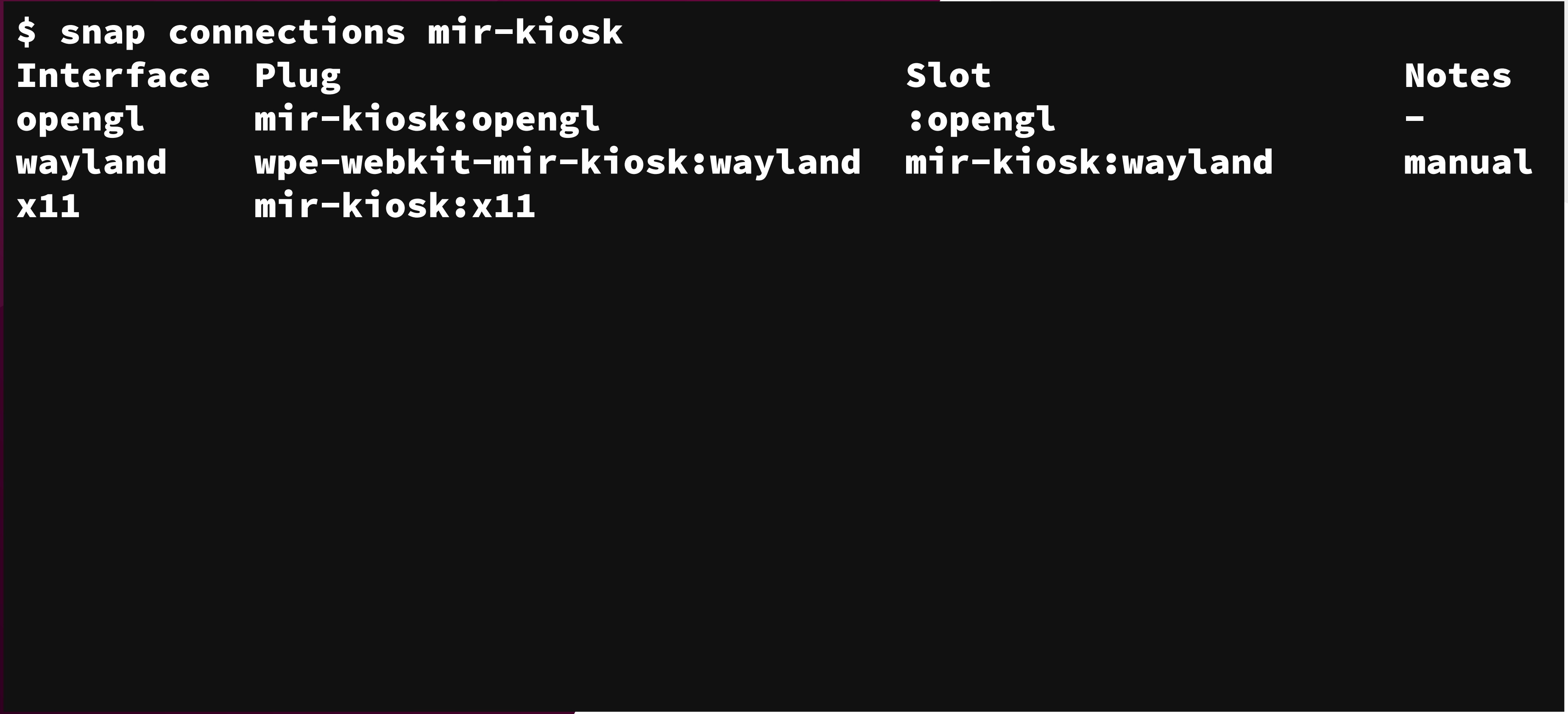
A graphical shell needs...

- ➢ User input & output
- ➢ Graphics

A web-kiosk needs...

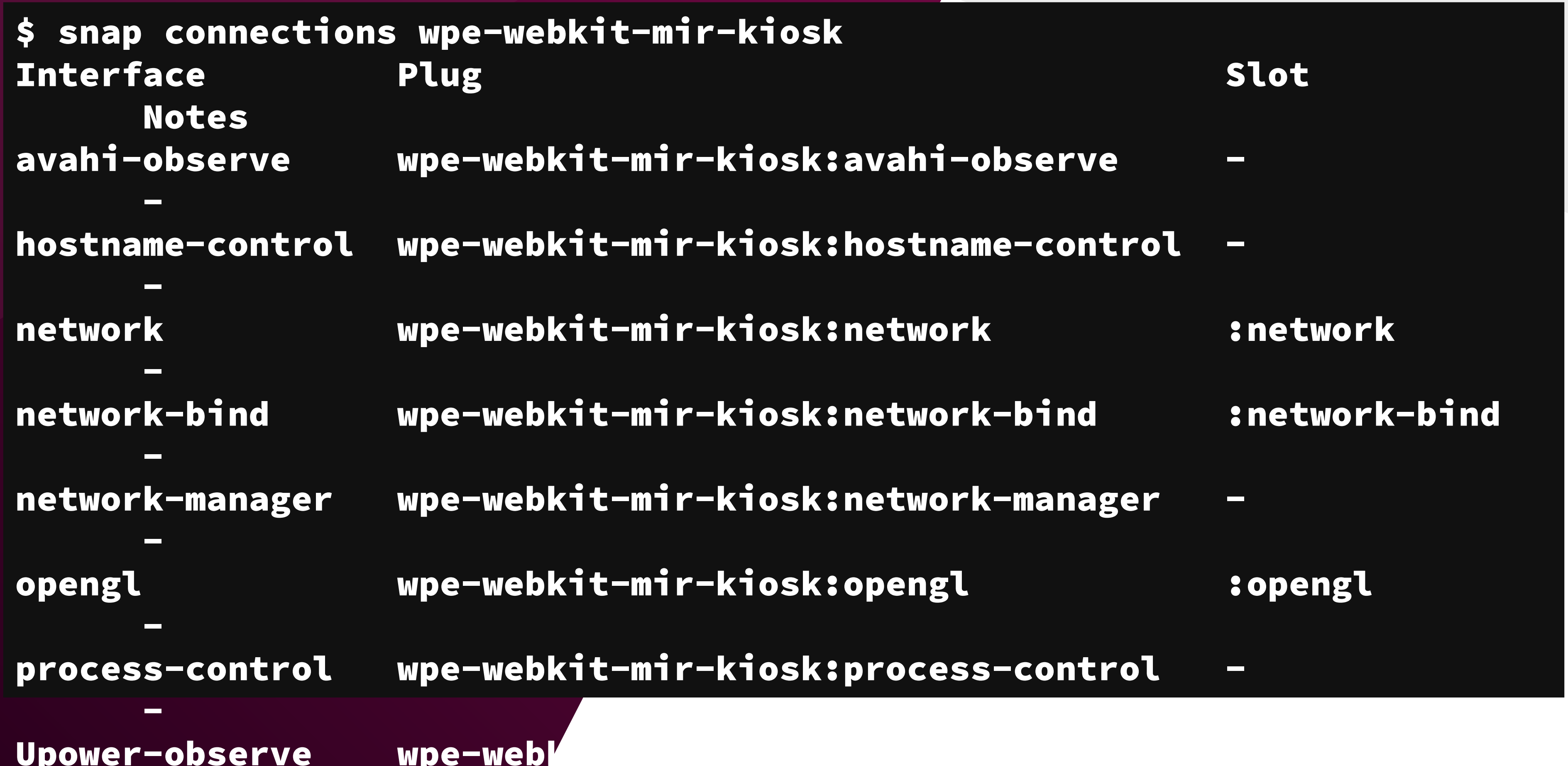
- ➢ Network
- ➢ Graphics

Shell and apps are different



```
$ snap connections mir-kiosk
Interface  Plug                          Slot                   Notes
opengl     mir-kiosk:opengl              :opengl                -
wayland    wpe-webkit-mir-kiosk:wayland  mir-kiosk:wayland      manual
x11        mir-kiosk:x11
```

Shell and apps are different



```
$ snap connections wpe-webkit-mir-kiosk
Interface         Plug                                   Slot
      Notes
avahi-observe     wpe-webkit-mir-kiosk:avahi-observe     -
      -
hostname-control  wpe-webkit-mir-kiosk:hostname-control  -
      -
network           wpe-webkit-mir-kiosk:network           :network
      -
network-bind      wpe-webkit-mir-kiosk:network-bind      :network-bind
      -
network-manager   wpe-webkit-mir-kiosk:network-manager   -
      -
opengl            wpe-webkit-mir-kiosk:opengl            :opengl
      -
process-control   wpe-webkit-mir-kiosk:process-control   -
      -
Upower-observe    wpe-webk
```

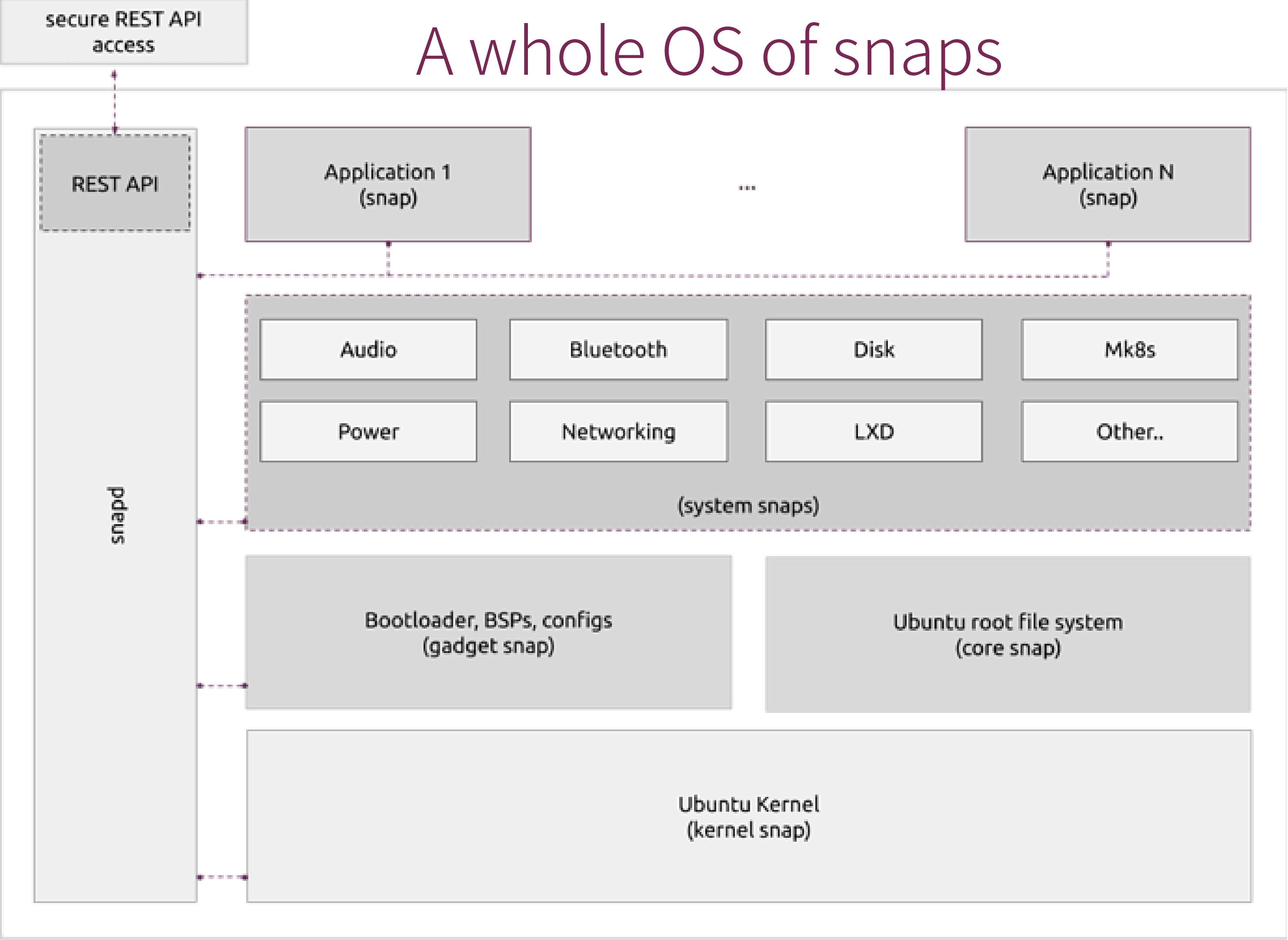
Ubuntu Core

A whole OS of snaps

Confinement can be applied to more than apps and shells.

A whole operating system can be built with this technology

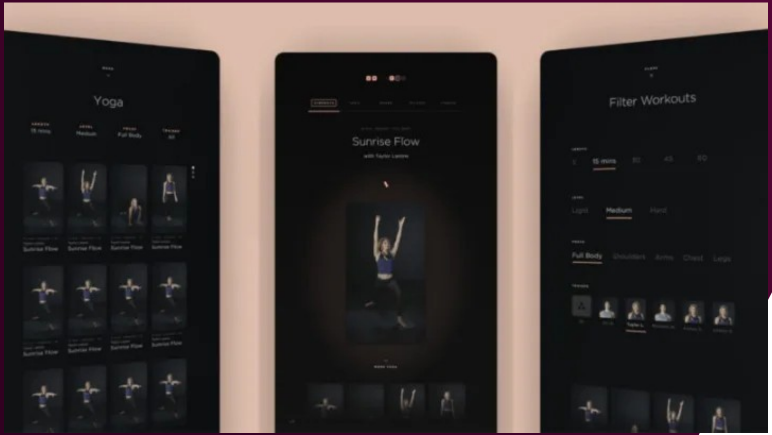
A whole OS of snaps



Graphical shell

Forme Life is a company based in California developing Studio, the full-length mirror that transforms into personal training.

Mir-kiosk was used on this innovative mirror to provide the foundation for the graphical implementation.



More Mir on mirrors

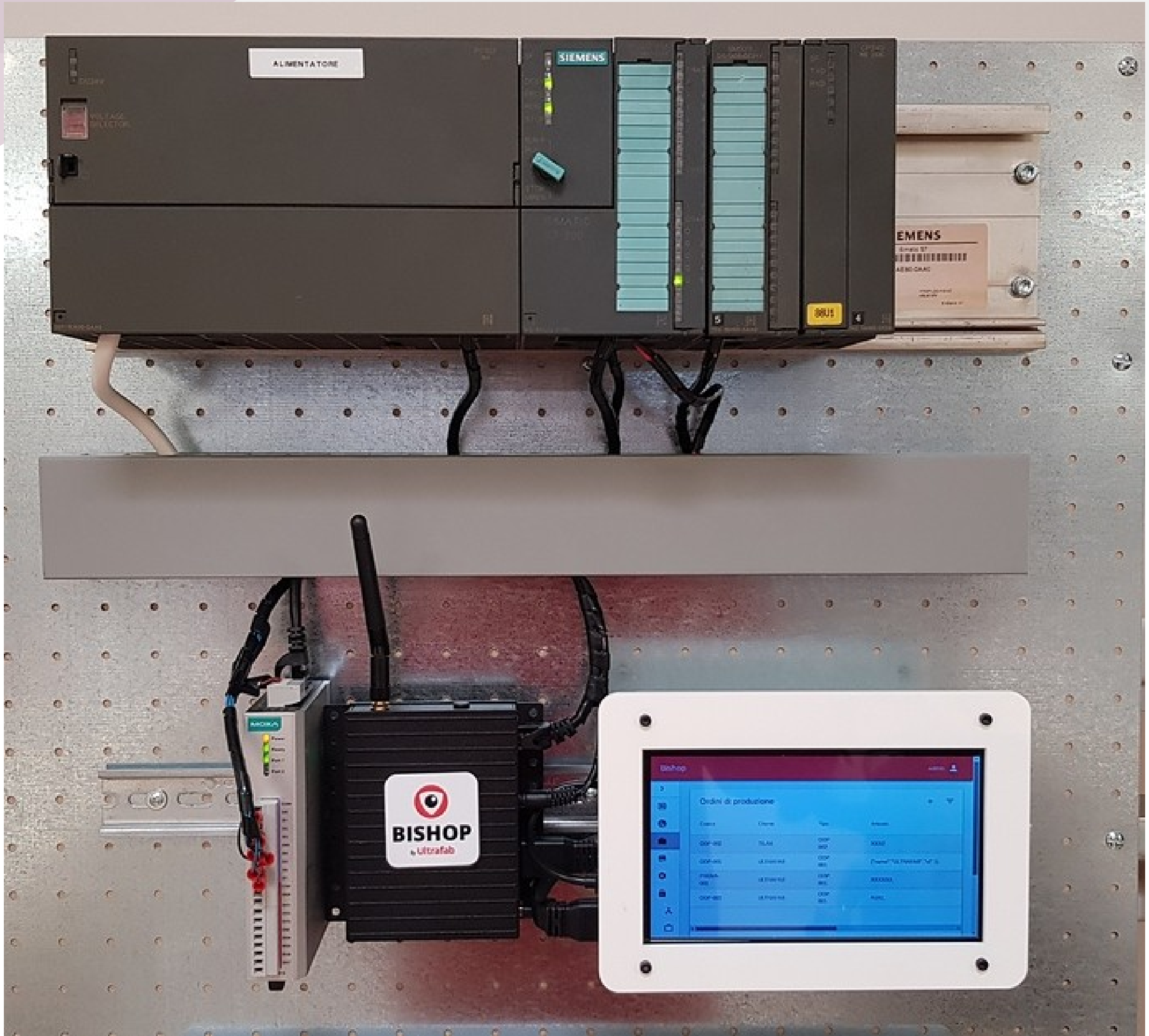
mirr.OS one is the further development of the individual smart home concept. The completely revised system now adapts even better to your needs. mirr.OS one comes with its own web app and uses the new security advantages of Ubuntu Core. On the new board with a grid you can arrange your widgets how you want and as often as you want.

glancr.de

Embedded in IoT devices

This is a picture of a test gateway running a Raspberry CM3 module, connected to a Siemens S7-300 PLC through a MOXA E1212 remote-io

http://tiny.cc/85ebtz



Different shells

Kiosk mode

We’ve see few examples of a minimal shell…

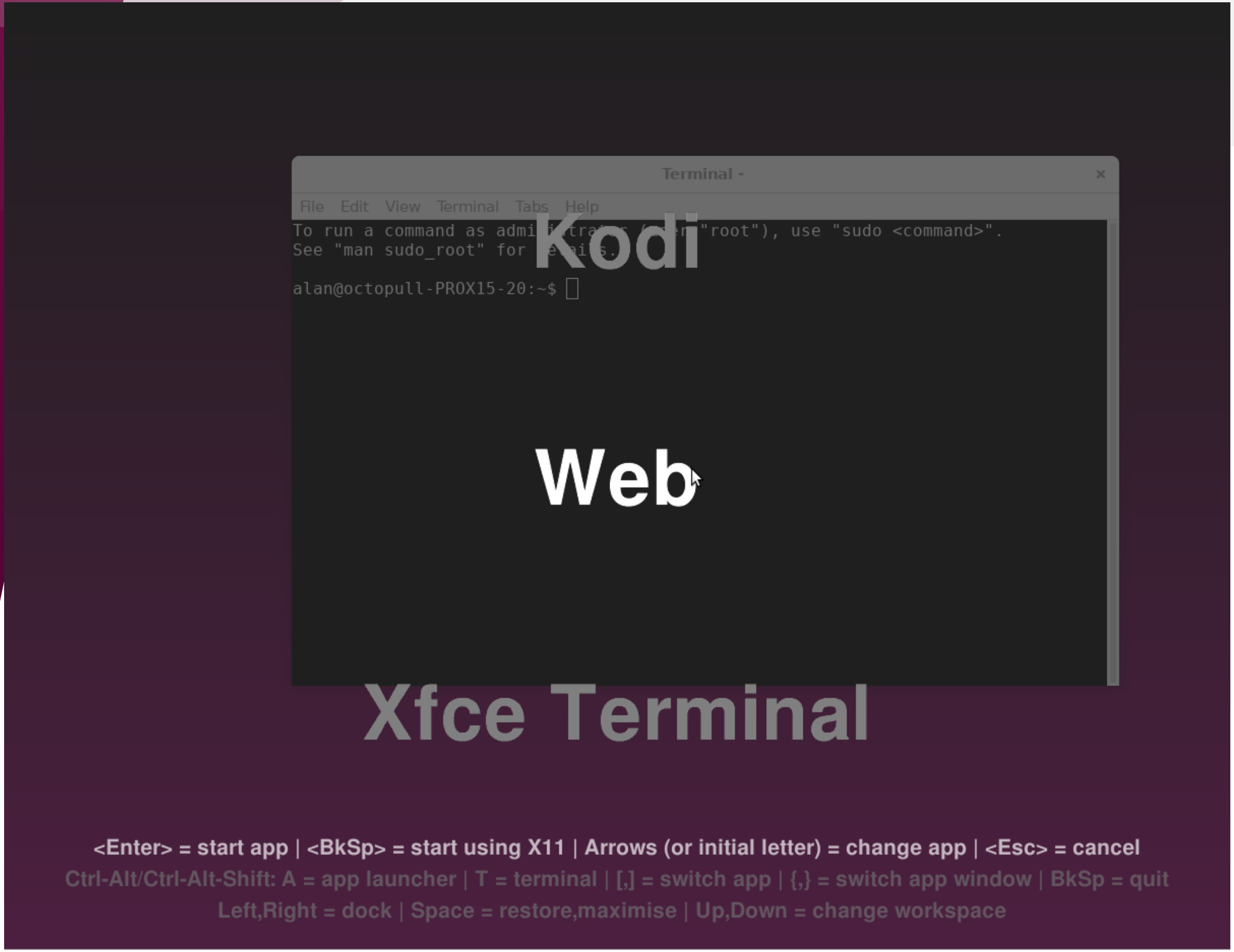
- ➢ A single fullscreen app
- ➢ Launched automatically

Desktop environment

- ➢ Multiple windowed applications
- ➢ Launched by the user

Egmde shell

Egmde is an “example desktop environment” used for testing and demonstration



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

alan@octopull-PROX15-20:~$ 
```

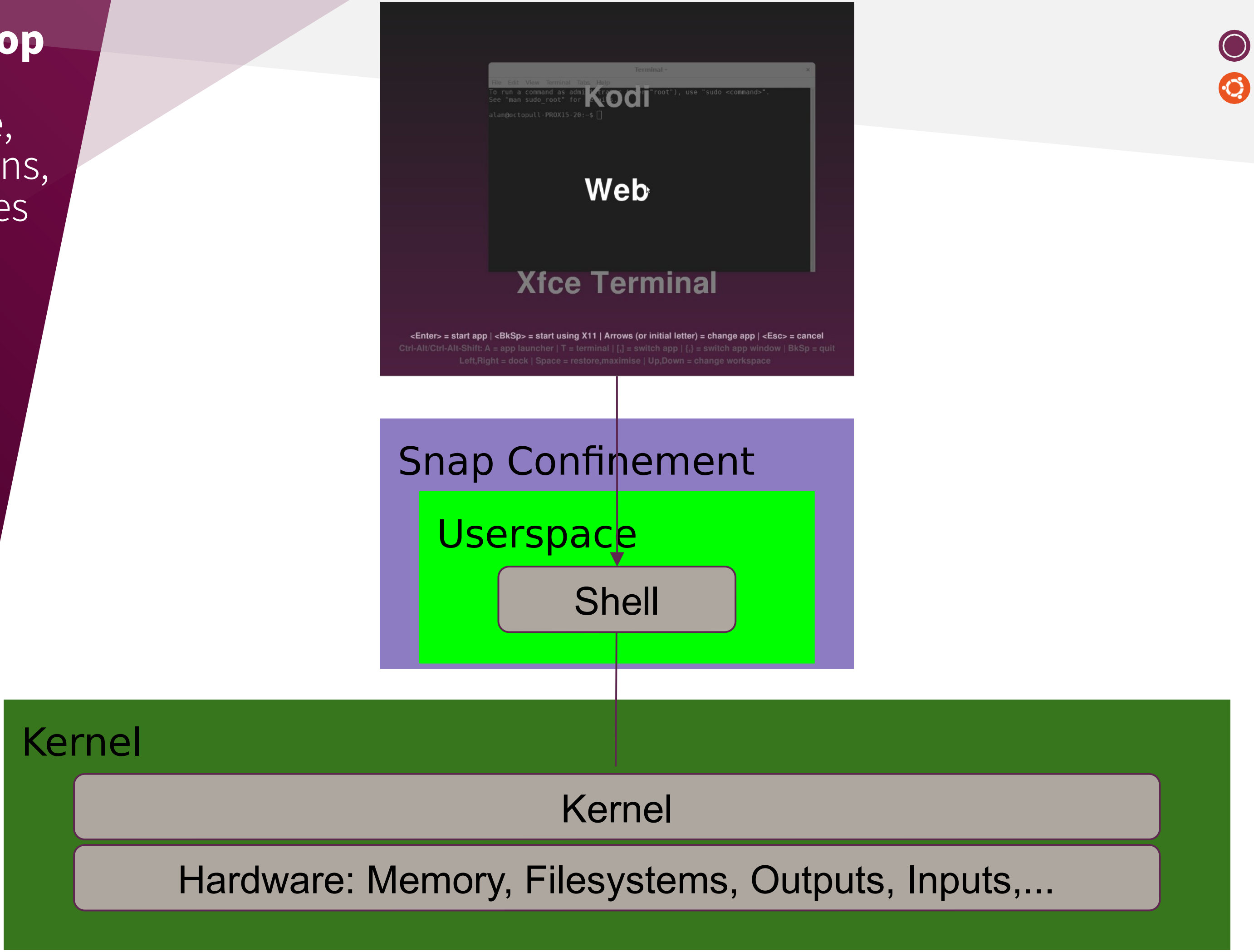
<Enter> = start app | <BkSp> = start using X11 | Arrows (or initial letter) = change app | <Esc> = cancel

Ctrl-Alt/Ctrl-Alt-Shift: A = app launcher | T = terminal | [,] = switch app | {,} = switch app window | BkSp = quit

Left,Right = dock | Space = restore,maximise | Up,Down = change workspace

egmde-confined-desktop

This snap confines egmde, and a variety of applications, to illustrate the possibilities and limitations



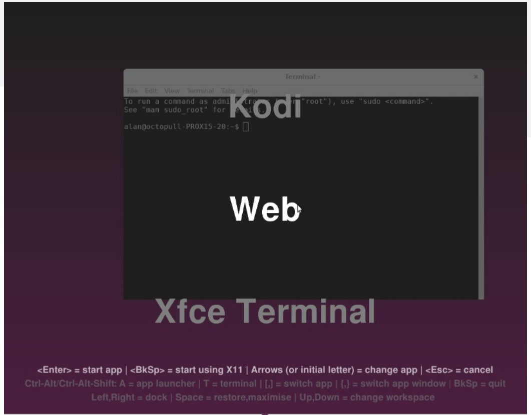
```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

alan@octopull-PROX15-20:~$
```

<Enter> = start app | <BkSp> = start using X11 | Arrows (or initial letter) = change app | <Esc> = cancel
Ctrl-Alt/Ctrl-Alt-Shift: A = app launcher | T = terminal | [,] = switch app | {,} = switch app window | BkSp = quit
Left,Right = dock | Space = restore,maximise | Up,Down = change workspace

Terminal -
File Edit View Terminal Tabs Help
Kodi
Web
Xfce Terminal
Snap Confinement
Userspace
Shell
Kernel
Kernel
Hardware: Memory, Filesystems, Outputs, Inputs,...

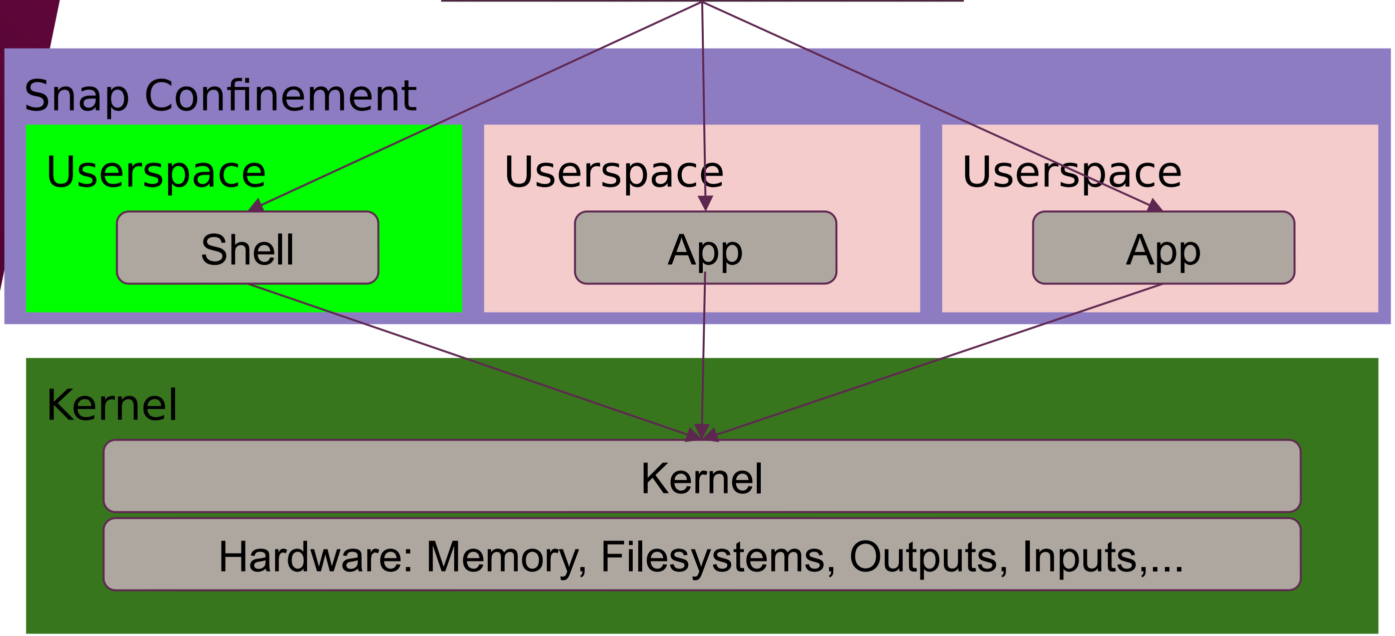
egmde-confined-desktop

This snap confines egmde, and a variety of applications, to illustrate the possibilities and limitations



```
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

alan@octopull-PROX15-20:~$ 
```

<Enter> = start app | <BkSp> = start using X11 | Arrows (or initial letter) = change app | <Esc> = cancel
Ctrl-Alt/Ctrl-Alt-Shift: A = app launcher | T = terminal | [,] = switch app | {,} = switch app window | BkSp = quit
Left,Right = dock | Space = restore,maximise | Up,Down = change workspace



Snap Confinement

Userspace

Shell

Userspace

App

Userspace

App

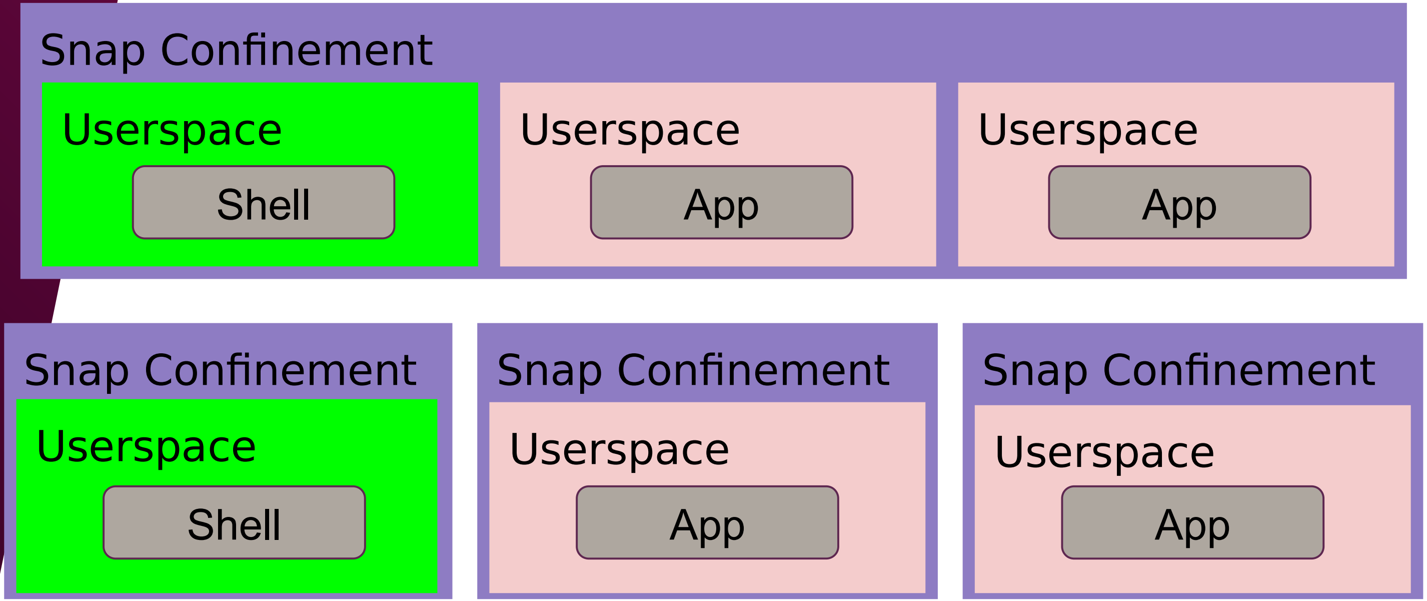
Kernel

Kernel

Hardware: Memory, Filesystems, Outputs, Inputs,...

egmde-confined-desktop

Having to package and confine all the applications and shell in a single snap is a limitation, and I'll talk about that shortly.

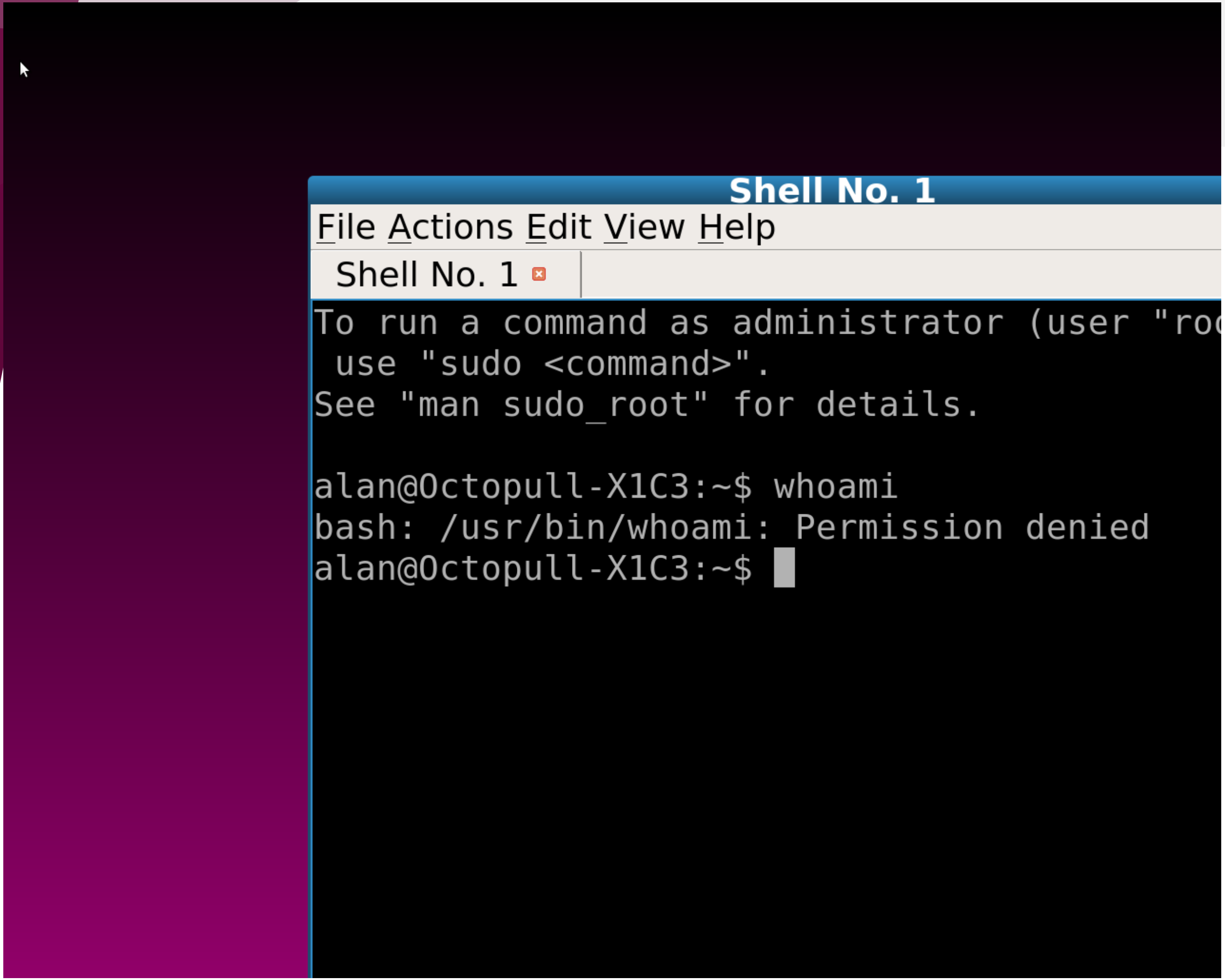


egmde-confined-desktop

[DEMO] confined “desktop” shell

egmde-confined-desktop

We can run a terminal emulator included in the snap

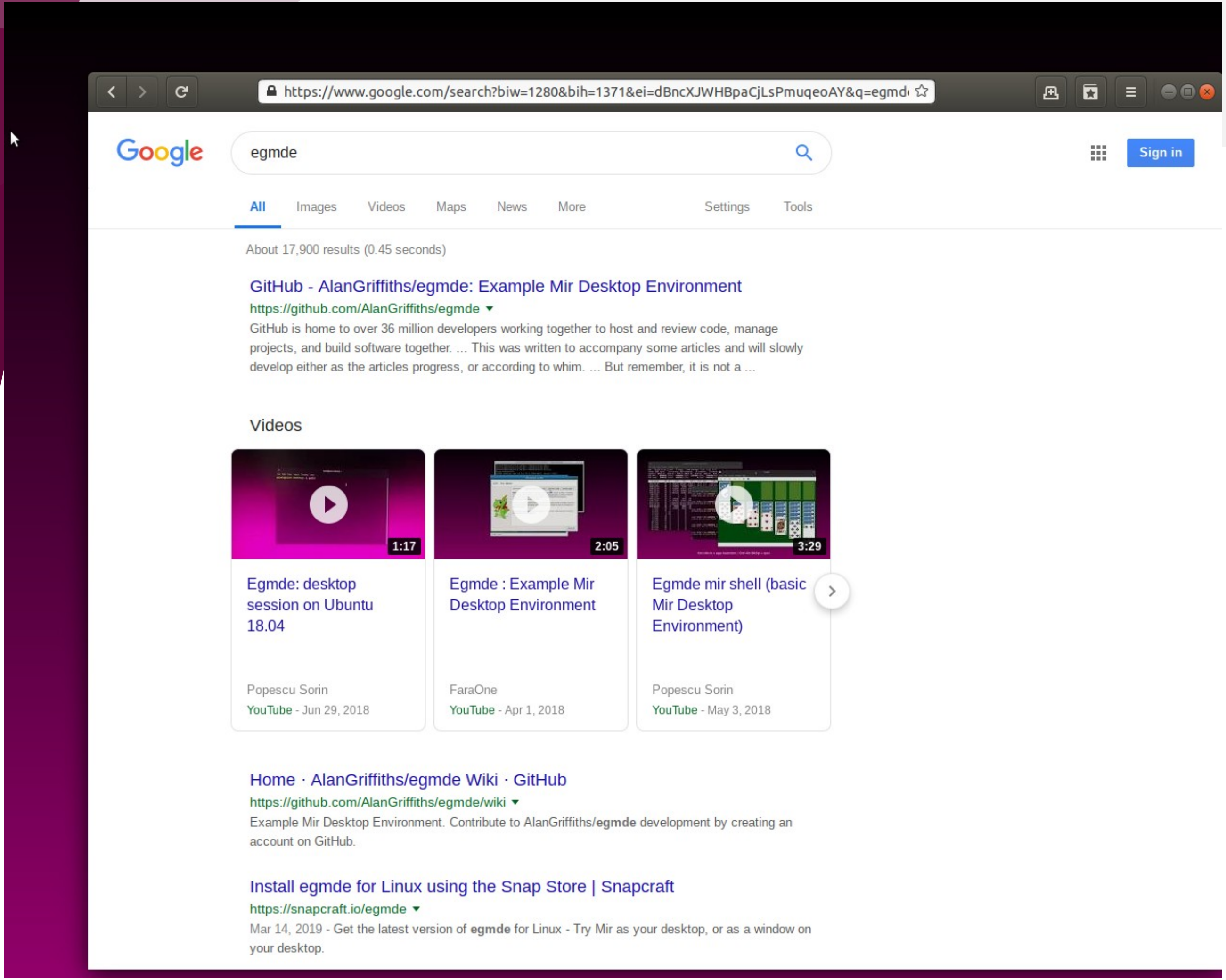


```
To run a command as administrator (user "roo
 use "sudo <command>".
See "man sudo_root" for details.

alan@Octopull-X1C3:~$ whoami
bash: /usr/bin/whoami: Permission denied
alan@Octopull-X1C3:~$ 
```

egmde-confined-desktop

We can run a browser included in the snap

The confinement restrictions apply to the browser, so even if compromised by a website it cannot access the host environment



https://www.google.com/search?biw=1280&bih=1371&ei=dBncXJWHBpaCjLsPmuqeoAY&q=egmd
Google
egmde
Sign in
All
Images
Videos
Maps
News
More
Settings
Tools

About 17,900 results (0.45 seconds)

GitHub - AlanGriffiths/egmde: Example Mir Desktop Environment
https://github.com/AlanGriffiths/egmde ▾
GitHub is home to over 36 million developers working together to host and review code, manage projects, and build software together. ... This was written to accompany some articles and will slowly develop either as the articles progress, or according to whim. ... But remember, it is not a ...

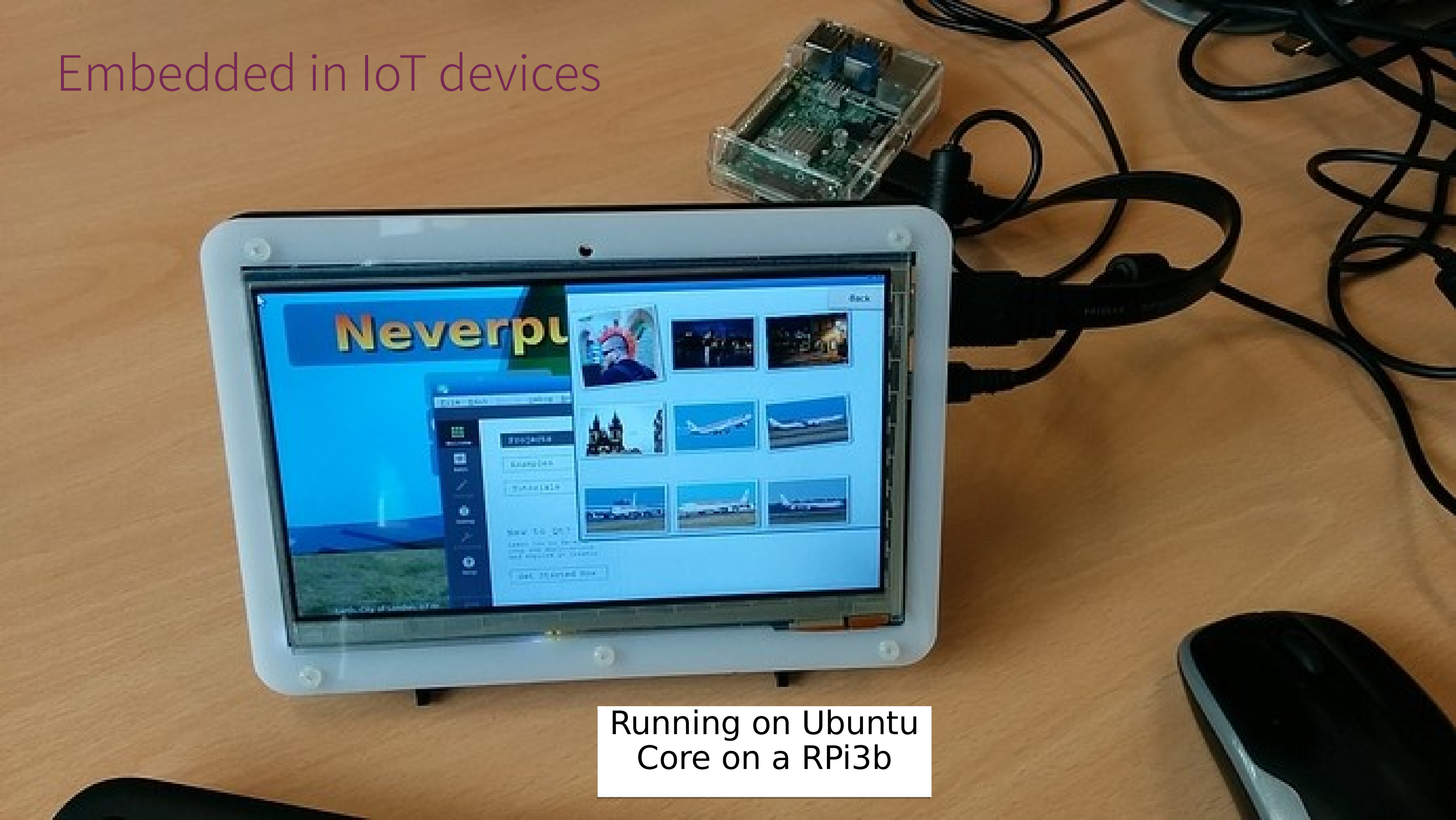
Videos

1:17	2:05	3:29
Egmde: desktop session on Ubuntu 18.04	Egmde : Example Mir Desktop Environment	Egmde mir shell (basic Mir Desktop Environment)
Popescu Sorin	FaraOne	Popescu Sorin
YouTube - Jun 29, 2018	YouTube - Apr 1, 2018	YouTube - May 3, 2018

Home · AlanGriffiths/egmde Wiki · GitHub
https://github.com/AlanGriffiths/egmde/wiki ▾
Example Mir Desktop Environment. Contribute to AlanGriffiths/**egmde** development by creating an account on GitHub.

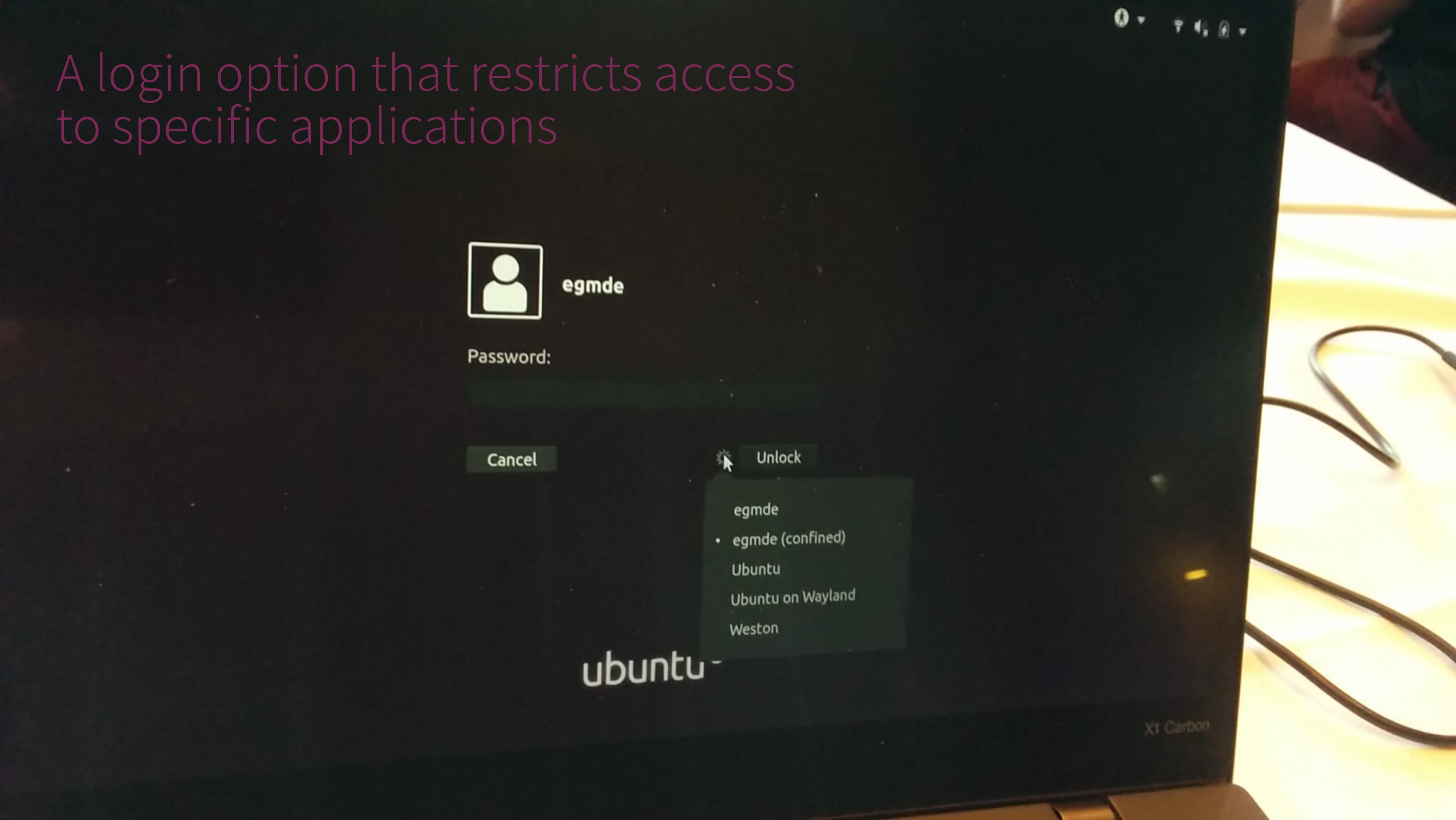
Install egmde for Linux using the Snap Store | Snapcraft
https://snapcraft.io/egmde ▾
Mar 14, 2019 - Get the latest version of **egmde** for Linux - Try Mir as your desktop, or as a window on your desktop.

Embedded in IoT devices

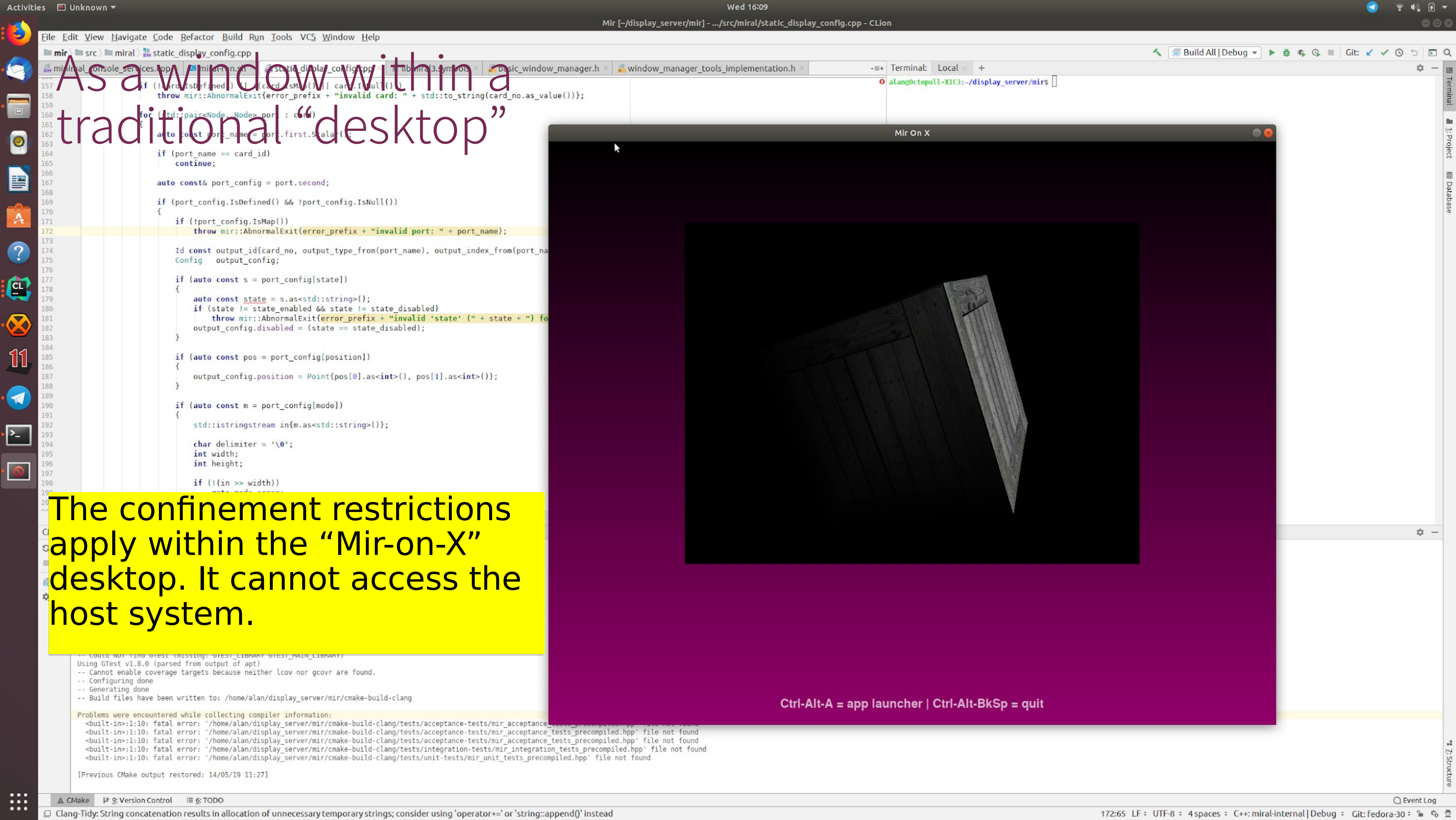


Running on Ubuntu Core on a RPi3b

A login option that restricts access to specific applications



As a window within a traditional “desktop”


Mir [~/display_server/mir] - .../src/miral/static_display_config.cpp - CLion
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
mir › src › miral › static_display_config.cpp
minimal_console_services.cpp
miral-run.sh
static_display_config.cpp
libmiral3.symbols
basic_window_manager.h
window_manager_tools_implementation.h
Build All | Debug
Terminal: Local
Mir On X
CMake
9: Version Control
6: TODO
Event Log

```
        if (!card.IsDefined() || !(card.IsMap() || card.IsNull()))
            throw mir::AbnormalExit{error_prefix + "invalid card: " + std::to_string(card_no.as_value())};

        for (std::pair<Node, Node> port : card)
        {
            auto const port_name = port.first.Scalar();

            if (port_name == card_id)
                continue;

            auto const& port_config = port.second;

            if (port_config.IsDefined() && !port_config.IsNull())
            {
                if (!port_config.IsMap())
                    throw mir::AbnormalExit{error_prefix + "invalid port: " + port_name};

                Id const output_id{card_no, output_type_from(port_name), output_index_from(port_na
                Config   output_config;

                if (auto const s = port_config[state])
                {
                    auto const state = s.as<std::string>();
                    if (state != state_enabled && state != state_disabled)
                        throw mir::AbnormalExit{error_prefix + "invalid 'state' (" + state + ") fo
                    output_config.disabled = (state == state_disabled);
                }

                if (auto const pos = port_config[position])
                {
                    output_config.position = Point{pos[0].as<int>(), pos[1].as<int>()};
                }

                if (auto const m = port_config[mode])
                {
                    std::istringstream in{m.as<std::string>()};

                    char delimiter = '\0';
                    int width;
                    int height;

                    if (!(in >> width))
```

```
alan@Octopull-X1C3:~/display_server/mir$
```

Ctrl-Alt-A = app launcher | Ctrl-Alt-BkSp = quit

```
Using GTest v1.8.0 (parsed from output of apt)
-- Cannot enable coverage targets because neither lcov nor gcovr are found.
-- Configuring done
-- Generating done
-- Build files have been written to: /home/alan/display_server/mir/cmake-build-clang

Problems were encountered while collecting compiler information:
  <built-in>:1:10: fatal error: '/home/alan/display_server/mir/cmake-build-clang/tests/acceptance-tests/mir_acceptance
  <built-in>:1:10: fatal error: '/home/alan/display_server/mir/cmake-build-clang/tests/acceptance-tests/mir_acceptance_tests_precompiled.hpp' file not found
  <built-in>:1:10: fatal error: '/home/alan/display_server/mir/cmake-build-clang/tests/acceptance-tests/mir_acceptance_tests_precompiled.hpp' file not found
  <built-in>:1:10: fatal error: '/home/alan/display_server/mir/cmake-build-clang/tests/integration-tests/mir_integration_tests_precompiled.hpp' file not found
  <built-in>:1:10: fatal error: '/home/alan/display_server/mir/cmake-build-clang/tests/unit-tests/mir_unit_tests_precompiled.hpp' file not found

[Previous CMake output restored: 14/05/19 11:27]
```

Clang-Tidy: String concatenation results in allocation of unnecessary temporary strings; consider using 'operator+=' or 'string::append()' instead

172:65 LF ÷ UTF-8 ÷ 4 spaces ÷ C++: miral-internal | Debug ÷ Git: fedora-30 ÷

The confinement restrictions apply within the “Mir-on-X” desktop. It cannot access the host system.

egmde-confined-desktop

The “egmde-confined-desktop” snap is a proof-of-concept, not a finished product

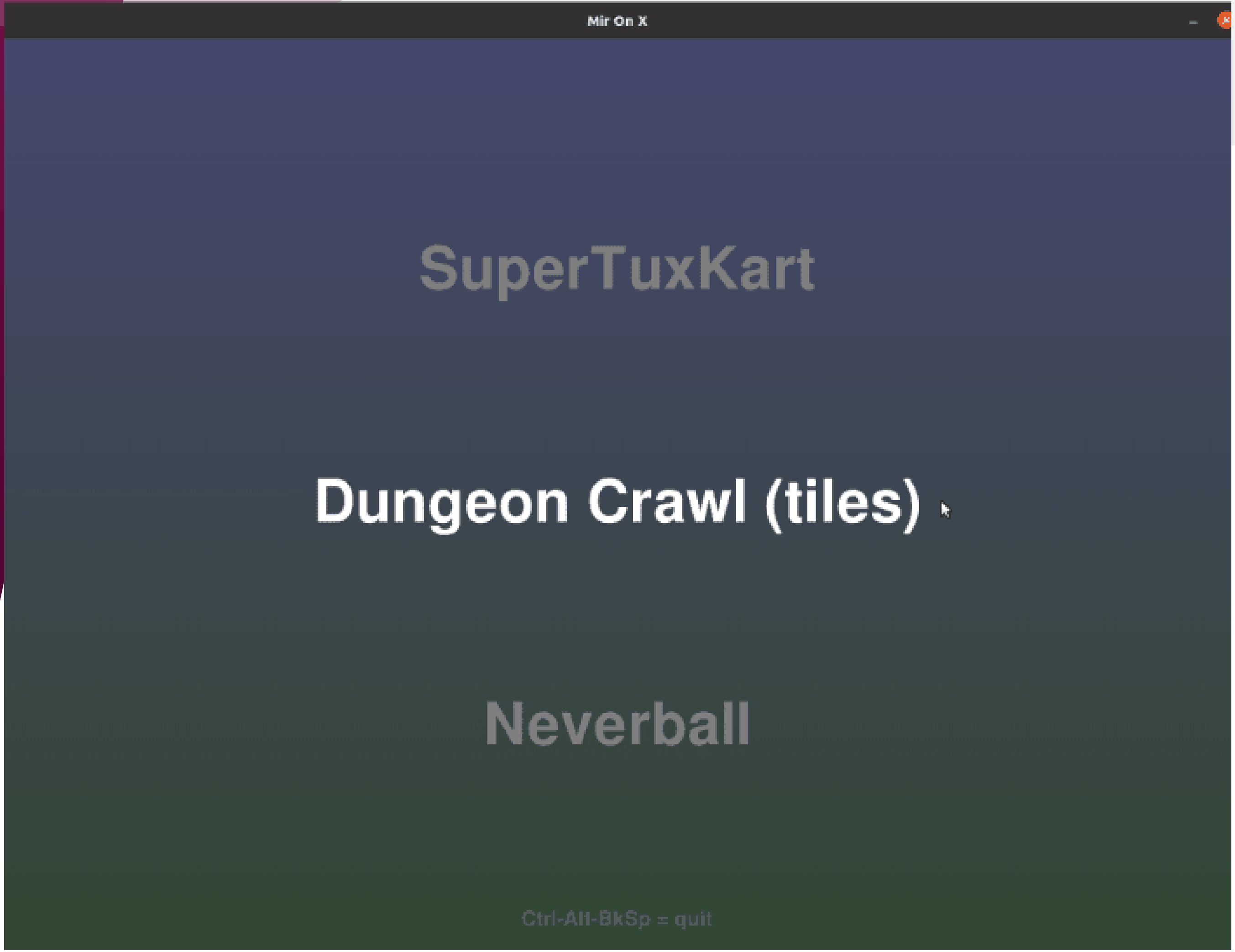
- ➢ A confined desktop environment
  - ○ On Ubuntu Core
  - ○ On “classic” Linux (where snapd is supported)
  - ○ A variety of applications included

Including a bespoke set of applications in a new snap is the simplest way to customize this

mircade

Mircade is a example snap based on a modified egmde and some games from the Ubuntu archive

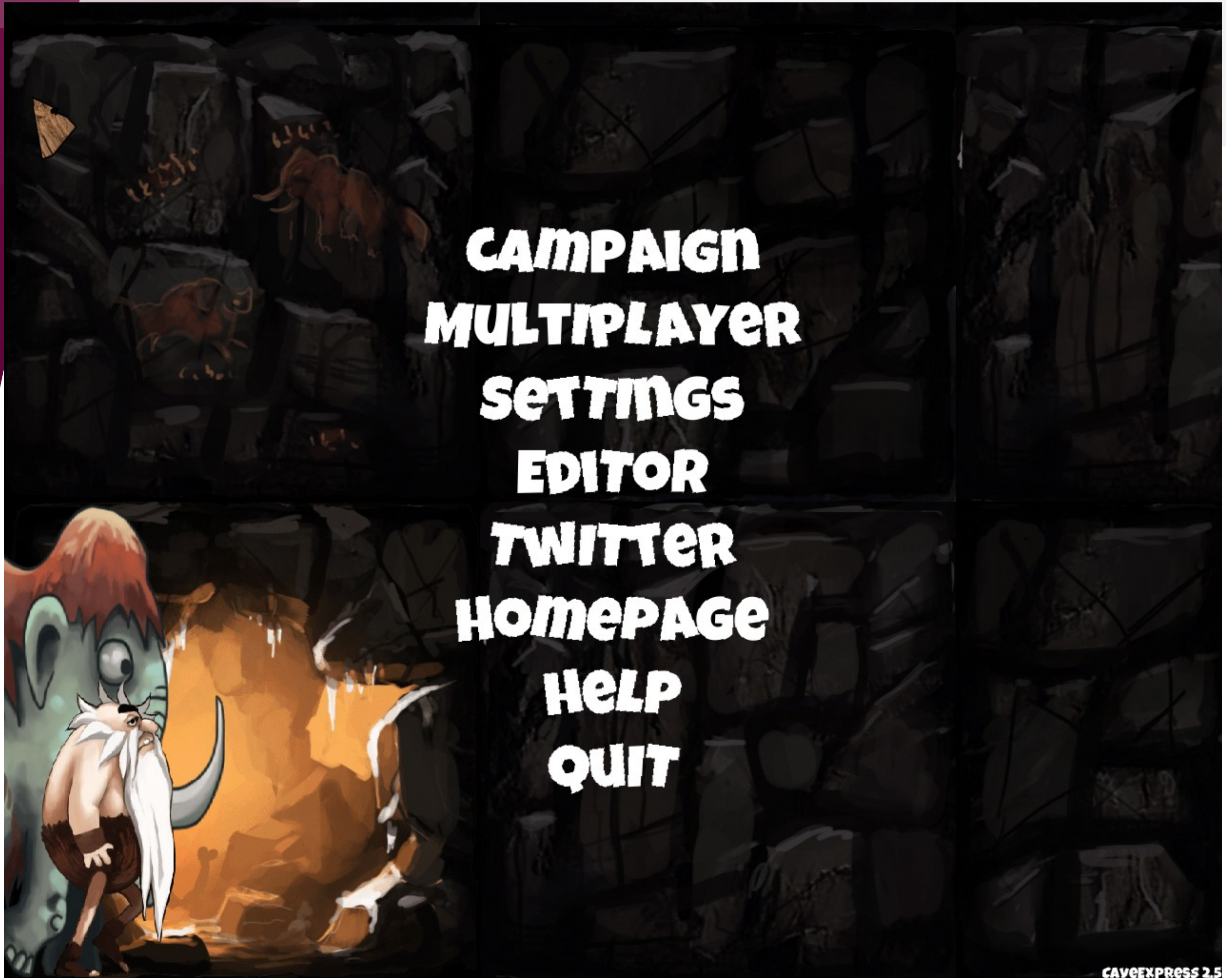
This shell launches a single fullscreen app



mircade

Mircade is a example snap based on a modified egmde and some games from the Ubuntu archive

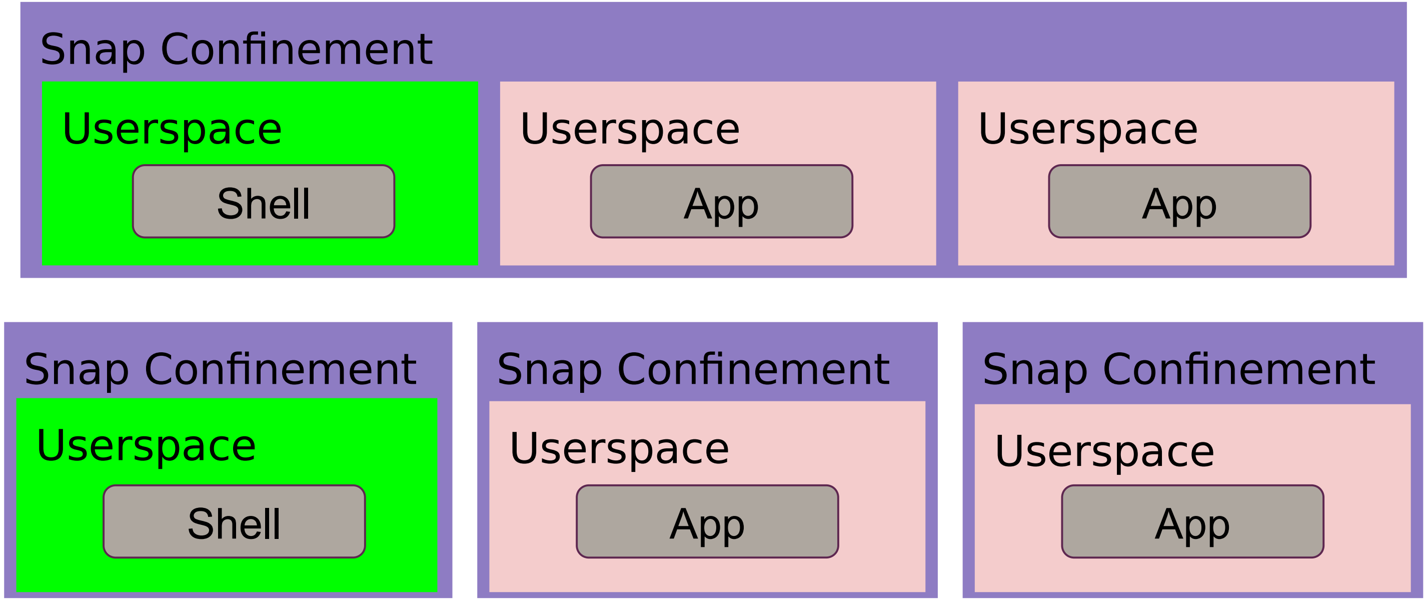
This shell launches a single fullscreen app



What use is a confined user shell?

Having to package and confine all the applications in a shell snap is a limitation…

…and I'll talk about that now!



Shell and apps are different

A shell: needs access to...

- ➢ User input and output
- ➢ Launching apps

A desktop environment...

- ➢ Helpers for keyring & policy kit
- ➢ Screensaver, screen lock, suspend, logout, shutdown

An app needs access to...

- ➢ $HOME directory & your files
- ➢ Network
- ➢ Removable media
- ➢ Other devices & filesystems

Shell and apps are different

A shell: needs access to...

- User input and output
- Launching apps

A desktop environment...

- Helpers for keyring & policy kit
- Screensaver, screen lock, suspend, logout, shutdown

An app needs access to...

- $HOME directory & your files
- Network
- Removable media
- Other devices & filesystems

✓

Shell and apps are different

A shell: needs access to…

- ➢ User input and output ✓
- ➢ Launching apps (currently only within the same snap)

A desktop environment…

- ➢ Helpers for keyring & policy kit ✕
- ➢ Screensaver, screen lock, suspend, logout, shutdown ✕

Launching apps

There are several issues to be addressed in order to launch apps...

- Identifying the available apps
  - There is a standard
    - *The "Desktop Entry Specification"*

      https://specifications.freedesktop.org/desktop-entry-spec/desktop-entry-spec-latest.html
    - But how does it apply in a confined environment?
- Confined snaps cannot directly invoke other snaps

Launching apps

Confined snaps cannot directly invoke other snaps

- ➢ But there is a “userd” process that can…
  - ○ …so we can send it a message
  - ○ But, how does userd “police” the requests?
  - ○ And, userd only runs on “Classic” systems
- ➢ There’s “Prior art” in Ubuntu Touch
  - ○ Clicks have lomiri-app-launch (formerly ubuntu-app-launch)

What use is a confined user shell?

Mir-kiosk

- “Kiosk” shell
  - On Ubuntu Core
  - Apps must “launch themselves”
    - E.g. **wpe-webkit-mir-kiosk**

Egmde-confined-desktop

- A confined desktop environment
  - A variety of applications included
  - On classic Linux
  - On Ubuntu Core

Mircade

- Bespoke shell with some games
  - On classic Linux
  - On Ubuntu Core

Future directions

We’re working on ways to enable other snaps to be launched from within a confined snap

Other desktop environments could be confined with some effort

Before the “hands on” Questions?

Making a confined user shell

To “play along” you need a computer with:

1. Linux
2. Snaps working:

   https://snapcraft.io/docs/installing-snapd
3. Git installed
4. A working internet connection

Making a confined shell

Installing the build tools

- Snapcraft
- Multipass

```
$ snap install --classic snapcraft
$ snap install --classic multipass
```

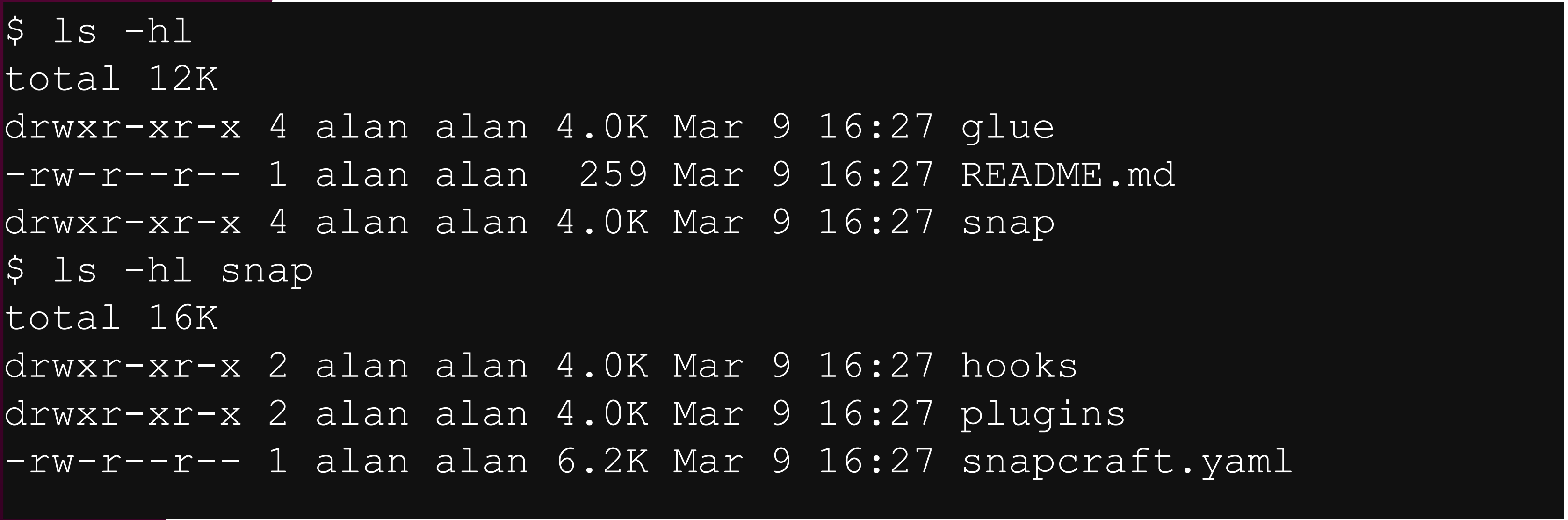
Making a confined shell

Cloning the confined desktop example

- ➢ Get egmde confined desktop
- ➢ Switch to the project directory

```
$ git clone \
https://github.com/MirServer/egmde-confined-desktop.git

$ cd egmde-confined-desktop
```

Making a confined shell



```
$ ls -hl
total 12K
drwxr-xr-x 4 alan alan 4.0K Mar 9 16:27 glue
-rw-r--r-- 1 alan alan  259 Mar 9 16:27 README.md
drwxr-xr-x 4 alan alan 4.0K Mar 9 16:27 snap
$ ls -hl snap
total 16K
drwxr-xr-x 2 alan alan 4.0K Mar 9 16:27 hooks
drwxr-xr-x 2 alan alan 4.0K Mar 9 16:27 plugins
-rw-r--r-- 1 alan alan 6.2K Mar 9 16:27 snapcraft.yaml
```

Making a confined shell

```
$ snapcraft
Launching a VM.
…
Snapped egmde-confined-desktop_139-mir2.3.2-snap80_amd64.snap

$ snap install --dangerous *.snap
egmde-confined-desktop 139-mir2.3.2-snap80 installed

$ /snap/egmde-confined-desktop/current/bin/setup.sh
…

$ egmde-confined-desktop
```

egmde-confined-desktop

A look at the snapcraft.yaml

Making a confined shell

egmde-confined-desktop

The “egmde-confined-desktop” snap is a proof-of-concept, not a finished product

- A confined desktop environment
  - On Ubuntu Core
  - On “classic” Linux (where snapd is supported)
  - A variety of applications included

Including a bespoke set of applications in a new snap is the simplest way to customize this

Future directions

Other snapped applications can be run on the “egmde-confined-desktop” desktop, but need to be launched from outside the snap. We’re investigating ways to enable other snaps to be launched from within a confined snap

Other desktop environments could be confined with some effort. There’s work being done with GNOME

Thank you. Questions?