Data Service Scalability

UWB HACKATHON: 2020

Let’s Build a Service

- Spec
  - Scenario: Photo Repository
  - Upload / Download from repository
  - Scale: dozen users
  - Durability: Sure
  - Security: No

v1

Availability: Poor
Scalability: Poor
Latency: ok
Consistency: Excellent

Definitions

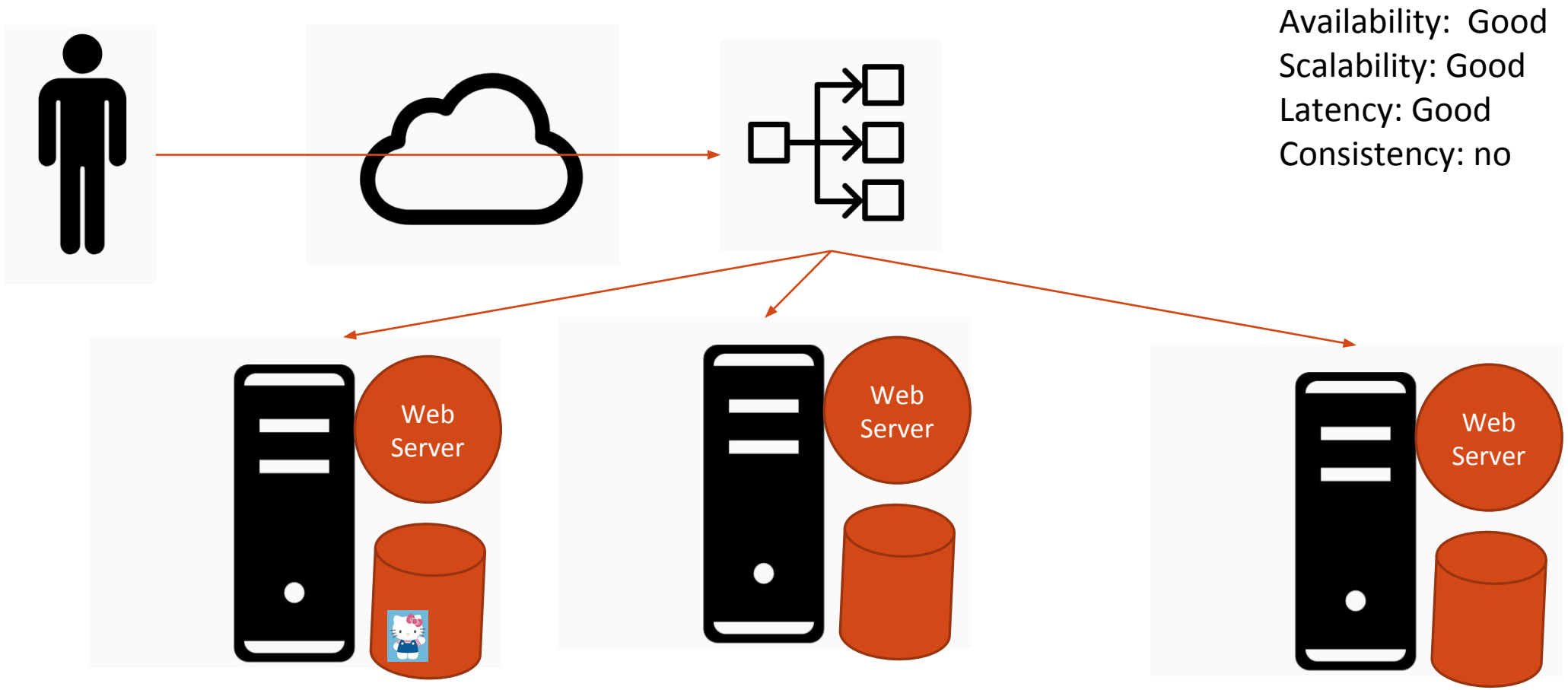
Availability: Is the service available?

Scalability: Can the service grow to support many users (dozen->google) and many photos (dozen->youtube)

Latency: How long for a user to get a response

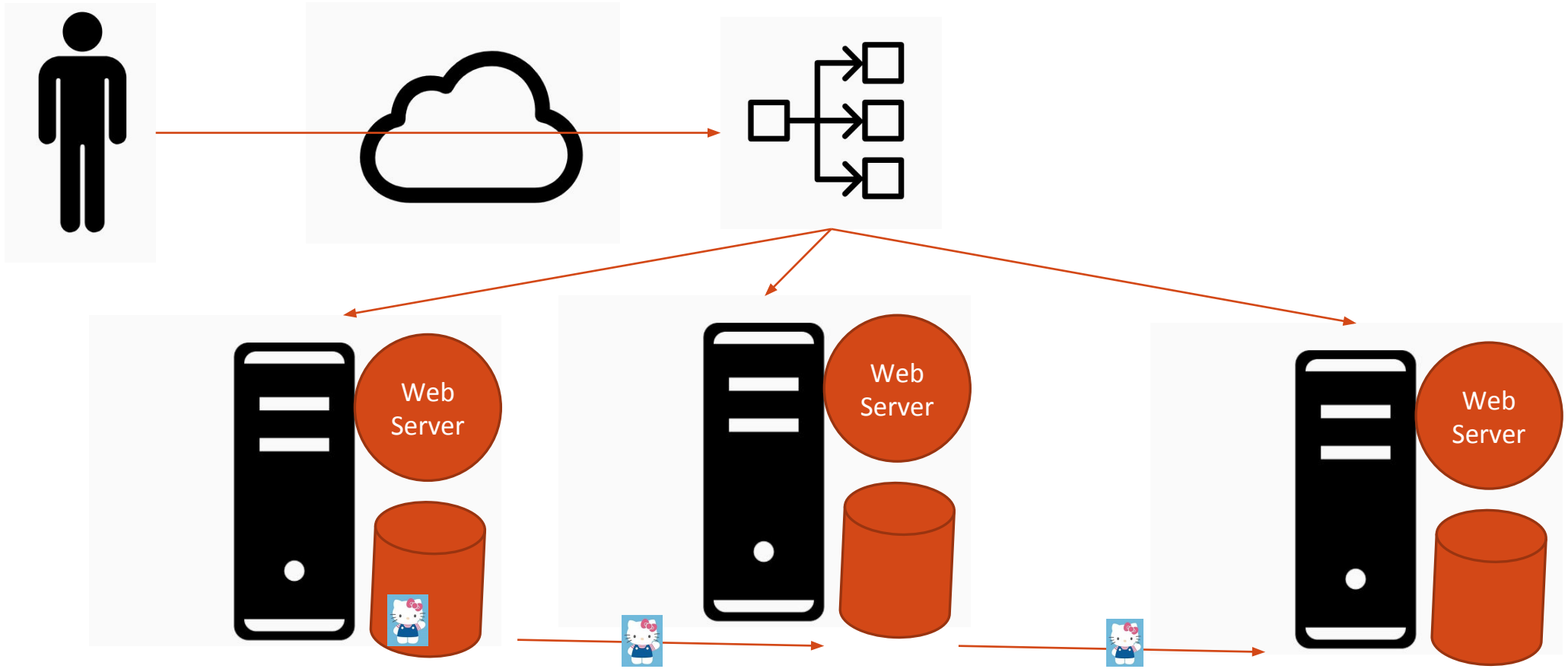
Consistency: Do I, and other users, see updated photos (and if not, how long does it take to get eventually consistent)

V2: Scale out w/Load balancer

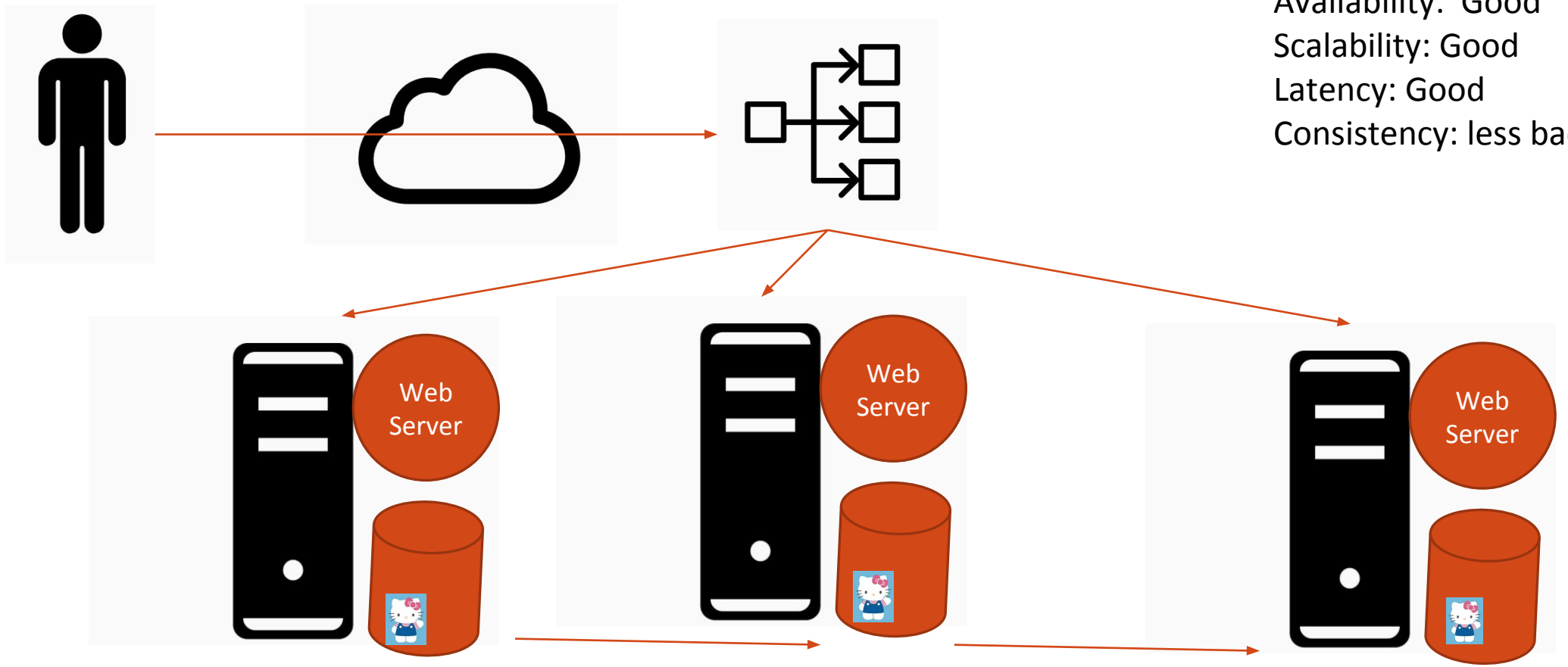


Availability: Good
Scalability: Good
Latency: Good
Consistency: no

V3: Add Backgroud Batch Job



V3: Background Batch Job

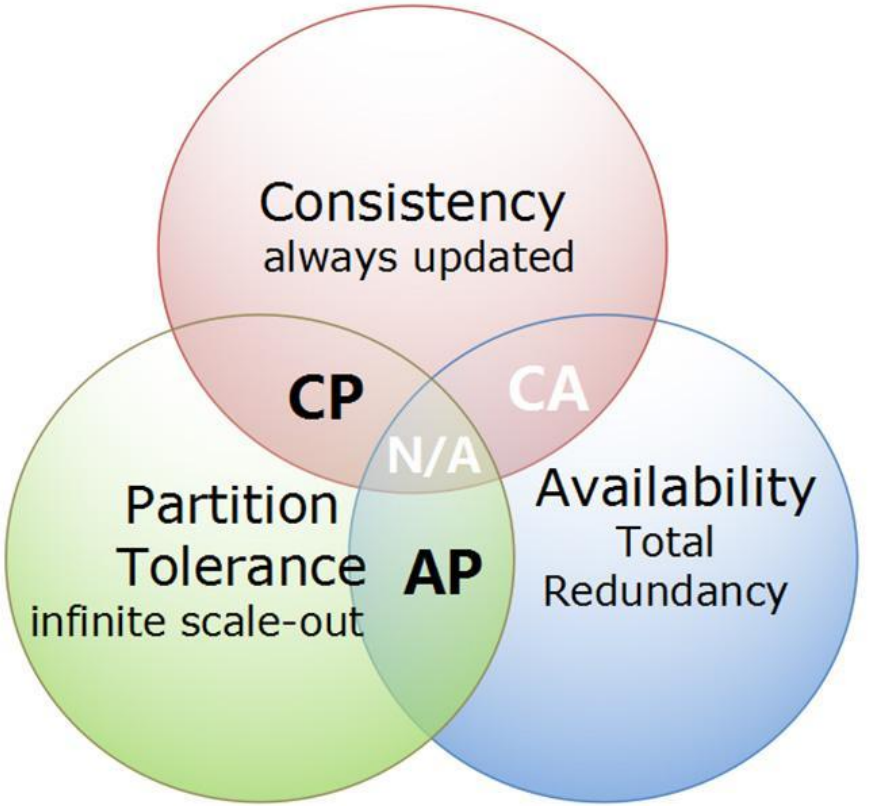


Availability: Good
Scalability: Good
Latency: Good
Consistency: less bad

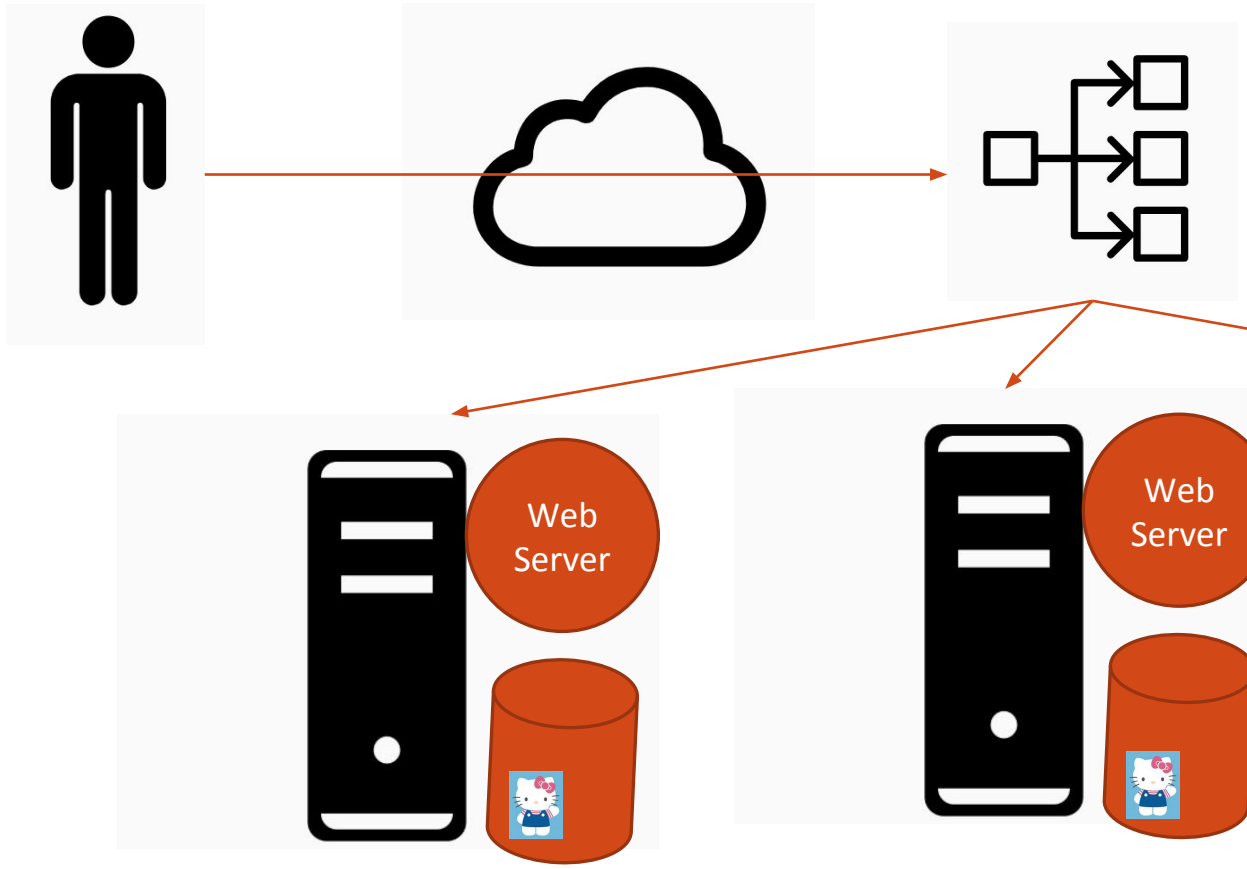
Highly Technical Observations

- Lumpy stuff is generally bad
- Background jobs cause lumpiness

CAP Theorem: You can have two



V4: Sync Writes (all servers)

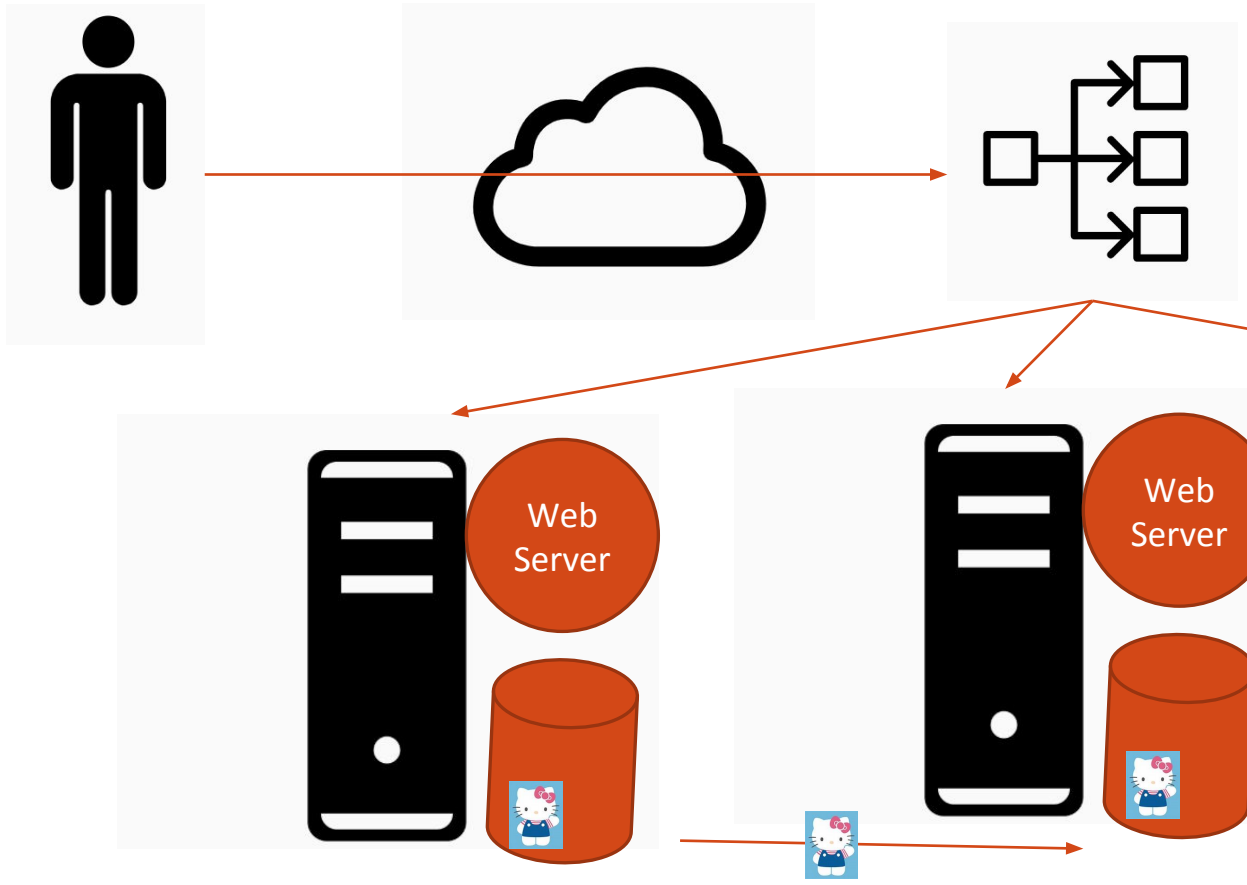


Read Availability : Good
Write Availability: Less Good
Read Latency: Good
Write Latency: Poor
Scalability: Terrible
Consistency: Good (we think)

V4 Issues

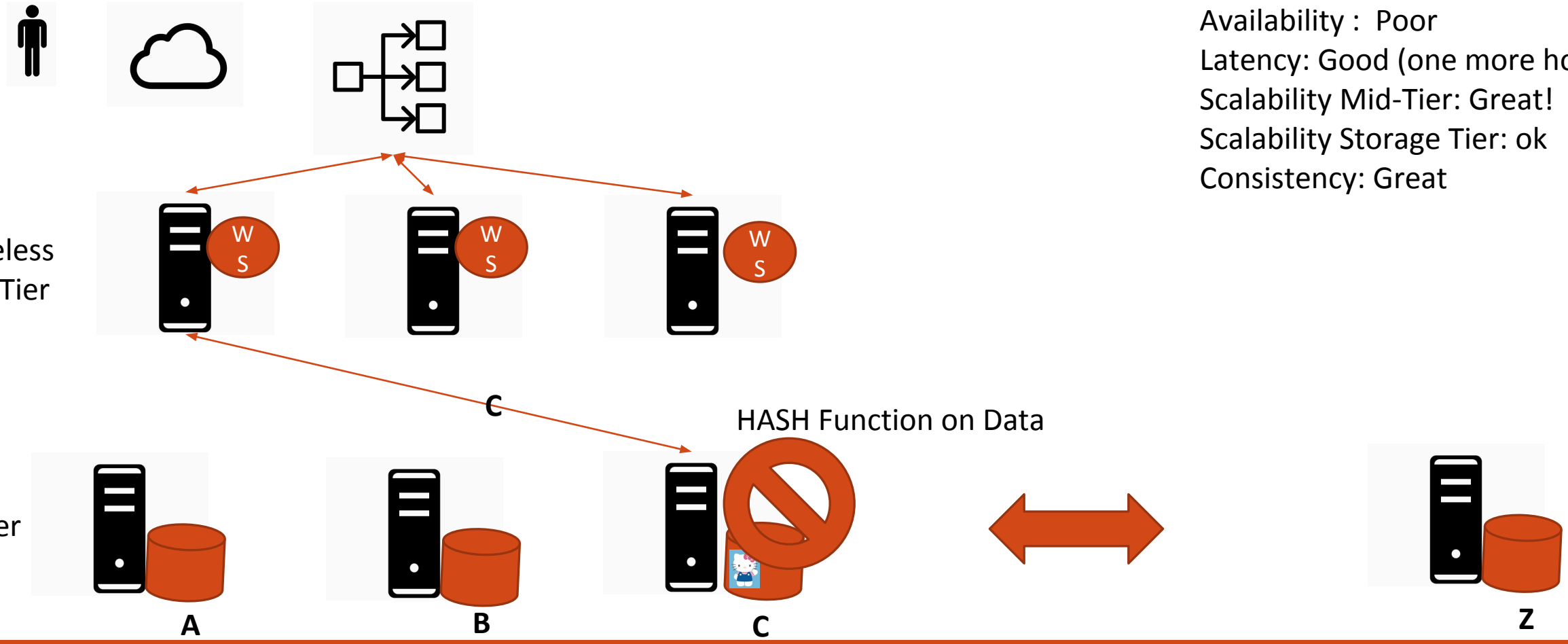
- Single machine is down during Write
  - Fail write?
    - Could be problems in roll back as another machine goes down
    - Not likely with 3 machines, how 100s? 1000s?
  - Ignore machine that is down?
- Bad Design
  - Diverges towards chaos
  - No self-healing

V5: Sync writes
Add Backgroud Batch Job



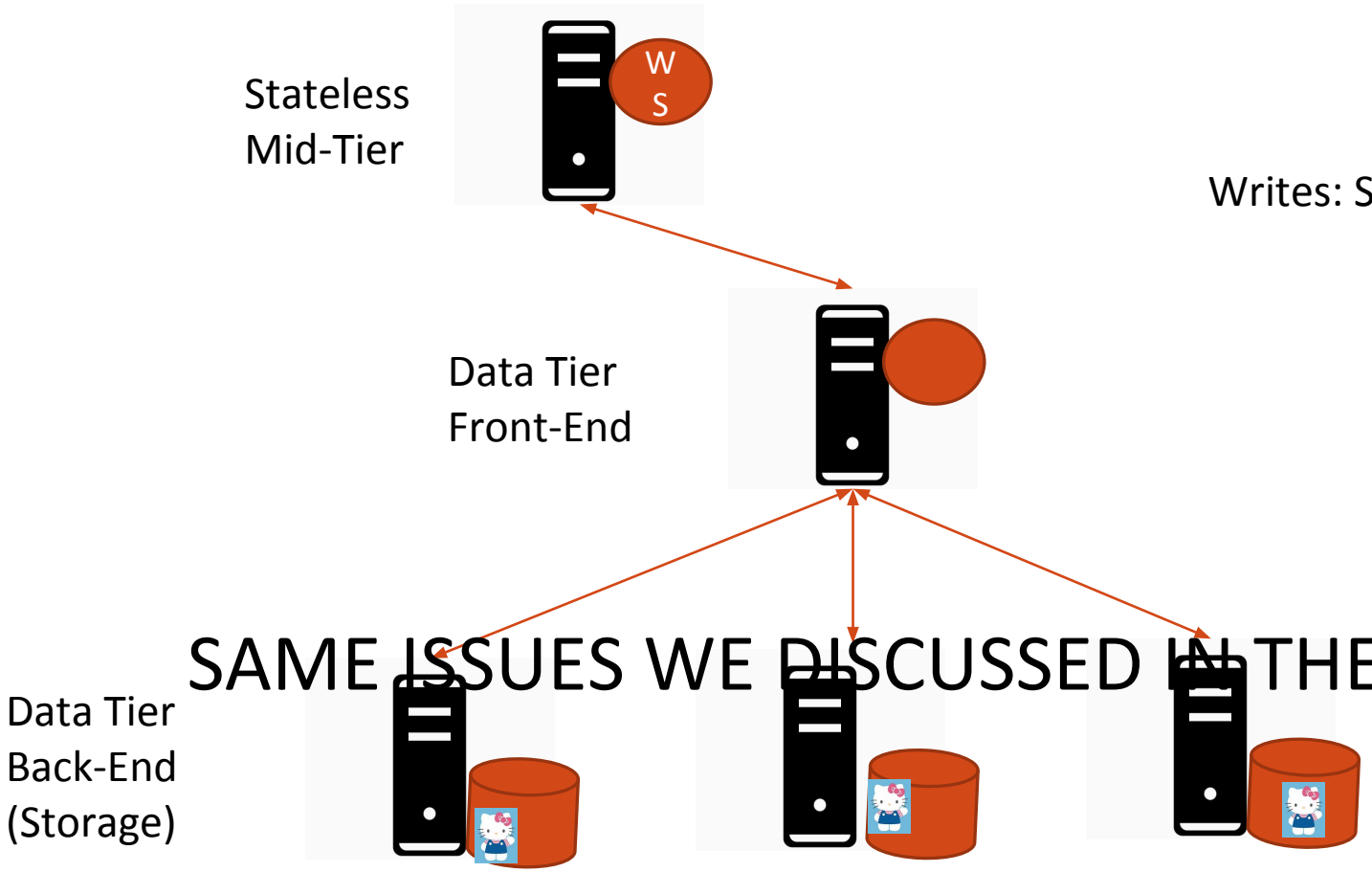
Read Availability : Good
Write Availability: Less Good
Read Latency: Good
Write Latency: Poor
Scalability: Terrible
Consistency: Actually ok

V6: Two Tier Arch

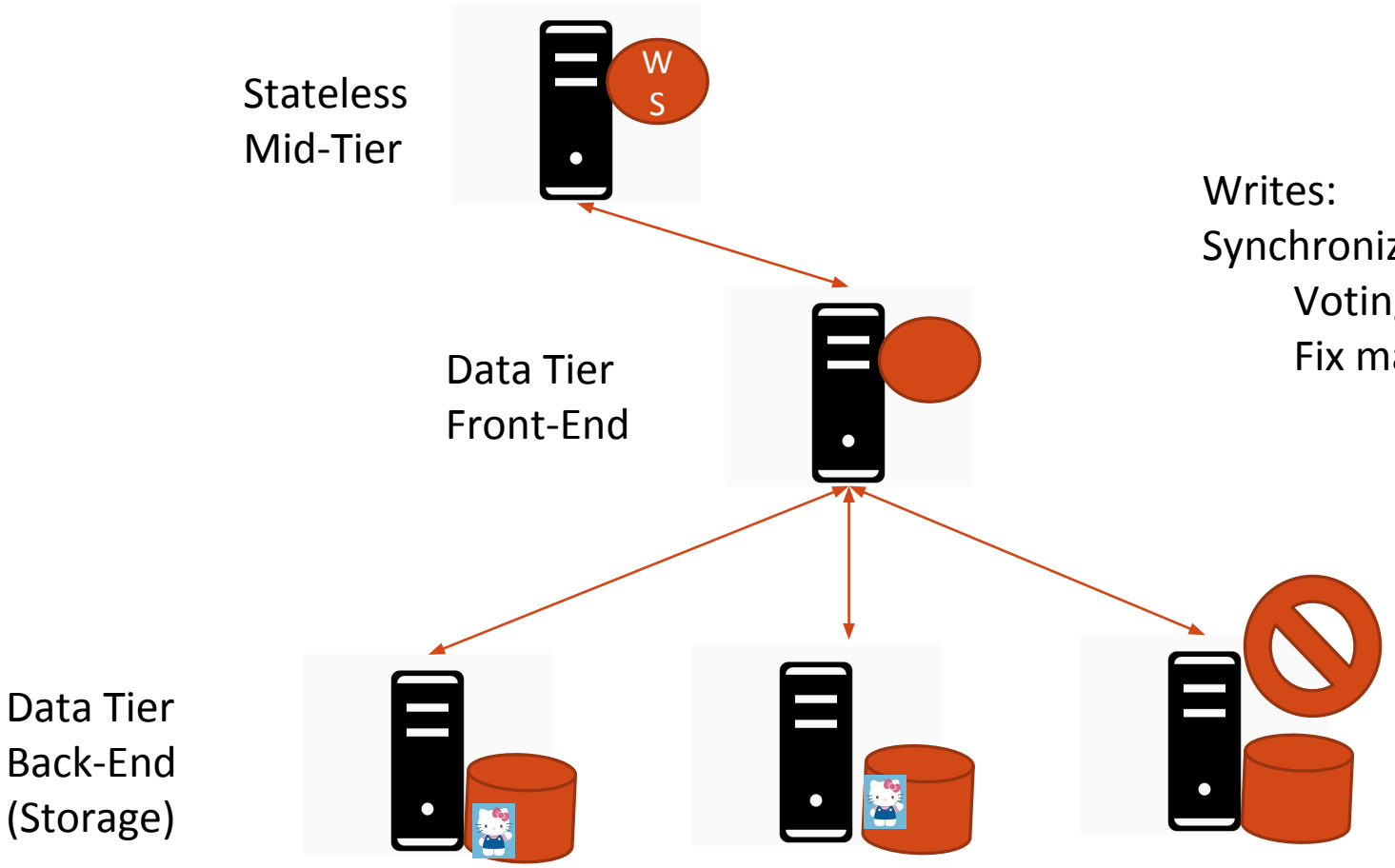


Availability : Poor
Latency: Good (one more hop)
Scalability Mid-Tier: Great!
Scalability Storage Tier: ok
Consistency: Great

V6b: Redundancy at storage tier



V6c: Redundancy at storage tier

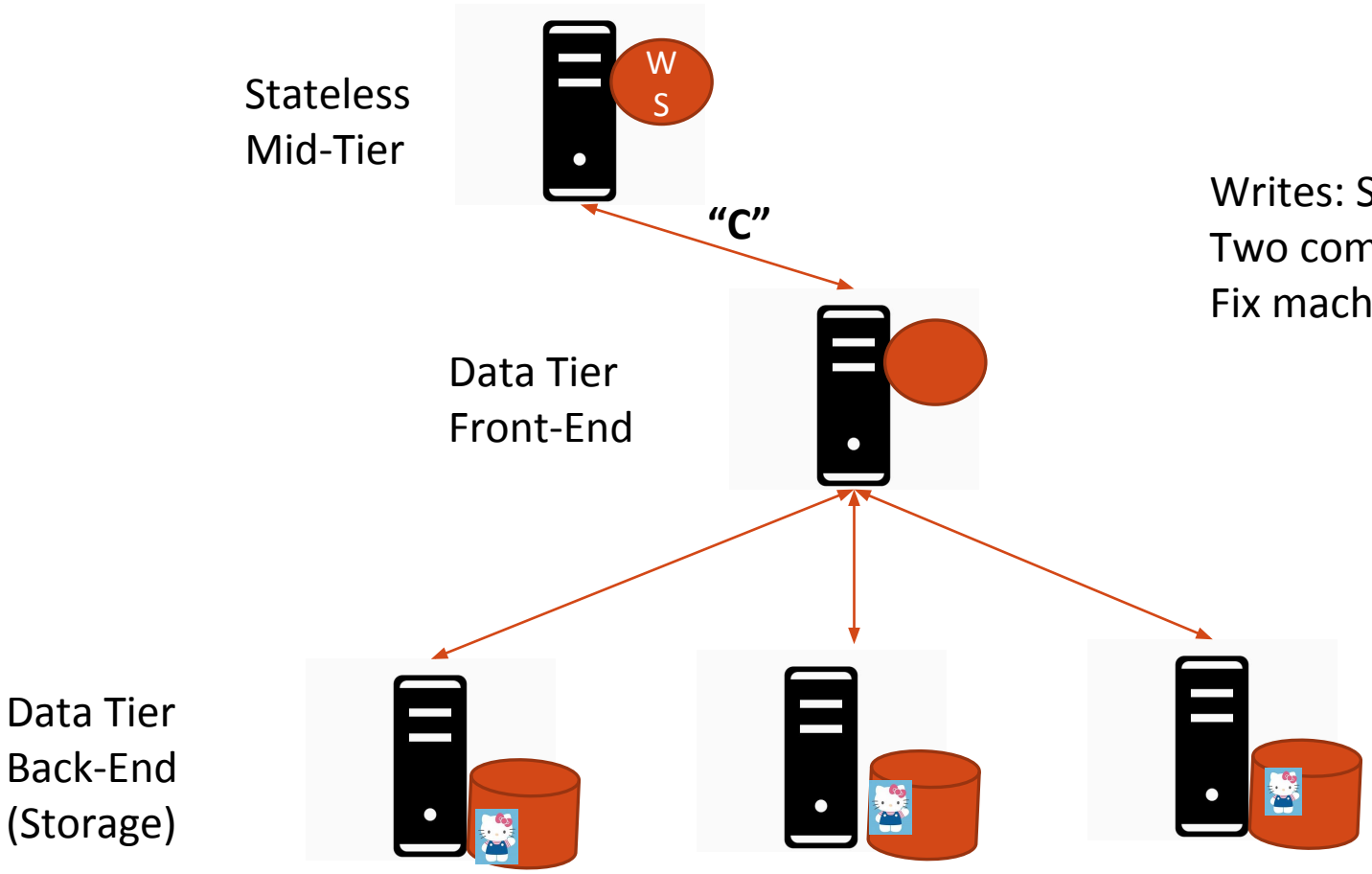


Writes:

Synchronize writes to all three, but two is fine

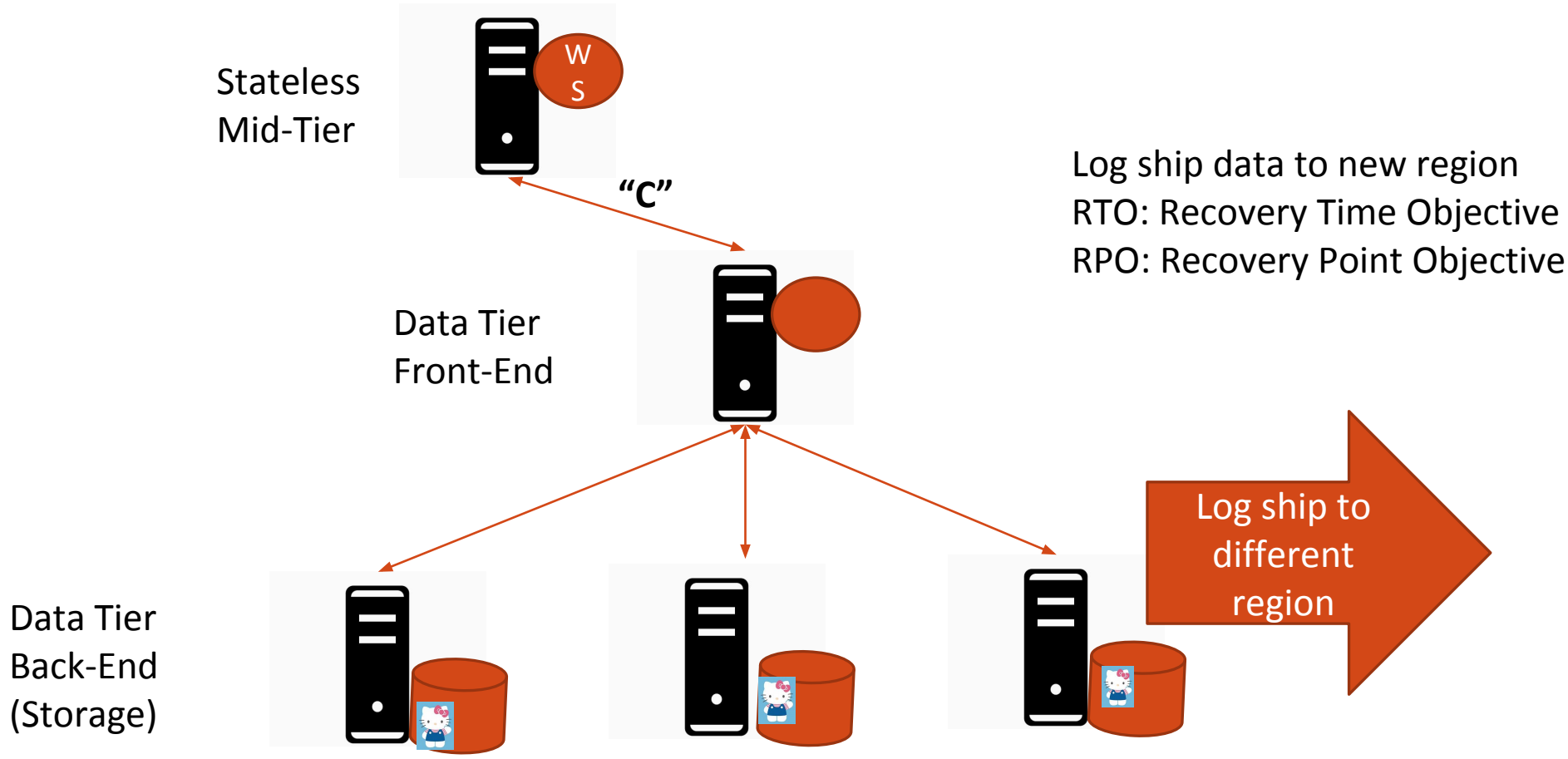
- Voting Mechanism
- Fix machine later

V6c: Redundancy at storage tier



Writes: Synchronize Writes to all three
Two compete is fine
Fix machine later

V6d: Redundancy at storage tier + Region



Log ship data to new region
RTO: Recovery Time Objective
RPO: Recovery Point Objective