

You are not alone

Scaling Multiplayer Games

Jesper Richter-Reichhelm - @jrrei

Knut Nesheim - @knutin



Wooga













> 1,000,000 daily users

- > 1,000,000 daily users
- > 50,000 concurrent users

- > 1,000,000 daily users
- > 50,000 concurrent users
- > 10,000 requests / second

Ok, you can
handle that.



Ok, you can
handle that.

But why only
single player?



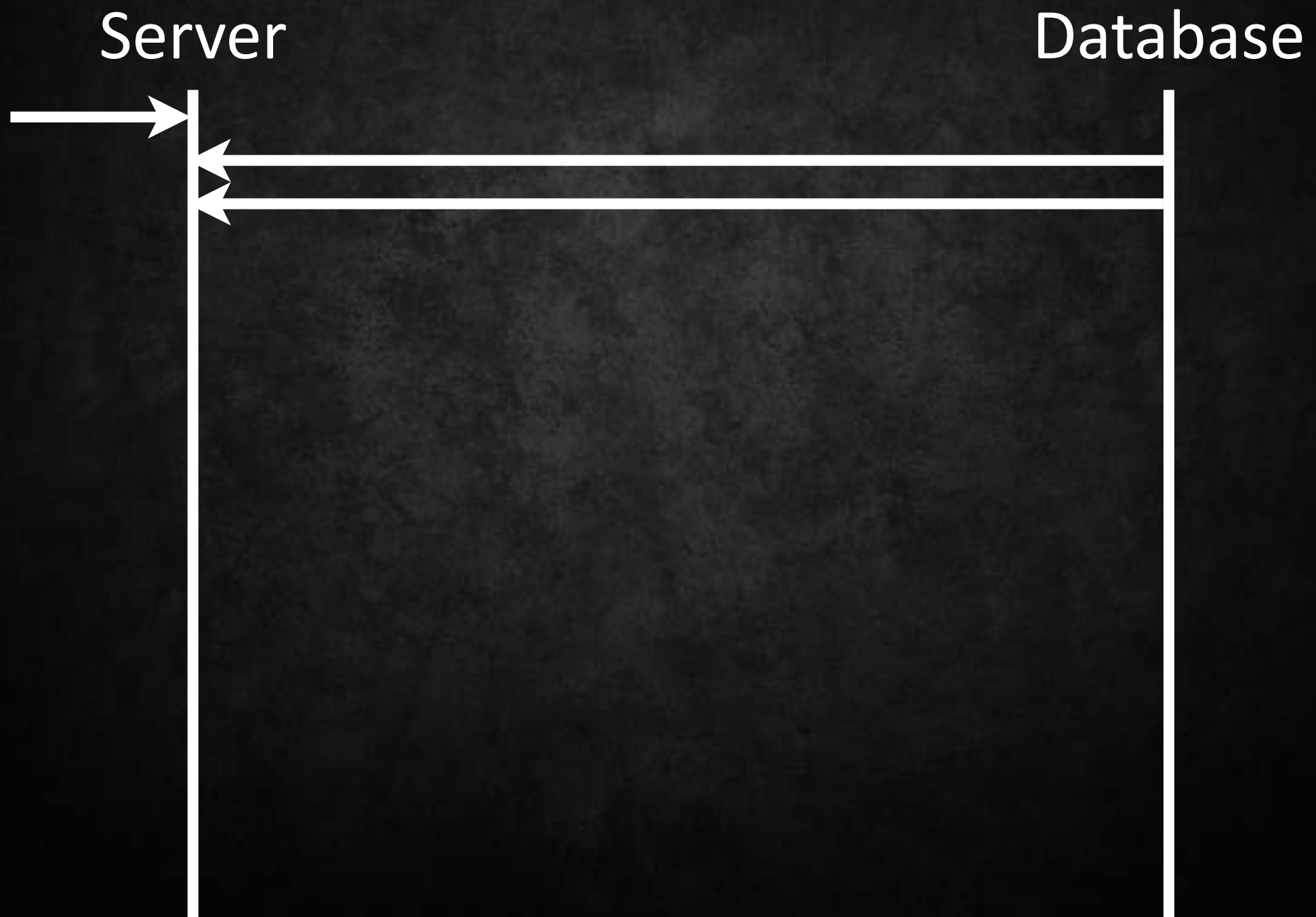
The Games
The Past
The Present
The Future
The Foundation

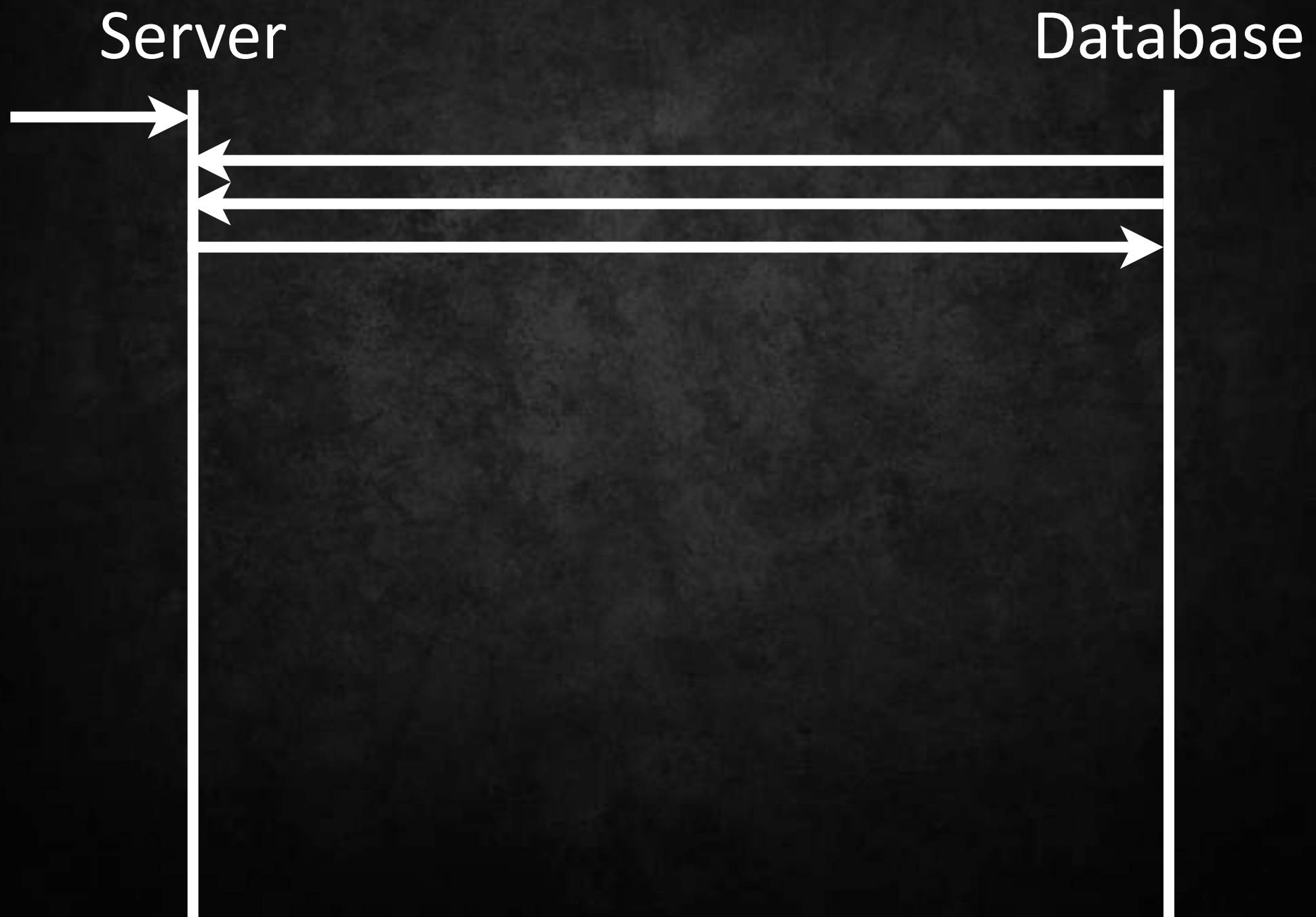
Server

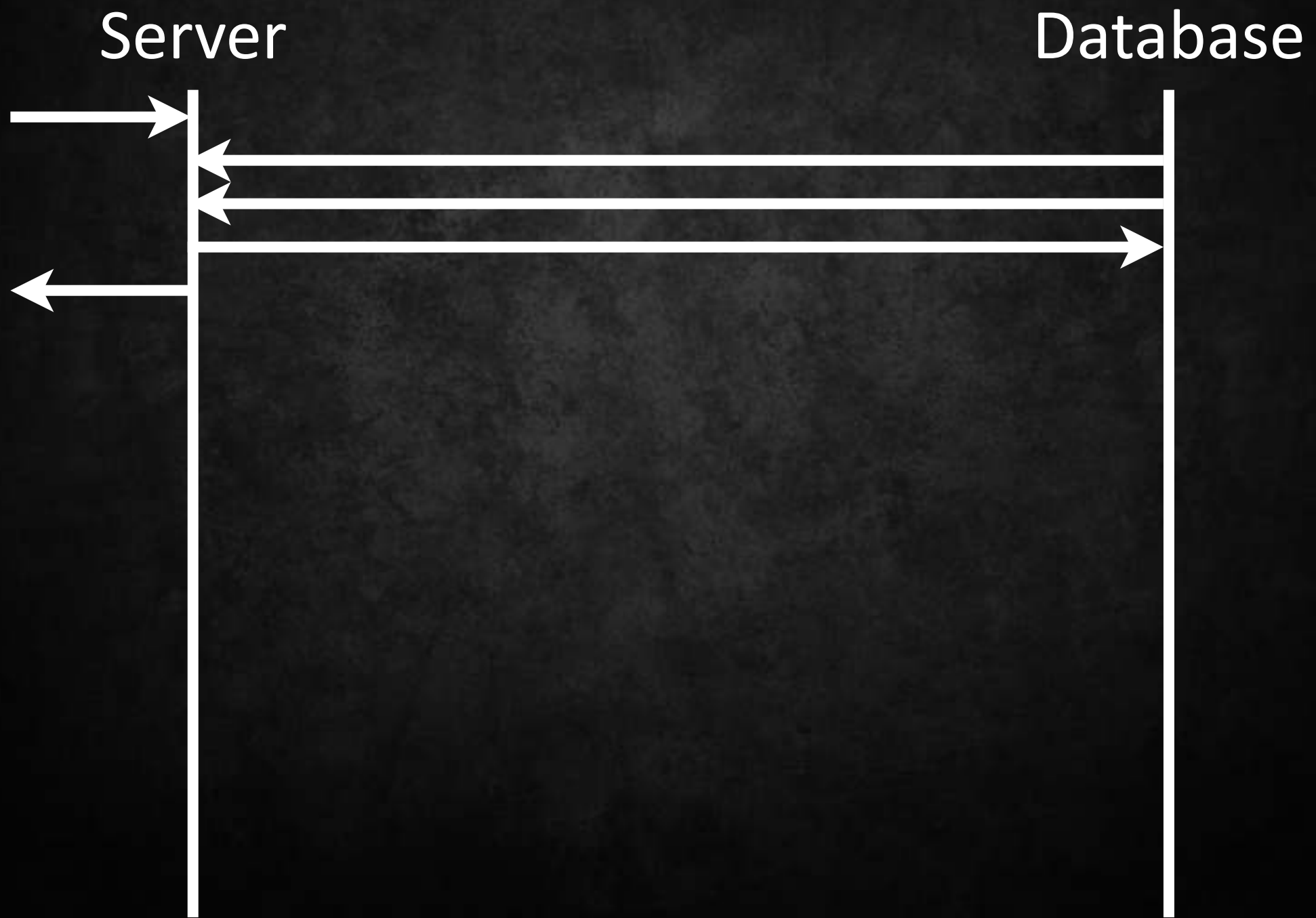


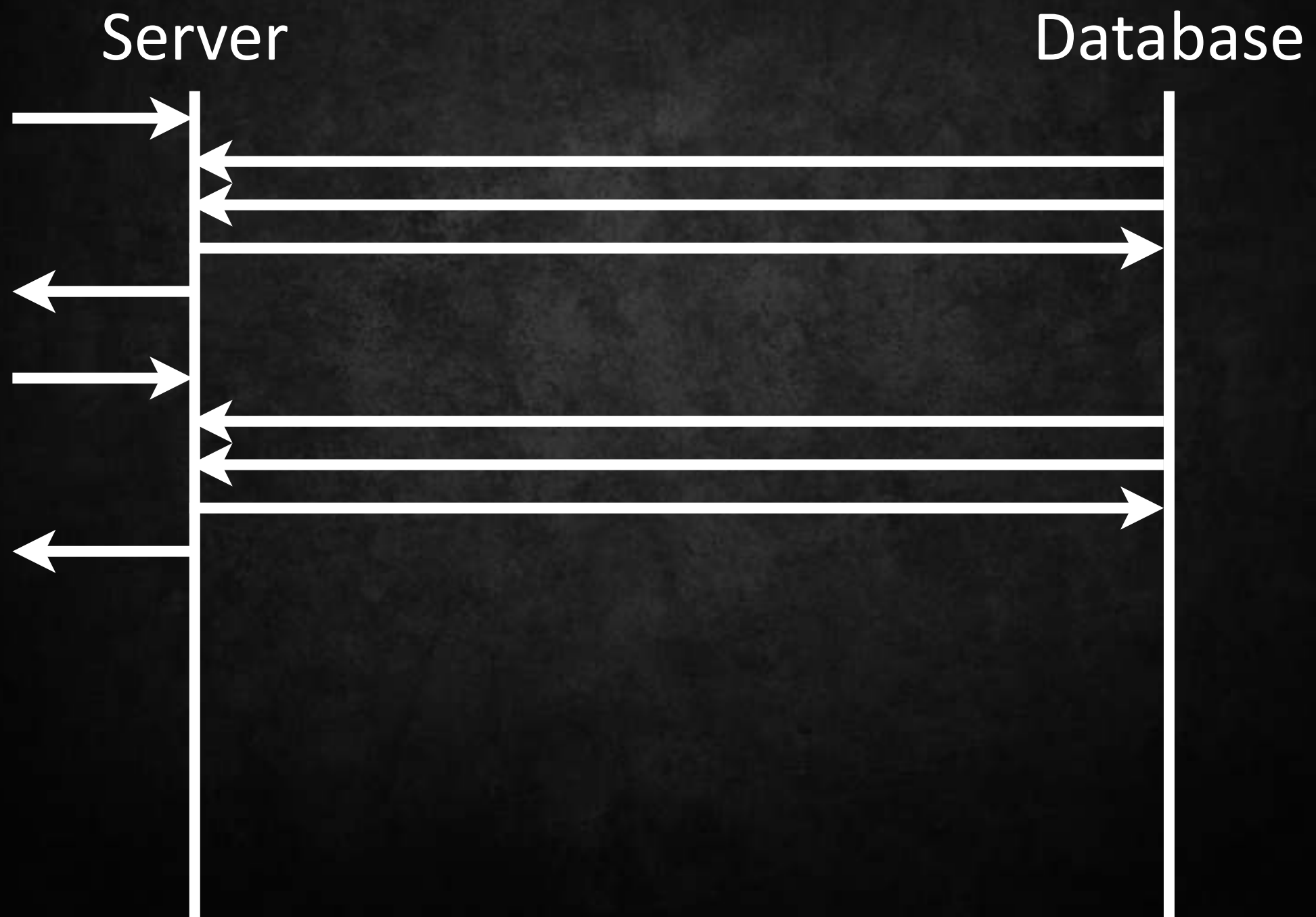
Database

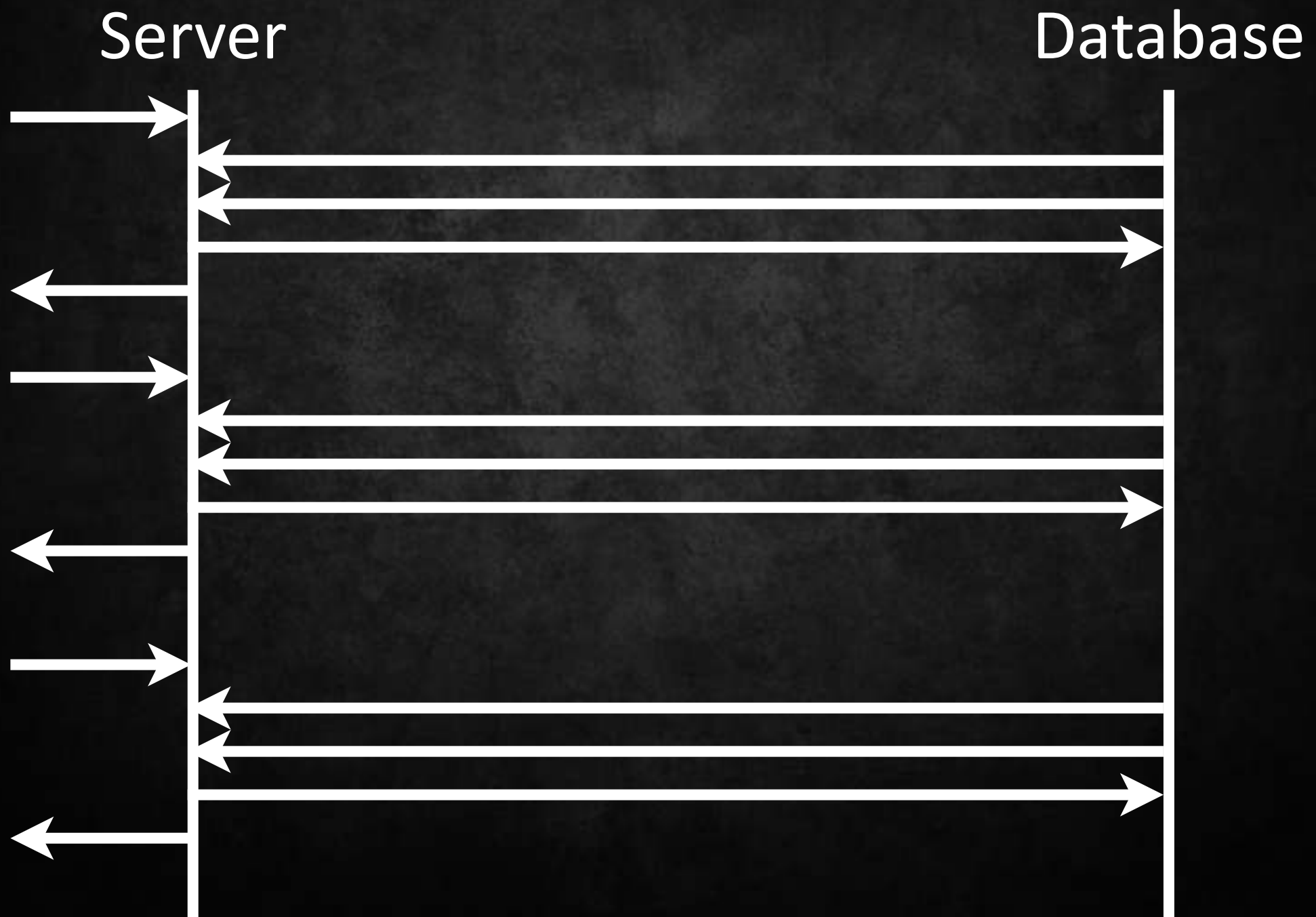












DB is the bottleneck



**Magic
Land**

Server

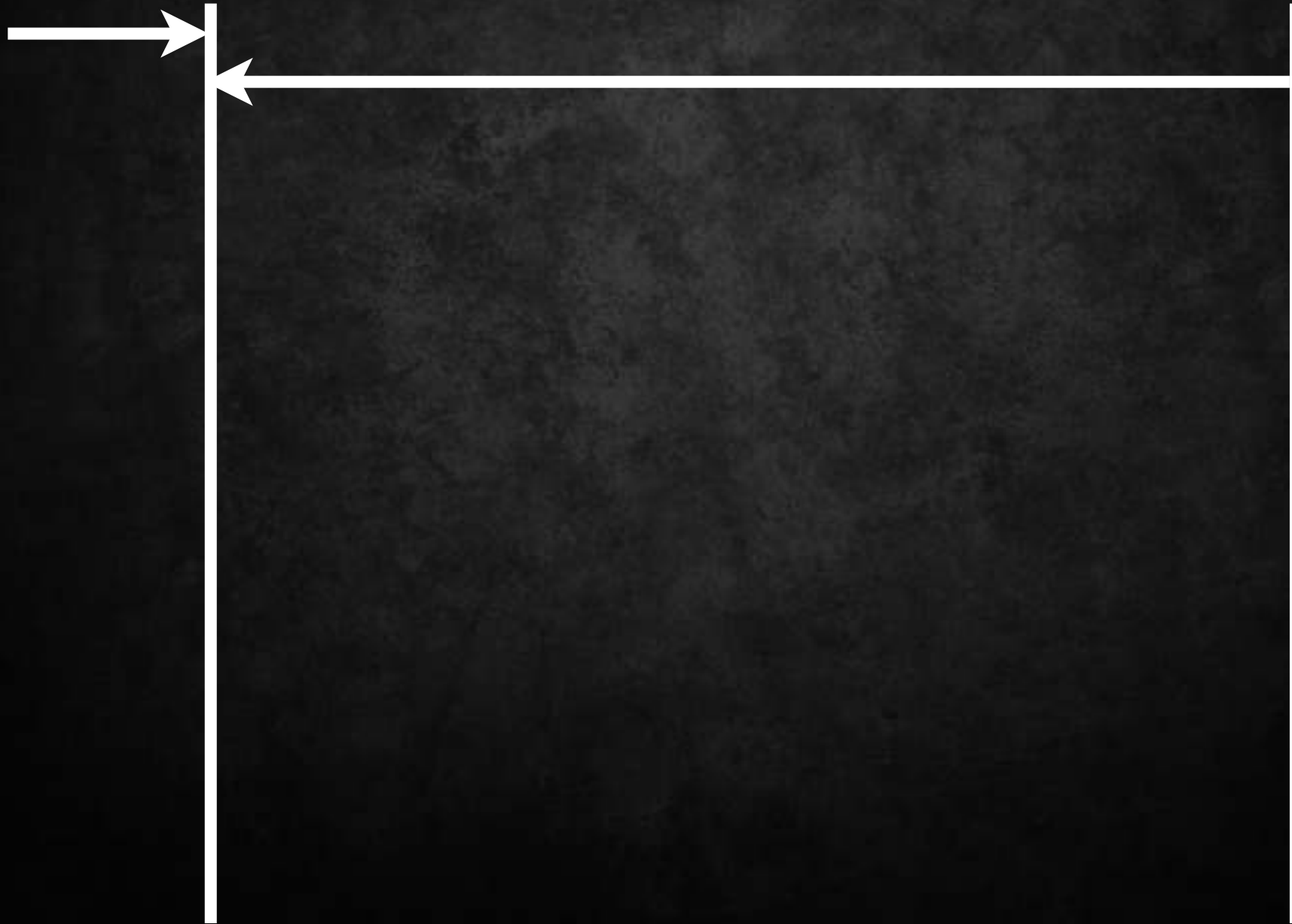


Database



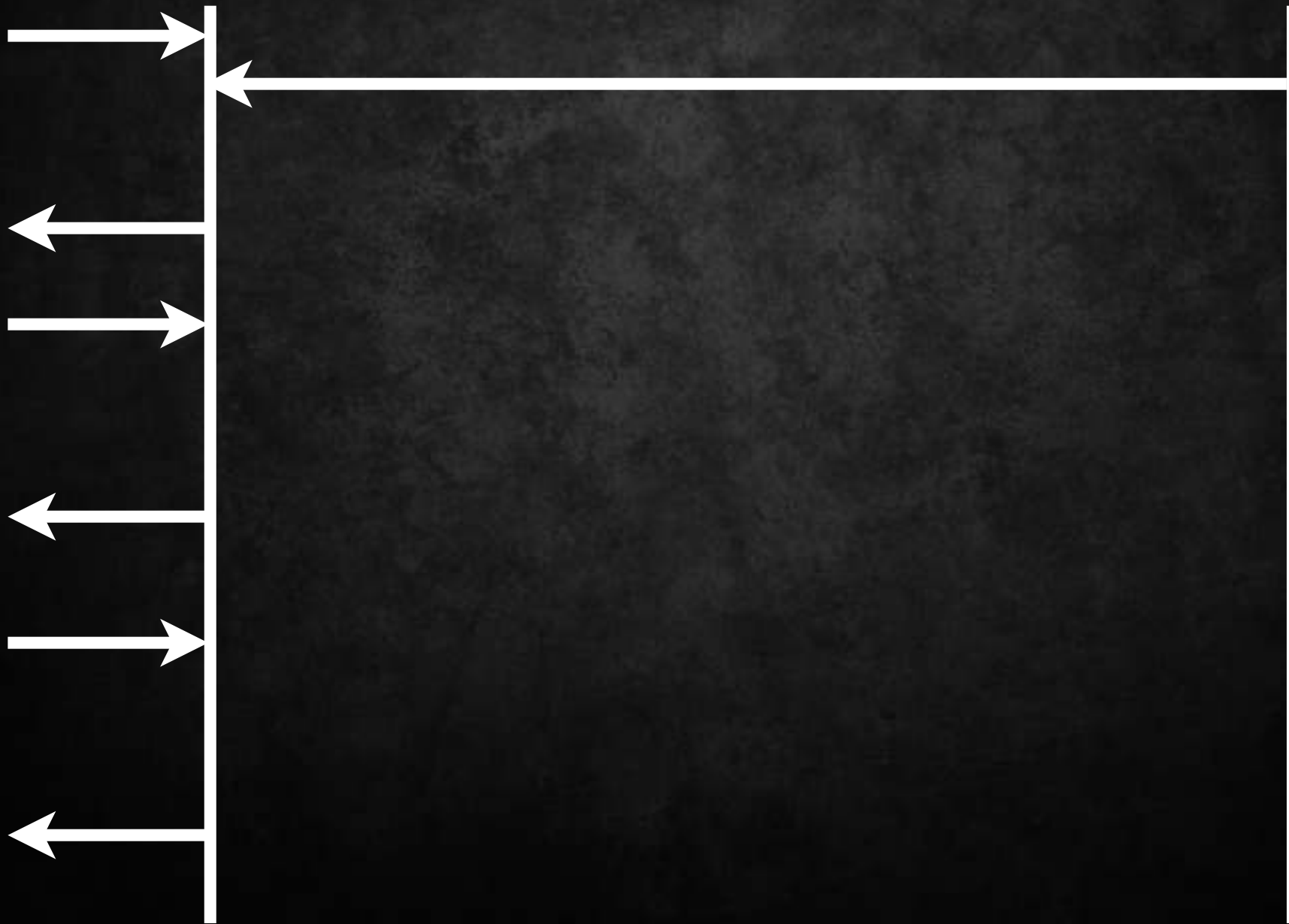
Server

Database



Server

Database



Server

Database



One Game Session

Server

Database

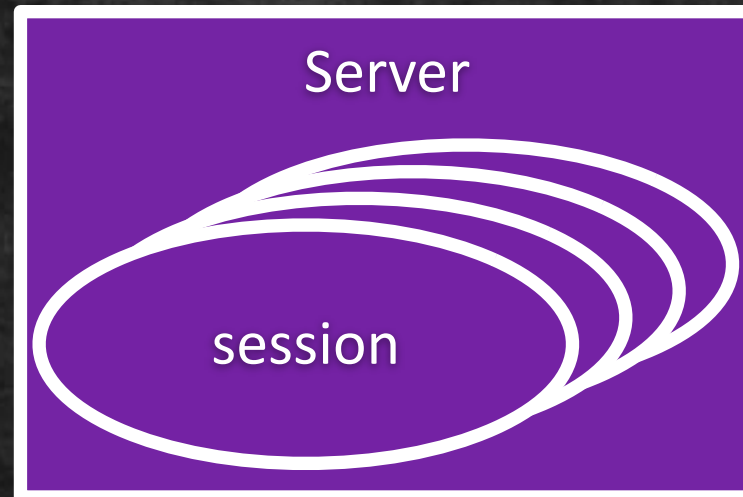


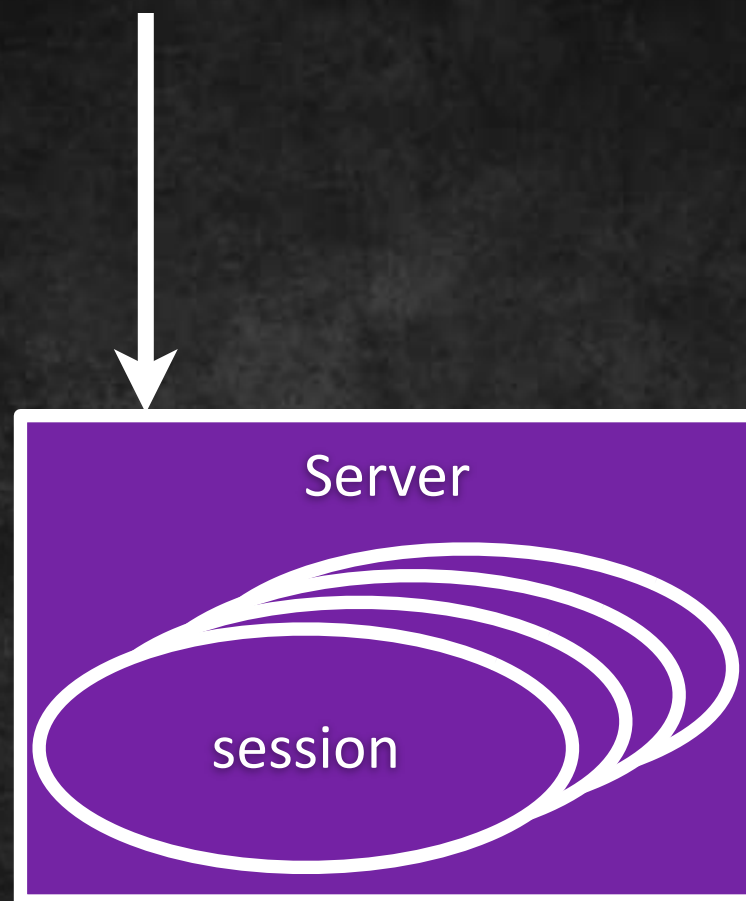
One Game Session

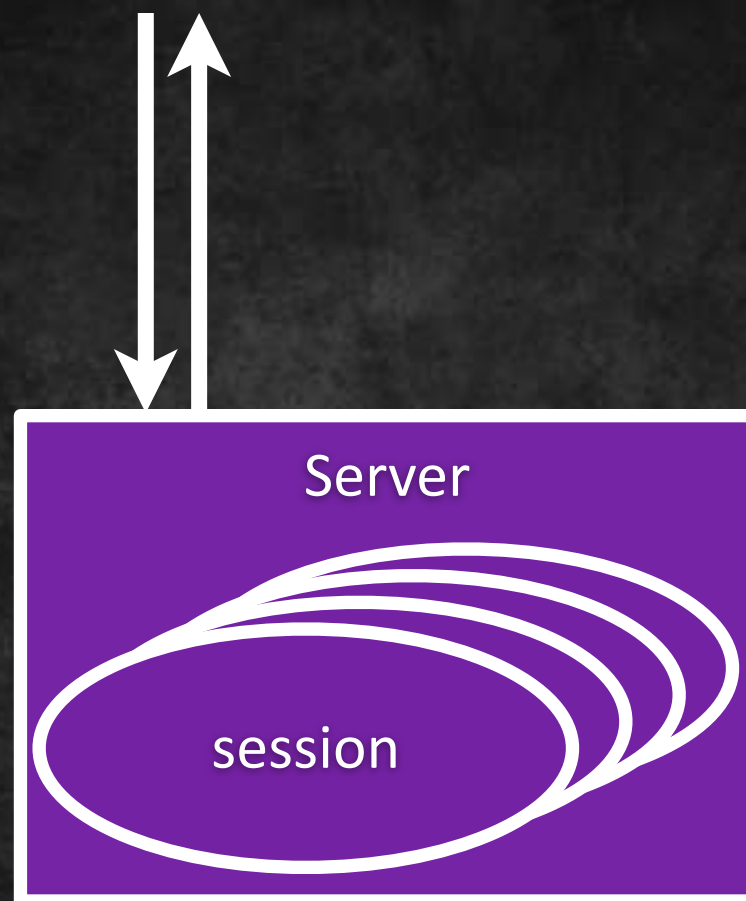


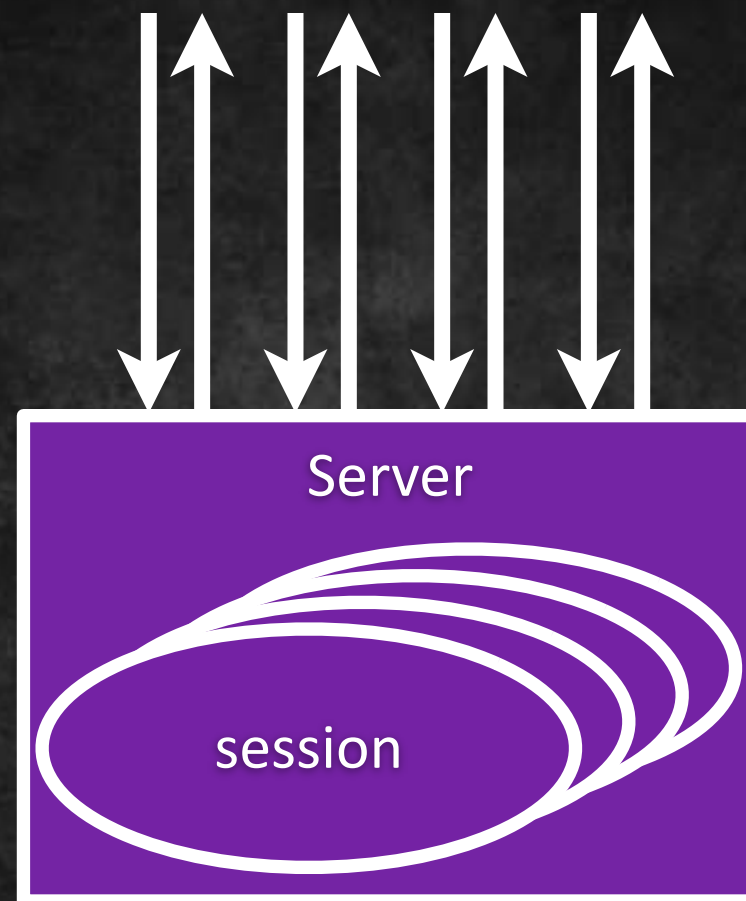
session

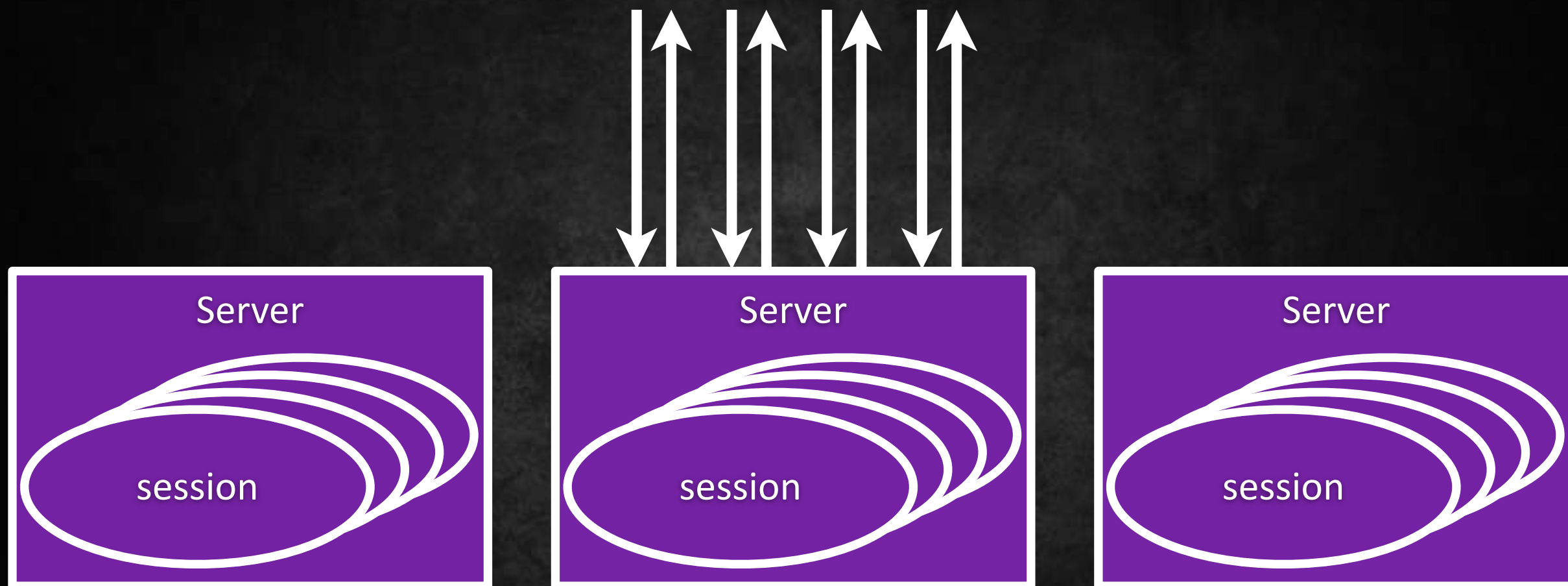












The Games
The Past
The Present
The Future
The Foundation

Thoughts
Client
Server
Operation

Thoughts
Client
Server
Operation



You are not alone

Pessimistic

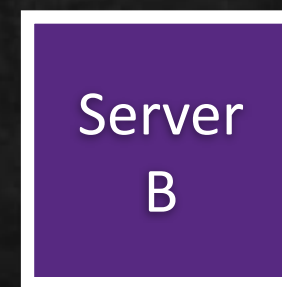
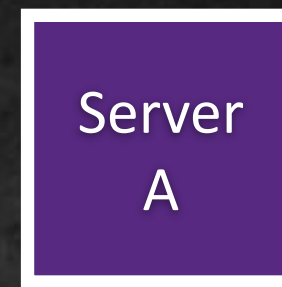
Pessimistic
Schizophrenic

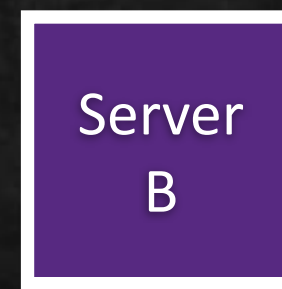
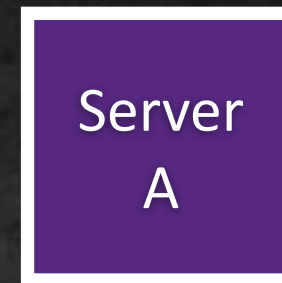
Pessimistic

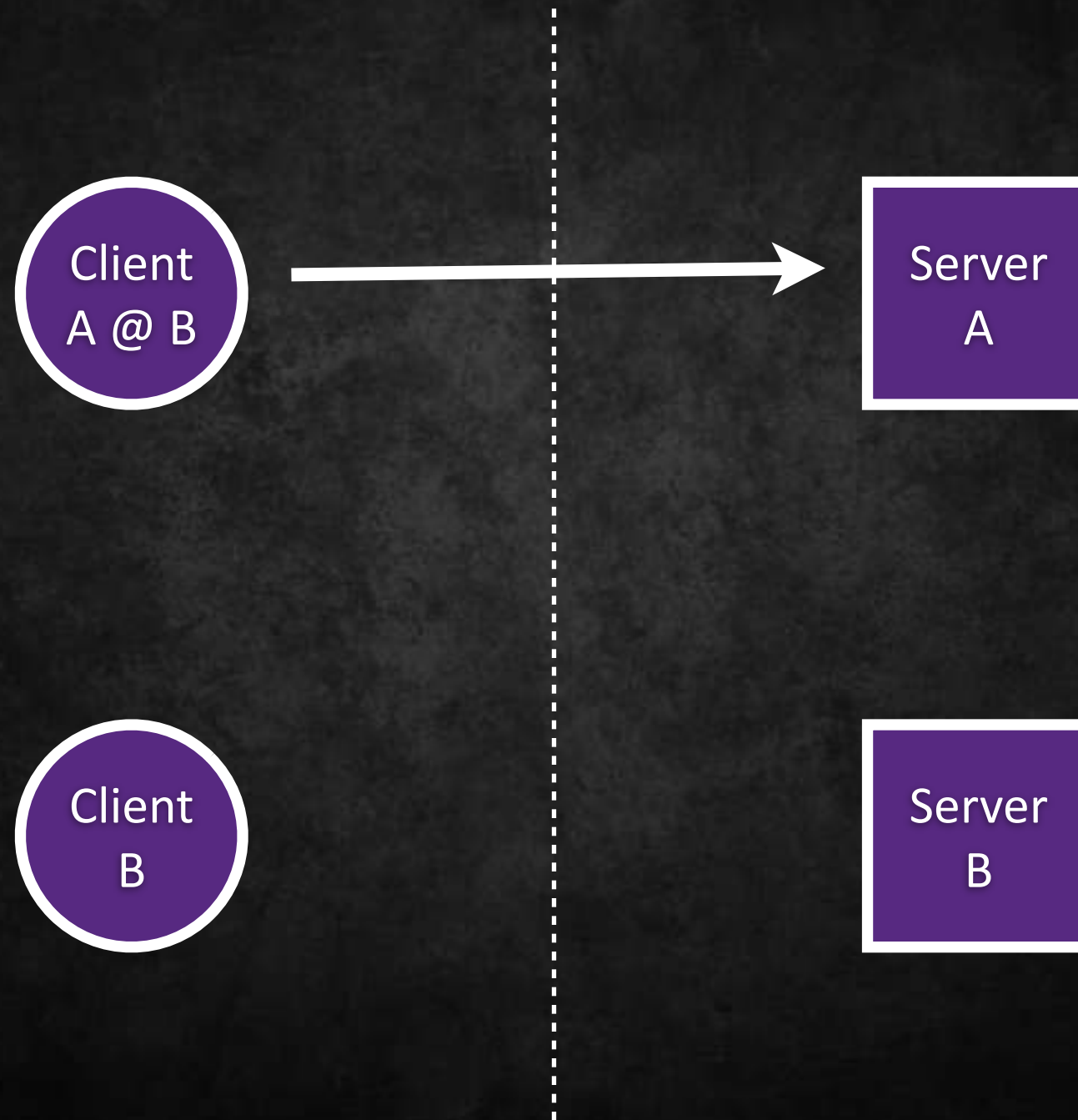
Schizophrenic

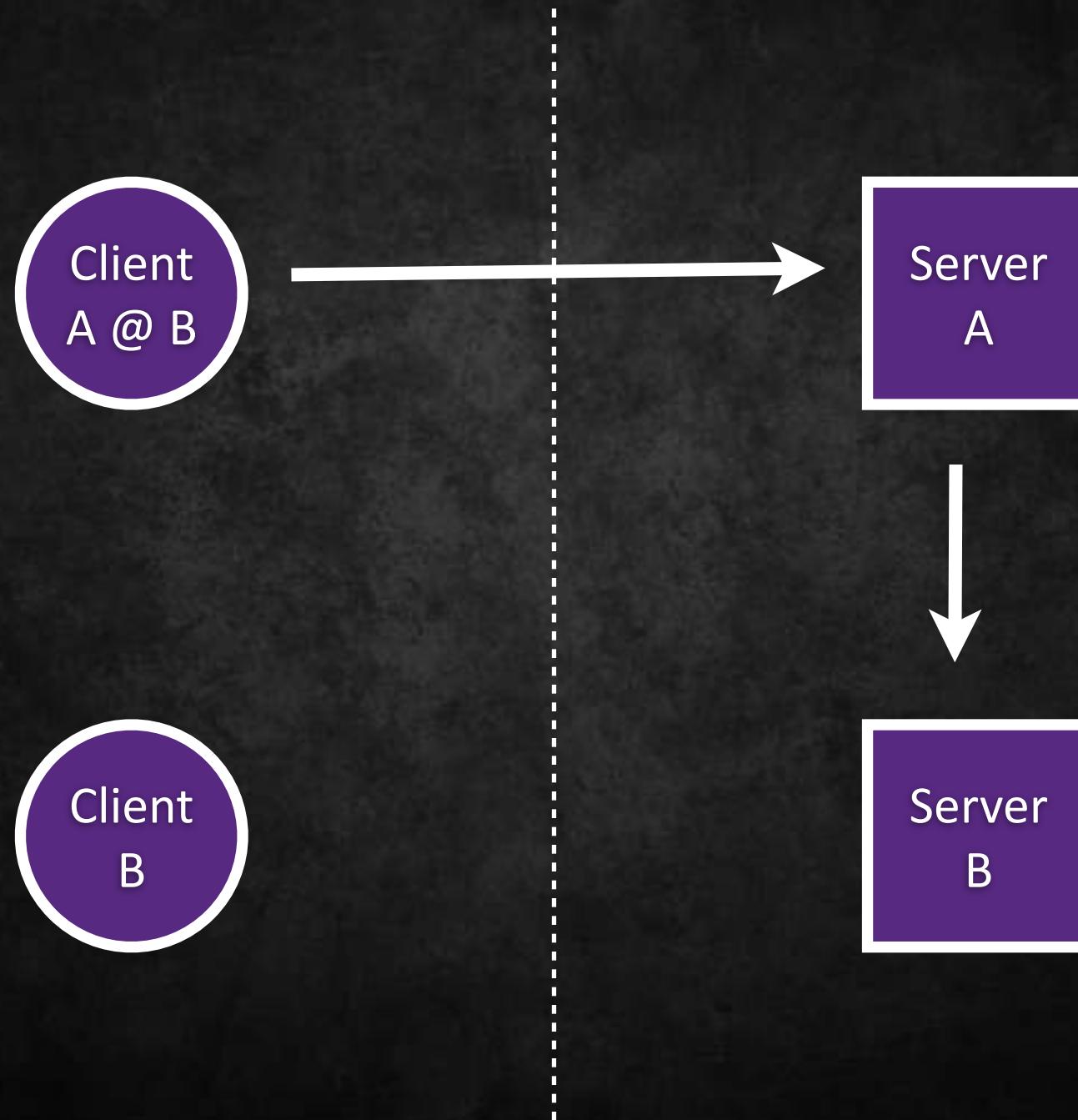
Optimistic

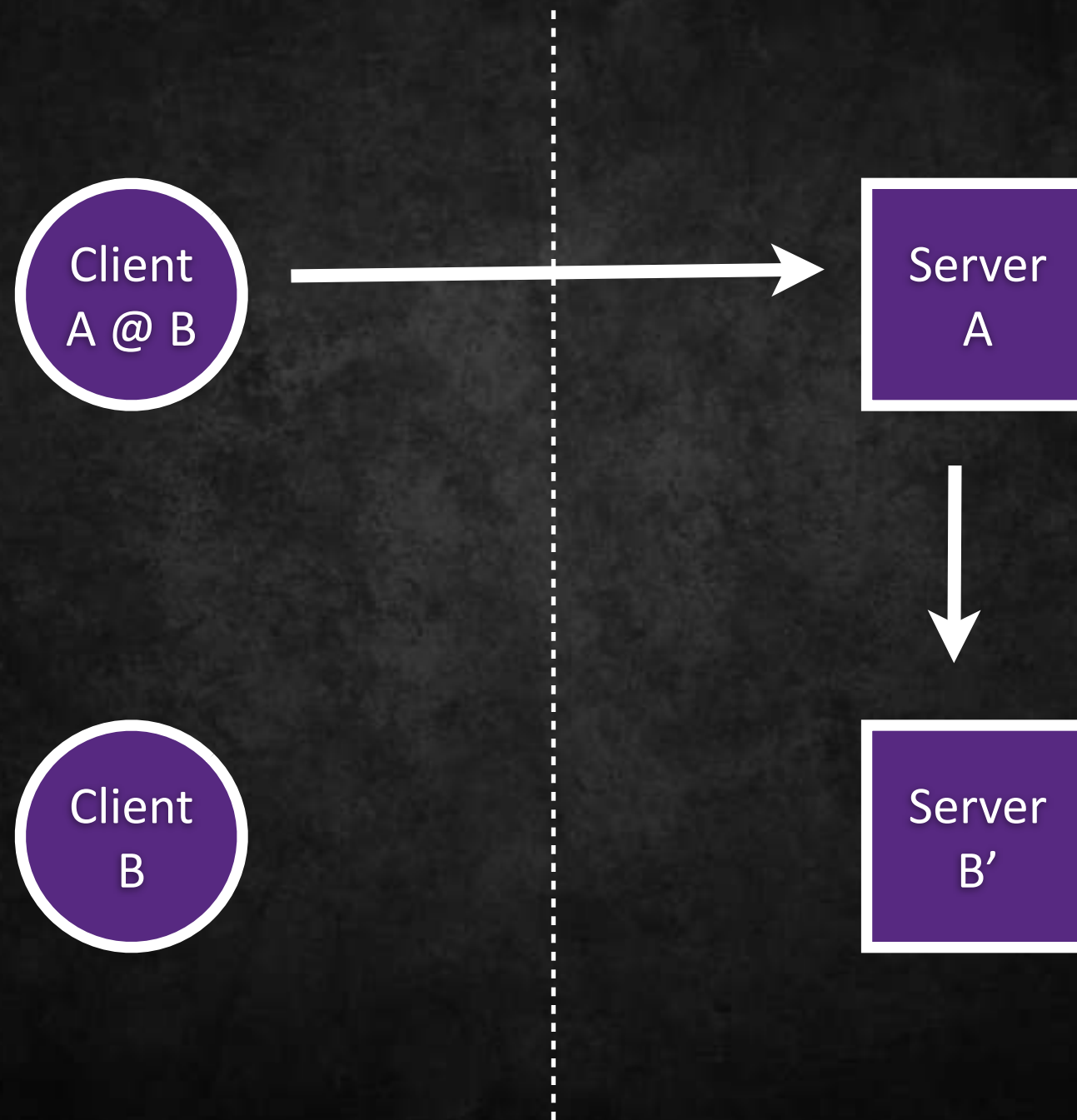
Pessimistic

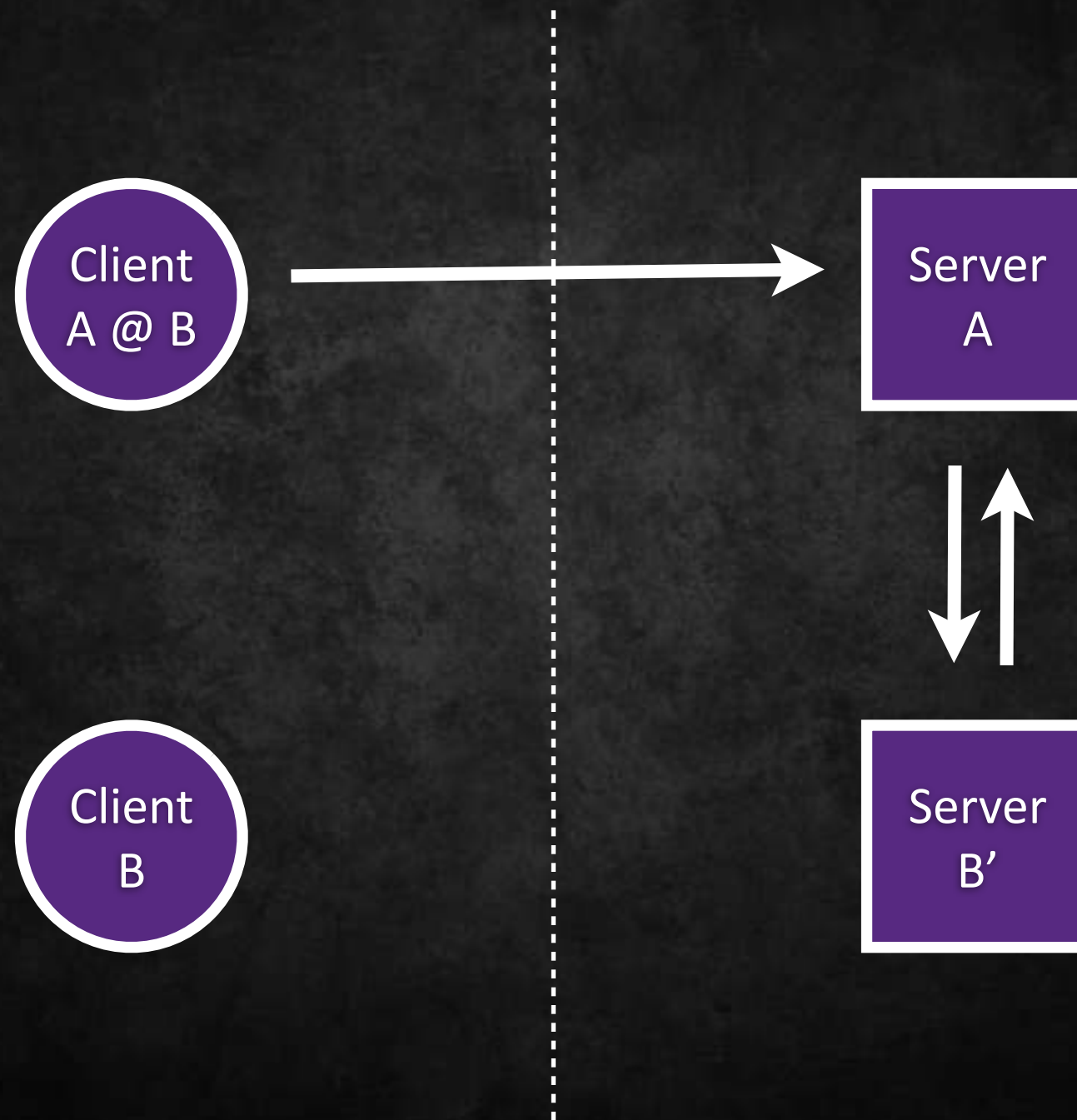


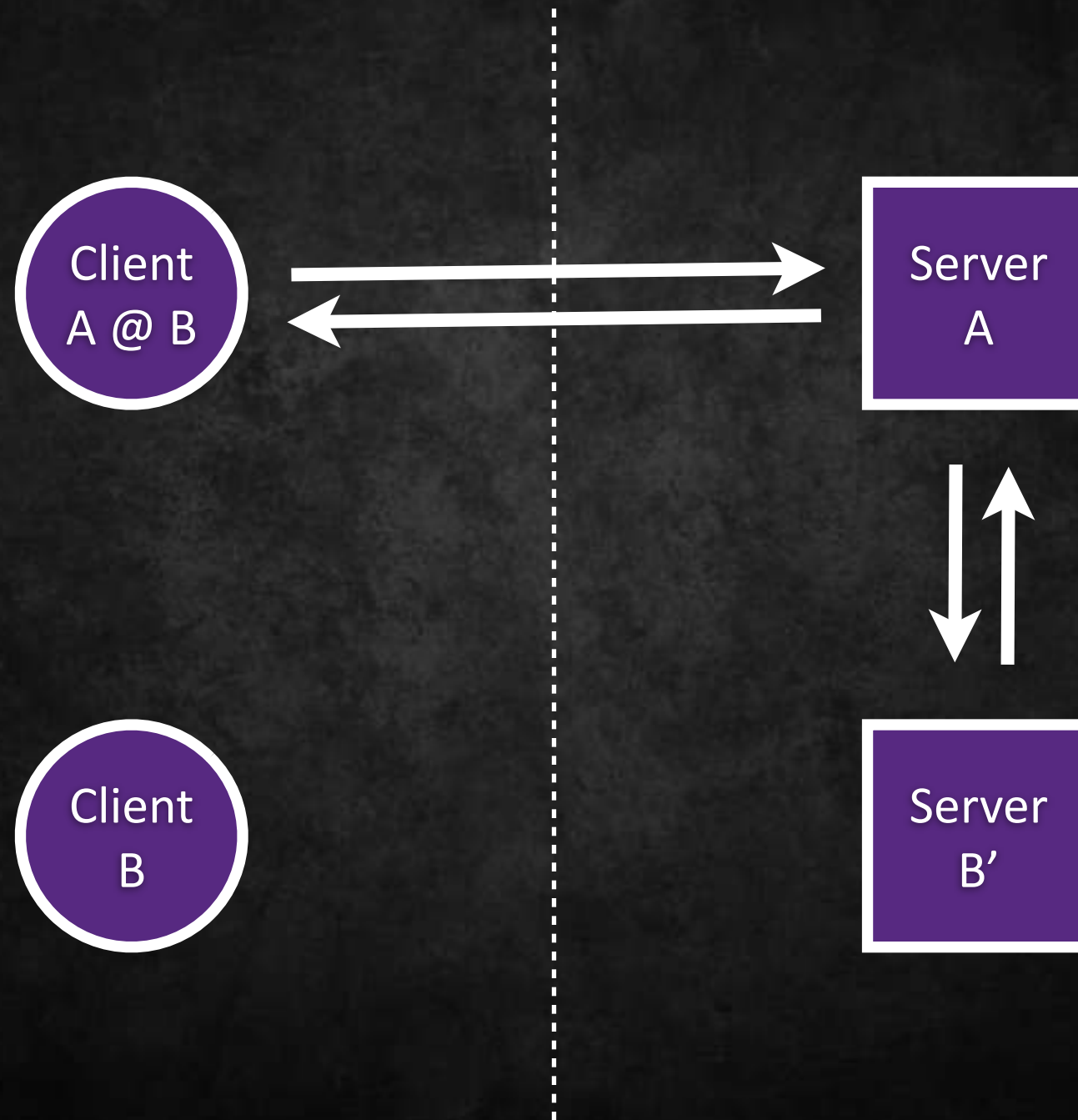


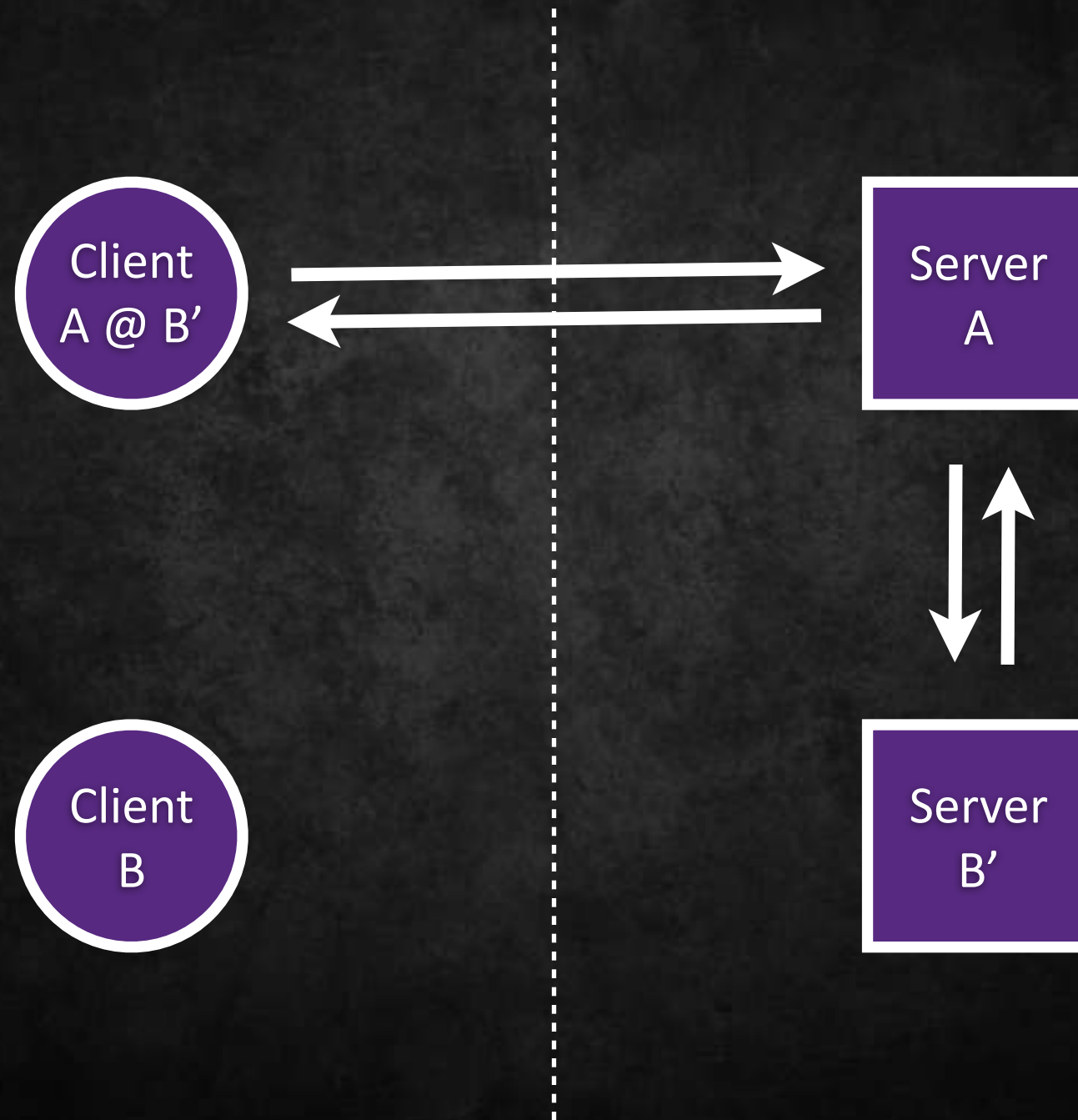


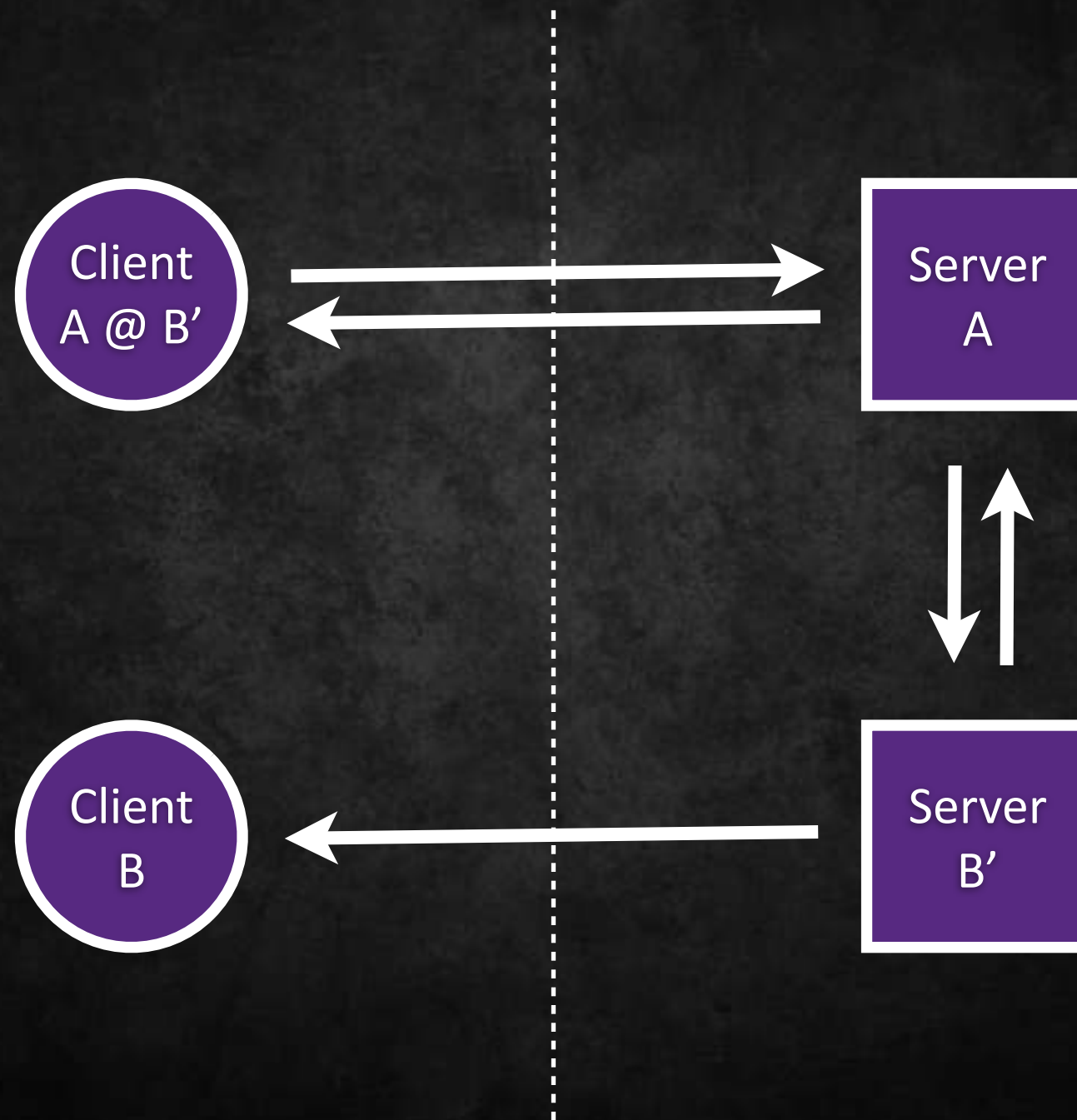


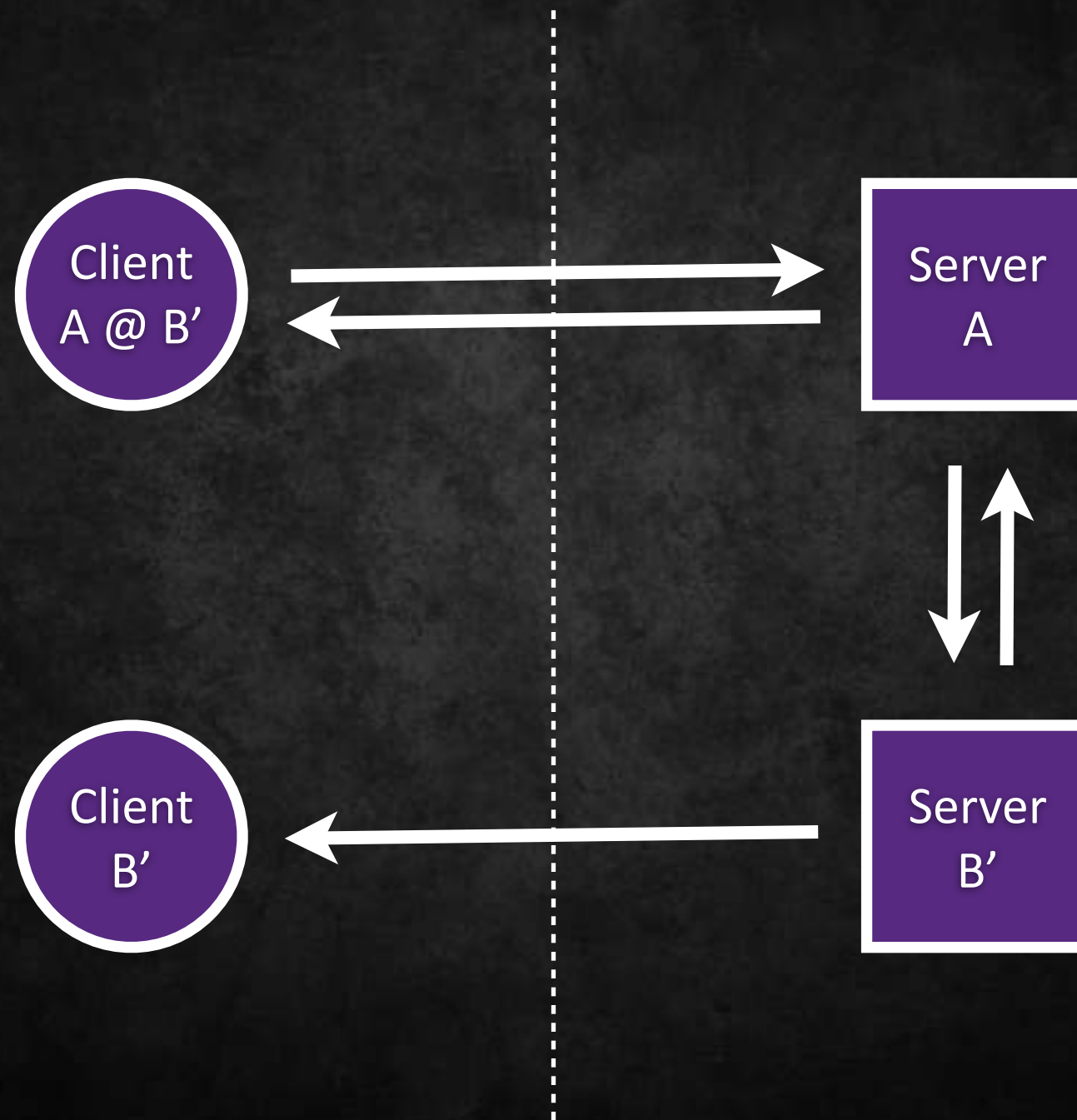














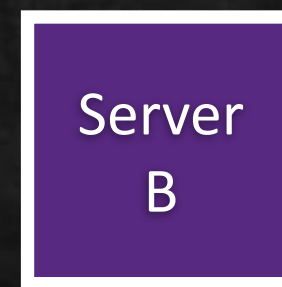
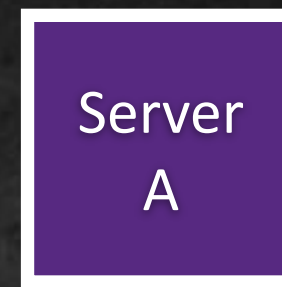
Waiting on server
responses?

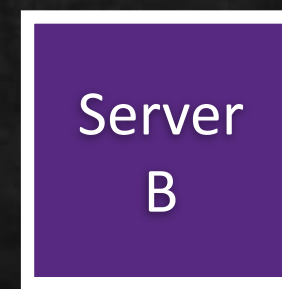
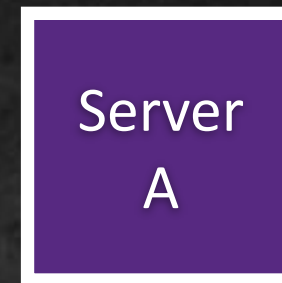


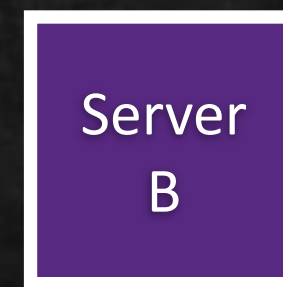
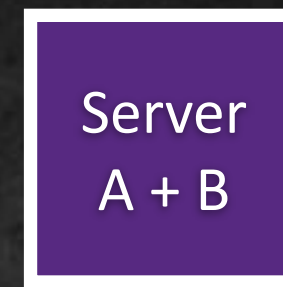
Waiting on server
responses?

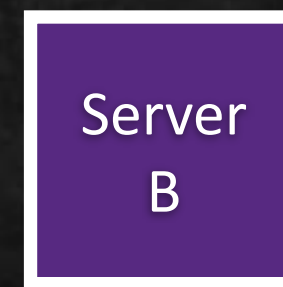
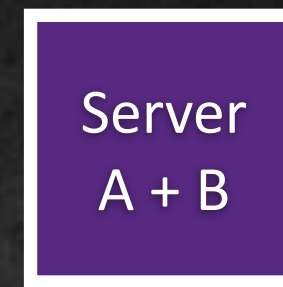
Are you out
of your mind?

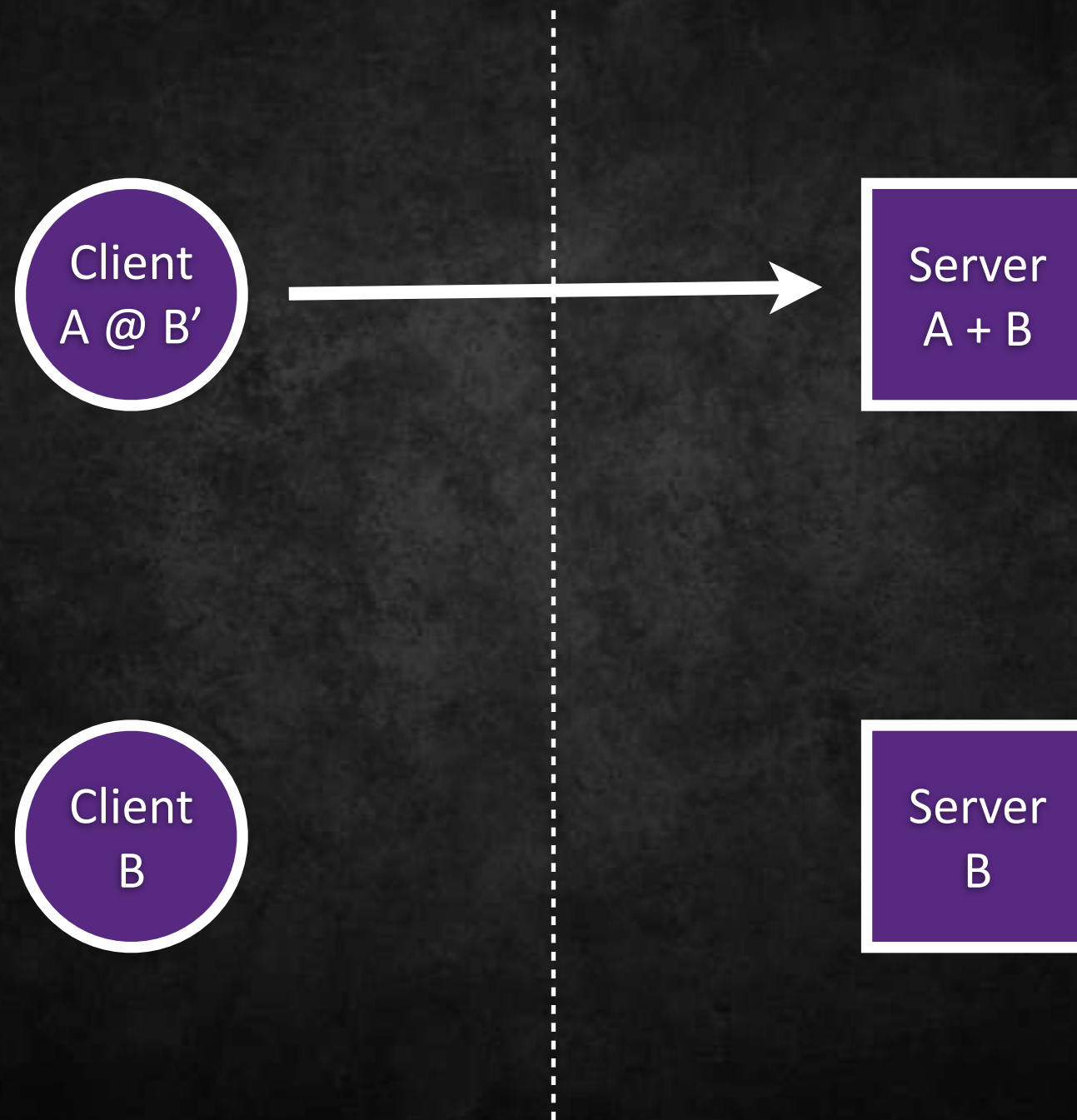
Schizophrenic

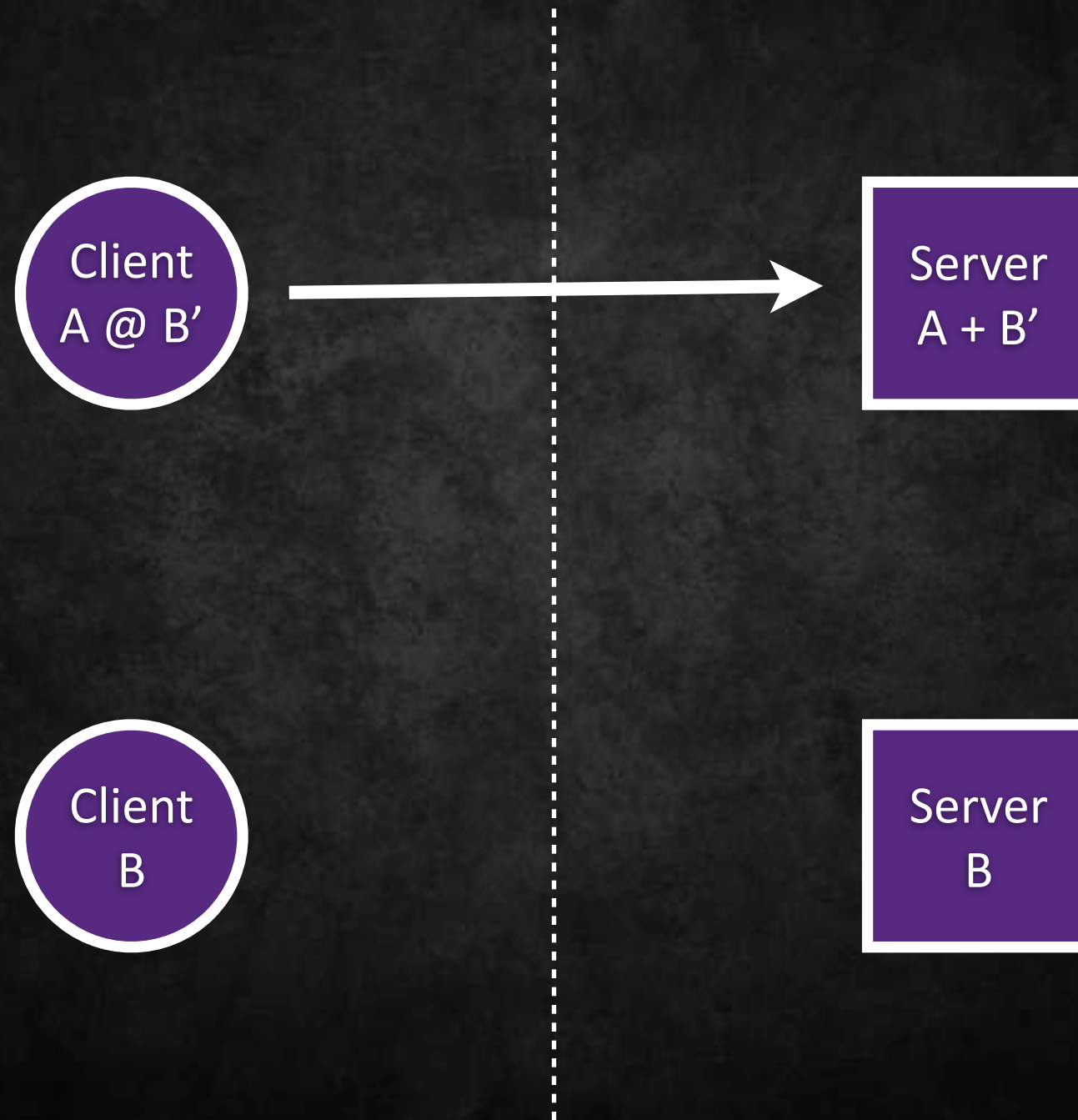


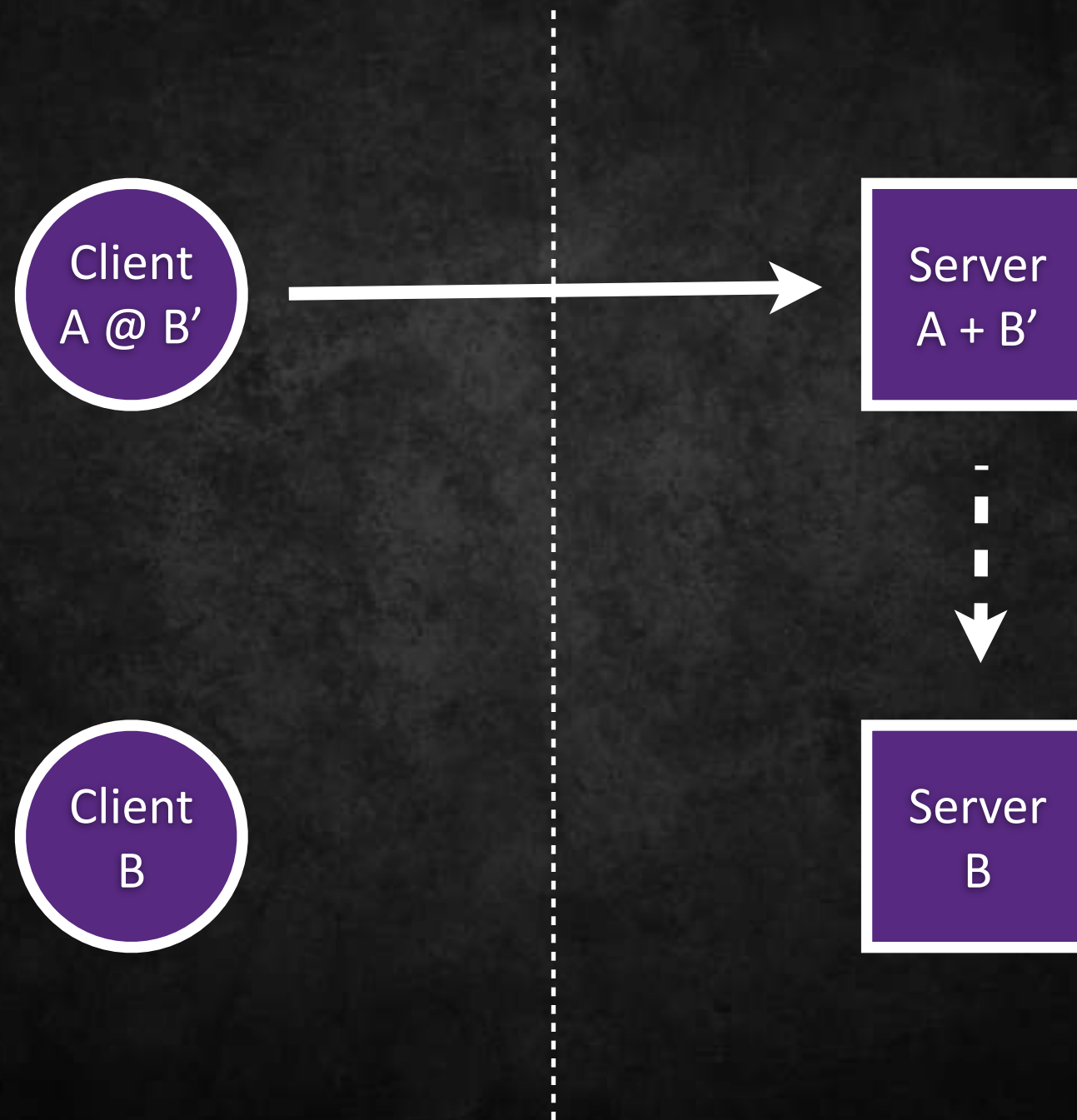


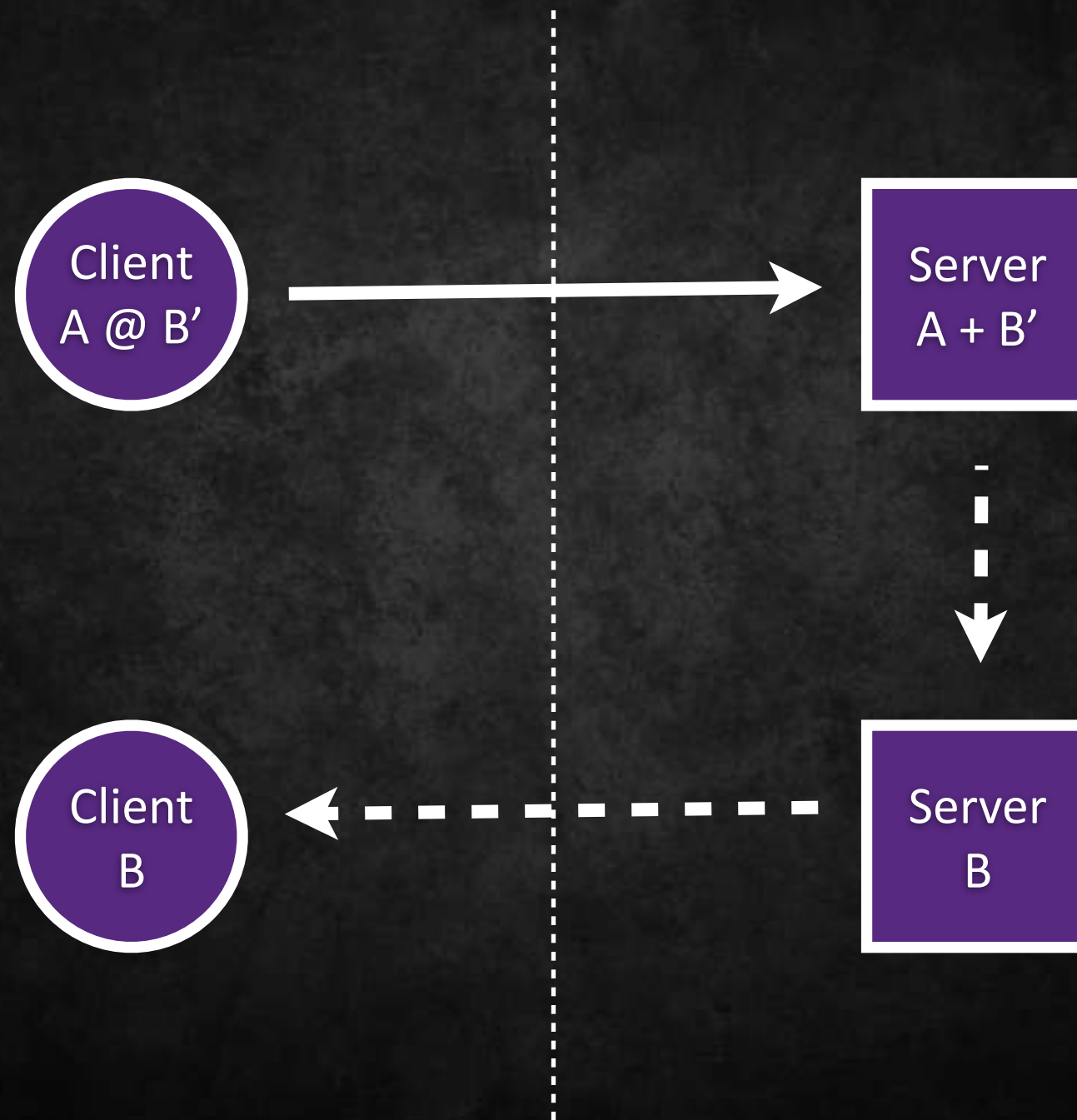


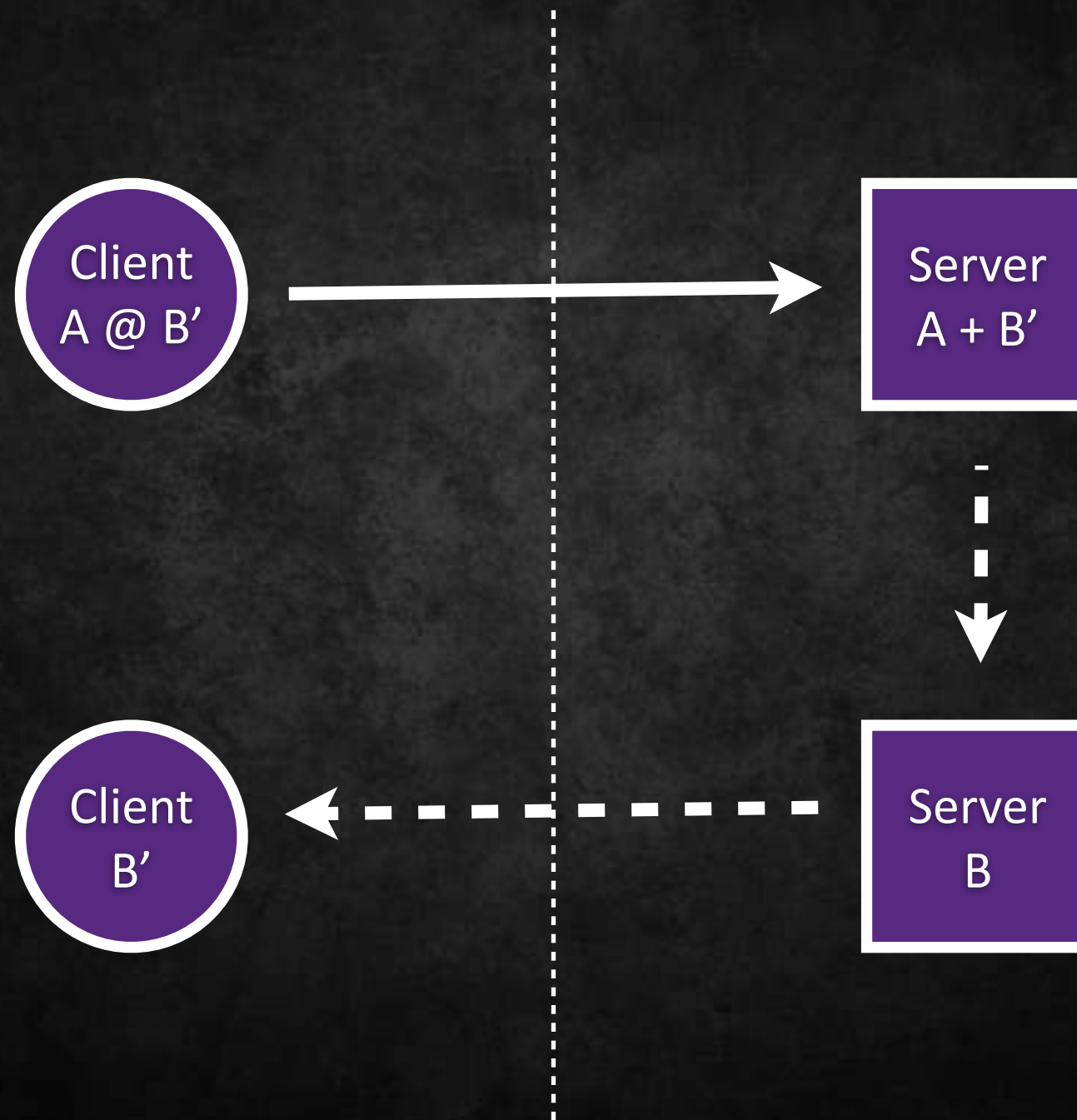


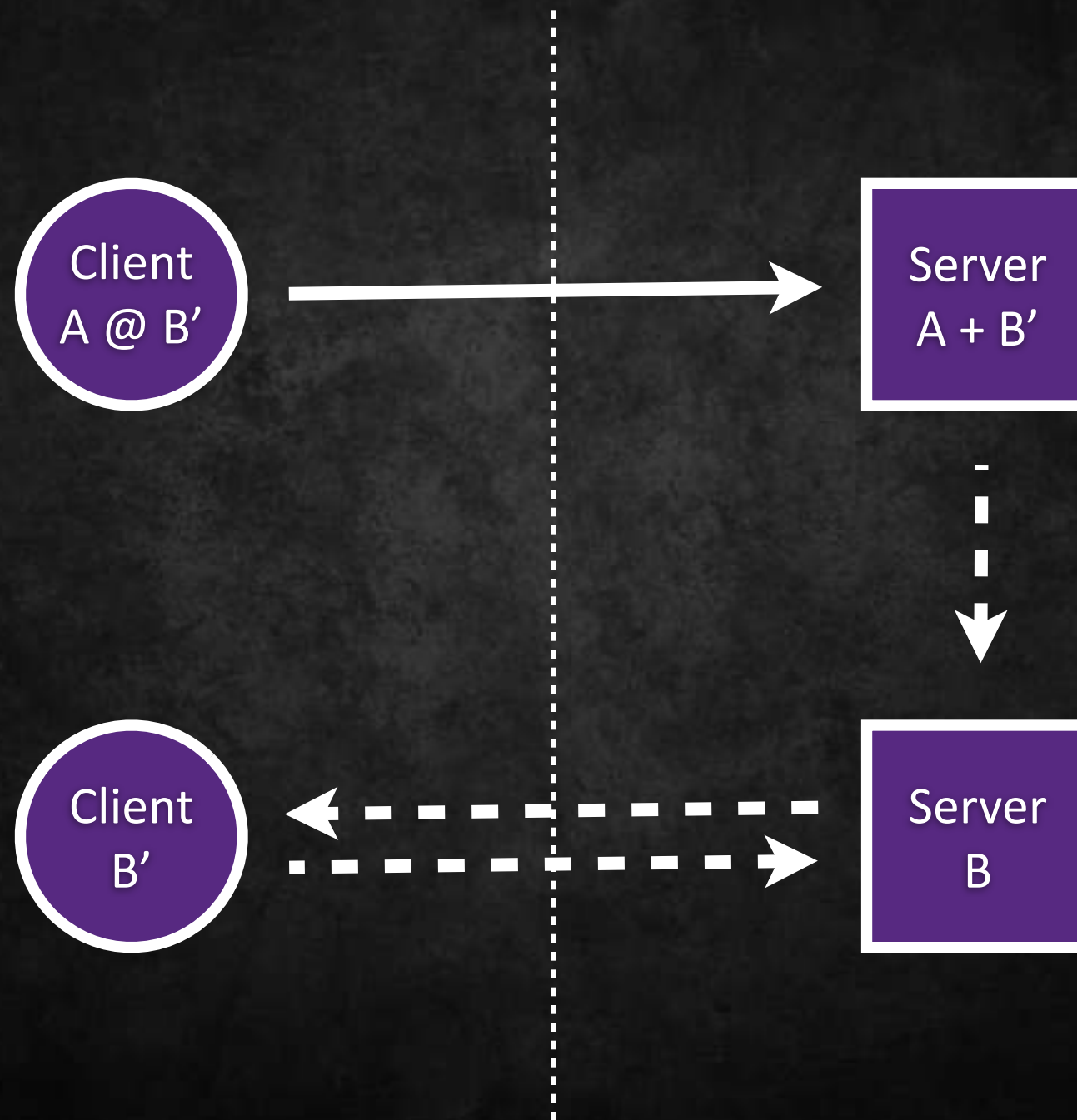


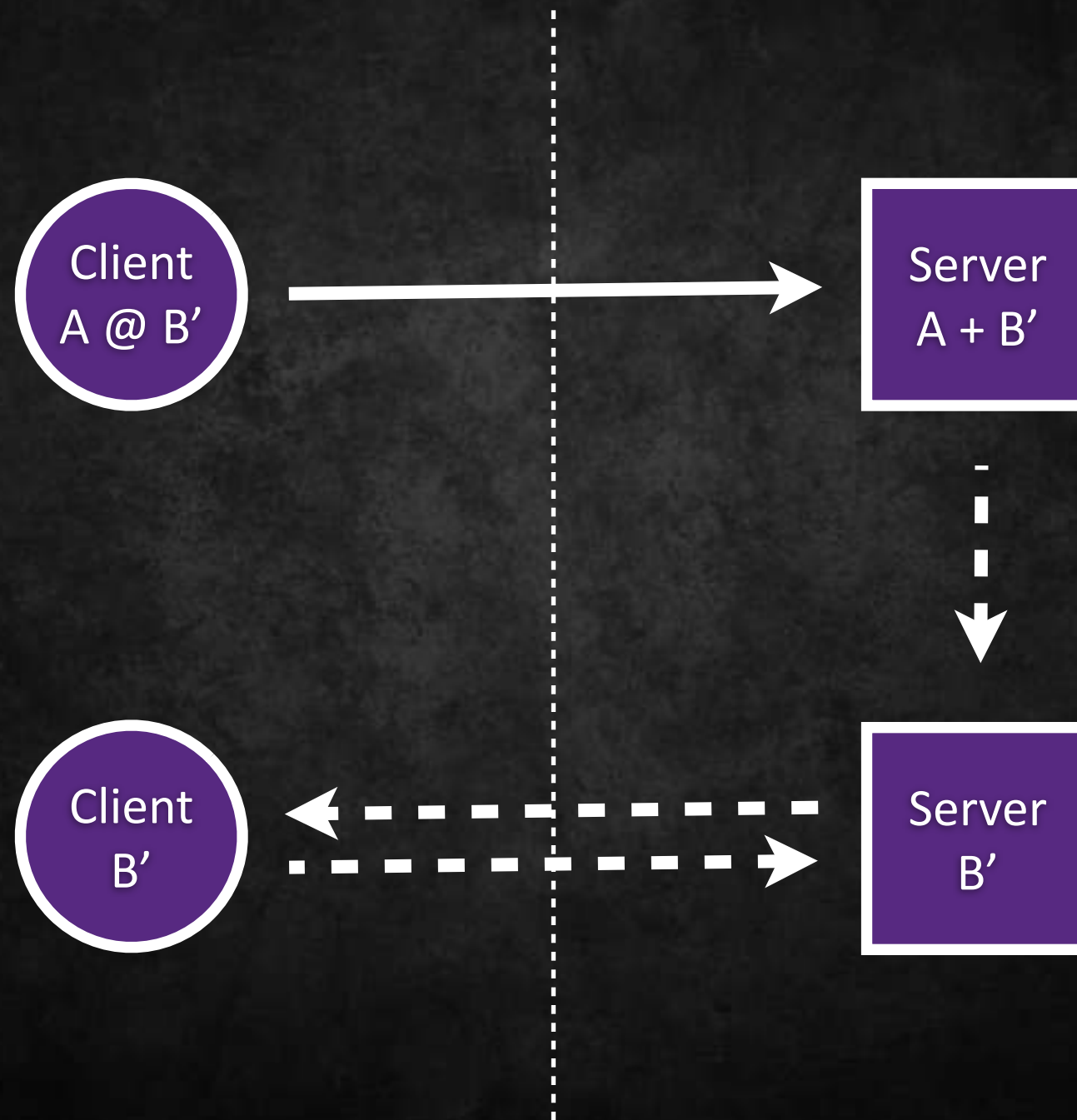














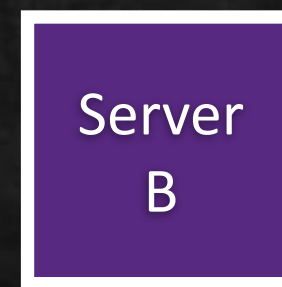
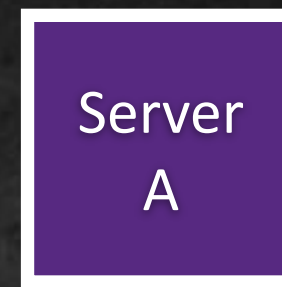
What about real interactions?

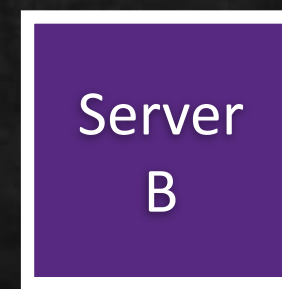
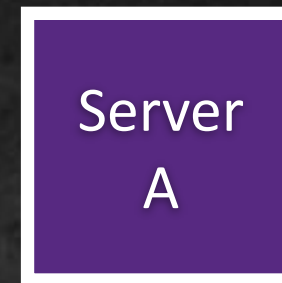


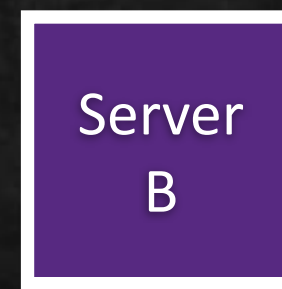
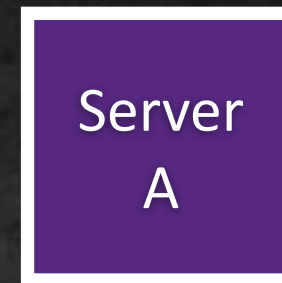
What about real
interactions?

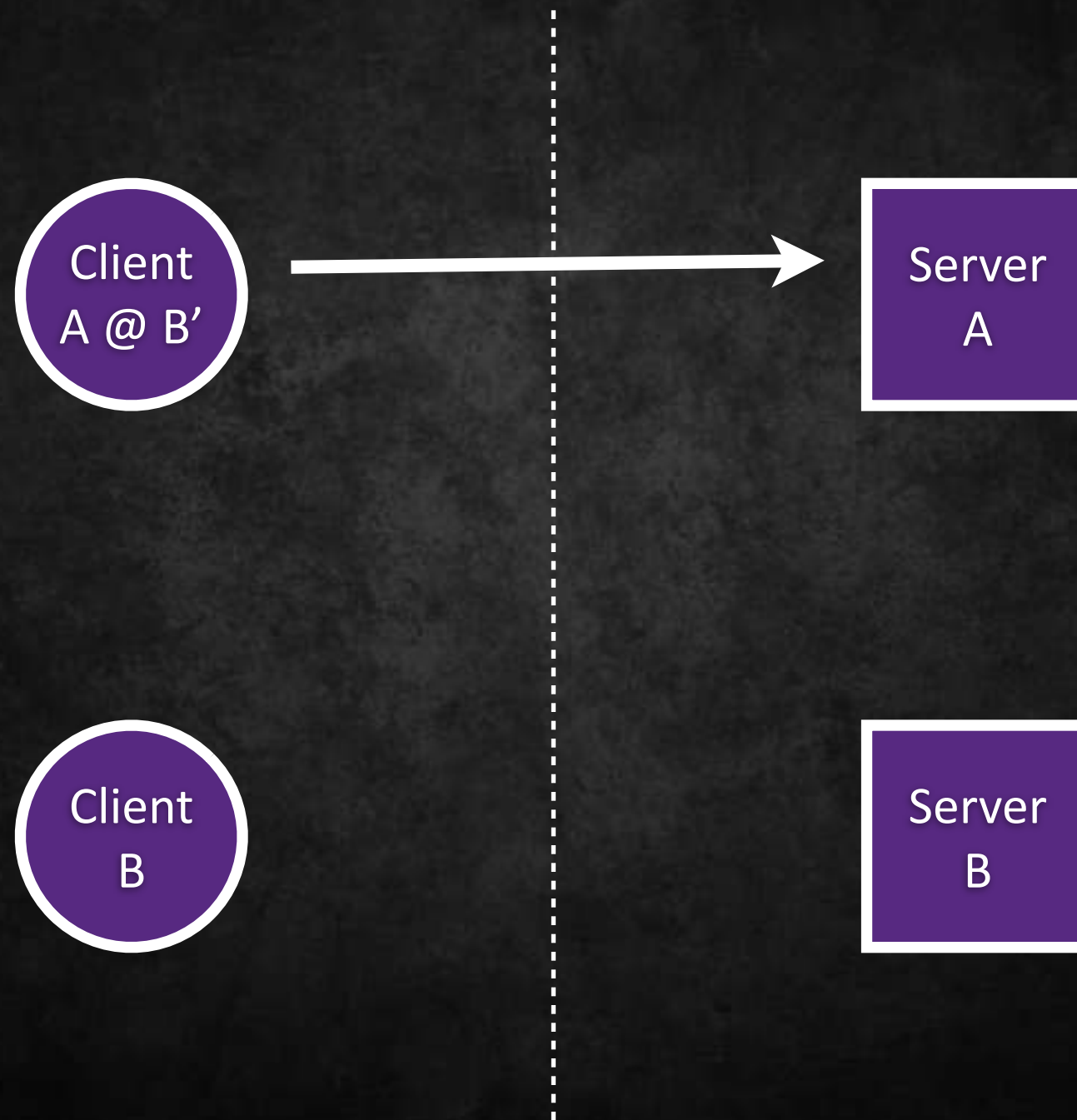
And what about
complexity
in the backend?

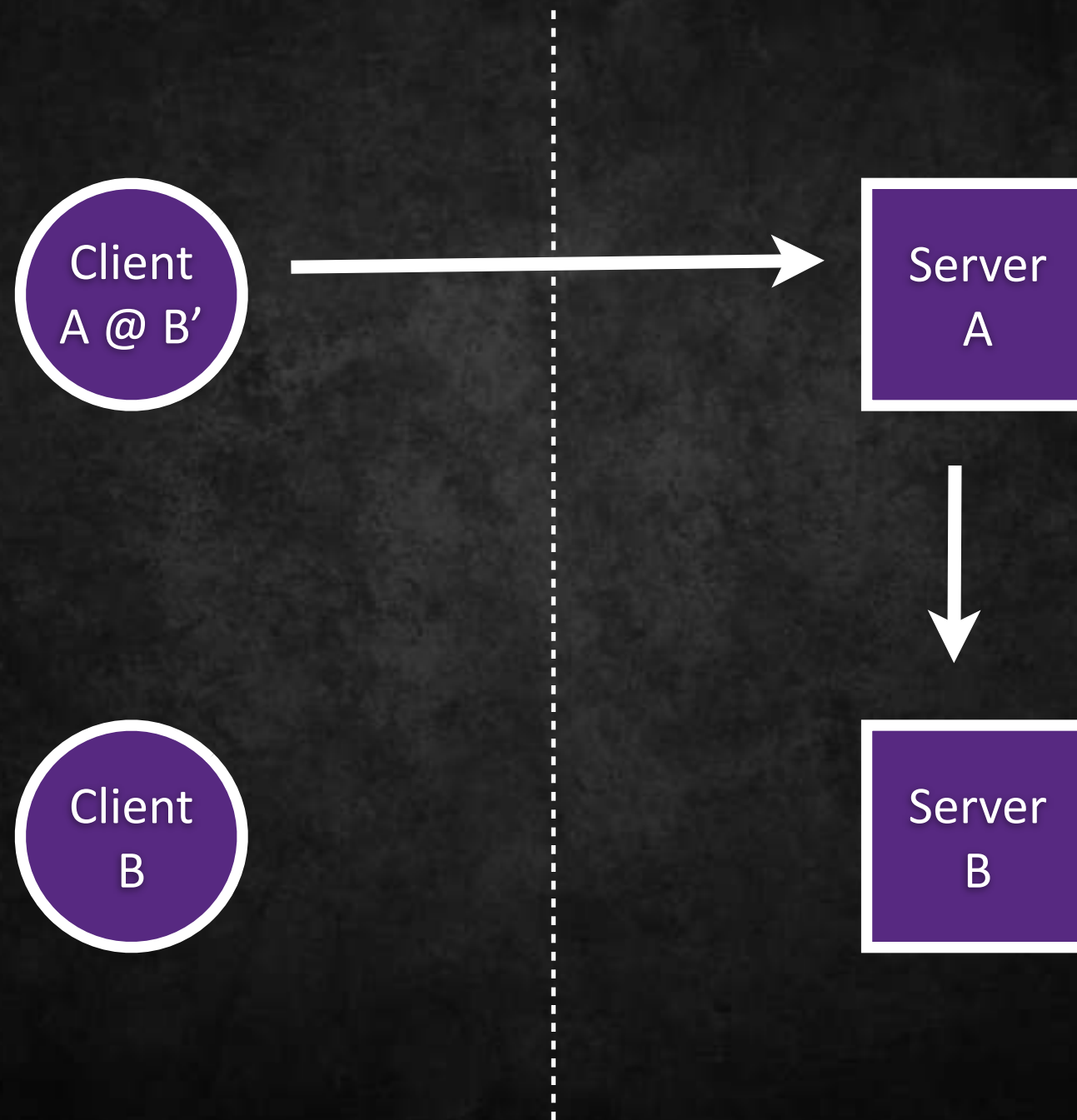
Optimistic

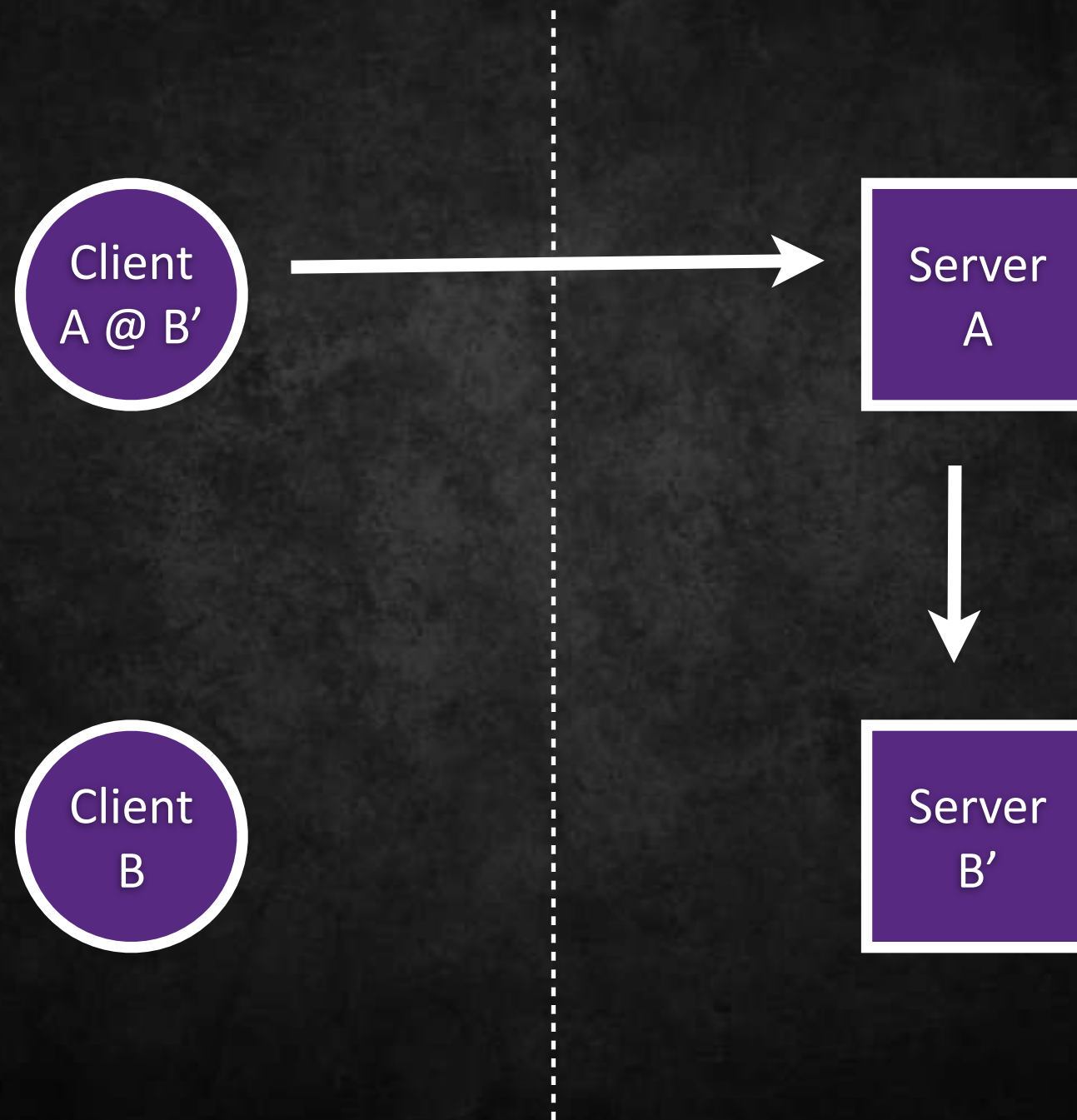


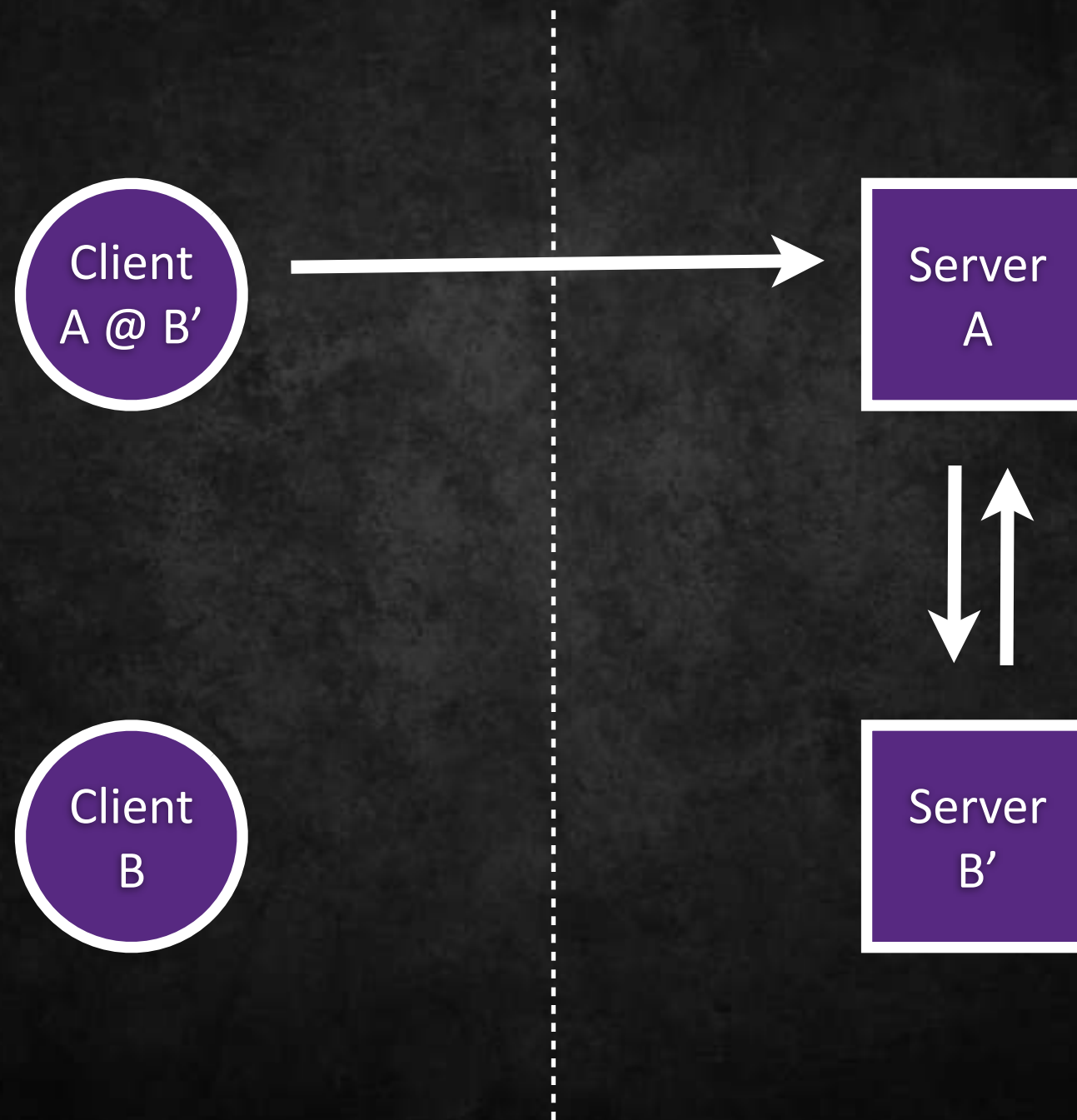


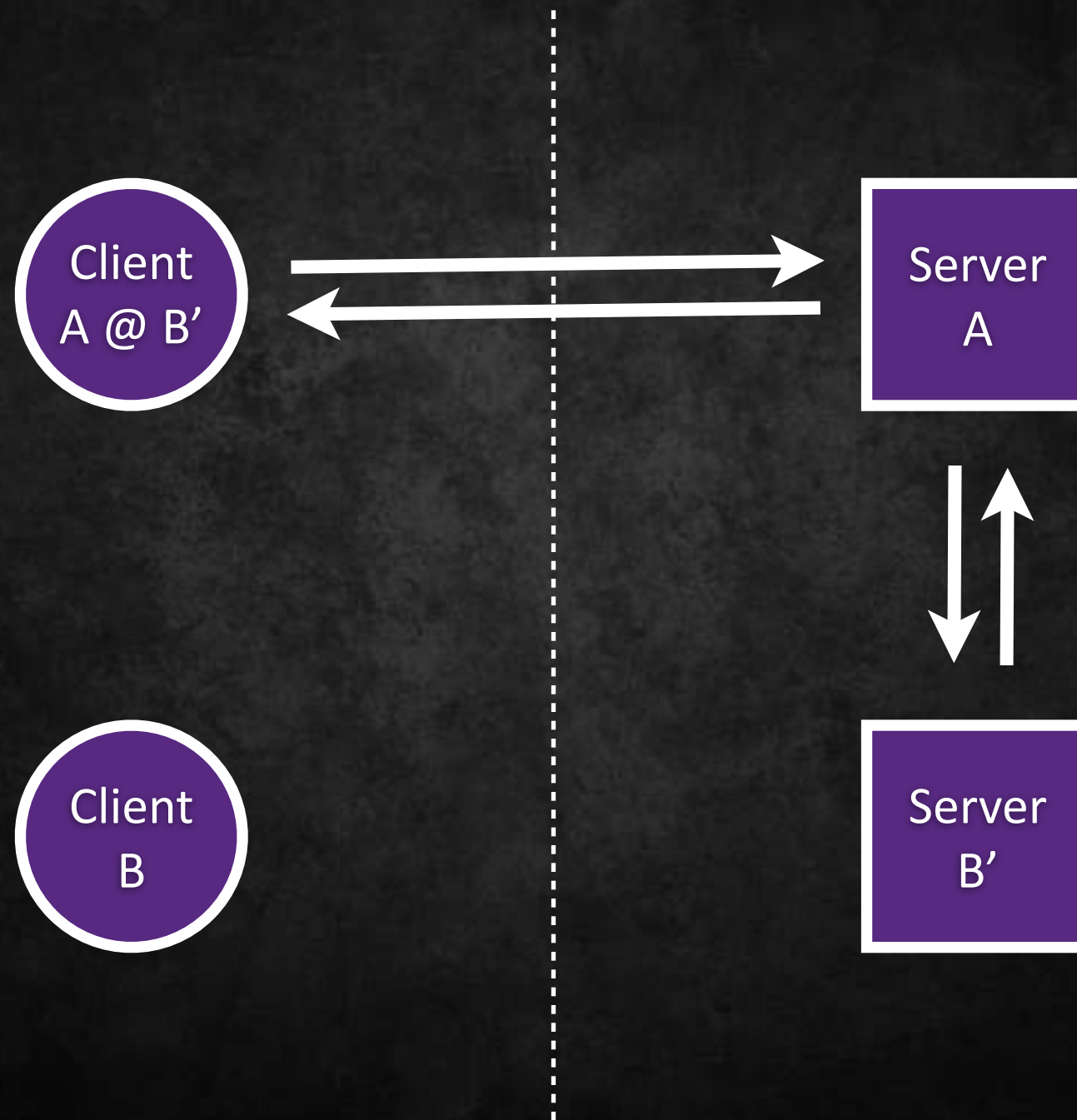


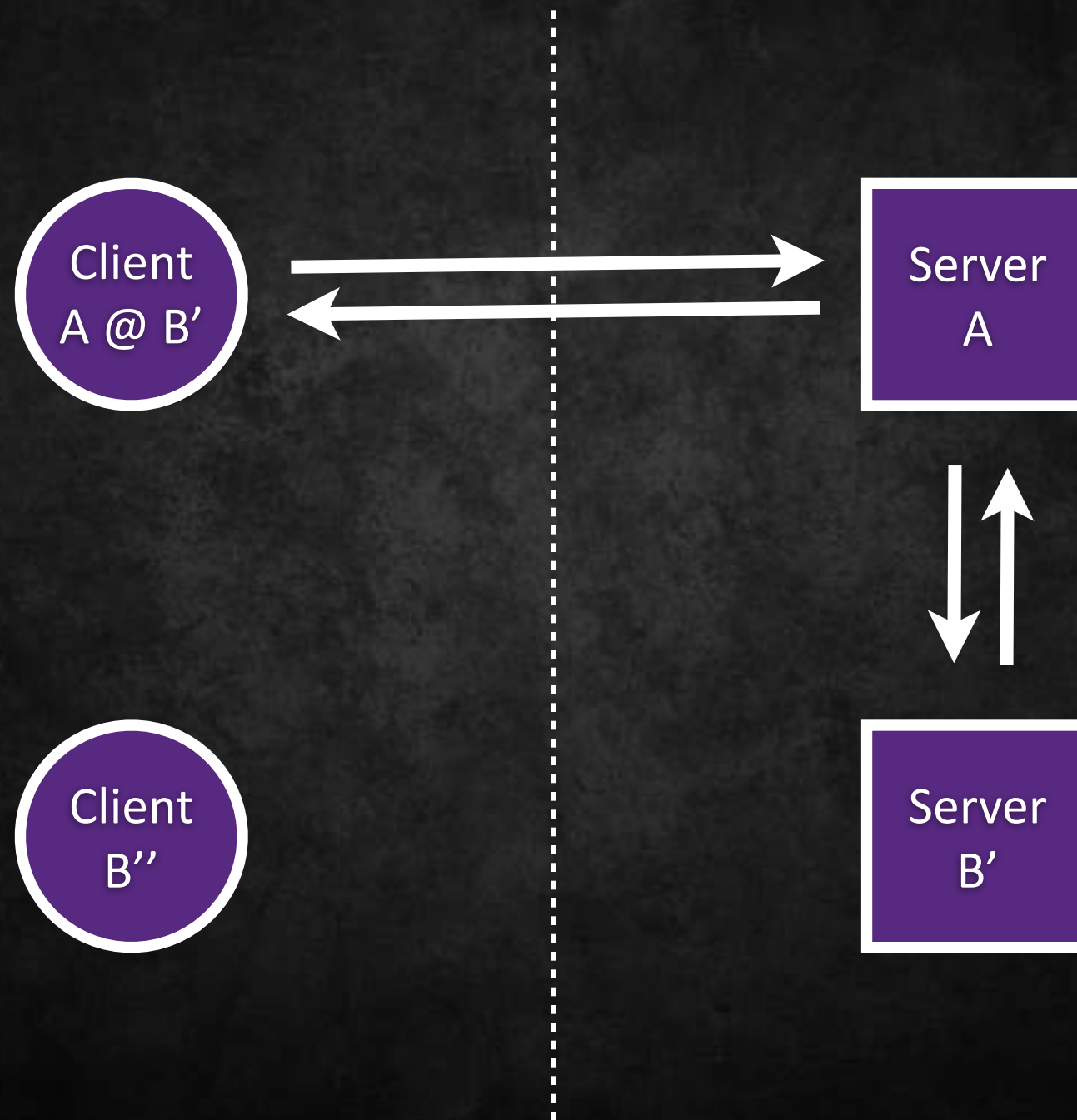


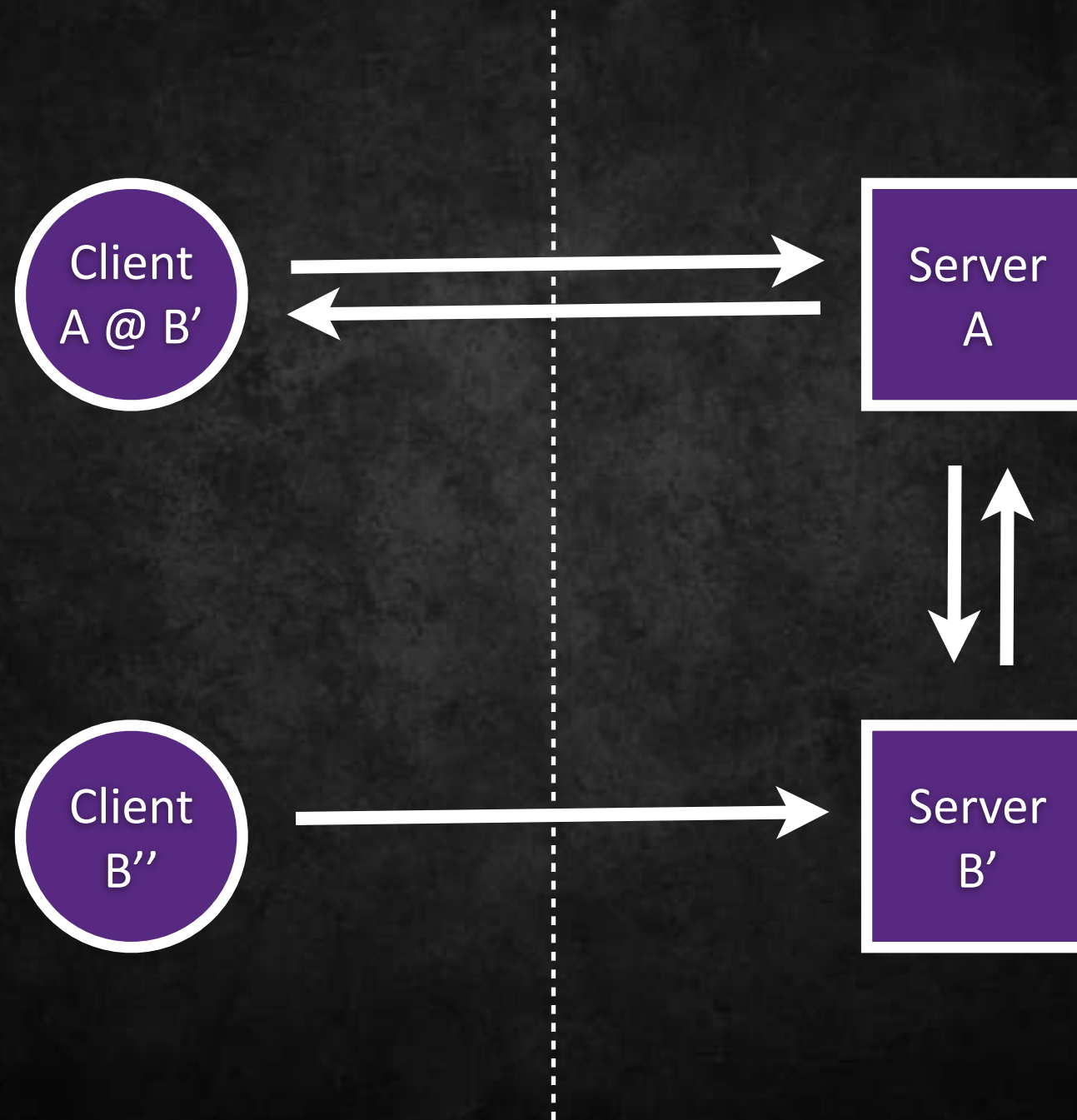


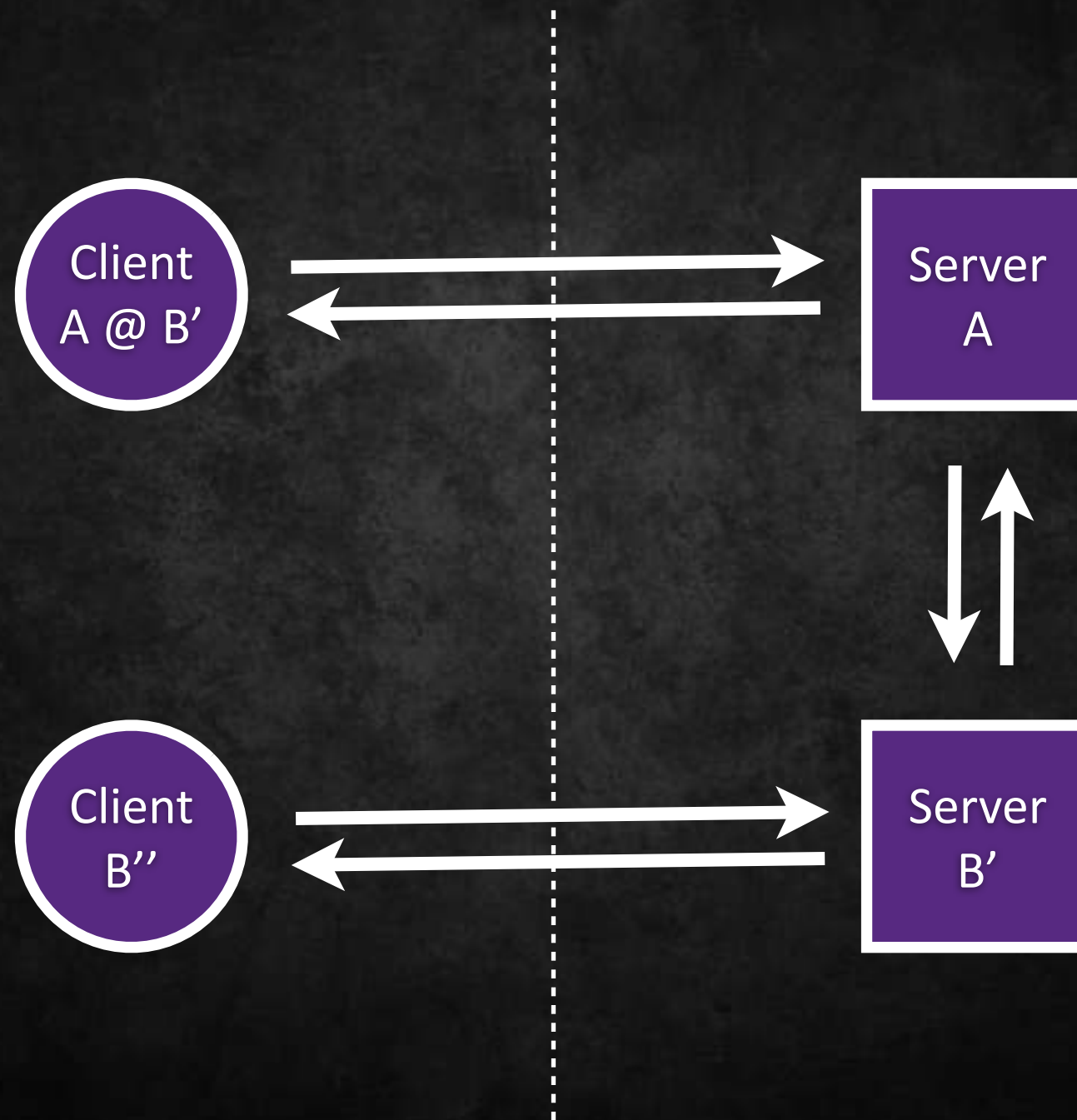


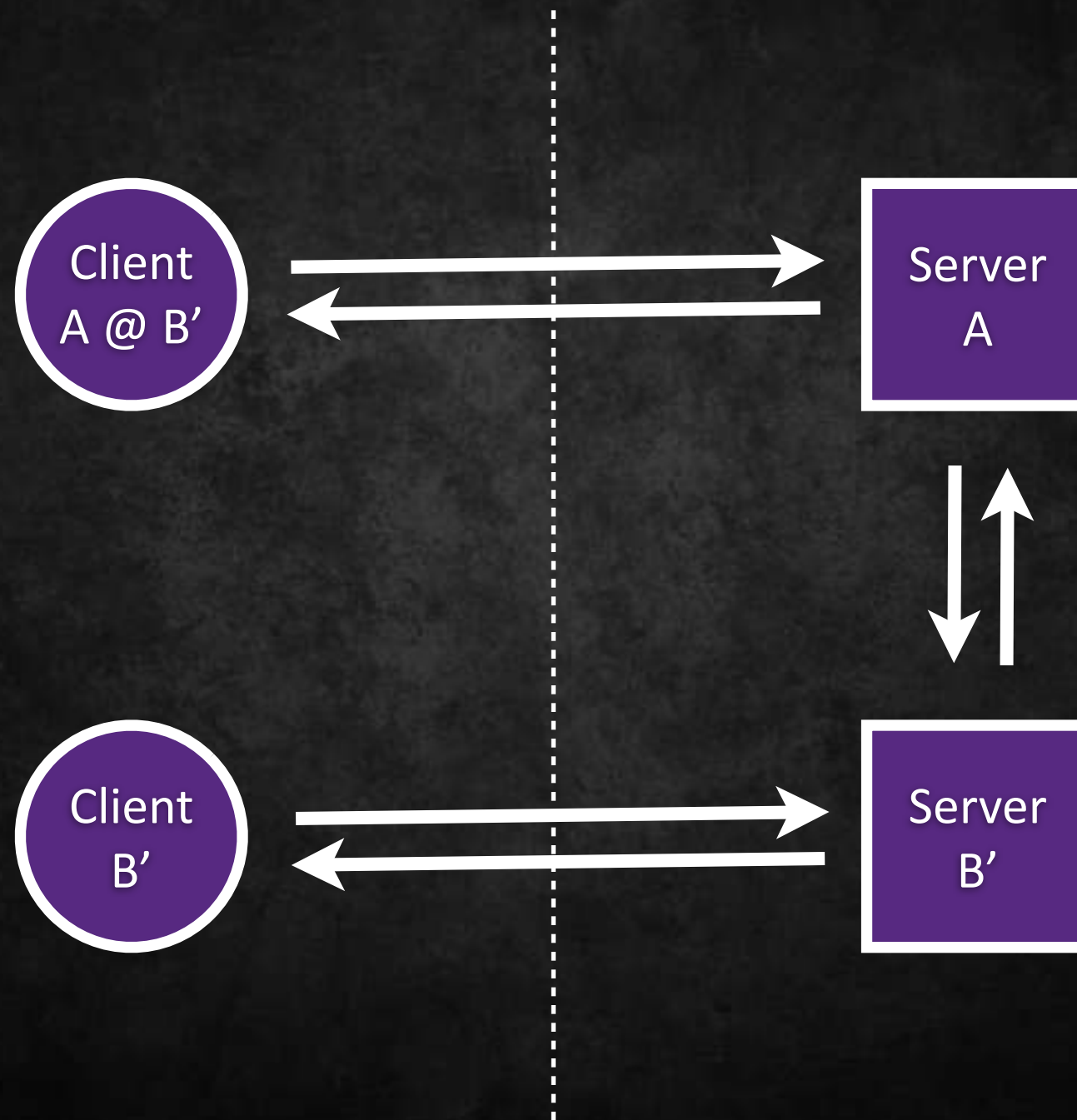














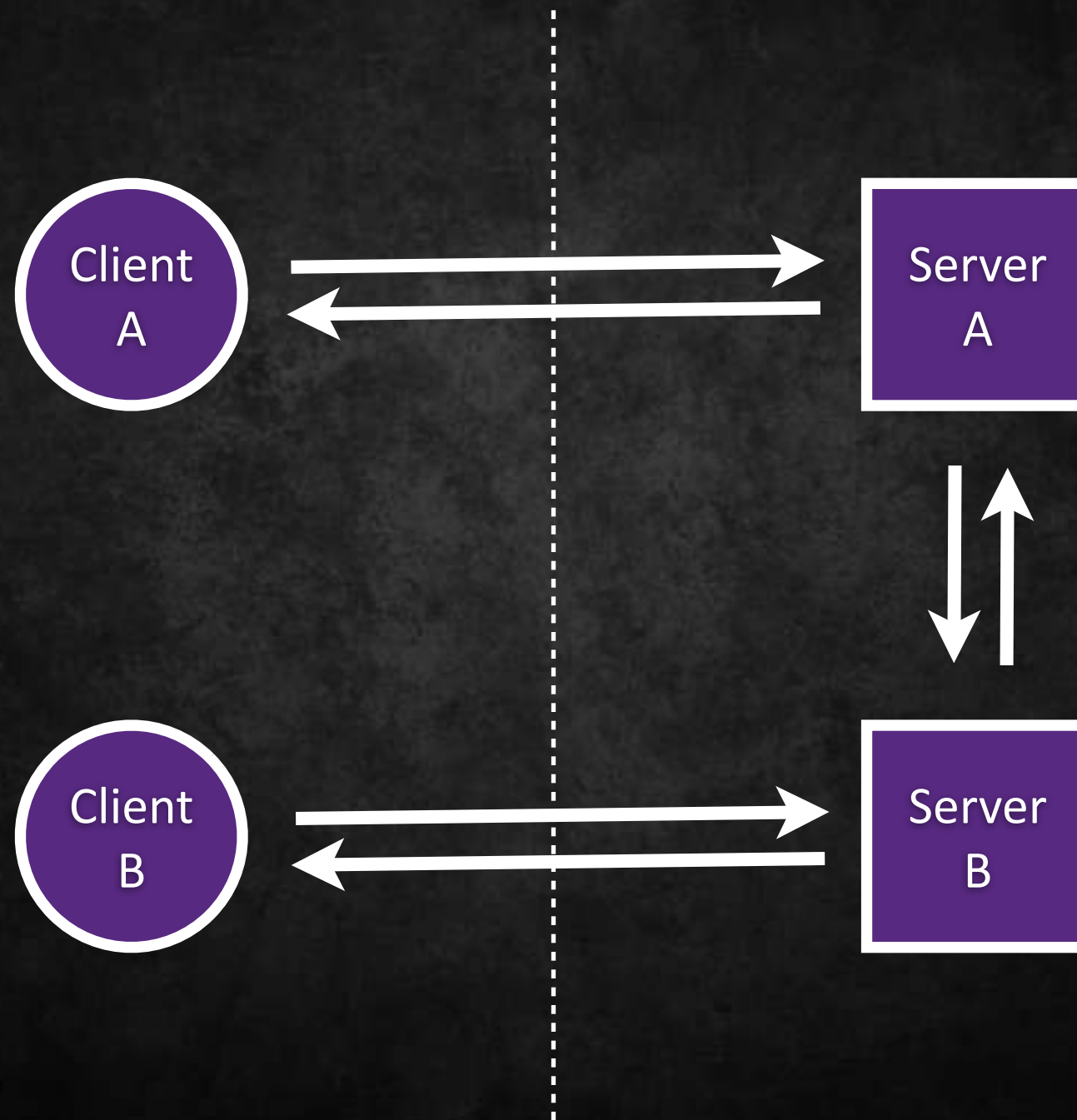
Hmm, client
rollbacks?

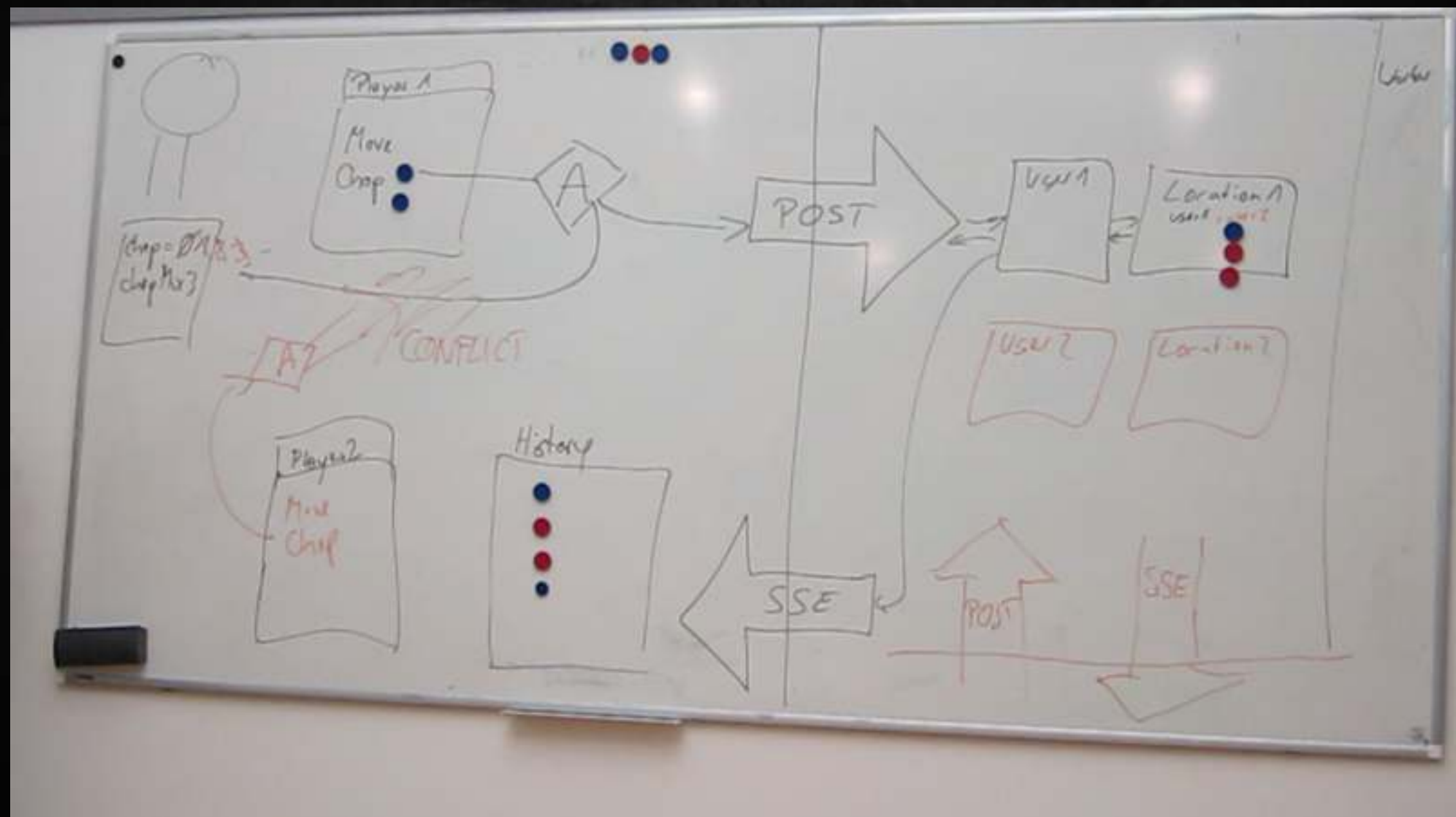


Hmm, client
rollbacks?

Are you sure you
can build that?

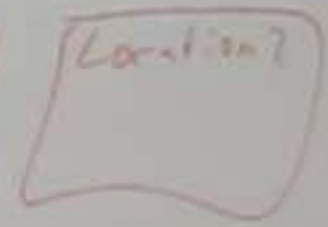
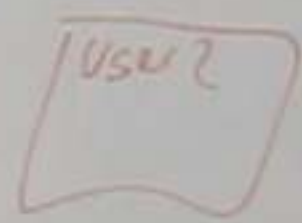
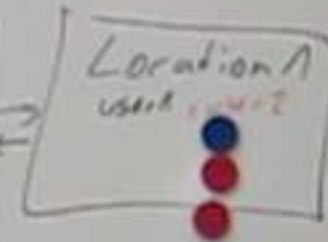
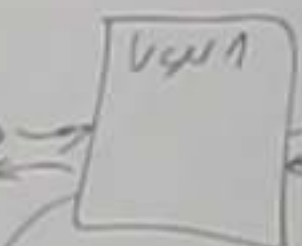
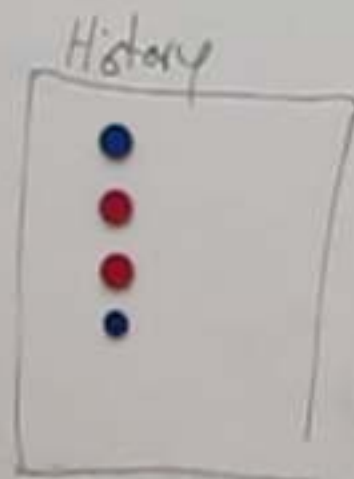
Thoughts
Client
Server
Operation







CONFLICT





Client A



state
count 0 max 3

Client A



state
count 0
max 3

user A
move

Client A



state
count 0
max 3

user A
move
pick 

Client A



user A

move
pick

state

count 1
max 3

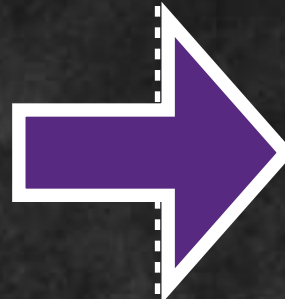


Client A



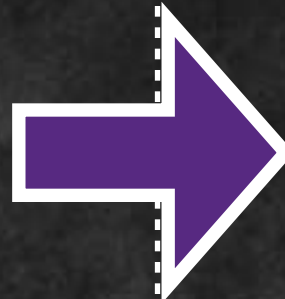
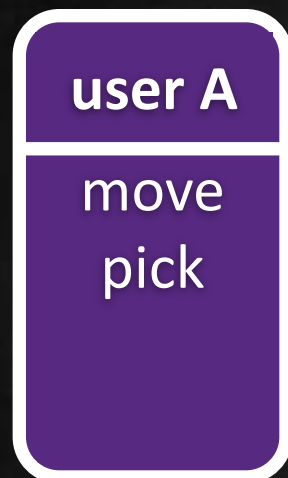
state
count 1
max 3

user A
move
pick



Client A

Server



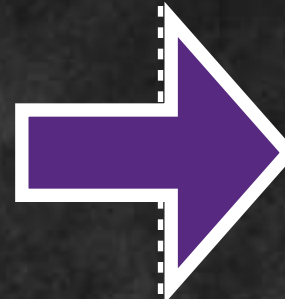
Client A

Server



state
count 1 max 3

user A
move pick



user A

location
count 0

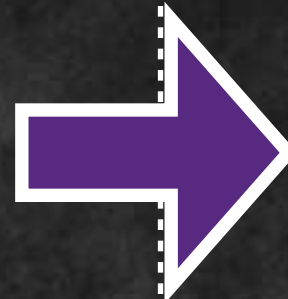
Client A

Server



state
count 1 max 3

user A
move pick



user A
location
count 1

Client A

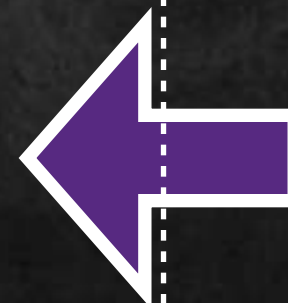
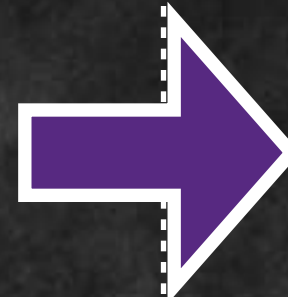
Server



state
count 1 max 3

user A
move pick

Client A



user A

location
count 1



Server



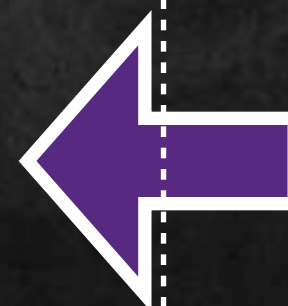
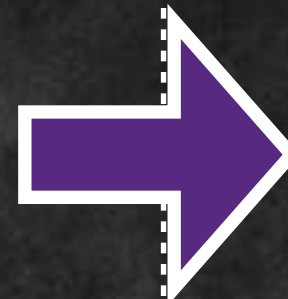
state
count 1 max 3

user A
move pick

history
move A pick A



Client A



user A

location
count 1

Server

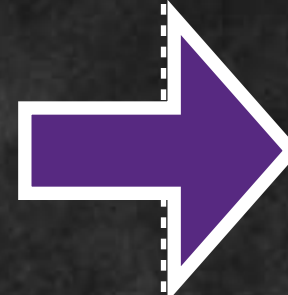


state
count 1 max 3

user A
move pick

history
move A pick A

Client A



user A

location
count 1

Server

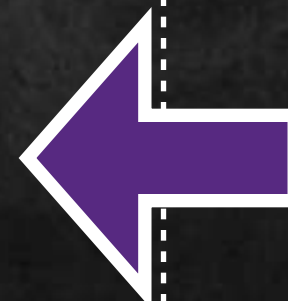
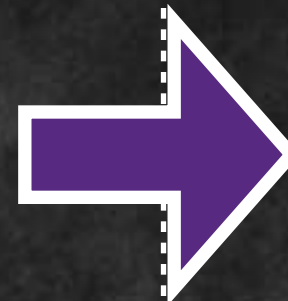


state
count 1
max 3

user A
move
pick

history
move A
pick A

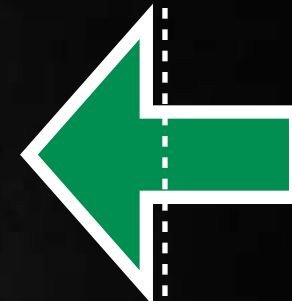
Client A

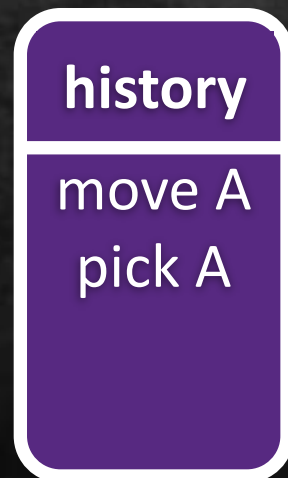
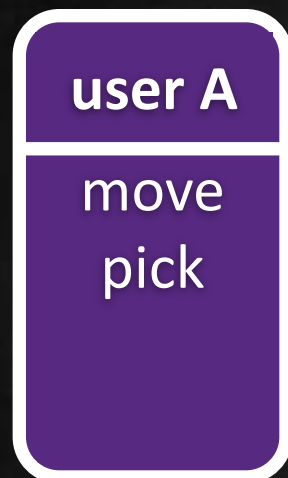


user A

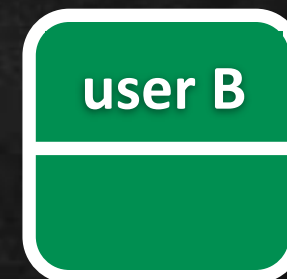
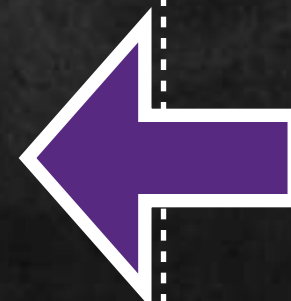
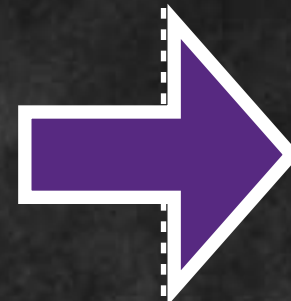
location
count 1

Server

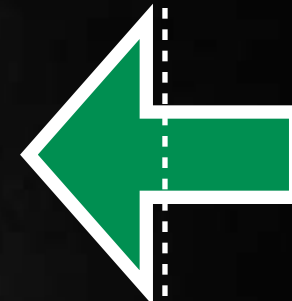


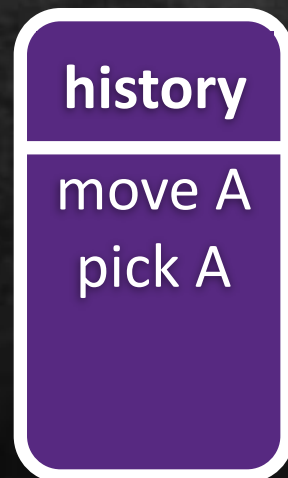
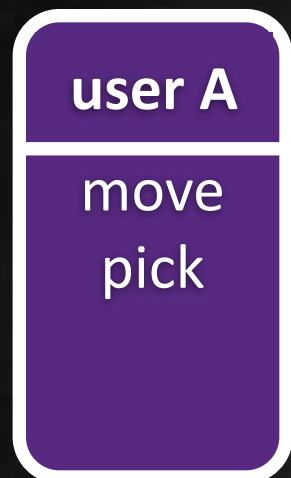


Client A

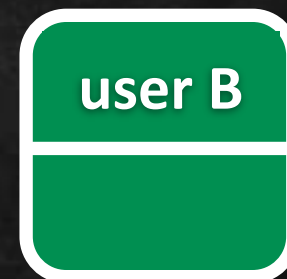
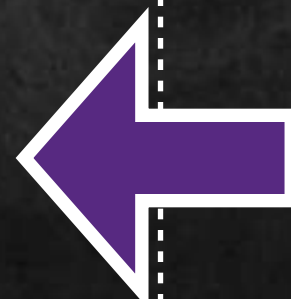
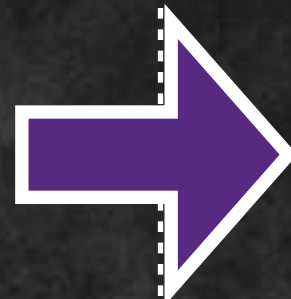


Server

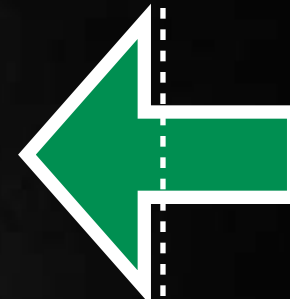


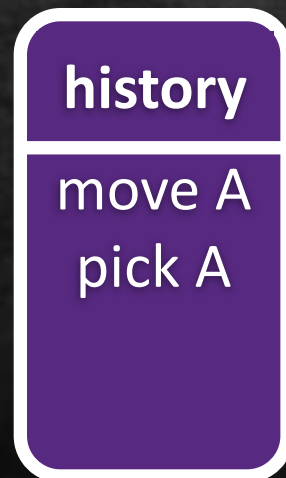
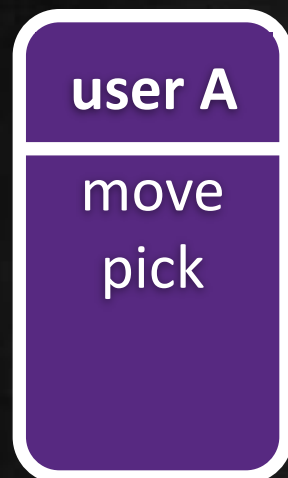


Client A

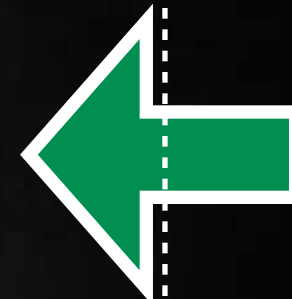
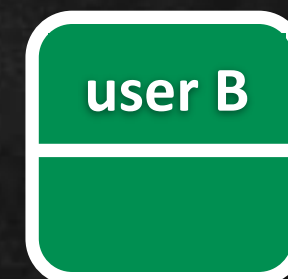
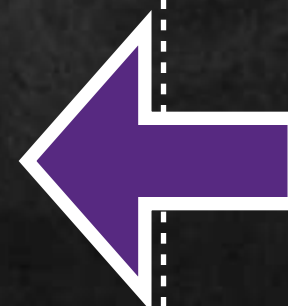
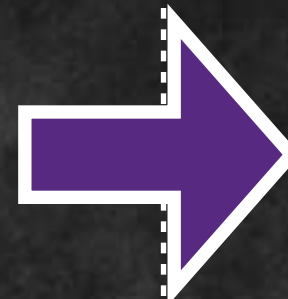


Server

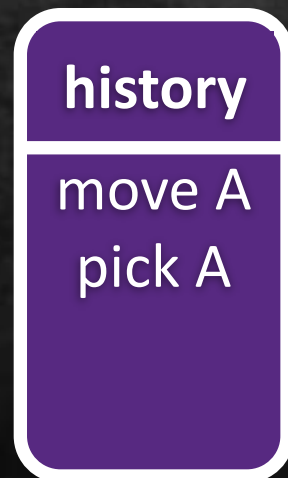
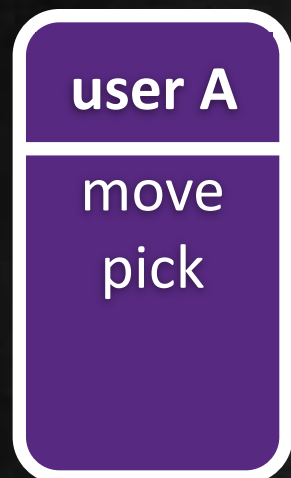




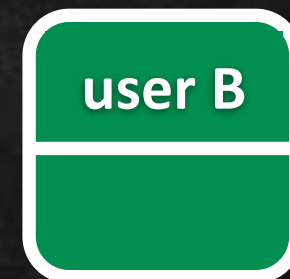
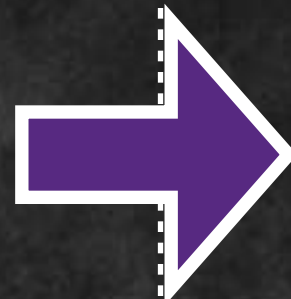
Client A



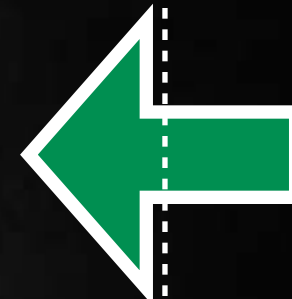
Server

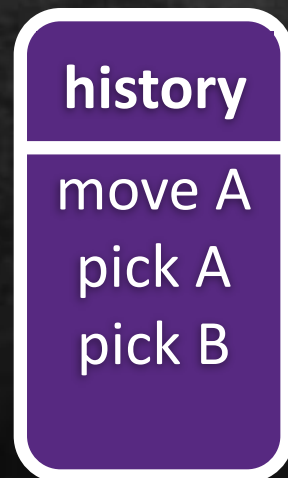
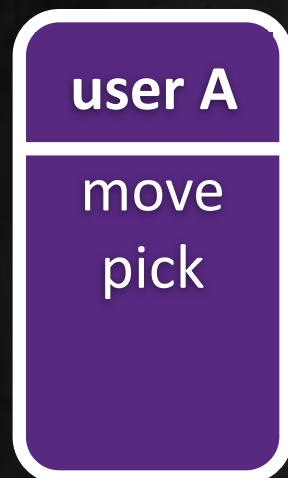


Client A

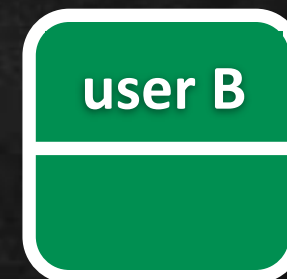
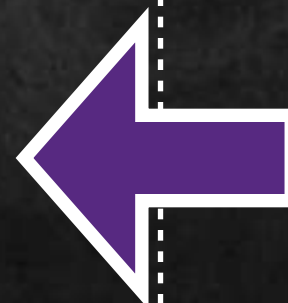
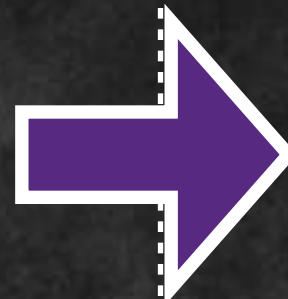


Server

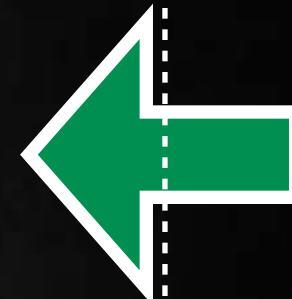


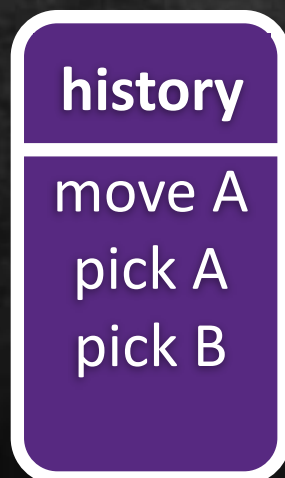
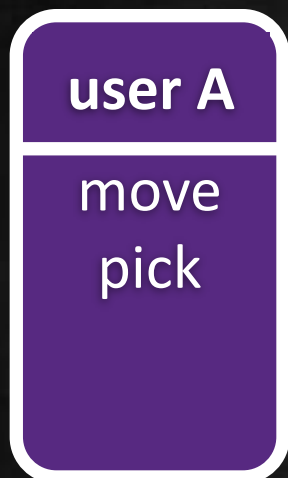


Client A

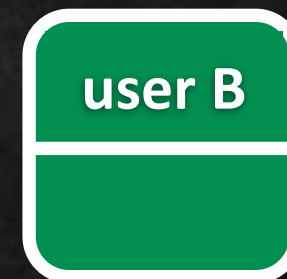
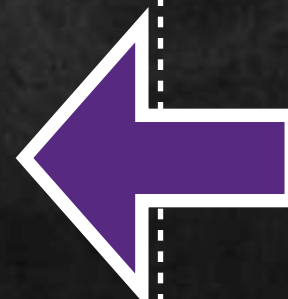
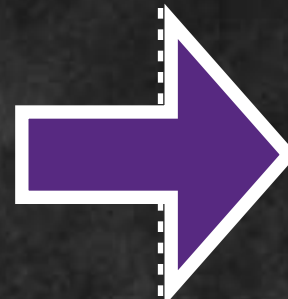


Server

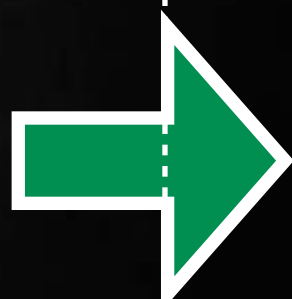
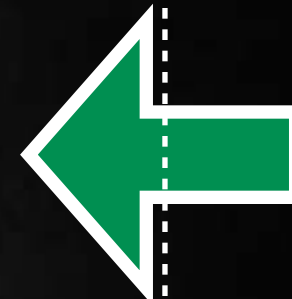




Client A



Server





state
count 2
max 3

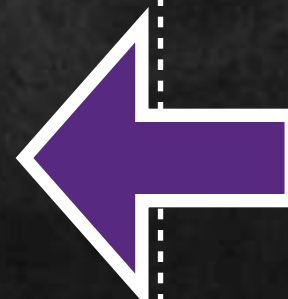
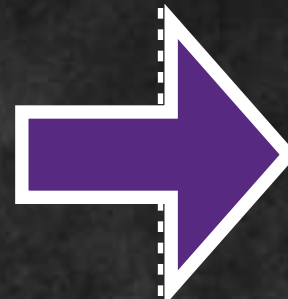


user A
move
pick

user B
pick

history
move A
pick A
pick B

Client A

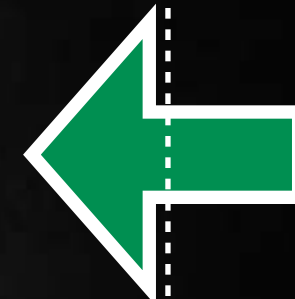


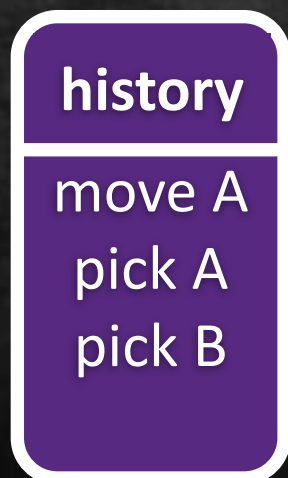
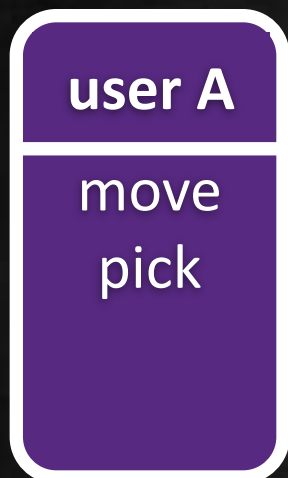
user A

location
count 2

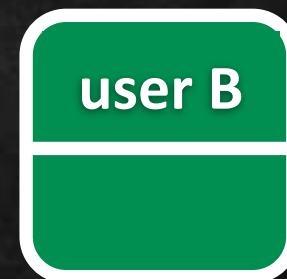
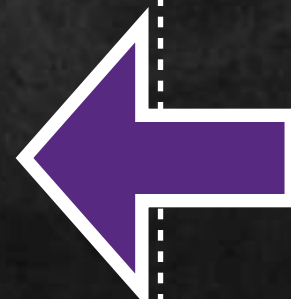
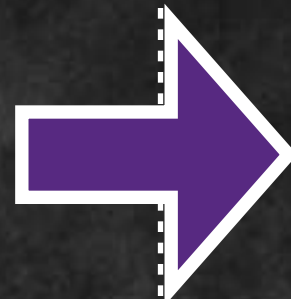
user B

Server

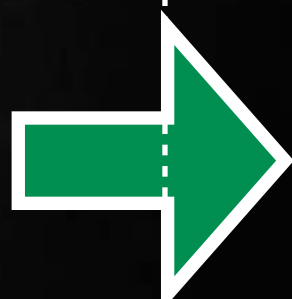
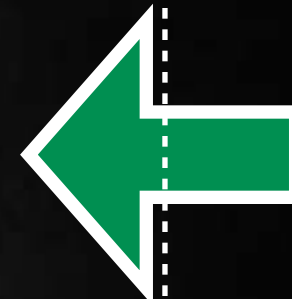


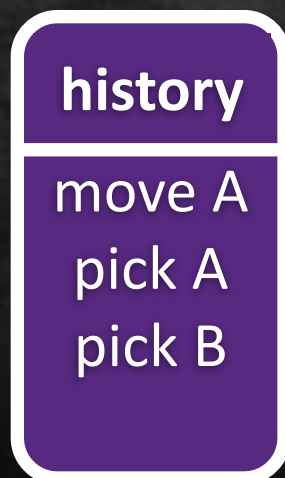


Client A

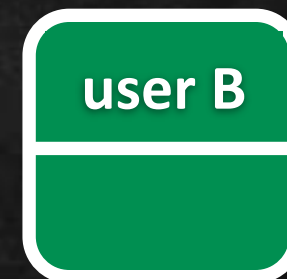
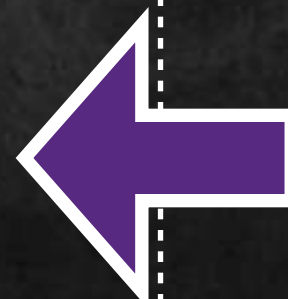
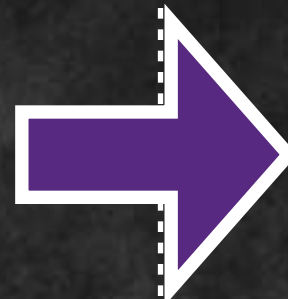


Server

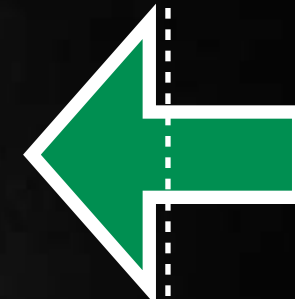




Client A



Server





state
count 3
max 3

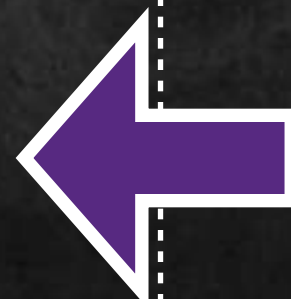
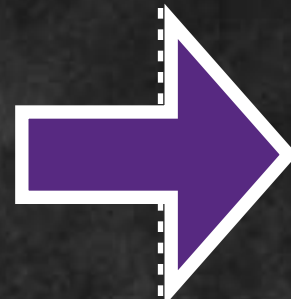


user A
move
pick
pick

user B
pick

history
move A
pick A
pick B

Client A

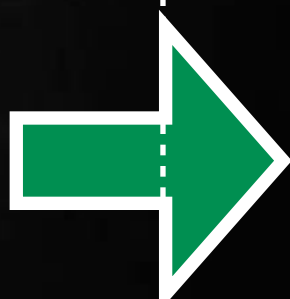
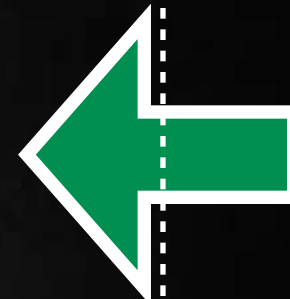


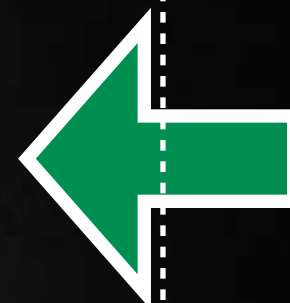
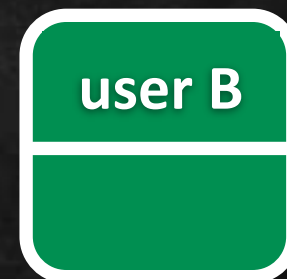
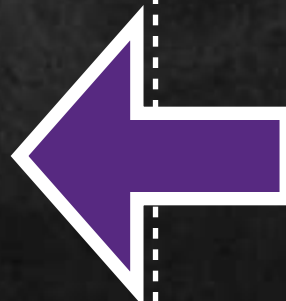
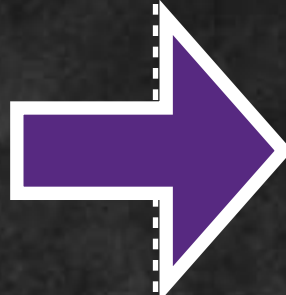
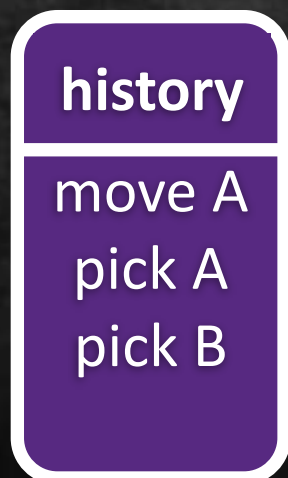
user A

location
count 2

user B

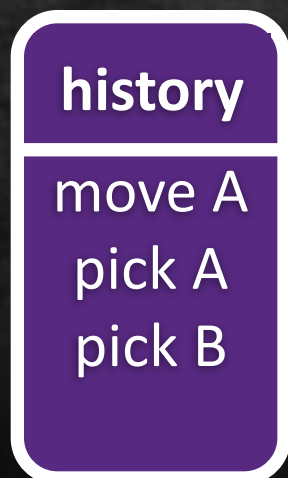
Server



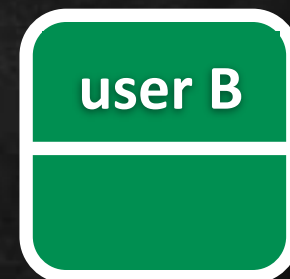
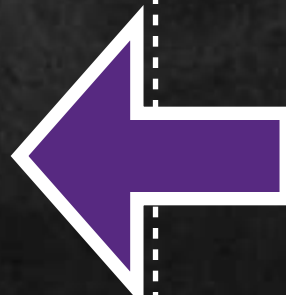
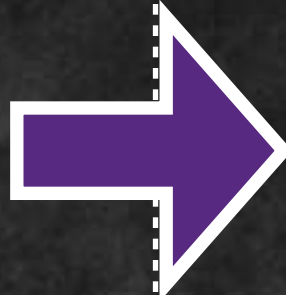


Client A

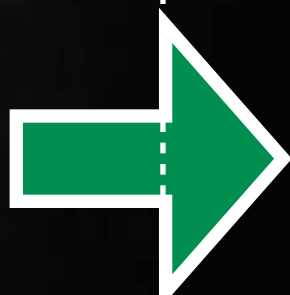
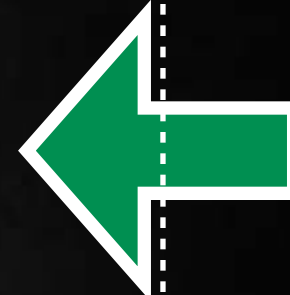
Server



Client A



Server





state

count 3
max 3

user A

move
pick
pick

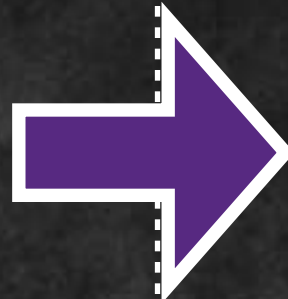
user B

pick

history

move A
pick A
pick B

Client A

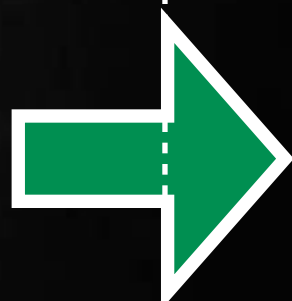
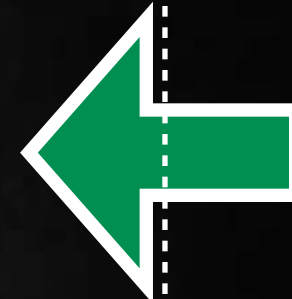


user A

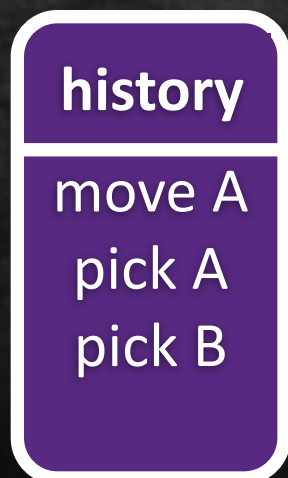
location

count 3

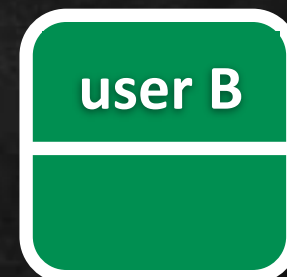
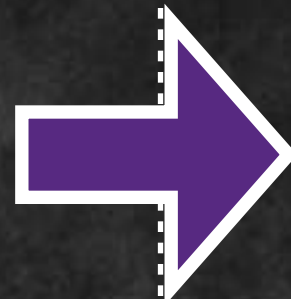
user B



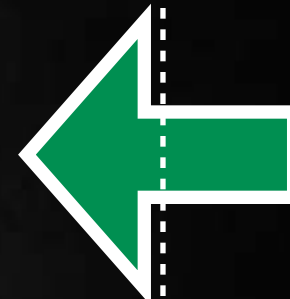
Server

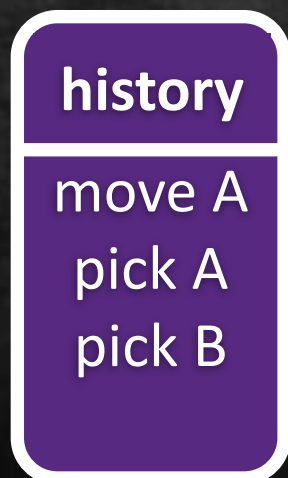


Client A

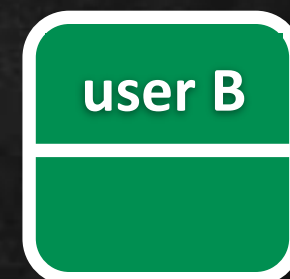
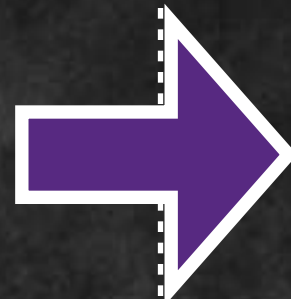


Server

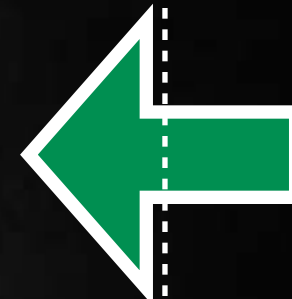


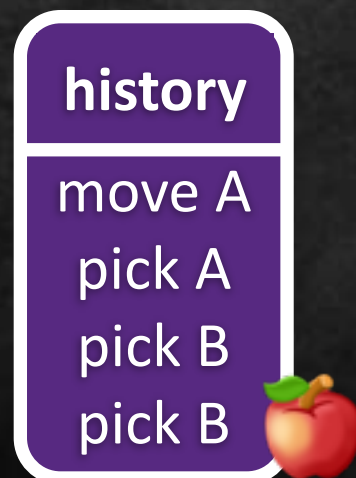


Client A

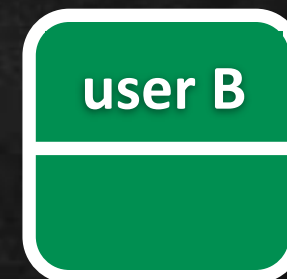
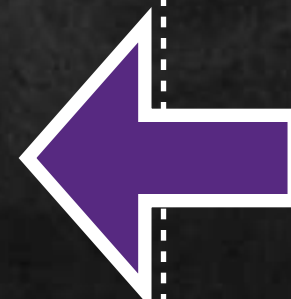
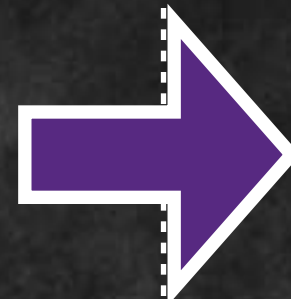


Server

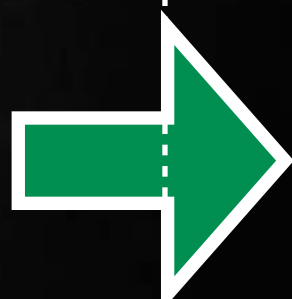
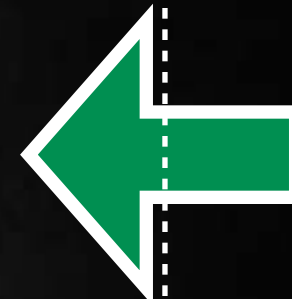


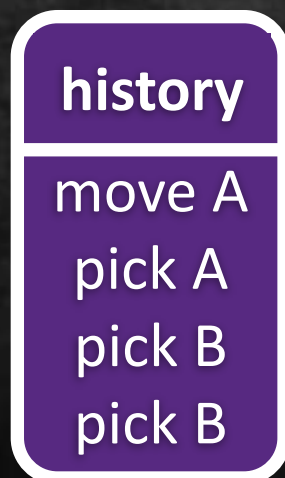
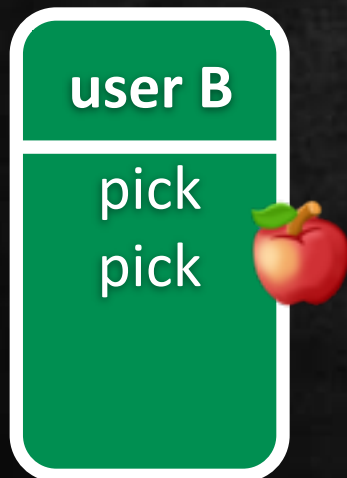


Client A

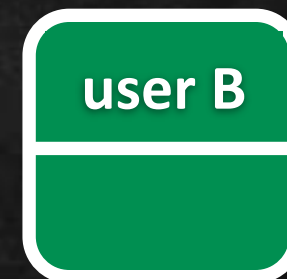
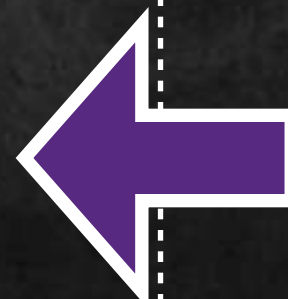
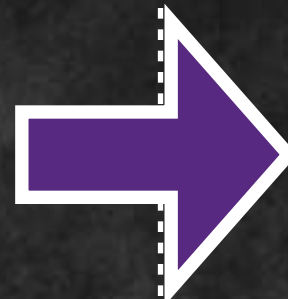


Server

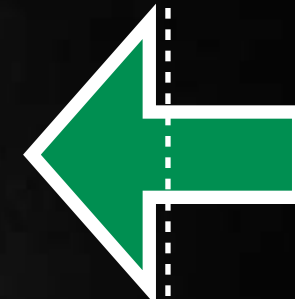




Client A



Server





state
count 3
max 3

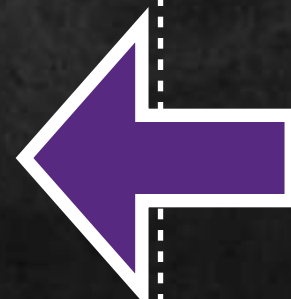
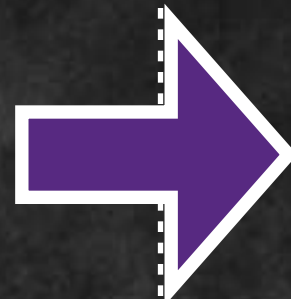


user A
move
pick
pick

user B
pick
pick

history
move A
pick A
pick B
pick B

Client A

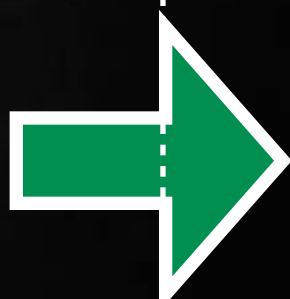
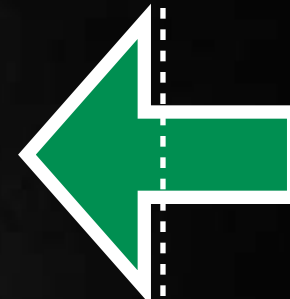


user A

location
count 3

user B

Server





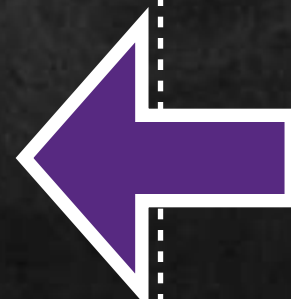
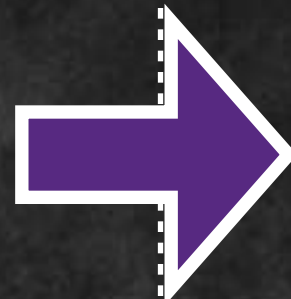
state
count 3
max 3

user A
move
pick
pick

user B
pick
pick

history
move A
pick A
pick B
pick B

Client A

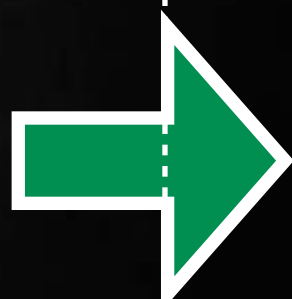
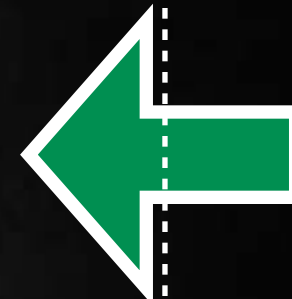


user A

location
count 3

user B

Server





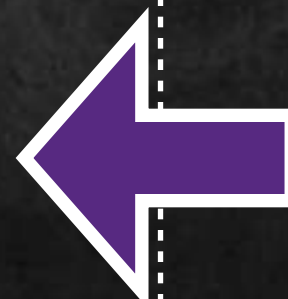
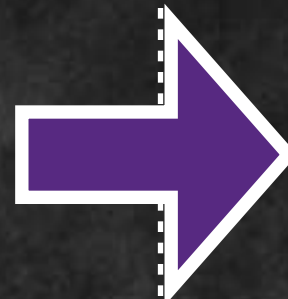
state
count 3
max 3

user A
move
pick
pick

user B
pick
pick

history
move A
pick A
pick B
pick B

Client A

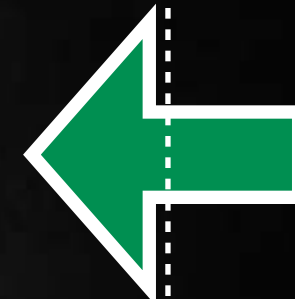


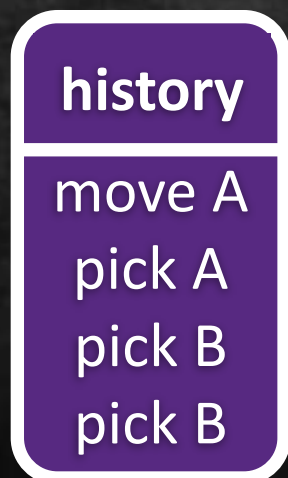
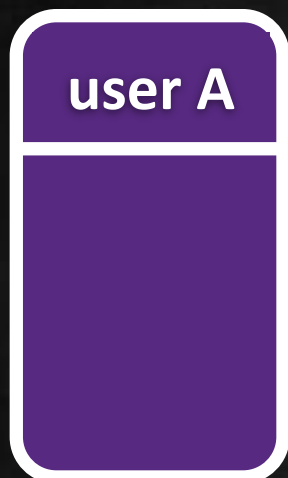
user A

location
count 3

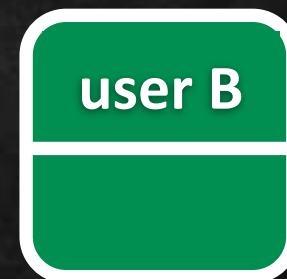
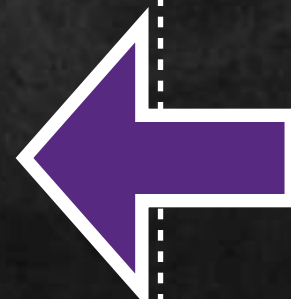
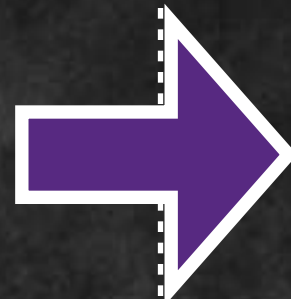
user B

Server

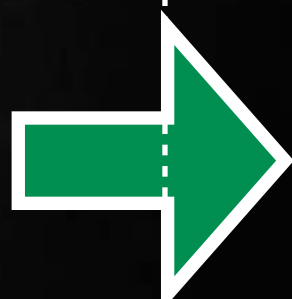
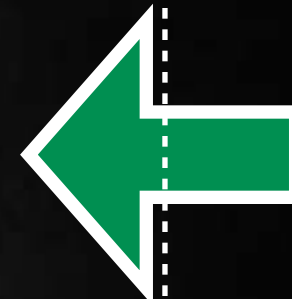


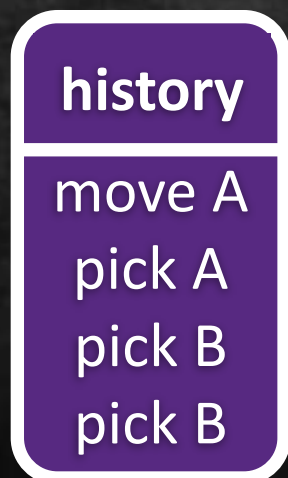


Client A

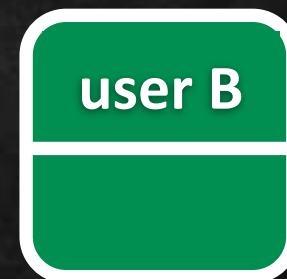
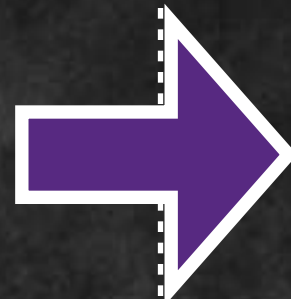


Server

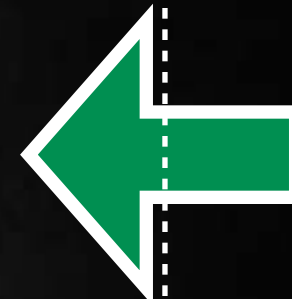




Client A



Server





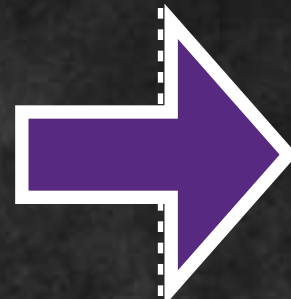
state
count 1
max 3

user A
move
pick

user B

history
move A
pick A
pick B
pick B

Client A

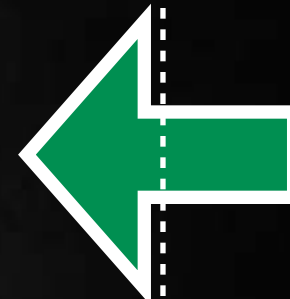


user A

location
count 3

user B

Server





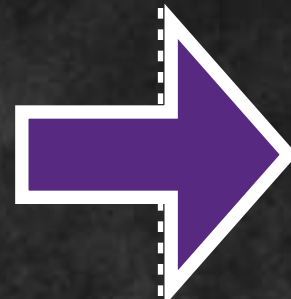
state
count 2
max 3

user A
move
pick

user B
pick

history
move A
pick A
pick B
pick B

Client A

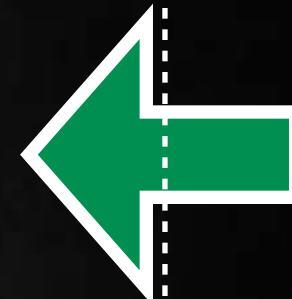


user A

location
count 3

user B

Server





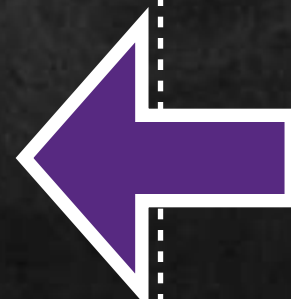
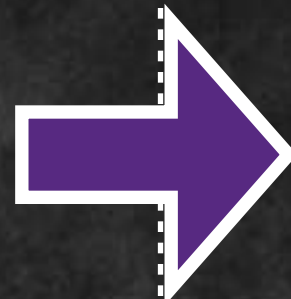
state
count 3
max 3

user A
move
pick

user B
pick
pick

history
move A
pick A
pick B
pick B

Client A

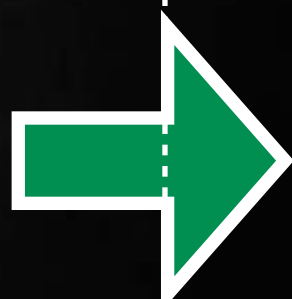
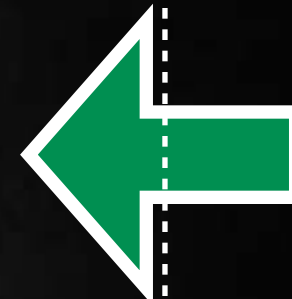


user A

location
count 3

user B

Server





...

(Let's look
inside)

Thoughts
Client
Server
Operation

Erlang

Erlang

Process isolation

Erlang

Process isolation

Networking

Erlang

Process isolation

Networking

Distribution

User process

User process

Stores user state

User process

Stores user state

Real-time push to client

World process

World process

Serializes updates

World process

Serializes updates

Concurrency “control”

World process

Serializes updates

Concurrency “control”

Runs game logic

Game logic

Game logic

Pure

Game logic

Pure

$f(\text{State}, \text{Cmd}) \rightarrow \{\text{NewState}, \text{Reply}\}$

Game logic

Pure

$f(\text{State}, \text{Cmd}) \rightarrow \{\text{NewState}, \text{Reply}\}$

Transactions

Real-time push

Real-time push

Browser support

Real-time push

Browser support

“Transfer-Encoding: chunked”

Real-time push

Browser support

“Transfer-Encoding: chunked”

State updates as JSON

Thoughts
Client
Server
Operation

Elli

Elli

Efficient, robust

Elli

Efficient, robust

High-throughput, low latency

Elli

Efficient, robust

High-throughput, low latency

Request-response model

Elli

Efficient, robust

High-throughput, low latency

Request-response model

github.com/knutin/elli

Distribution

Distribution

Idea: Identical nodes, Riak

Distribution

Idea: Identical nodes, Riak

No centralization/serialization

Distribution

Idea: Identical nodes, Riak

No centralization/serialization

No SPOF

Distribution needs

Distribution needs

Locking

Distribution needs

Locking

Dynamic cluster

Meet “locker”

Meet “locker”

Distributed lock table

Meet “locker”

Distributed lock table

CAS

Meet “locker”

Distributed lock table

CAS

Write quorum, 2PC

Meet “locker”

Distributed lock table

CAS

Write quorum, 2PC

Asynchronous replication

Meet “locker”

Distributed lock table

CAS

Write quorum, 2PC

Asynchronous replication

330 SLOC, one gen_server

The Games
The Past
The Present
The Future
The Foundation

Research

Research

Aka “hacking”

Everything is possible

Everything is possible

First person

Everything is possible

First person

Real-time strategy

Everything is possible

First person

Real-time strategy

Turn-based

Everything is possible

First person

Real-time strategy

Turn-based

Etc

Nothing is possible

Nothing is possible

Network latency kills
gameplay

Lag compensation

Lag compensation

Many solutions

Lag compensation

Many solutions

Depends on game

FPS

FPS

Speed is king

FPS

Speed is king

Divergent simulations

FPS

Speed is king

Divergent simulations

Server keeps history

FPS

Speed is king

Divergent simulations

Server keeps history

Validates delayed commands

RTS

RTS

Synchronization is king

RTS

Synchronization is king

Identical simulations

RTS

Synchronization is king

Identical simulations

Server buffers commands

RTS

Synchronization is king

Identical simulations

Server buffers commands

Broadcasts, lockstep

The Games
The Past
The Present
The Future
The Foundation

Small independent teams

Each team adds something

Based on previous learnings

Standing on the
shoulders of giants

Slides: <http://woo.ga/backend>

Slides: <http://woo.ga/backend>

Code: <http://github.com/wooga>

Questions?

Questions?

@jriri @knutin

<http://wooga.com/jobs>