CS333: Storage Systems

Williams College
Spring 2021

Conference 0: Course Trajectory & Themes

Course Description

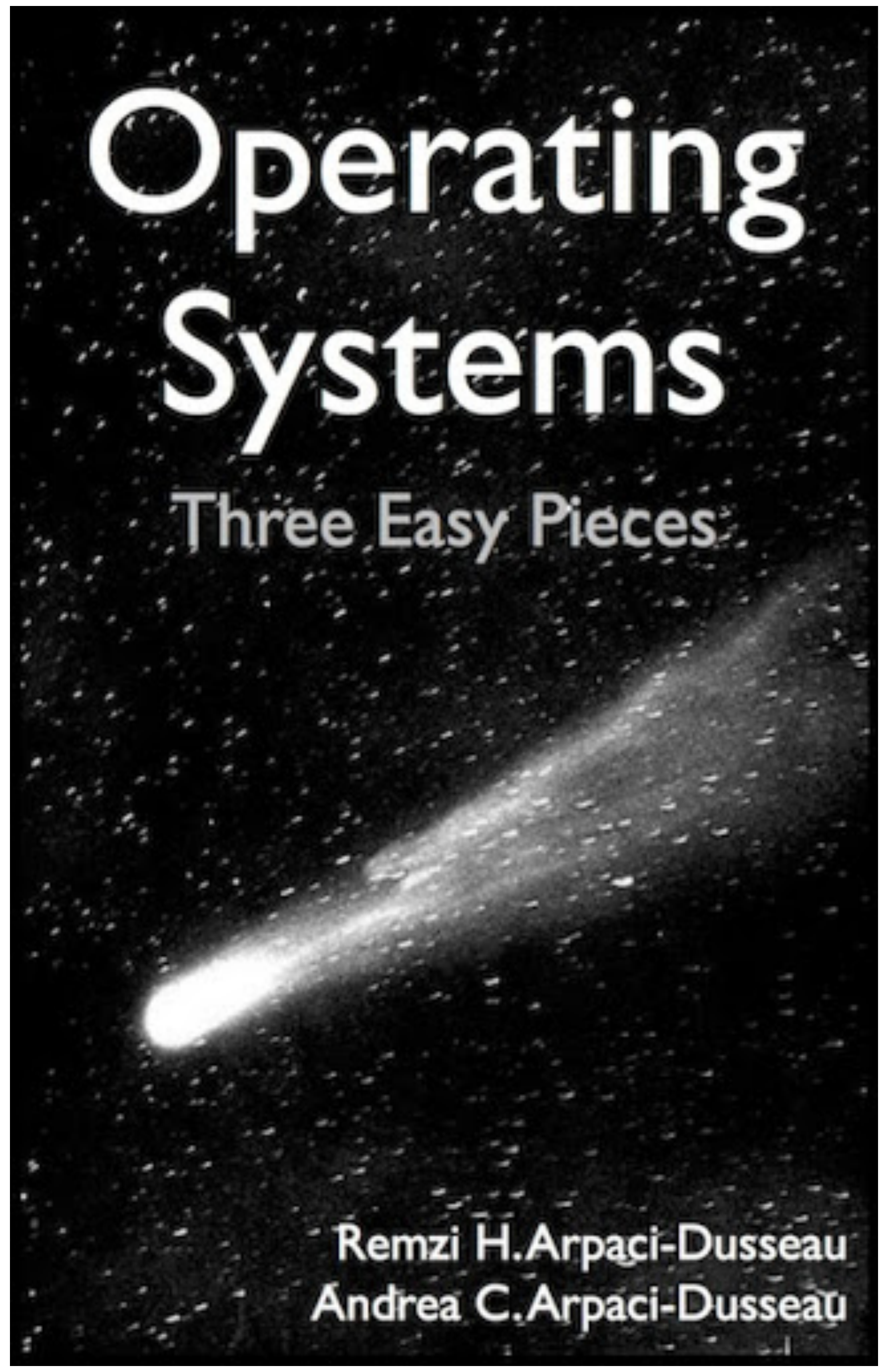
This course will examine topics in the **design**, **implementation**, and **evaluation** of storage systems. Topics include:

- the memory hierarchy;
- ways that data is organized (both logically and physically);
- storage hardware and its influence on storage software designs;
- data structures;
- performance models; and
- system measurement/evaluation.

Readings will be taken from OSTEP v1.0 (first half of the semester) and recent/influential technical literature (second half). Our readings and discussions will place an emphasis on identifying and evaluating **design trade-offs**.

Course Trajectory

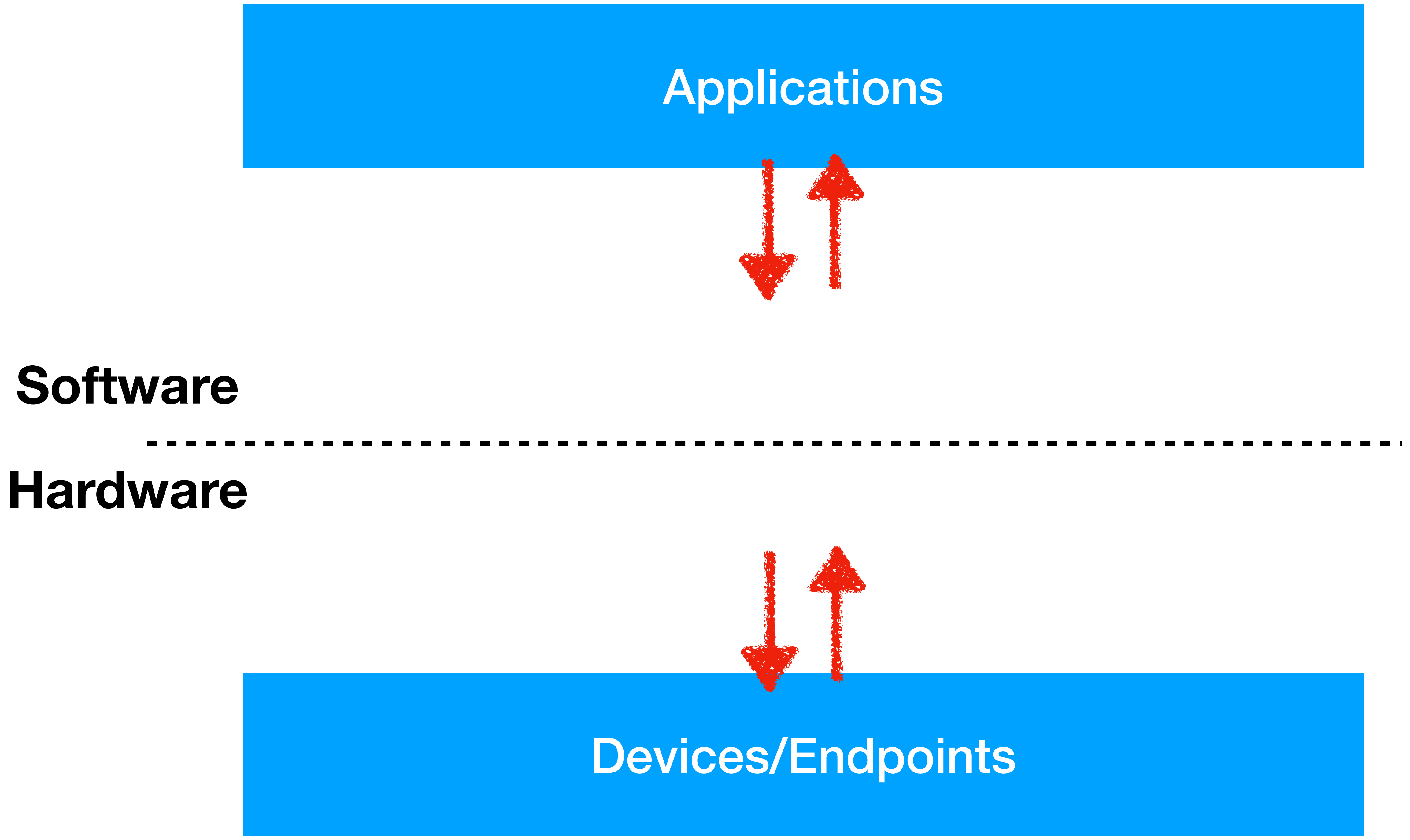
- First half: Operating Systems: Three Easy Pieces (v1.0)
  - Build foundation from bottom (devices) up through applications
  - Available for free; most up-to-date PDFs linked on webpage
- Second half: Current/influential research papers
  - Papers posted on schedule like rest of readings
    - Chosen with a trajectory in mind, but there is some flexibility if you are excited about a certain topic— let me know!
    - **Optional** readings often posted for diving deeper—often the main reading is chosen because it is clearly written or influential, not necessarily most recent



Overview and Themes

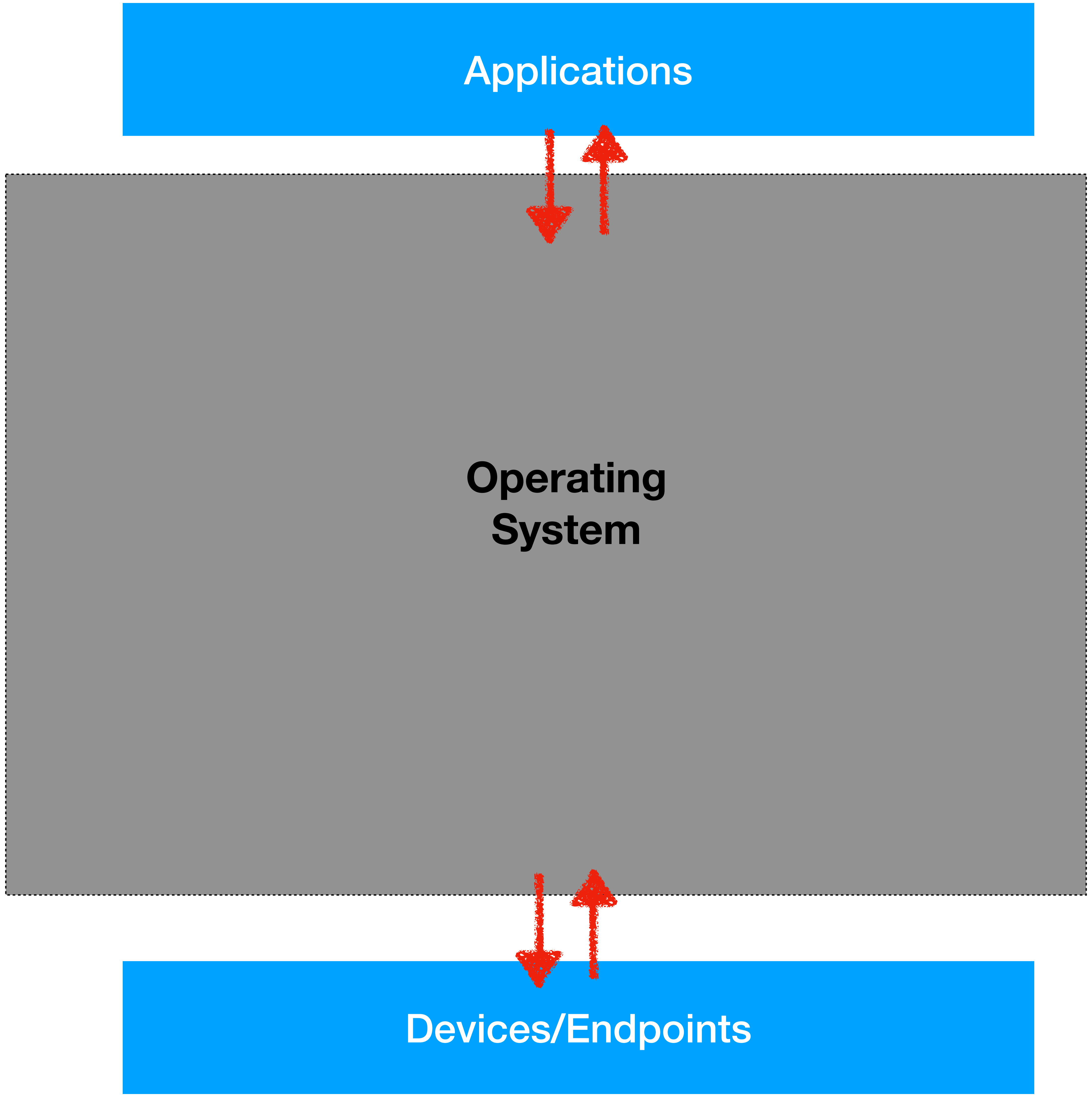
Schedule and Course Topics

Simplified "Storage Stack"



- What are some common/important applications that store/access *persistent* data?
- What interfaces do applications use to read/write their data?
- What is missing from this diagram?
- What interfaces do devices provide?
- What are some devices/endpoints that we might use to store data *persistently?*

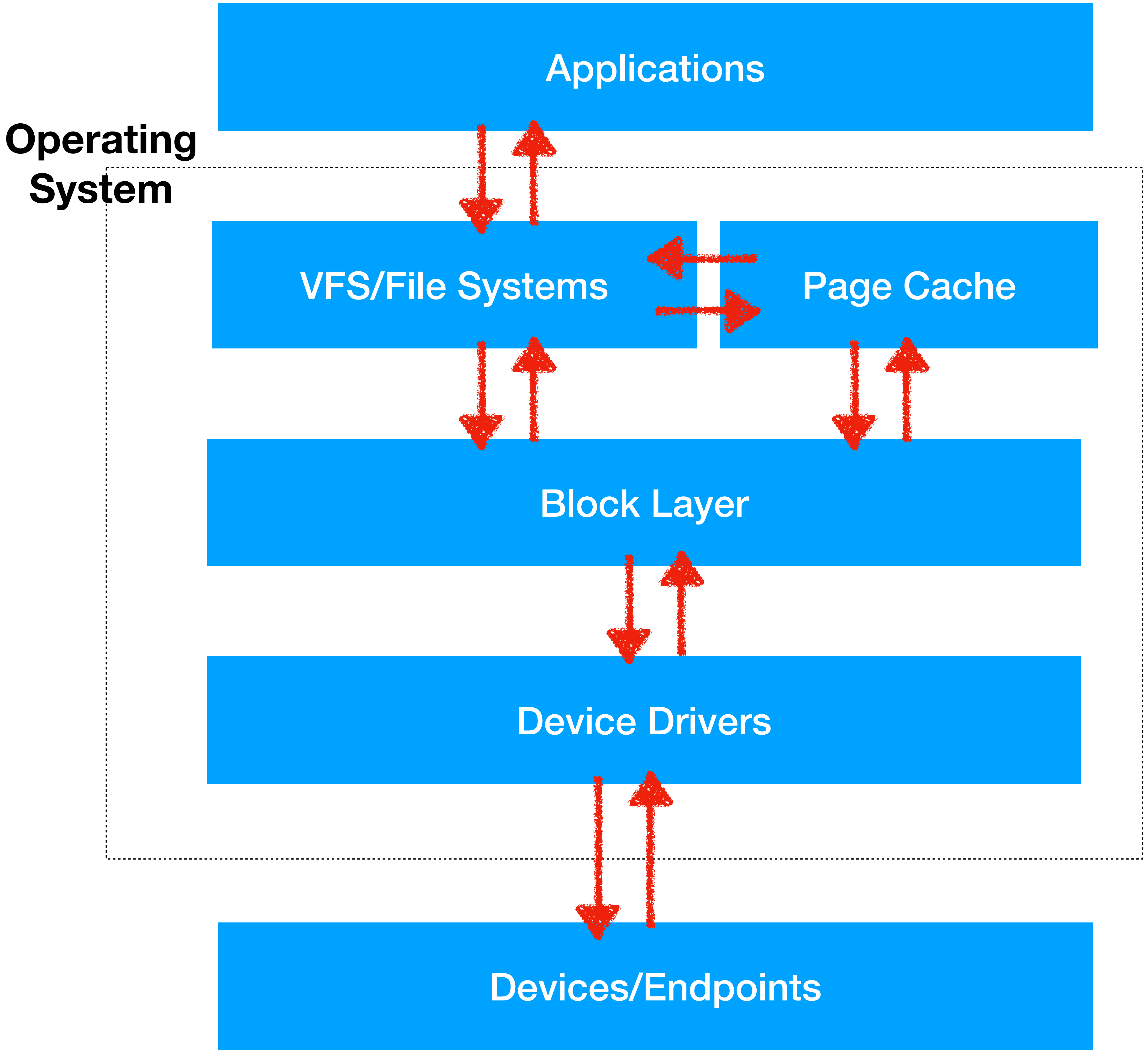
Simplified "Storage Stack"



The Operating System wears a lot of hats, but one of its jobs is to give applications a way to index their data in a uniform way.

- What pieces of the OS are responsible for managing storage?

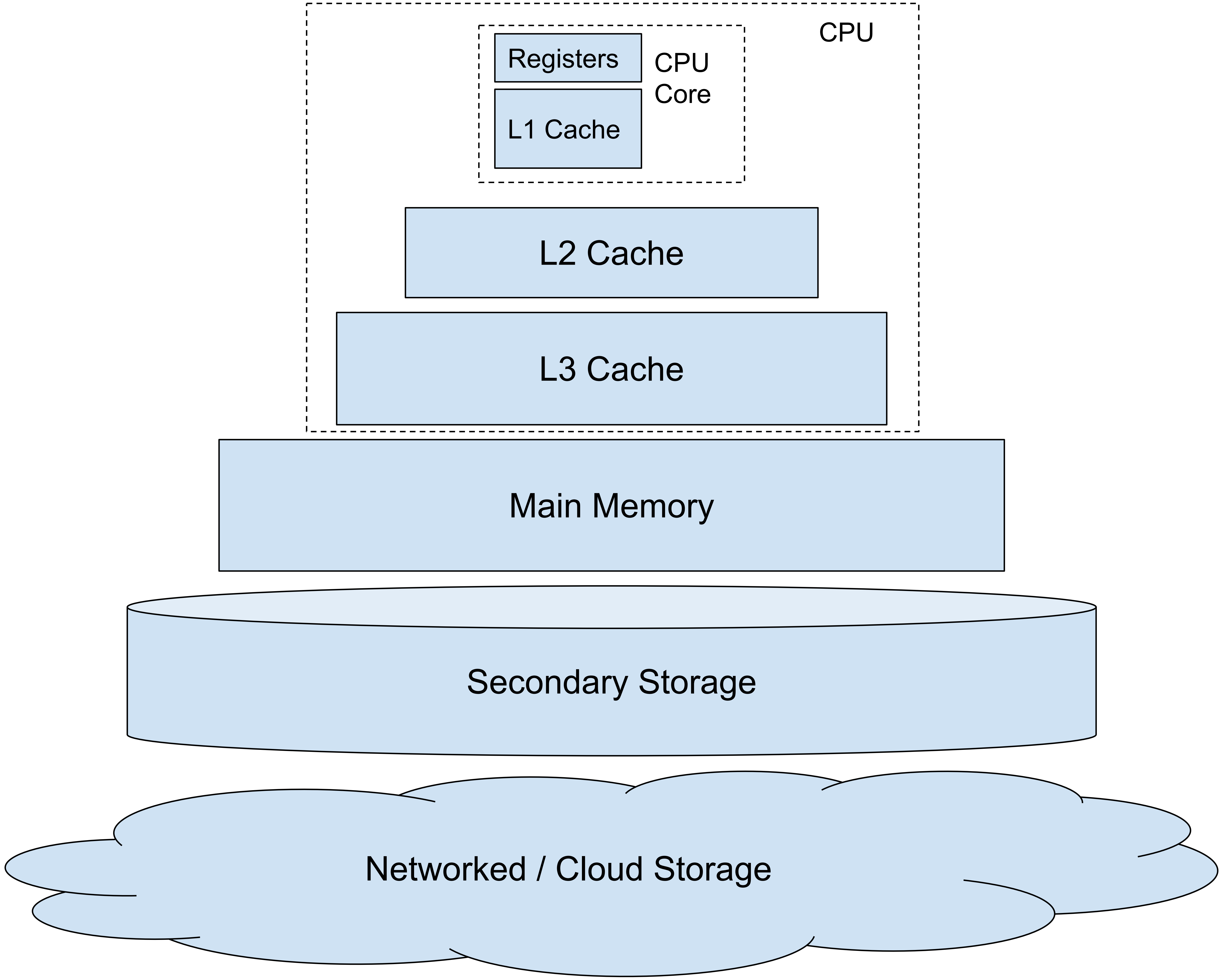
Simplified "Storage Stack"



The Operating System consists of a series of interacting layers.

- What are the interfaces at layer boundaries?
- What are the requirements/needs at each layer?
- What are the costs/challenges of changing the details at each layer?
  - Adding APIs?
  - Removing APIs?
  - Modifying APIs?

The Memory Hierarchy



- **What are the rough sizes at each level (units)?**
- **What are the rough speeds at each level?**
  - **Latency**
  - **Bandwidth**
- **Are there different choices available to systems designers/builders at any of the levels? Why would you favor one choice over another?**
- **What is the unit of transfer between each level, and what influences the sizes of those data transfer units at each level?**

Now, let's take a moment and try to relate the memory hierarchy to the design of the software storage stack from the previous slide.

Course Trajectory

- We will identify different design decisions and trade-offs made all throughout the storage stack
  - How do these decisions affect the way we design applications?
  - How do these decisions affect performance?
  - How do these decisions affect correctness?
- We'll apply performance models in order to better understand these design tradeoffs
- We'll explore different devices and how those devices affect the ways we design software
- We'll explore new data structures, and how those data structures can be used to improve system designs

Course Structure

- Any questions about logistics?

- Course website:
  - http://cs.williams.edu/~jannen/teaching/s21/cs333/index.html