Plan

Intro: Consistency

Recitation Questions

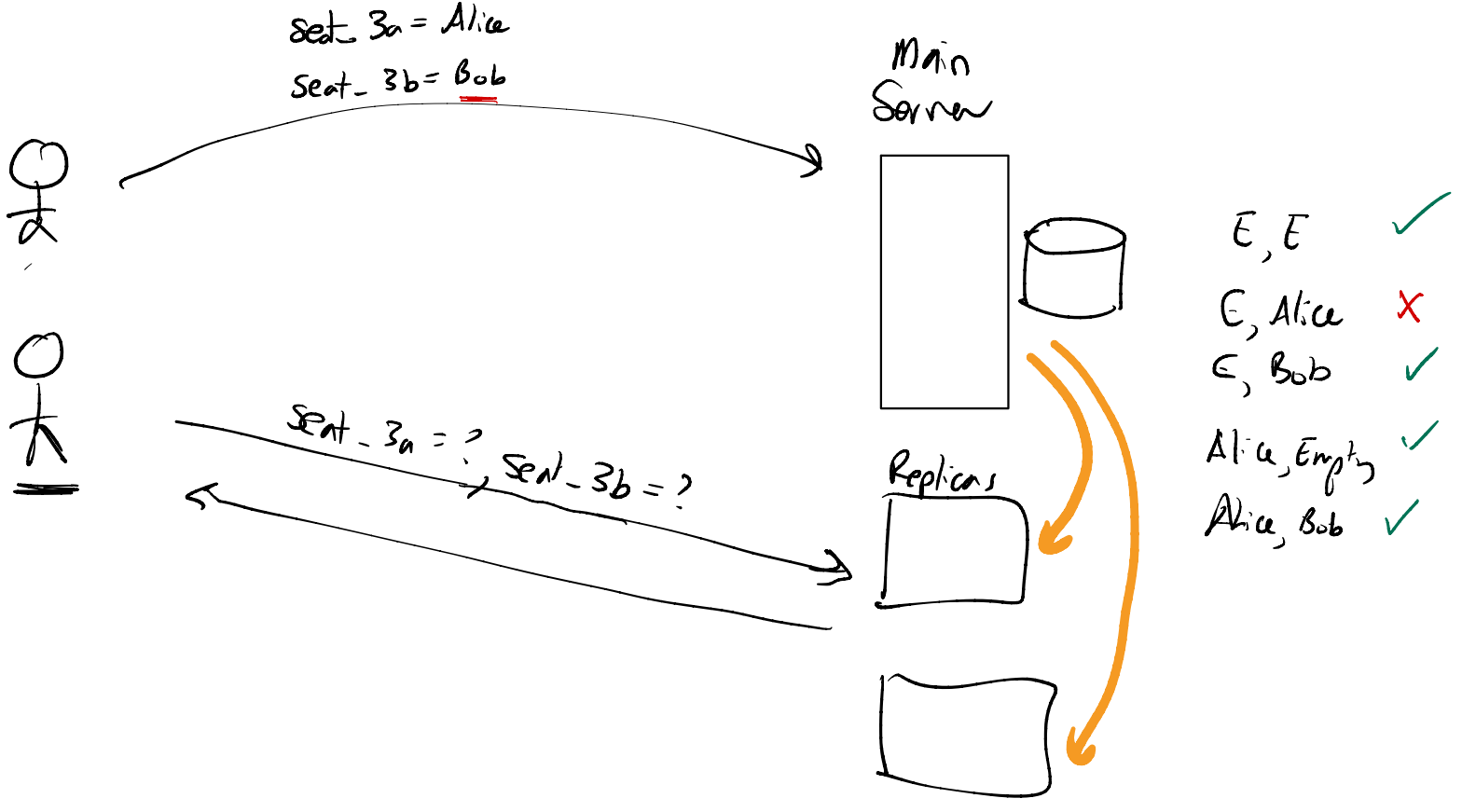
Types of Consistency

Consistency Game

Logistics

* Welcome back!
* Amir leading recitation on Thursday
* Hands-on due tomorrow
* Design project report due in two weeks

In simpler times...



Recitation Questions

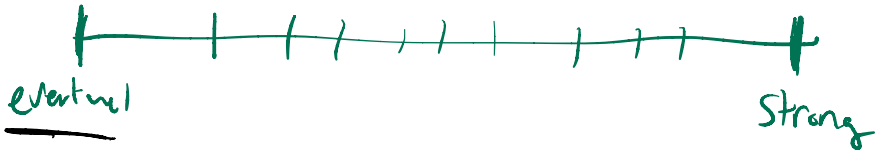
1. What is a consistency guarantee? What aspects of system does it affect?
   - ↳ R/W distributed ... who sees which writes
   - ↳ applies to multicore CPUs
2. How does a system designer choose a "good" guarantee?
   - ↳ Tradeoffs! Perf, avail, flexibility, cost
   - → Min guarantees necessary.

   Dyamo DB
   - → Eventual $
   - → Strong $$
3. Why does the choice matter?
   - ↳ Apps have diffent needs

Amazon DB

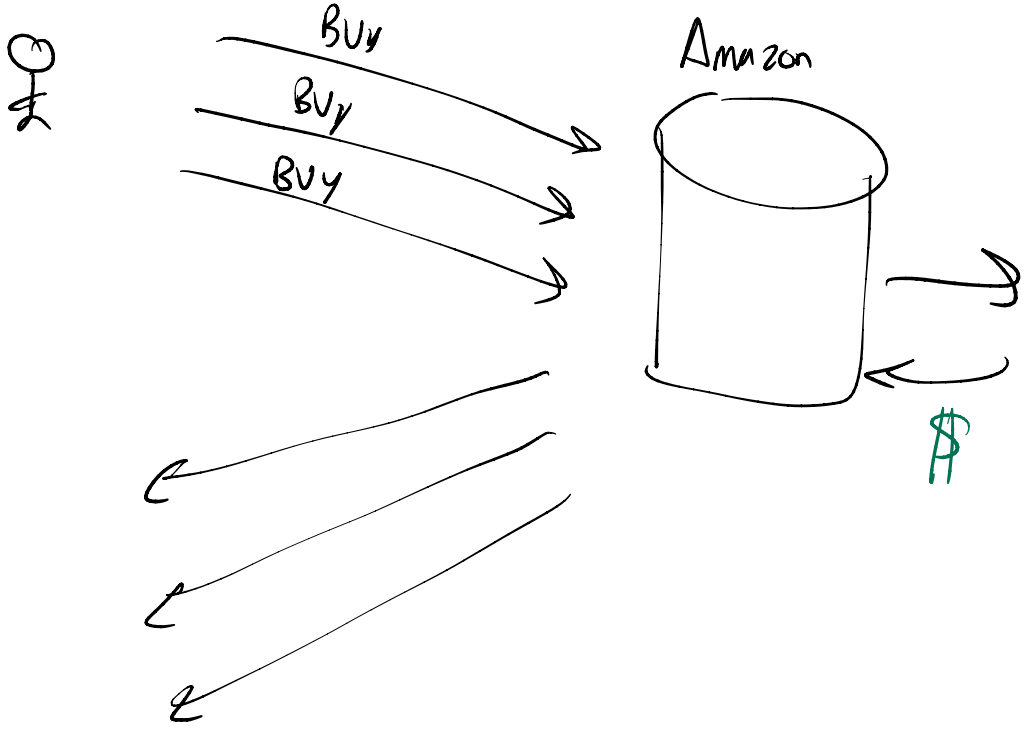
(1) ↳ Why does Amazon only offer 2 options?

↳ simplicity {



(2) Why is that good enough?

↳ Some delay is okay

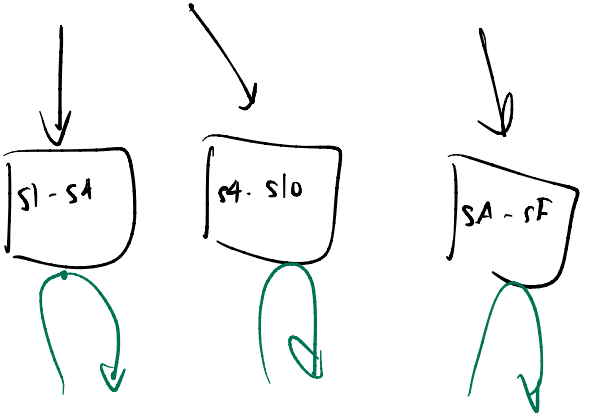

Buy
Buy
Buy
Amazon
$

Consistency Game

1) Eventual
2) Strong

write → s3 = 7

s5 = 2



"Sharding"

Plan

Intro: Consistency

Recitation Questions

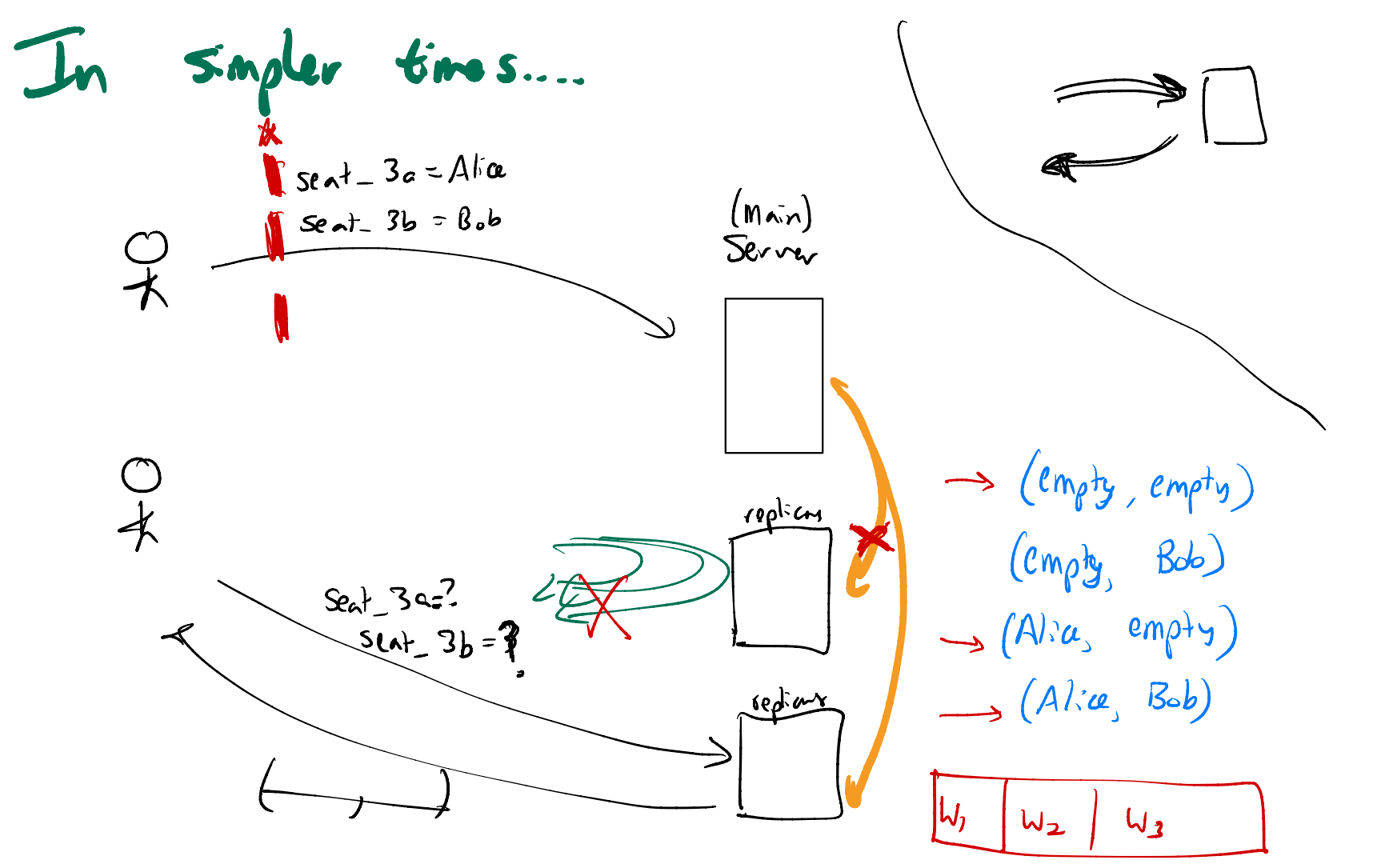
Types of Consistency

Consistency Game

Logistics

* Welcome back!
* Amir leading recitation on Thursday
* Hands-on due tomorrow
* Design project report due in two weeks

In simpler times....



Recitation Questions

1. What is a consistency guarantee?
   - promise about how up-to-date data will be
   - distributed system

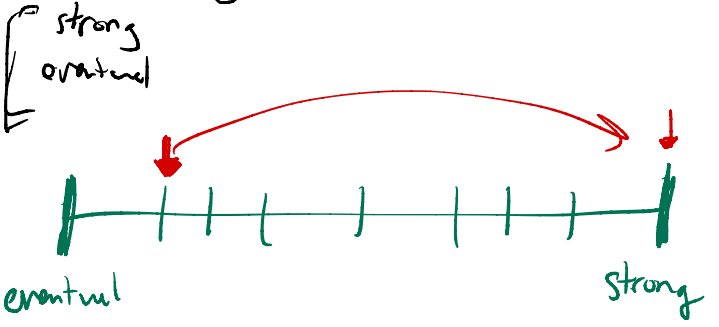
2. How does a system designer choose a consistency guarantee?
   - app need vs. cost
   - DynamoDB
     - Strong $$
     - Eventual $

3. Why does the spice of a consistency guarantee matter?
   - correctness! vs cost/perf

→ Why does Amazon only offer 2 guarantees?

↳ complicated for Amazon

↳ complicated for app developer



→ Why is eventual consistency enough ever?

↳ "less than 1 second"

↳

Consistency Game

WRITER → send msgs to Amir (private room)

s3 = 7

s4 = 2

sC = 5

MAIN (Amir)

REPLICAS

READERS