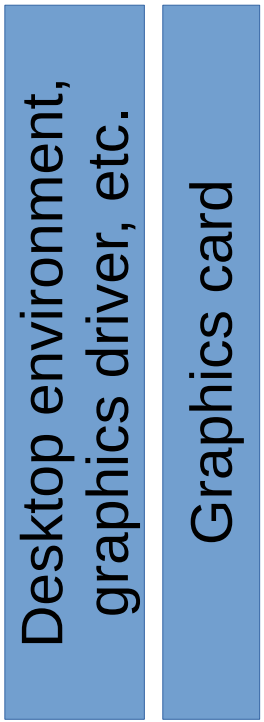
What is a graphics engine?

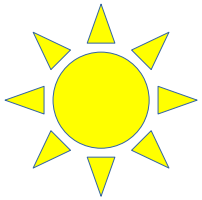
(very generally speaking)

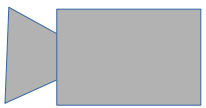
What is a graphics engine?



Let’s say that we do have a computer with a 3D graphics card and a desktop environment. You know, a computer.

What is a graphics engine?



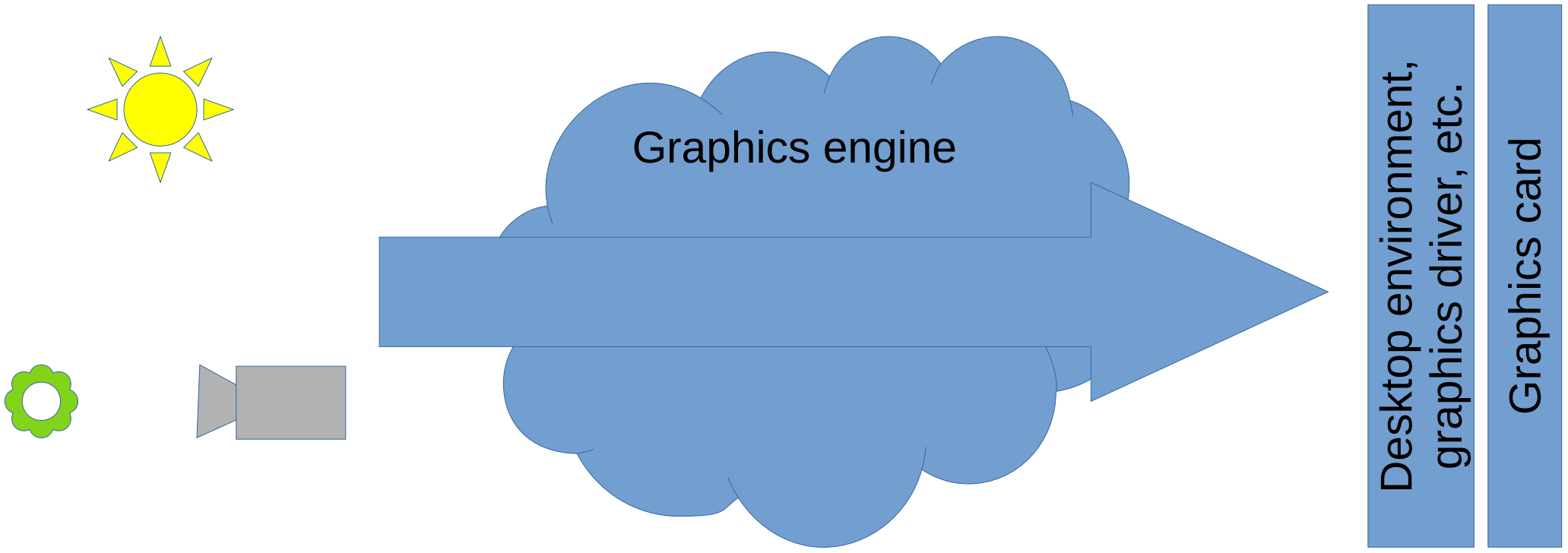


Desktop environment, graphics driver, etc.

Graphics card

And we have a bunch of digital art assets.

What is a graphics engine?

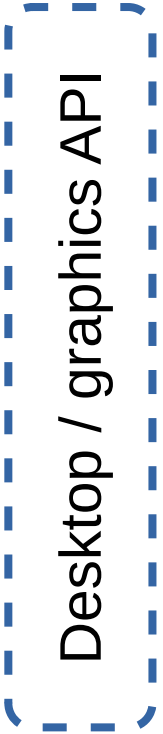


A graphics engine is a heap of software that makes the computer turn scenes filled with assets into images on the screen.

And in Panda3D?

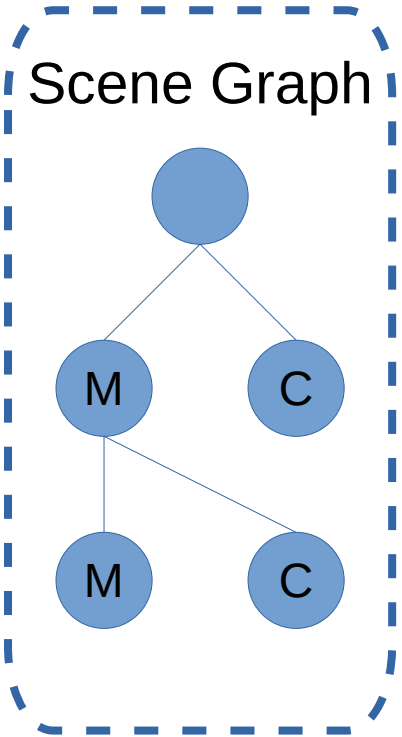
(specifically speaking)

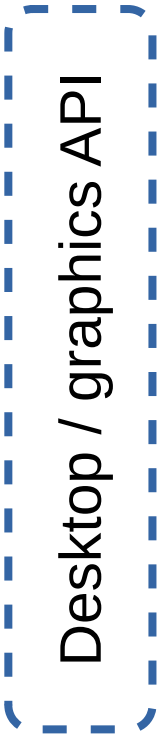
And in Panda3D?



Panda3D runs on several operating systems with different graphics APIs, and we will not care about the specifics of those right now. Or, ideally, ever.

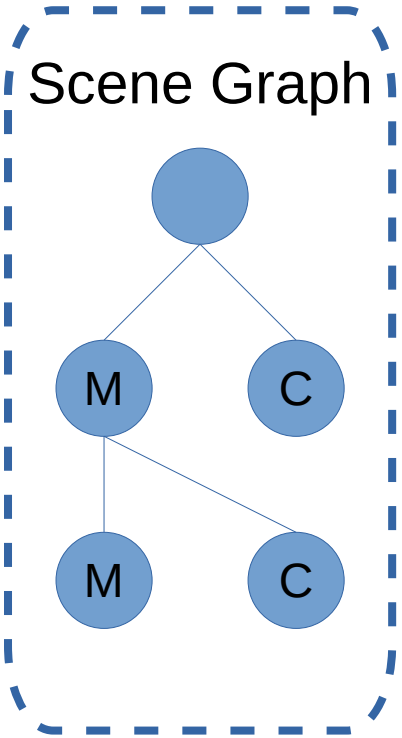
And in Panda3D?

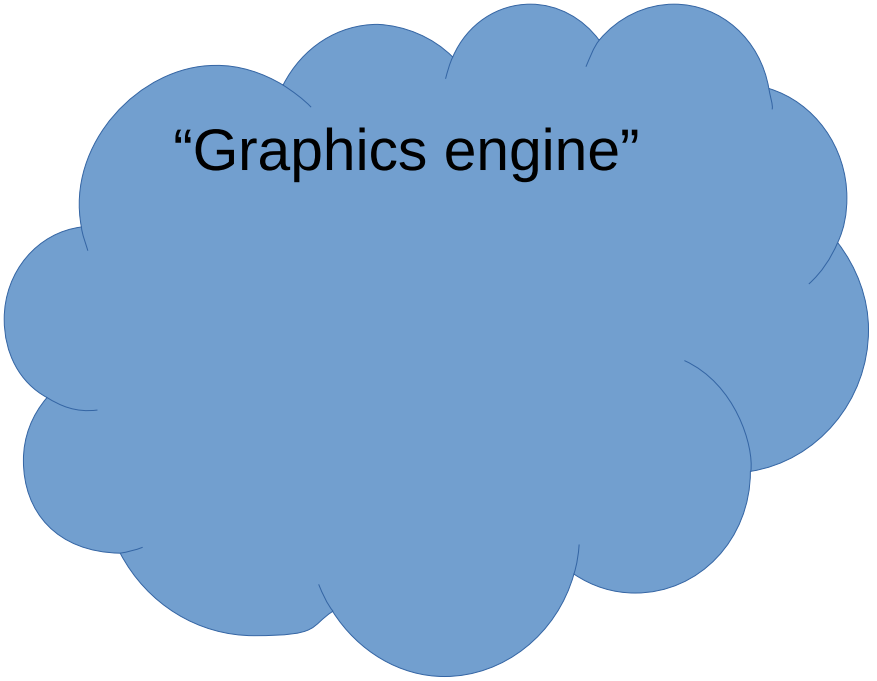


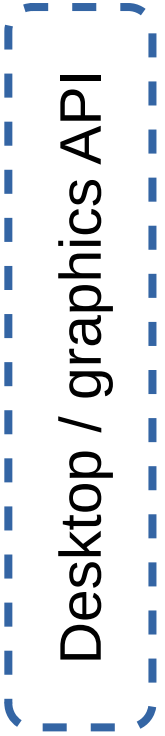


Scene graphs arrange **M**odels and **C**ameras (and other things) into scenes.

And in Panda3D?

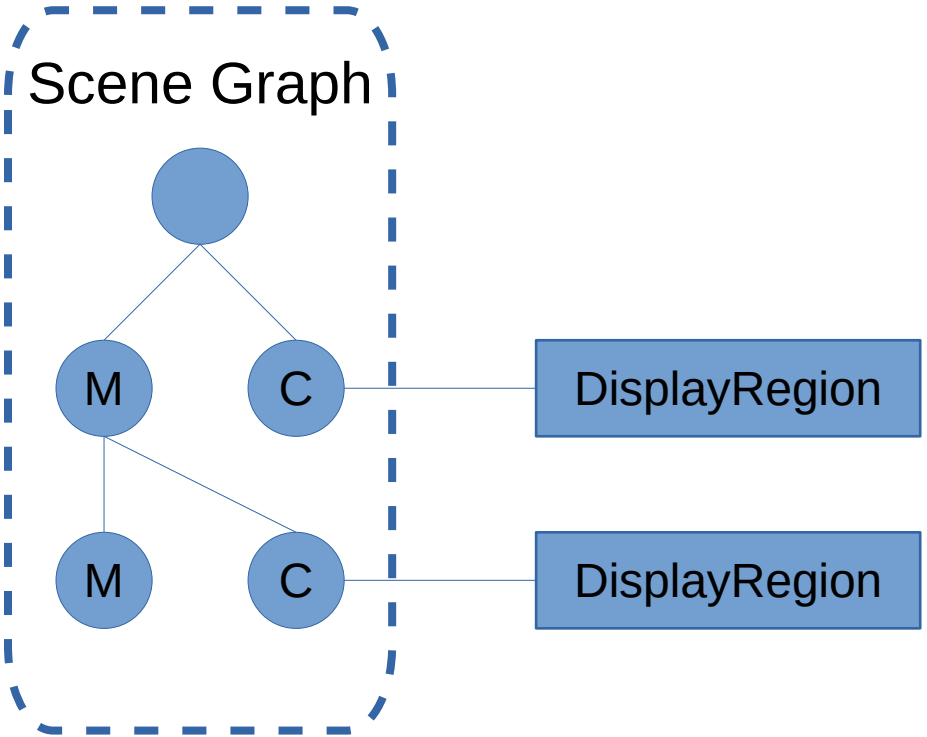


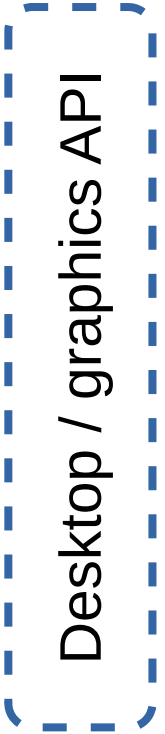




Here, we will be dealing with the technology between those ends.

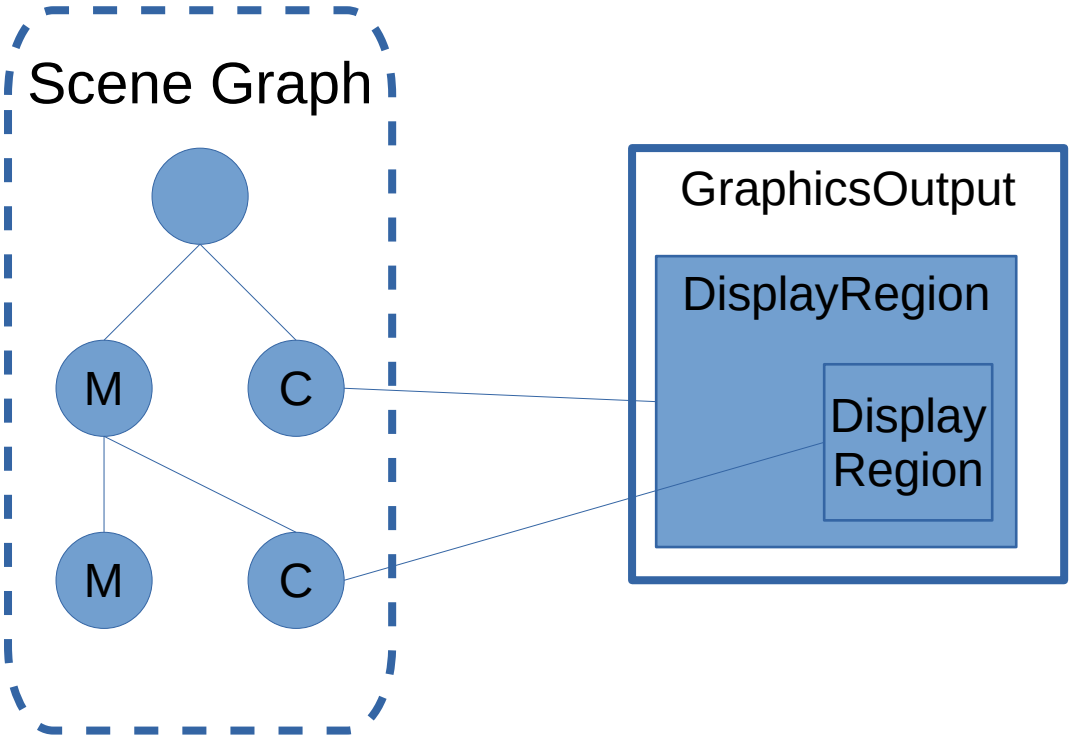
Between the ends





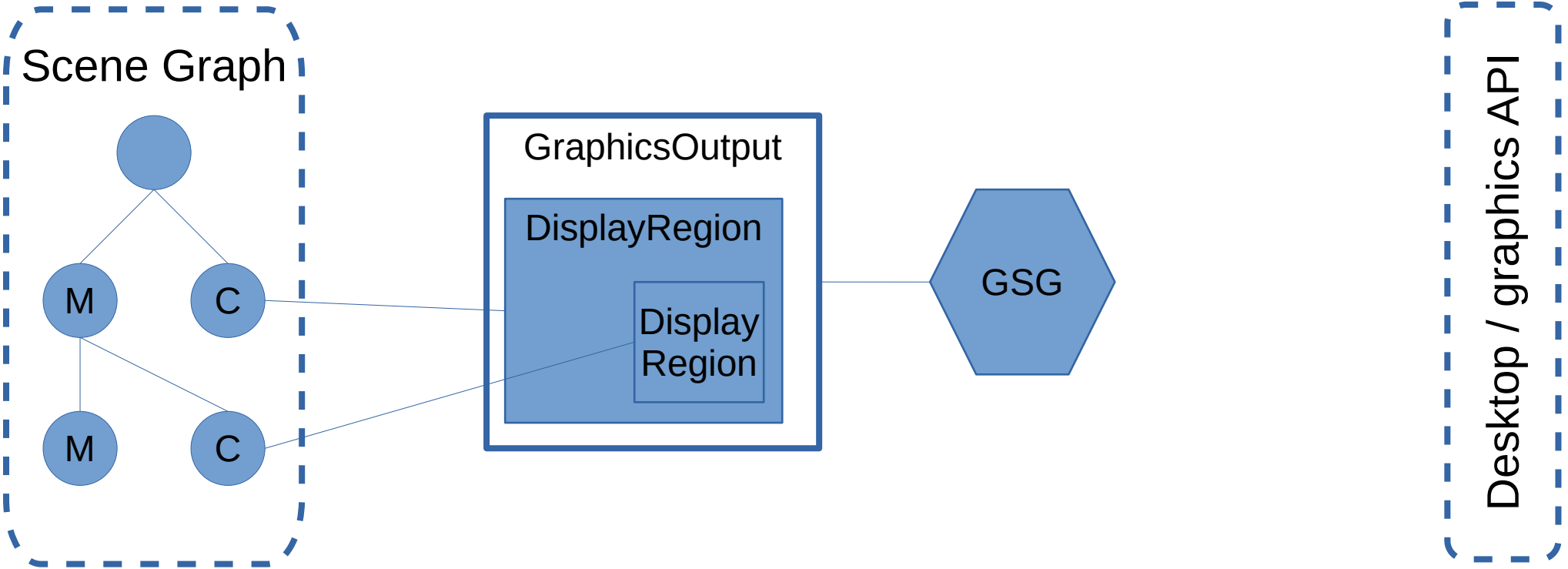
A camera’s view (if it is relevant to the final output) is rendered into a DisplayRegion.
Several DisplayRegions may use the same camera.

Between the ends



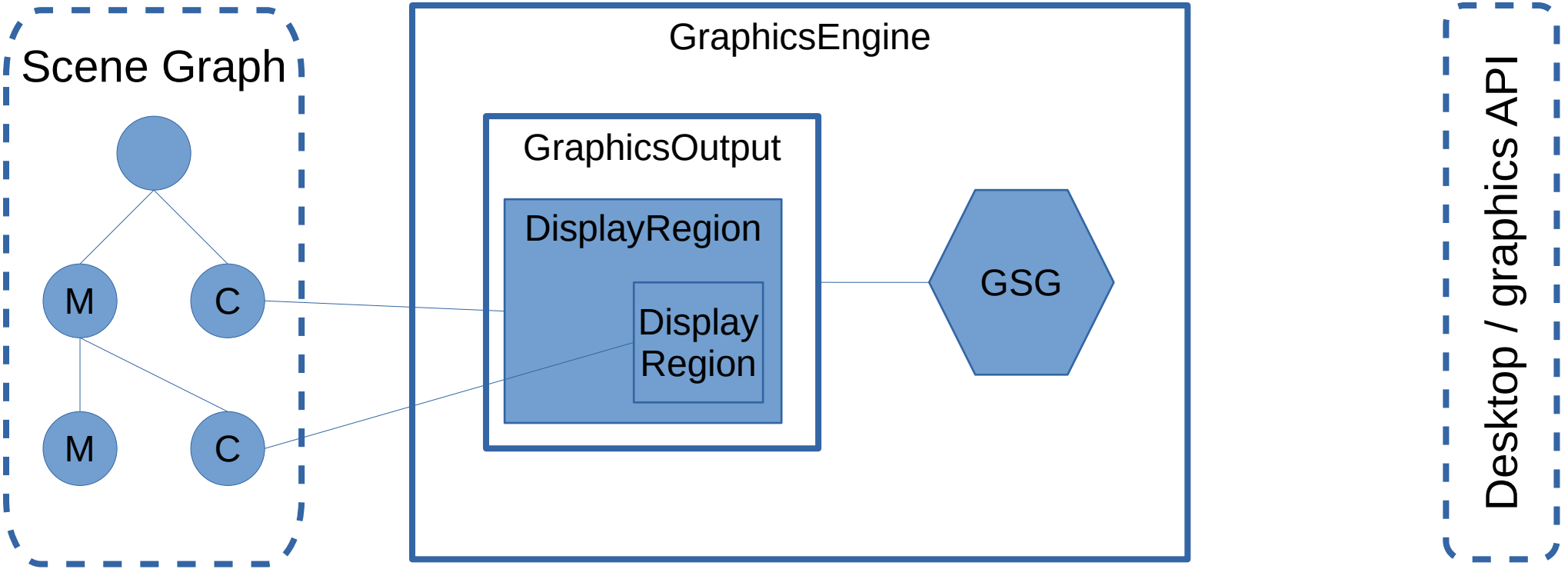
DisplayRegions occupy an area in a GraphicsOutput.
Windows are a type of GraphicsOutput.

Between the ends



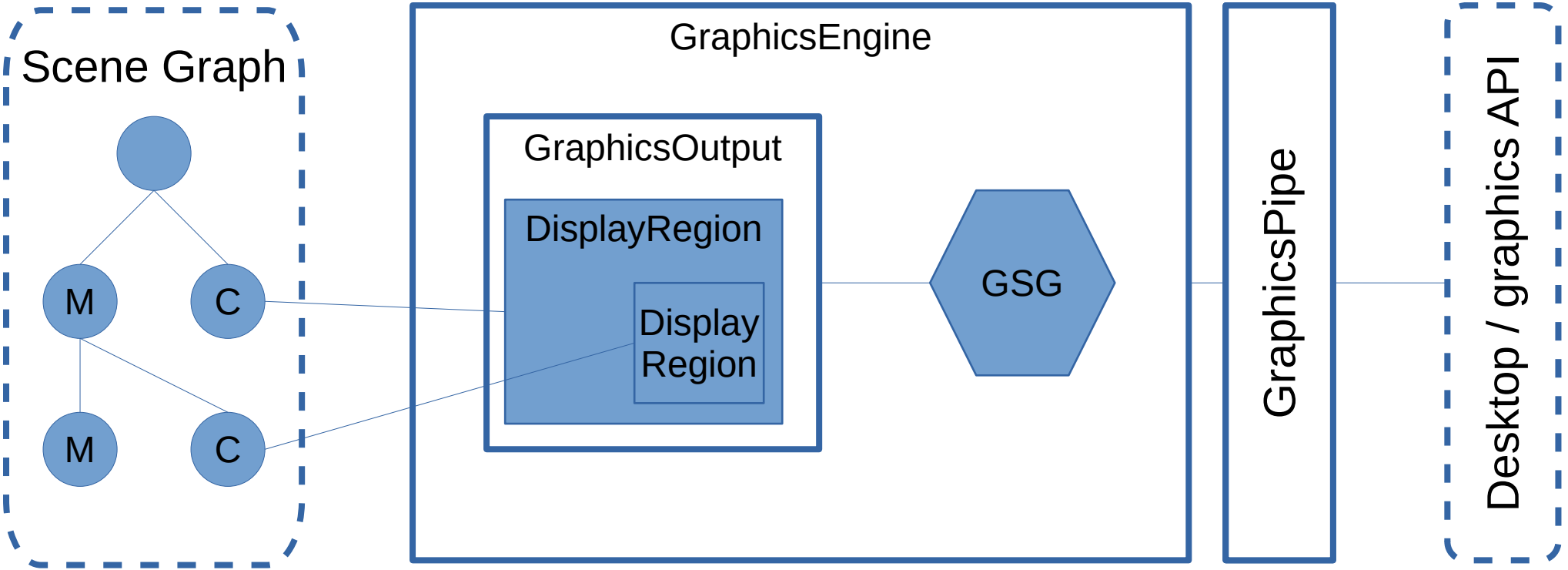
Each GraphicsOutput uses a GraphicsStateGuardian to render its content in a safe manner.
Several GraphicsOutputs may share the same GraphicsStateGuardian, improving efficiency.

Between the ends



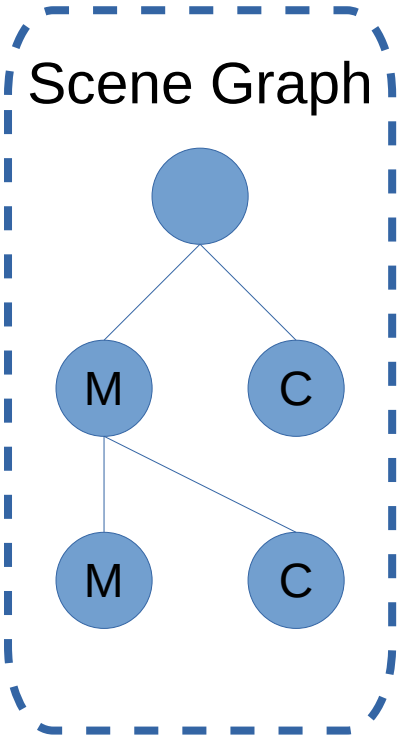
These objects are all created by a GraphicsEngine, which also manages the rendering.

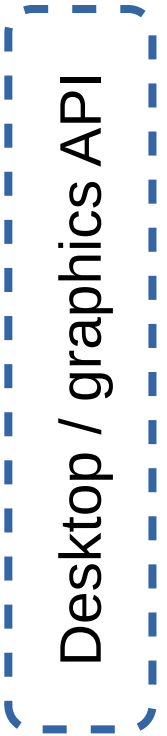
Between the ends



Each GraphicsEngine is bound to a GraphicsPipe, which abstracts the computer's particulars away.

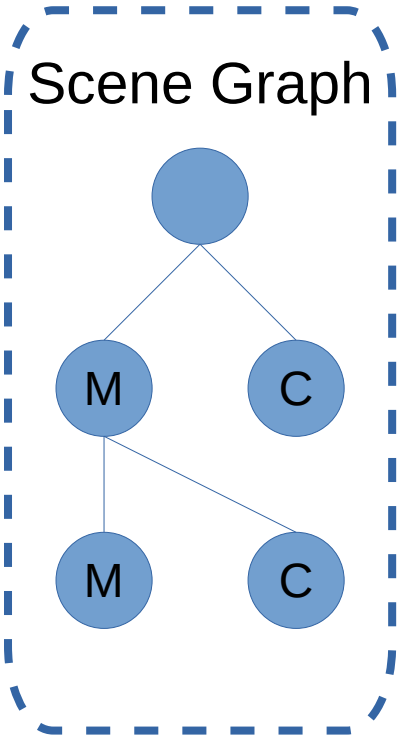
From scratch

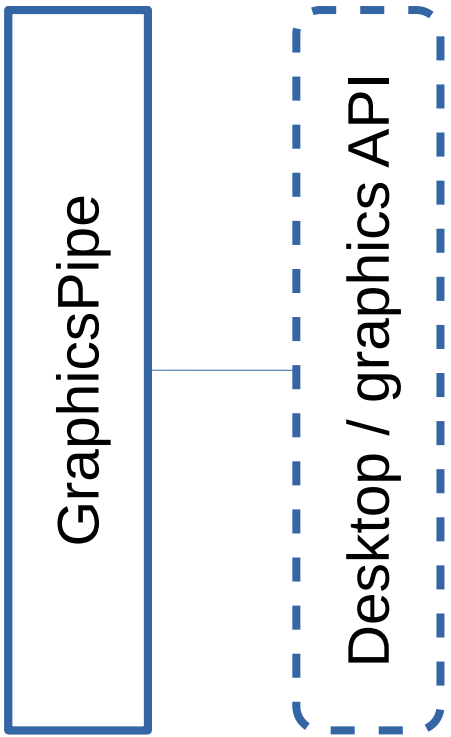




Starting with a blank slate, let us create all this in code.

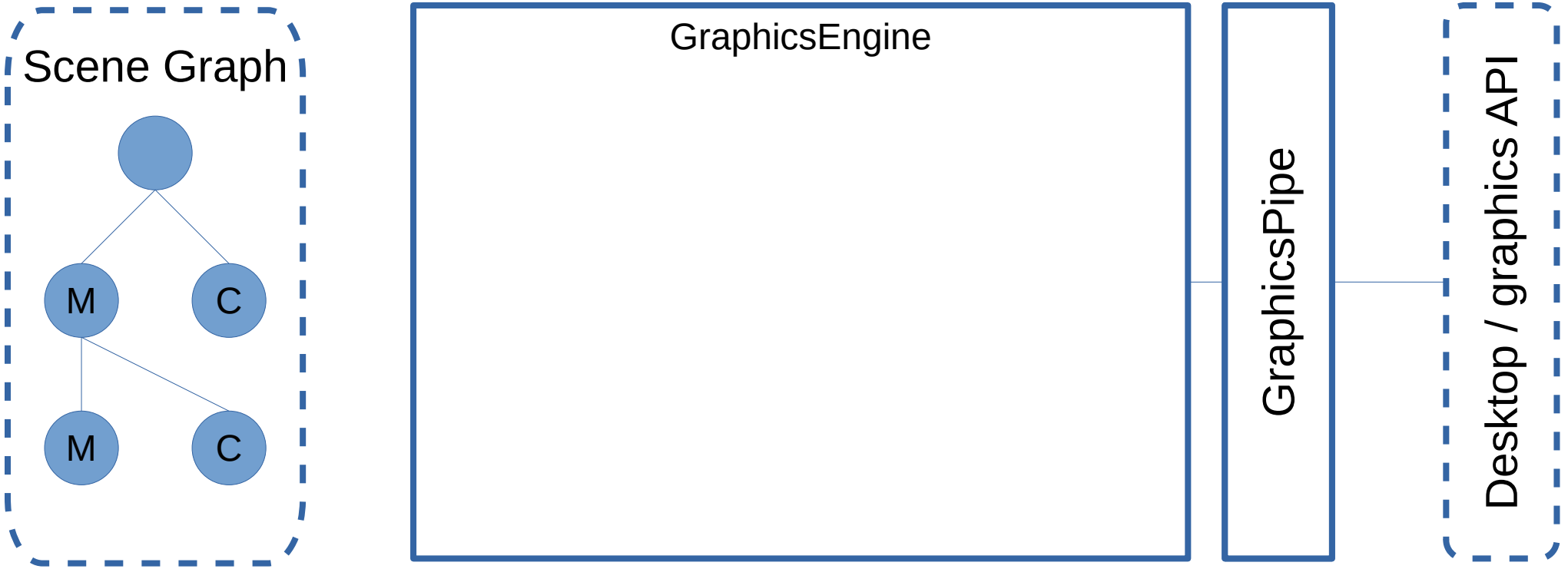
From scratch





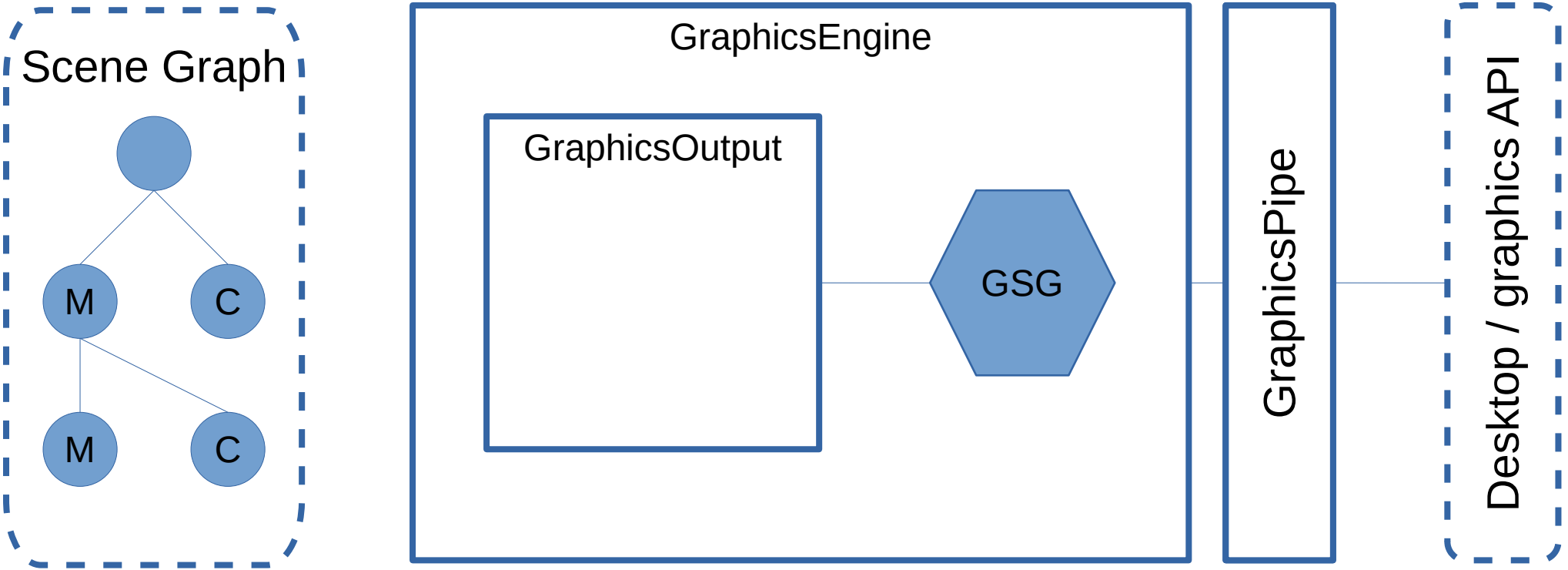

```
selection = GraphicsPipeSelection.getGlobalPtr()
pipe = selection.make_default_pipe()
```

From scratch



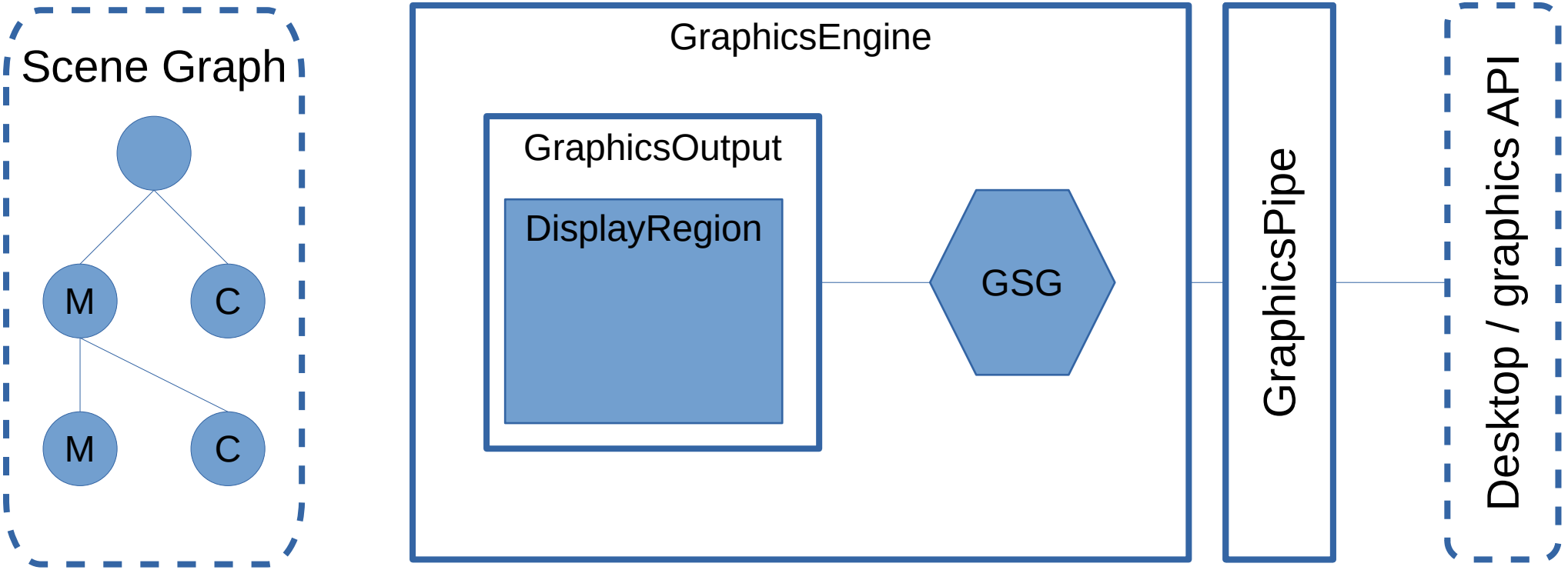

```
graphics_engine = GraphicsEngine(pipe)
```

From scratch



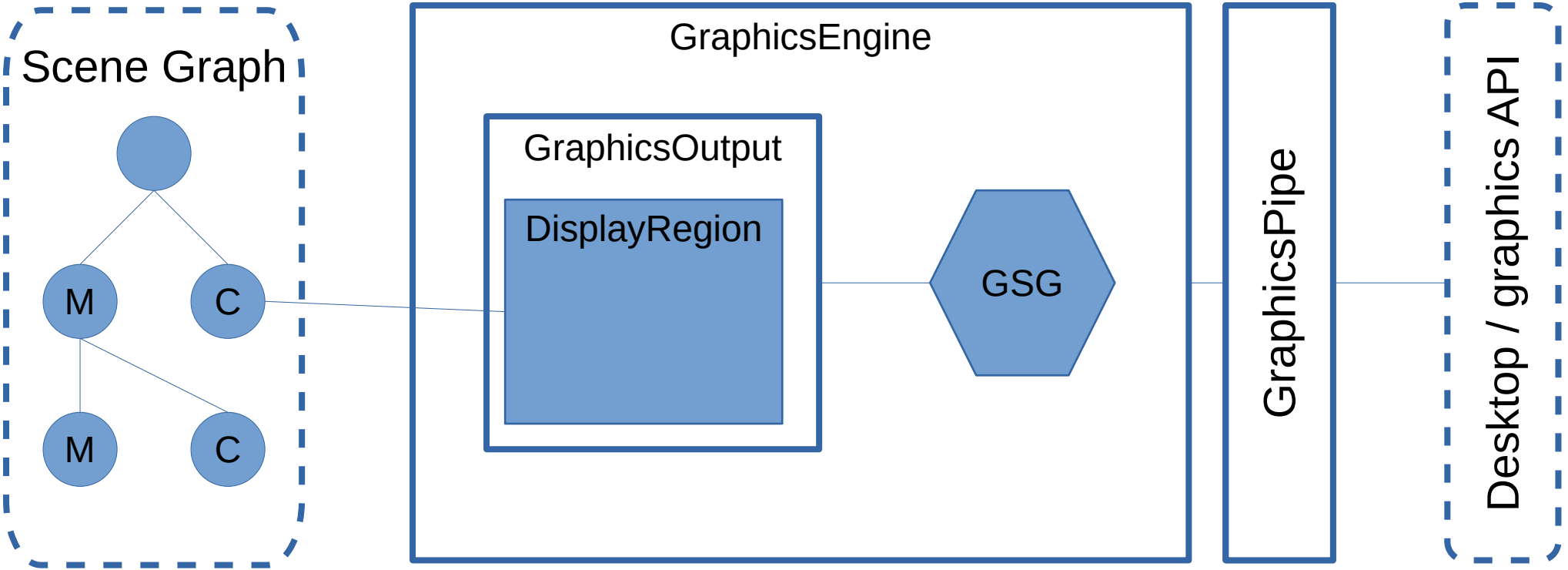

```
window = graphics_engine.make_output(pipe, …)
# Implicitly binds to the default GSG
```

From scratch



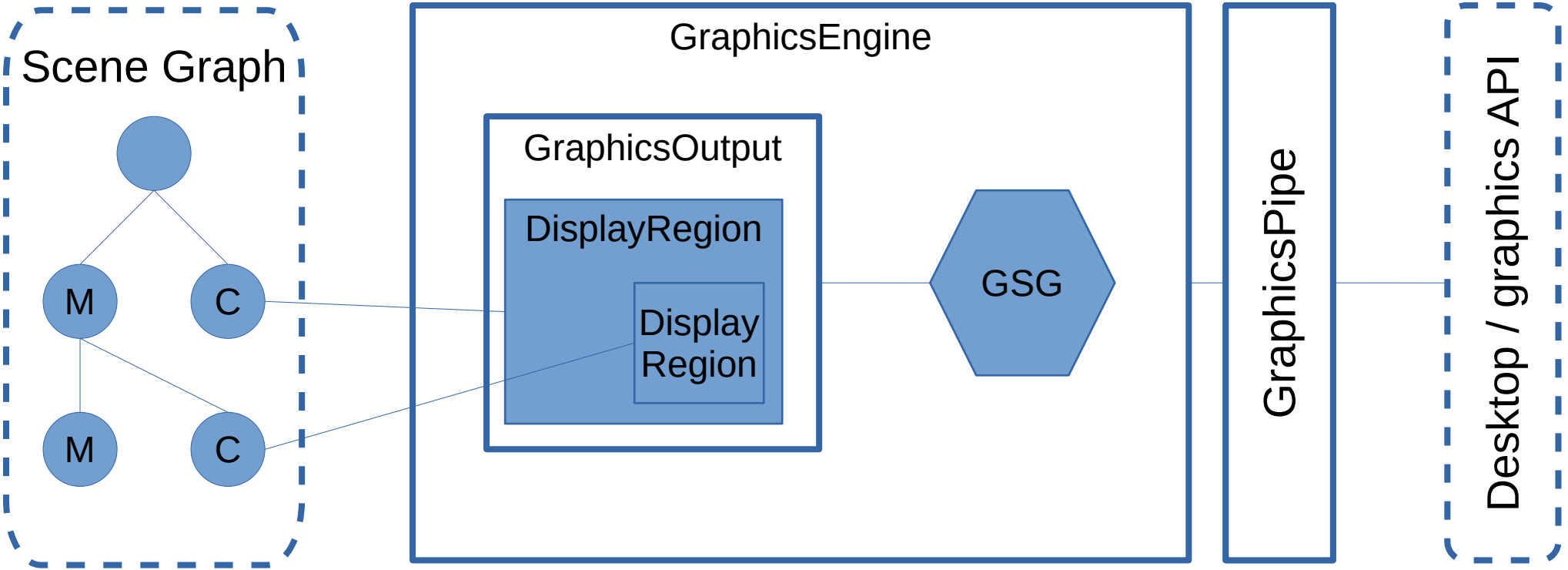

```
display_region = window.make_display_region()
```

From scratch




```
display_region.set_camera(cam_node_path)
```

From scratch

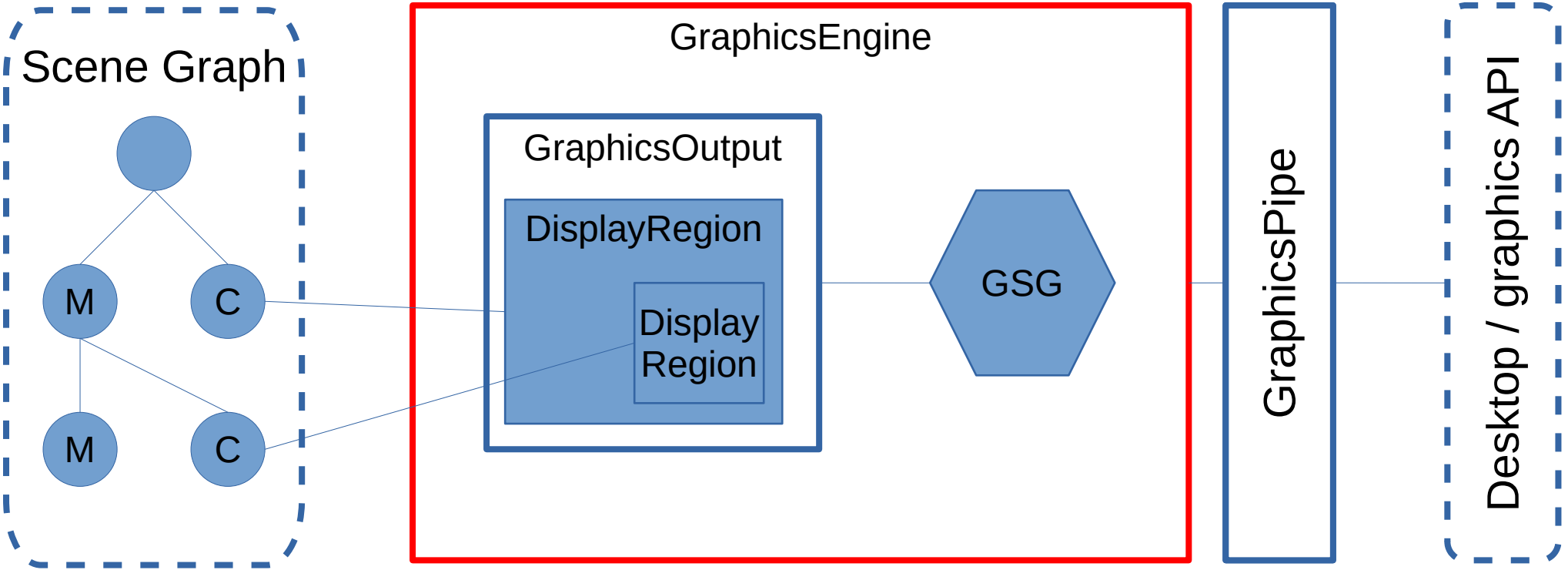



```
# The second DisplayRegion works the same.
```

Rendering

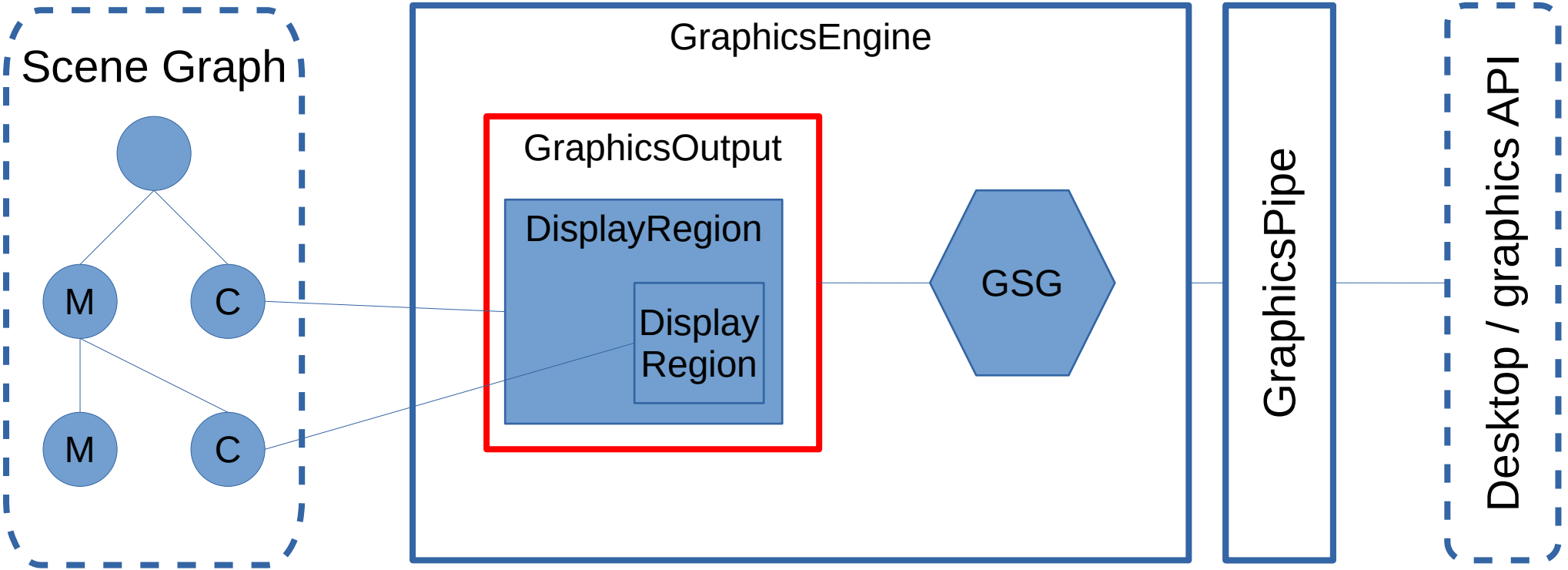
(The process this side of the graphics driver)

Rendering



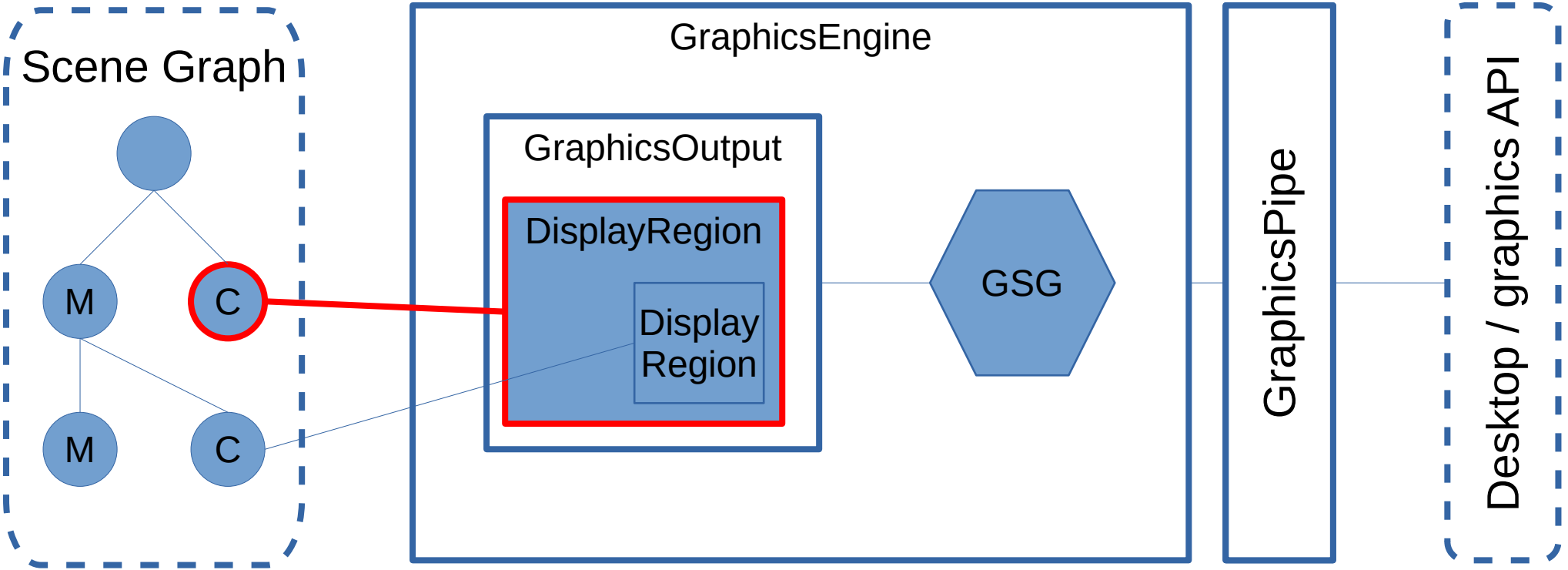
We kick off the process: `graphics_engine.render_frame()`

Rendering



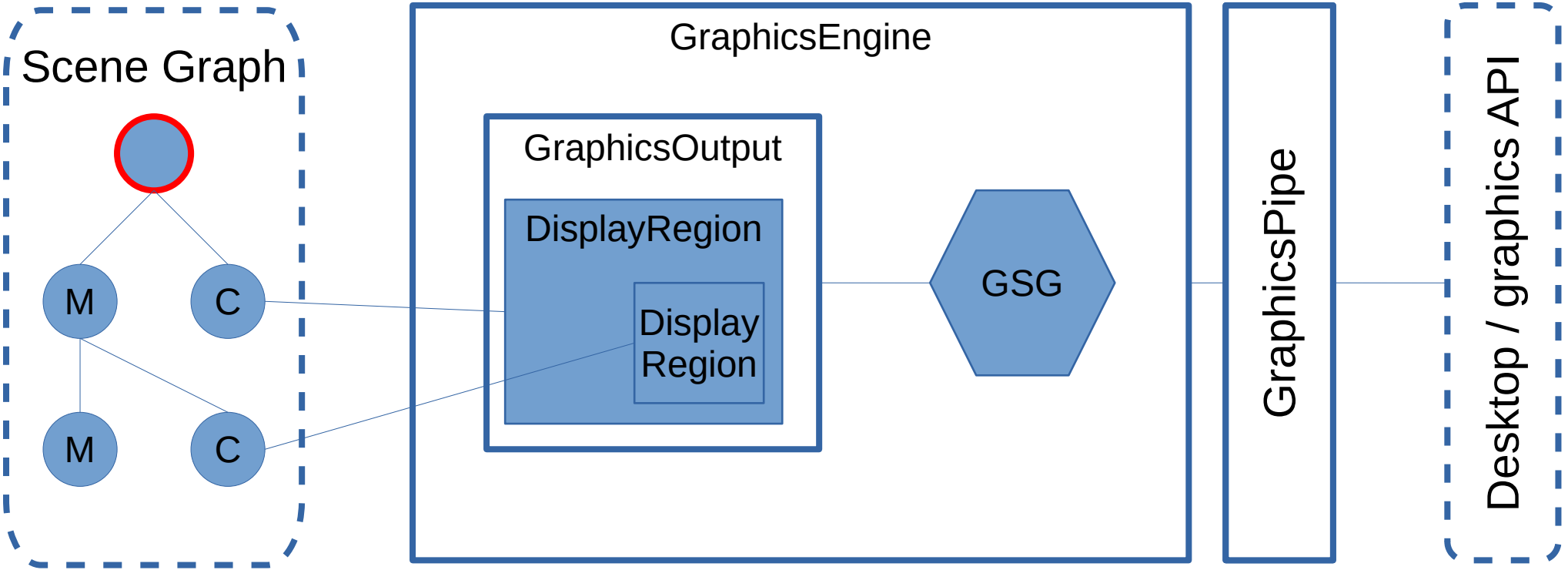
The engine goes through each of its GraphicsOutputs.

Rendering



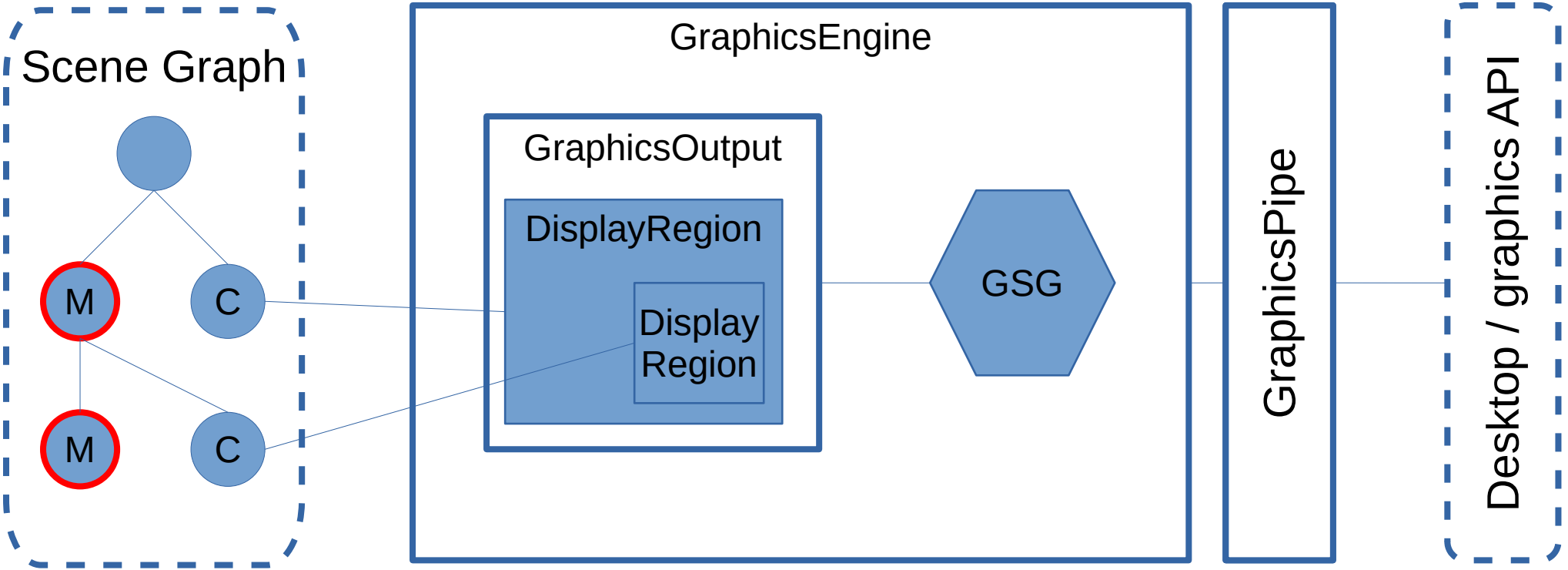
There, it goes through each DisplayRegion (and its camera).

Rendering



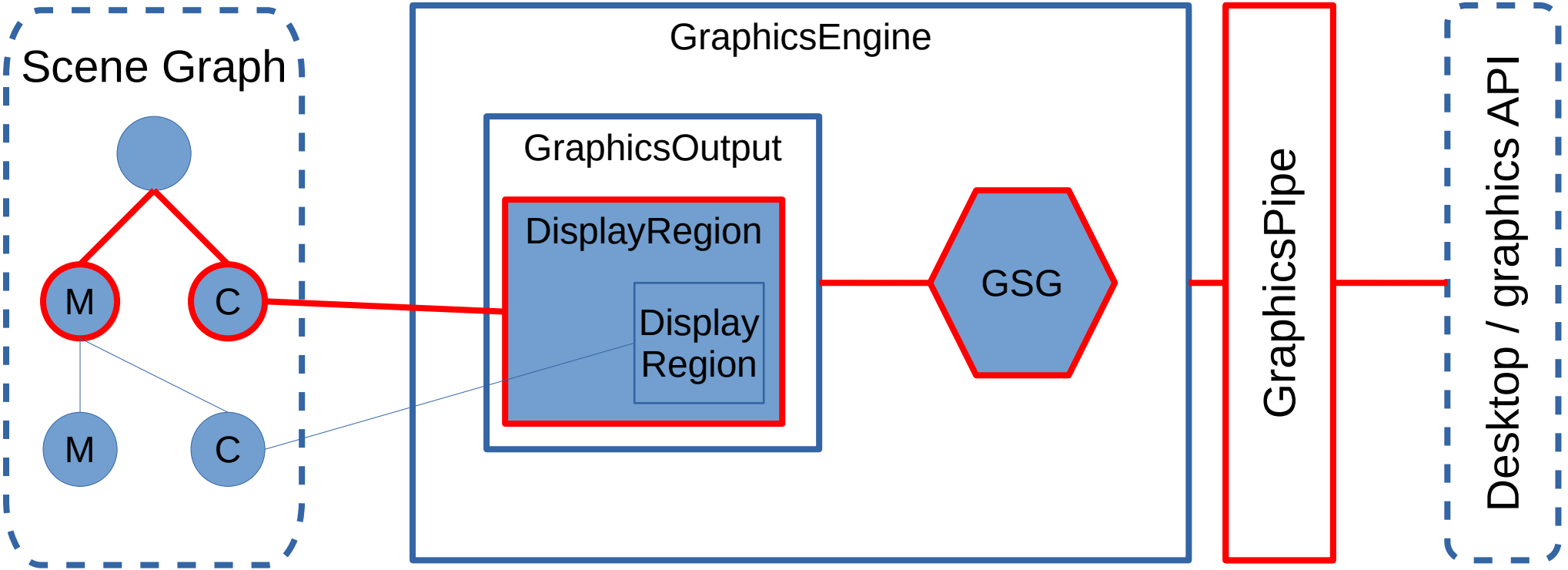
The root of the scene graph to which the camera is attached is determined.

Rendering



Culling is performed to determine which objects are (able to be) visible in the scene.

Rendering



Draw commands are issued for each object.
Once all commands are processed, the image is ready.