



# A Look Inside Mutter / GNOME Shell

**Georges Stavracas**

**Endless OS Foundation**

Hi



# Mutter

## What's Mutter

- Based on Clutter
  - Mutter = Metacity + Clutter
- A library to write compositors
- A tiny compositor
- Supports plugins
  - GNOME Shell is a Mutter plugin
- X11 / Wayland

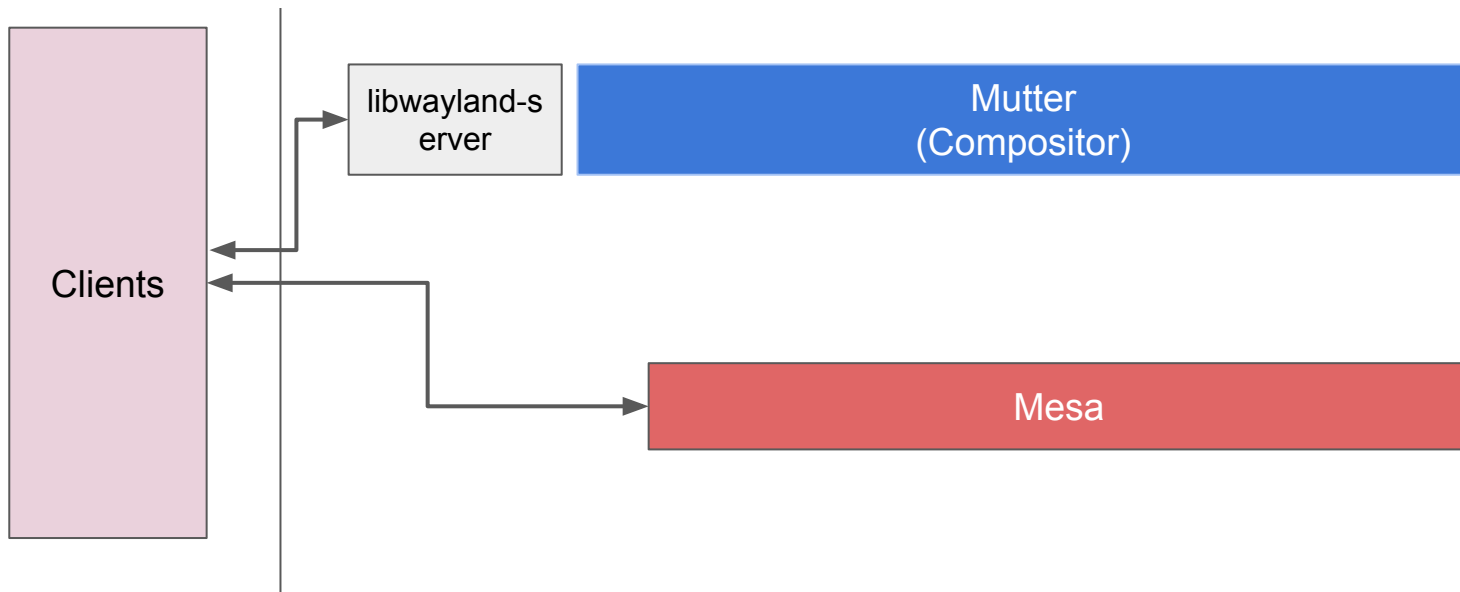
## Mutter, Clutter, Cogl, What?

Mutter  
(Compositor)

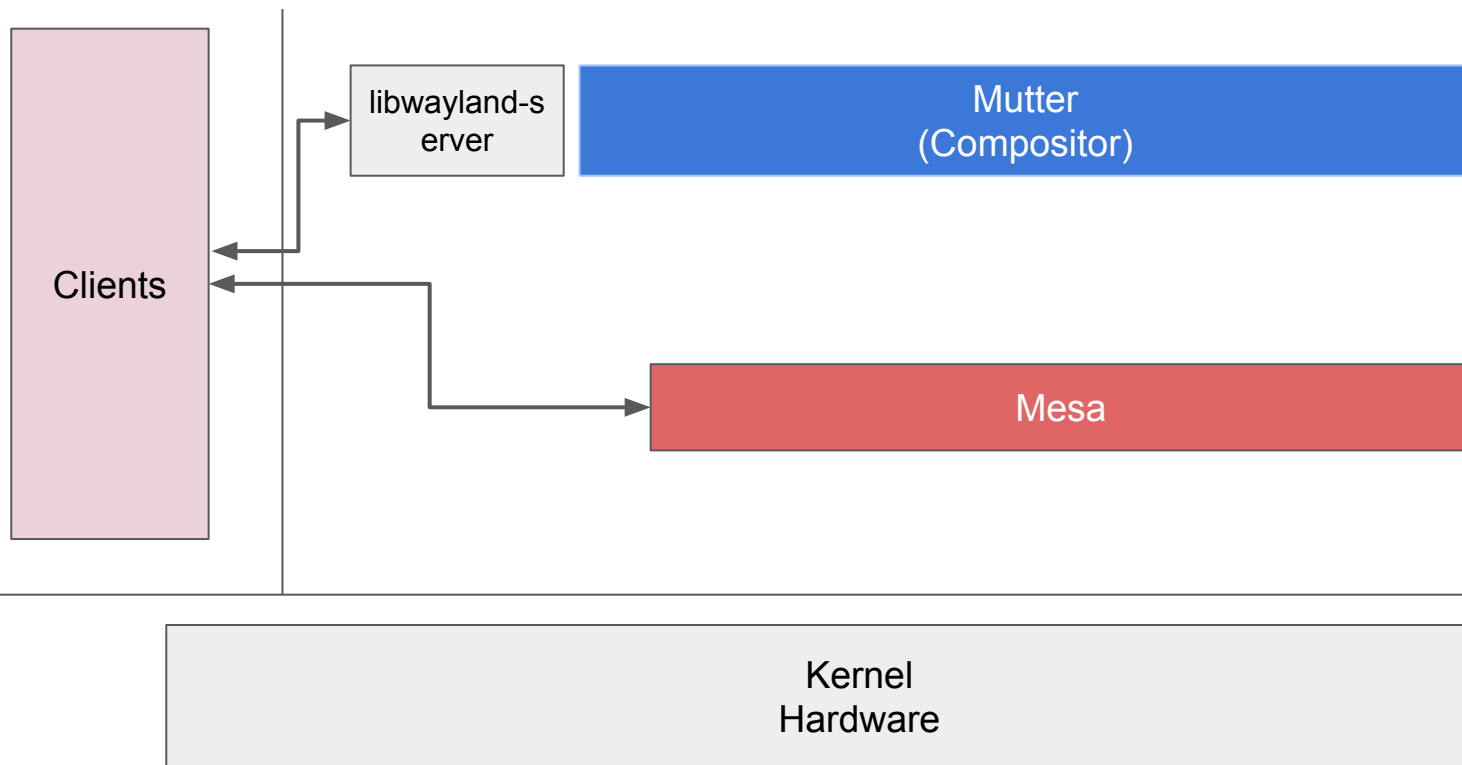
## Mutter, Clutter, Cogl, What?



## Mutter, Clutter, Cogl, What?

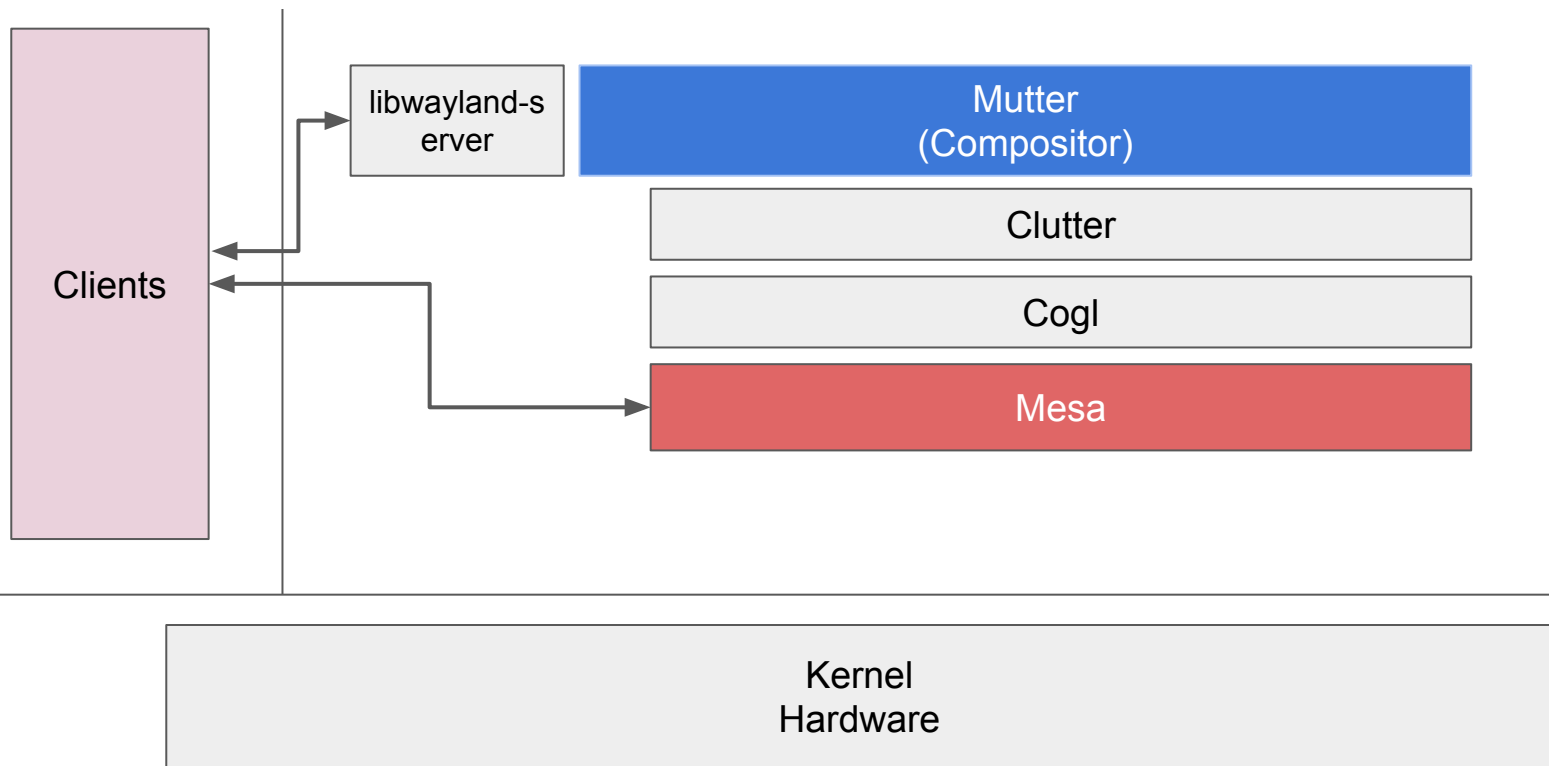


## Mutter, Clutter, Cogl, What?

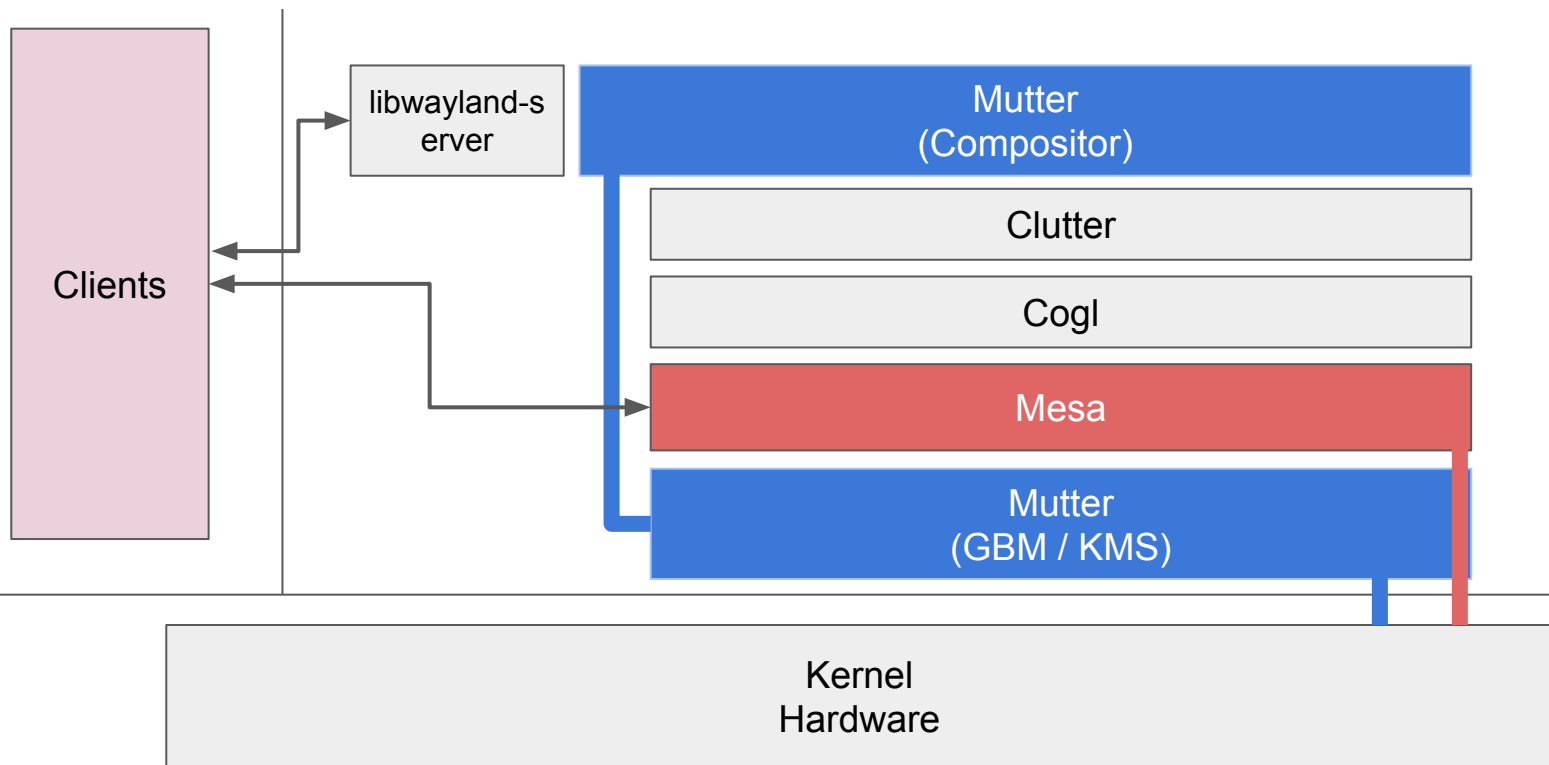




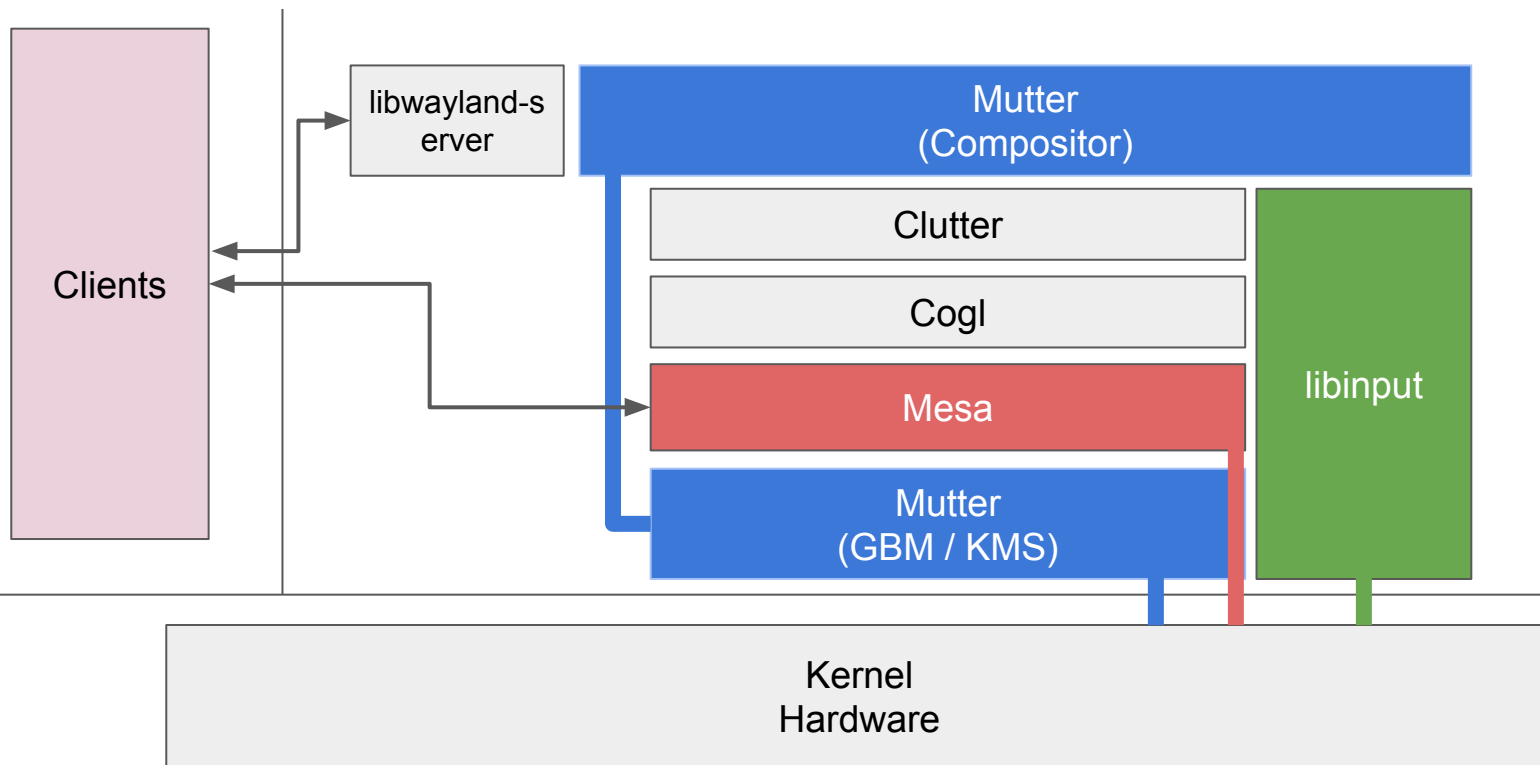
## Mutter, Clutter, Cogl, What?



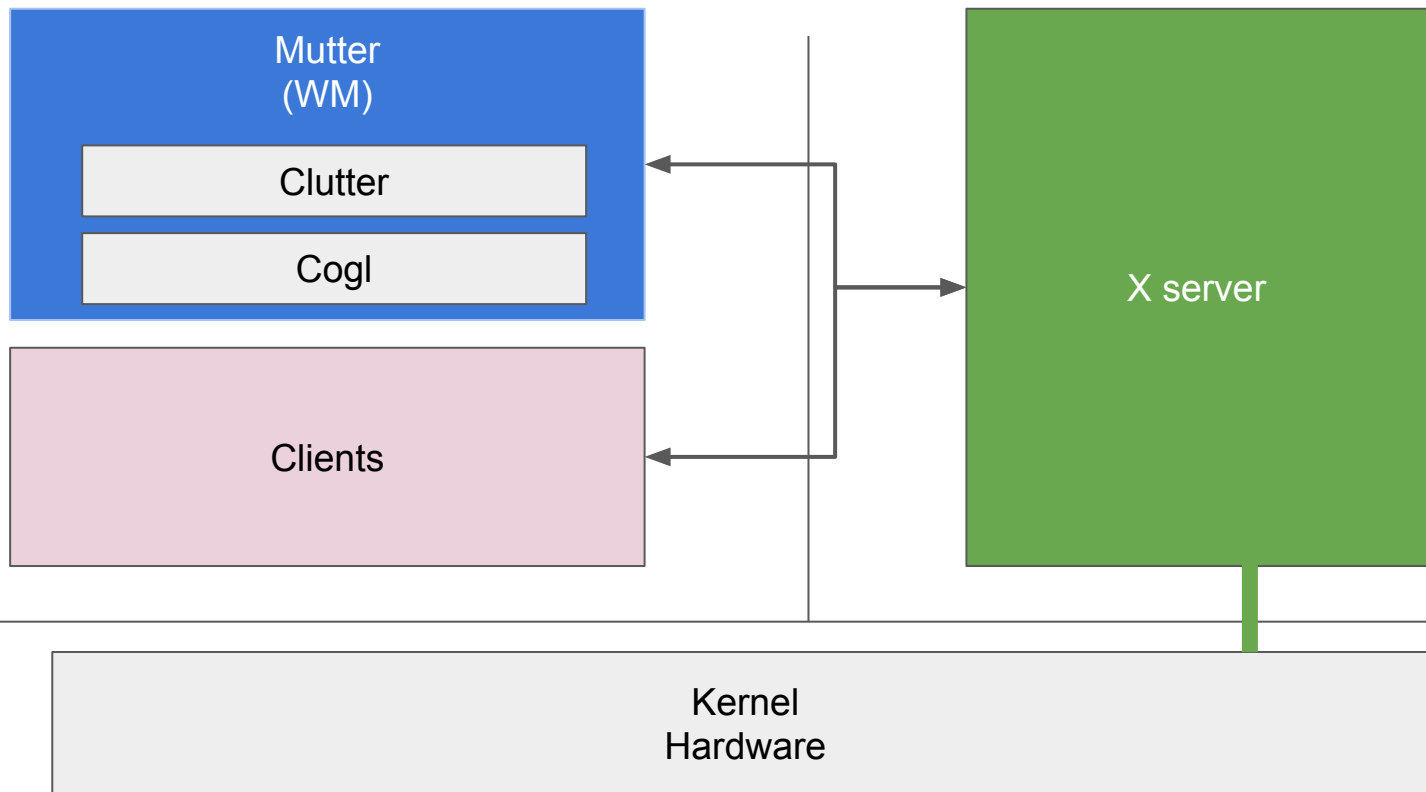
## Mutter, Clutter, Cogl, What?



## Mutter, Clutter, Cogl, What?



## On X11...



**Cogl**

# Cogl

An abstraction layer over OpenGL

- Allows dealing with GL in an object-oriented manner
- Predates Vulkan by many years, yet surprisingly similar!
- Was an independent project before being merged with Mutter
- Most APIs revolve around CoglContext and CoglFramebuffer

**Clutter**



# Clutter

An OpenGL-based toolkit

- Previously, and application toolkit
- Forked, copy lives inside Mutter
- Slowly progressing to be a compositor toolkit
- Actors, stage, views, etc, allow creating nice effects
- Animation framework
- Input handling

# Clutter

- Traditional toolkit update cycle:
  - Layout: places actors somewhere
  - Paint: paint actor contents
  - Pick: determine what's beneath the cursor
- 2D actor tree on 3D space
- Uses paint volumes for basic culling
- Affine transformations everywhere



## Frame Scheduler

- Recently the frame scheduler was improved
- Goal was to give more room for applications to draw
- ... and higher chances for reducing latency

## Frame Scheduler



| → monitor vsync

■ → next frame

■ → skip frame

## Frame Scheduler (improved)



| → monitor vsync

■ → next frame

■ → skip frame

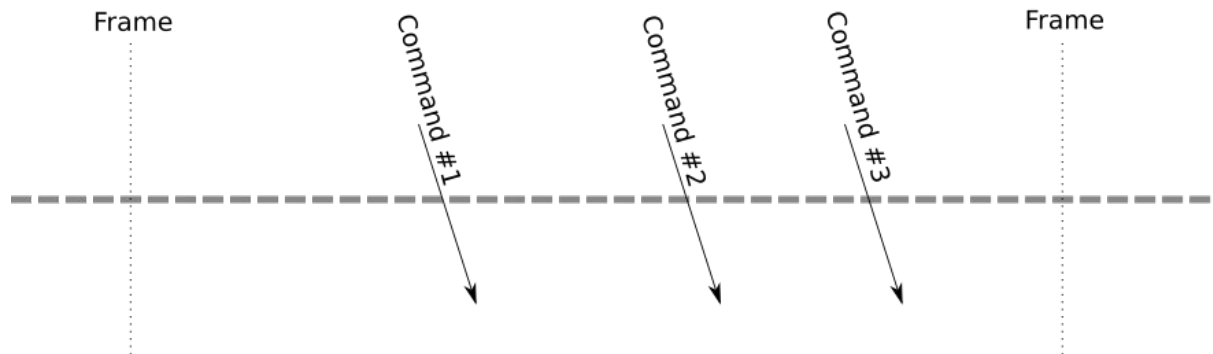
# Transactional KMS

*... or, at least, the path to*



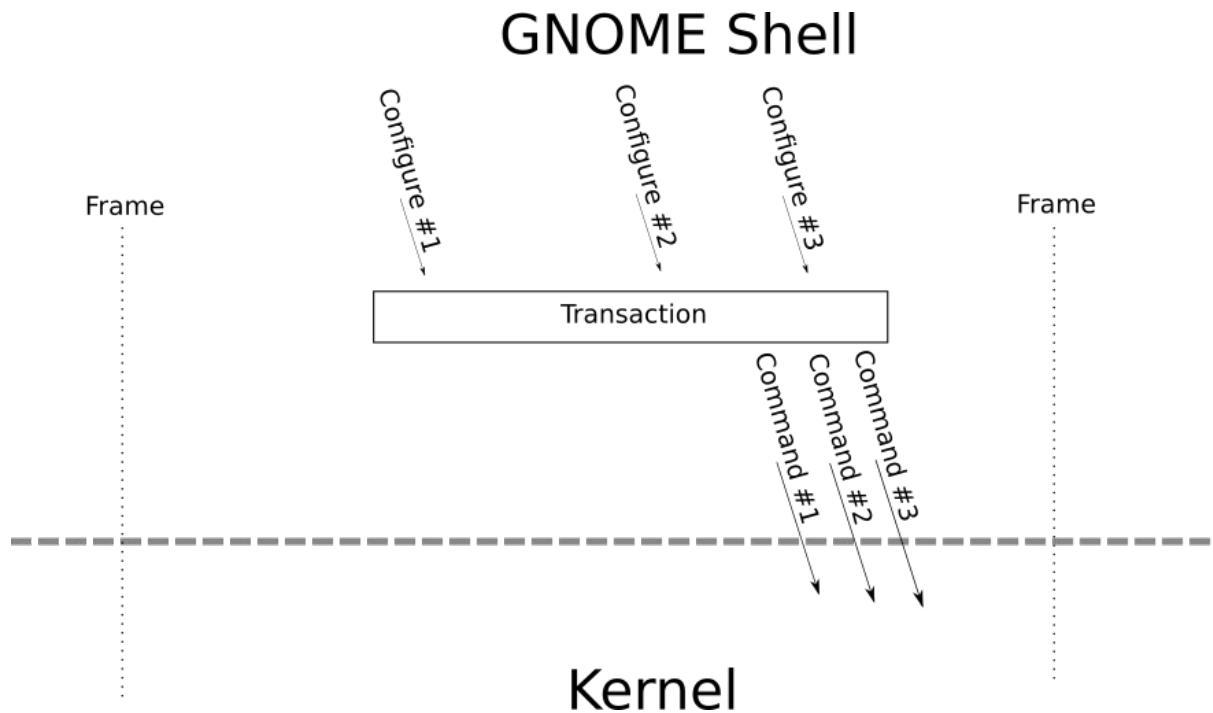
## Non-transactional (until GNOME 3.32)

GNOME Shell

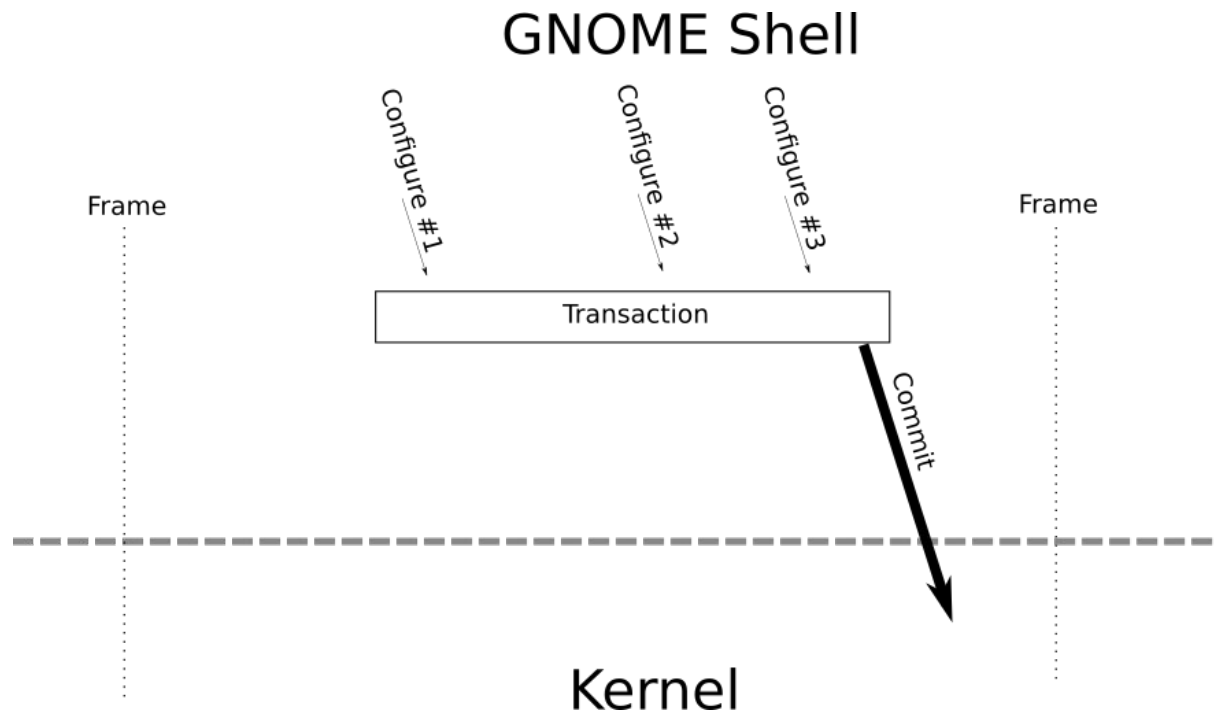


Kernel

## Fake Transactional (current)



## Atomic Transactional (TBD)



## Transactional KMS

- Updates KMS state and composited image atomically
- Needs to be able to switch between OpenGL and:
  - Hardware overlays for compositing
  - Accelerated cursor plane
  - CRTC gamma for color management
- DRM buffer modifiers
- Most of the code lives under `src/backends/native/meta-kms-*`
- Thread safety in mind
  - Eventually, a dedicated KMS thread
  - Multi-threaded KMS transaction setup

## While Talking About KMS...

- Basically no support for overlay planes yet
  - Mutter uses the cursor plane at most
- Will require deep surgery in Mutter's Clutter
- Use *libliftoff* perhaps?
  - Seems like the most reasonable option for now

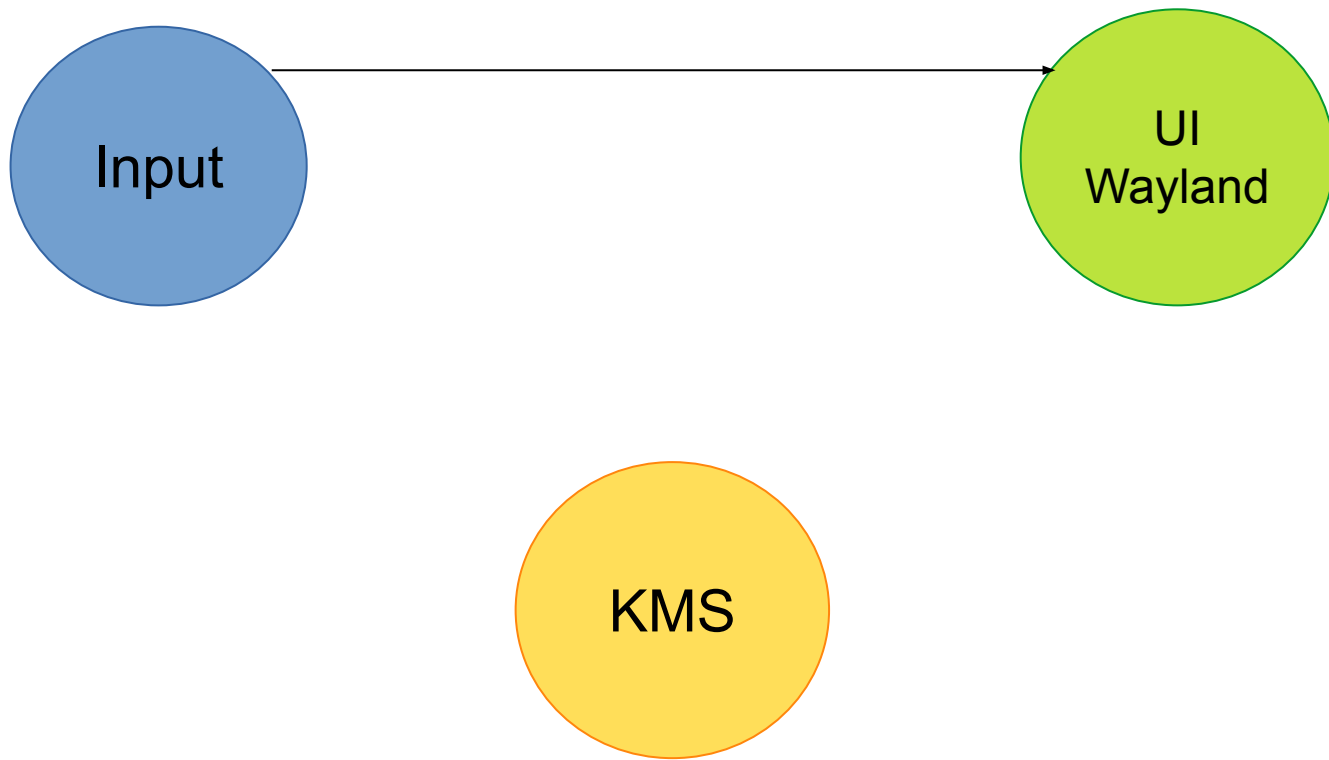
**Input**

# Input

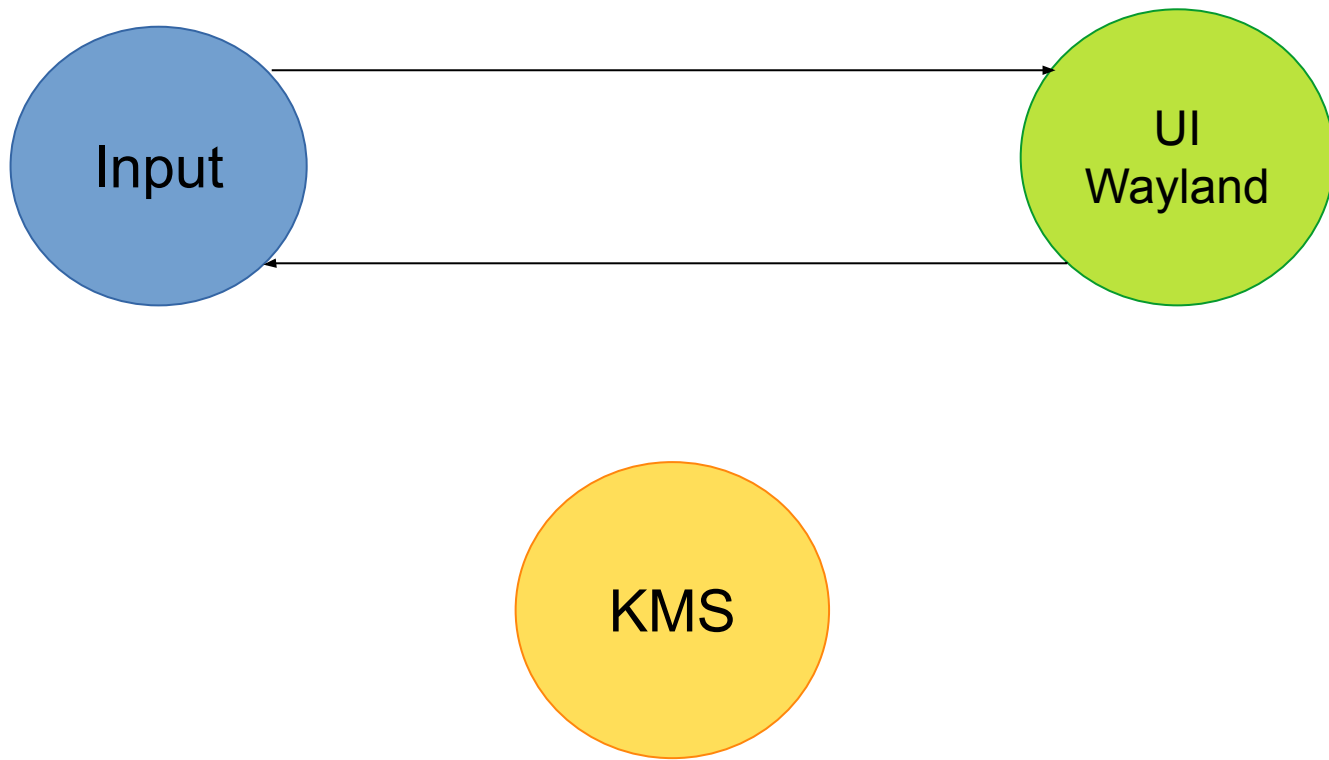
- Uses libinput and xcb
- Implemented as seats
  - Device owners and source of events
  - Wayland: MetaSeatNative
  - X11: MetaSeatX11
- Next steps: input thread
  - Work in progress:  
[https://gitlab.gnome.org/GNOME/mutter/-/merge\\_requests/1403](https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/1403)



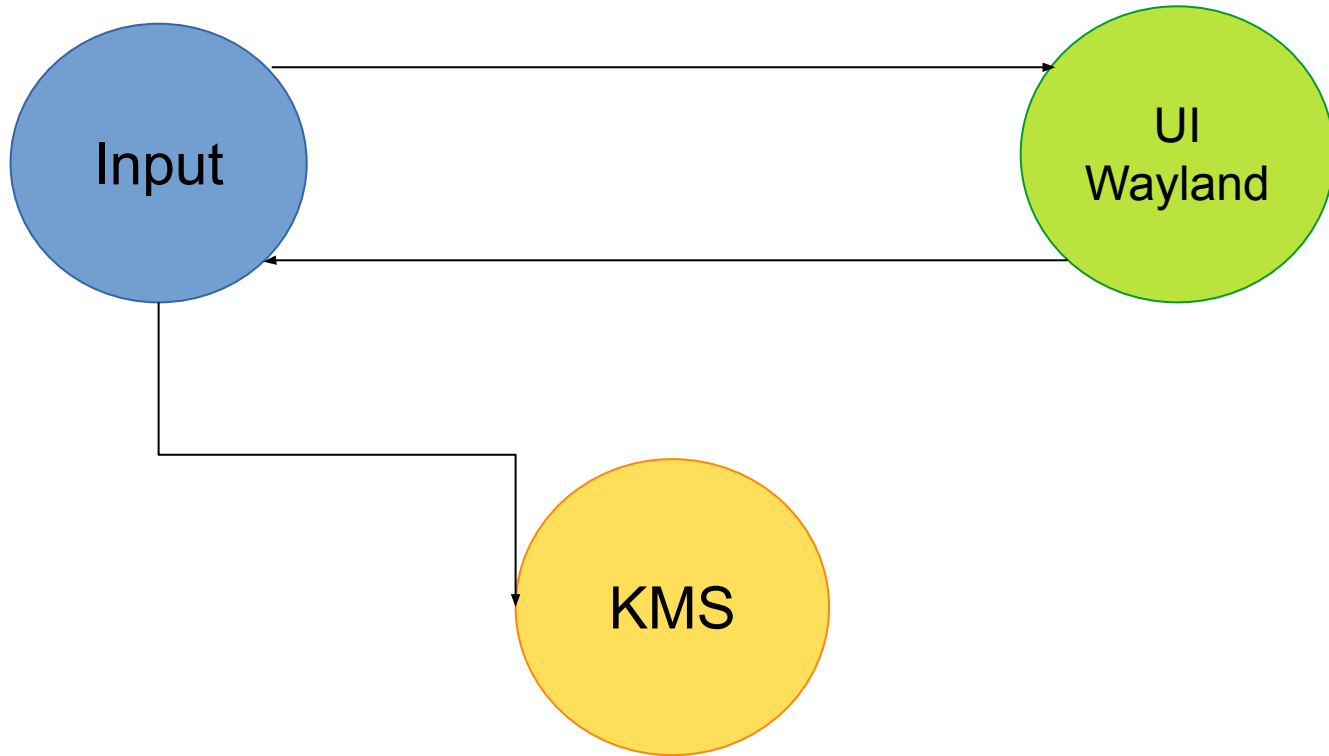
## Input Thread



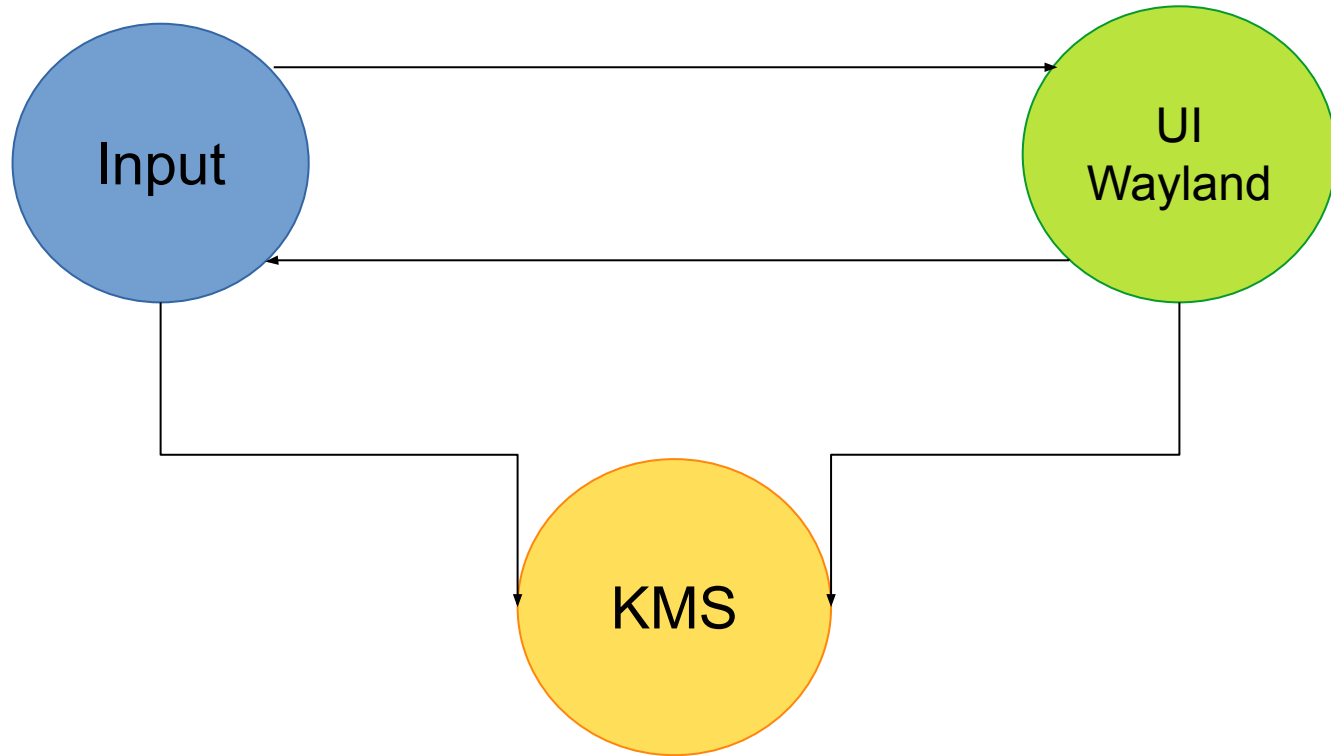
## Input Thread



## Input Thread



## Input Thread



## Potential Benefits

- No missed libinput events
- No blocked cursor pointer
- Better handling of high frequency devices
- Reuse of cursor plane (e.g. for tablets)
- In general, peace of mind

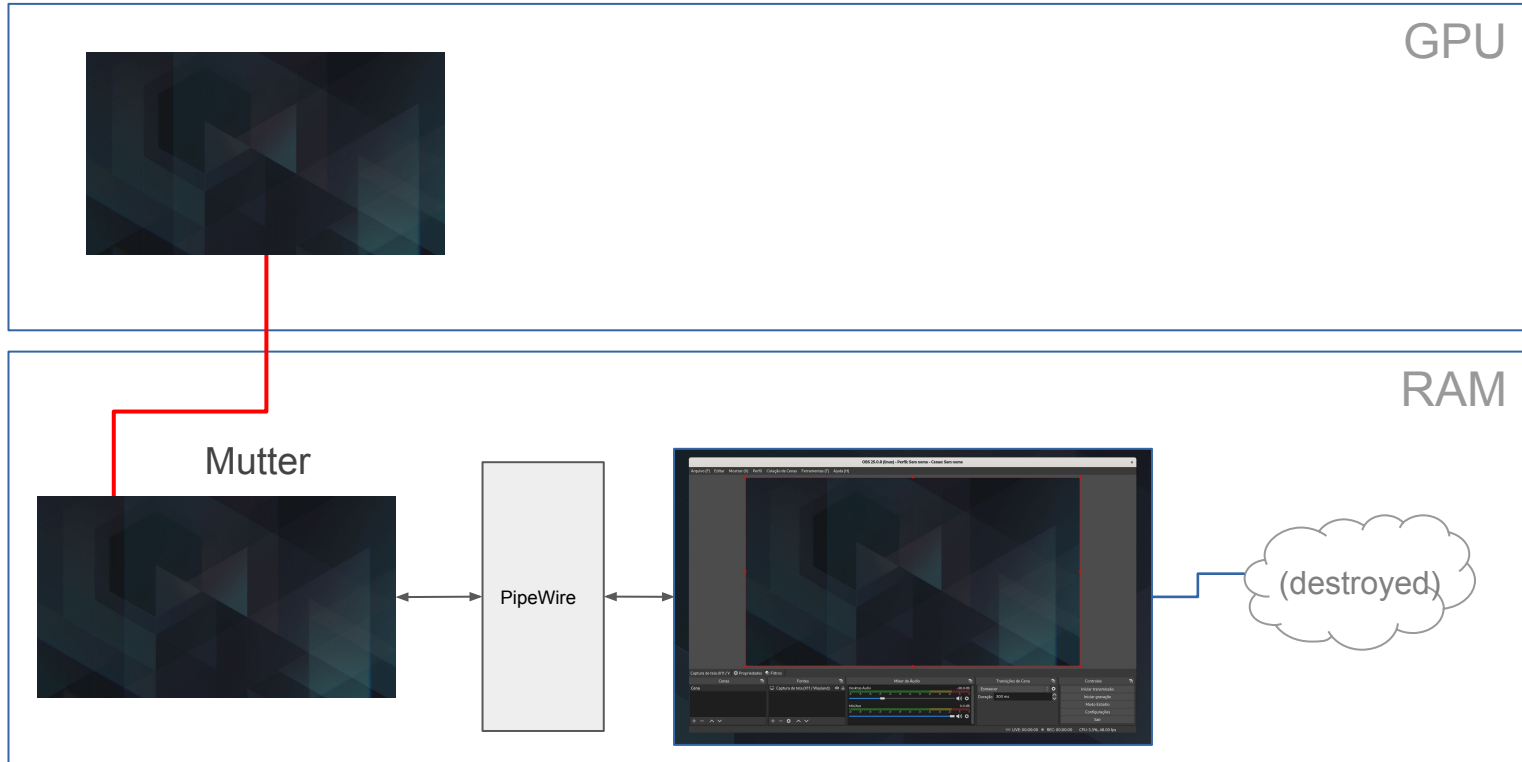


## Screen Sharing

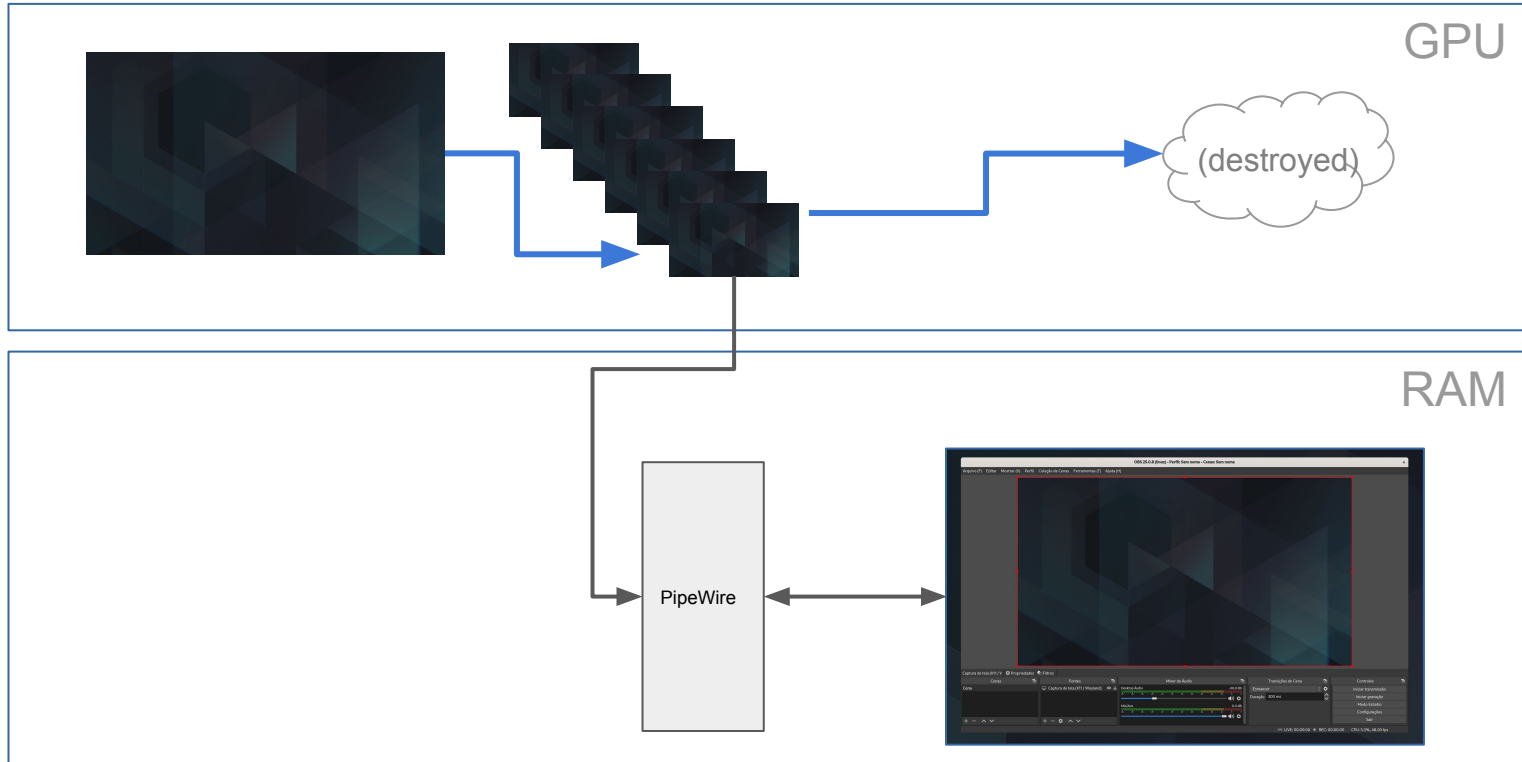
- Based on PipeWire
- Pre-allocates a number of buffers
- Cycles through them
- Supports streaming metadata
- Fast path: DMA-BUF sharing
- Slow path: *glReadPixels()*



## Screen Sharing (fallback)



## Screen Sharing (DMA-BUF)



## Screen Sharing

- Demo!

## Screen Sharing

- No NVidia support
- No X11 support

Some thoughts:

- Dealing with GBM is a breeze
- PipeWire doesn't send critical DMA-BUF info, such as modifiers
  - This shouldn't be hard to add to PipeWire though
  - Great task to start contributing

## Screen Sharing

- Links
  - <https://github.com/obsproject/obs-studio/pull/3338>
  - [https://gitlab.gnome.org/feaneron/obs-xdg-portal/-/merge\\_requests/5](https://gitlab.gnome.org/feaneron/obs-xdg-portal/-/merge_requests/5)
- If you have a modern Linux distribution:

```
$ flatpak install --user https://flathub.org/beta-repo/appstream/com.obsproject.Studio.flatpakref
```

## Other Plans

## Other Plans

- Finish transition to Graphene
- Retained rendering tree
- Deeper, more detailed profiling

# Graphene

- A graphics data types library
- Platform-specific optimizations: SSE 2 & 4.1, ARM NEON, etc
- Jackpot: graphene matrices!
- Mutter is halfway through the transition
  - Finish porting to Graphene
  - Only basic data types ported
  - Last battle is CoglMatrix → graphene\_matrix\_t
  - Help needed



# Graphene

# Q&A



Thanks!