CS 4530: Fundamentals of Software Engineering
Lesson 2.2 The Architectural Scale

Jonathan Bell, Adeel Bhutta, Ferdinand Vesely, Mitch Wand

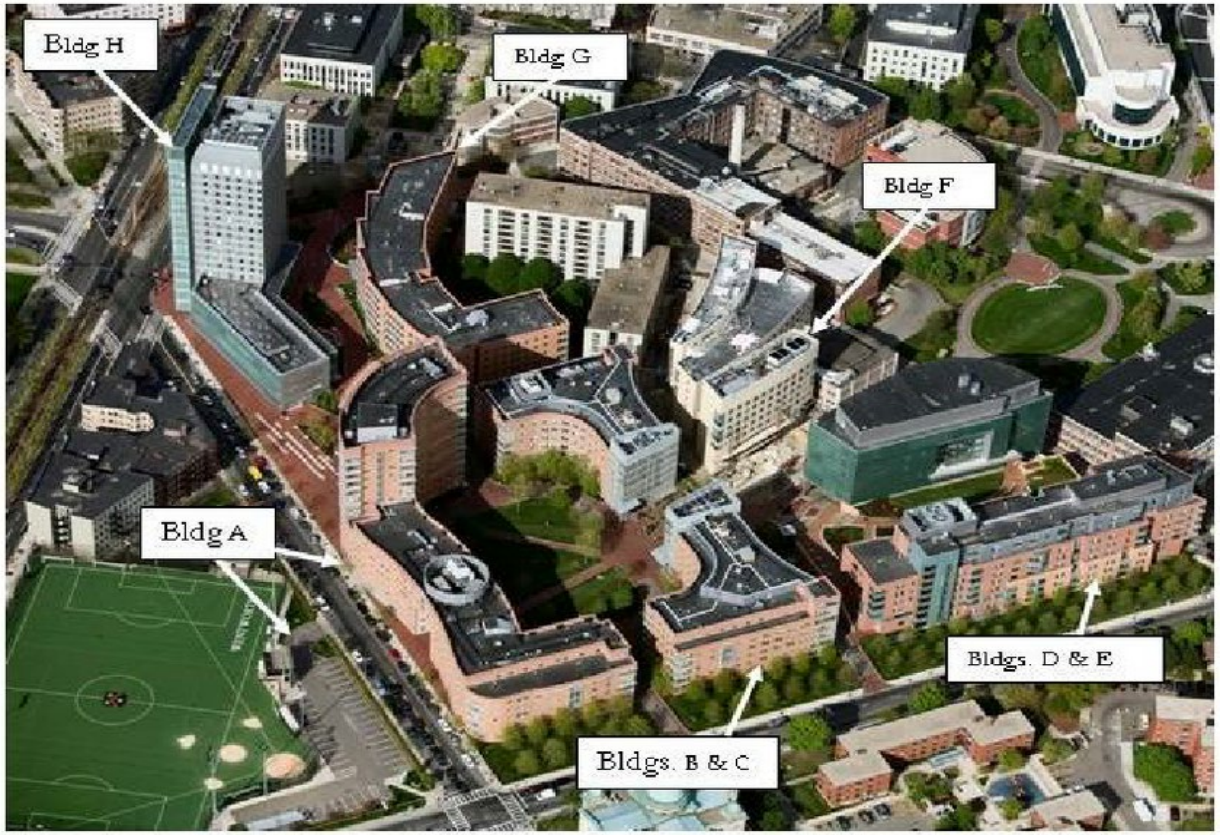
Khoury College of Computer Sciences

Learning Goals for this Lesson

- At the end of this lesson, you should be able to
  - Give 5 examples of software architectural styles and their distinguishing characteristics
  - Draw a picture or give an example to illustrate each one

The Architectural Scale

- key questions: what are the pieces? how do they fit together to form a coherent whole?



What do we learn at this scale?

- Knowing the top-level organization gives you the first clue about
  - how to understand the system
  - where to look for bugs or explain behaviors
  - how to organize into teams
  - how to find modification and extension points

Examples of Architectural Styles

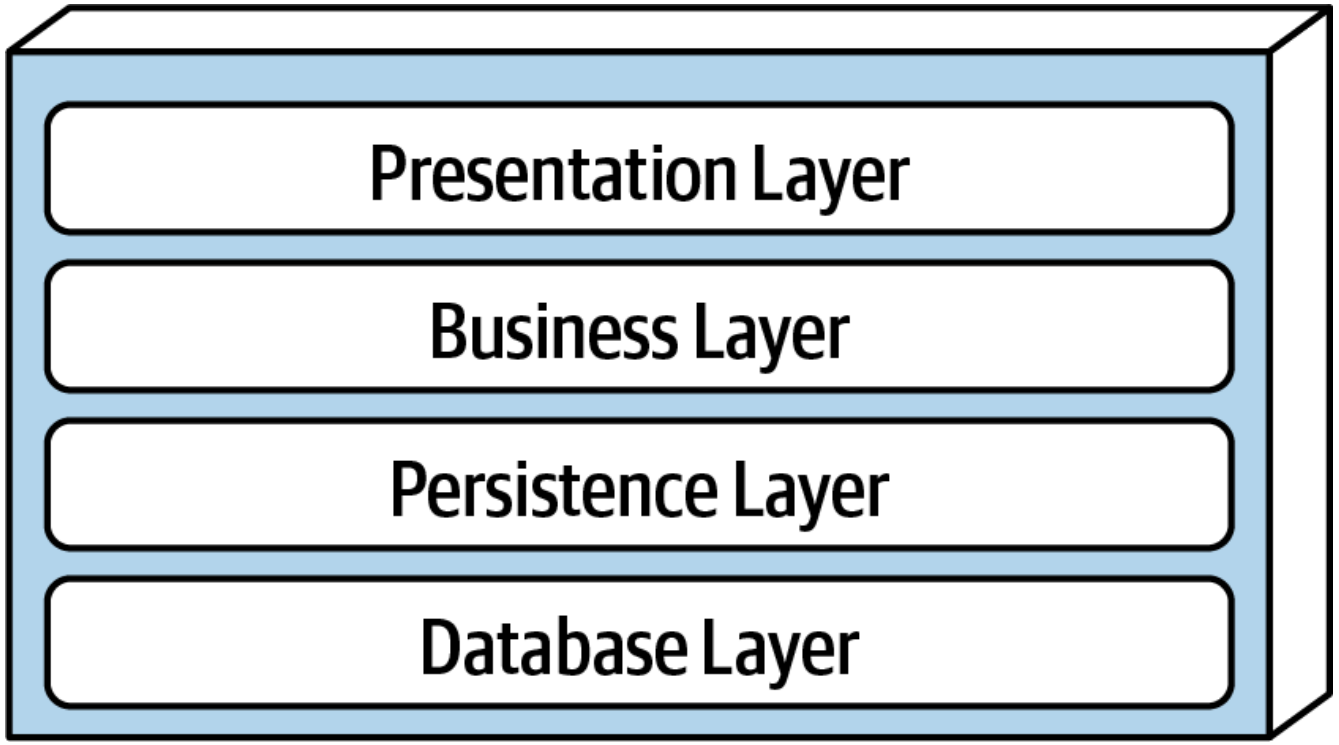
1. Object-oriented
2. Layered
3. Pipeline
4. Microkernel
5. Event-driven

Example 1: Object-Oriented Architecture

- The pieces of the program correspond to entities in the real world.
- Properties & operations correspond to operations in the real world
- Example: solitaire:
  - Entities:
    - card
    - pile
    - layout
  - Operations:
    - find the number and suit of a card
    - move a card from one pile to another
    - check to see if a given move is legal
    - check to see if the layout is in a winning state

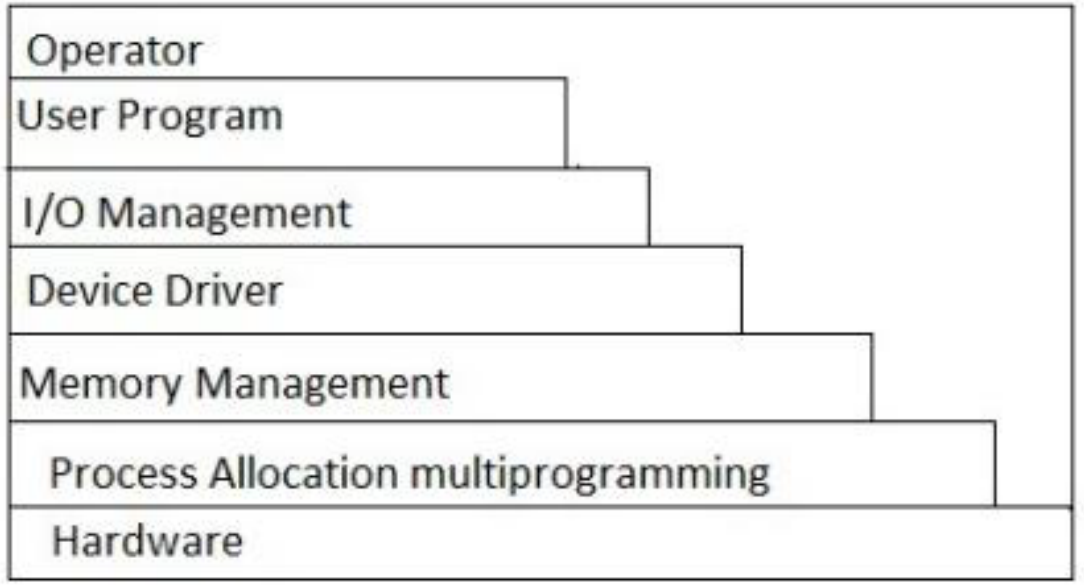
Example 2: Layered Architecture

- The pieces correspond to level of concern.
- Each layer depends on services from the layer or layers below
- Organize teams by Layer
  - different layers require different expertise
- When the layers are run on separate pieces of hardware, they are sometimes called "tiers"



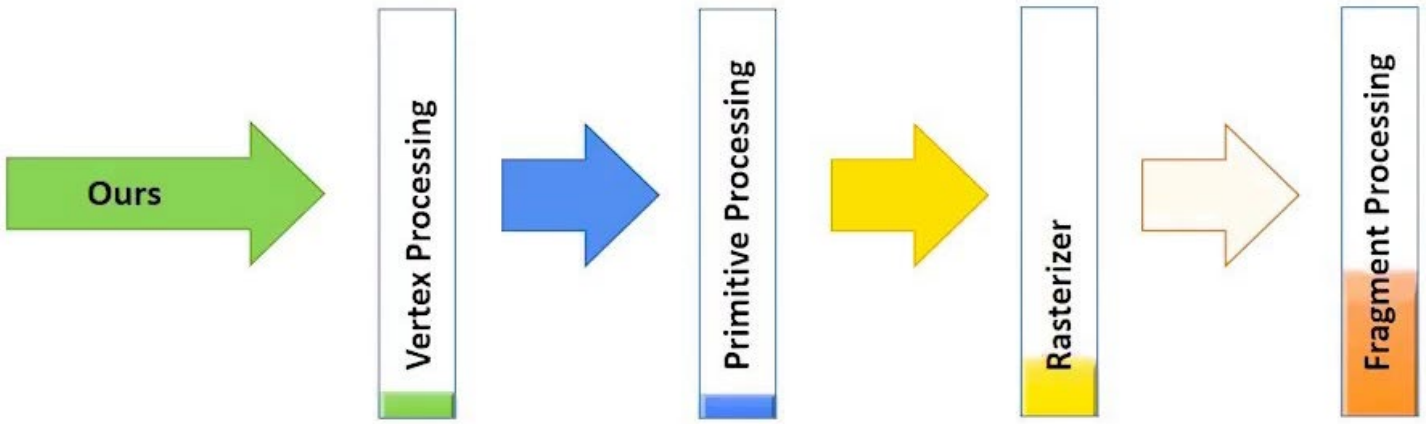
Layered Architecture (contd)

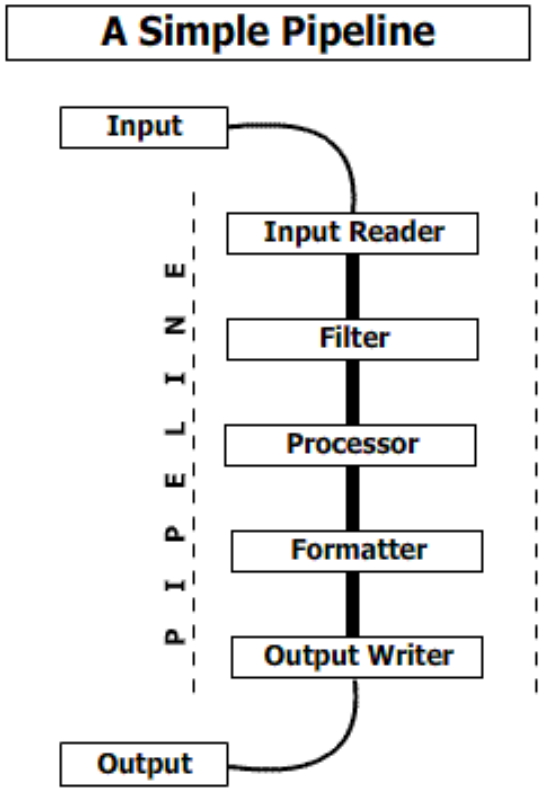
- Typical organization for operating systems
- Layers communicate through procedure calls and callbacks (sometimes called "up-calls")



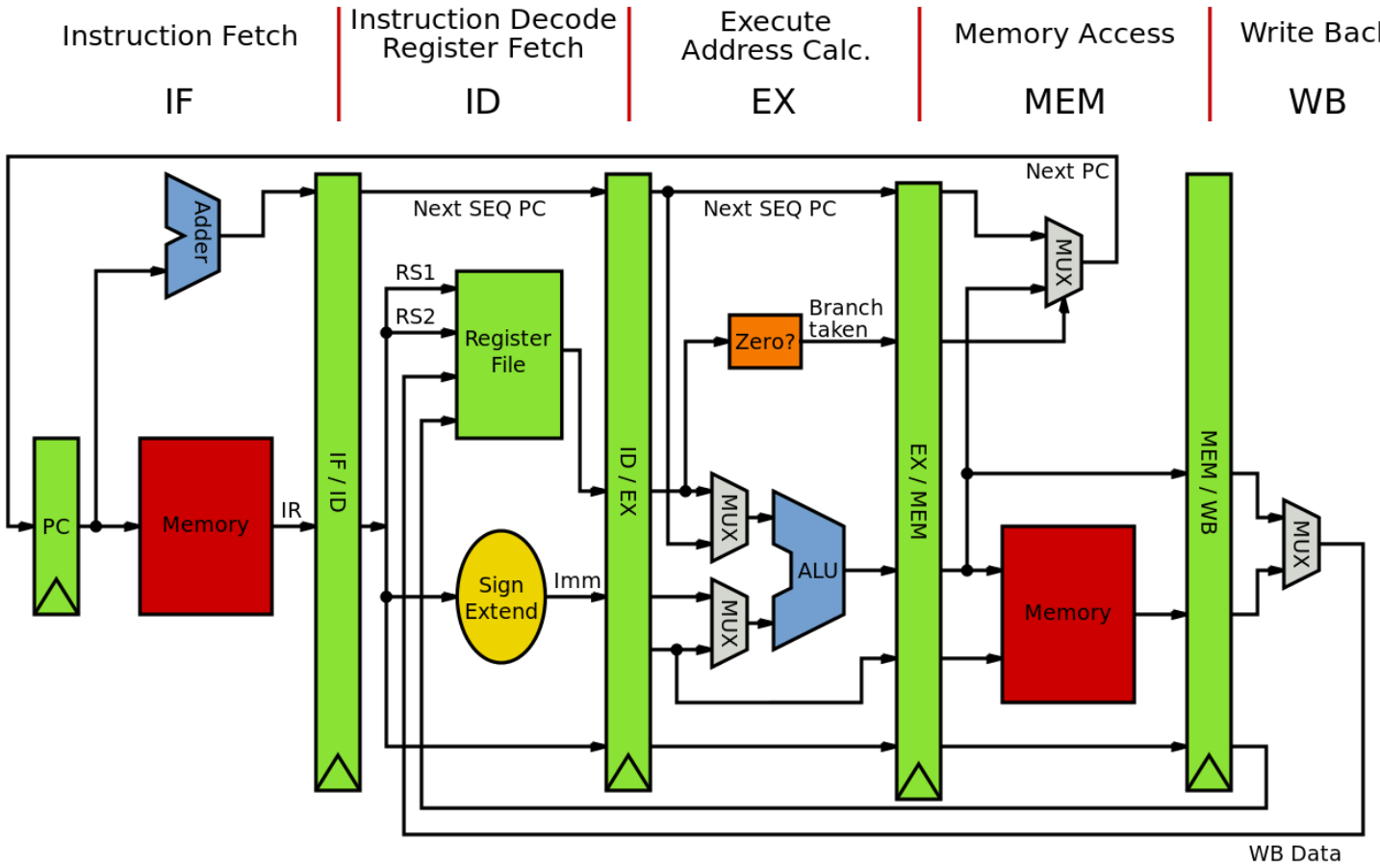
Example 3: Pipeline Architecture

- The pieces correspond to stages in the transformation of data in the system
- Good for complex straight-line processes, e.g. image processing





Also good for visualizing hardware

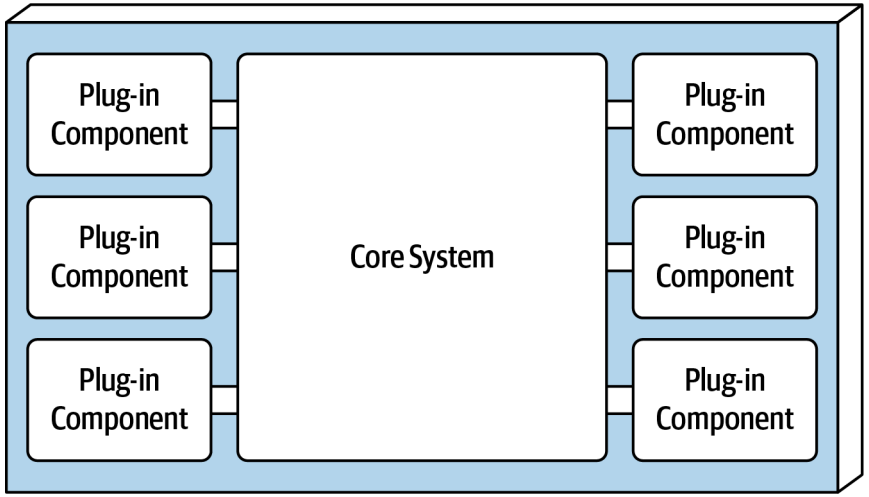


How do the stages communicate?

- That's the next-level decision
  - data-push (each stage invokes the next)
  - demand-pull (each stage demands data from its predecessor)
  - queues? buffers?
  - This is at the interaction scale (coming up next lesson)

Example 4: A Plugin Architecture ("microkernel")

- Components consist of
  - a small core (the "microkernel") for essential functions, and lots of hooks for adding other services
  - Plug-ins corresponding to different user functions
- Highly extensible
- Plug-ins can be designed by small, less-experienced teams– even by users!
- Connection methods may vary



Analogy: Affordances

- An “affordance” is a property of an object that enables some operation to be performed with/on that object
  - Eg: zipper handles enable opening/closing
  - Door handles enable opening/closing
  - Spouts enable pouring
- In a microkernel architecture, the core contains affordances for extension
- These affordances may have many possible forms

Plugin Examples

- Many examples:
  - Visual Studio Code (internal org. + extension marketplace)
  - emacs (emacs-lisp + hooks)
  - git clients

```
$ ls .git/hooks
applypatch-msg.sample      pre-applypatch.sample      pre-rebase.sample
commit-msg.sample          pre-commit.sample          pre-receive.sample
fsmonitor-watchman.sample  prepare-commit-msg.sample  update.sample
post-update.sample         pre-push.sample
```

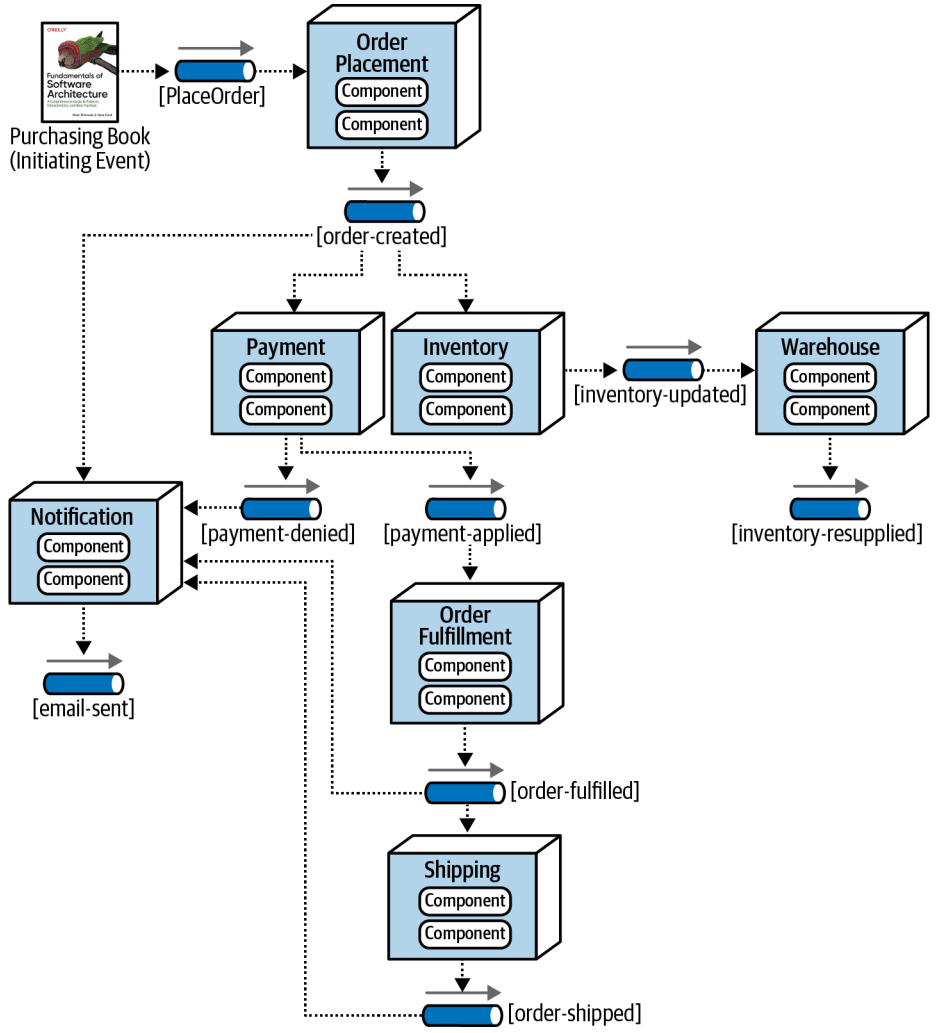
Express.js provides methods for modifying its built-in actions

```
1  const express = require('express' 4.17.2 )
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    res.send('Hello World!')
7  })
8
9  app.listen(port, () => {
10   console.log(`Example app listening on port ${port}`)
11 })
```

https://expressjs.com/en/starter/hello-world.html

Example 5: Event-Driven Architecture

- Metaphor: a bunch of bureaucrats shuffling papers
- Components correspond to stages in the flow of data through the system (not necessarily a straight-line flow)
- Each processing unit has an in-box and one or more out-boxes
- Each unit takes a task from its inbox, processes it, and puts the results in one or more outboxes.
- Stages are typically connected by asynchronous message queues.
- Conditional flow



Each piece can have its own architecture

- We can use the same ideas to talk about each piece of the overall architecture.
- For example, the backend of covey.town (the subject of our team project) is a layered architecture. The higher levels (those closest to the user) are event-driven, but the lower layers are more object-oriented.

Learning Goals for this Lesson

- At this point, you should be able to
  - Give 5 examples of software architectural styles and their distinguishing characteristics
  - Draw a picture or give an example to illustrate each one