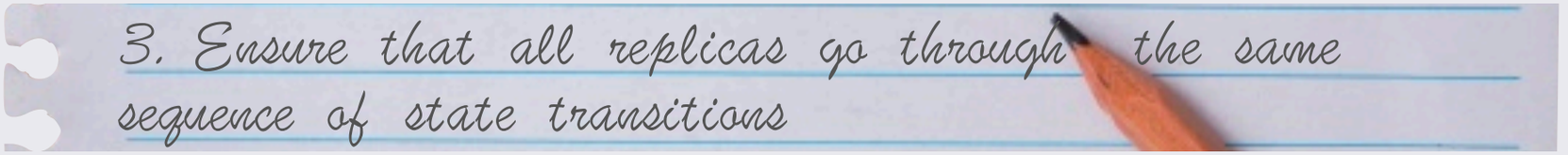
EECS 591
Distributed Systems

Manos Kapritsos
Fall 2021

Using (multi)Paxos to implement State Machine Replication

The original Paxos algorithm achieves agreement on **one** value

SMR required replicas to agree on the **sequence** of commands that will be executed

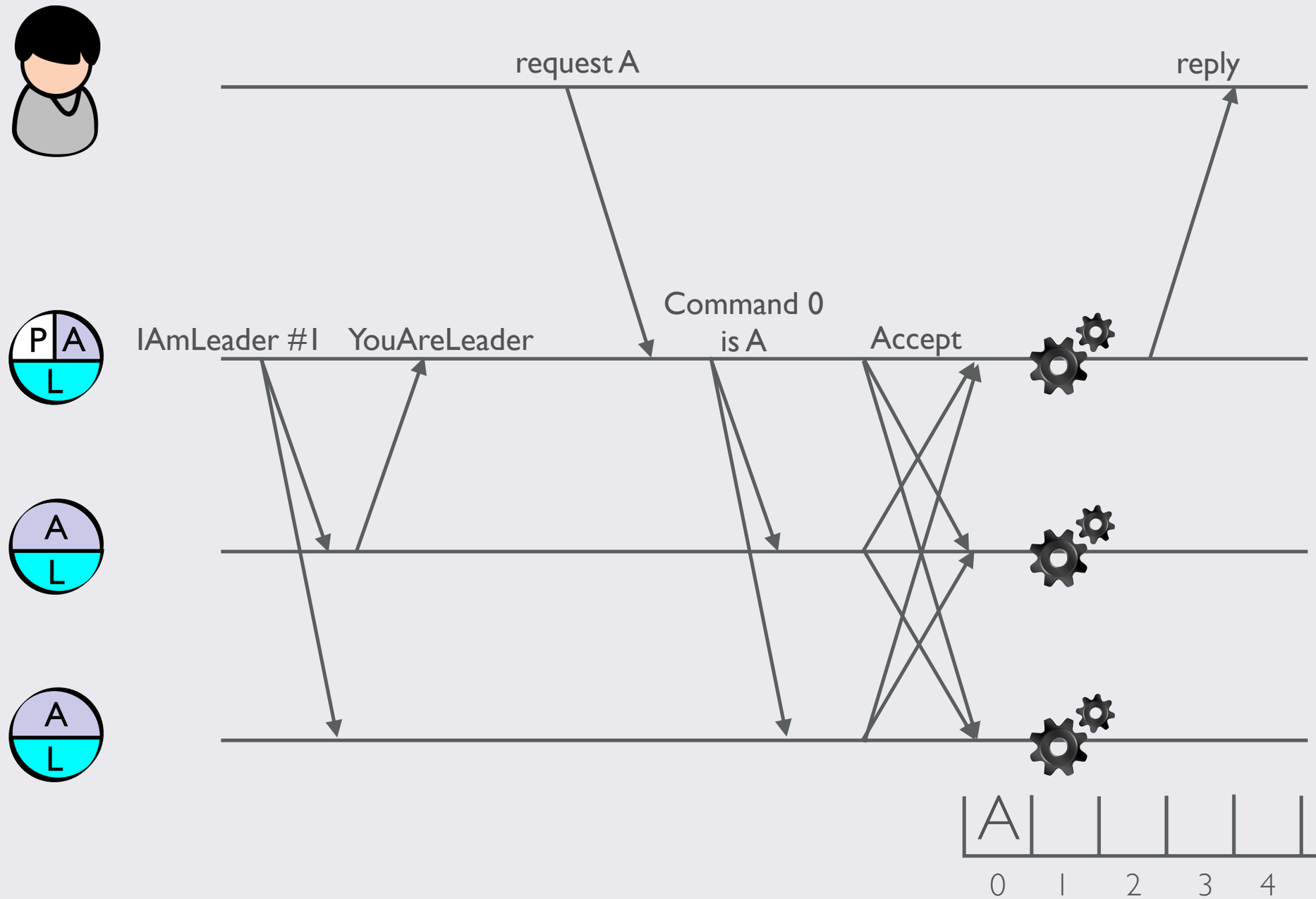


MultiPaxos: Run an instance of Paxos for each slot in the sequence

Important: we don't need to run phase 1 (election) every time!

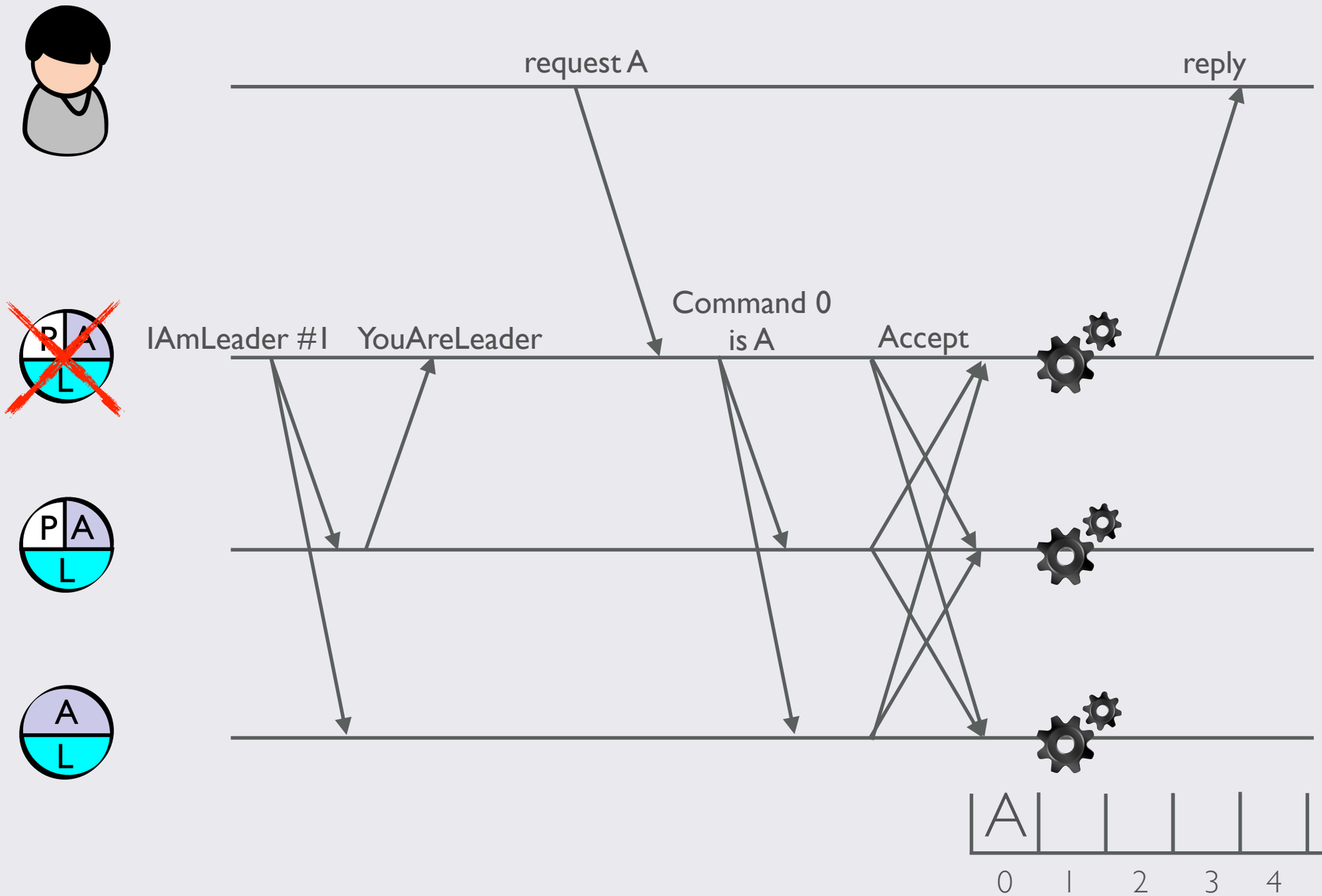
PAXOS/SMR IN REAL LIFE

Proposers, acceptors and learners
are all collocated on $2f + 1$ replicas



PAXOS/SMR IN REAL LIFE

Proposers, acceptors and learners
are all collocated on $2f + 1$ replicas



Administrivia

Midterm

- Wednesday 10/27, 12-1:20pm, during class
  - You can use any material listed on the course website

No class the next two Mondays

- Monday 10/18, UM study day
- Monday 10/15, conflict with SOSP workshops

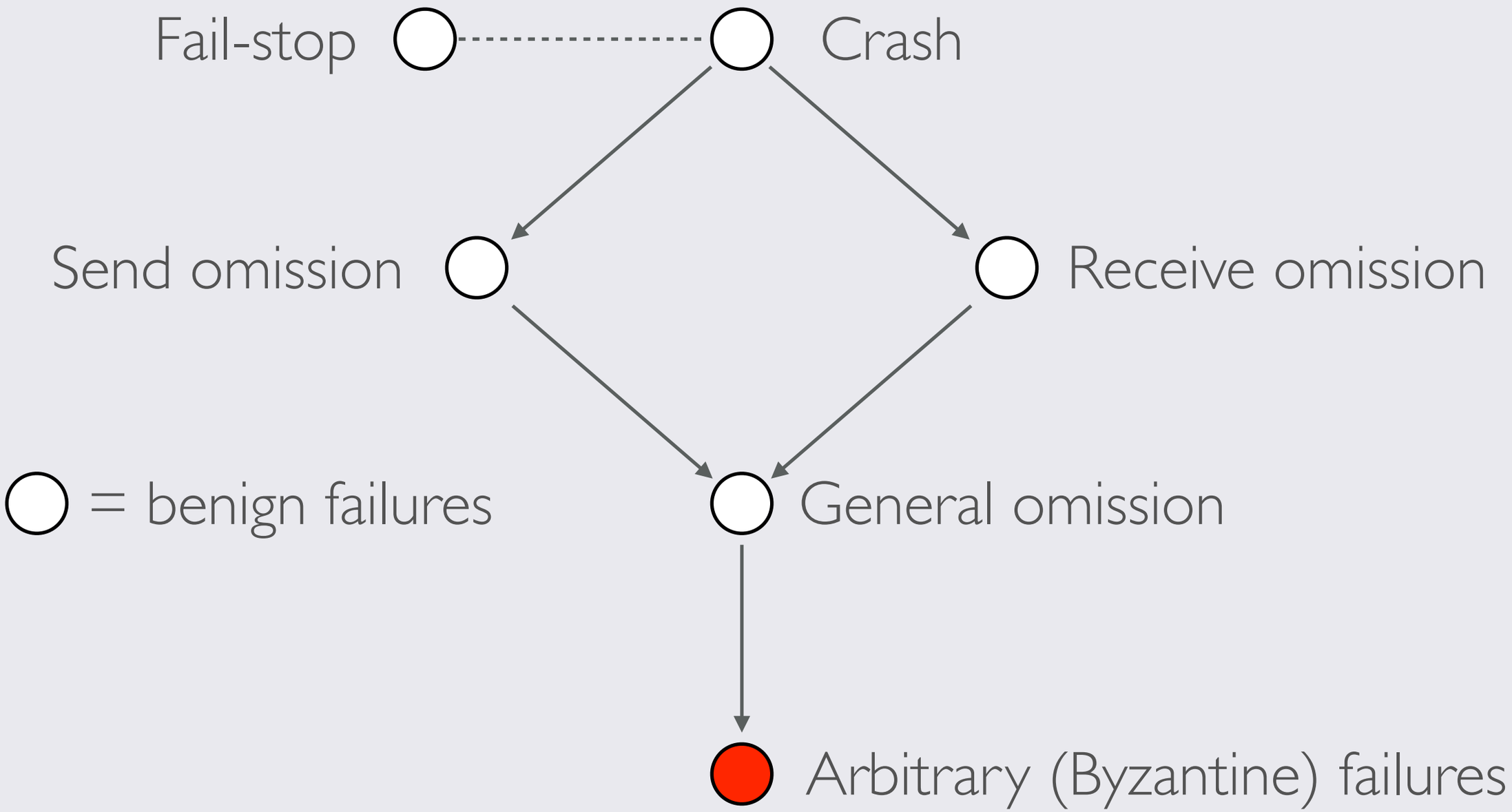
Research part

- Starts after midterm, Monday 11/1 with Fast Paxos and Flexible Paxos
  - You should read both papers and you can review either one

Byzantine Fault Tolerance

Slides by Lorenzo Alvisi

A HIERARCHY OF FAILURE MODELS



What are Byzantine Failures

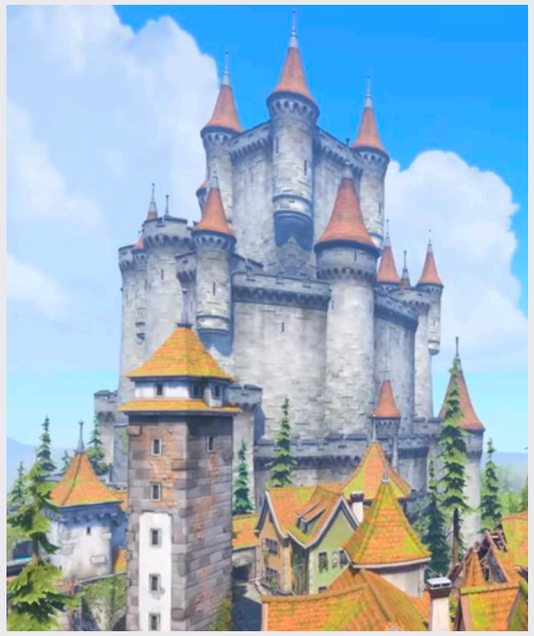
The short answer: they can be ***anything!***

(they can even be crash/omission failures)

Examples of commission failures

- A bit flip in memory
  - Manufacturing defect
  - Alpha particles
- Network card malfunction
- Intentional behavior
  - Rational node: trying to game the system for personal gain
  - Malicious node: trying to bring the system down

The Byzantine Generals



- Synchronous communication
- One general may be a traitor

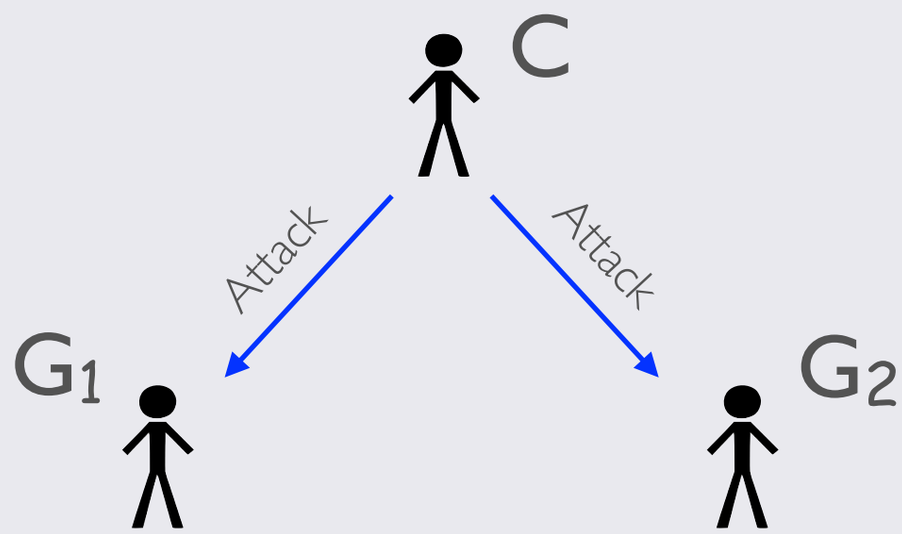
The Byzantine Generals

- Synchronous communication
- One general may be a traitor
- One of the generals is the commander **C**
  - The commander decides **Attack** or **Retreat**

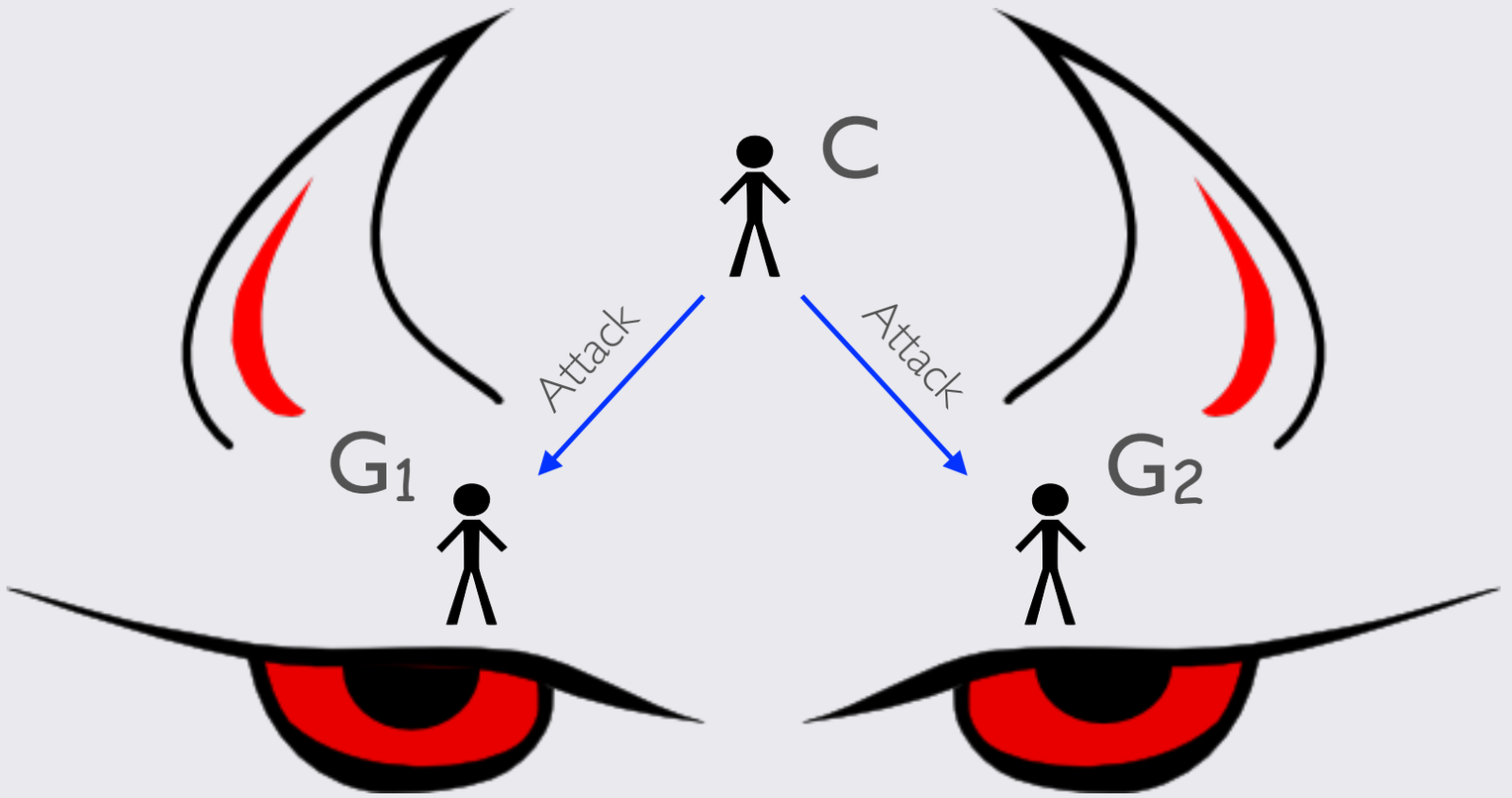
Goals

1. If **C** is trustworthy, every trustworthy general must follow **C**'s orders
2. Every trustworthy general must follow the same battle plan

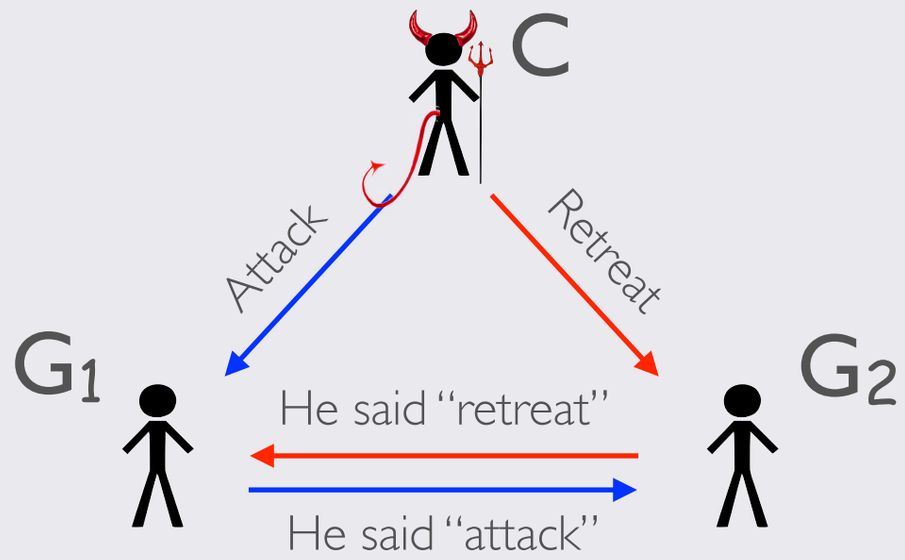
Remember when things were simpler?



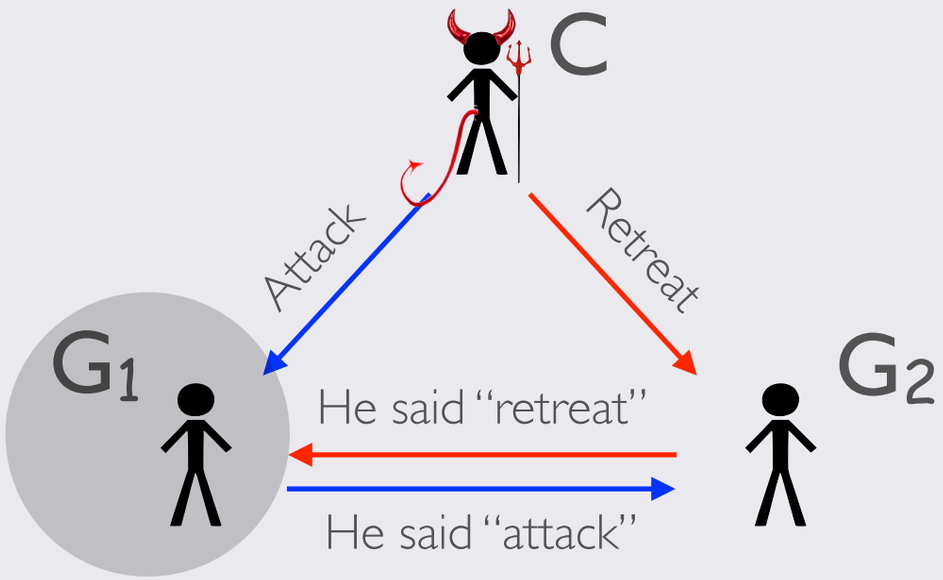
YOU CAN'T TRUST ANYONE THESE DAYS…

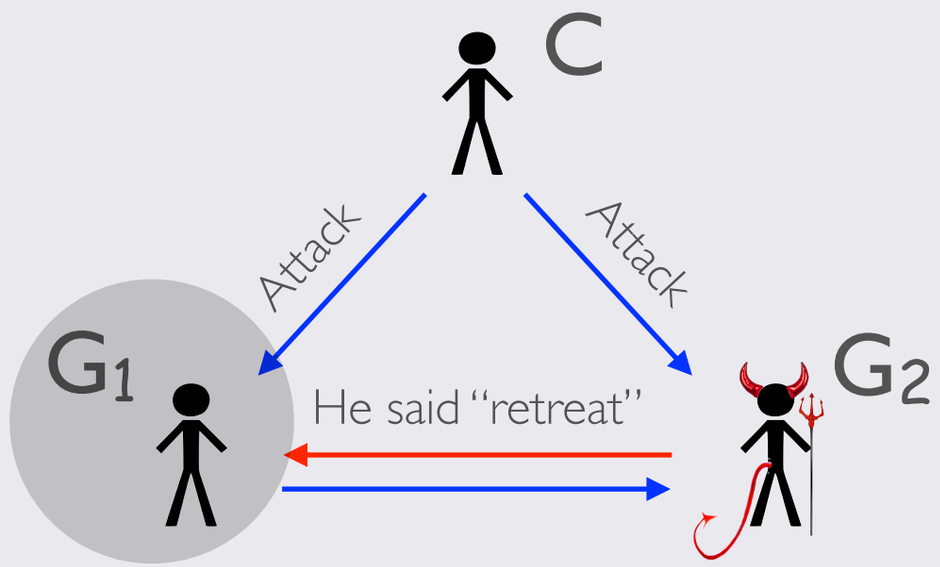


YOU CAN'T TRUST ANYONE THESE DAYS…

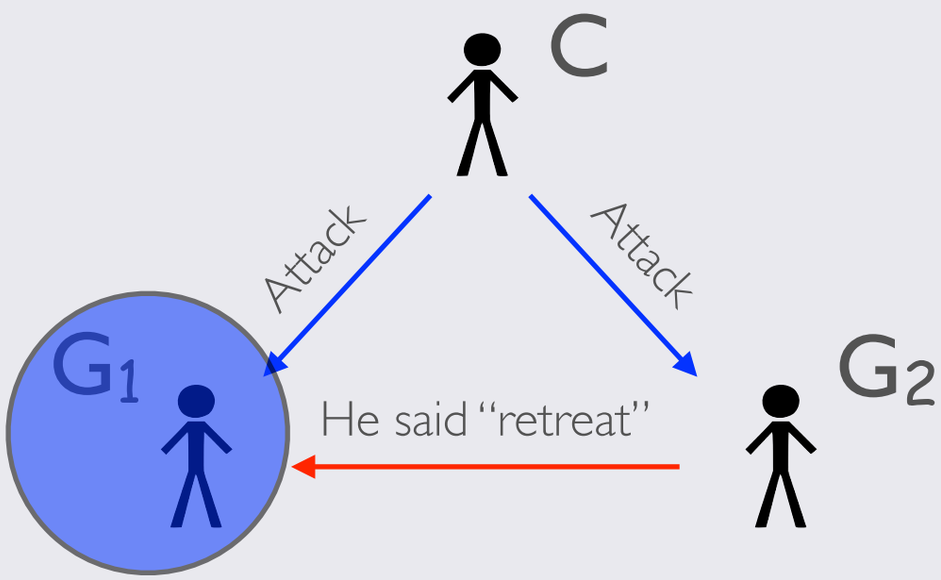


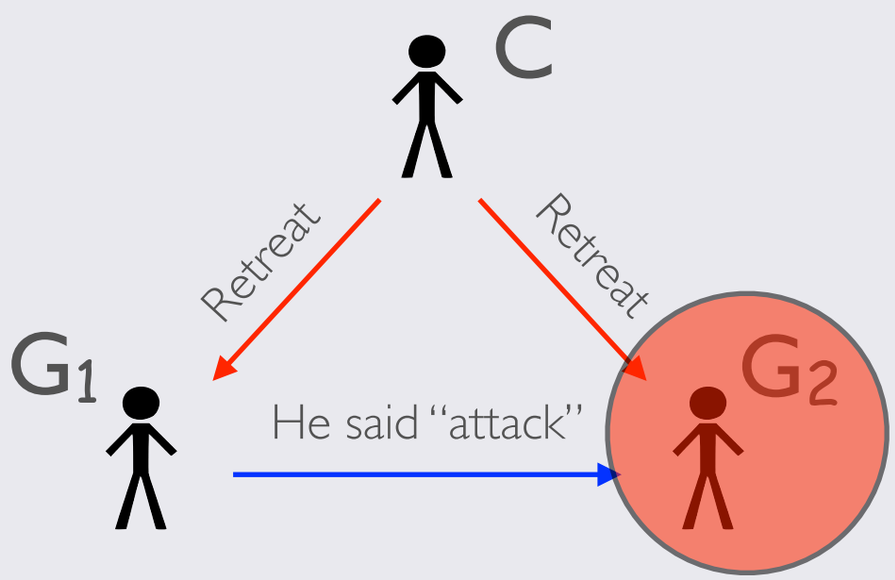
YOU CAN'T TRUST ANYONE THESE DAYS…

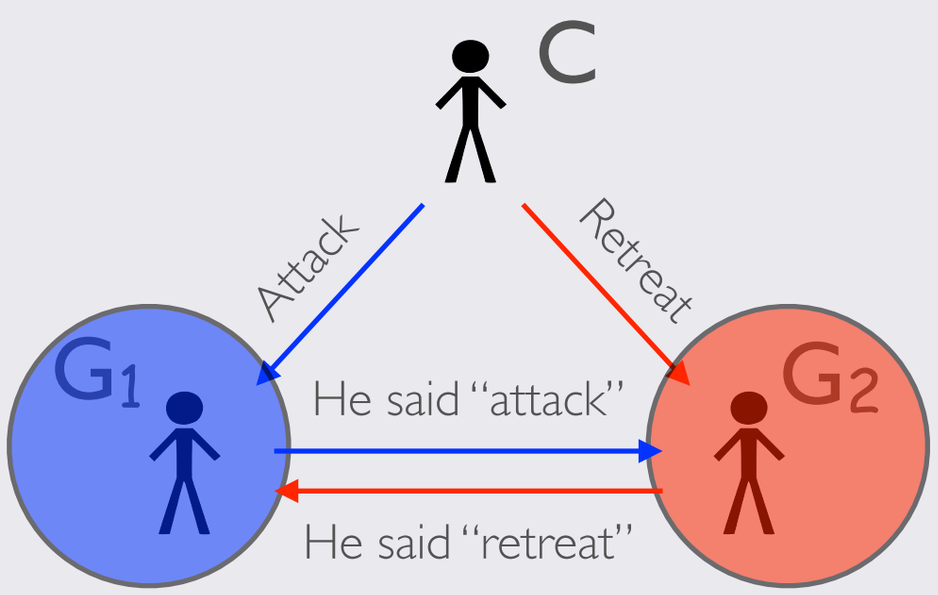




"BUT THEY WERE ALL OF THEM DECEIVED…"







A LOWER BOUND

Theorem

There is no algorithm that solves TRB for Byzantine failures if $n \leq 3f$

Lamport, Shostak and Pease, The Byzantine Generals Problem, 1982

PBFT: A Byzantine Renaissance

Practical Byzantine Fault Tolerance

(Castro, Liskov 1999-2000)

- First practical protocol for **asynchronous BFT replication**
- Like Paxos, PBFT is safe all the time, and live during periods of synchrony

Barbara Liskov

Turing Award 2008

The setup

System model

- Asynchronous system
- Unreliable channels

Crypto

- Public/private key pairs
- Signatures
- Collision-resistant hashes

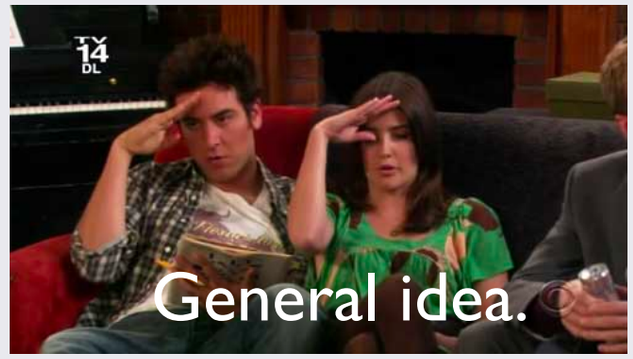
Service

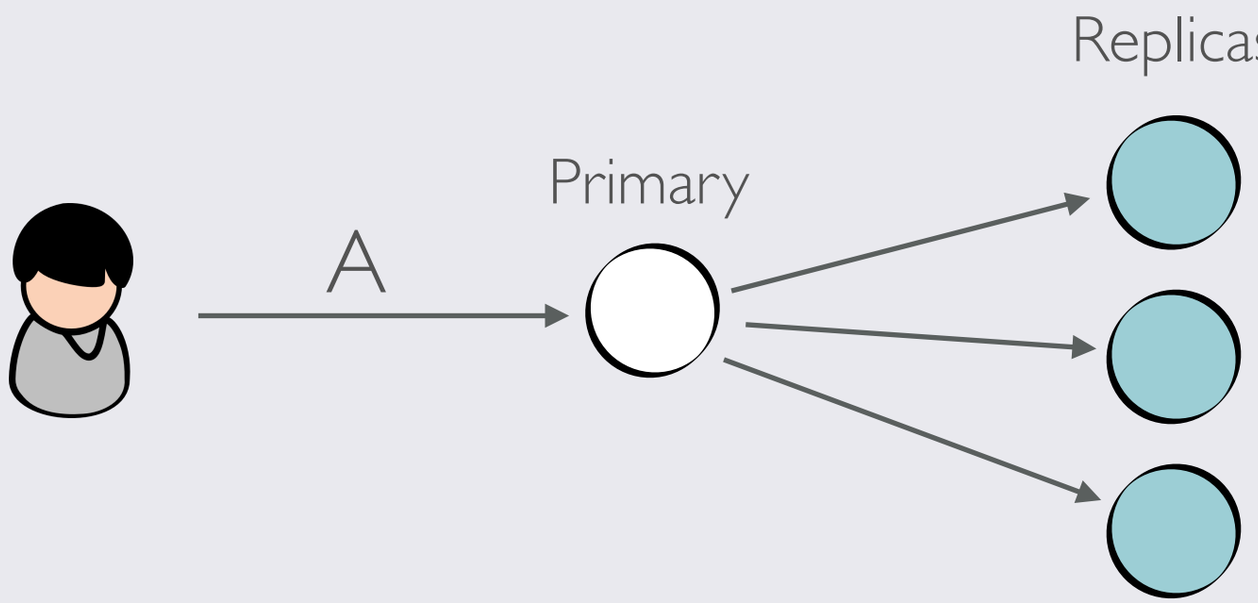
- Byzantine clients
- Up to f Byzantine servers
- $n = 3f + 1$ total servers

System goals

- Always safe
- Live during periods of synchrony

THE GENERAL IDEA





- One primary, 3f replicas
- Execution proceeds as a sequence of **views**
  - A view is a configuration with a well-defined primary
- Client sends signed commands to primary of current view
- Primary assigns sequence number to client's command
- Primary is responsible for the command eventually being decided

What could possibly go wrong!?

- The primary could be faulty!
  - could ignore commands, assign same sequence number to different requests, skip sequence numbers, etc.
  - ☑ Backups monitor primary's behavior and trigger **view changes** to replace a faulty primary

- Replicas could be faulty!
  - could incorrectly forward commands received by a correct primary
  - ☑ any single request may be misleading; need to rely on **quorums** of requests

  - could send incorrect responses to the client
  - ☑ client waits for $f + 1$ matching responses before accepting

Certificates

Protocol steps are justified by **certificates**

- Sets (quorums) of signed messages from distinct replicas proving that a property holds

Certificates are of size at least $2f + 1$

- Any two quorums intersect in at least **one correct** replica (for safety)
- There is always a quorum of correct replicas (for liveness)