

A night sky with a full moon and a silhouette of a tree. The sky is filled with stars, and the moon is in the upper right. A tree silhouette is on the left side of the image.

GFXReconstruct

Brad Grantham, LunarG, Inc.
XDC 2022

GFXReconstruct

Tool suite for capturing and replaying graphics API calls

Speaker: Brad Grantham, LunarG Inc.

GFXReconstruct Technical Lead

Slides are available here:

<https://www.lunarg.com/news-insights/white-papers/GFXReconstruct-2022-XDC/>

GFXReconstruct - Agenda

- Overview
- Use Cases
- Capture, replay, and other tools
- Architecture - file format, repository structure, classes
- Live demo

GFXReconstruct - Overview

Source Available : <https://github.com/LunarG/GFXReconstruct>

- Captures commands to a file (aka “a capture”)
- Replays captures
- C++ and Python
- Linux, Android, Windows
- API-agnostic; Vulkan available in the Github repository
- Handful of additional tools

GFXReconstruct - Use Cases

Save an app's Vulkan commands and replay them repeatably and consistently

- Driver regression testing
- Architecture simulation
- Silicon bringup
- Bug reporting

Currently in use by several GPU, chipset, platform vendors

GFXReconstruct - Capturing An Application

Vulkan API layer “libVkLayer_gfxreconstruct.so” or .dll

- Use VK_INSTANCE_LAYERS or “gfxrecon.py capture” or VkConfig
- Hooks all core 1.3 Vulkan function calls and many extensions
- Captures most function inputs and outputs
 - (almost - e.g. outputs not stored on error return)
- Writes a blocks per call to capture file
 - Compressed by default

GFXReconstruct - Capturing An Application

(continued)

- Memory tracking - `GFXRECON_MEMORY_TRACKING_MODE` or `--memory-tracking-mode`

<code>pageguard</code> (default)	Attempt to detect writes to mapped buffers, write only changed pages
<code>assisted</code>	If application uses <code>vkFlushMappedMemoryRanges</code>
<code>unassisted</code>	Can just write all buffers all the time before Submit

GFXReconstruct - Capturing An Application

Trimming

- Can capture a range of frames or (on desktop) using a hotkey
- Can capture multiple ranges
 - E.g. `GFXRECON_CAPTURE_FRAMES=1, 2, 10-20`
- All graphics state up to the range is tracked
- When beginning a trimmed range, tracked state is stored in the capture file as state setup
- But conservative - can't know what future frames will reference

GFXReconstruct - `gfxrecon.py replay`

Plays Vulkan function call stream as close to the original as possible

- Select one of multiple GPUs: “`--gpu`”
- Can support replay on other driver or GPU model or even GPU family
 - Memory alignment, types not guaranteed
 - Can translate using “`-m`”: “`remap`”, “`realign`”, “`rebind`”
 - Chooses subclasses of `VulkanResourceAllocator`
 - Can mask off extensions: “`--remove-unsupported`”
- Will warn if function has different return codes in replay
 - Can also skip `Allocate` calls that failed in capture with “`--sfa`”

GFXReconstruct - gfxrecon.py replay

(continued)

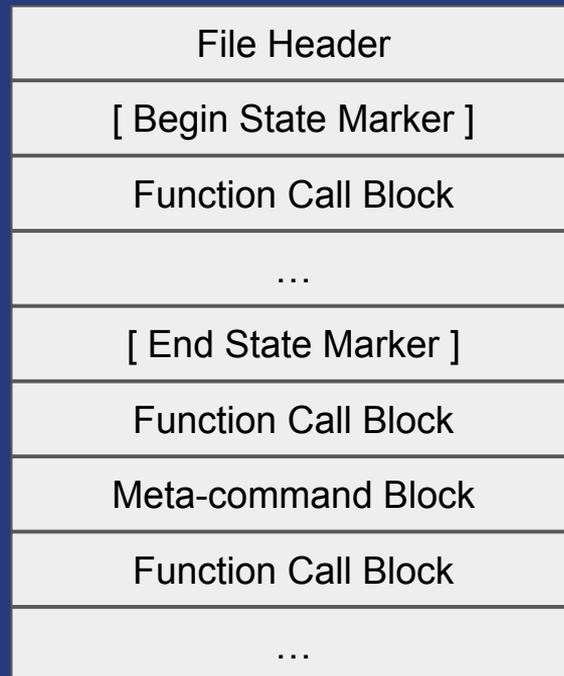
- Choose one of multiple captured surfaces to replay: “--surface-index”
 - Helpful on Android
- Save presented images: “--screenshots”, “--screenshot-all”
- Swapchain acquisition order not guaranteed even on same GPU
 - “Virtual Swapchain” renders to texture, then blits to swapchain image by replay index (can be disabled)
- Can attempt replay even on different platform with “--wsi”

GFXReconstruct -- Other Tools in the Package

- `info` - print useful information about a capture
- `optimize` - remove unused resources
- `compress` - change compression format
- `extract` - extract shader binaries for inspection or replacement
- `convert` - convert to e.g. JSONlines (command per line)
(new tool, will be in next release)

GFXReconstruct File Format

- File header followed by a series of blocks
- Meta-command blocks, e.g.
 - Blocks to mark beginning and end of state setup
 - Set up swapchain index for trimmed range
- Each Vulkan function call generates a block
 - API call ID
 - e.g. `ApiCall_vkCreateInstance = 0x1001`
 - Input, output parameters



BUT, the preferred way to process a capture is **to subclass Consumer !**

GFXReconstruct Architecture

Components

- CaptureManager - deal with API specifics, trimming, misc
- Encoder - serialize API call info and parameters
- FileProcessor - read blocks from a file, decode and call Decoders
- Decoder - deserialize API call info, call Consumers
- Consumer - take API call info, do *something* with it
 - E.g. VulkanReplayConsumer
 - E.g. VulkanStatsConsumer

GFXReconstruct

Source code directory structure

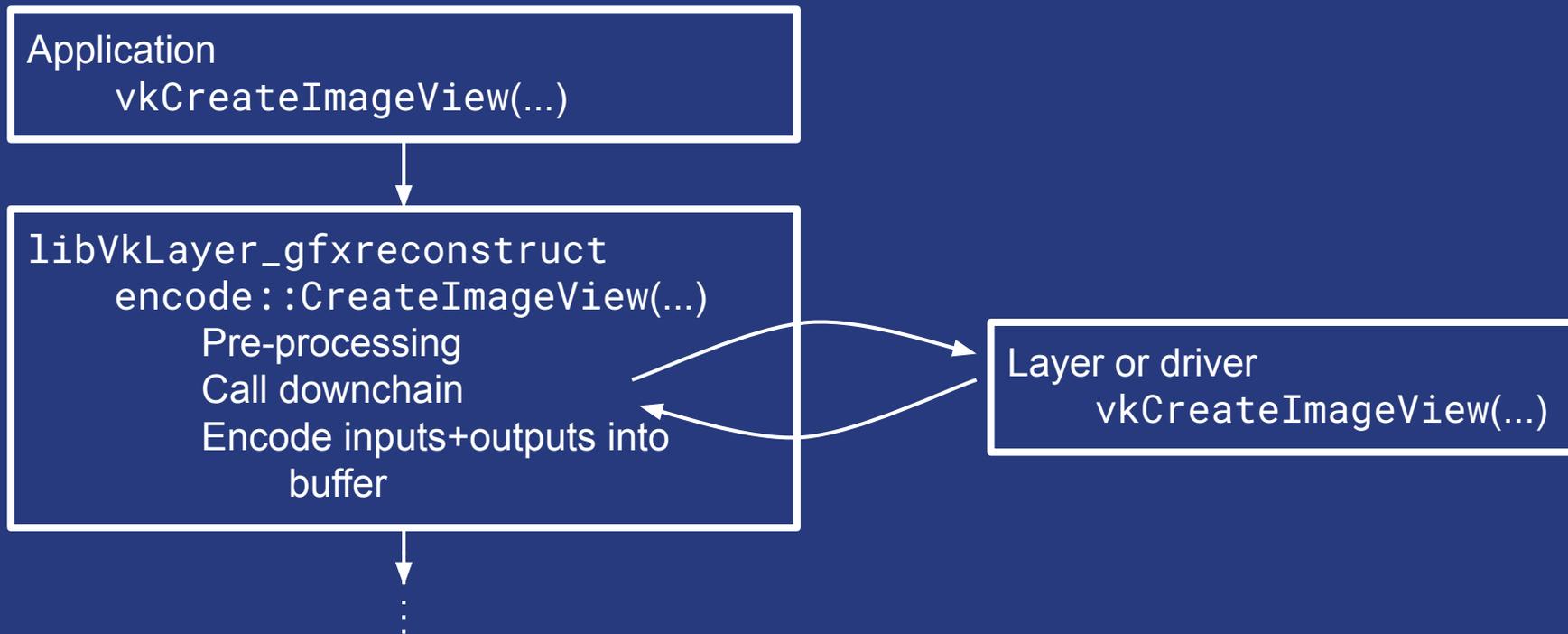
- `framework/`
 - `generated/` - generators & generated code *is checked in*
 - `encode/` - capture manager, handwritten capture, state tracking
 - `decode/` - file processing, decoding, replay, and other consumers
 - `format/` - file format metacommand structs, API call IDs
 - `util/` - etc

GFXReconstruct Architecture

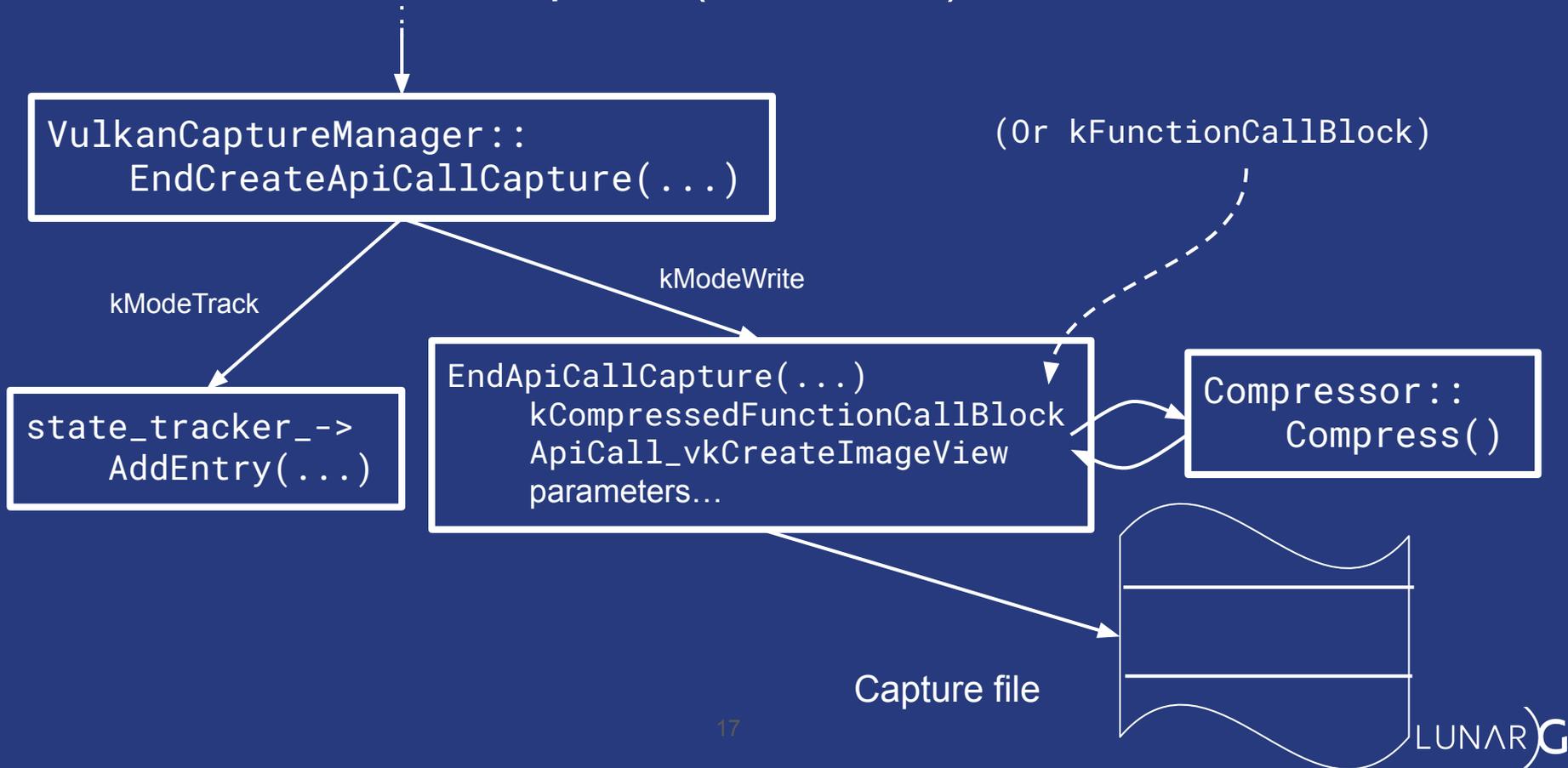
Source code directory structure - cont.

- `tools/` - settings, tool `main()`s, etc
- `layer/` - Vulkan API layer dispatch table, boilerplate, manifest

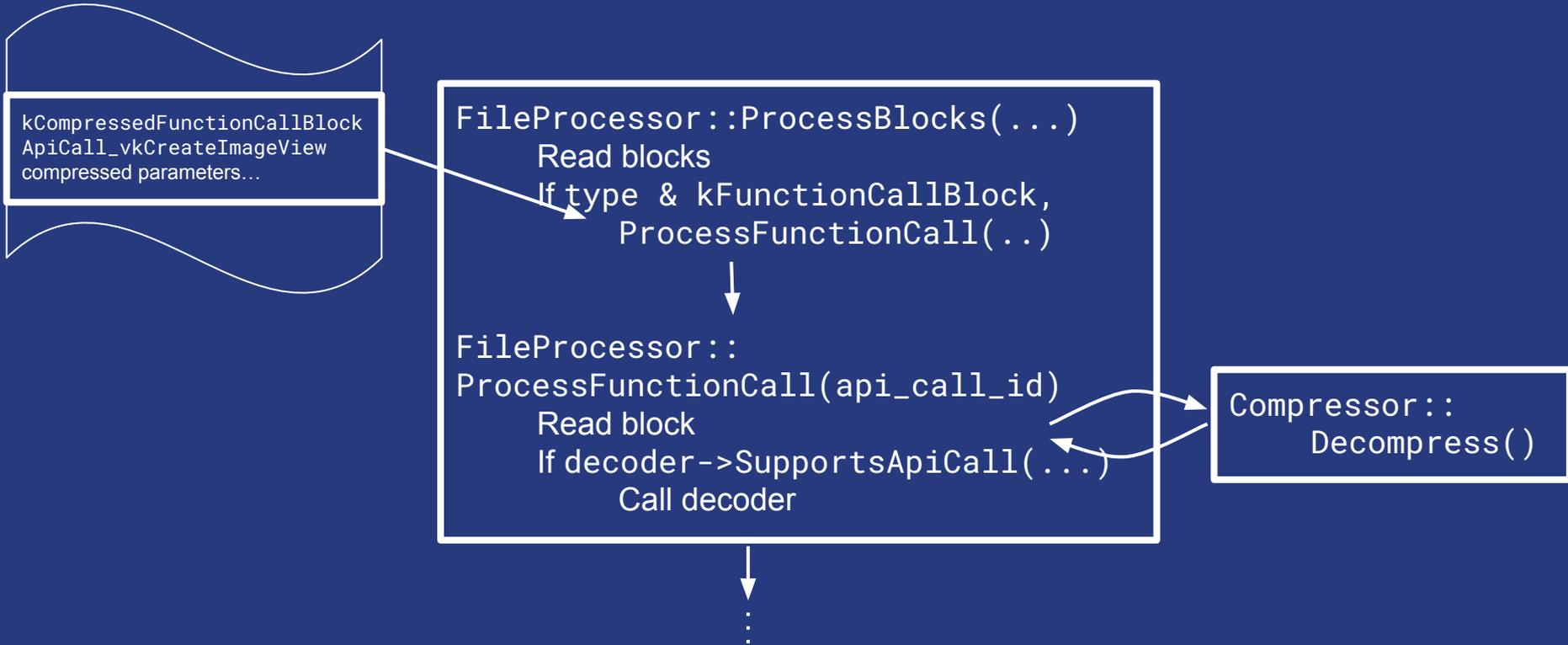
GFXReconstruct - Capture



GFXReconstruct - Capture (continued)



GFXReconstruct - Replay



GFXReconstruct - Replay (continued)



```
VulkanReplayDecoder::  
    DecodeFunctionCall  
  
Decode_vkCreateImageView  
    Decode all parameters for vkCreateImageView  
    Handle IDs, simple types, struct chains  
    For all consumers,  
        consumer->Process_vkCreateImageView(..)
```



GFXReconstruct - Replay (continued)



```
VulkanReplayConsumer::  
    Process_vkCreateImageView(...)  
        Map handles  
        Preprocess data  
        vkCreateImageView(...)  
        Postprocess data  
        Store created handles
```

GFXReconstruct - 1.0 Release (imminent)

Tasks left to be done

- Version the file format - backward compatibility
- Identify and tackle any open issues and PRs that really should be resolved

GFXReconstruct Demo

- Capture vkcube with VkConfig and replay with installed SDK
- Capture vkcube with trimming and replay that capture
- `gfxrecon.py info`
- Build from source
- Capture vkcube and replay the capture
- Convert to human-readable and view the results

Thanks!

<https://github.com/LunarG/GFXReconstruct>

<https://lunarg.com>

brad@lunarg.com

LUNAR)G[®]

