Maybe sticky sessions weren't a bad idea


©IRES MLS


©IRES MLS

Trends

Tradeoffs

Nuance

Status Quo

1. Write my PHP / Java / Rails
2. Connect it to an RDBMS

Buy lots of web servers & really big databases

Storing state **in** an application

- Just what the heck is state anyway?
- How do we store it today
- Some history
- Ok, but why?
- Caveats
- More information

State of the state

- Data vs. behavior

- A coupling with time and space

Time

- "happens before"
- memory barriers
- synchronization primitives

Space

- threads
- processes
- servers

State is a lens

```
def fact(i: Int) = { return i * fact(i - 1) }
```

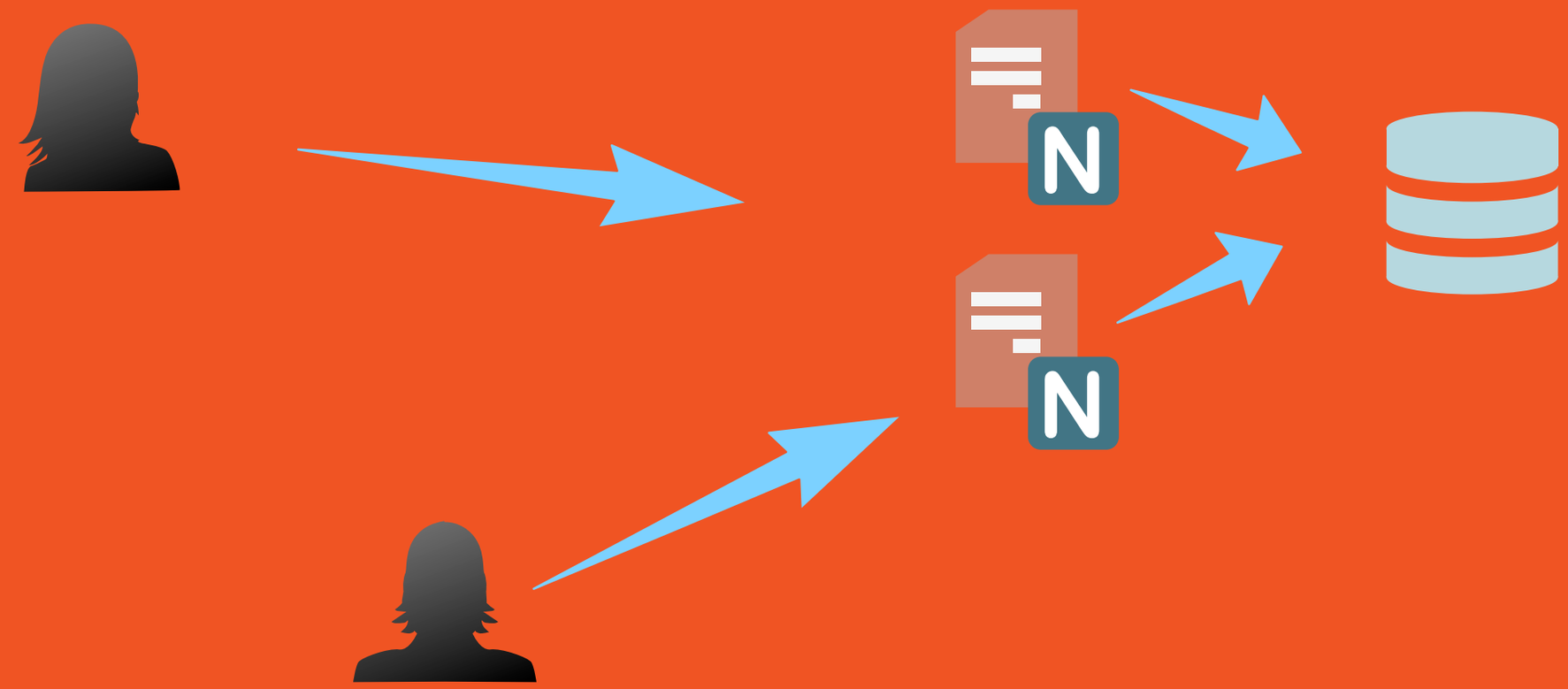
State is a lens

```
def doIt(i: Int) = { log("you sent $i") }
```

State-**less** vs. State-**ful**

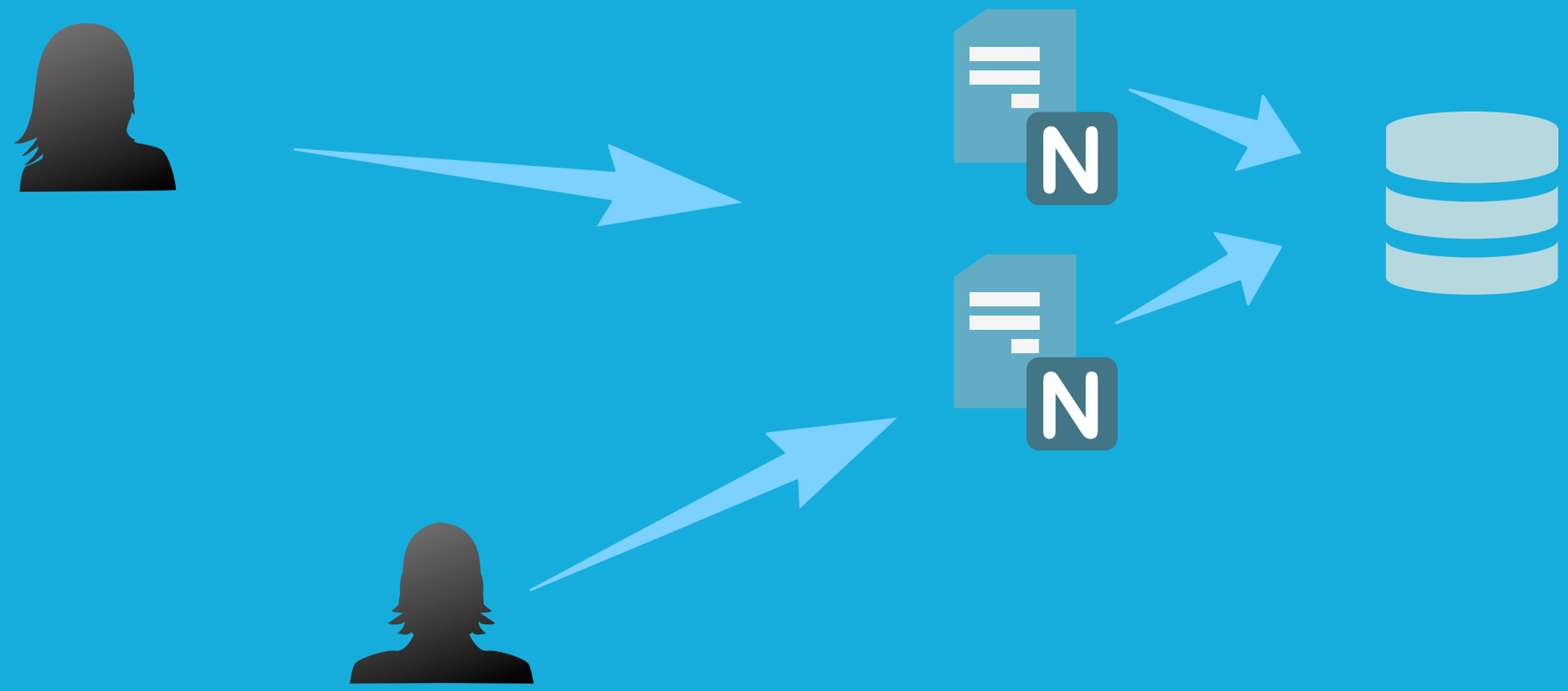
- Store data in a database

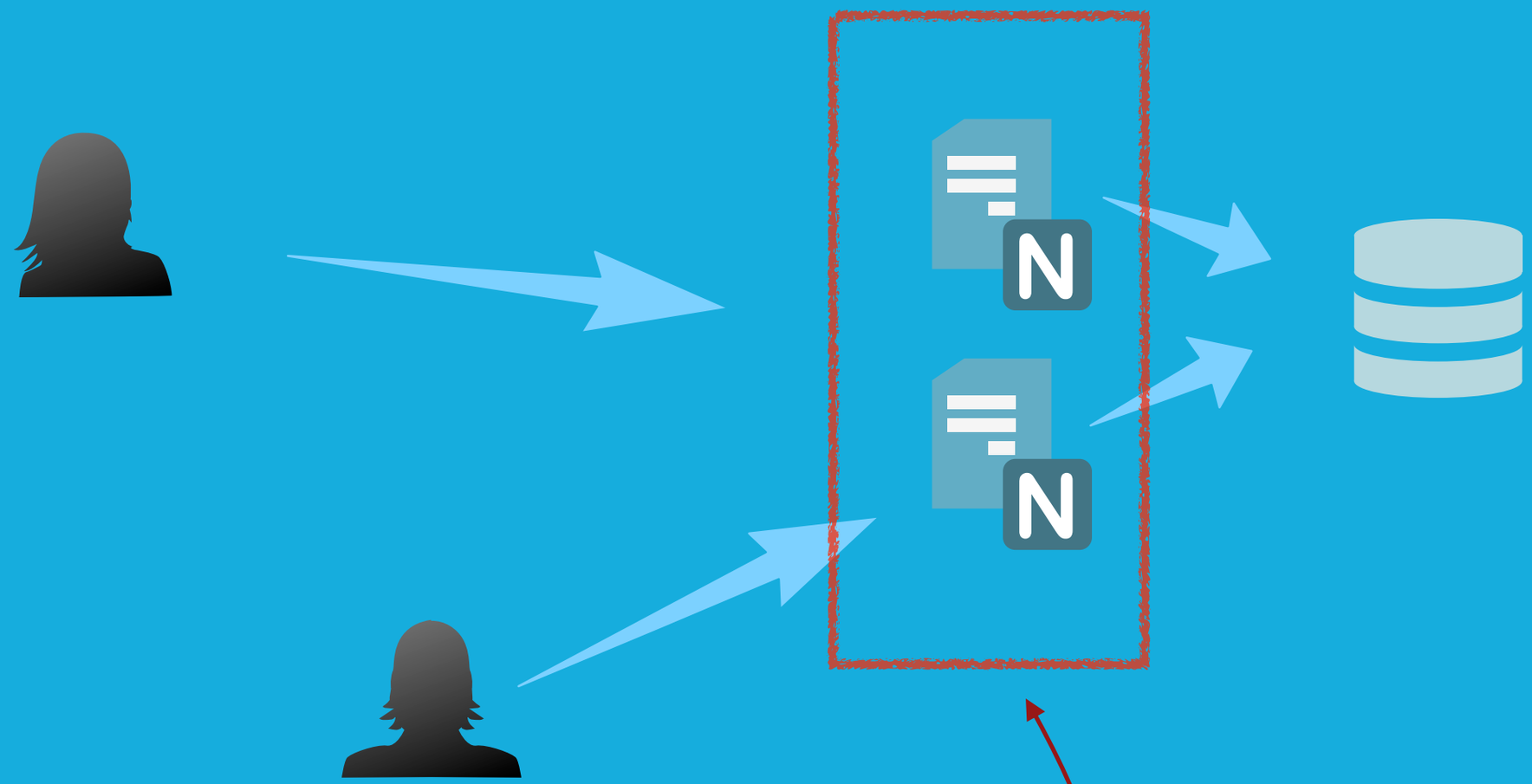
- Ship data => behavior


N
N

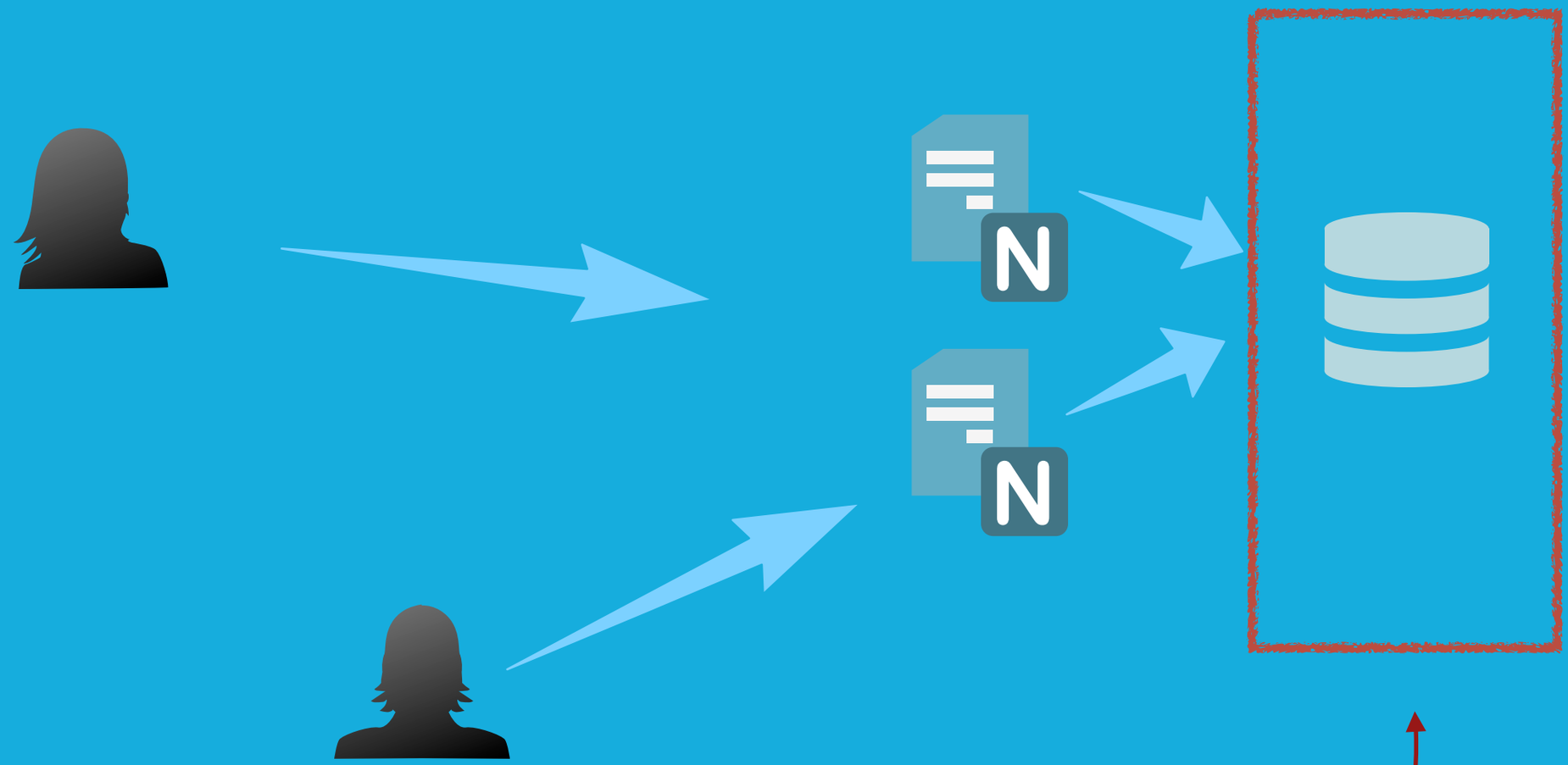
State-**less** applications

- Deployed behind a load balancer
- Most common CRUD applications

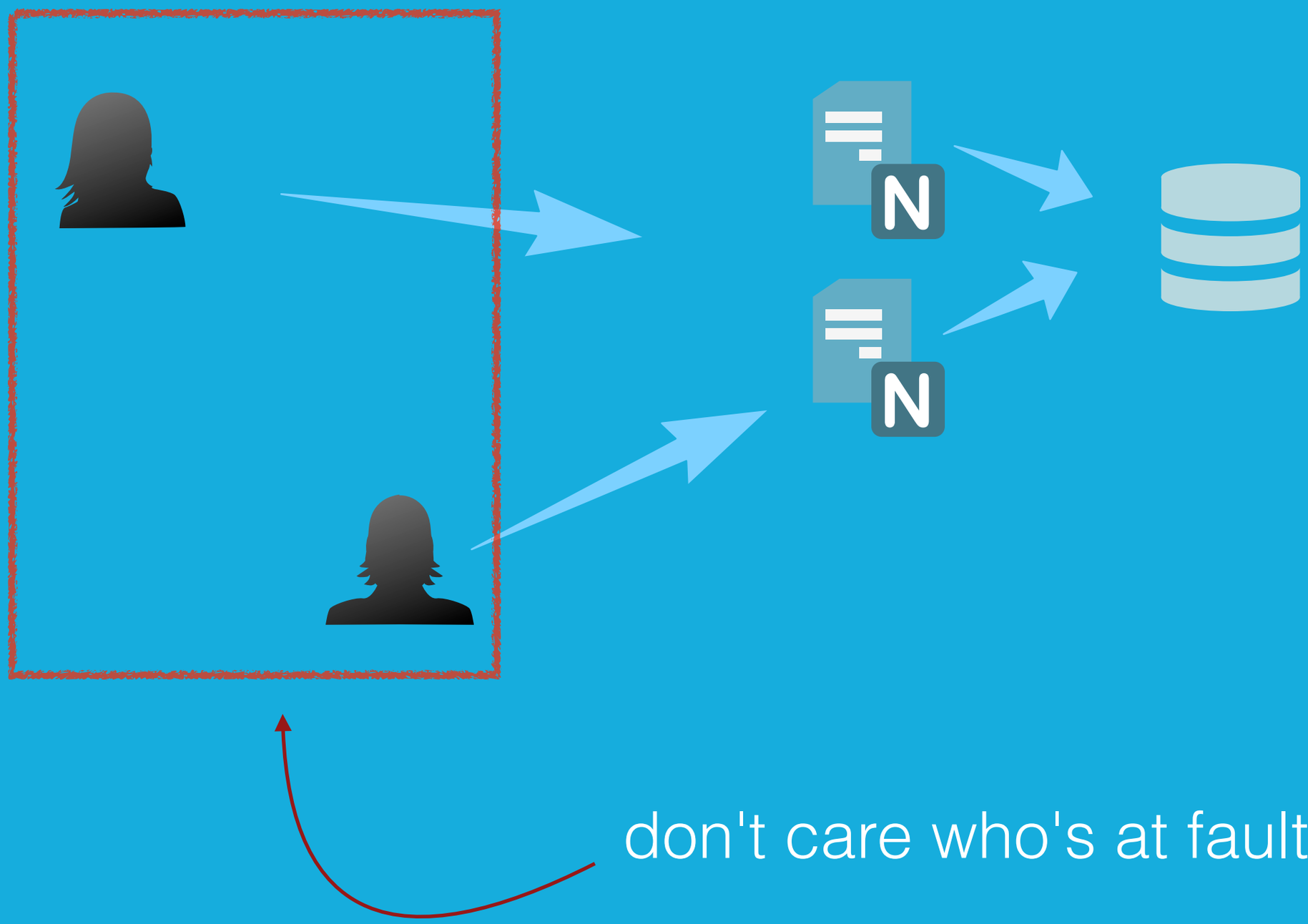




my engineers wrote this 😱

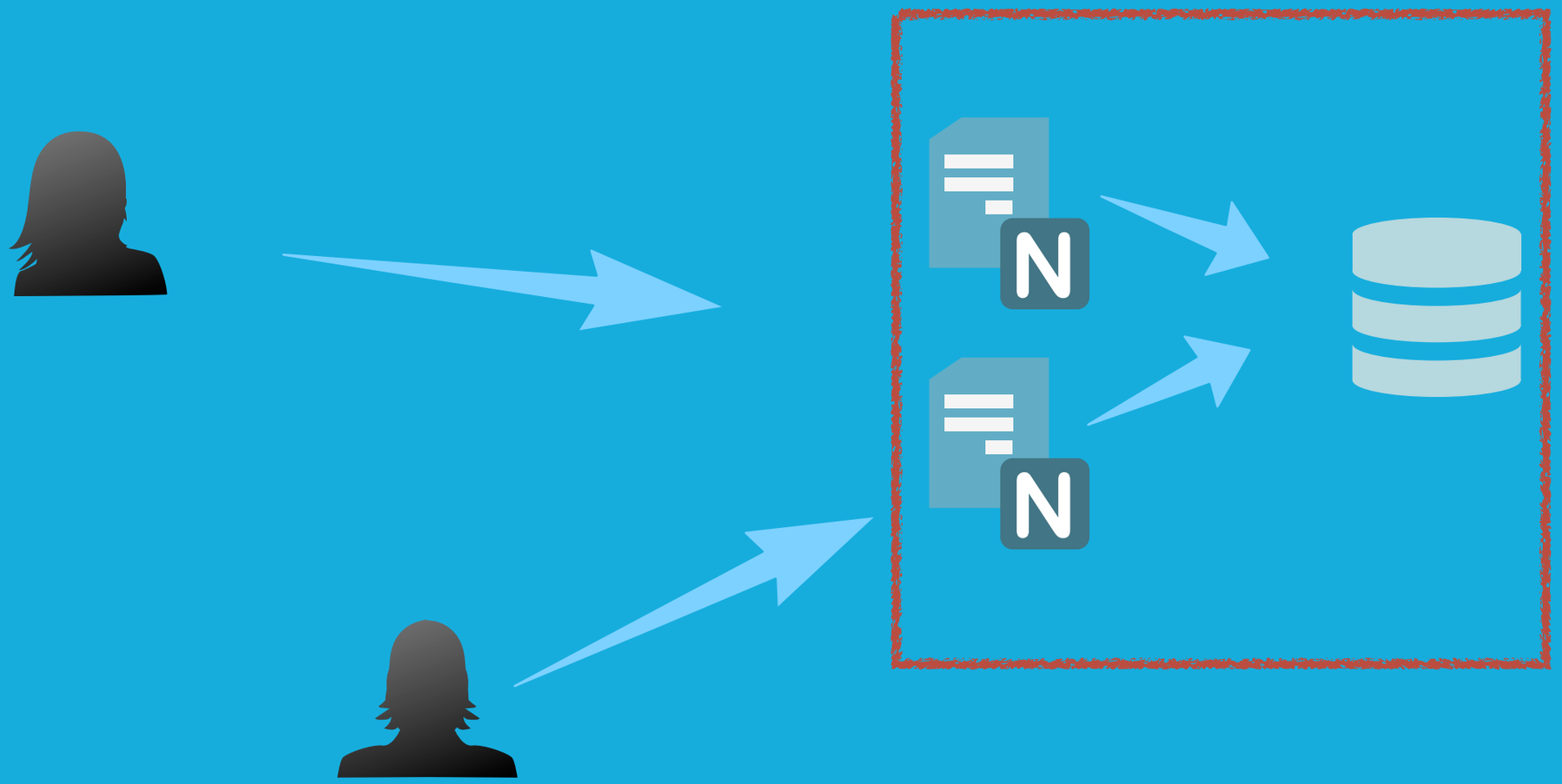


This thing "works"



don't care who's at fault

State-ful



State-**ful** applications

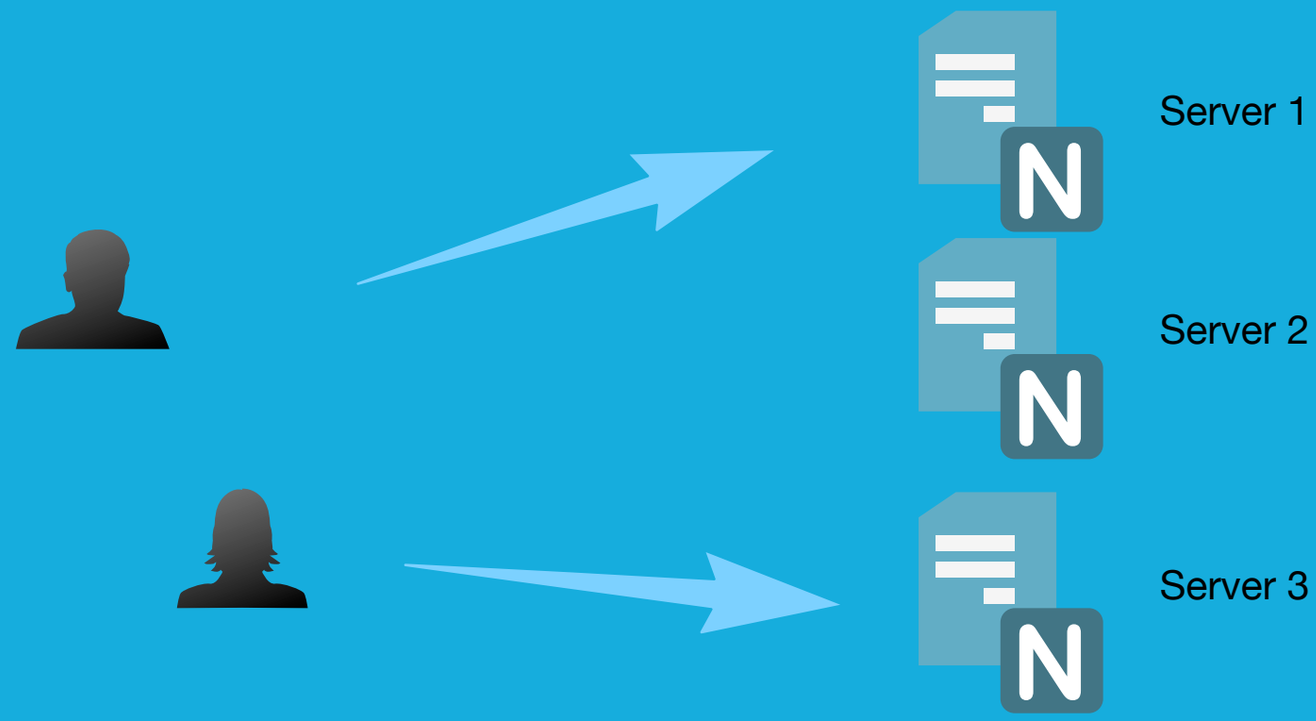
- Store data with the behavior
- Data does not move when worked on

Sticky sessions

- Sticky sessions are an http concept that glues a *session* to a server

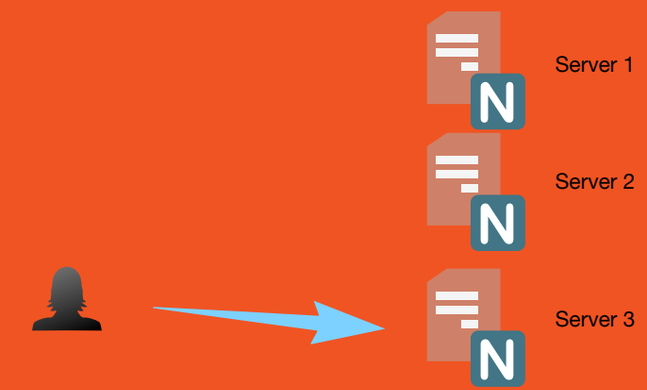
- Keeps sessions in a single place

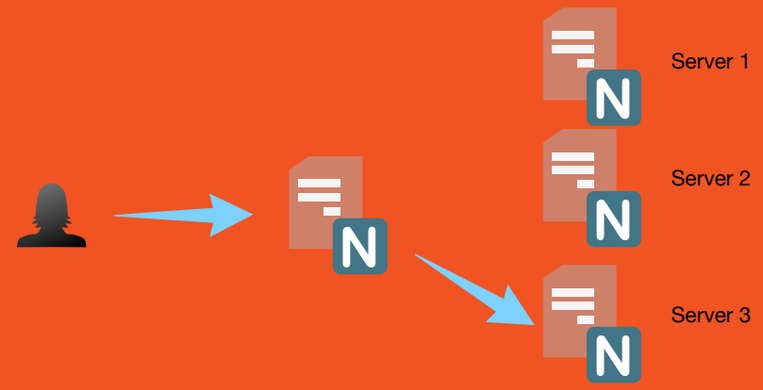
Sticky session



How

- DHT
- Non-**D** HT





Major motivators

- Performance
- Correctness
- Programmer ergonomics
- Resilience

Performance

CPU cache	1ns	1s
Main memory	120ns	2 min
Disk	50-150µs	14 hours
Network	500 µs	6 days

Performance

CPU cache	1ns	1s
Main memory	120ns	2 min
Disk	50-150µs	14 hours
Network	500 µs	6 days

Performance

CPU cache	1ns	1s
Main memory	120ns	2 min
Disk	50-150µs	14 hours
Network	500 µs	6 days

Correctness

- Data & behavior co-existing means we can reason about safely changing state

Programmer Ergonomics

Resilience

- Classes of error around txns
- Connection pools
- Failure can be handled

How do we do this...

Runtimes

- Long lived processes
- Threading model
- Control over memory

Frameworks

- Supports some way to make remote calls
- Treats concurrency as a first class citizen
- A concept of clustering

Some examples

The downside

Serialization

- De-serialize the future
- De-serialize the past

Thundering herds

- Startup time (deployment)
- Rebalance performance

But it worked on my computer?!?

Delicious memory

- Unbounded, in-memory data structures

But it worked on my computer?!?

~~Copying~~ Inspiration

- Any distributed database: Riak, Cassandra, Dynamo
- Akka distributed data / cluster sharding
- Orleans
- Unison
- CRDTs

- Just what the heck is state anyway?
- How do we store it today
- Some history
- Ok, but why?
- Caveats
- More information

@boulderdanh

@boulderdanh