



IMPORTANCE OF MODIFIER IN LINUX

GNOME-ASIA 2022

Sameer Lattannavar

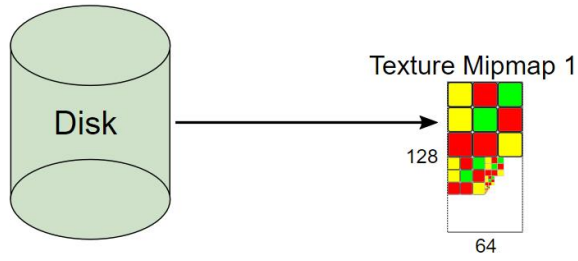


OUTLINE OF THE TALK

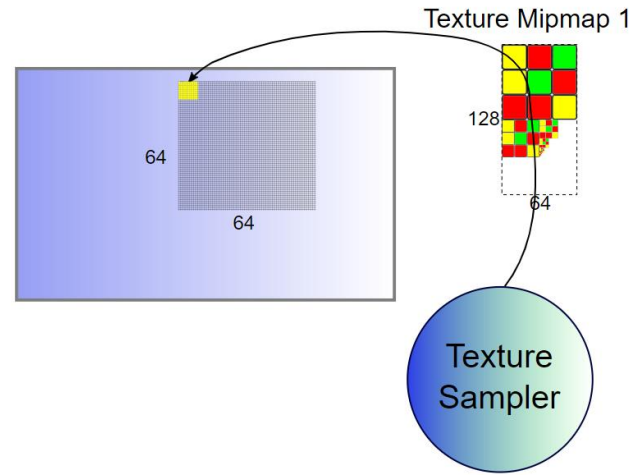
- Understanding modifiers
- Modifiers Flow
- Modifiers implementation in Linux
- Performance Boost
- Current status of modifiers support in mutter
- Q&A

Typical Composition Flow

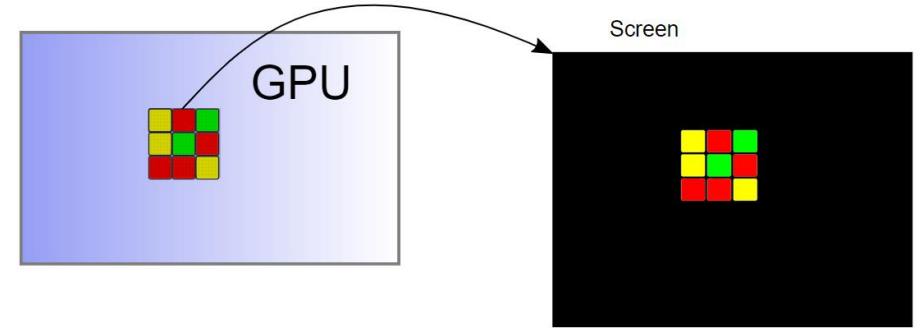
Texture Upload



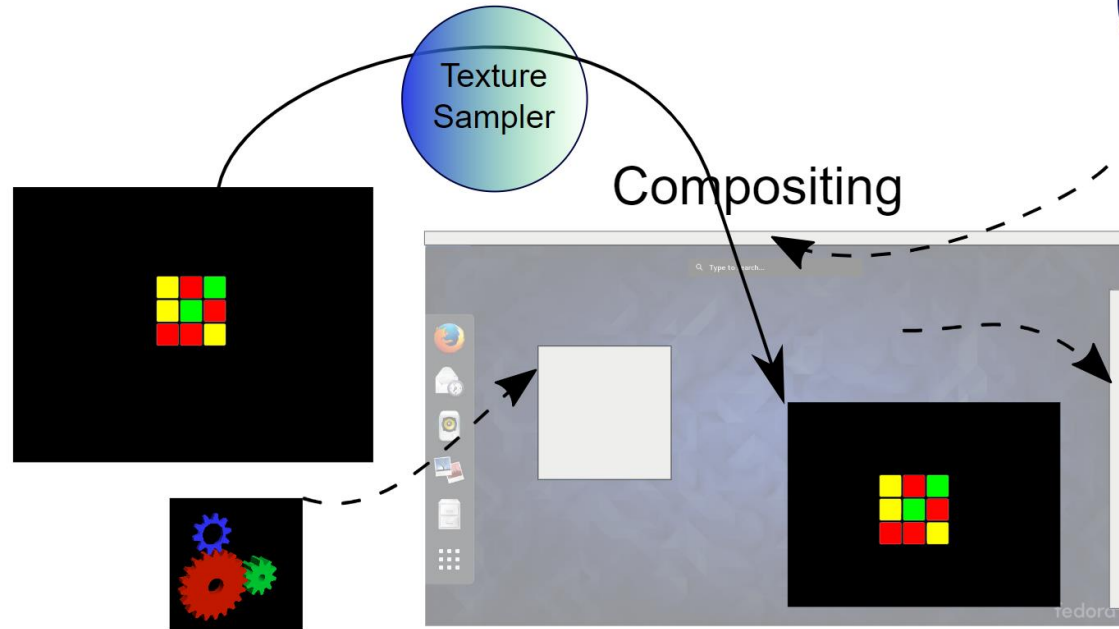
Texel Fetch



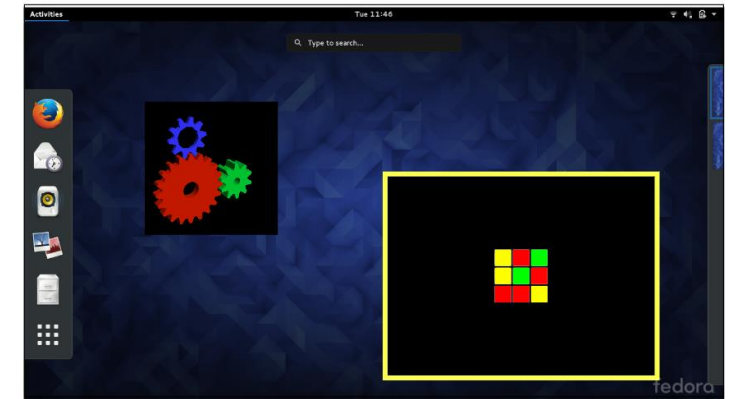
Framebuffer Write



Compositing



Display Scanout



WHAT ARE MODIFIERS ? AND WHY?



Modifiers solves the End-to-End Render Buffer compression



They are modifications that are applied to a buffer's layout



Compliment to the Tiled Framebuffer



Way to implement Lossless Compression for Rendering and Media buffers in Graphics Stack

END TO END, LOSSLESS COMPRESSION

End to end lossless compression attempts to provide both "end to end" and "lossless" compression transparently to software.

End to End

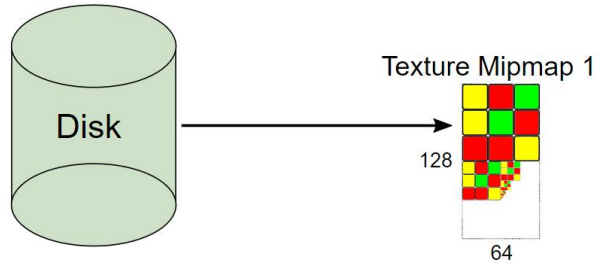
- At all stages of the rendering flow to composition stage, modifiers must be made visible.

Lossless

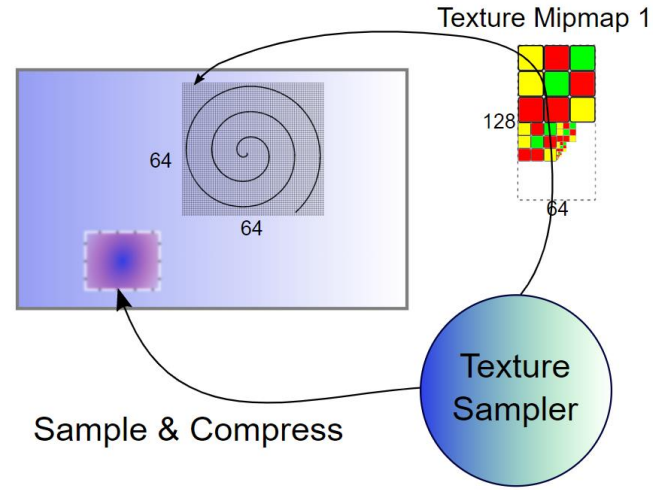
- Since every stage of the stack is involved with the compressed buffer, no one would like to lose the quality of the frame
- Lossless algorithms are mostly inefficient in terms of savings

Composition Flow with Modifiers support

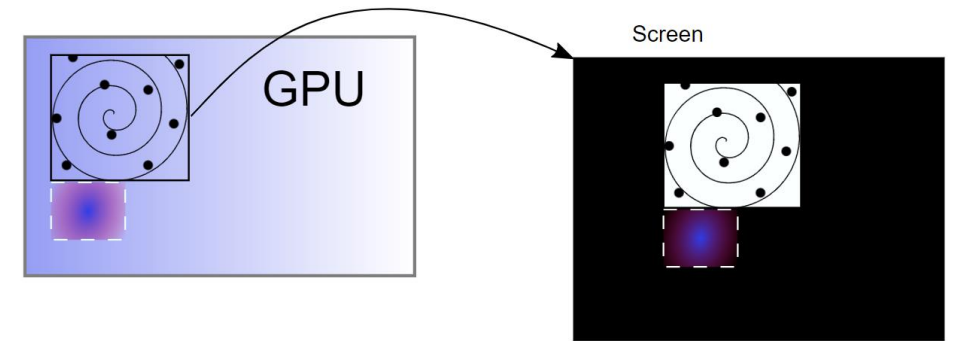
Texture Upload



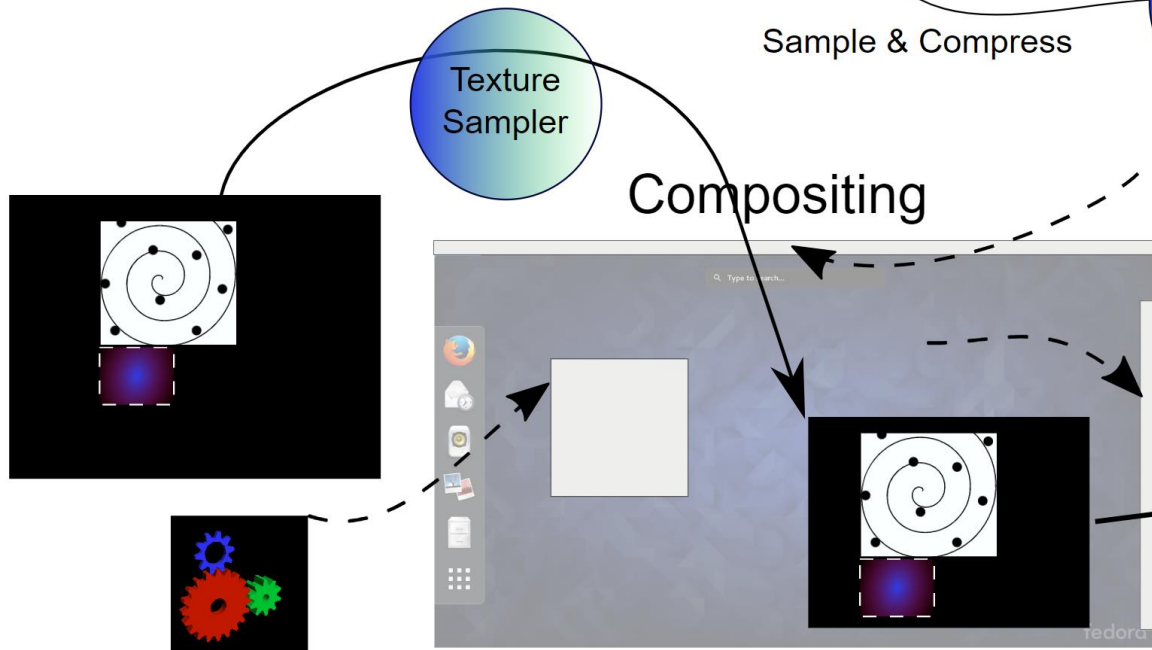
Texel Fetch



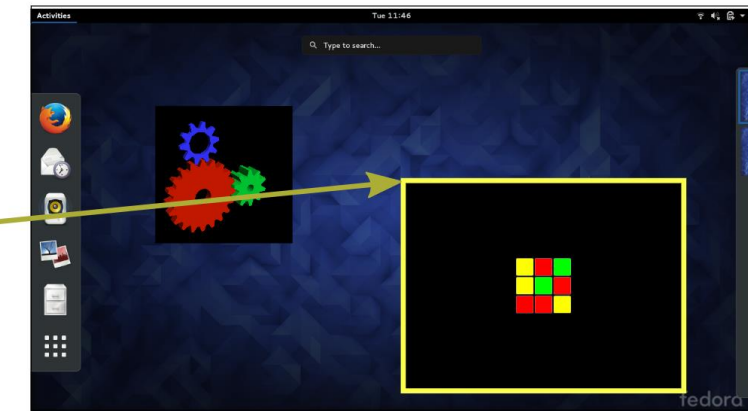
Framebuffer Write



Compositing

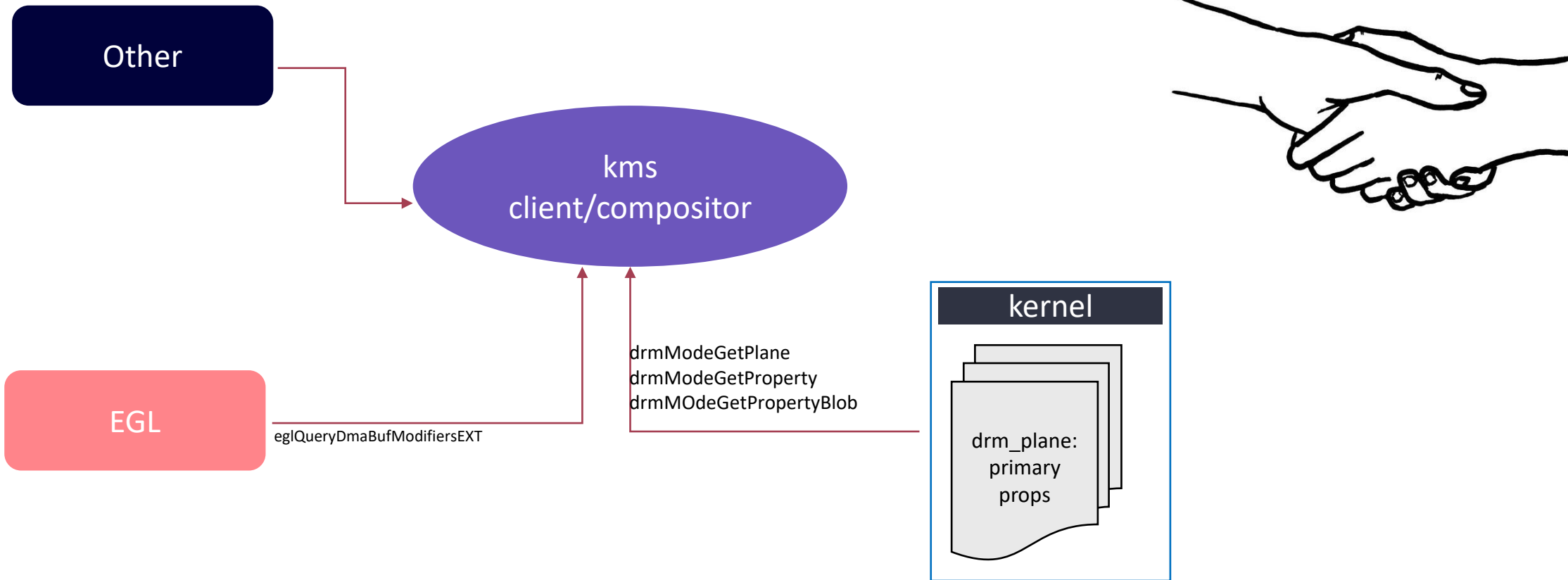


Display Scanout



STEP#1: COMPOSITOR NEGOTIATES

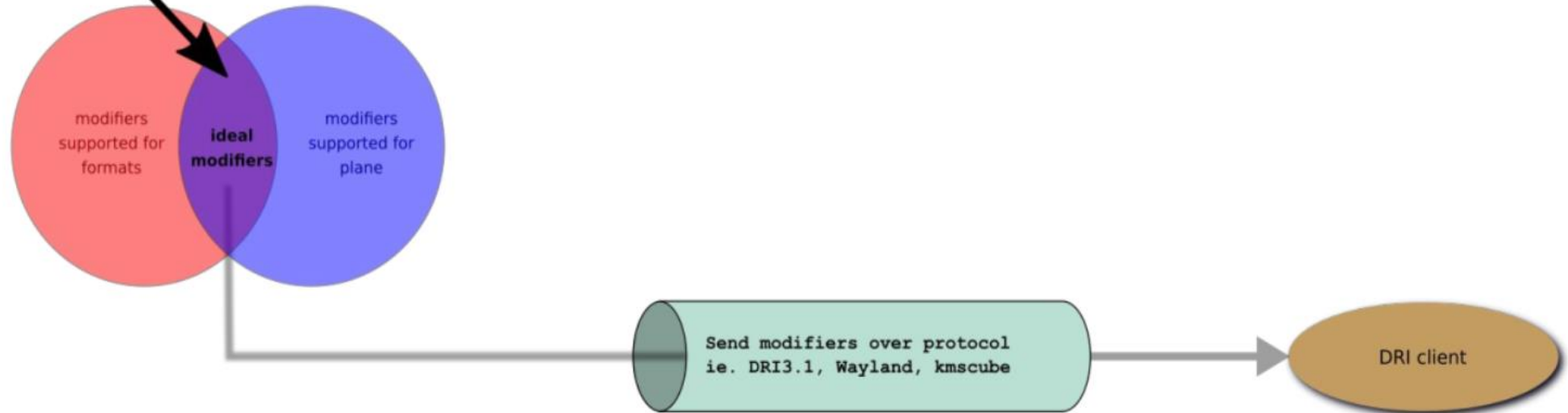
Query all “sink” APIs to find out what modifiers are supported for the currently requested format.



STEP#2: DIRECT CLIENT TO USE THE RIGHT MODIFIER

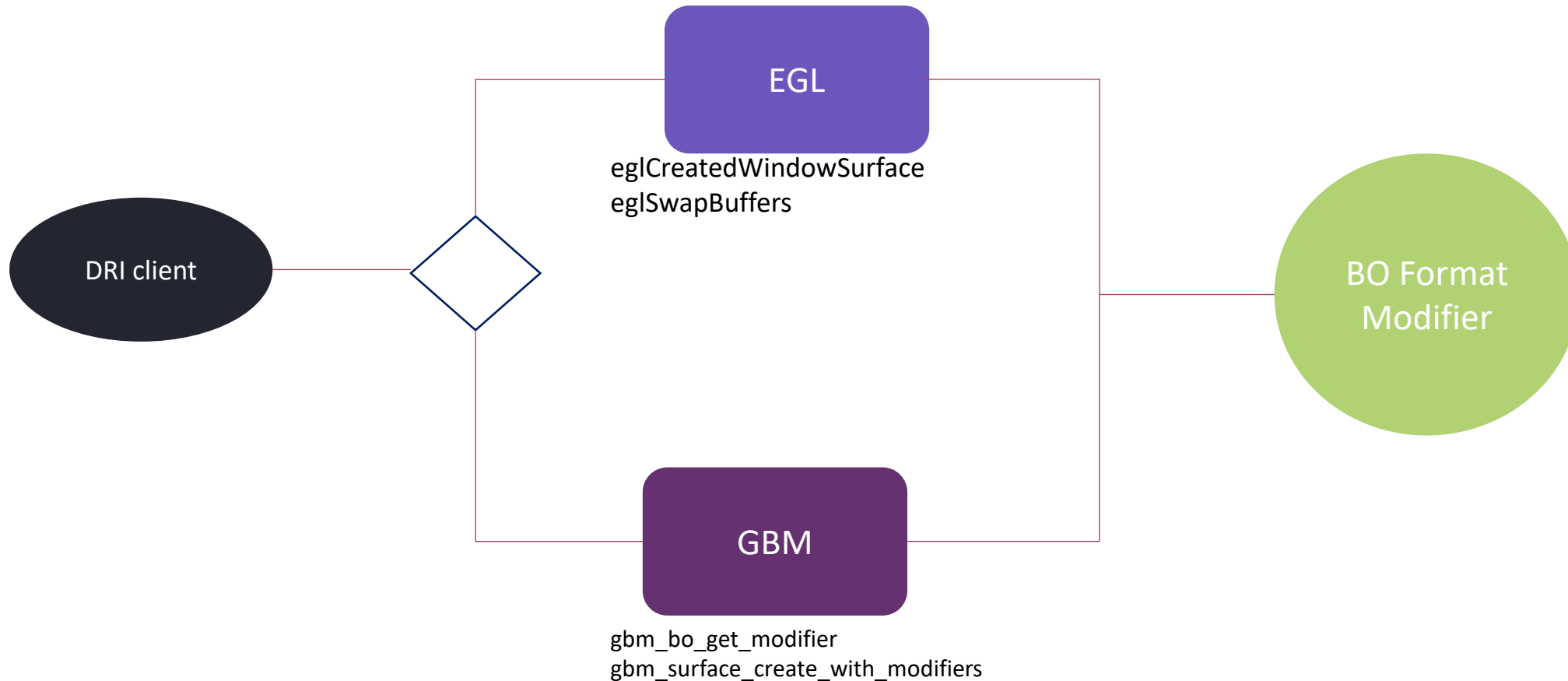
With desired modifiers selected, client will be informed which modifiers to use.

Modifiers which can be used for direct scanout, and used to sample from EGLImages

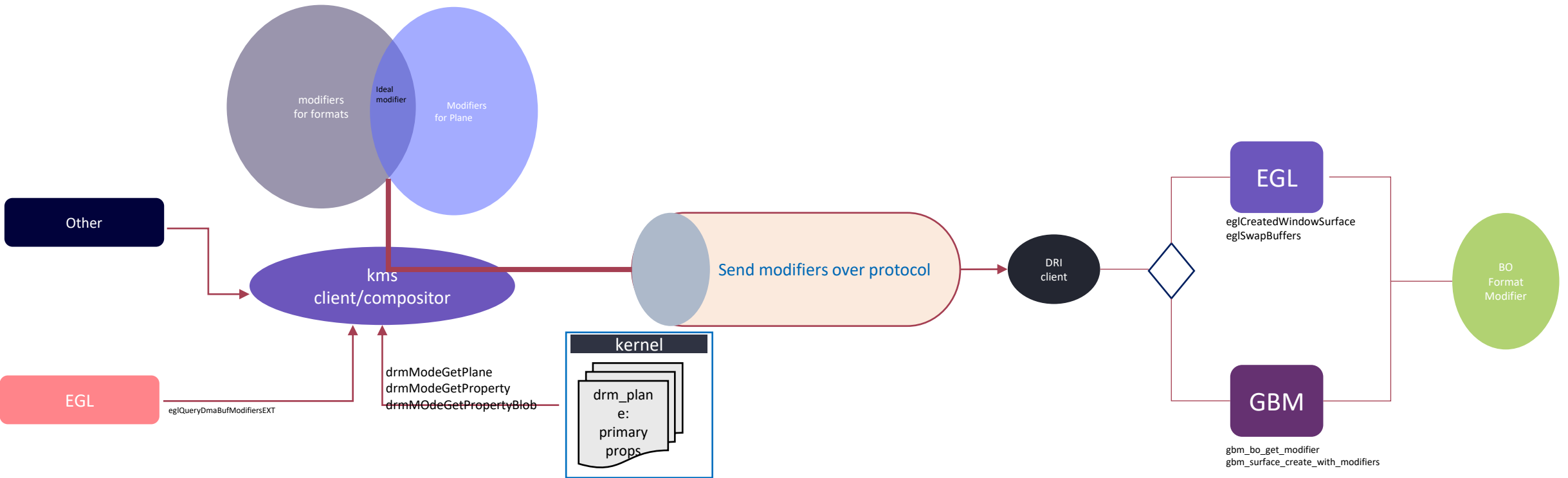


STEP#3: CREATING THE BUFFER OBJECTS

The client created the buffer either directly or indirectly using the formats and modifiers it wants.



STEP#4: PUTTING TOGETHER

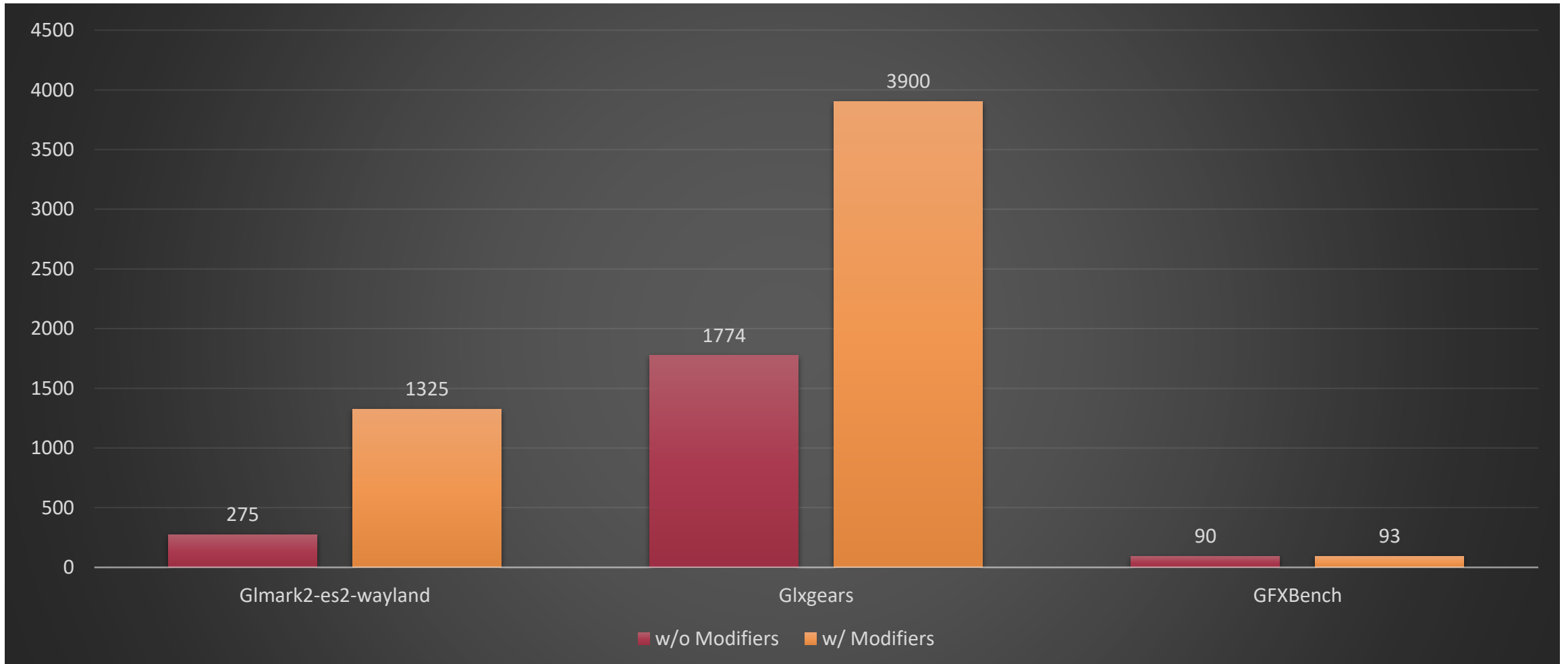


MODIFIERS IMPLEMENTATION IN LINUX

Modifiers are implemented across these pieces of Graphics Software stack:

- Linux DRM
 - blobifier
 - AddFB2
- Linux i915
 - blobifier
 - AddFB2
- Mesa
 - DRI
 - EGL etc.
- Protocol
 - Wayland-protocols
 - Mutter
 - Weston
 - Sway etc
- Khoronos
 - image_dma_buf_import_modifiers

SOME PERFORMANCE DATA





ISSUE FACED BY MUTTER COMMUNITY

- While enabling modifiers on i915 systems Mutter community complained 'Black external monitor after login': <https://gitlab.gnome.org/GNOME/mutter/-/issues/1618>
- This forced to disable modifiers in udev-rules on i915 systems, so the clients will start using Linear DMA Buffer instead of Tiled buffers
- **Configurations used with issue reported:**
 - Platform: KabyLake (Gen-9 iGPU)
 - Monitors connected : eDP & DP
- **Use case:**
 - With the above configurations, user logged in to **find External monitor (DP) was black**
 - Community root caused and disabling modifiers resolved the issue
- **Decision taken on this error:**
 - Multi Head/Multi Display use cases started failing with modifiers enabled, so decided to disable modifiers via udev-rules for i915 systems
 - There are NO failures on Single Display use cases

RE-ENABLE THE MODIFIER SUPPORT IN MUTTER FOR I915 SYSTEMS

- Short Term solution:
 - udev-rule to enable modifier for latest i915 systems:
 - By conditionally disabling modifiers in udev-rules for Gen9 or lower Intel GPUs.
 - MR-2641 is merged to enable modifiers on latest i915 systems will be part of GNOME-44 release
- Long Term Solution:
 - TEST_ONLY Commit check for all modifiers until working modifiers
 - This is suggested by both Mutter community and #dri-devel IRCs
 - Solid plan of implementing TEST_ONLY commit is yet to be decided
 - Estimated timeline: 3-4 weeks for development efforts upon final decision, Review and merge subject to community's responses
 - Some progress is made towards modifiers with these MRs to 'Send modifier to client despite udev-rules' to enable client to use modifiers:
 - Send modifiers to client despite udev rule
 - **[Merged]** https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/2546
 - Add support for multi-monitor direct scanout
 - **[Merged]** https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/2526

REFERENCES:

- Block diagram courtesy : Ben Widawsky's blog:
<https://bwidawsk.net/blog/2021/2/modifiers/>

Modifiers implementation MRs:

- Mutter: https://gitlab.gnome.org/GNOME/mutter/-/merge_requests/1959
- Wayland-protocols: https://gitlab.freedesktop.org/wayland/wayland-protocols/-/merge_requests/8
- Mesa: https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/11248

Future steps:

- Long term Solution with TEST_ONLY Commit:
<https://gitlab.gnome.org/GNOME/mutter/-/issues/1623>

THANK YOU



Sameer Lattannavar



+91 9986017846



Sameer.lattannavar@intel.com

Nick: @slattann