Distributed systems

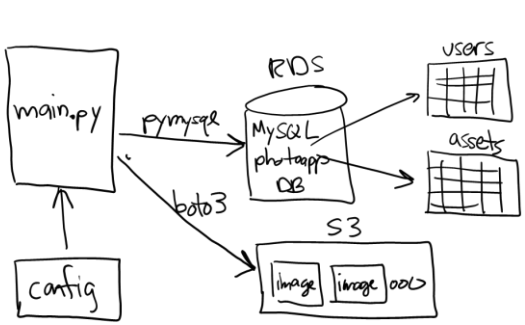
- **Distributed systems are hard**
- **Types of distributed systems**
- **Properties of distributed systems**
  - *Performance, security, scalability, availability, consistency*

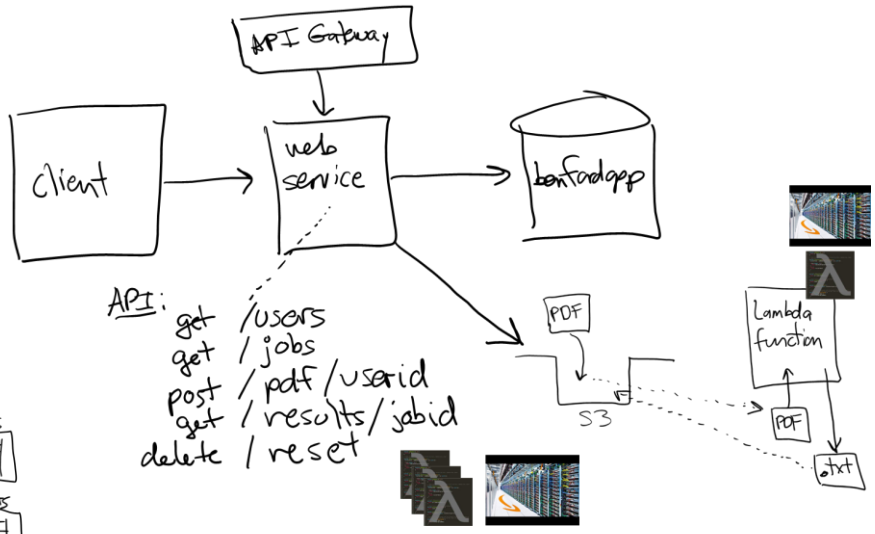
- [CS 340/440, CS 345/445]

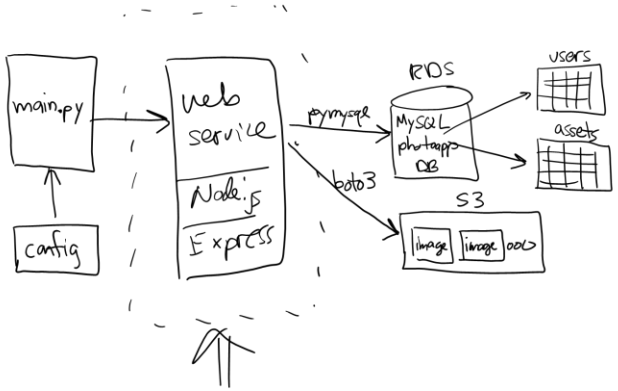
Distributed systems

- **Cloud-based apps are distributed systems**
  - *N > 1 computer programs communicating via messages*

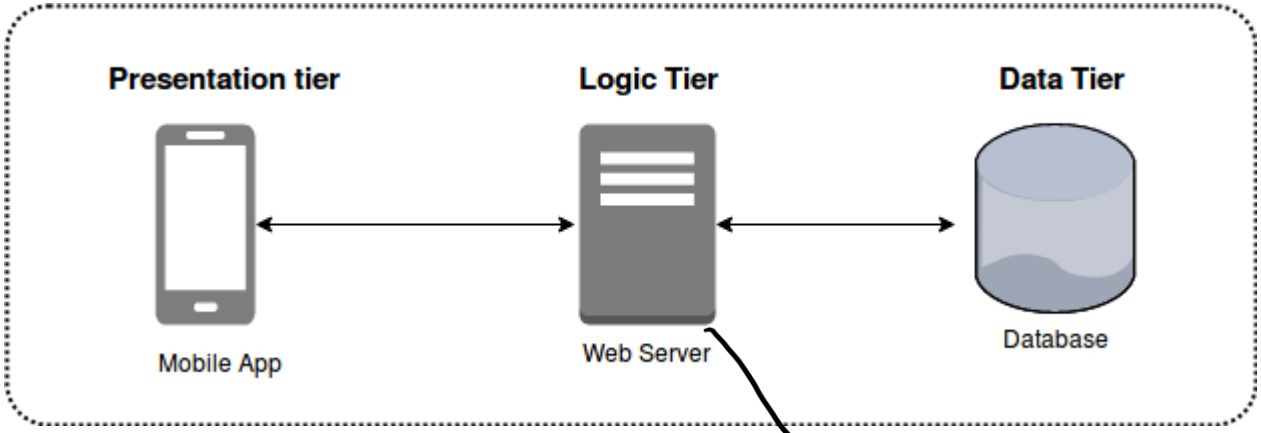
- **All our projects are distributed systems:**

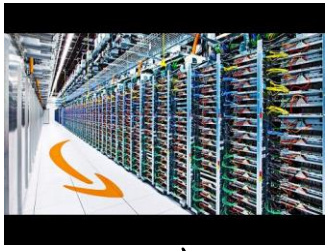






Our projects: basic multi-tier design

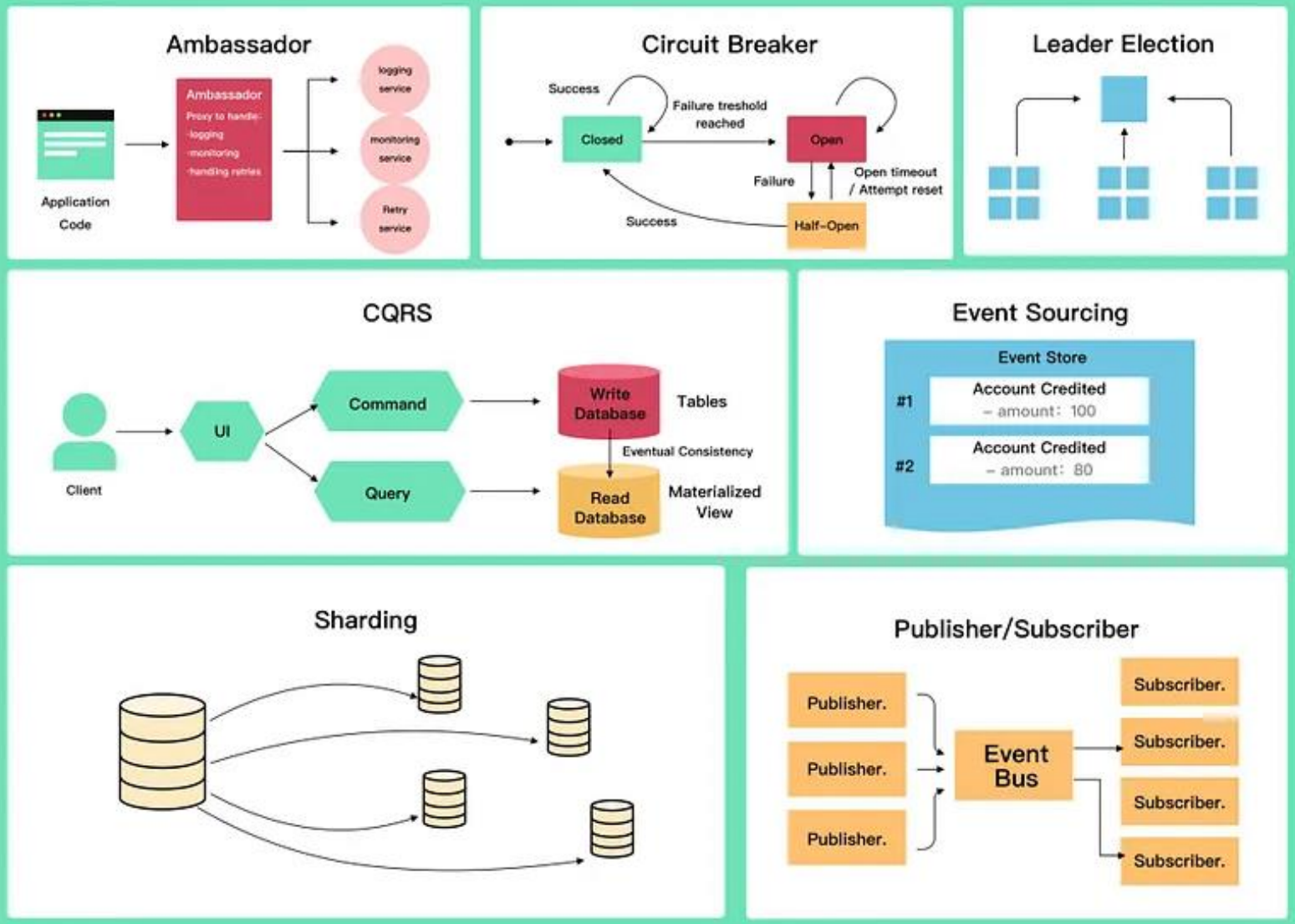




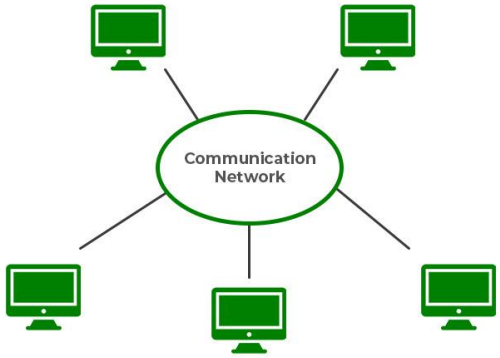
execute tier

Top 7 Most-Used Distributed System Patterns

ByteByteGo.com



Distributed systems are HARD

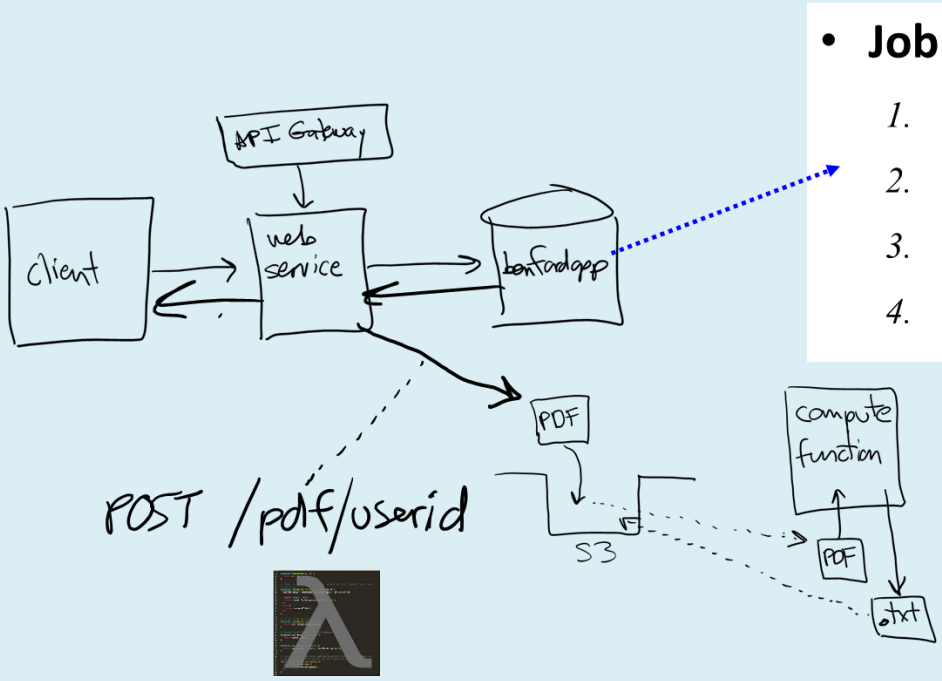


- **Distributed systems are HARD to build**
  - *What if the network goes down?*
  - *What if one of the machines crashes?*
  - *How to secure? What if one of the machines is hacked?*
  - *How to prevent hackers from impersonating a machine?*

Courses in distributed systems:
CS 345/455

Example

- **Example:**
  - *In project 03, **POST /pdf** uploads PDF to S3 & updates DB*

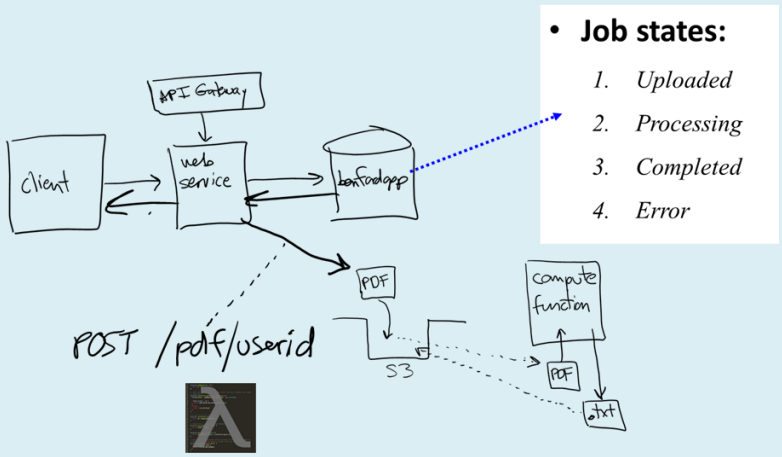


- **Job states:**
  1. *Uploaded*
  2. *Processing*
  3. *Completed*
  4. *Error*

What scenario(s) did we ignore in programming proj03_upload?

Answers

- **In the proj03_upload function:**
  - *What if the PDF upload works but the update fails?*
  - *What if the DB update works but the upload fails?*
  - *If PDF upload fails, should we retry? How many times?*
  - *If the DB update fails, should we retry? How many times?*
  - *How do we get errors back to the client in all cases?*

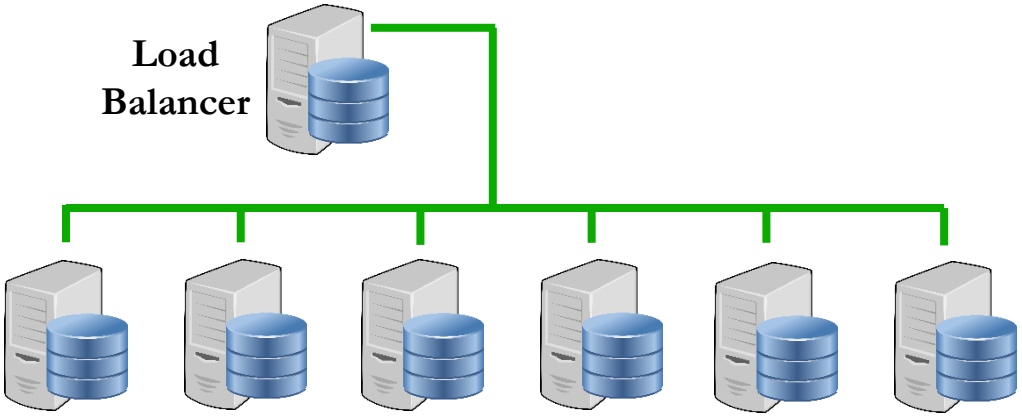


Properties of distributed systems

- **The most important properties to consider:**
  - *Performance (latency, bandwidth, throughput)*
  - *Scalability*
  - *Security*
  - *Availability / fault tolerance*
  - *Consistency*

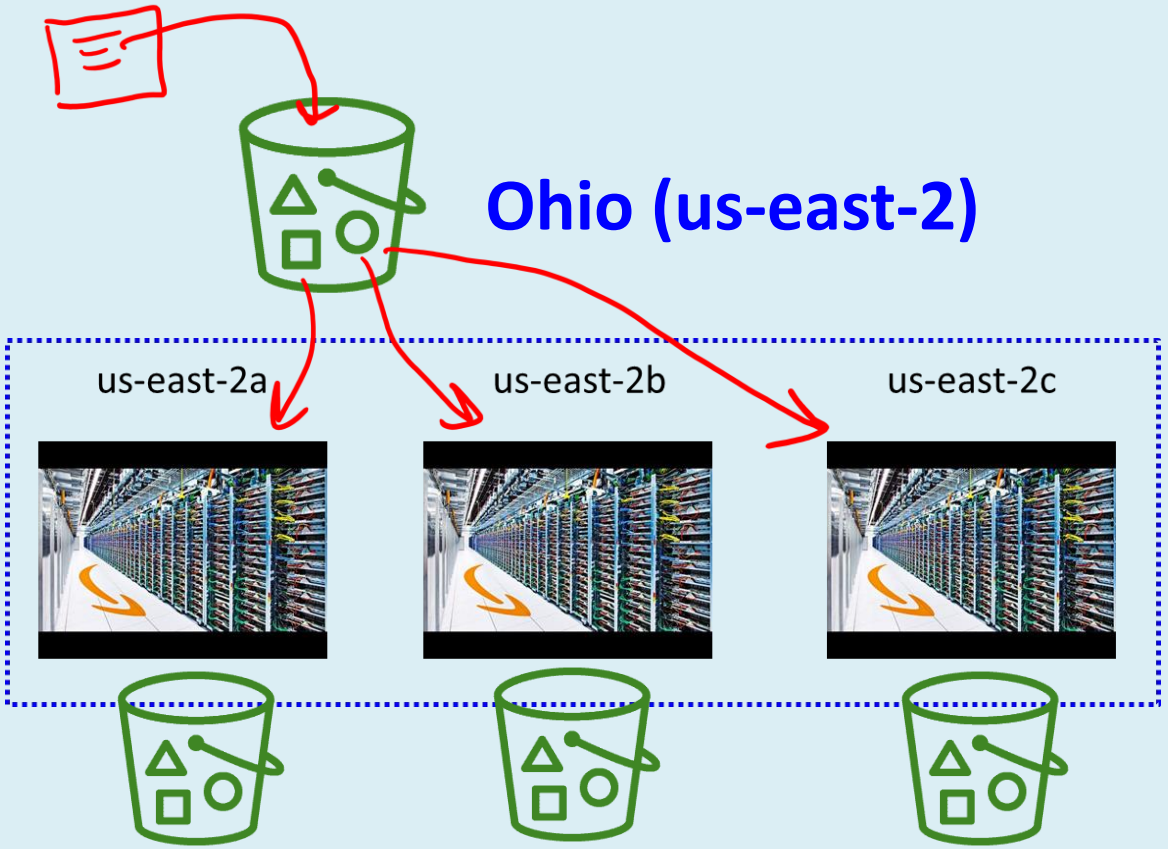
Availability / fault tolerance

- Computers crash, it will happen…
- Only way to keep your system **available** is with multiple computers
- A system that keeps running in presence of failures is **fault tolerant**



Example: S3 is fault tolerant

- **Buckets are replicated across all sites in a region in case one of the sites loses power / network connectivity…**

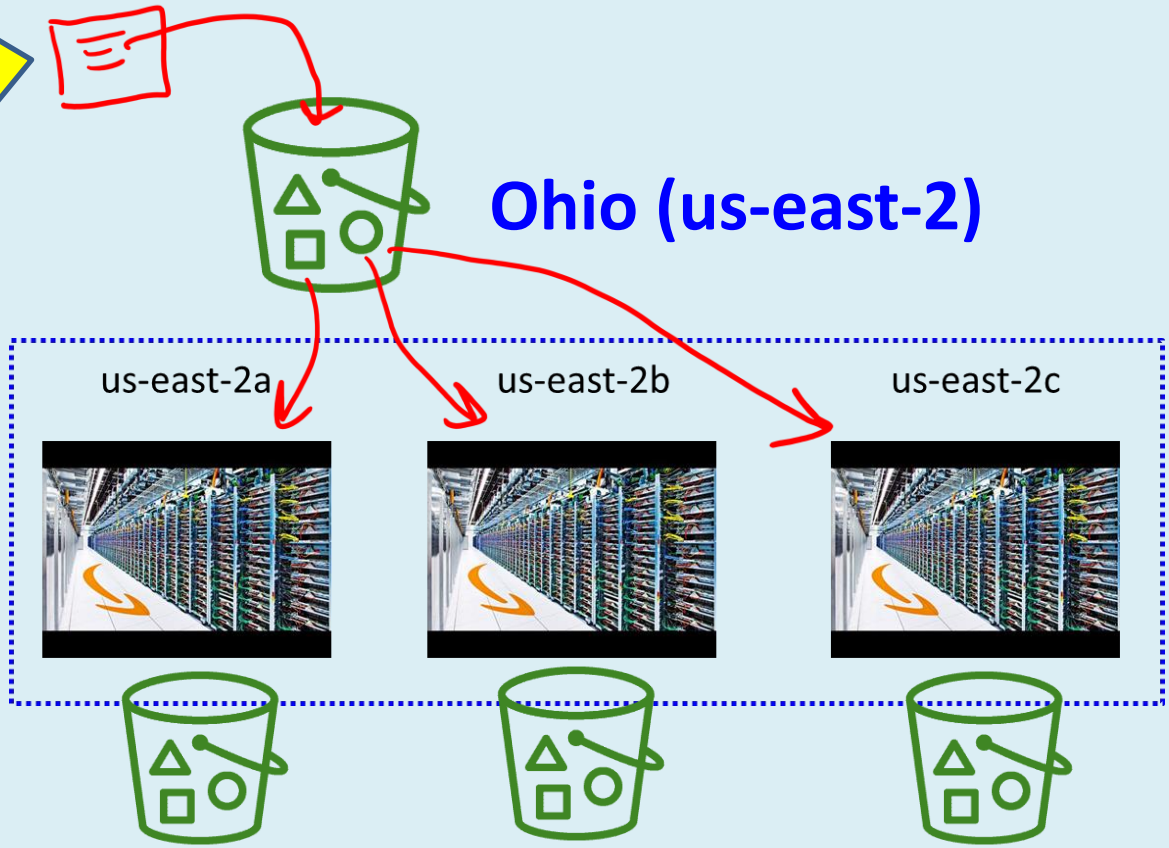


Interesting question…

- **Replicating data leads to an interesting problem…**

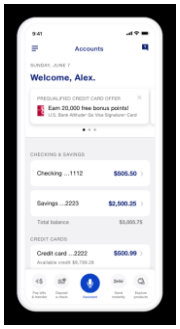
If we upload an image to S3, how soon is it available in all 3 sites?

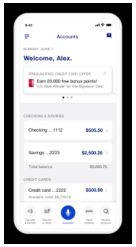
What if a download request is routed to a site that hasn't finished replicating? What happens? What does AWS guarantee, if anything?



Another version of the same question…

- Suppose a bank account has exactly $1,000
- How does the bank prevent two different people from withdrawing $1,000 from that account at the same time?





That's it, thank you!