Gallium3D

Graphics Done Right

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

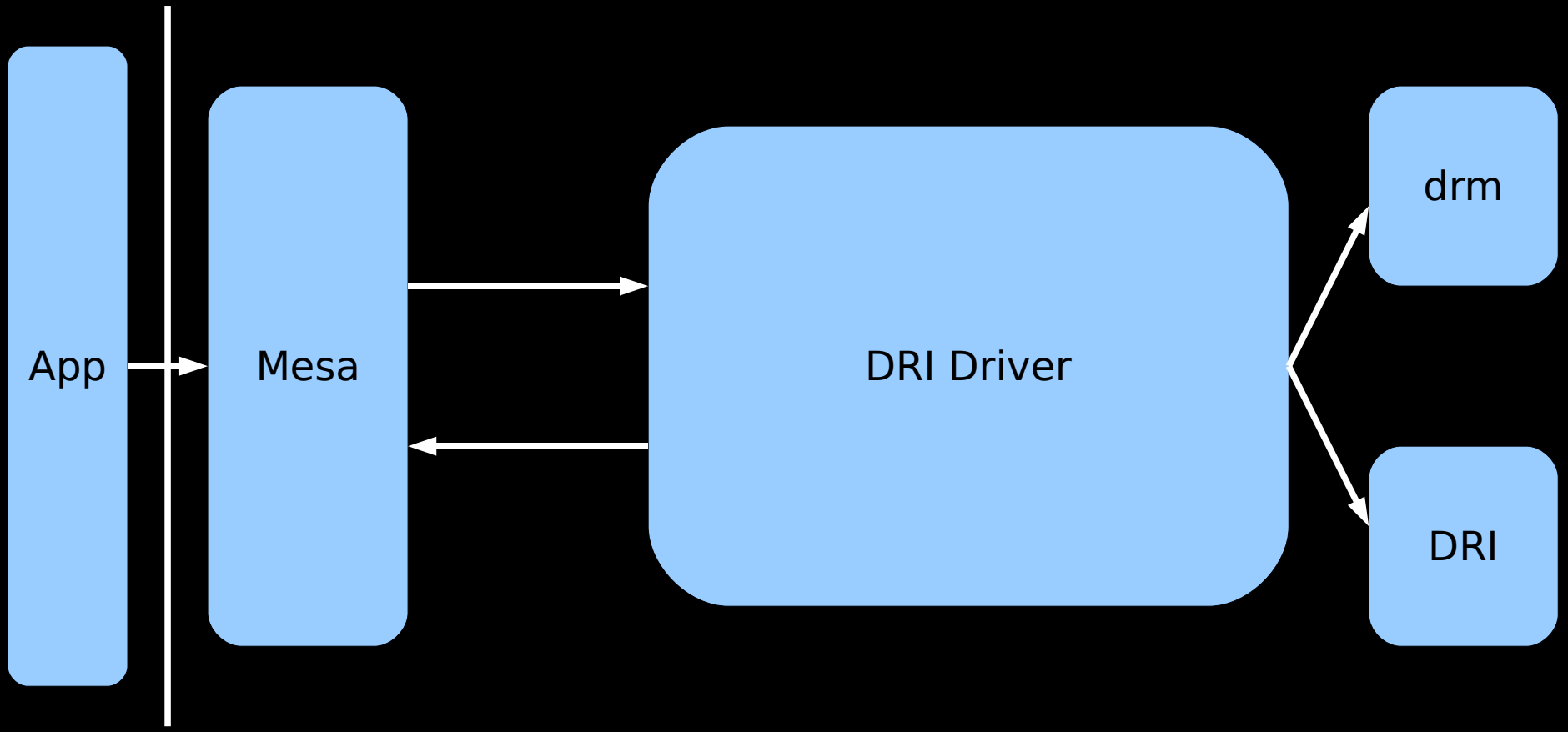
Contents

- Recap
  - Gallium3D
    - General summary
    - Why would you want to use it.
- Gallium3D latest changes
- Taking request (no singing)

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

DRI Driver Model



- Drivers were tied to OS, API, window system.
- EG, dealing with DRI cliprects in DrawArrays.
- Driver interface becoming unmanageable.

Zack Rusin zack@tungstengraphics.com

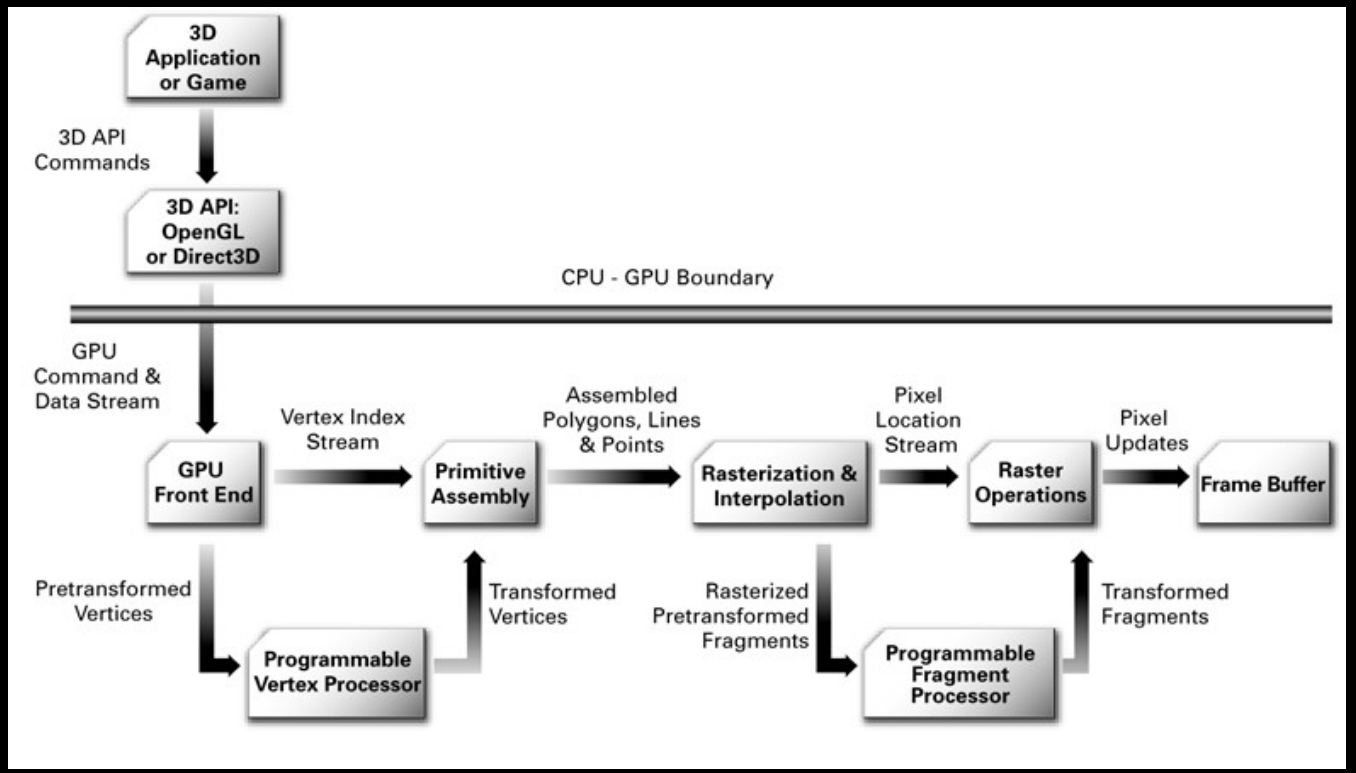
Tungsten Graphics

DRI Driver Model

Zack Rusin zack@tungstengraphics.com

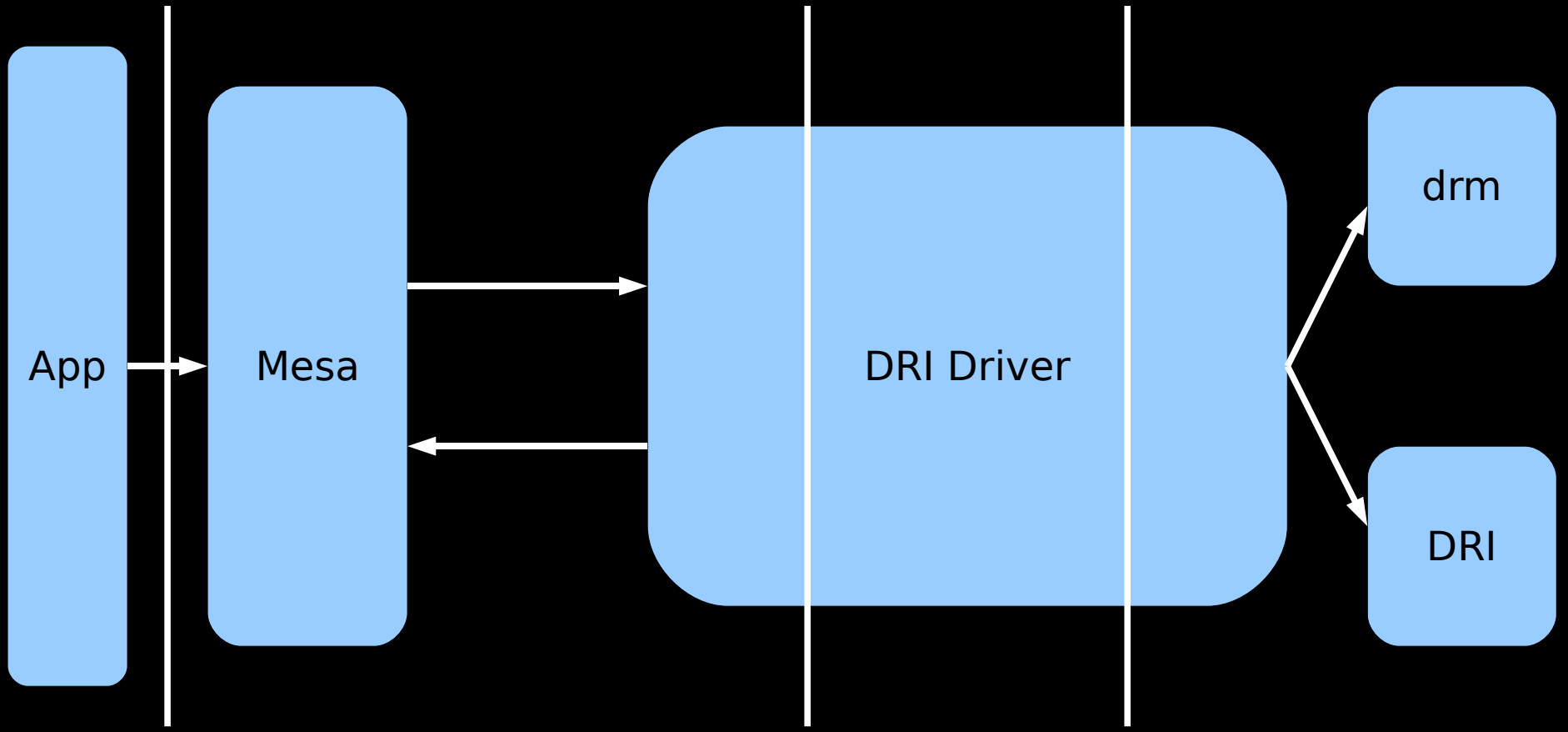
Graphics Pipeline

- Essentially the same for all modern API's



Zack Rusin zack@tungstengraphics.com

Impose new interfaces

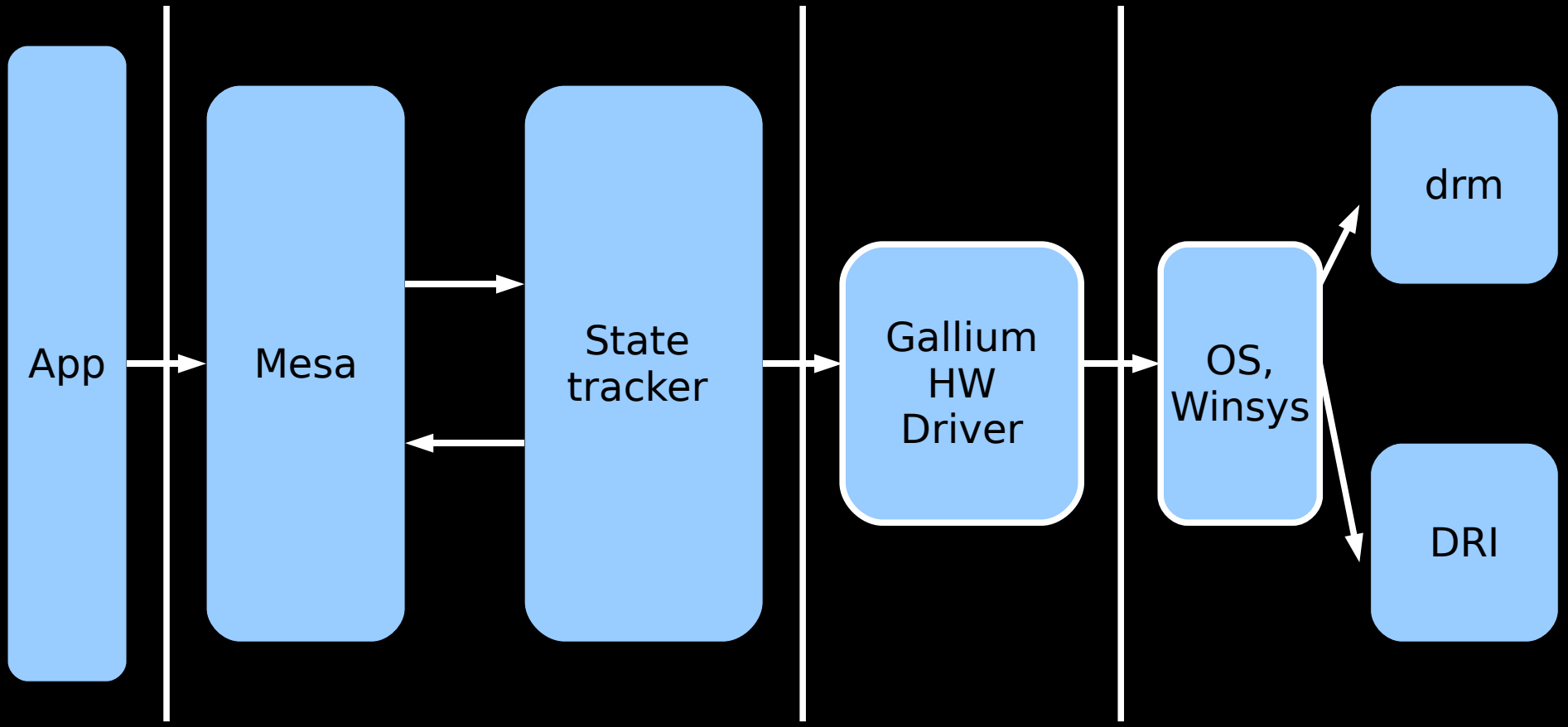


- Isolate interactions with API, OS, HW.
- Identify new interfaces.
- Split the driver.

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

Gallium in 2007



- The original plan for Gallium3D.
- Still more or less correct.

Zack Rusin zack@tungstengraphics.com

Since then...

- Rapid interface evolution
- Hopefully starting to stabilize, but there are still some minor issues outstanding.
- On the horizon: simplify TGSI shader representation
- Changes in the draw module
- New insights into fallbacks, driver structure.
- New utility code

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

Since then...

- Got some hardware drivers working
  - I915 (updated to head)
  - softpipe
  - Cell driver
  - i965
- External driver projects:
  - Nouveau
  - R300 work

Tungsten Graphics

Building blocks

- Gallium3D at its core is just an interface
- The actual functionality is split across different modules
  - Those modules can be mix-and-matched to produce a complete solution

Tungsten Graphics

Building blocks

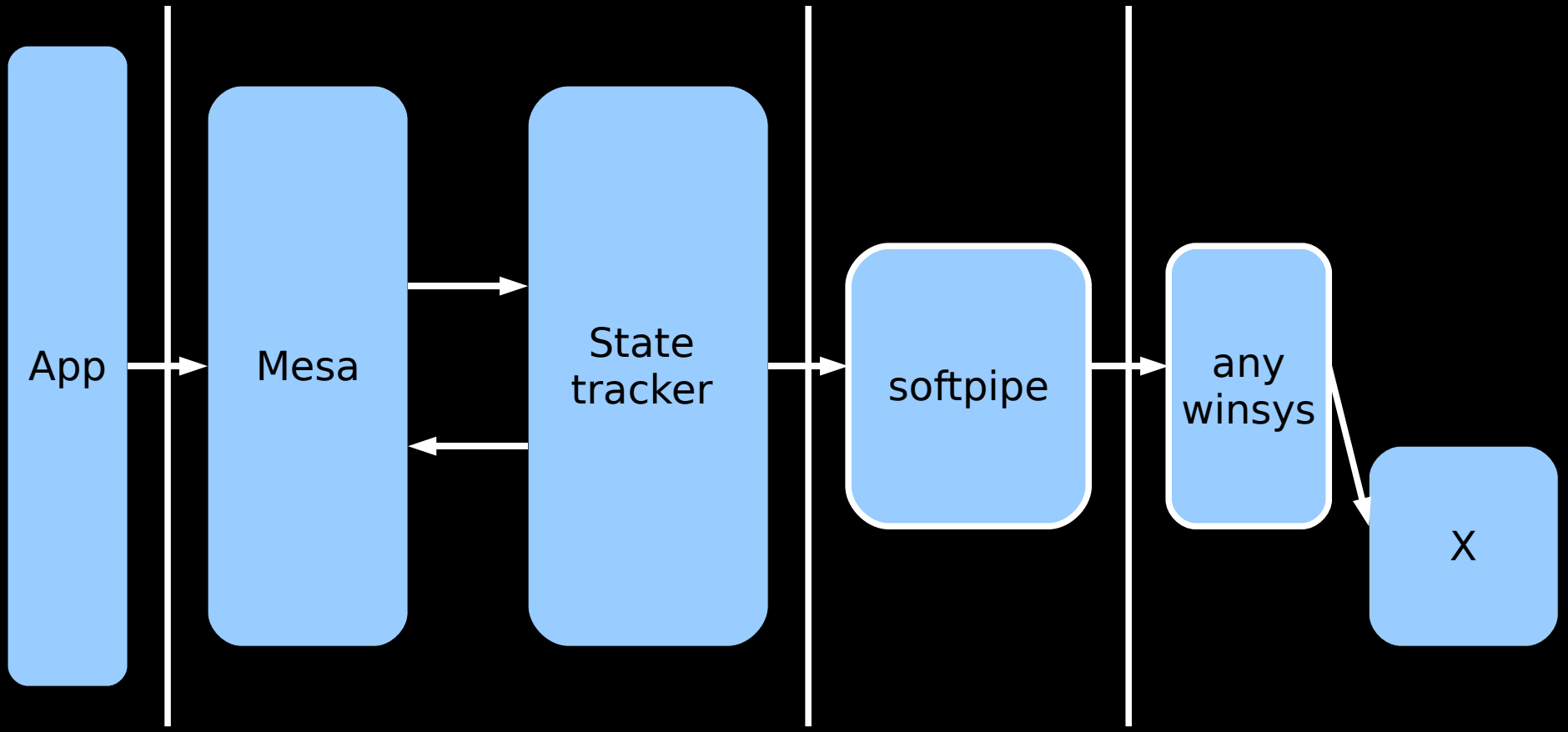
- Important modules within the framework include:
  - State trackers
    - Implement API on top of Gallium3D
  - Winsys
    - Integration with a windowing system, low level management (surfaces, buffers and fencing)
  - Gallium3D driver
    - Implements the Gallium3D interface

Tungsten Graphics

Building blocks

- Important modules within the framework include:
  - Draw
    - Software vertex paths
  - CSO
    - constant state objects management
  - Buffers management code
  - TGSI code
  - LLVM integration
  - A few others (sct, util)

Software Rasterizer

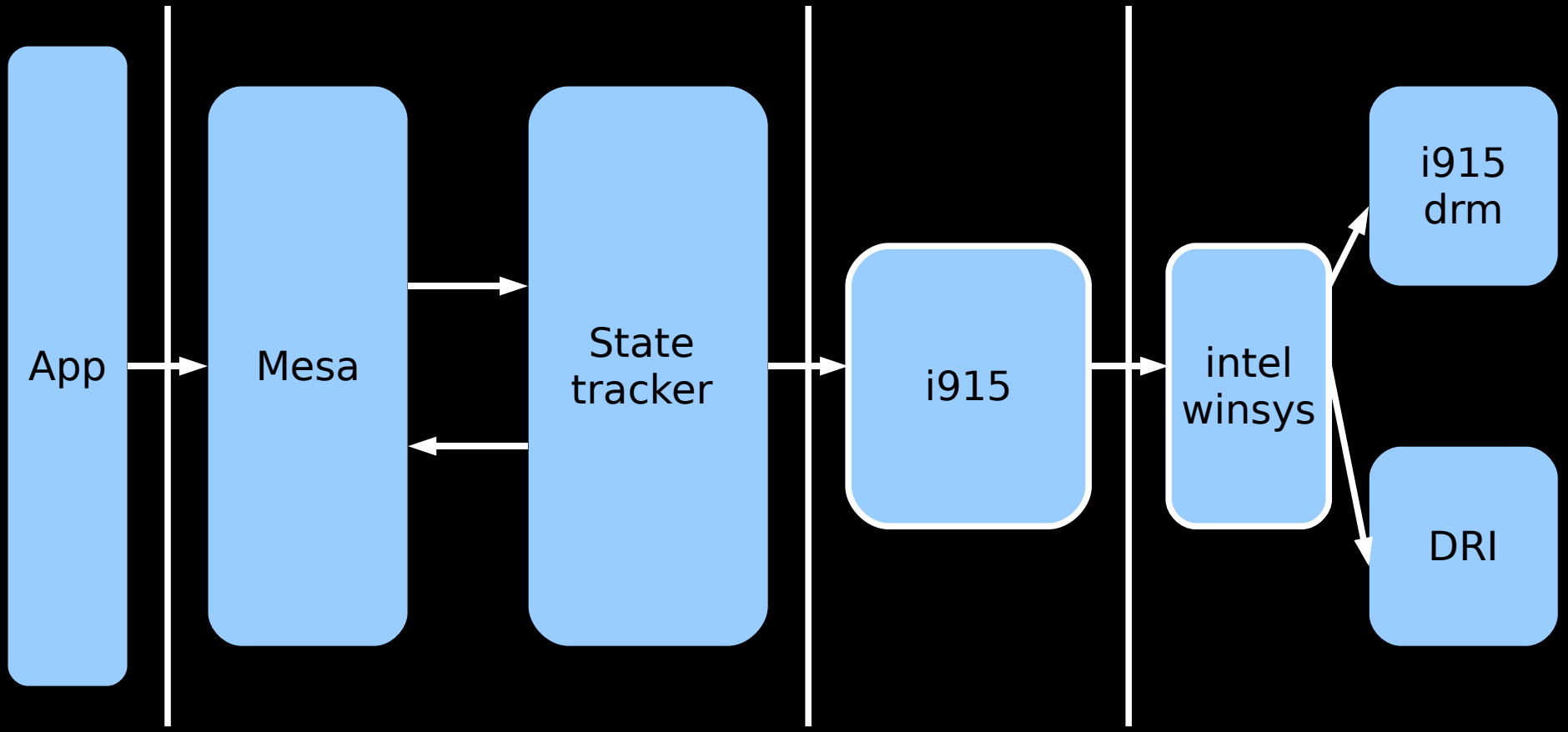


- Codegen through LLVM and simple rtasm.
- A fairly clear path to performance.
- A good project for someone?

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

Hardware: i915

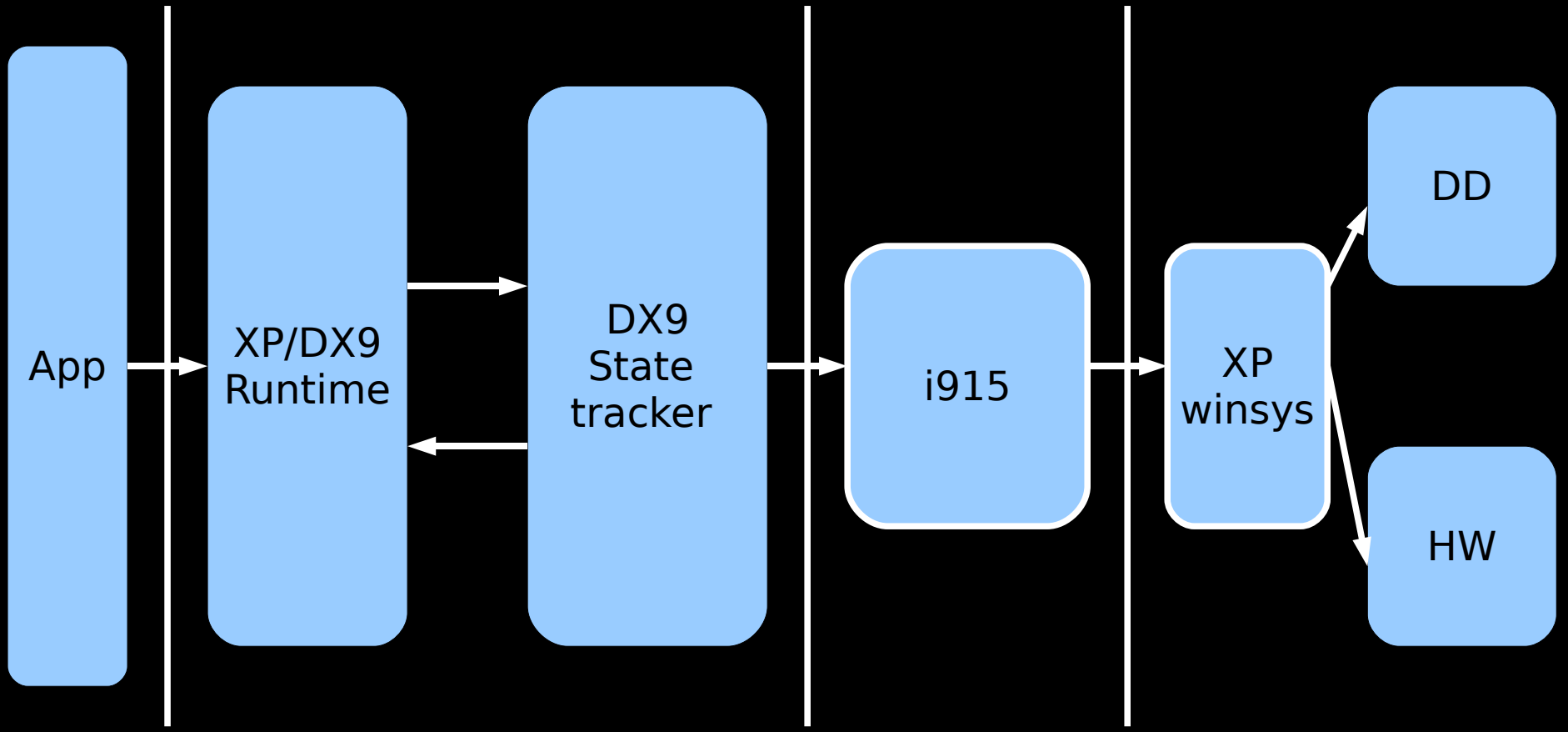


- Updated to the latest DRM changes.
- Near term goal: Rebase to X, DRM head.
- Later: DRI2, Polish, Performance...

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

It works on Windows

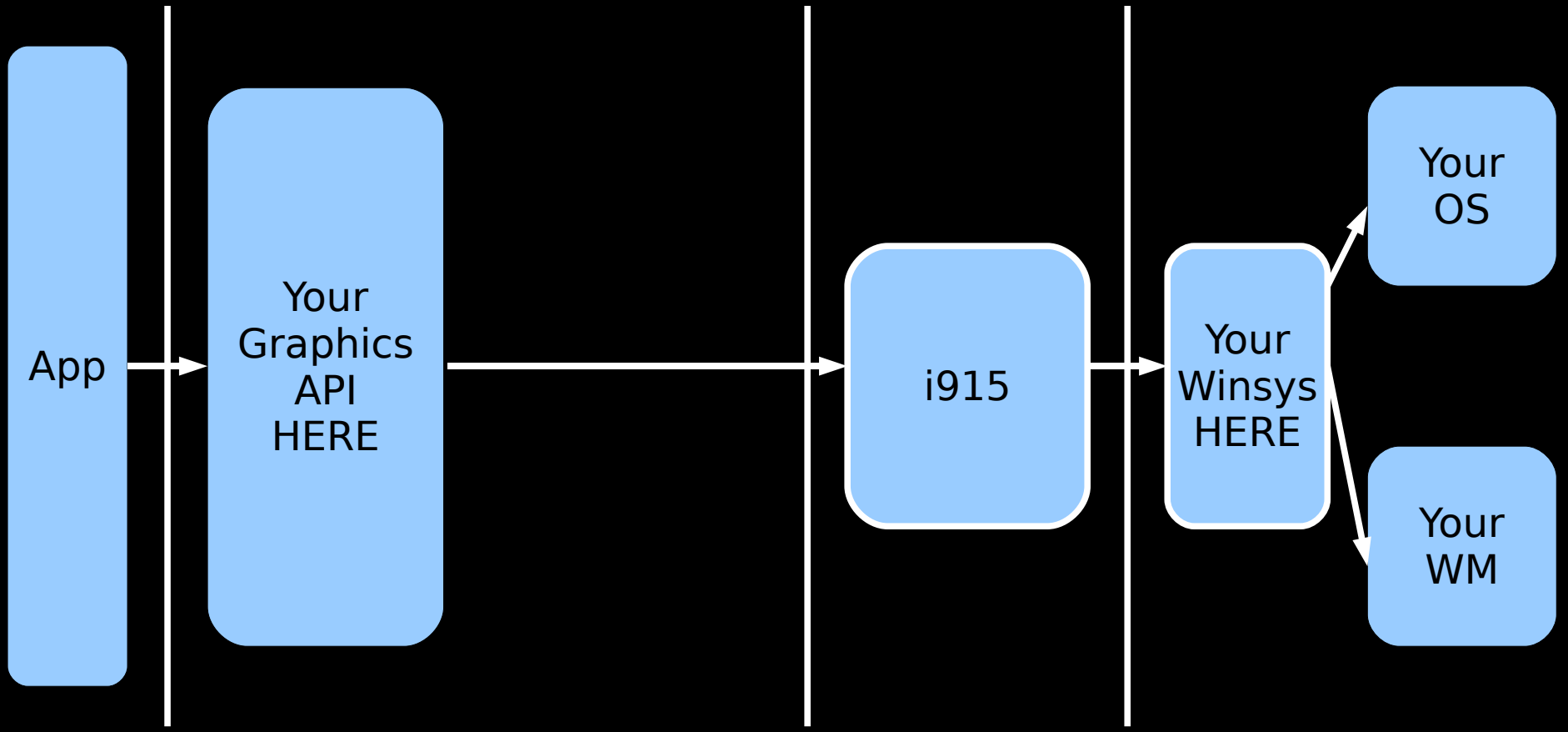


- This is actually working.
- Validates the portability claims for Gallium.

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

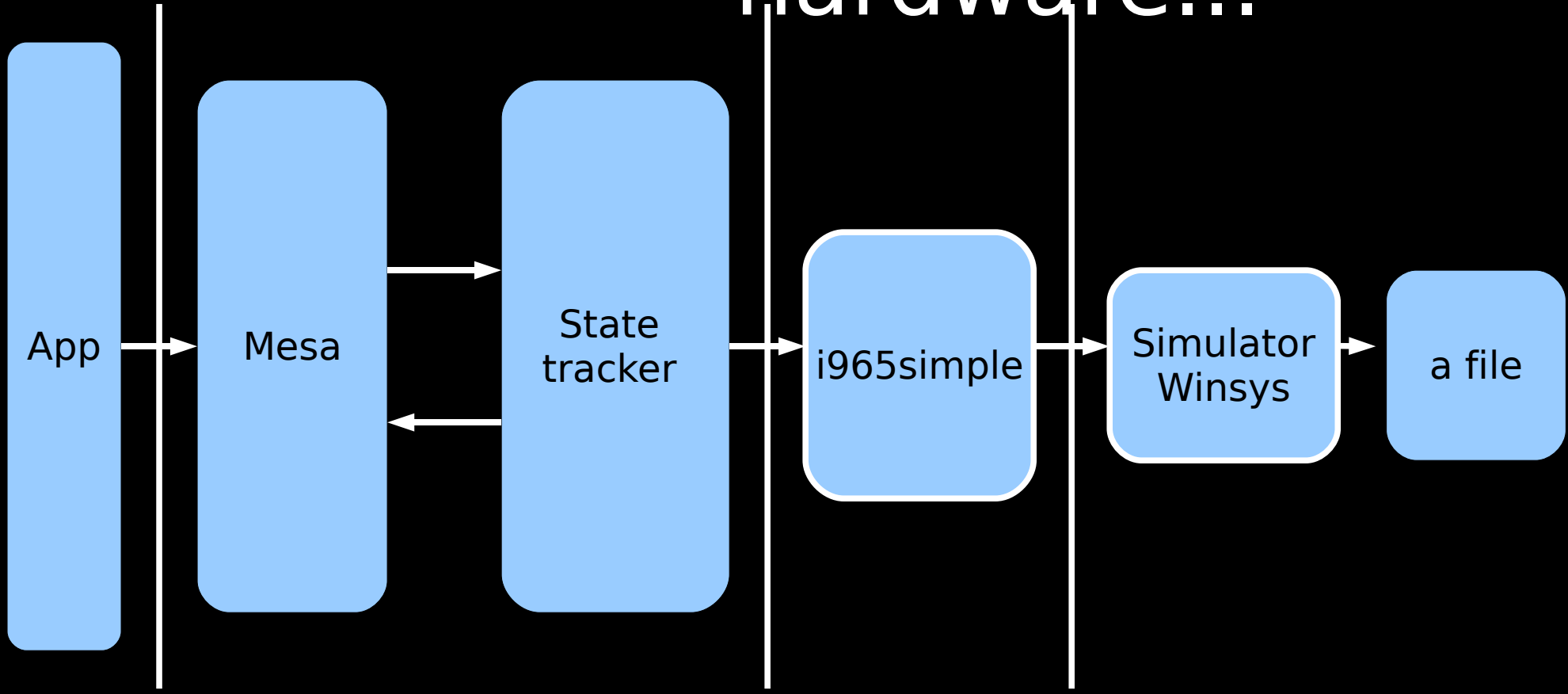
...It'll work anywhere



- DirectFB, VxWorks, Kdrive, GLES, Cellphones, Robots, FreeBSD, MiniGLX, EGL, Clusters, etc.
- Wider audience --> better drivers.

Zack Rusin zack@tungstengraphics.com

You don't even need hardware...



- A nice way to work on hardware you don't actually have available.
- Easy to capture, analyze dumps offline.
- TODO: Replay

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

Shaders

- At the very core of Gallium3D
- TGSI used throughout
  - Drivers can either:
    - Use TGSI directly
    - Employ LLVM code-generation facilities

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

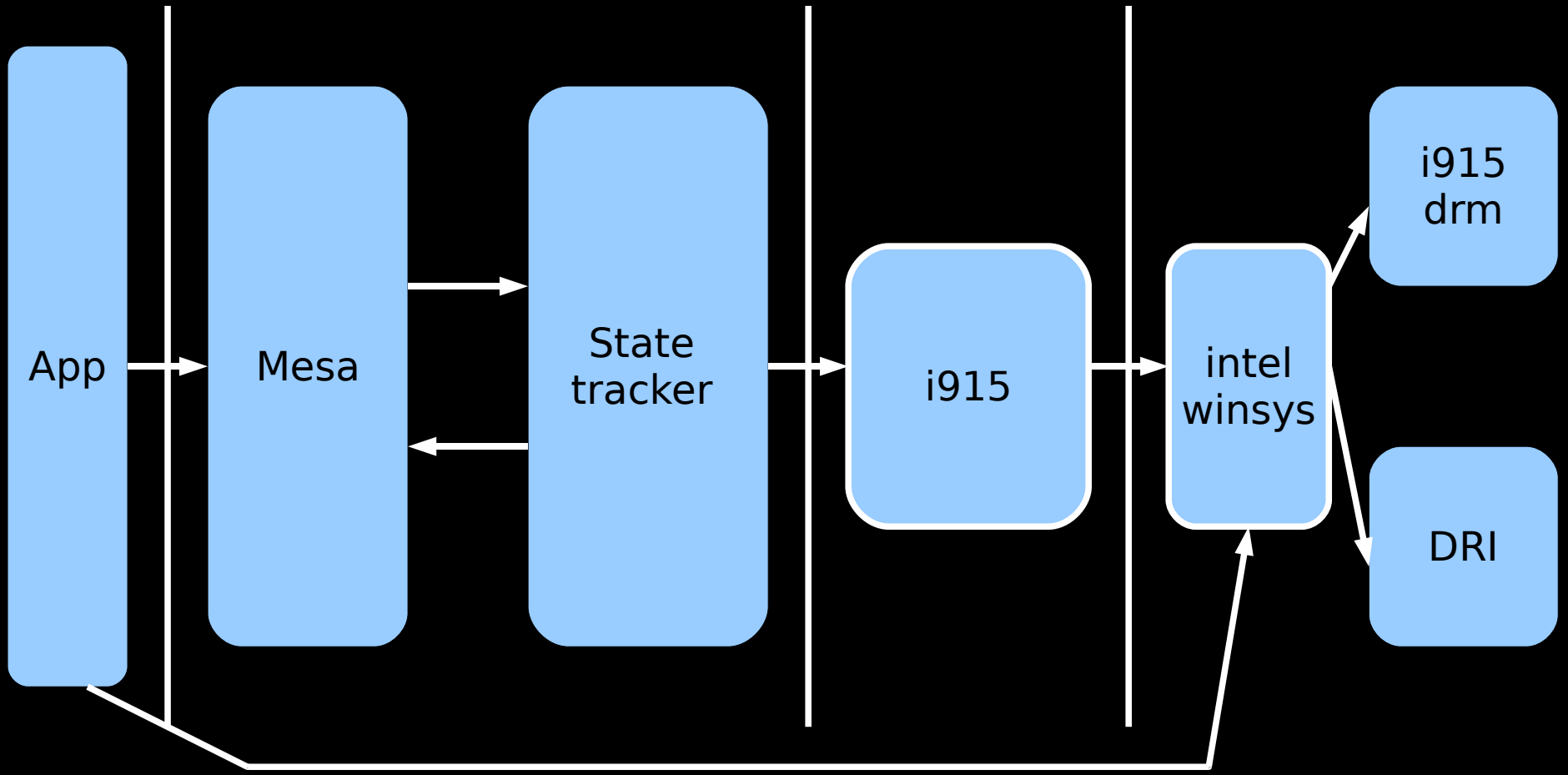
LLVM

- TGSI compiled into LLVM IR
- LLVM optimization passes used
- Drivers implement LLVM code-generator

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

Winsys issues



- GLX implemented by DRI + the Winsys layer
- Swapbuffers, create surface, etc, seem to bypass this nice stack.

Zack Rusin zack@tungstengraphics.com

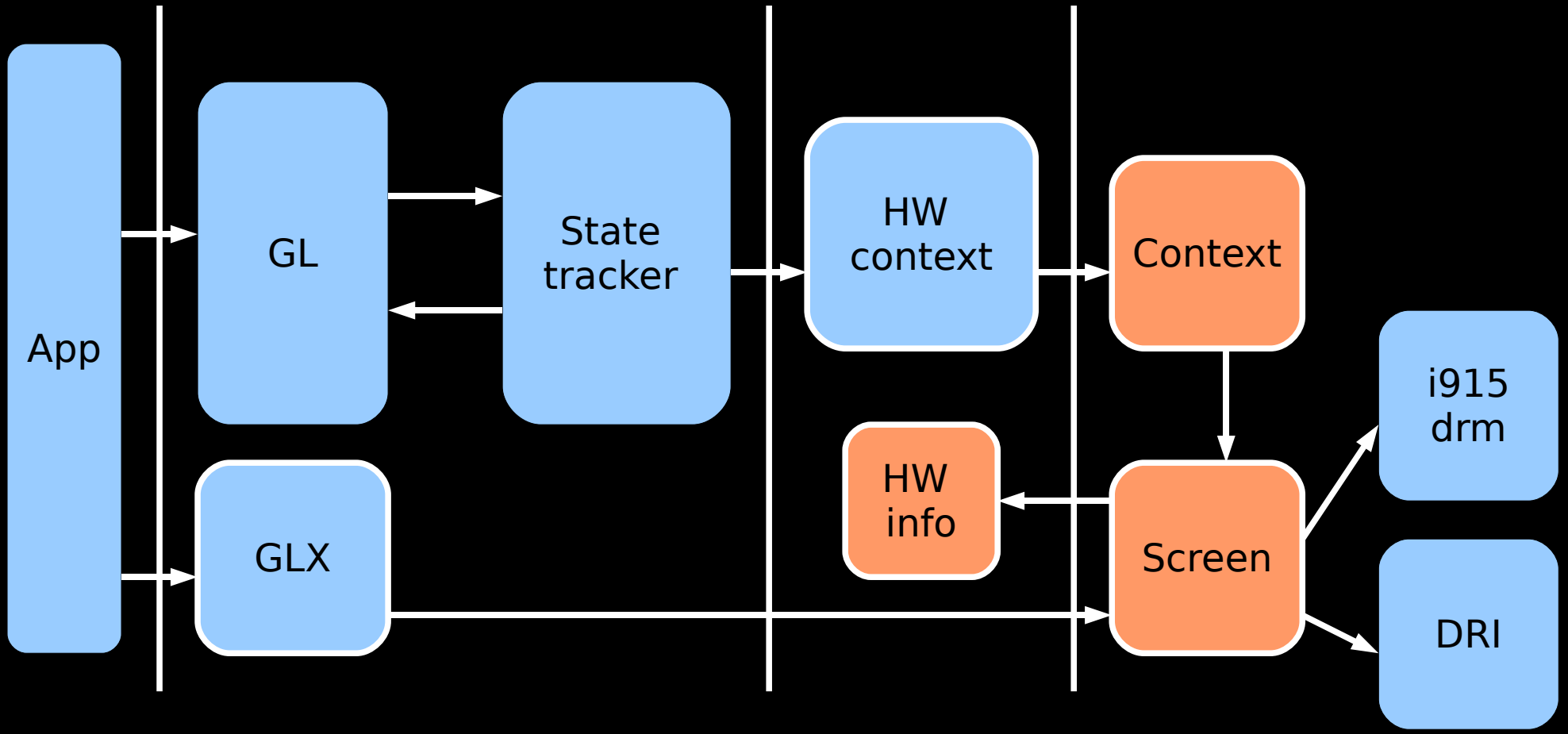
Tungsten Graphics

Winsys issues

- Neat diagram above ignores non-drawing aspects of the driver.
- There is real complexity here:
  - Surface allocation – happens before context creation
  - GL extensions – need to know (approximately) before context creation.
  - Swapbuffers
- Currently winsys is splitting into two entities: per-screen and per-context.
- May end up with a parallel stack, ie:

Tungsten Graphics

What's in a winsys?

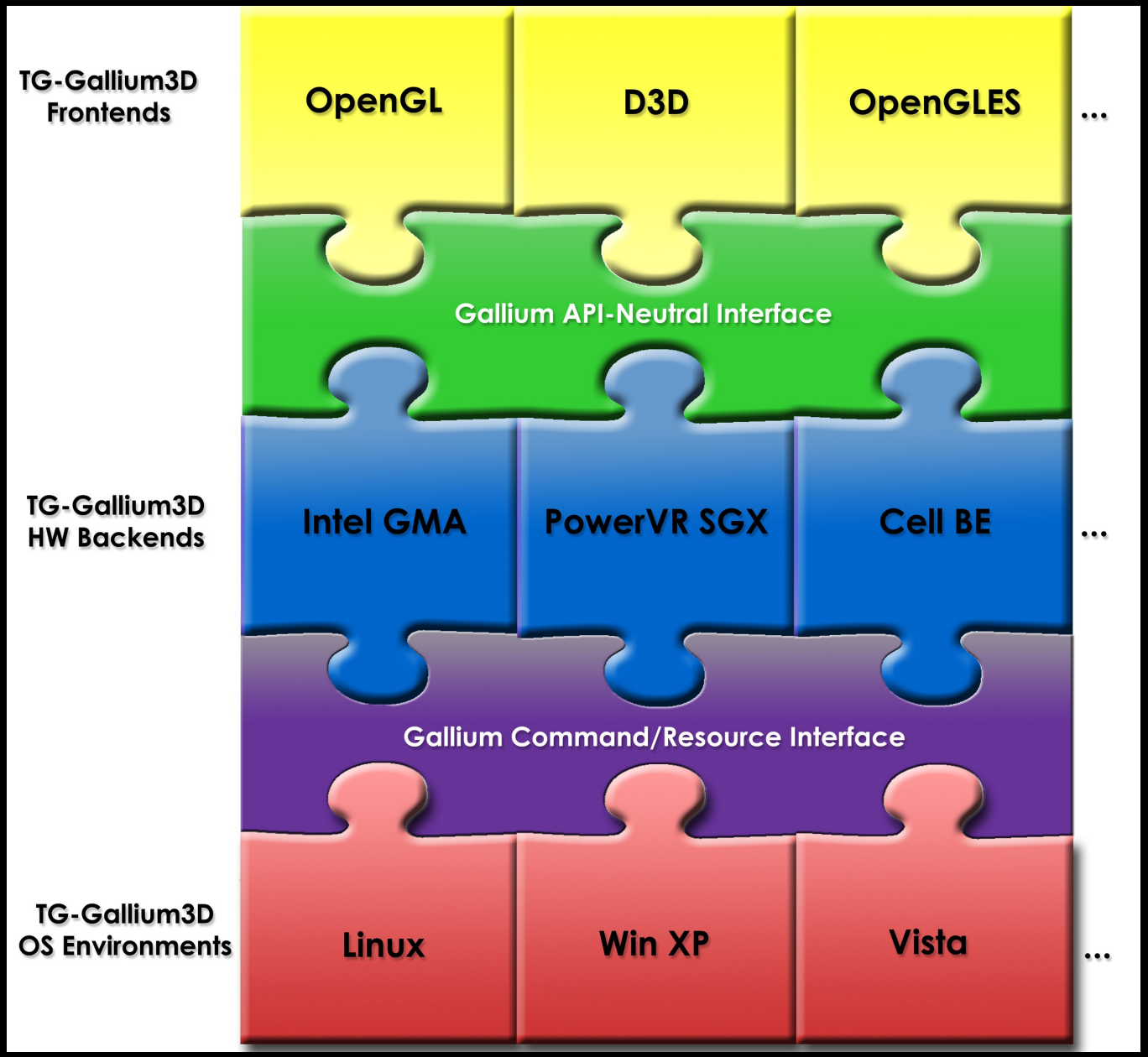


- Orange components... A lot of interfaces...
- Small piece of code, but complex.
- SOON: Split it up for a clearer stack.

Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

New diagram



Zack Rusin zack@tungstengraphics.com

Tungsten Graphics

Summary

- We're getting there.
- Interface churn should start to slow down, but some pain still to come.
- Focus to shift:
  - Performance
  - Conformance & correctness
  - Stabilization

Zack Rusin zack@tungstengraphics.com