Mobile: The Future of VR

ARM

Sam Martin, ARM
Juan Wee, Samsung
Alon Or-bach, Samsung

Game Developers Conference
1st March 2017

SAMSUNG

http://www.telephonearchive.com/phones/sc/sc-mckinley-fig8.html

Not Samsung

SAMSUNG

Flickr/Julien Prive

SAMSUNG

http://rlogin.dk/sites/messenger-dead-due-to-Skype-but-good-replacements-exist/page2.html

SAMSUNG

https://firefoxos.mozfr.org/post/2015/09/Nicolas-Hoizey-pionnier-creation-jeu-Firefox-OS

SAMSUNG

SAMSUNG

SAMSUNG

SAMSUNG

https://en.wikipedia.org/wiki/File:Game-Boy-FL.png

SAMSUNG

SAMSUNG

https://www.cnet.com/products/samsung-gear-vr/


4,800
MISSILE WARNING

VR HMD

6m+

2016
GearVR, PSVR, Rift, Vive

VR HMD

?

2017
GearVR, PSVR, Rift, Vive
Daydream, Windows 10 VR

SAMSUNG

Data: SuperData

Mobile VR Session

< 10 min

Mobile VR Sessions per Month

50 times

SAMSUNG

Data: SuperData

VR Content Consumption

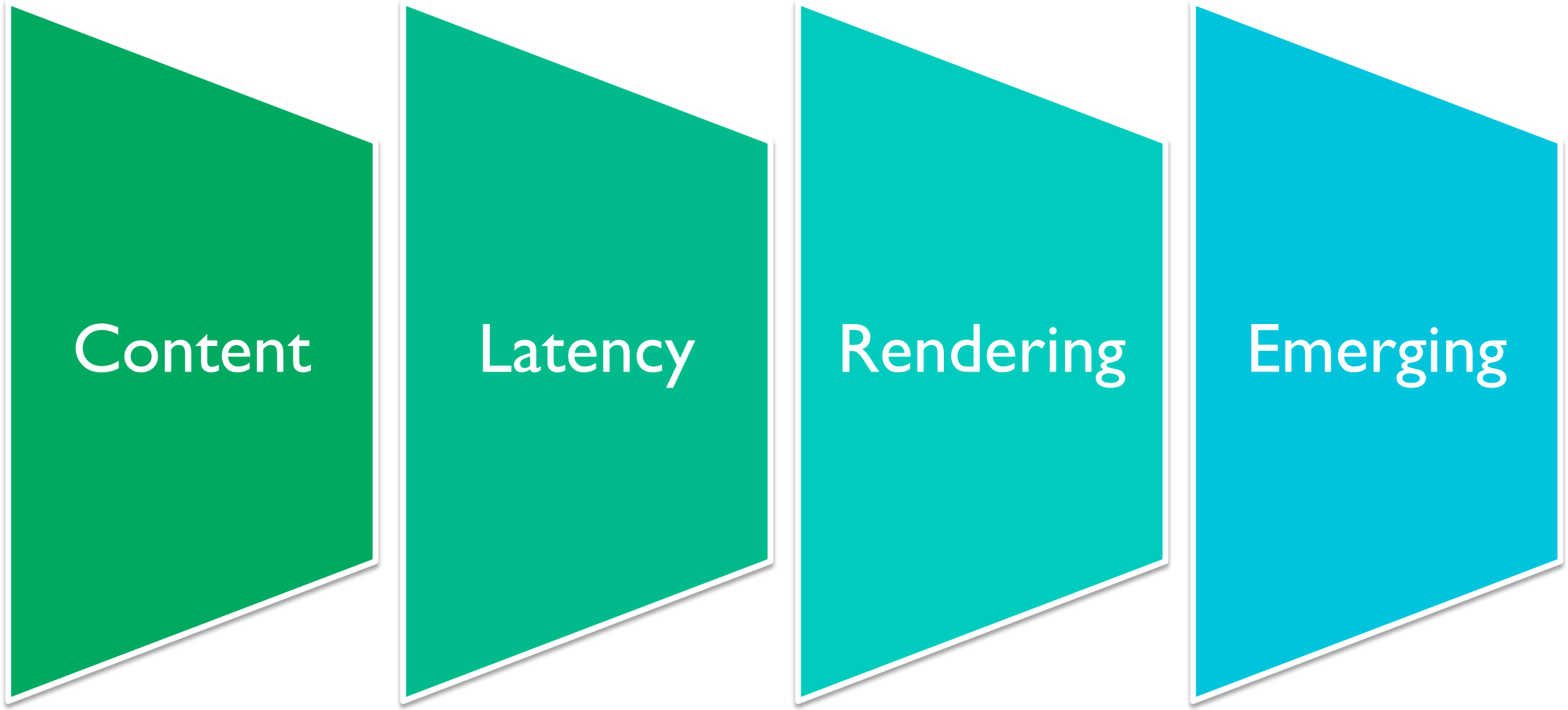
44 %

Games

SAMSUNG

Data: SuperData

Frontiers



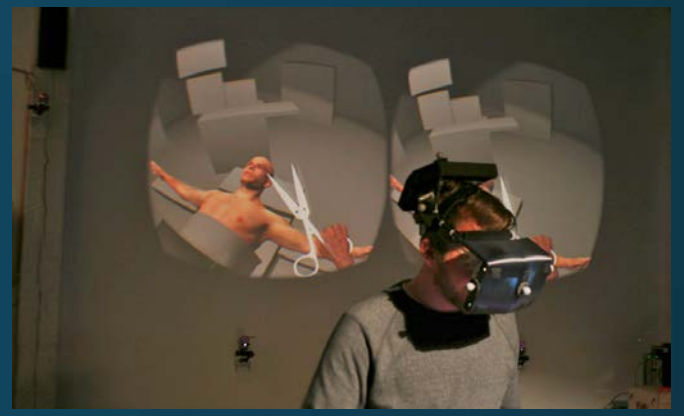
Content frontier

SAMSUNG

Juan Wee, Samsung Electronics

Games

Experiences



Training

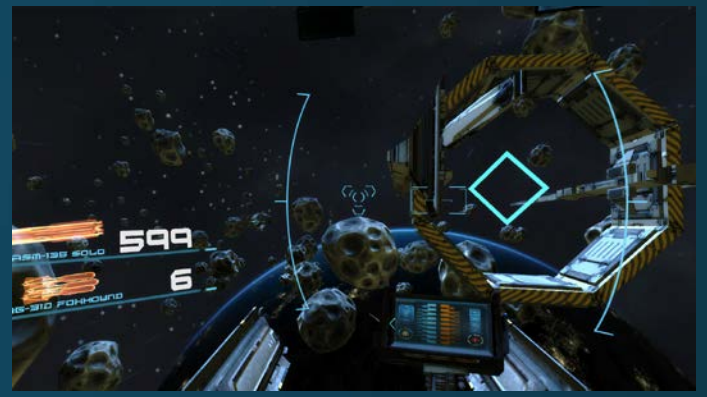
Education

TV and Film

Advertising

SAMSUNG

Immersive but stationary



Use impressive scale and perspective

Captive audience

SAMSUNG

VR Video Content Consumption

10m+

Hours

2016 GearVR

SAMSUNG

SAMSUNG

SAMSUNG

Bridges physical distances of places and experiences

Bridges personal distances

Brings people together to share experiences

SAMSUNG

All of this comes at a cost…

SAMSUNG

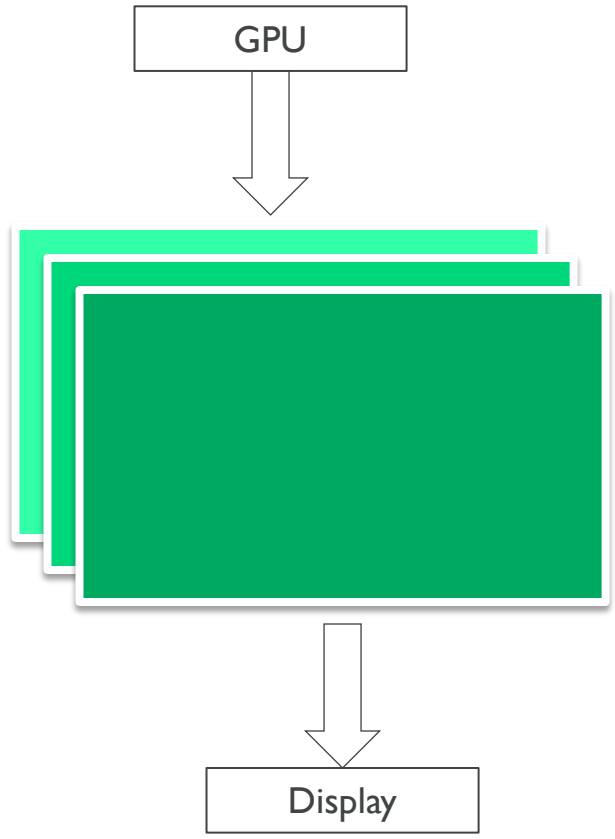
Latency frontier

Sam Martin, ARM

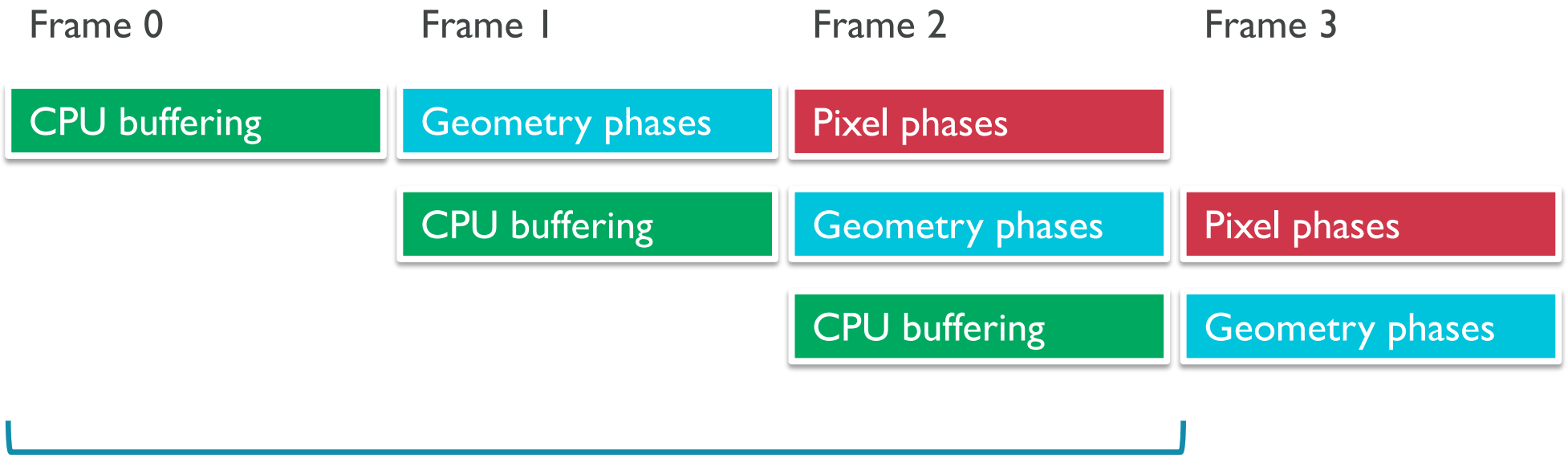
ARM

<20ms Latency

- "Motion to photons"
  - 20ms is accepted imperceptible latency
- But… GPUs are throughput processors
- Usually ok to increase latency if it improves throughput
  - Android can/may triple buffer
  - Graphics pipeline spread over multiple frames



Worst-case frame pipelining



Up to 50 ms @ 60 Hz!

Conclusion:
We need to shortcut this pipeline for latency-sensitive changes

Composition in “traditional” Android apps

App →

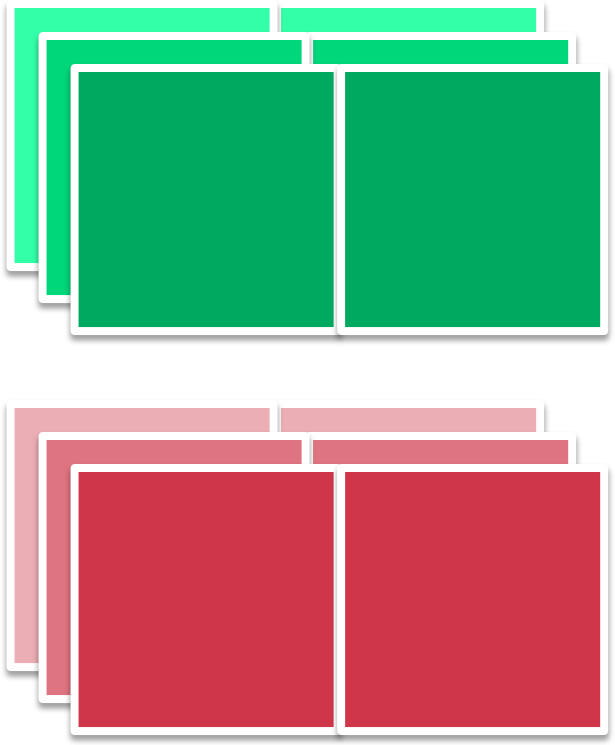
UI,
Notifications,
Video,
Buttons,
etc

Android
compositing
process

GPU

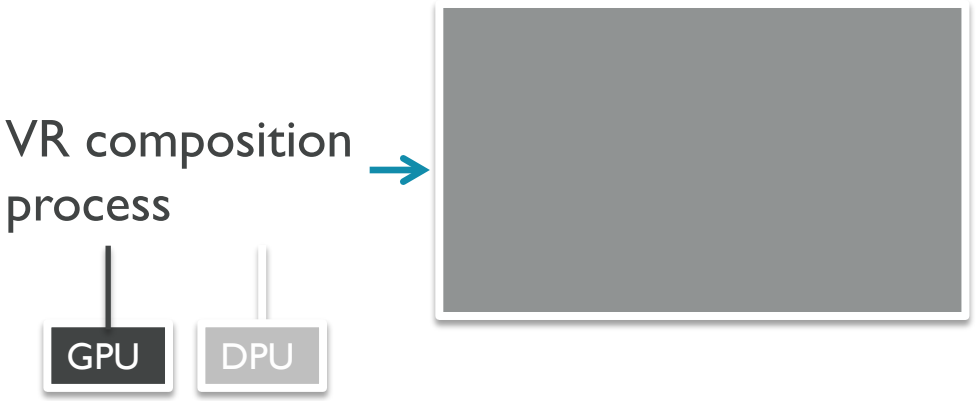
DPU

VR application



…

- Head-tracked layers
- Head-locked layers
- System layers
  - UI / notifications / input
- Not just flat textures!
  - Quad (video)
  - Cylindrical
  - Cubemaps



VR composition

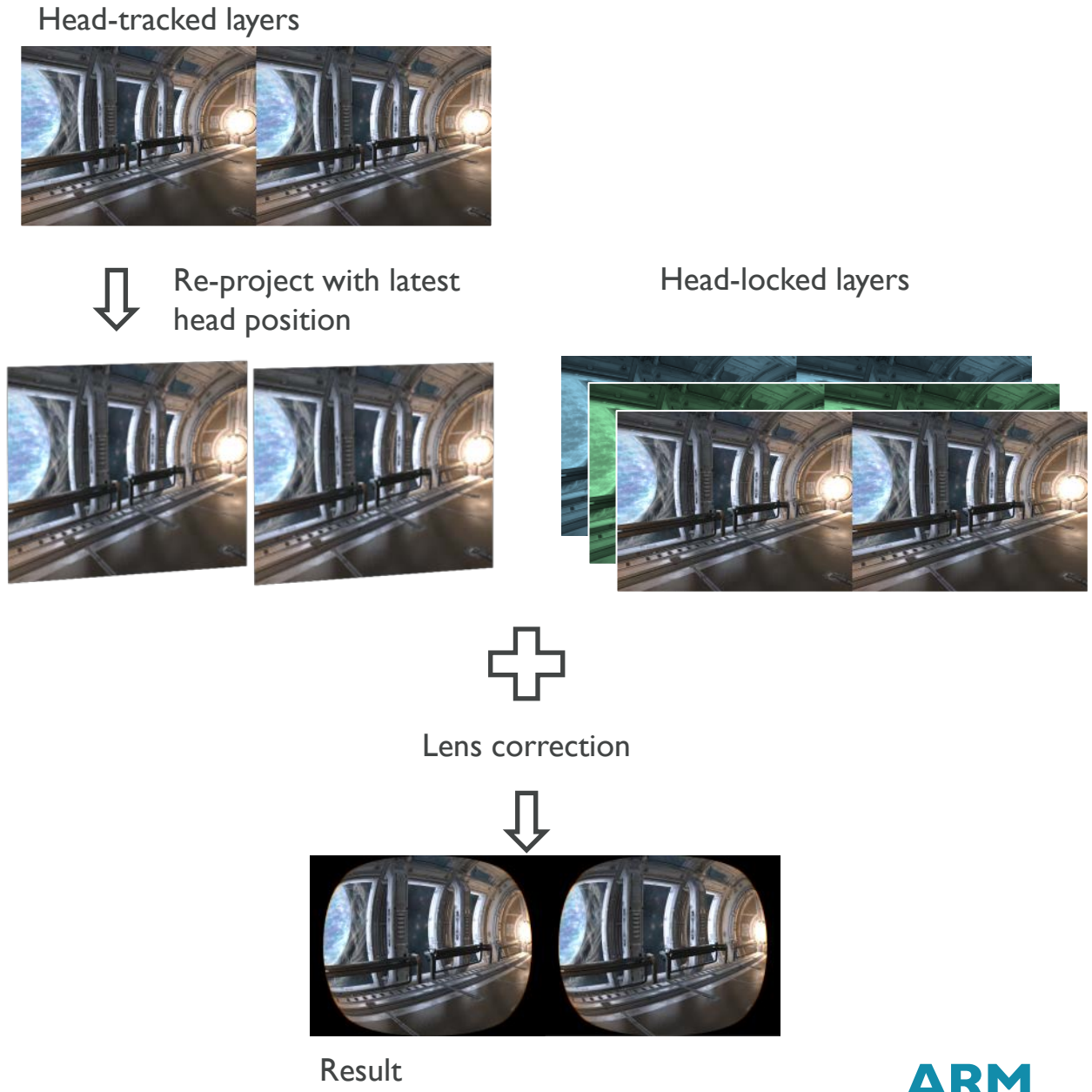
- Logically, a just-in-time process that :

  Re-projects head-tracked layers

  Composites all layers

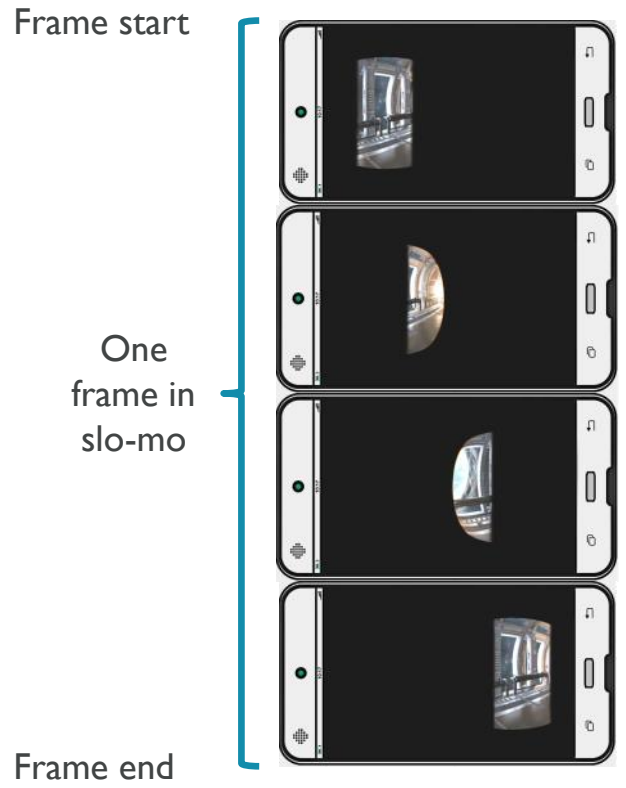
  Lens-corrects the result

- 15-30% of the GPU + a lot of bandwidth

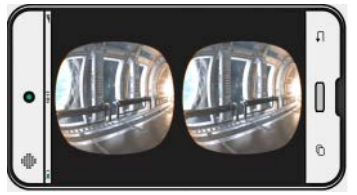


Lots of fiddly details

- Two more problems:
  - Display transfer time too long: 16.6ms @ 60 Hz
  - VR devices need low persistence panels to reduce blurring
- OLED panels are ideal – they progressively illuminate pixels
  - Low persistence and each pixel has a unique illumination time
- All GearVR and Daydream devices have OLED panels
- TLDR: Can use these properties to achieve latency goals



But looks like this to the end user…

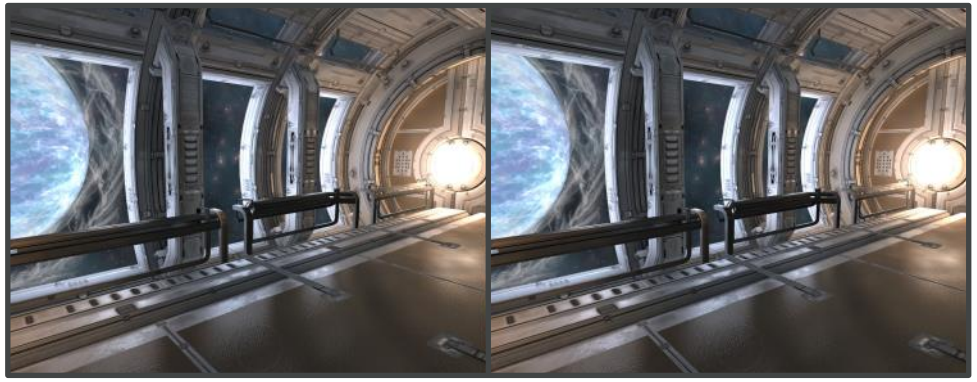


ARM

Result: 2 GPU interleaved processes

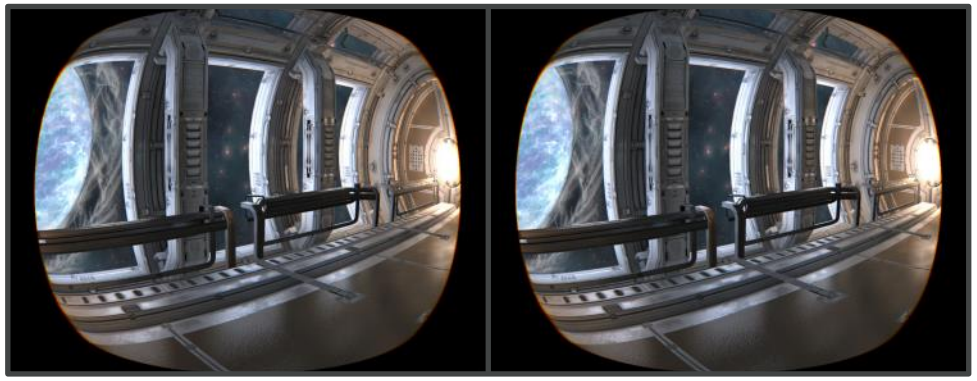
2x 1024x1024

Application draws eye buffers @ 30-60 Hz



- Rotation-only pipeline latency = ~8.3ms
- Decoupled from application
  - App can drop frames (sort of)
- OLED panels aren't cheap → limits availability
- Latency for other inputs is still perceivable

Re-project with latest head rotation 2 'slices' @ 60 Hz



2560x1440

Where does this go next?

- More stable system operation
  - Too much judder in real world systems today
    → Use logcat-based tools to see what happens today
  - Need more sweeping defences against this

- Latency of other inputs
  - E.g. animations, → Spacewarp?
  - Controller inputs, camera inputs, ... → active area of interest
  - Audio! (including 3d audio)

Where does this go next?

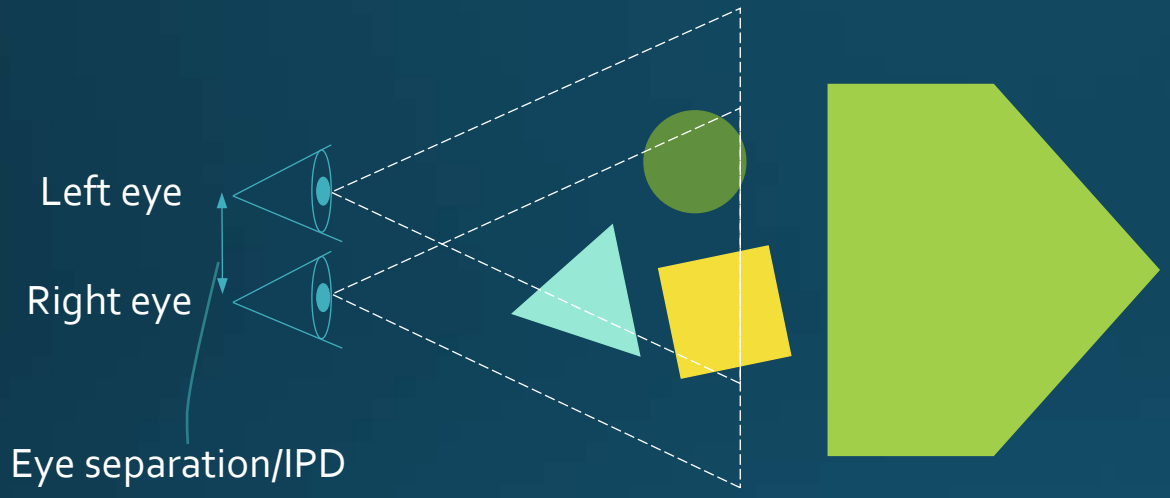
- 90Hz, 120Hz, … higher framerates?
  - Yes, but not about latency – diminishing returns beyond 20ms
  - Higher refresh rate → less 'flicker', less blur, brighter displays
  - Higher refresh rate → better panel response times
  - Higher refresh rate → LCD panels → lower cost → more devices
- Attention to detail

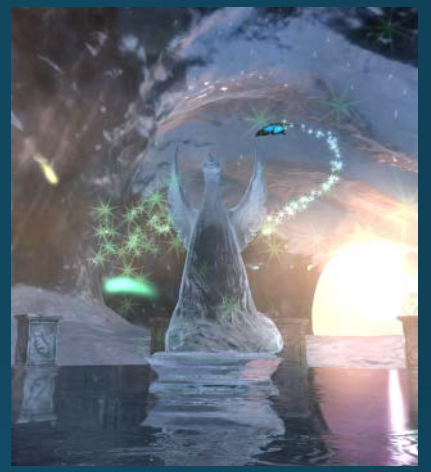
Rendering frontier

SAMSUNG

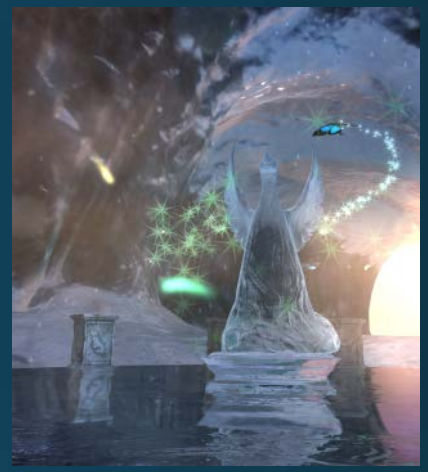
Alon Or-bach, Samsung Electronics

Stereo output 2x work in ½ the time!?





Left eye



Right eye

- Additional app overhead to prepare 2x draws
- 2x driver overhead to handle draws
- 2x vertex-tiling process

…but in ½ the time to get the same frame rate

CPU overhead

Geometry overhead

SAMSUNG

Multiview

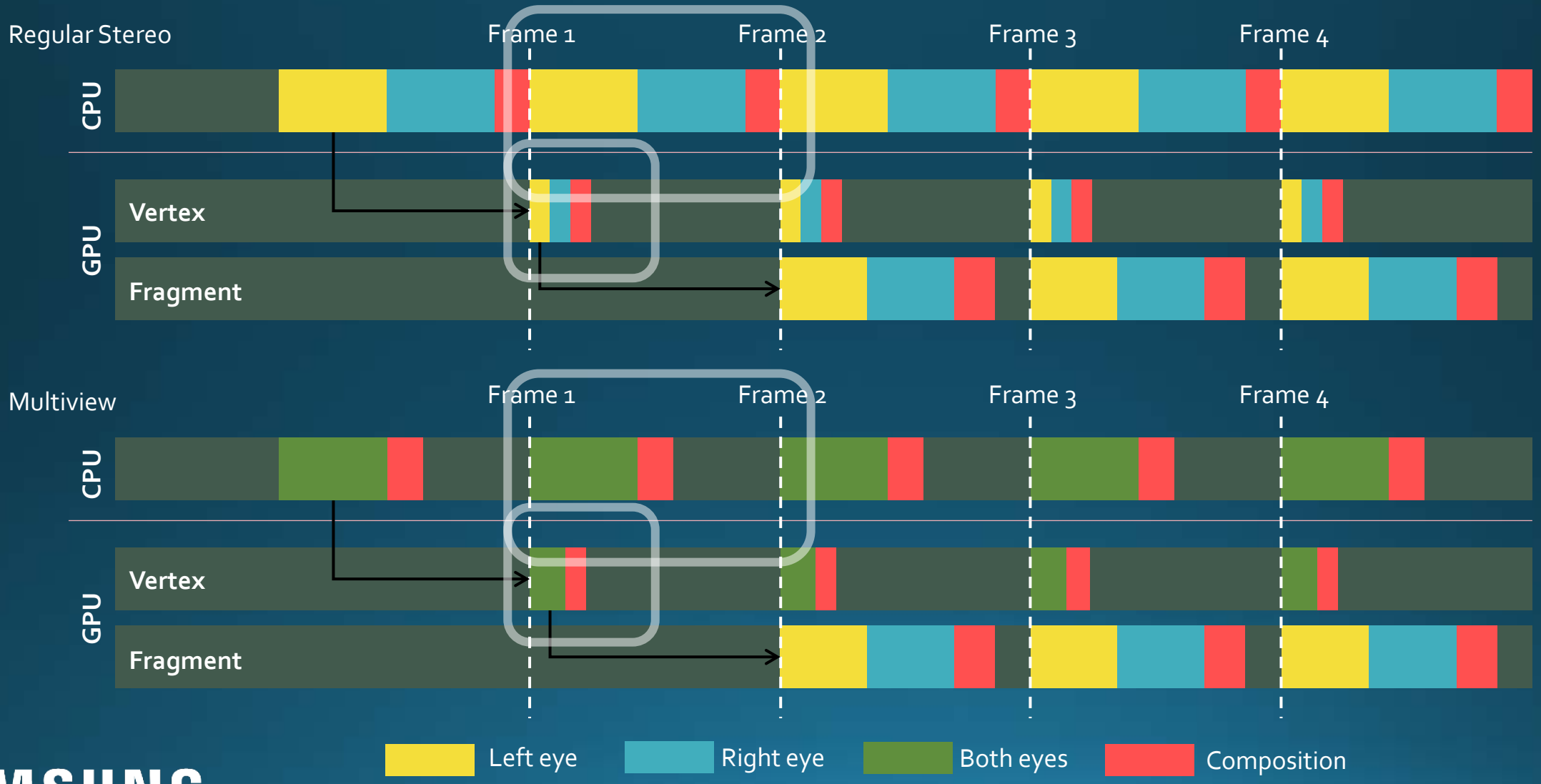
1. Draw into a texture array
2. Shaders see a new constant: `g_ViewID`
3. Use it to vary shading between eyes!

- Pros:
  - Simple!
  - Widely supported
  - Efficient on existing hardware
  - Generalises beyond 2 views

- Support in Unreal (4.14+) and Unity (5.6 beta)

SAMSUNG

Multiview illustration on Mali



SAMSUNG

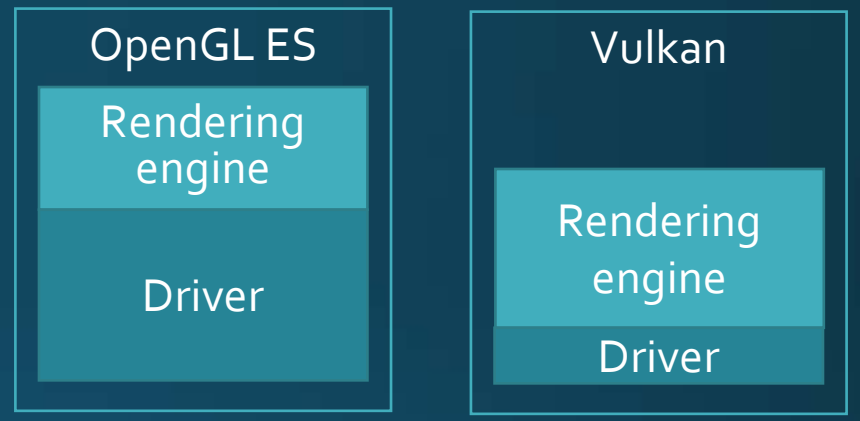
What is Vulkan?

- Cross-platform open standard next-gen graphics & compute API
- A more explicit alternative to OpenGL / OpenGL ES
- Features
  - Cross-platform & designed for modern GPU architectures
  - Low-overhead drivers
  - Explicit control / predictable performance
  - Multi-threading friendly

SAMSUNG

Why Vulkan benefits VR on mobile

- Reduced CPU overhead
  - Draw calls no longer the bottleneck!
- Decreased power consumption
  - Lower overhead and multithreading
- Or more impressive graphics



SAMSUNG

ARM

Vulkan: Improved framerate stability

- Engine/Application has far more control over when work happens
- Heavy lifting done away from critical path
  - Building resources (command buffers, pipelines)
  - Shader compilation
- Pipeline caches enable re-use of resources

=> Avoid draw-time stuttering!

Vulkan features enabling VR

Technique	Component	Feature / extension
Multiview rendering	• Game	• Multiview or instancing
Lens correction & image warp	• VR compositor	• Queue priorities or cross-process sharing and synchronization
Rendering to front buffer	• VR compositor	• Shared presentable images

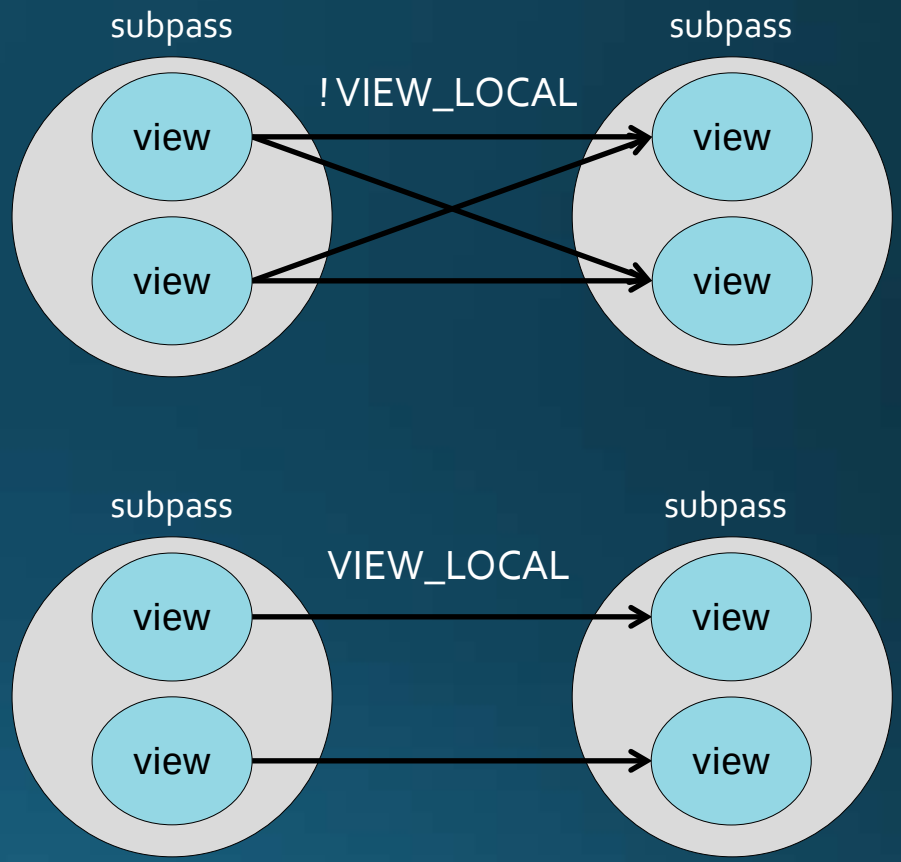
SAMSUNG

Vulkan: Multiview

- **VK_KHX_multiview** experimental extension just released!
  - Keen to get developer feedback
  - Warning! It **will** be removed when functionality finalized
- Accompanied by SPIR-V extension **SPV_KHR_multiview**
- Based on GL_OVR_multiview, redefined to use render passes
- Commands executed across multiple views
- Shaders can differ per-view using **ViewIndex**

Vulkan: Multiview

- Aim to achieve best performance across differing implementations
- Enables recording command buffers that differ between views
- Introduces mechanism to associate render pass with multiple views
- Supports tiling-friendly VIEW_LOCAL dependencies



SAMSUNG

Emerging frontiers

Sam Martin, ARM

ARM

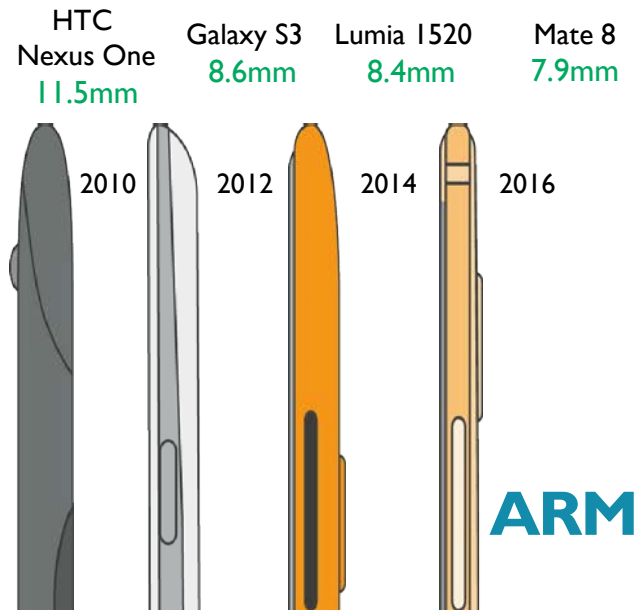
Performance limits as thermal limits

- Performance on premium mobile is limited by capacity to dissipate heat

- Phones are not getting bigger
- Process shrinkage is getting more expensive

Must find ways to do more in less

- Some inefficiencies that will give us a short term boost
- Need to look for domain-specific optimisations
  - Foveated rendering? HW composition? VR video?



Alternative facts

- Increase your thermal limit

- Phone-in-a-headset form factor
  - Low cost, compelling experience, largest potential market

- Standalone ("all in one") VR devices
  - Better cooling options, custom display

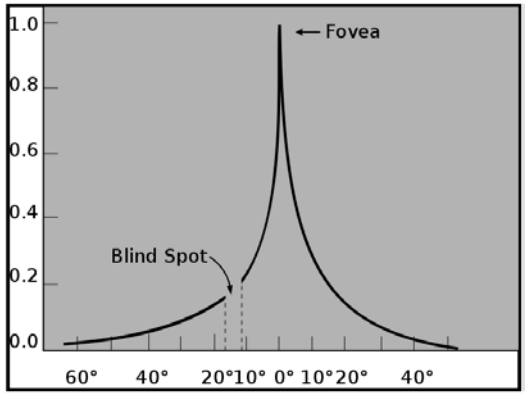
- Separate headset and device form factors
  - Thermal limits dependent upon non-headmounted device – phone to a rucksack
  - Lightest possible headset

Foveated rendering (two flavours)

- Lens-matched
- Gaze-tracked

- May / may not have the same solution
- Both need the app to ‘opt in’
- 4k panels
  - Need a 2x improvement to realise

- Many other uses for eye tracking
  - IPD measurement, biometric sign in, social



Other active fields

- Tracking – near future
  - Positional head tracking
  - Tracked controllers
  - Tracking people
  - Tracking environments
- Video
  - Position-independent (“lightfield”) video
  - 360 video
  - Voxel/”other” video

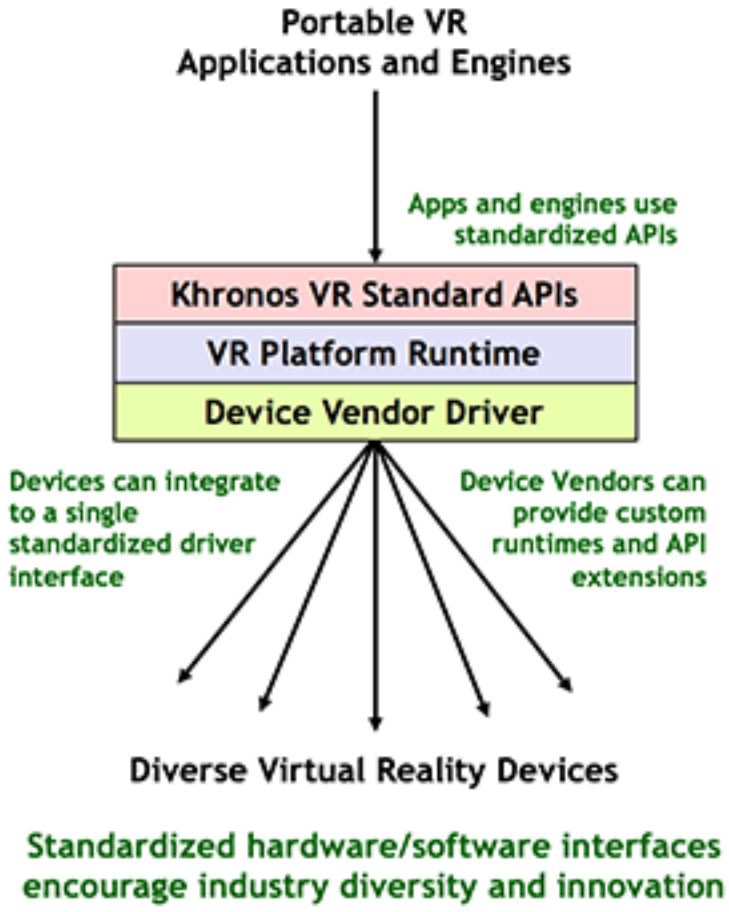
OpenXR™

AMD ARM EPIC GAMES Google Imagination intel

LG LUNARG MEDIATEK NOKIA NVIDIA oculus

QUALCOMM RAZER SAMSUNG sensics tobii

unity VALVE VeriSilicon 兆芯 zSpace



Standardized hardware/software interfaces encourage industry diversity and innovation

Thanks! Questions?

- Sam Martin, ARM — sam.martin@arm.com — @palgorithm
- Juan Wee, Samsung — juan.wee@samsung.com
- Alon Or-bach, Samsung — alon.orbach@samsung.com — @alonorbach

- Also highly recommended at GDC 2017:
  - "Get the Most from Vulkan in Unity with Practical Examples from Infinite Dreams"
    - 10-11am tomorrow (Thursday, room 3022)
  - "Achieving Console Quality Games on Mobile with Digital Legends Case Study"
    - 5-6pm today (Wednesday, room 3007)

ARM