

WebRender in the age of Vulkan



moz://a
all hands

WebRender
(today)

**Texture
updates**

**GPU
Cache
updates**

**Render
Task
submit**

Same thread!

| Render thread: WebRender

- Receive serialized frame data
- Update textures
- Update GPU cache rows
- Upload frame global data
- Bind render pass 1..N
 - a. Bind layer 1..M
 - i. Draw batch 1..K



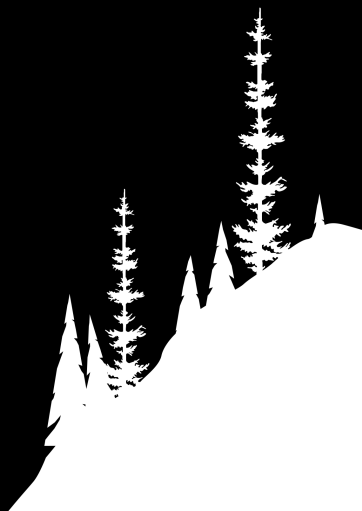
| Render thread: OpenGL driver

Manage
Staging
Memory

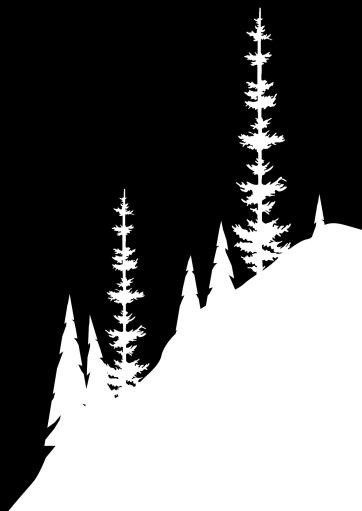
Manage
GPU
targets

Manage
resource
lifetimes

Match vertex buffers, render
targets, blend modes, and shaders
together...



| GL Driver:



| Meet Vulkan

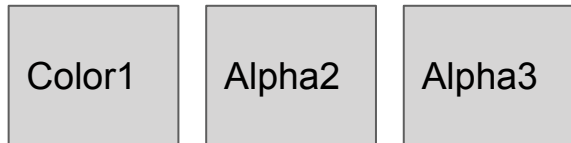
Low level graphics and compute API designed for multi-threading and minimal CPU overhead.



01: Let frame builder manage staging and device memory.

Frame & Scene builders

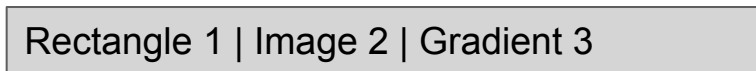
Render targets



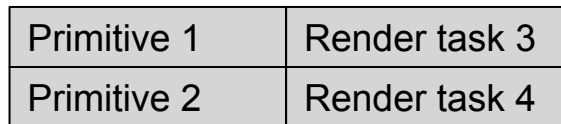
Staging memory



Instance data



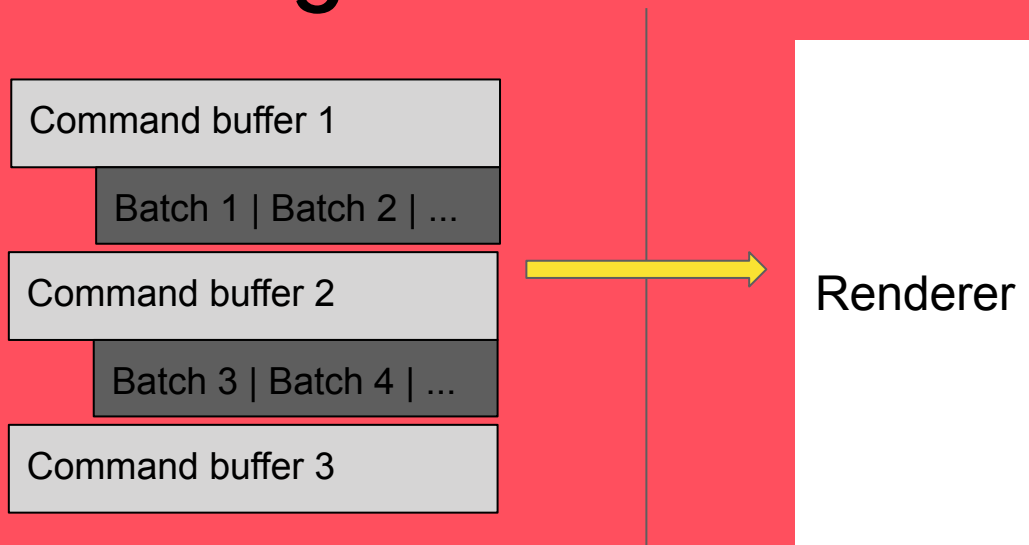
Metadata



Renderer

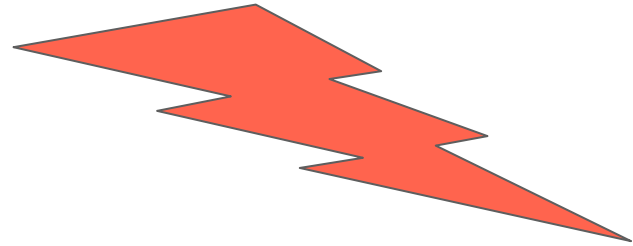


02: Record commands at frame building.



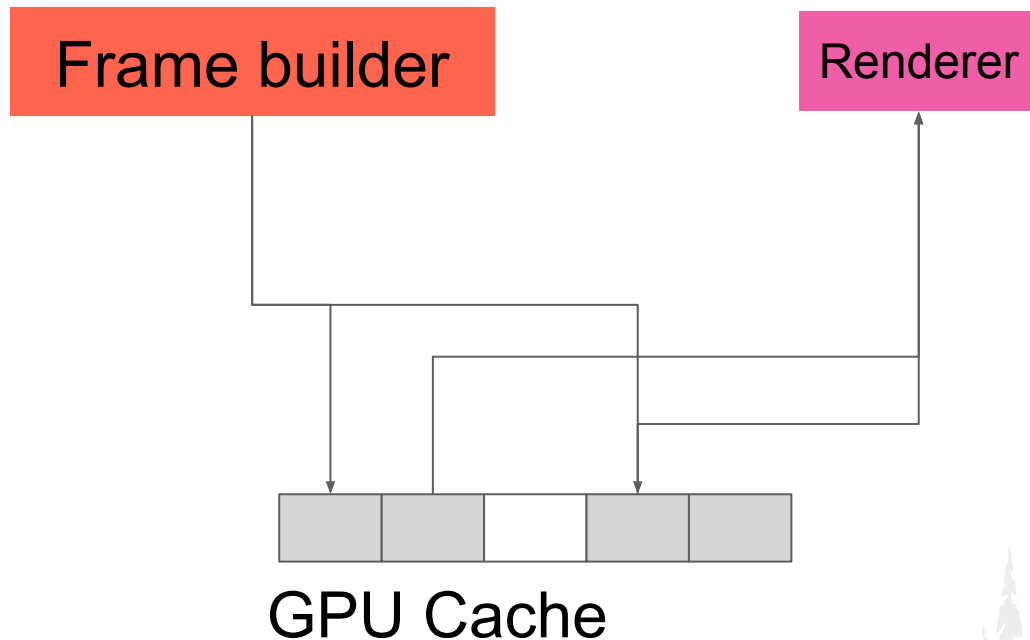
92-108	✧ alpha batches
92-92	✧ B_Image
93-93	✧ B_Image
94-94	✧ B_Image
95-95	✧ B_Image
96-96	✧ B_Blend
97-97	✧ B_Image
98-98	✧ TextRun
99-99	✧ B_Solid
100-100	✧ B_LinearGradient
101-101	✧ B_Solid
102-102	✧ B_Image
103-103	✧ B_Solid
104-104	✧ TextRun
105-105	✧ B_Blend
106-106	✧ B_Image

Batch breaks?..

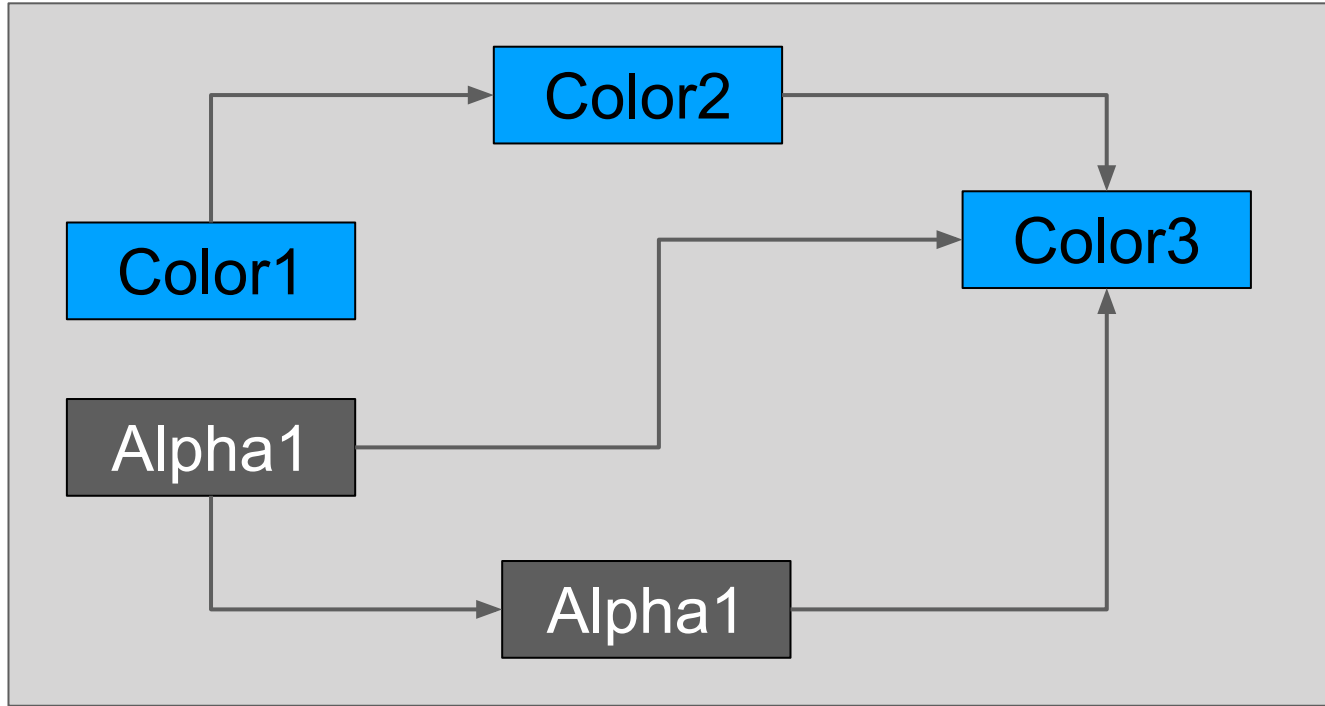


More batches!

| 03: Persistently-mapped GPU cache.



| 04: Über render pass.



WebRender (today)

Frame builder:

```
manage_memory()  
build_batches()
```

Render thread:

```
manage_memory()  
prepare_objects()  
record_commands()  
submit()
```

moz://a
all hands

WebRender
(future)

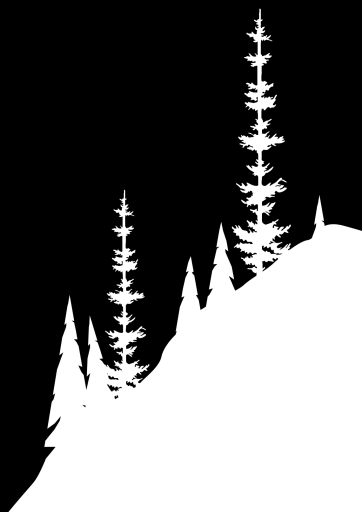
Frame builder:

```
manage_memory()  
prepare_objects()  
record_commands()
```

Render thread:

```
submit()
```

| Vulkan Driver:



Thank you!

