Vulkanised 2023

The 5th Vulkan Developer Conference
Munich, Germany / February 7–9

An Introduction to Vulkan

Johannes Unterguggenberger

TU Wien, Huawei

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

Schedule

PART 1:

Setup	Lecture	Coding Session
10 min	**20** min	**90** min
Starts at 09:00	Starts at 09:10	Starts at 09:30

PART 2:

Lecture	Coffee Break	Coding Session
15 min	**25** min	**80** min
Starts at 11:00	Starts at 11:15	Starts at 11:40

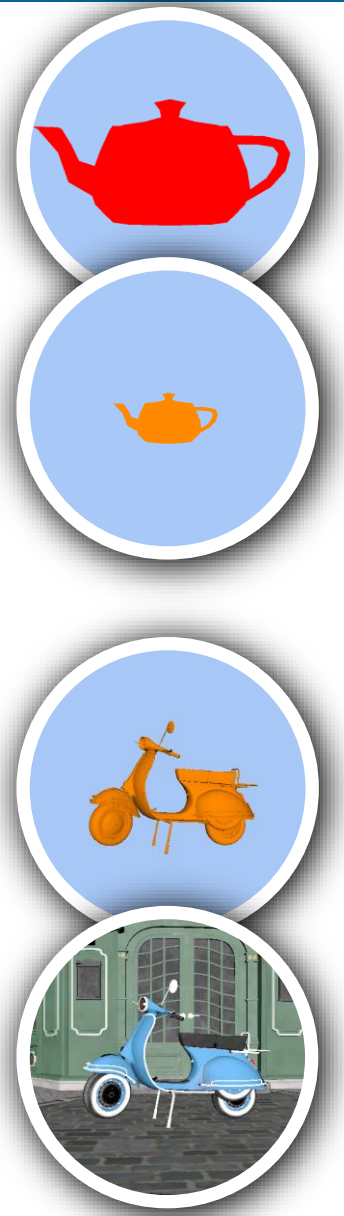
Lunch Break 13:00 – 14:00

PART 3:

Lecture	Coding Session	Coffee Break
15 min	**65** min	**30** min
Starts at 14:00	Starts at 14:15	Starts at 15:20

PART 4:

Lecture	Coding Session	Closing
20 min	**70** min	**10** min
Starts at 15:50	Starts at 16:10	Starts at 17:20



Vulkanised 2023

The 5th Vulkan Developer Conference
Munich, Germany / February 7–9

PART 2

- Graphics Pipelines
- Uniform Buffers
- Descriptors

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

PART 2

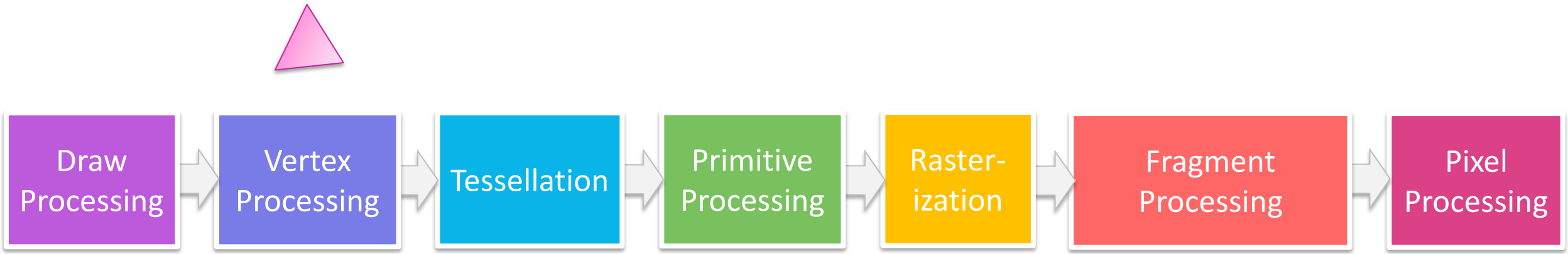
- **Graphics Pipelines**
- Uniform Buffers
- Descriptors

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

Graphics Pipelines

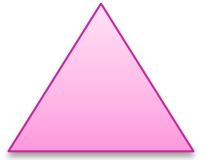
vkCmdDraw

Graphics Pipelines

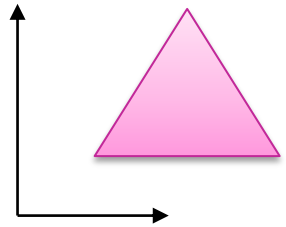


Graphics Pipelines

Graphics Pipelines

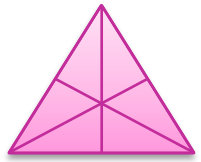


Graphics Pipelines



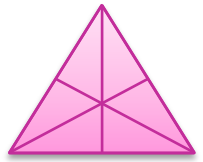
`vertex shader`

Graphics Pipelines



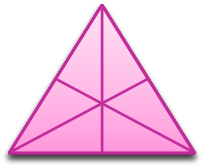
`vertex shader`

Graphics Pipelines



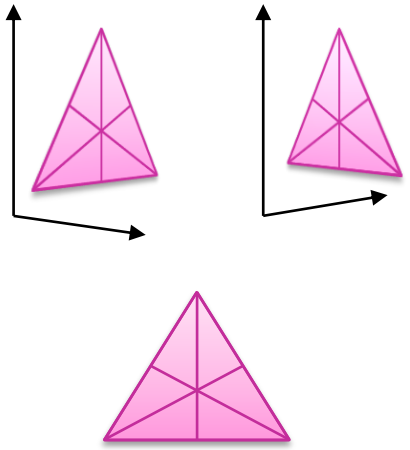
`vertex shader`

Graphics Pipelines



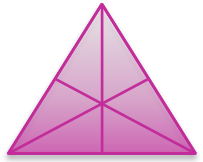
`vertex shader`

Graphics Pipelines



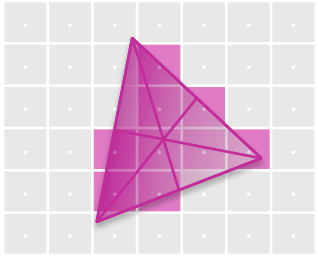
`vertex shader`

Graphics Pipelines



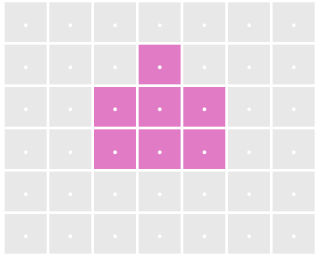
`vertex shader`

Graphics Pipelines



`vertex shader`

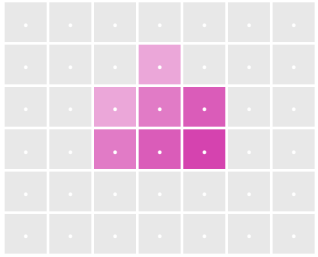
Graphics Pipelines



`vertex shader`

`fragment shader`

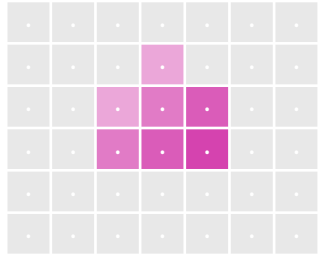
Graphics Pipelines



`vertex shader`

`fragment shader`

Graphics Pipelines



`vertex shader`

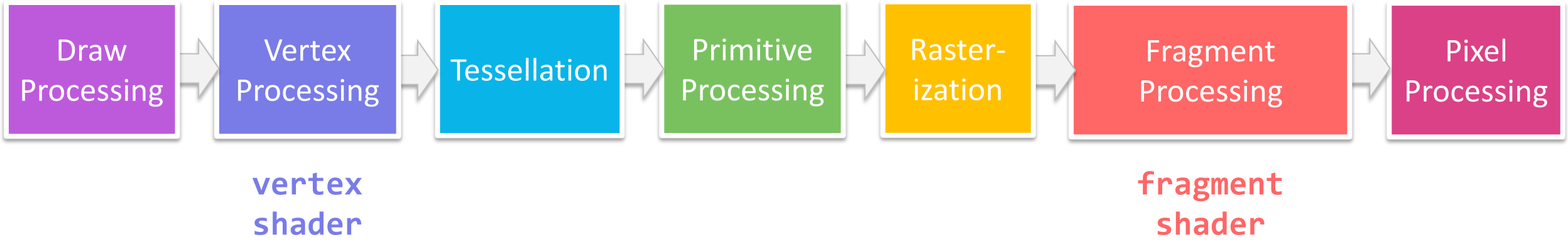
`fragment shader`

Graphics Pipelines

vkCmdDraw

=>

- Use a **graphics pipeline** (rasterizer!)
  - Which shaders to use? (e.g., **vertex** and **fragment**)
- Which geometry as **input** (e.g., triangles)
- Which **resources** to bind? (through descriptors)



PART 2

- **Graphics Pipelines**
- Uniform Buffers
- Descriptors

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

PART 2

- Graphics Pipelines
- **Uniform Buffers**
- Descriptors

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

GLSL vertex shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) in vec3 in_position;

void main() {
    gl_Position = uniform_buffer.transformationMatrix * vec4(in_position, 1.0);
}
```

vertex shader

fragment shader

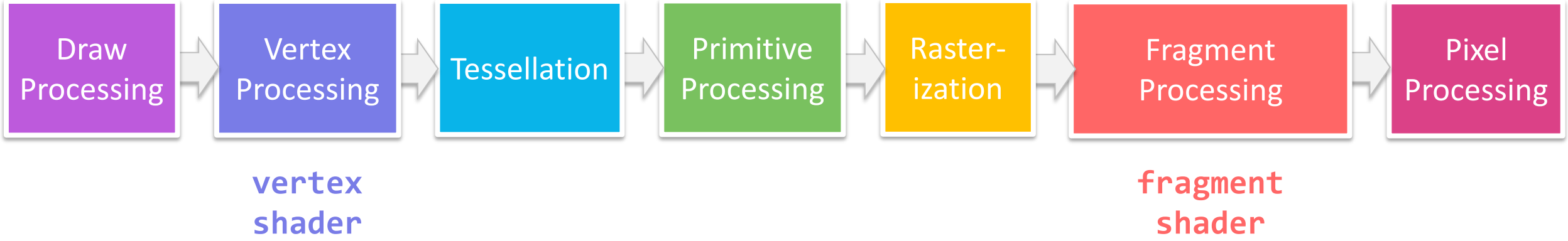
GLSL vertex shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) in vec3 in_position;

void main() {
    gl_Position = uniform_buffer.transformationMatrix * vec4(in_position, 1.0);
}
```



GLSL vertex shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) in vec3 in_position;

void main() {
    gl_Position = uniform_buffer.transformationMatrix * vec4(in_position, 1.0);
}
```

vertex shader

fragment shader

GLSL vertex shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) in vec3 in_position;

void main() {
    gl_Position = uniform_buffer.transformationMatrix * vec4(in_position, 1.0);
}
```

vertex shader

fragment shader

GLSL fragment shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) out vec4 out_color;

void main() {
    out_color = uniform_buffer.color;
}
```

vertex shader

fragment shader

GLSL fragment shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) out vec4 out_color;

void main() {
    out_color = uniform_buffer.color;
}
```

vertex shader

fragment shader

GLSL fragment shader

```
#version 450

layout (binding = 0) uniform UniformBuffer {
    vec4 color;
    mat4 transformationMatrix;
} uniform_buffer;

layout (location = 0) out vec4 out_color;

void main() {
    out_color = uniform_buffer.color;
}
```

vertex shader

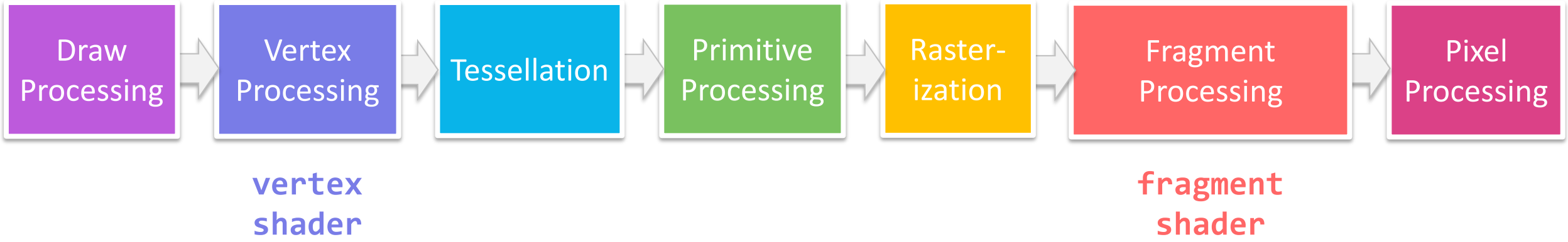
fragment shader

Buffers

vkCmdDraw

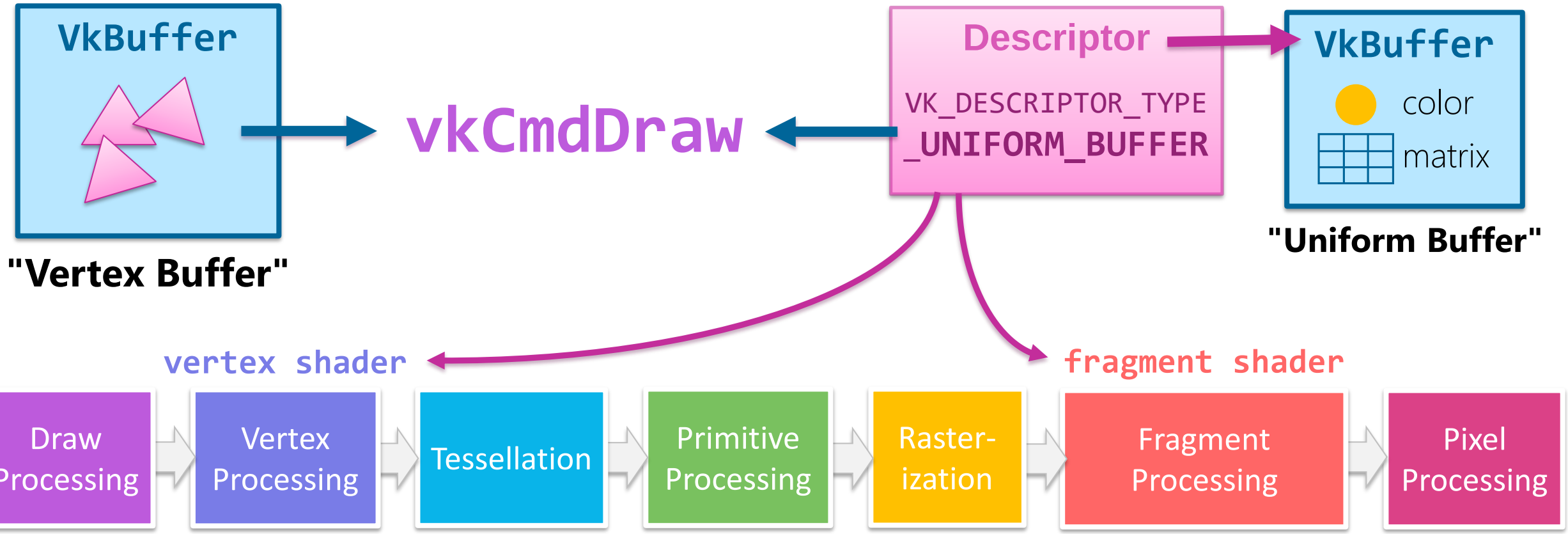
=>

- Use a **graphics pipeline** (rasterizer!)
  - Which shaders to use? (e.g., **vertex** and **fragment**)
- Which geometry as **input** (e.g., triangles)
- Which **resources** to bind? (through descriptors)



Buffers

- Which geometry as **input** (e.g., triangles)
- Which **resources** to bind? (through descriptors)



PART 2

- Graphics Pipelines
- **Uniform Buffers**
- Descriptors

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

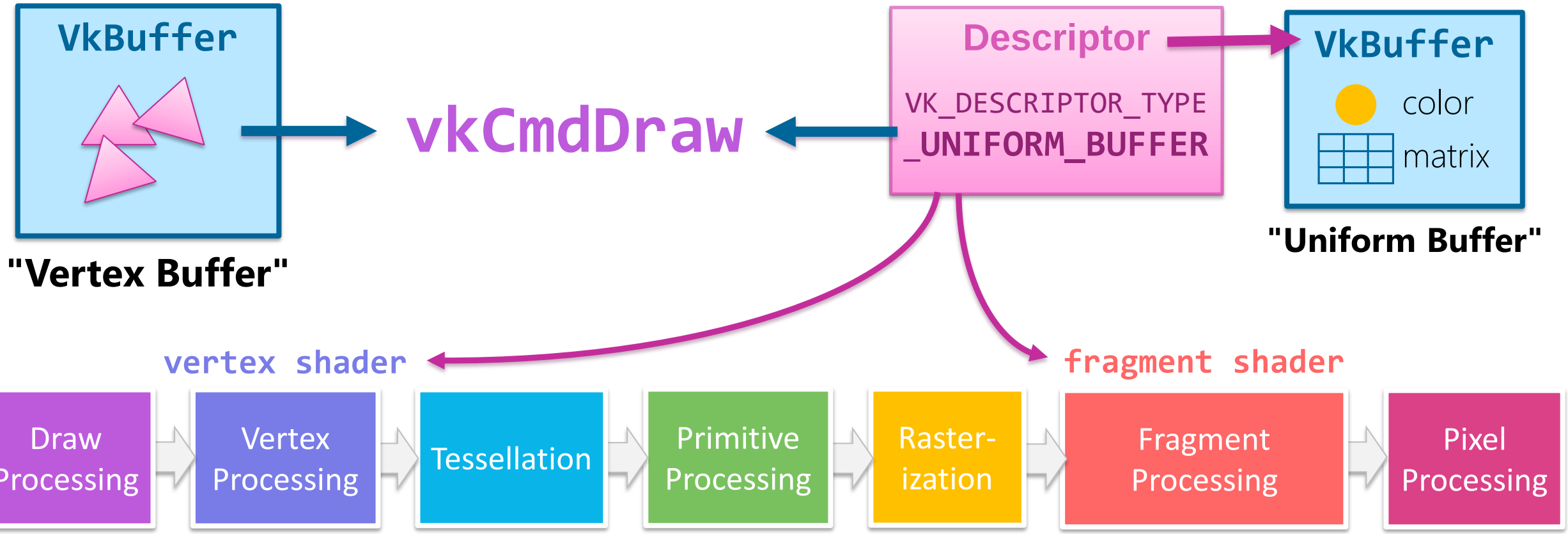
PART 2

- Graphics Pipelines
- Uniform Buffers
- **Descriptors**

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

Buffers

- Which geometry as **input** (e.g., triangles)
- Which **resources** to bind? (through descriptors)



Descriptors and Descriptor Sets

- One **descriptor** describes one **resource**

Descriptors and Descriptor Sets

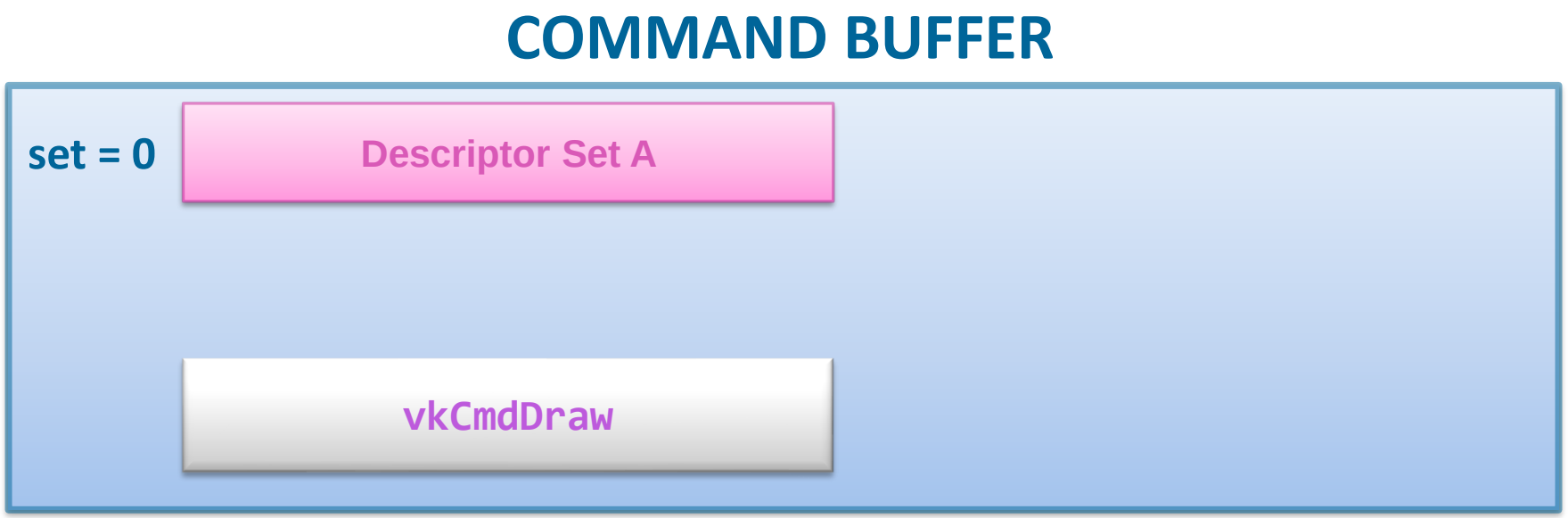
- One **descriptor** describes one **resource**
- Descriptors are organized in **descriptor sets**

Descriptors and Descriptor Sets

- One **descriptor** describes one **resource**
- Descriptors are organized in **descriptor sets**

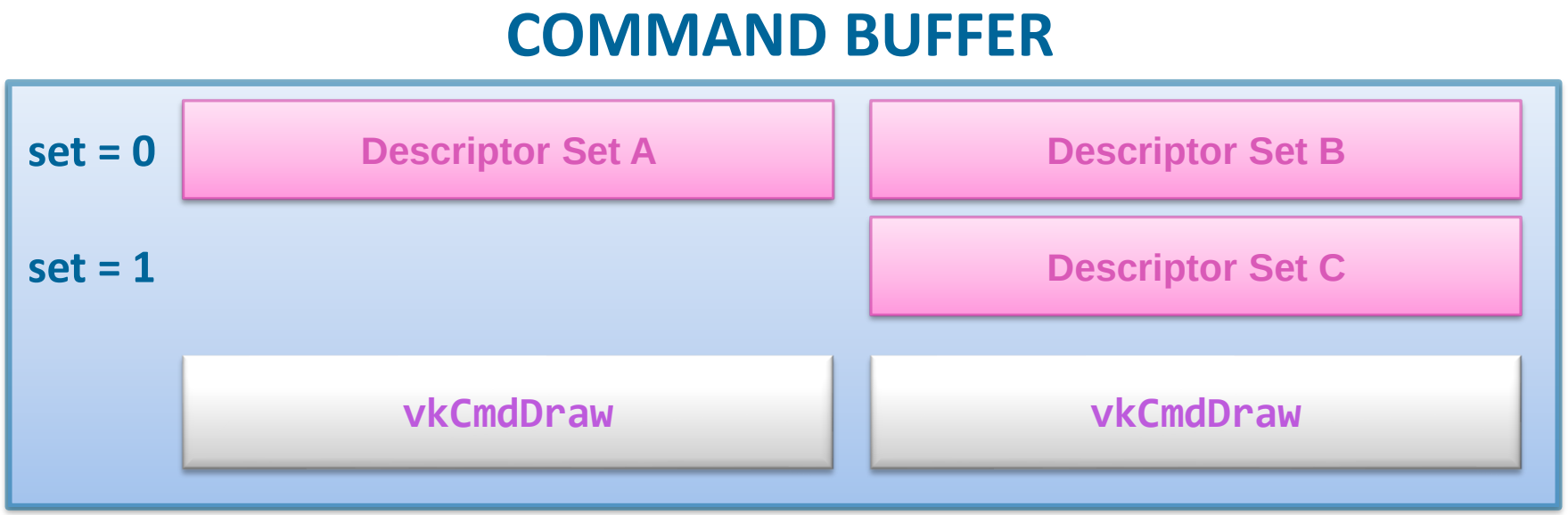
Descriptors and Descriptor Sets

- One **descriptor** describes one **resource**
- Descriptors are organized in **descriptor sets**
- Descriptor set bound while recording a command buffer

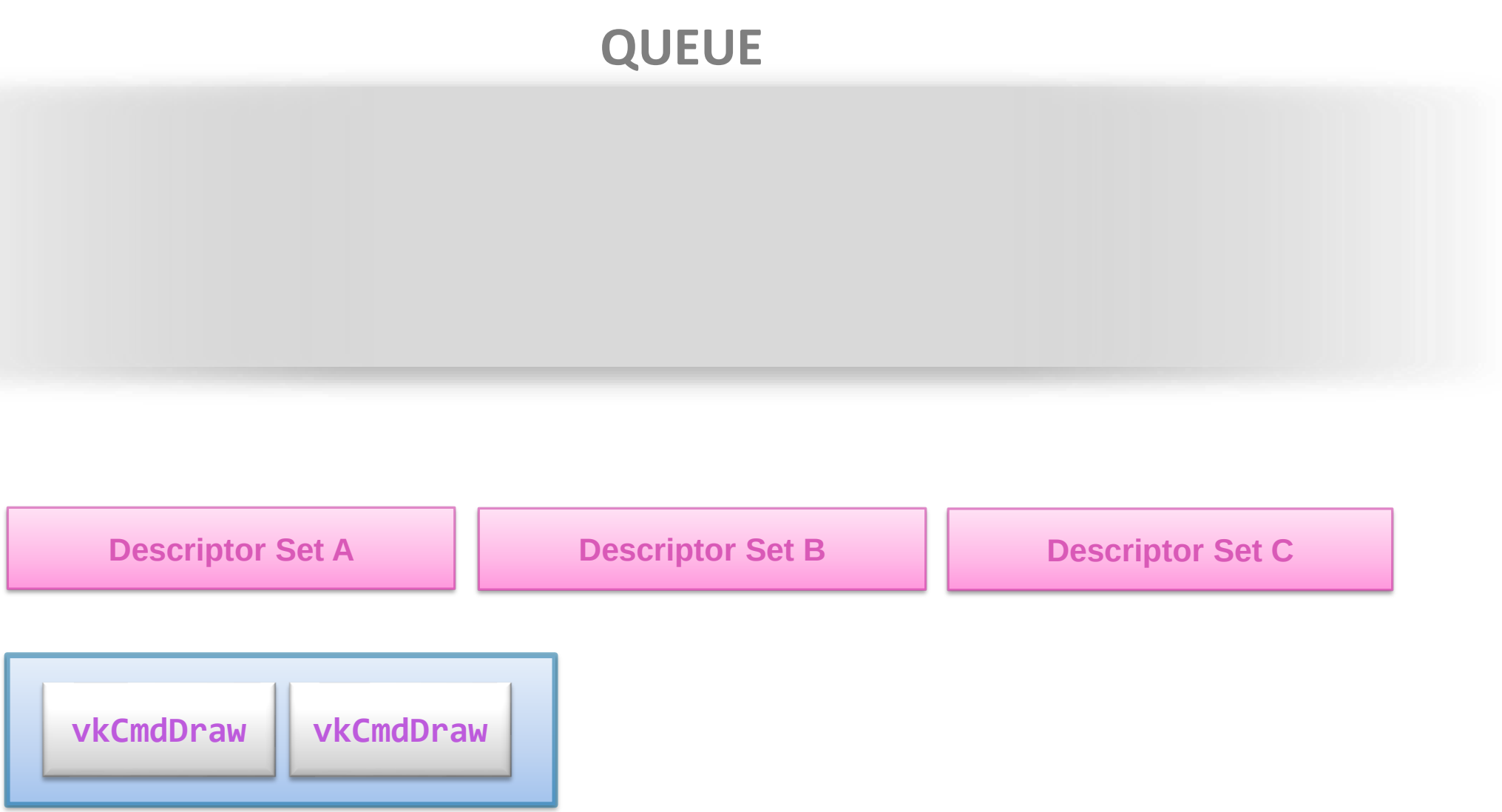


Descriptors and Descriptor Sets

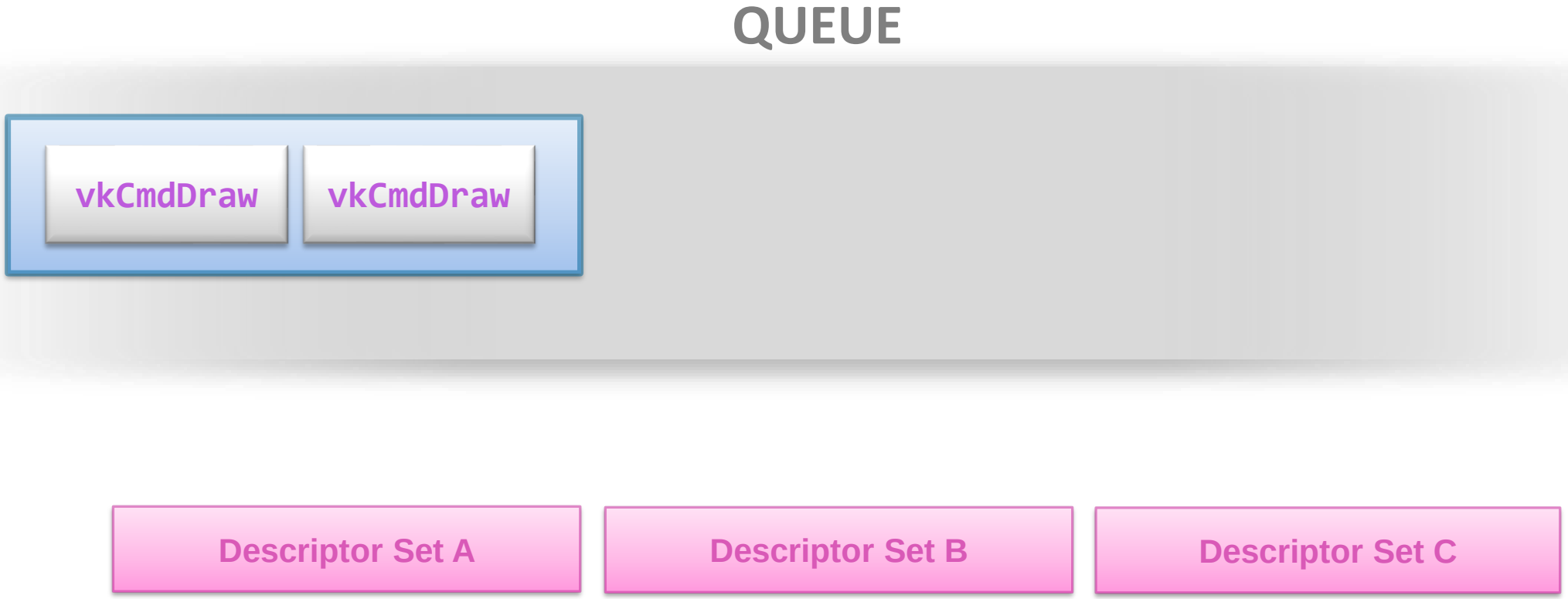
- One **descriptor** describes one **resource**
- Descriptors are organized in **descriptor sets**
- Descriptor set bound while recording a command buffer



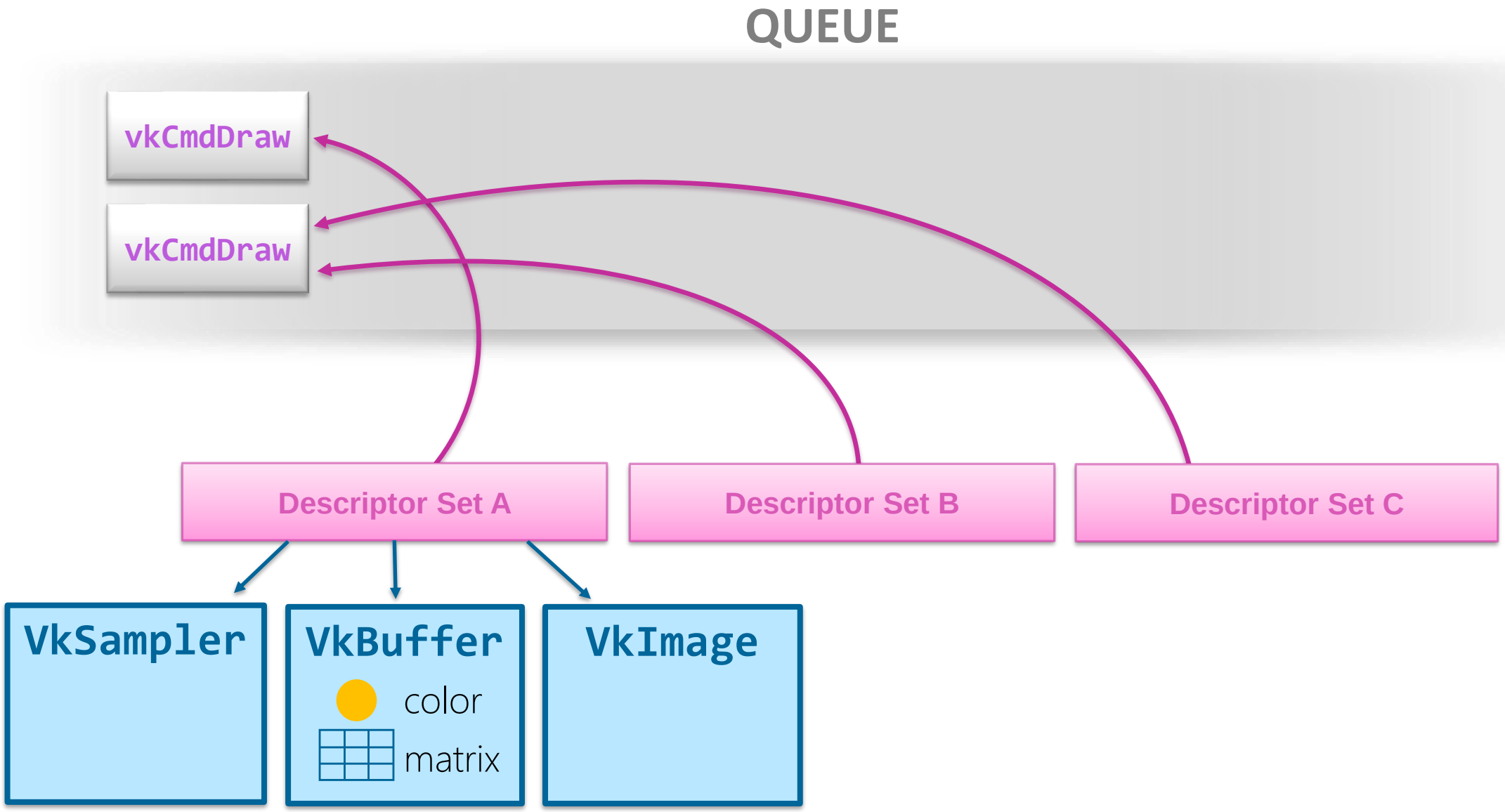
Descriptors and Descriptor Sets



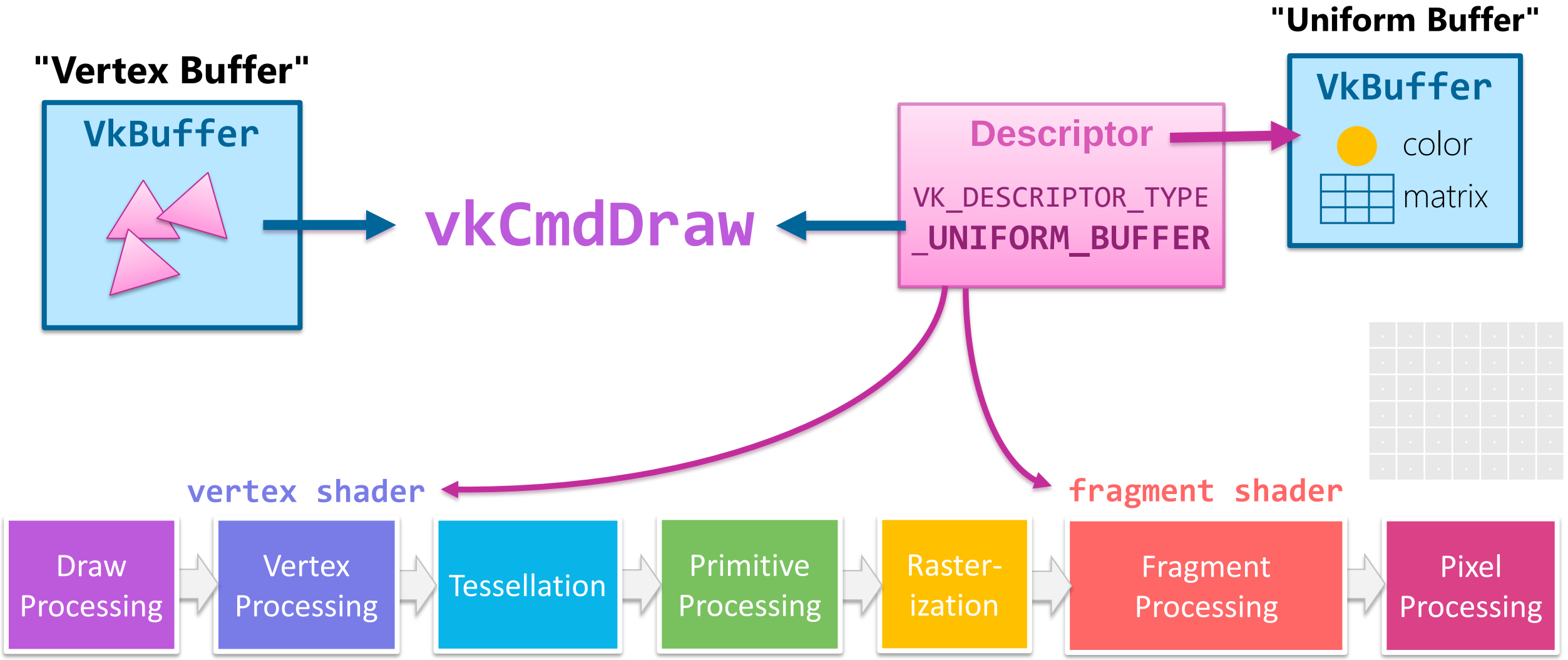
Descriptors and Descriptor Sets



Descriptors and Descriptor Sets



Full Picture



PART 2

- Graphics Pipelines
- Uniform Buffers
- **Descriptors**

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

PART 2

- Graphics Pipelines
- Uniform Buffers
- Descriptors

Platinum Sponsors: AMD arm Google LUNARG KHRONOS GROUP SAMSUNG

Schedule

PART 1:

Setup	Lecture	Coding Session
10 min	**20** min	**90** min
Starts at 09:00	Starts at 09:10	Starts at 09:30

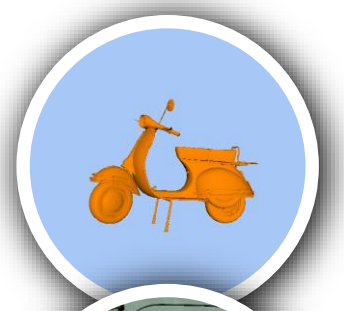
PART 2:

Lecture	Coffee Break	Coding Session
15 min	**25** min	**80** min
Starts at 11:00	Starts at 11:15	Starts at 11:40

Lunch Break 13:00 – 14:00

PART 3:

Lecture	Coding Session	Coffee Break
15 min	**65** min	**30** min
Starts at 14:00	Starts at 14:15	Starts at 15:20



PART 4:

Lecture	Coding Session	Closing
20 min	**70** min	**10** min
Starts at 15:50	Starts at 16:10	Starts at 17:20