Aegean: Replication beyond the client-server model

Remzi Can Aksoy, **Manos Kapritsos**
University of Michigan

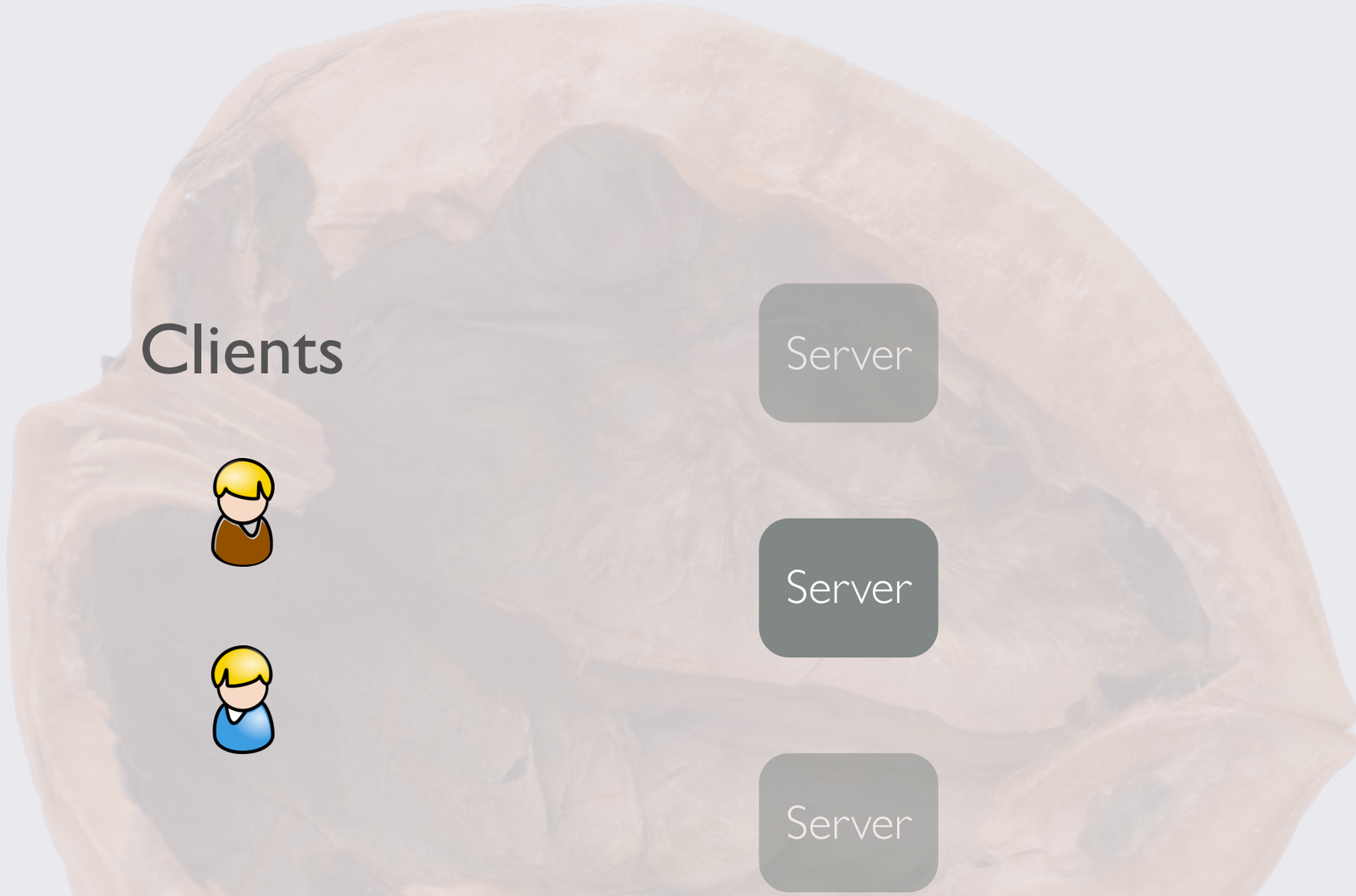
REPLICATION IN A NUTSHELL

Primary-backup protocols

*Paxos protocols

BFT protocols

Replication in a nutshell



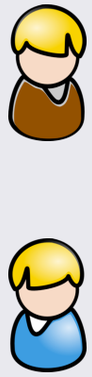
Goal of replication: provide the abstraction of a *single correct server*

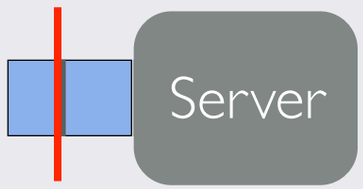
Beyond client-server

Clients

Expedia

Airline/Hotel/Bank





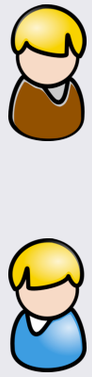
Backend server

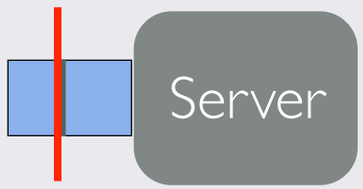
Beyond client-server

Clients

Expedia

Airline/Hotel/Bank



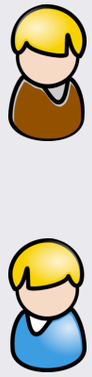


Beyond client-server

Clients

Expedia

Airline/Hotel/Bank



Server

Backend server

BEYOND CLIENT-SERVER

Clients

Expedia

Airline/Hotel/Bank

Goal of replication: provide the abstraction of a *single correct server*

Aegean

and mitigating

Identifying the impact of service interactions on the design of replication protocols

Outline

Problem statement

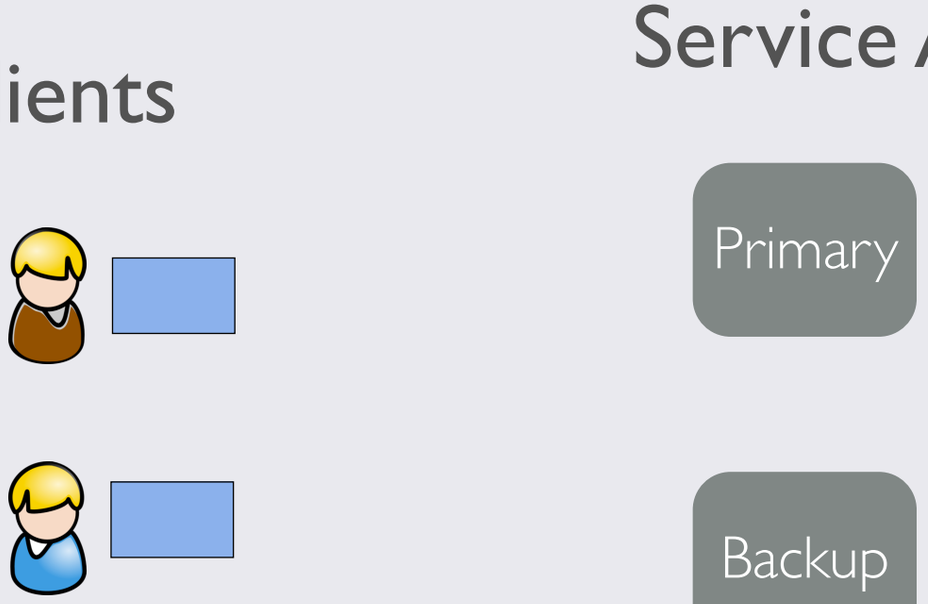
Correctness: the threat of speculation

Performance: the price of sequential execution

Evaluation of Aegean

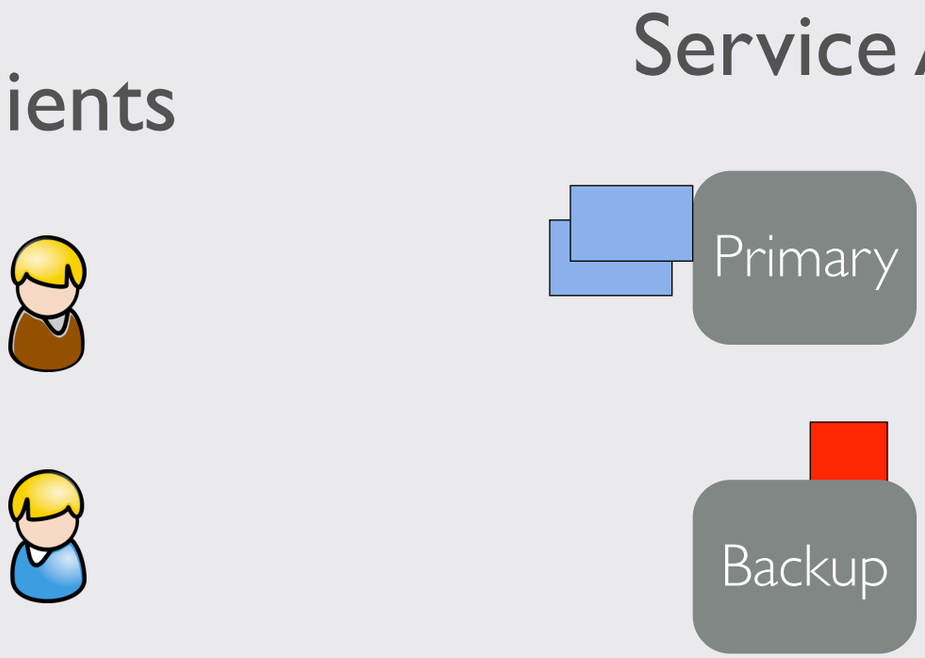
How interactions break our protocols

Example: Primary-backup



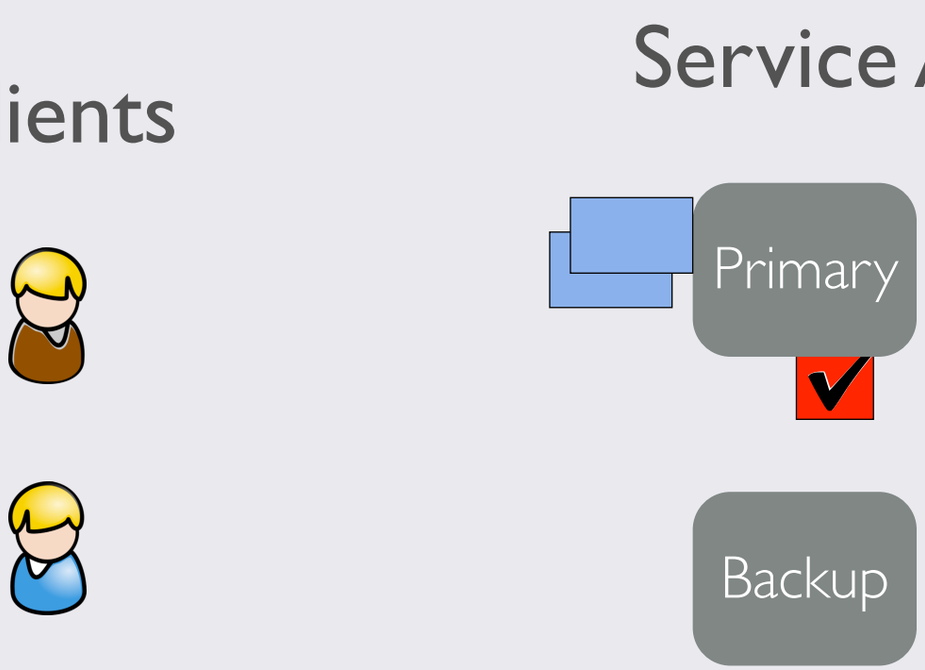
How interactions break our protocols

Example: Primary-backup



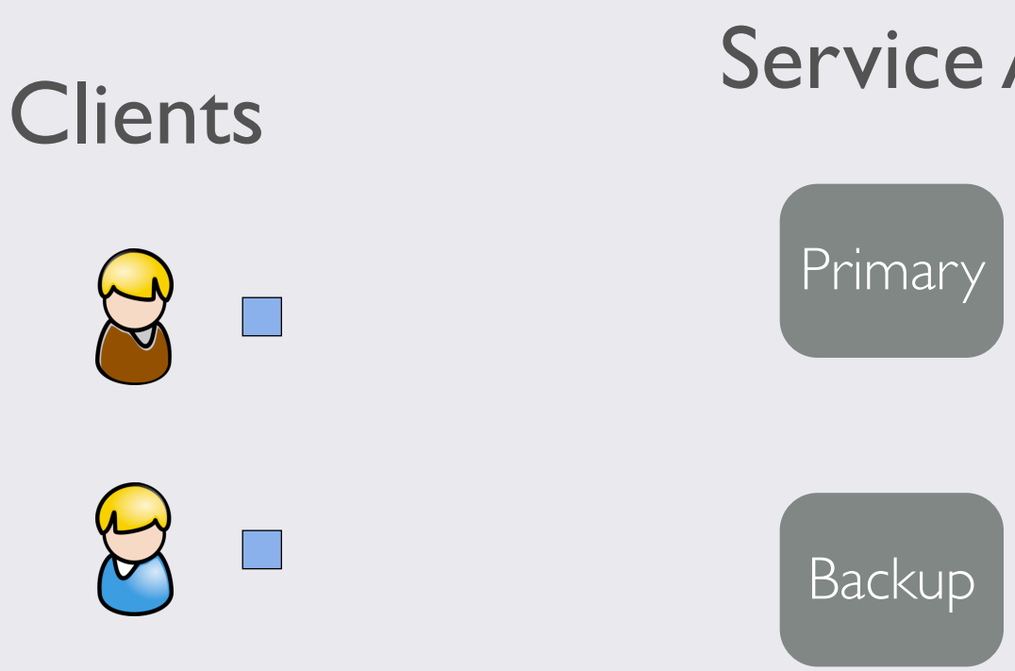
How interactions break our protocols

Example: Primary-backup



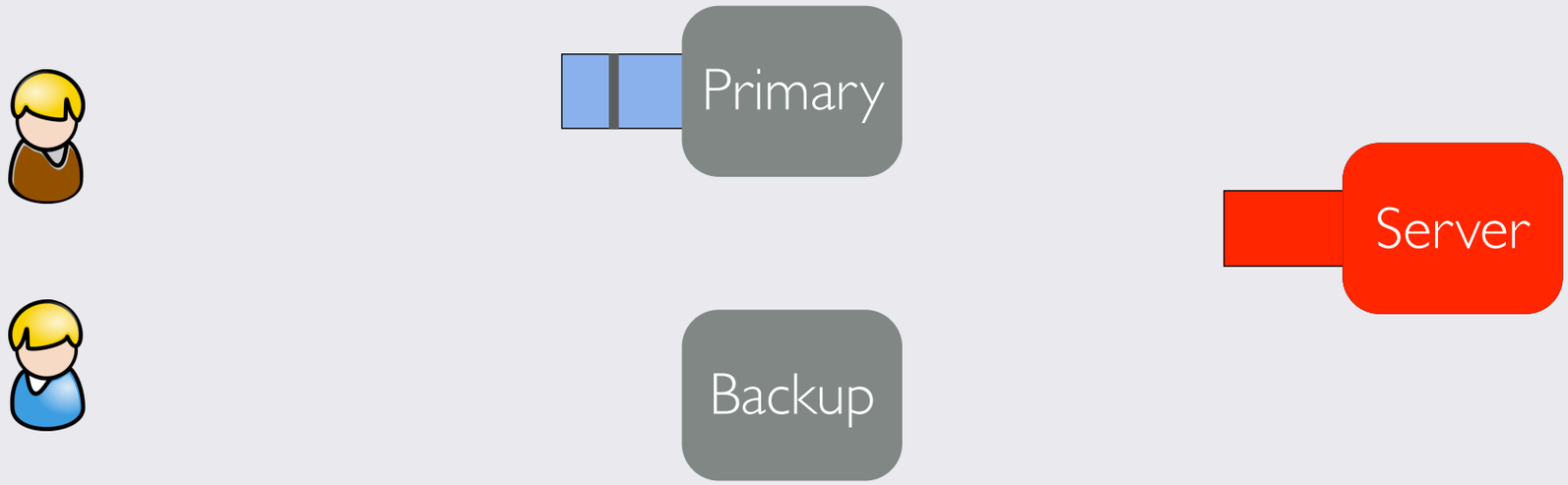
How interactions break our protocols

Example: Primary-backup



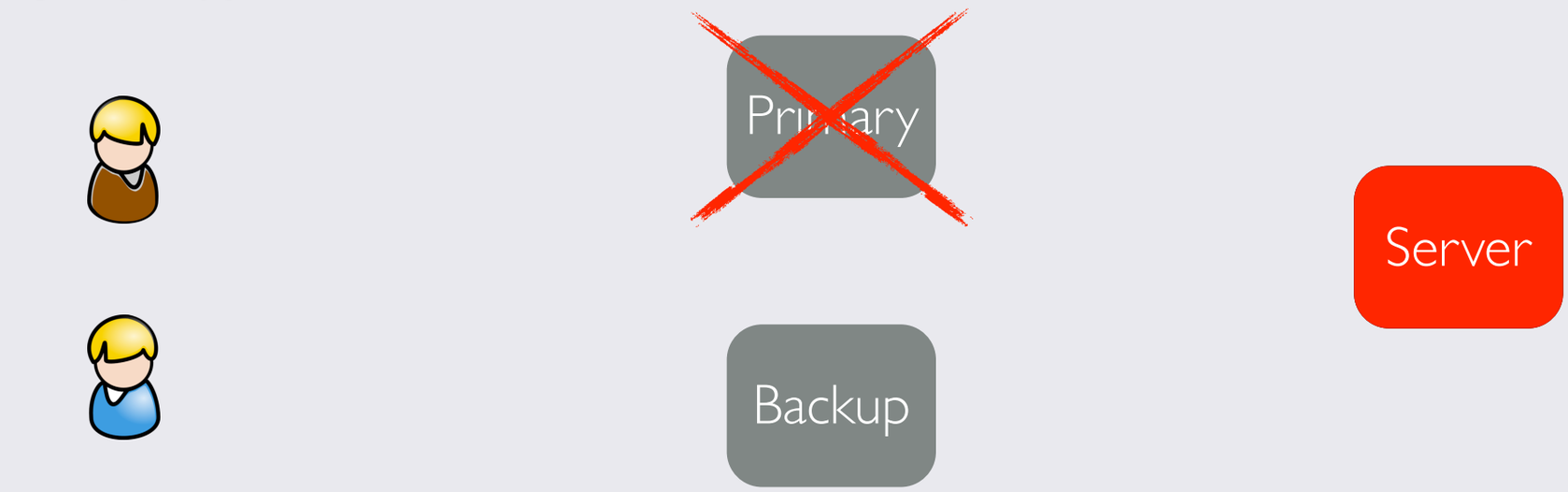
How interactions break our protocols

Example: Primary-backup



How interactions break our protocols

Example: Primary-backup




We the People
We hold these truths to be
self-evident, that all
output commits are created equal

You can still use speculation, if you **contain** its effects

Resolve speculation before sending a nested request

Rethink the sync

[OSDI '06]

Outline

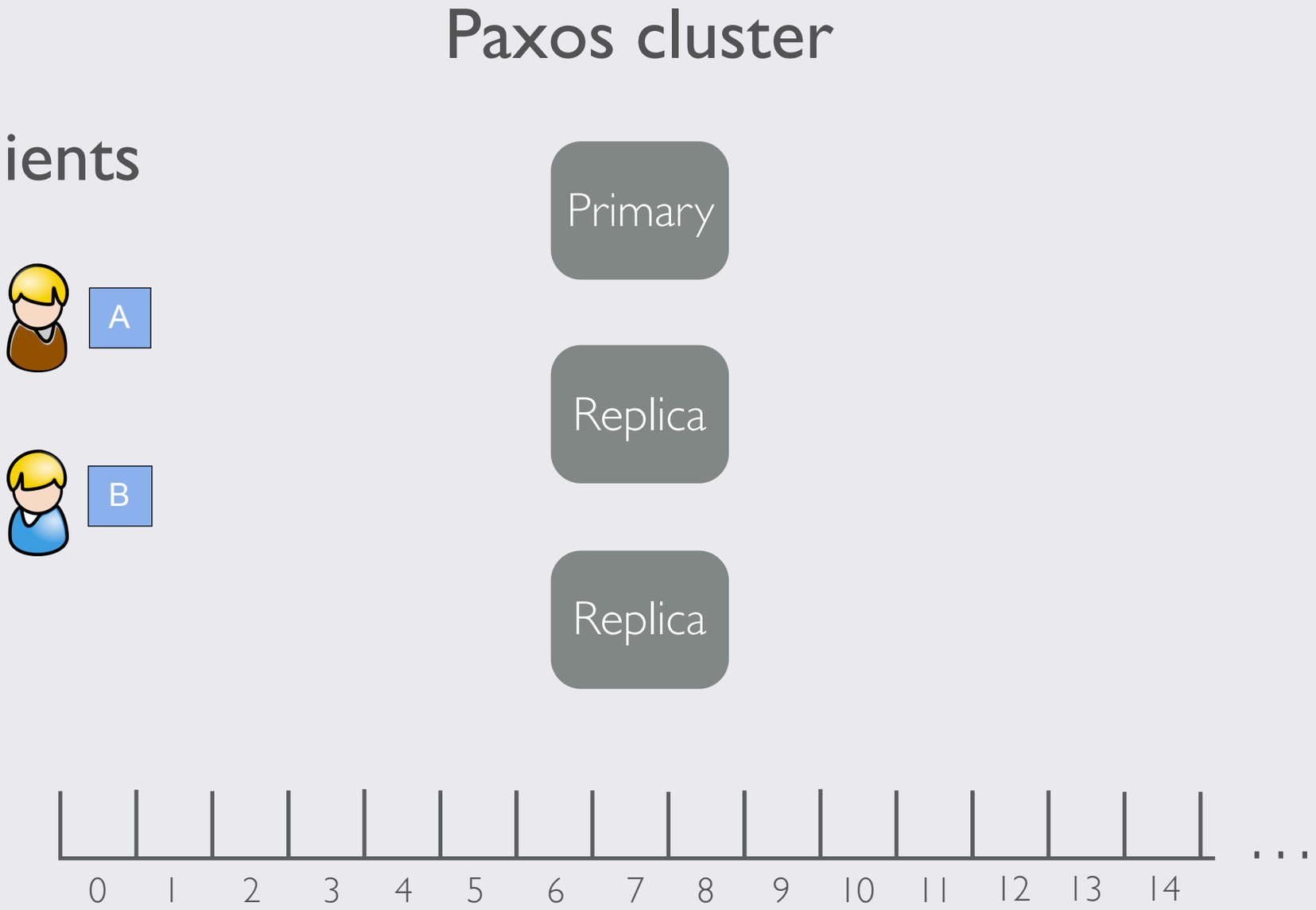
Problem statement

Correctness: the threat of speculation

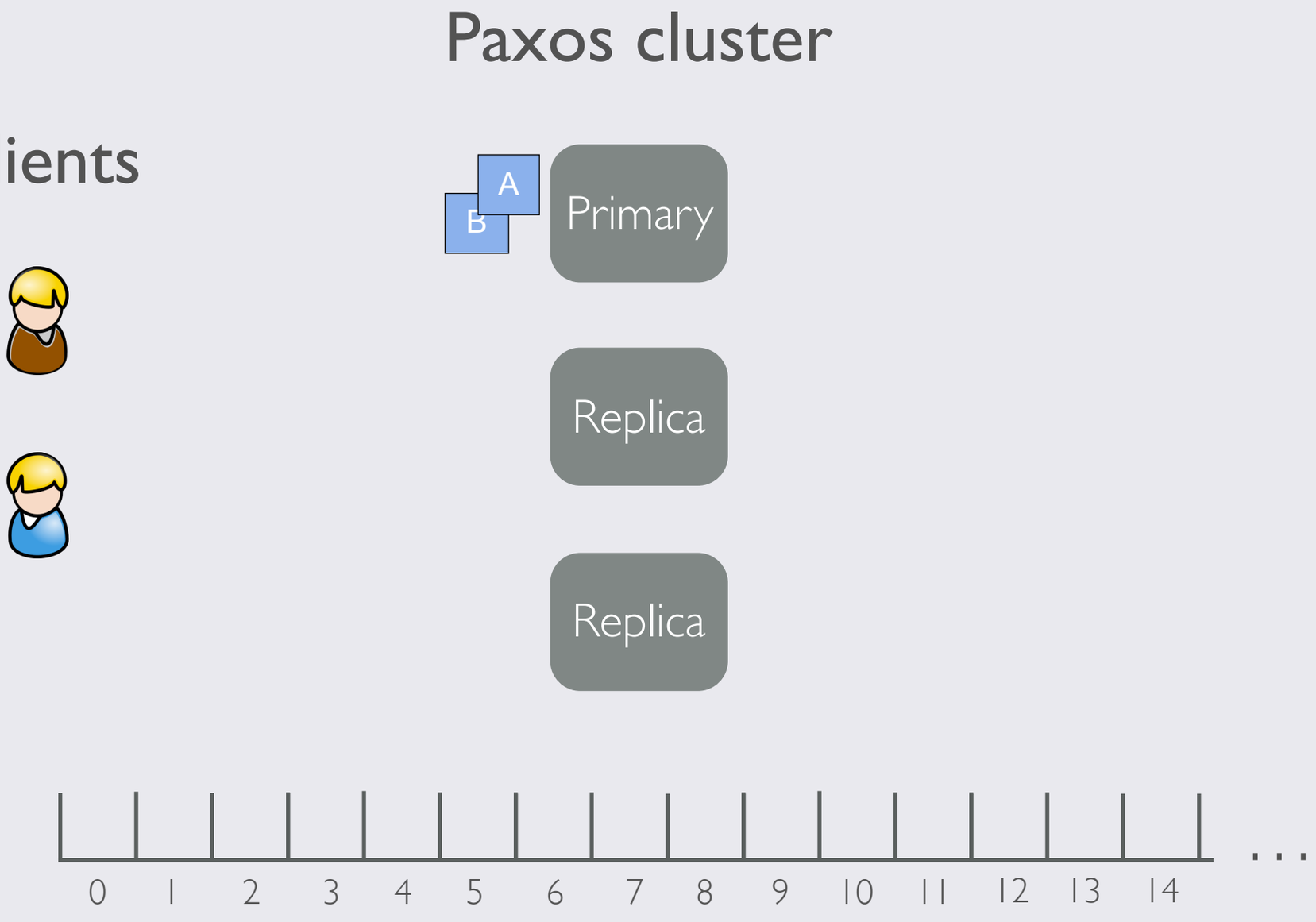
Performance: the price of sequential execution

Evaluation of Aegean

PAXOS AND SEQUENTIAL EXECUTION



PAXOS AND SEQUENTIAL EXECUTION



PAXOS AND SEQUENTIAL EXECUTION

Paxos cluster

Clients

Primary

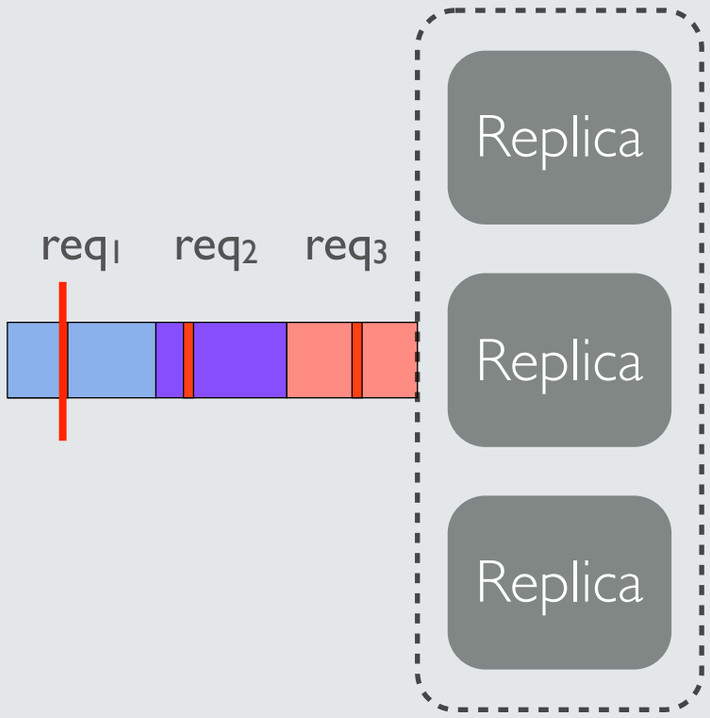
Replica

Replica

THE PRICE OF SEQUENTIAL EXECUTION

Service A

Service B



Backend server

THE PRICE OF SEQUENTIAL EXECUTION

Service A

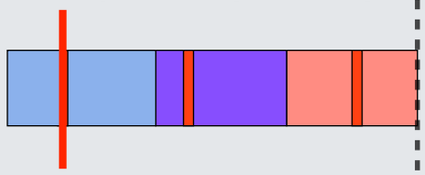
Service B

Replica

Replica

Replica

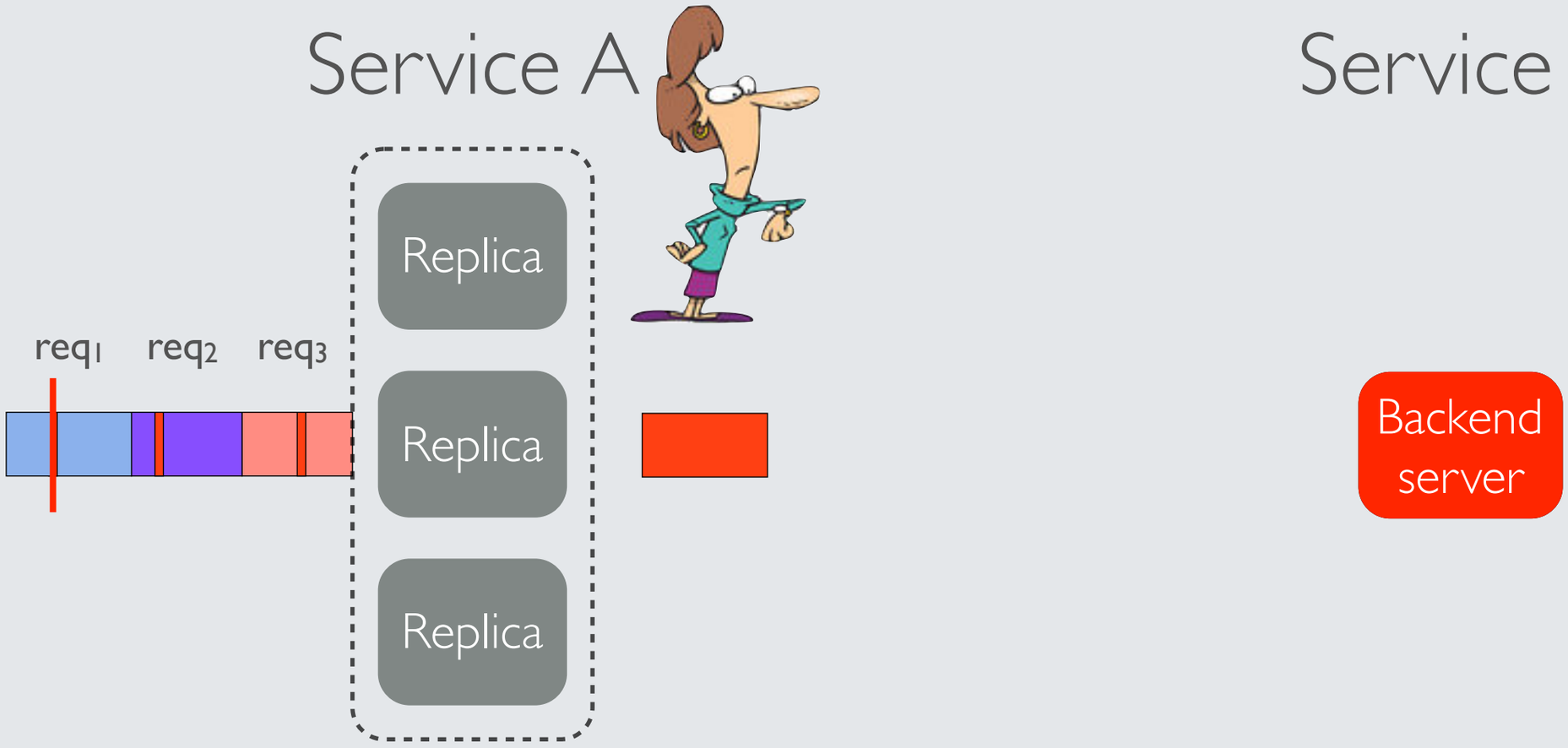
req_1 req_2 req_3



Backend server

Service A remains idle while waiting for B's reply.

THE PRICE OF SEQUENTIAL EXECUTION



Service A remains idle while waiting for B's reply.

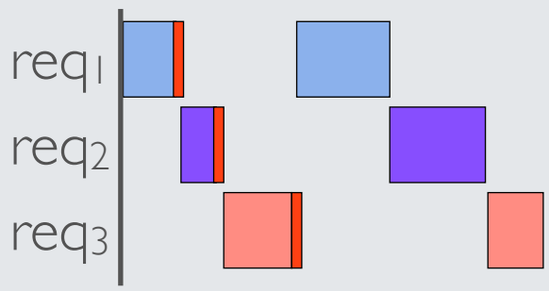
NO NEED FOR IDLENESS

Insight

We need a **deterministic** execution schedule
(not necessarily a sequential one)

Sequential

Request pipelining



NO NEED FOR IDLENESS

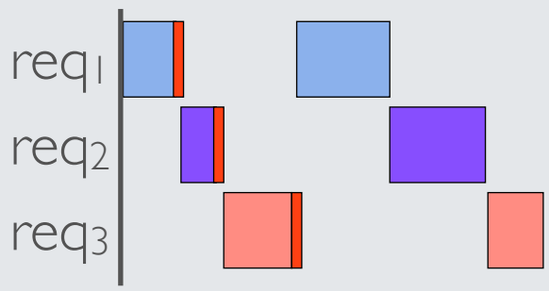
Insight

We need a **deterministic** execution schedule
(not necessarily a sequential one)

What about linearizability?

Sequential

Request pipelining



It is ***not*** the job of the replication protocol to enforce linearizability

Goal of replication: provide the abstraction of a ***single correct server***

Outline

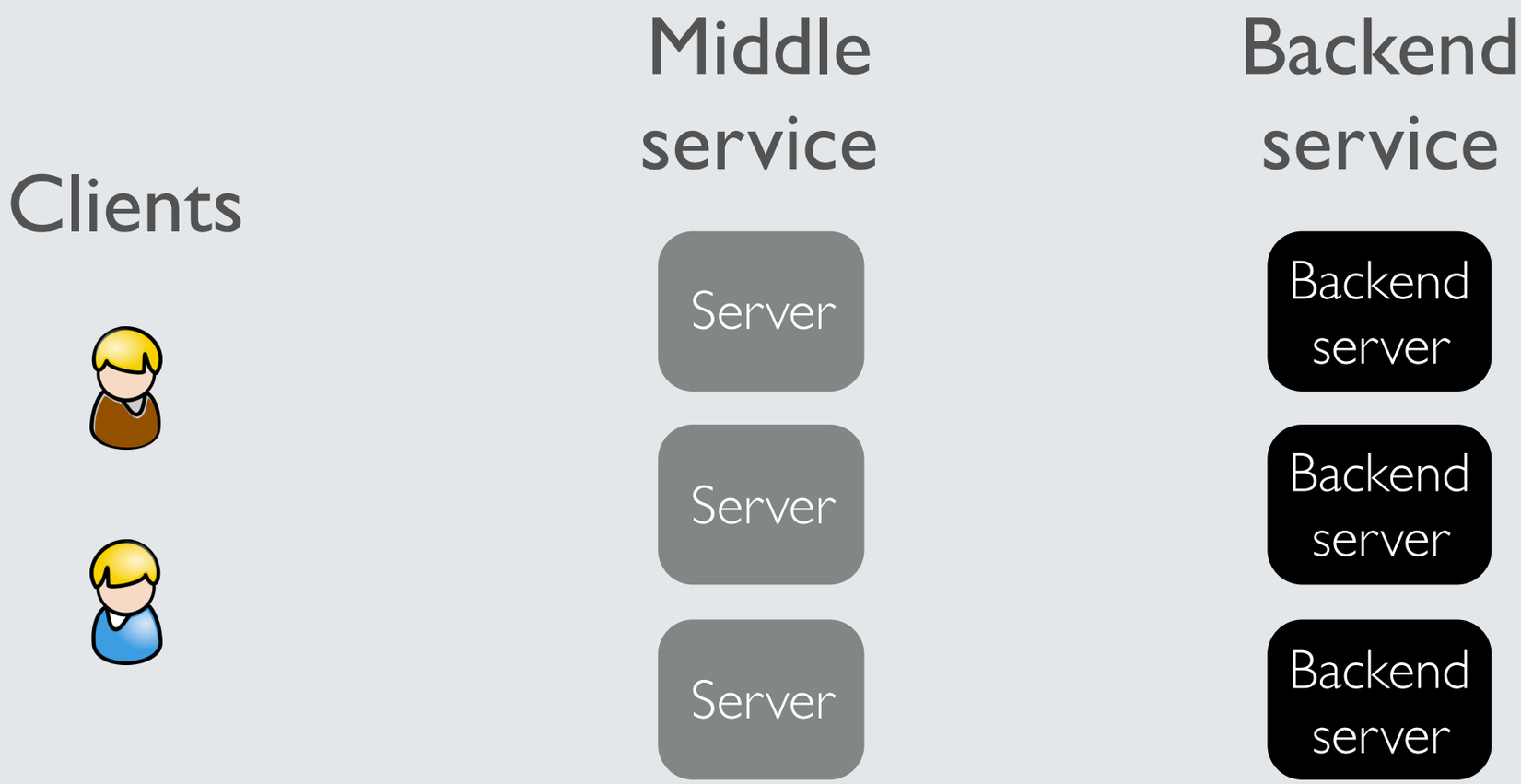
Problem statement

Correctness: the threat of speculation

Performance: the price of sequential execution

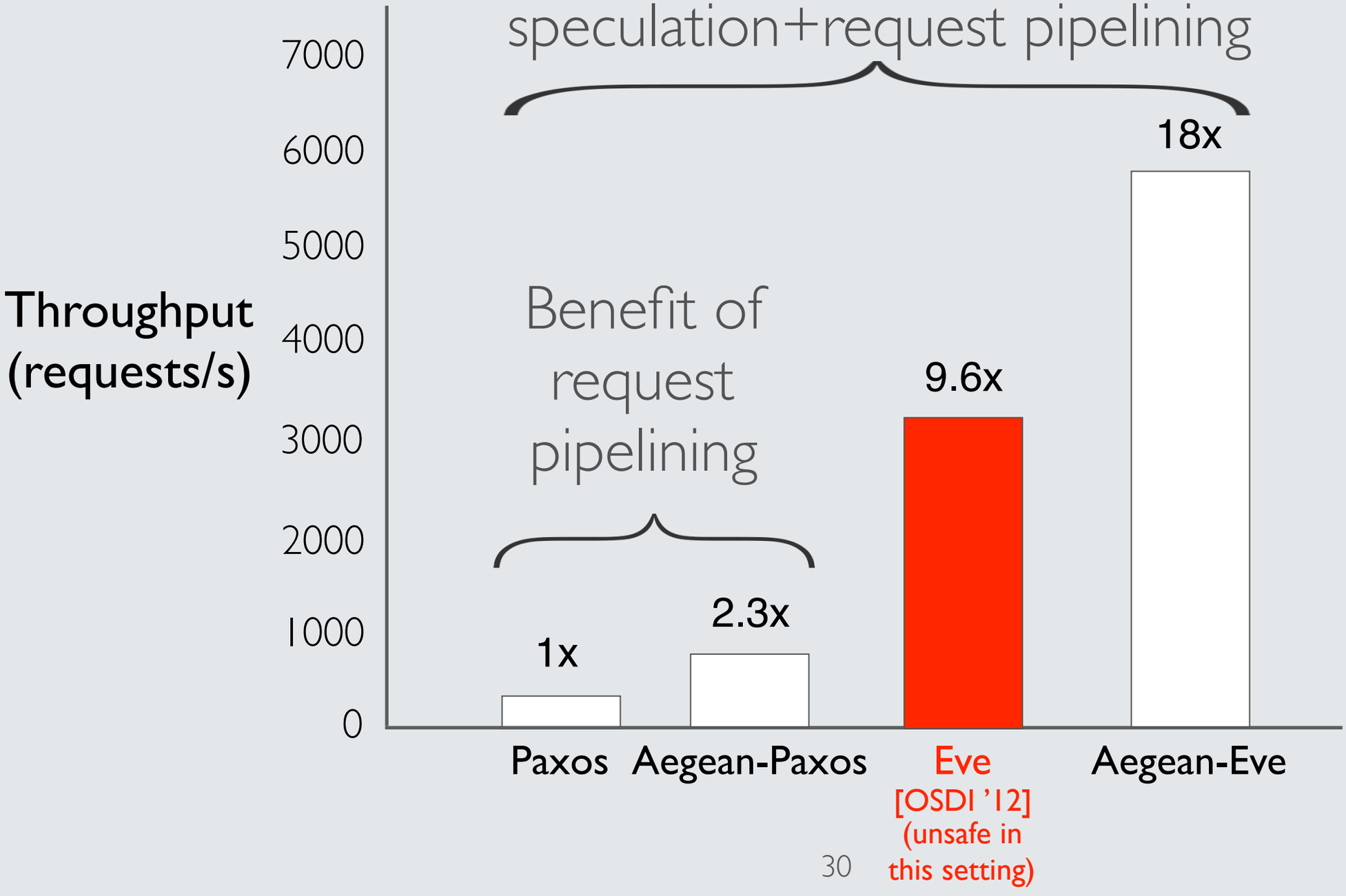
Evaluation of Aegean

EXPERIMENT SETUP



Requests incur a total of 1ms of execution time on each service

PERFORMANCE BENEFIT OF AEGEAN



CONCLUSION

We are moving **beyond the client-server** model

Adapt replication protocols to account for **interactions between services**

- Restore correctness
- Optimize performance

Thank you!
Questions?

https://github.com/GLaDOS-Michigan/Aegean