Portable Graphics Abstraction in Rust

Dzmitry Malyshau · Markus Siglreithmaier

ONE Rust API
MANY Backends

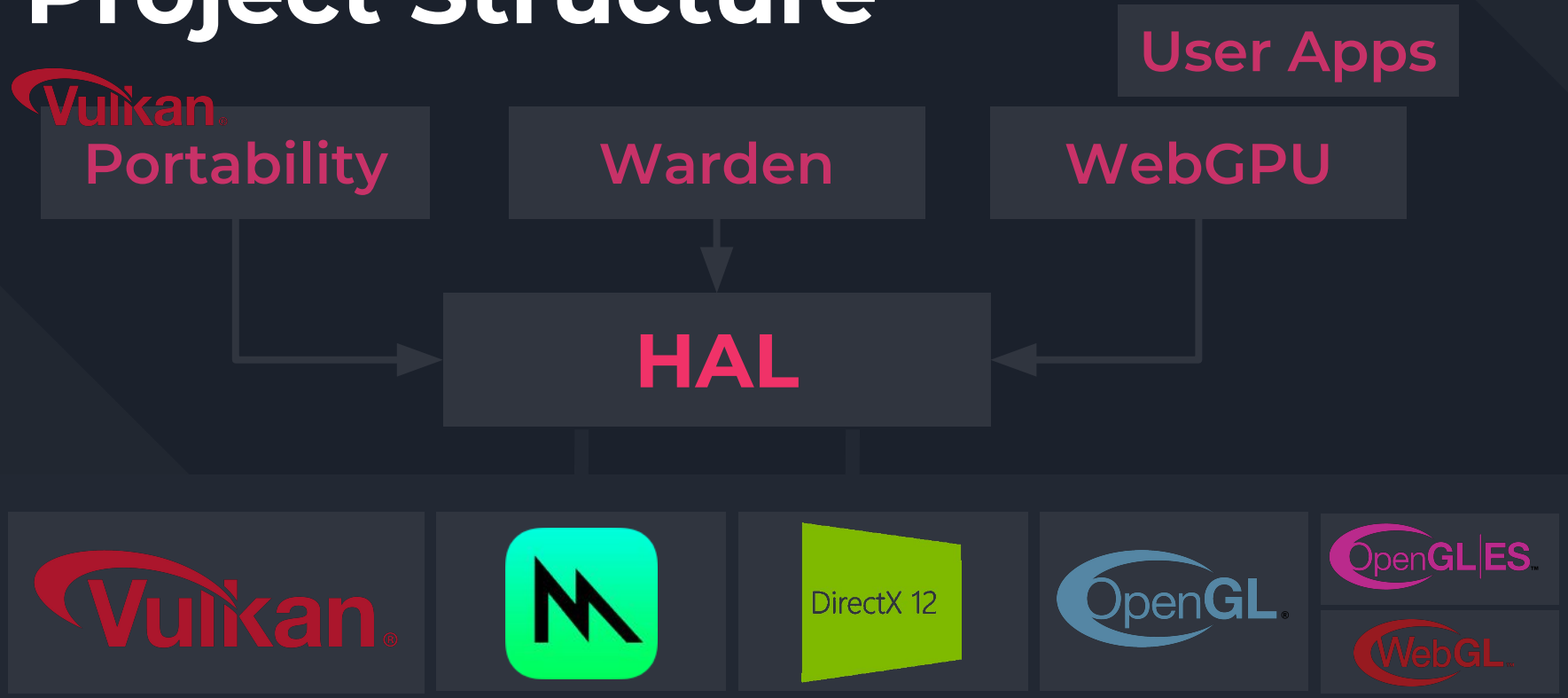
History

2014	Birth + Thread model
2015	Command buffers + Pipeline states
2016	D3D11 & Metal backends
2017	Hardware Abstraction Layer + SPIRV
2018	???

History

2014	Birth + Thread model
2015	Command buffers + Pipeline states
2016	D3D11 & Metal backends
2017	Hardware Abstraction Layer + SPIRV
2018	???

Project Structure



Warden ref-test framework

Scene	resources & jobs
Resource	buffer | image | shader | ...
Job	render pass | transfer commands
Test	jobs & expectation

Warden Example: Resource

```
"im-color": Image(
    kind: D2(1, 1, Single),
    num_levels: 1,
    format: Rgba8Unorm,
    usage: (bits: 4),
),
"pass": RenderPass(
    attachments: {
        "c": (
            format: Some(Rgba8Unorm),
            ops: (load: Clear, store: Store),
            layouts: (start: General, end: General),
        ),
    },
    subpasses: {
        "main": (
            colors: [("c", General)],
            depth_stencil: None,
        )
    },
    dependencies: [],
),
```

Warden Example: Pass

```
"pass-through": Graphics(
            descriptors: {},
            framebuffer: "fbo",
            pass: ("pass", {
                "main": (commands: [
                    BindPipeline("passthrough.pipe"),
                    SetViewports([
                        Viewport(
                            rect: Rect(x: 0, y: 0, w: 1, h: 1),
                            depth: (start: 0.0, end: 1.0),
                        )
                    ]),
                    SetScissors([
                        Rect(x: 0, y: 0, w: 1, h: 1),
                    ]),
                    Draw(
                        vertices: (start: 0, end: 3),
                    ),
                ]),
            }),
        ),
```

Rust features

Nice bits

- Send/Sync
- Immutability
- Traits and Associated Types
- Multiple repositories

Rust features

Missing bits

- `#[non_exhaustive]` enums
- Struct alignment (shaders)
- `impl<T> IntoIterator for [T; N]`
- exclusive cargo features

Backend Trait

- Main entrypoint for users
- Zero-cost abstractions!

```
pub trait Backend {
    type Device:                Device<Self>;
    type CommandQueue:          RawCommandQueue<Self>;
    type Surface:               Surface<Self>;
    ...

    type ShaderModule:          Debug + Any + Send + Sync;
    type Framebuffer:           Debug + Any + Send + Sync;
    type Buffer:                Debug + Any + Send + Sync;
    type ComputePipeline:       Debug + Any + Send + Sync;
    ...
}
```

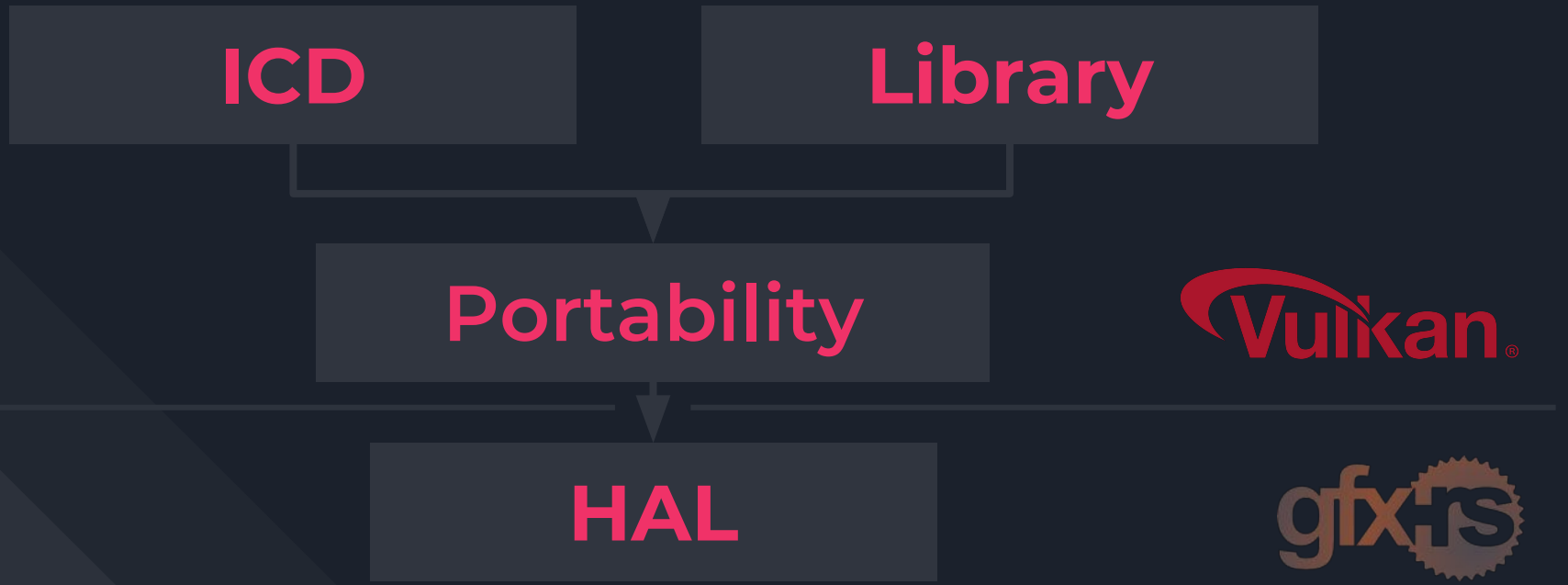
2018 Roadmap

Vulkan Portability

- Vulkan Portability Initiative by Khronos
- Portable **subset** of the Vulkan API

Goal: Use HAL for implementing the subset!

gfx-rs portability



WebGPU

- Intro to the W3C efforts
- https://github.com/kvark/webgpu-servo
- … WebVulkan?

Ecosystem

ggez

WR WebRender

Amethyst

Vulkano

Three-rs

Questions?

Blog: http://gfx-rs.github.io/

Code: https://github.com/gfx-rs/gfx