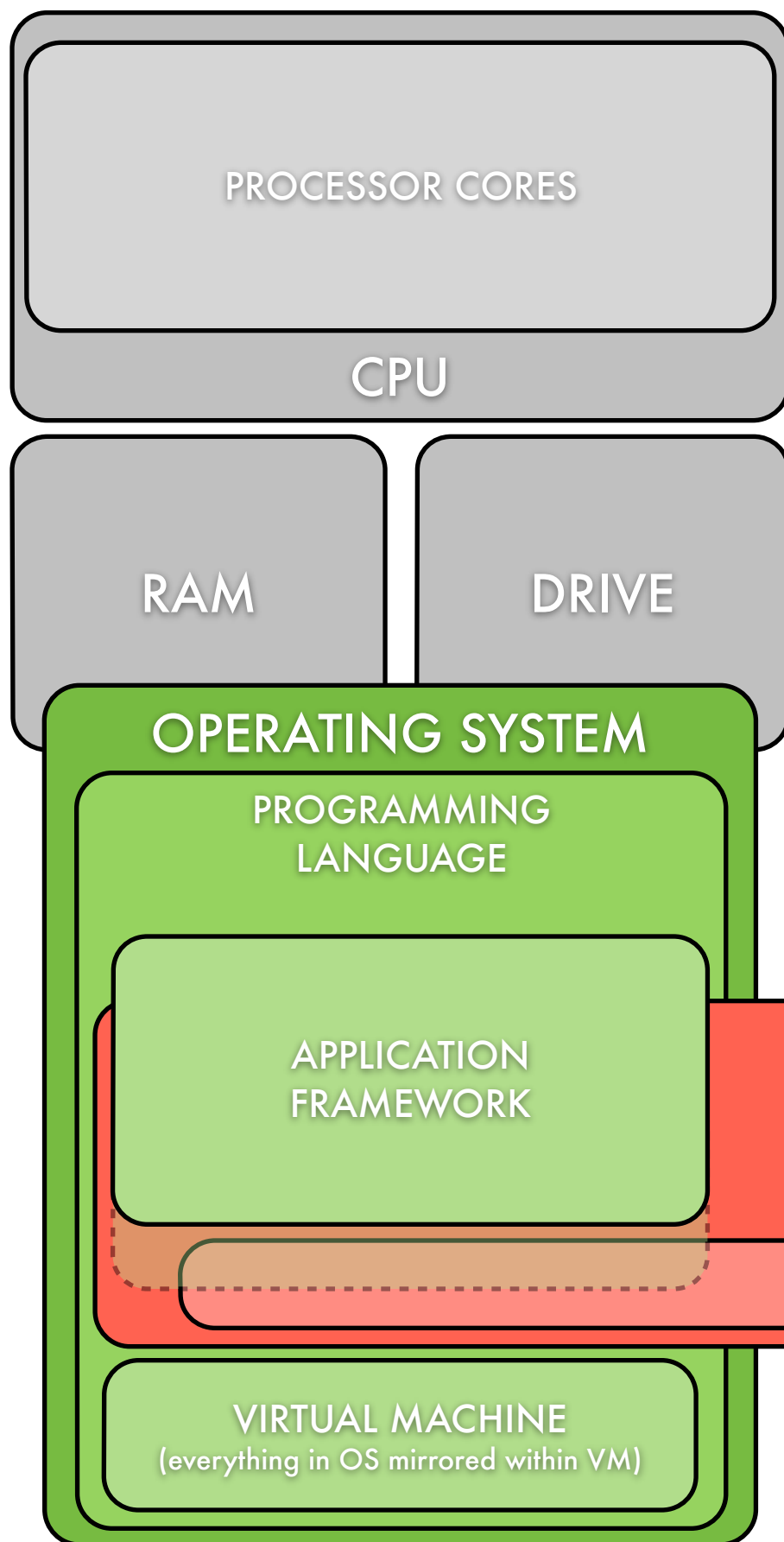


The Art of Graphics Programming

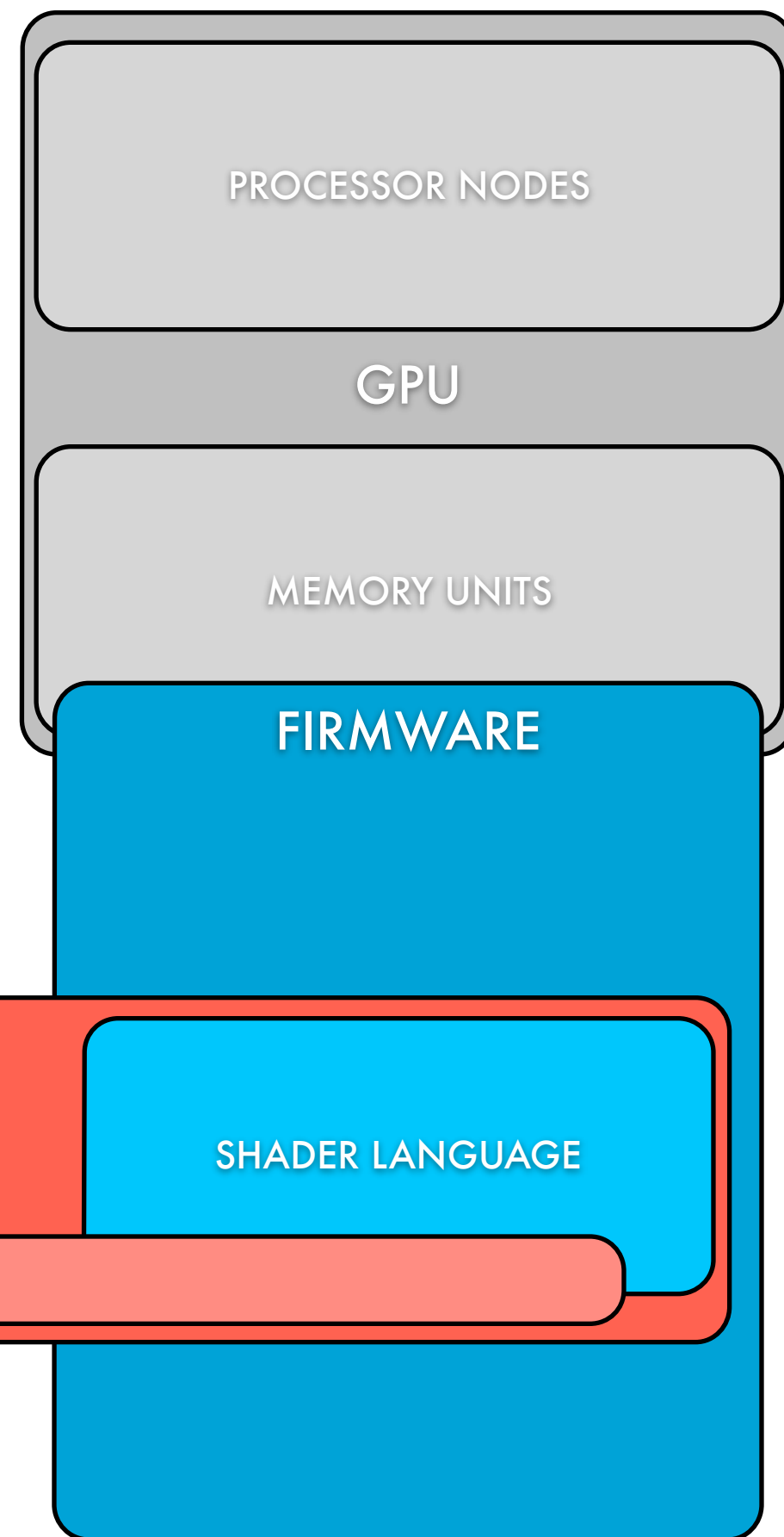
Technical Context

Hardware and Software
as an Integrated System

Patrick Hebron
NYU ITP



A Rough Sketch of How It All Fits Together



GRAPHICS API

GPGPU API

For optimal performance, an integrated system should distribute work amongst its components in accordance with the strengths and weaknesses of each.

The primary considerations for a real-time graphics pipeline are:

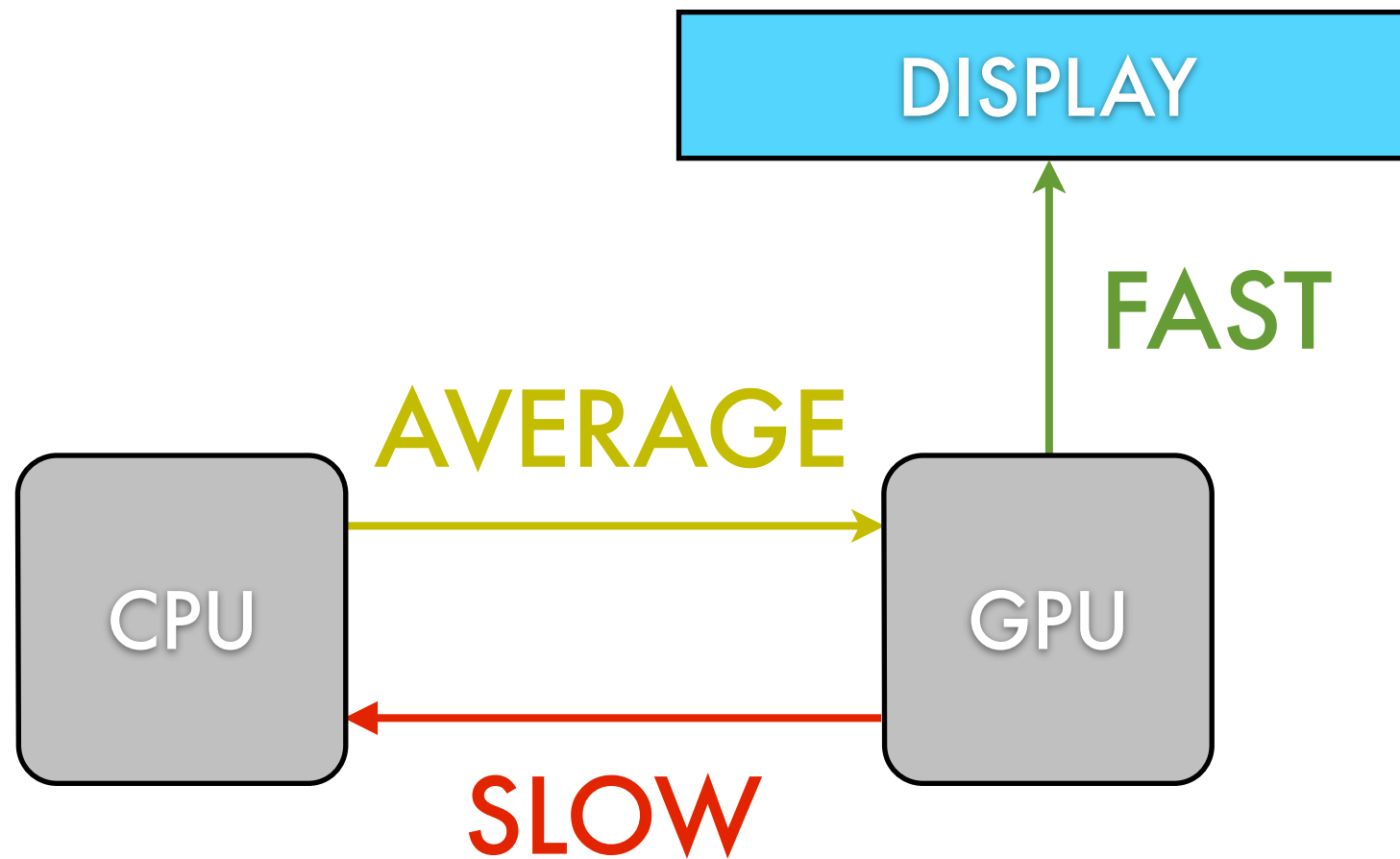
- Computation speed
- Data transfer speed

The GPU is extremely fast, but designed for specific conditions.
The CPU is slower, but designed for all terrains.



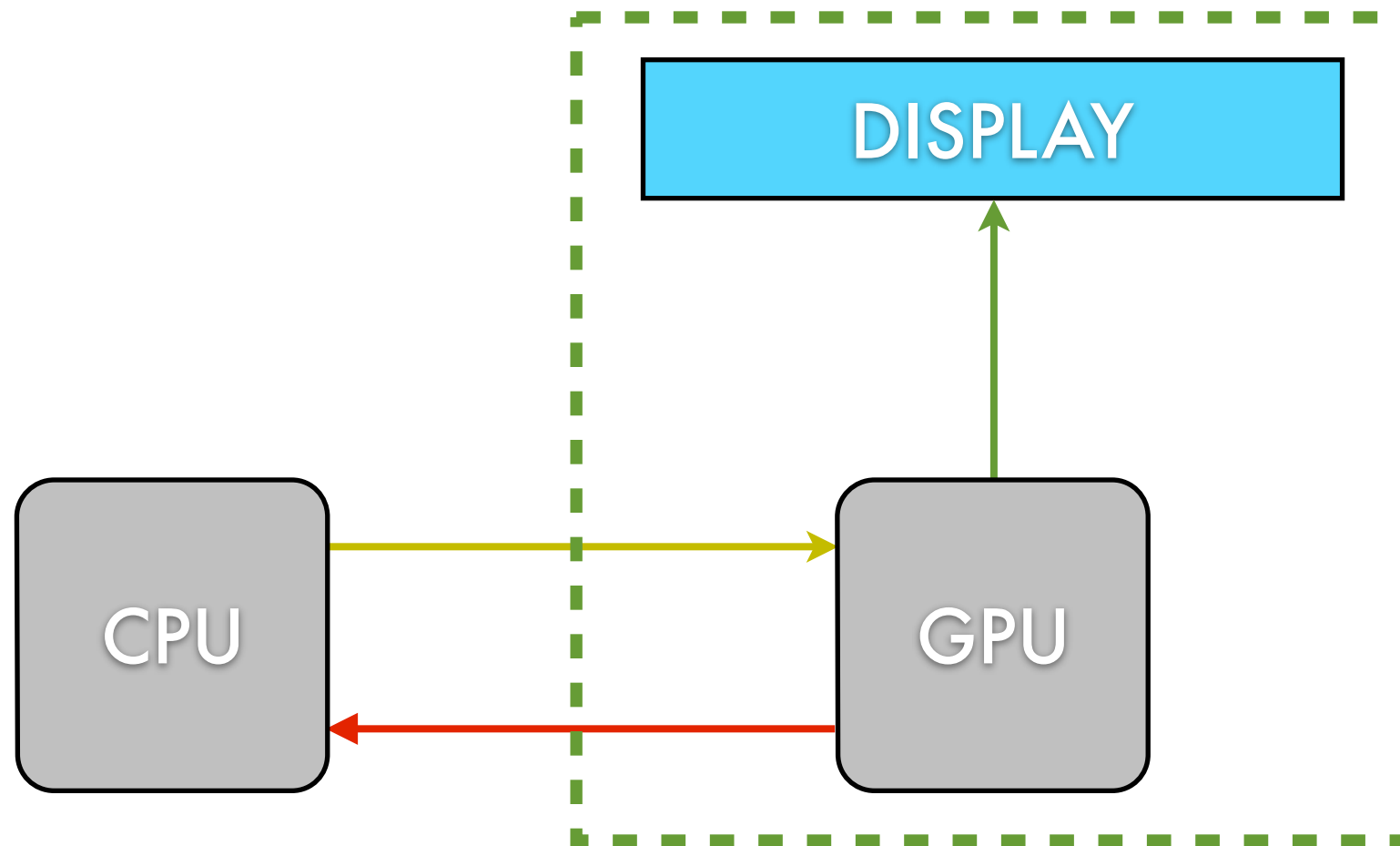
Therefore, any work that can be done on the GPU should be.
Everything else is left to the CPU.

Data Transfer Speeds



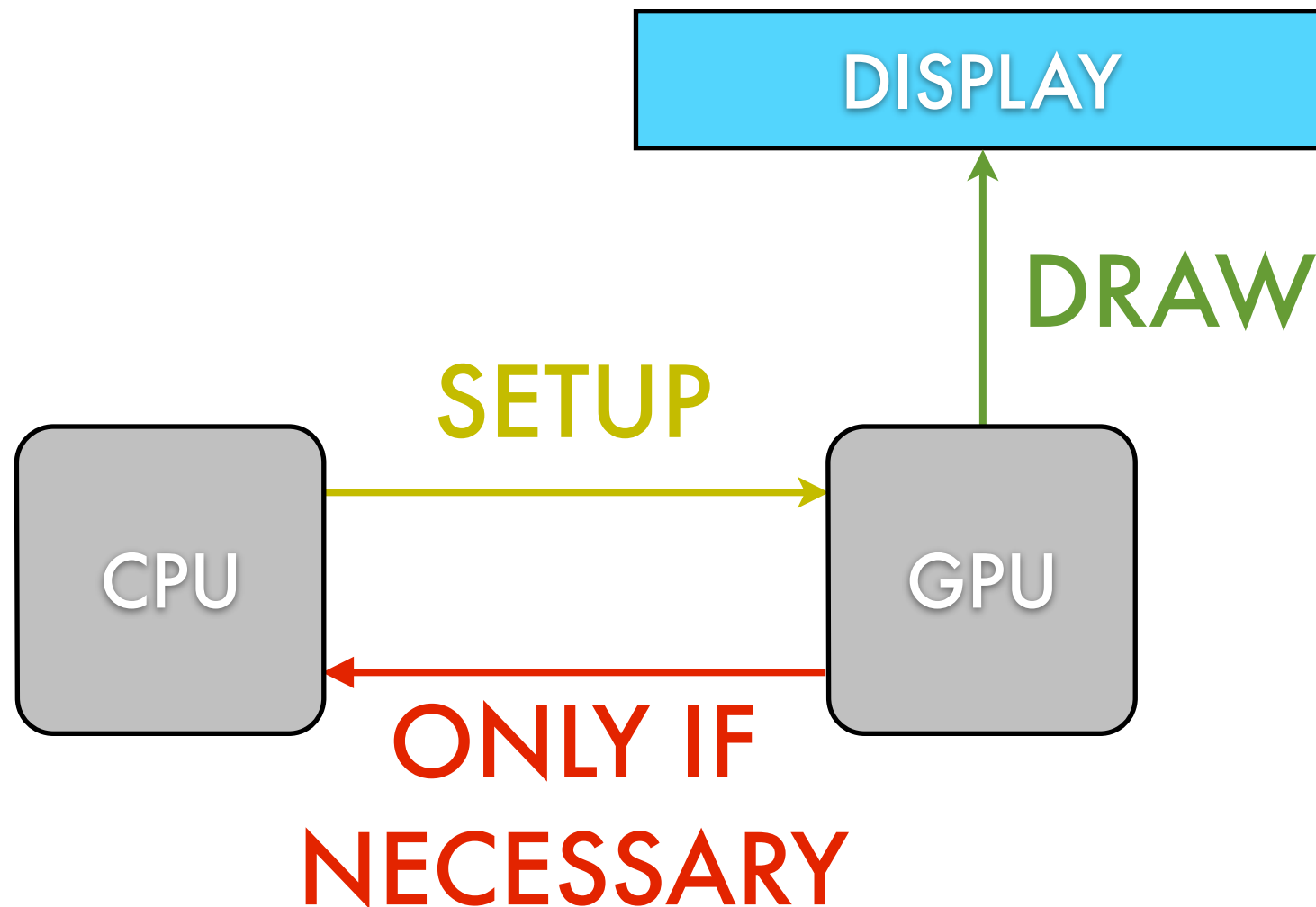
Data transfers between the various hardware components are not equally costly.

The Implications of Data Transfer Speeds



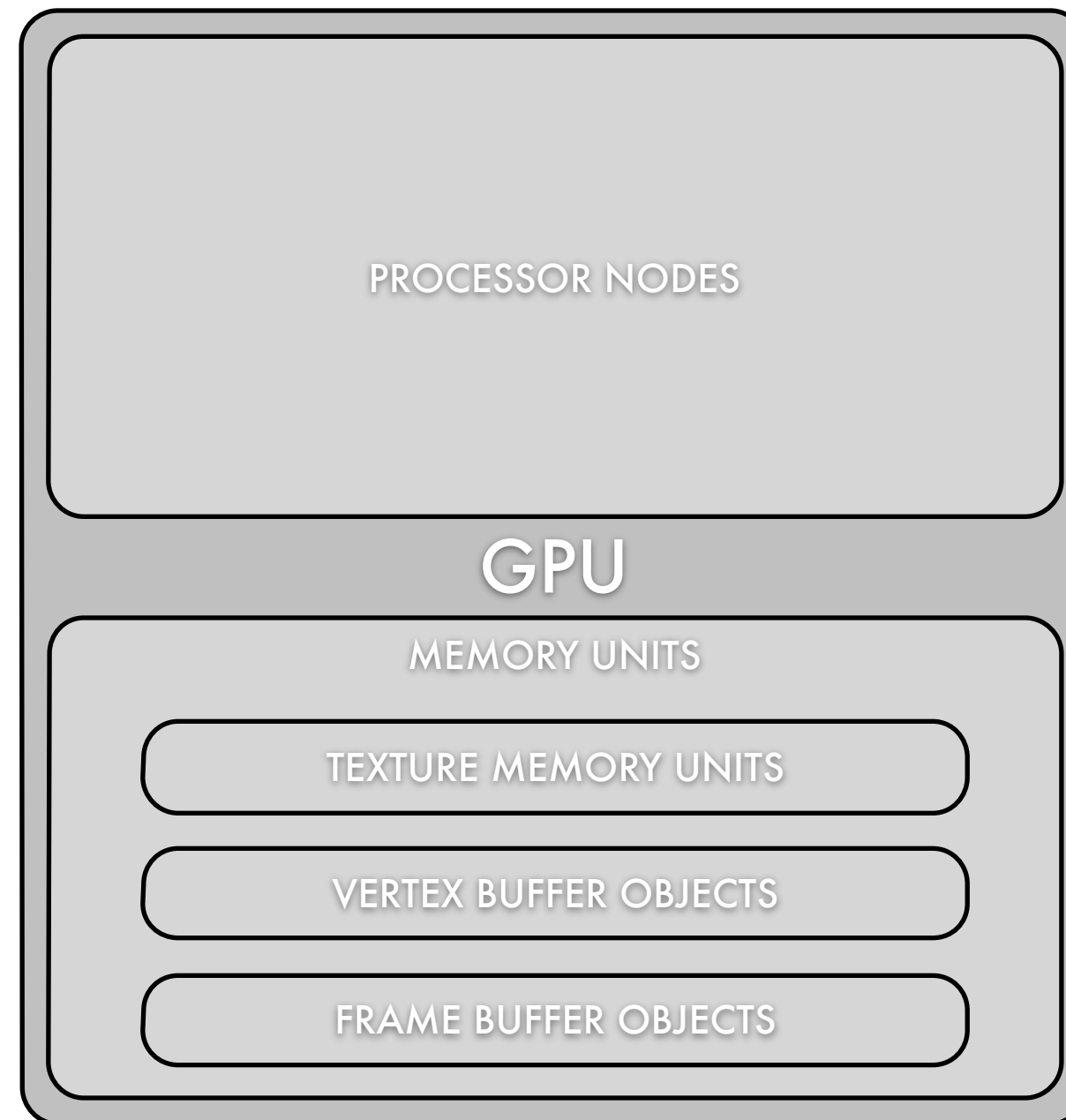
We should design software that keeps the majority of transfers between the GPU and DISPLAY.

In other words, it would be best to pass the bulk of our geometric data to the GPU during setup().



Within the draw() loop, the GPU should handle most data manipulations with minimal input from the CPU.

This is achieved by pre-loading large data elements such as images and vertex lists into the GPU's persistent memory units.



Graphics APIs and Shader Languages provide tools for accessing and manipulating the data stored in GPU memory with minimal CPU usage.