



THE GAME
INSIDE
THE GAME.

Lessons Learned: WebSockets

Martin Chaov - Architecture Senior Manager @ Draftkings Inc.



DISCLAIMER

Agenda

Quick overview of WebSockets

Lifecycle

Use-cases

Benefits

Challenges

Alternatives

Summary



What is a WebSockets?



References > The WebSocket API (WebSockets) ⋮

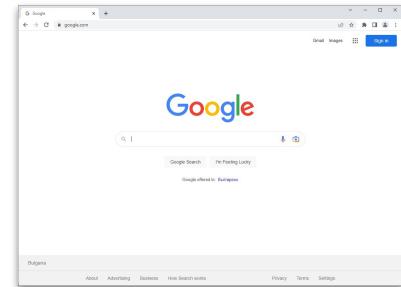
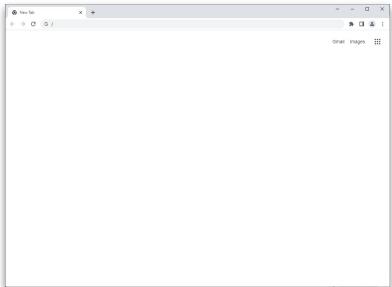
The WebSocket API (WebSockets)

The **WebSocket API** is an advanced technology that makes it possible to open a two-way interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses without having to poll the server for a reply.

Can I use?



Typical HTTP connection



DNS Lookup

Connect

Send

Wait

Load

Close

WebSocket Life Cycle

WebSocket Life Cycle

```
const ws = new WebSocket("wss://domain.ext")
```

WebSocket Life Cycle

```
const ws = new WebSocket("wss://domain.ext")  
// ws.readyState = ?
```

WebSocket LC

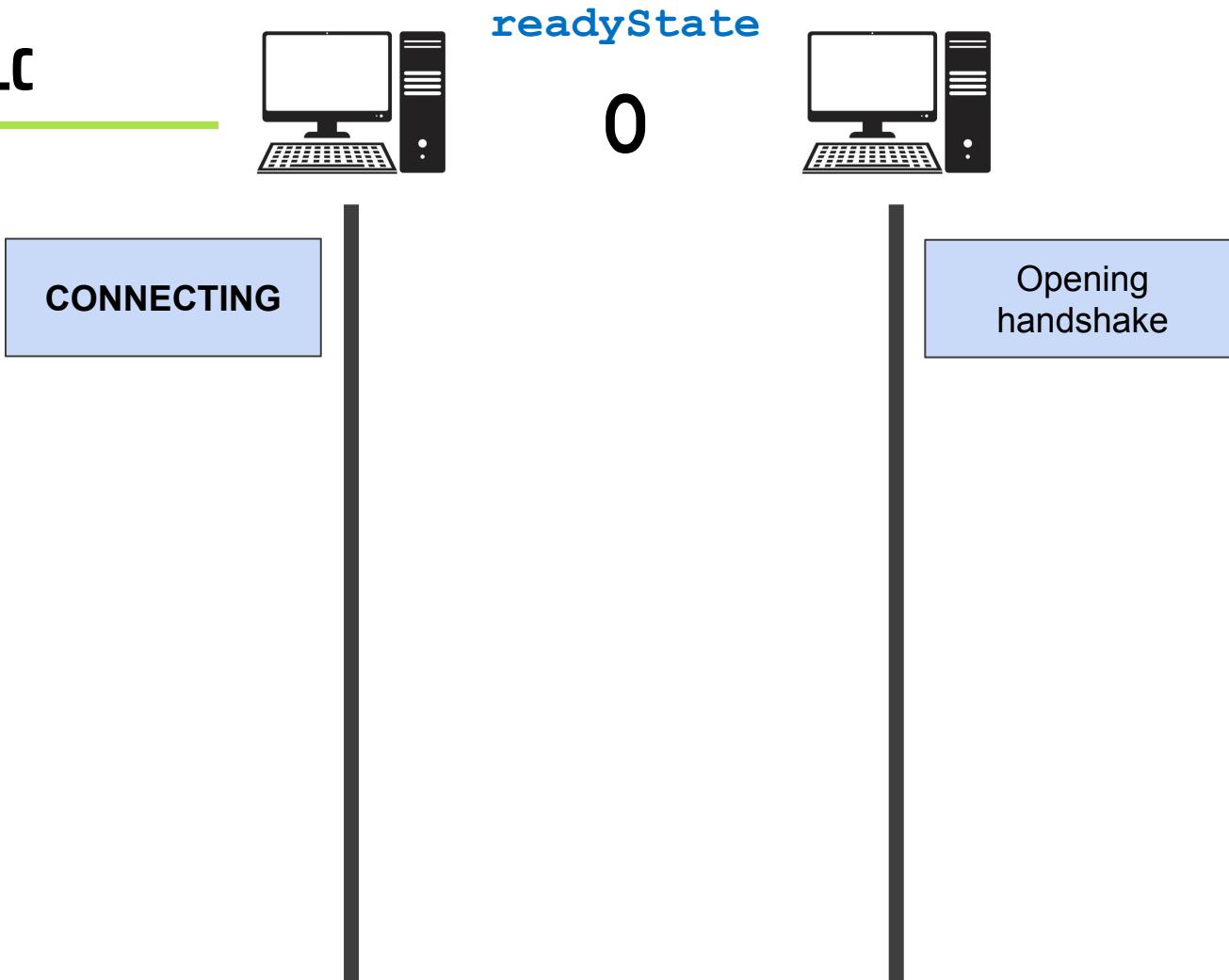


readyState

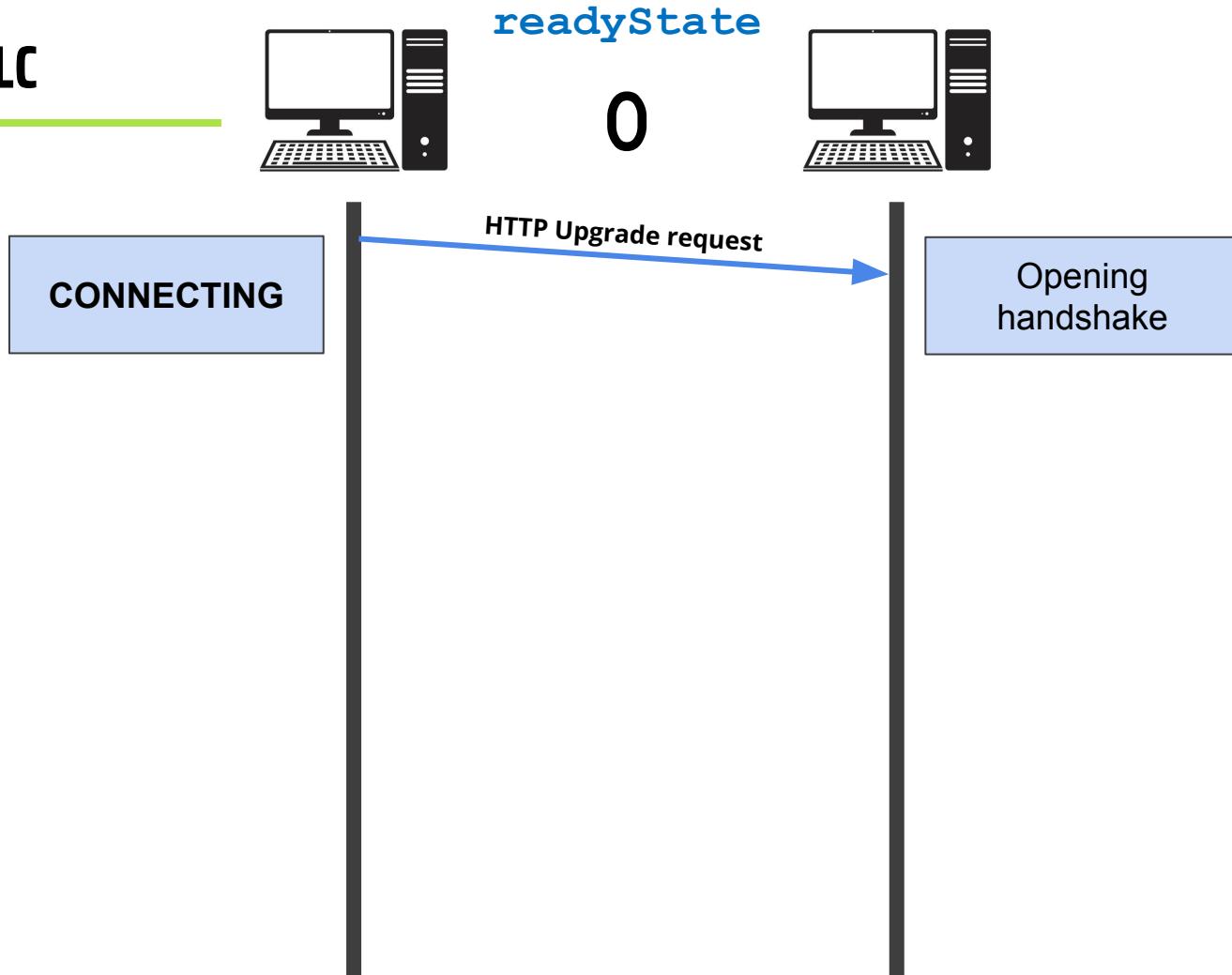
0



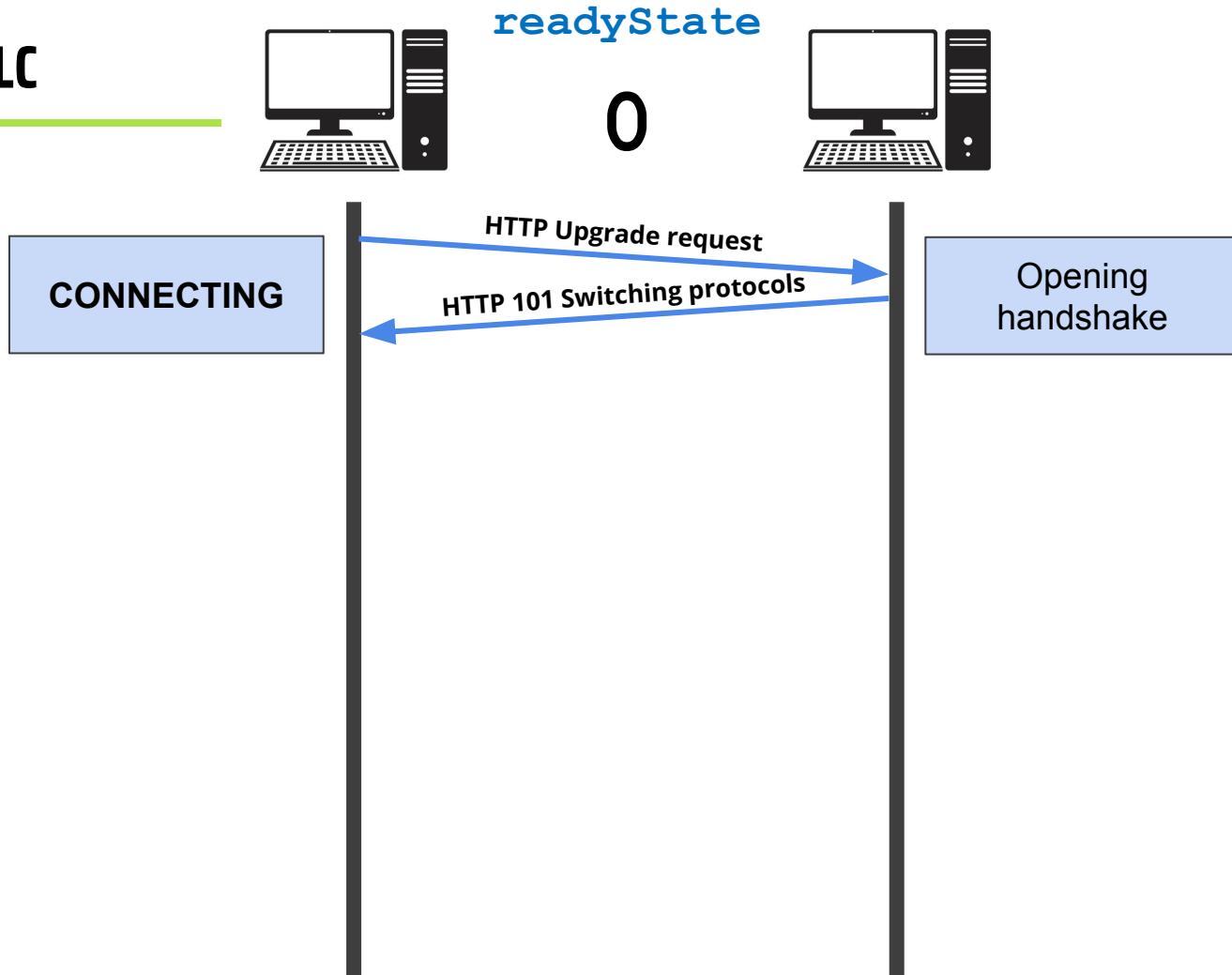
WebSocket LC



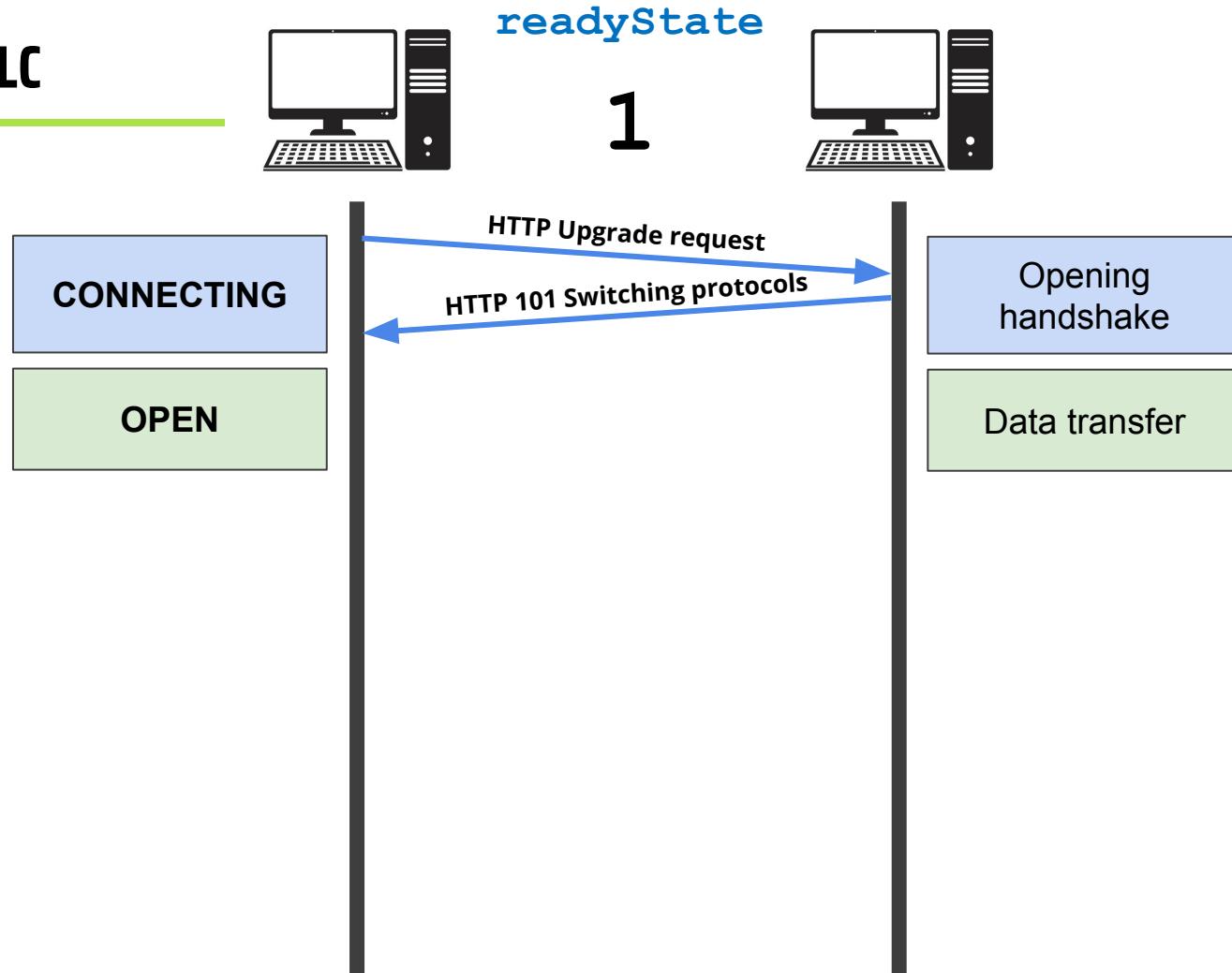
WebSocket LC



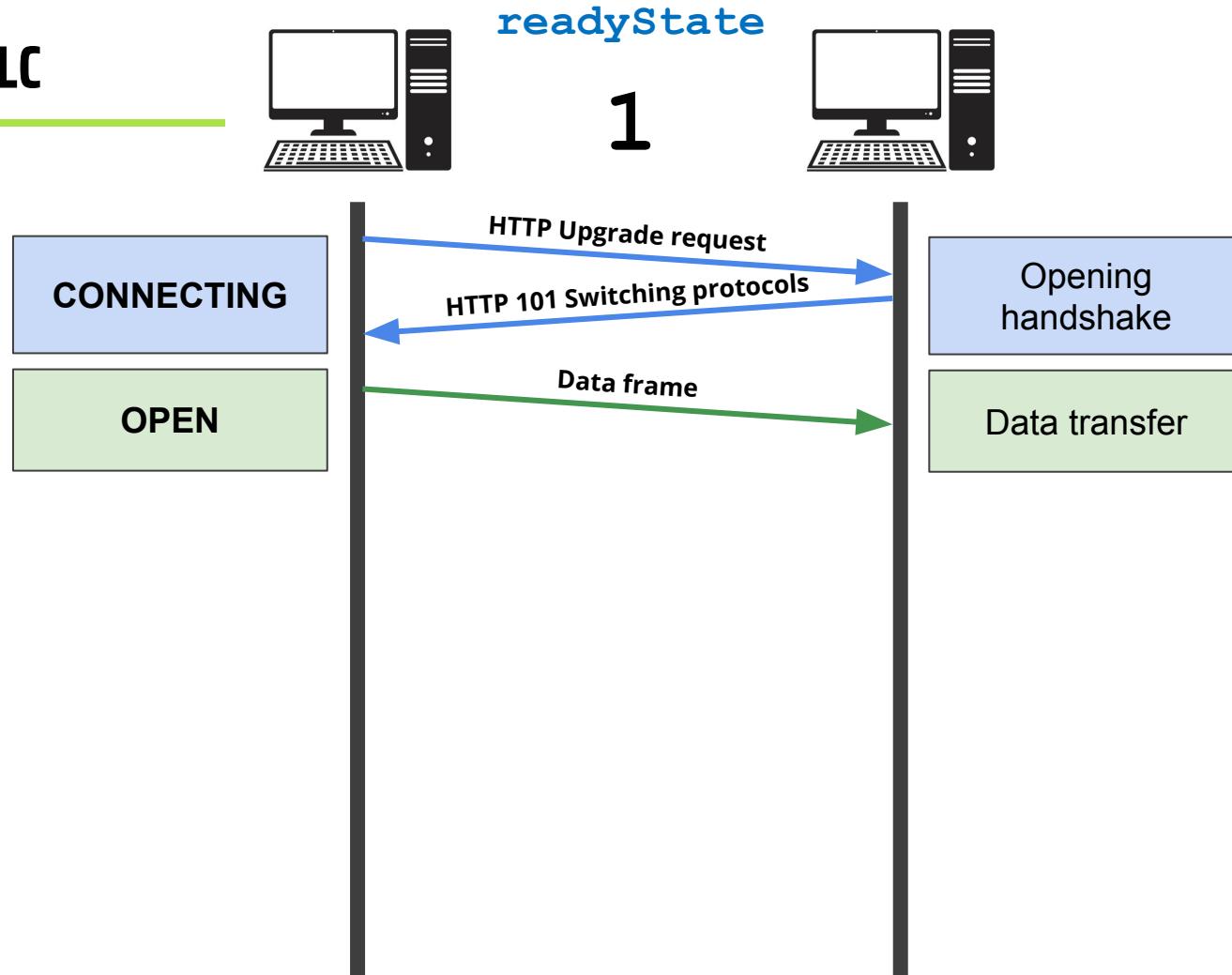
WebSocket LC



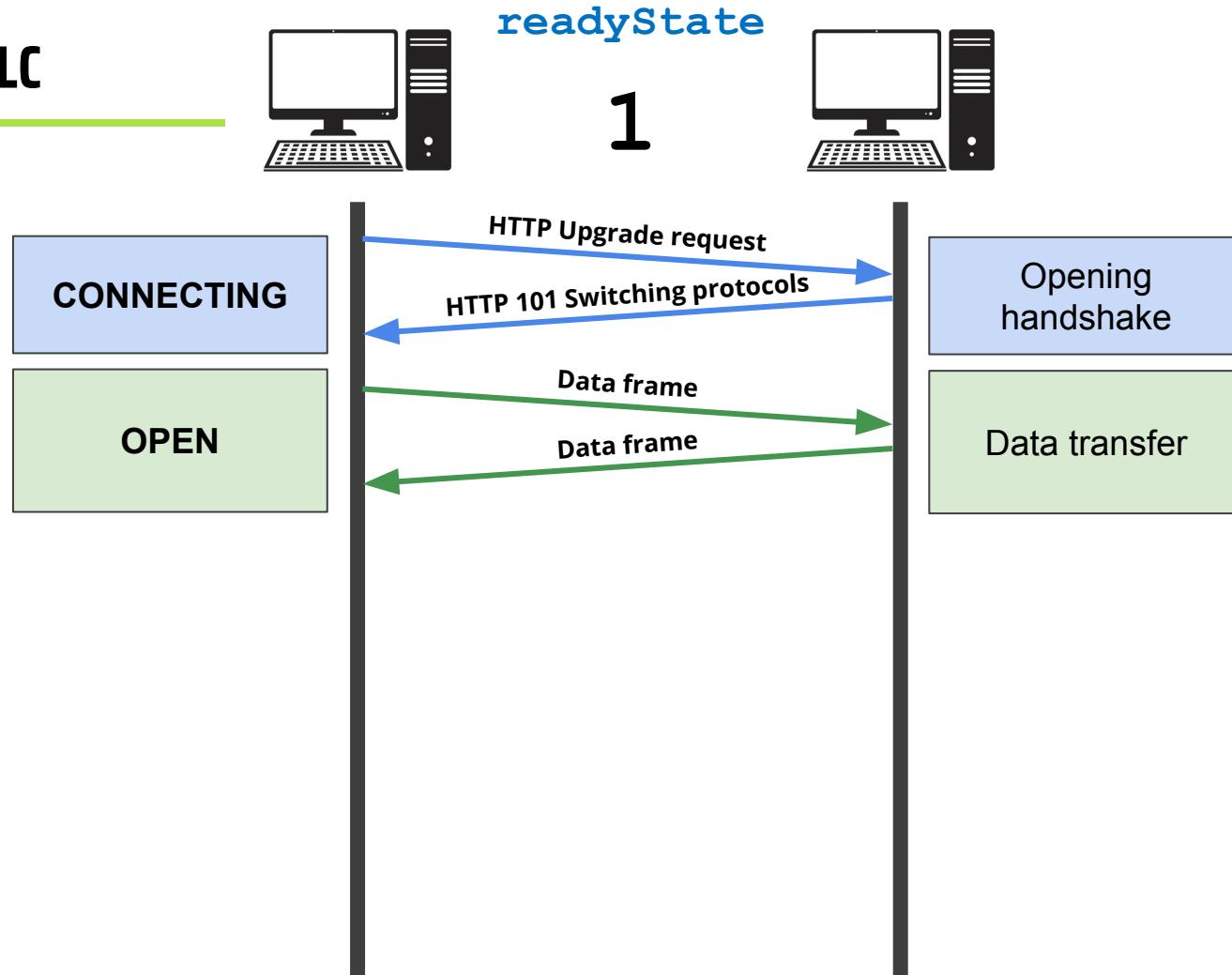
WebSocket LC



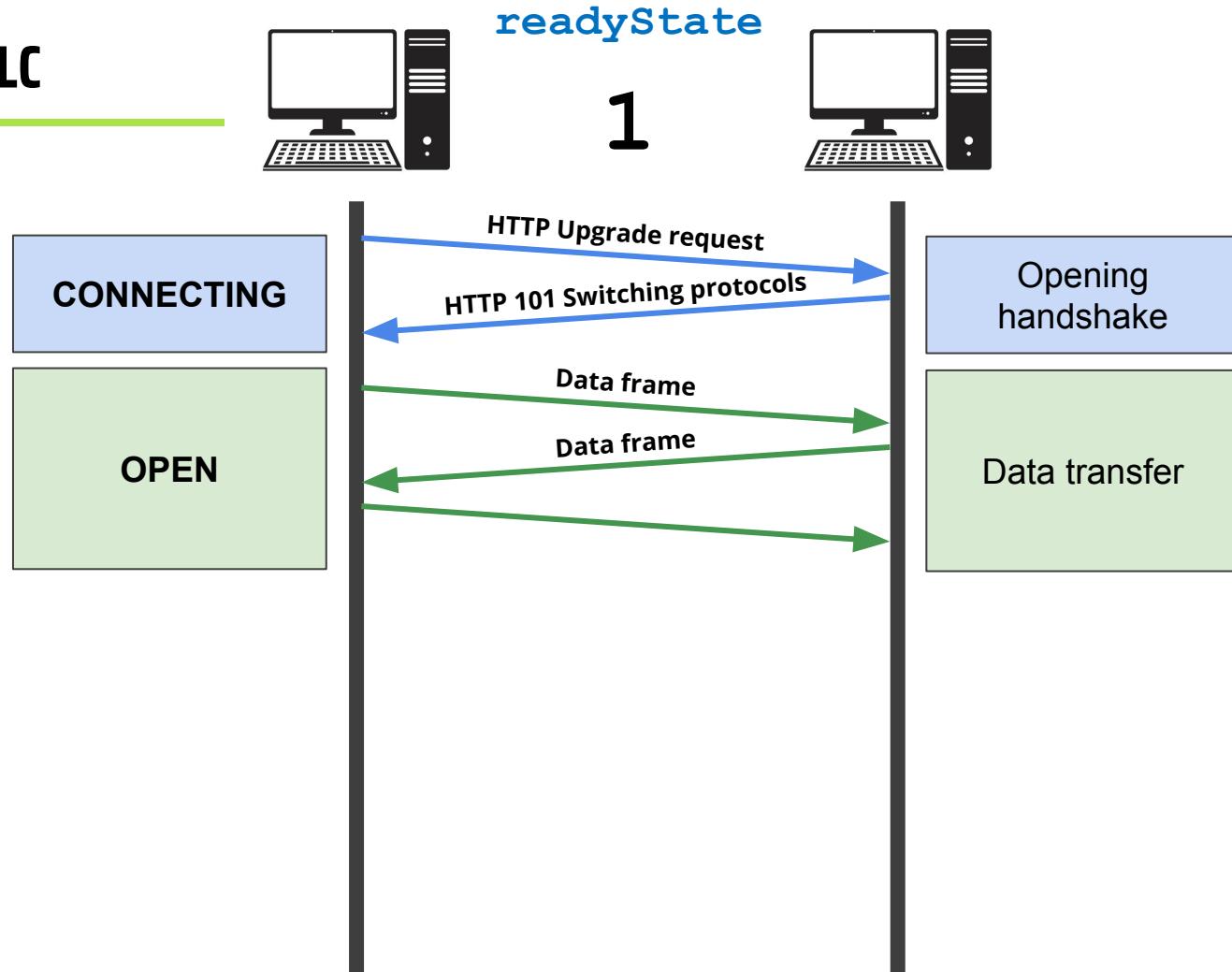
WebSocket LC



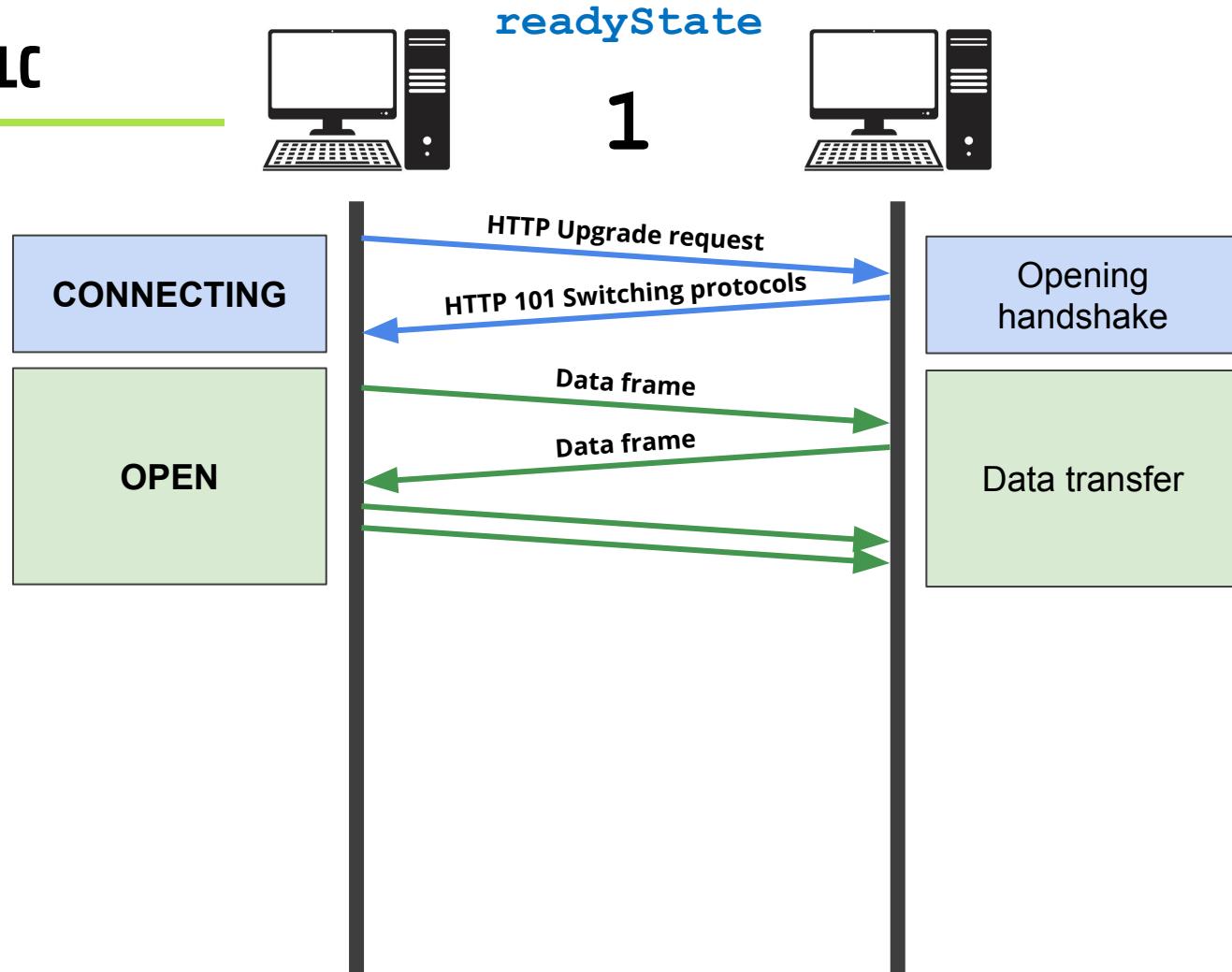
WebSocket LC



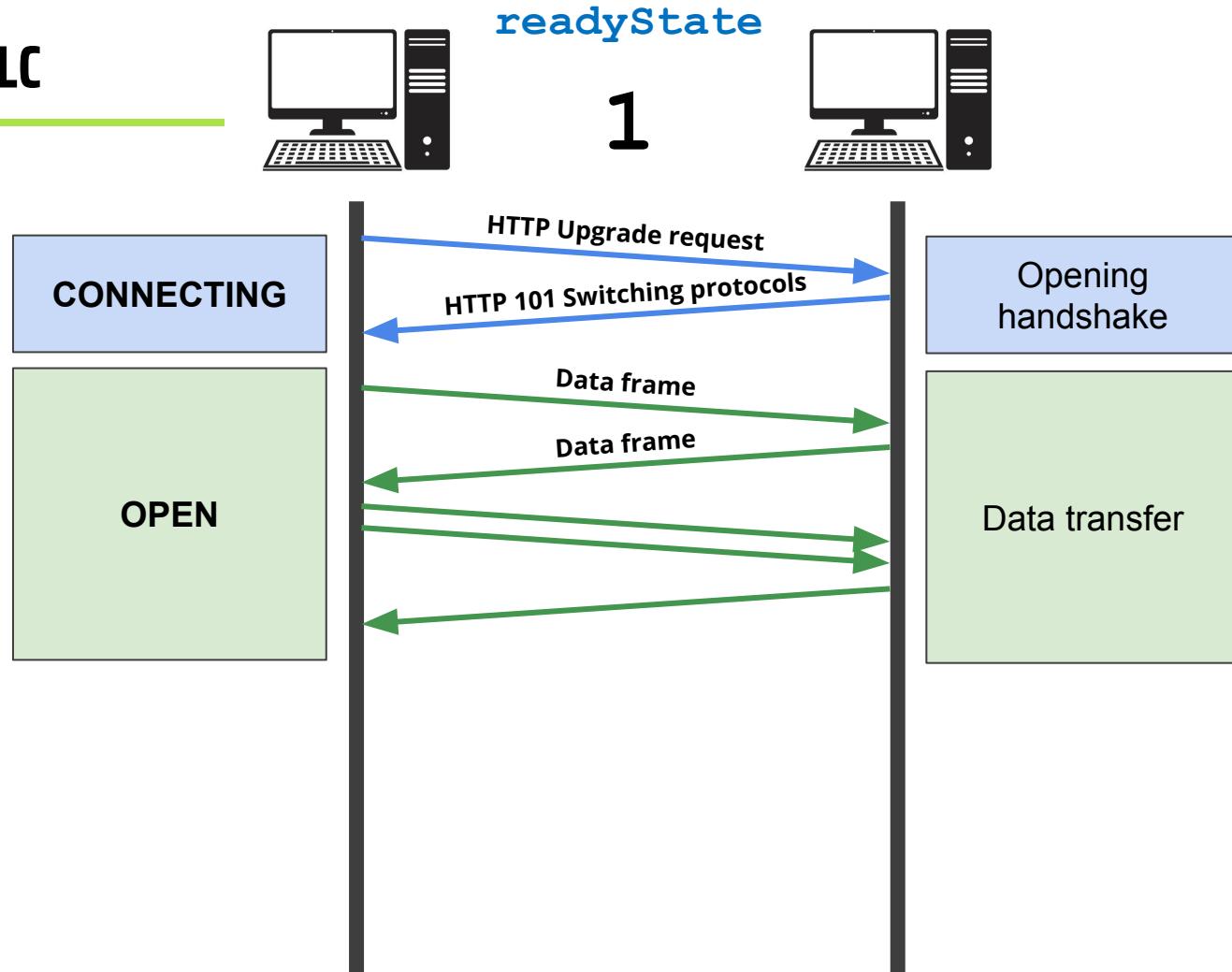
WebSocket LC



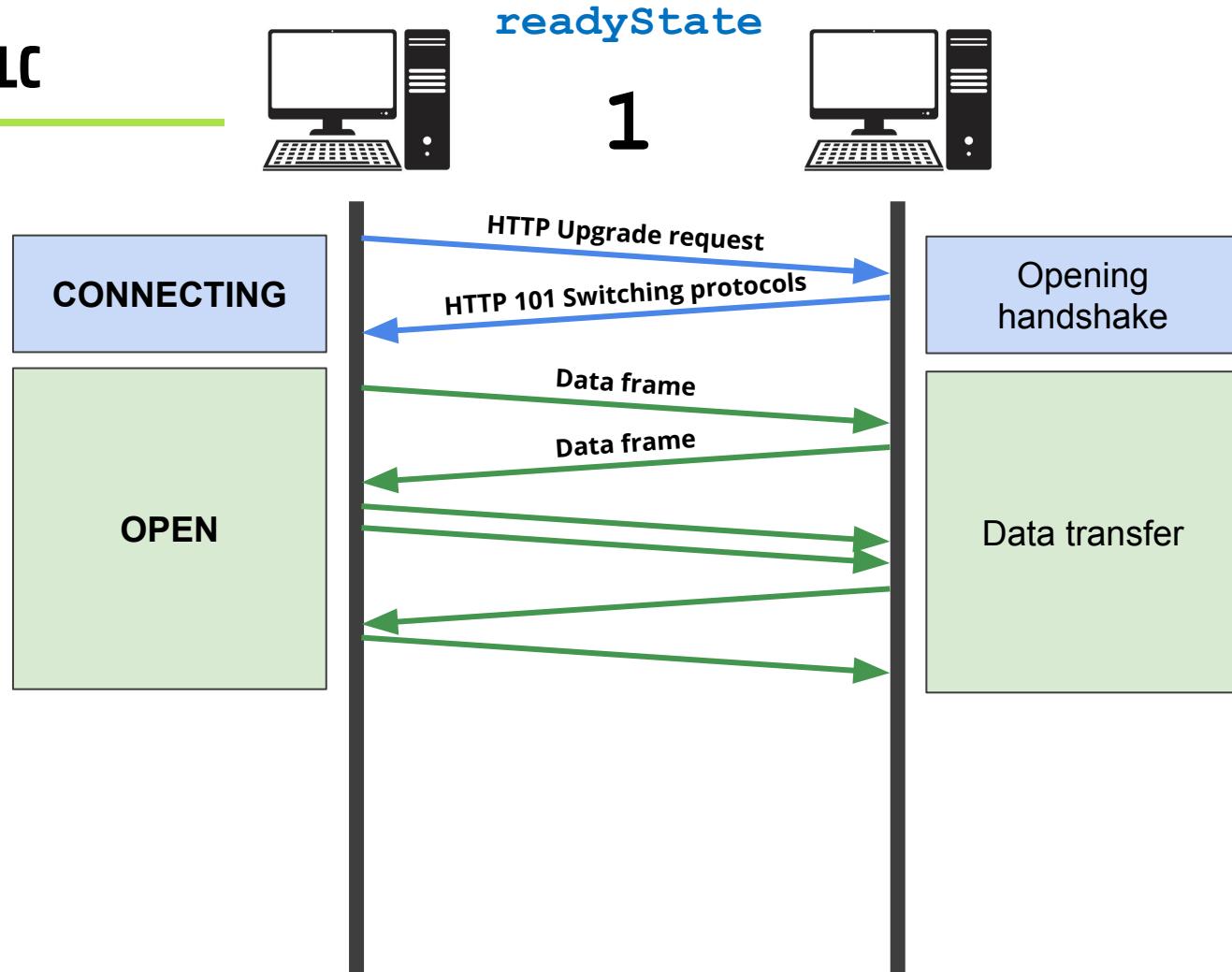
WebSocket LC



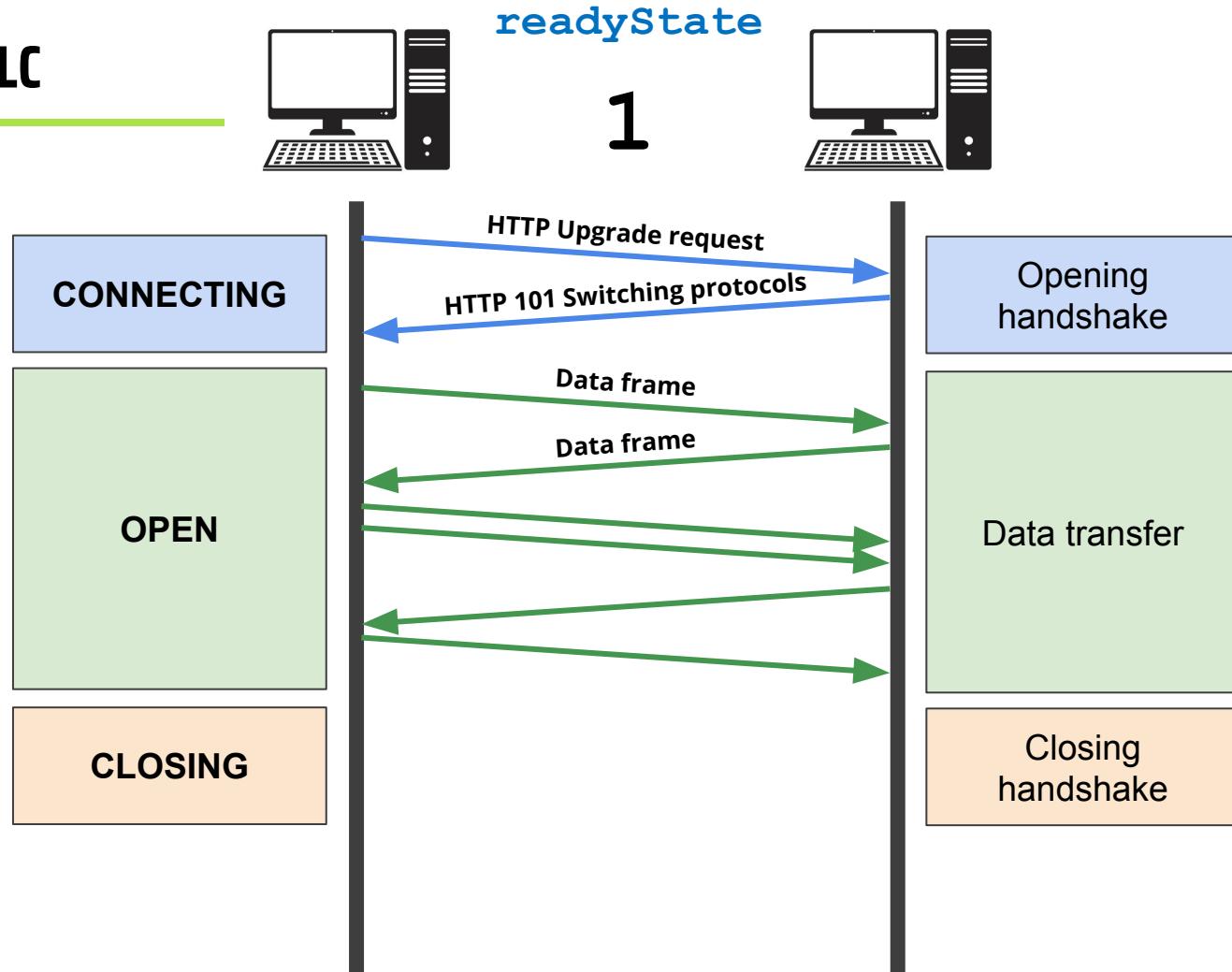
WebSocket LC



WebSocket LC



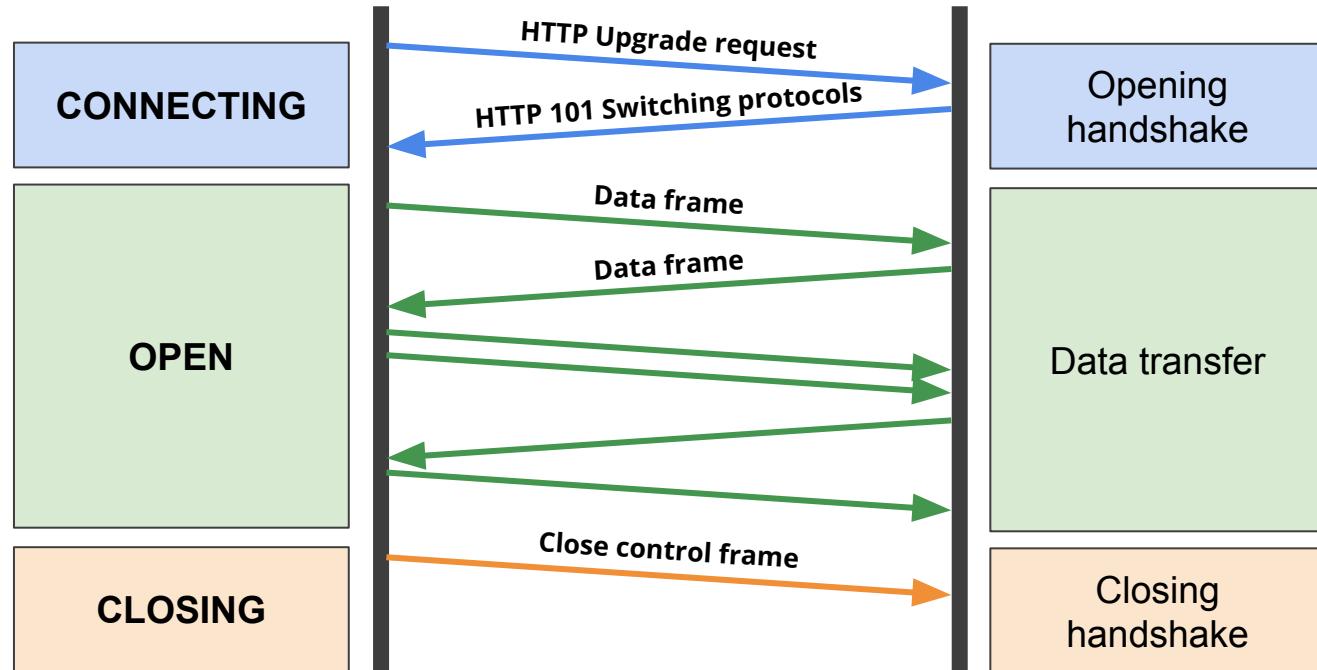
WebSocket LC



WebSocket LC

readyState

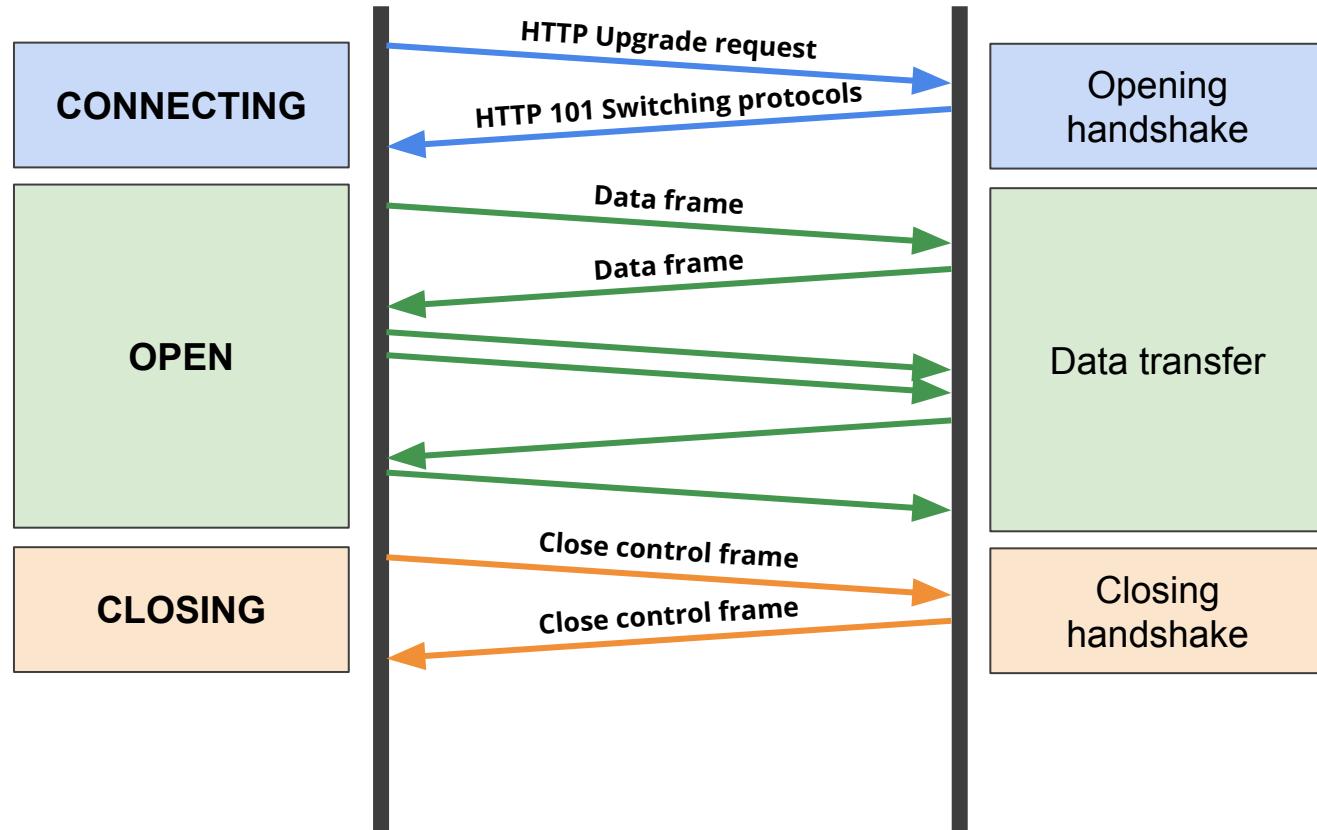
2



WebSocket LC

readyState

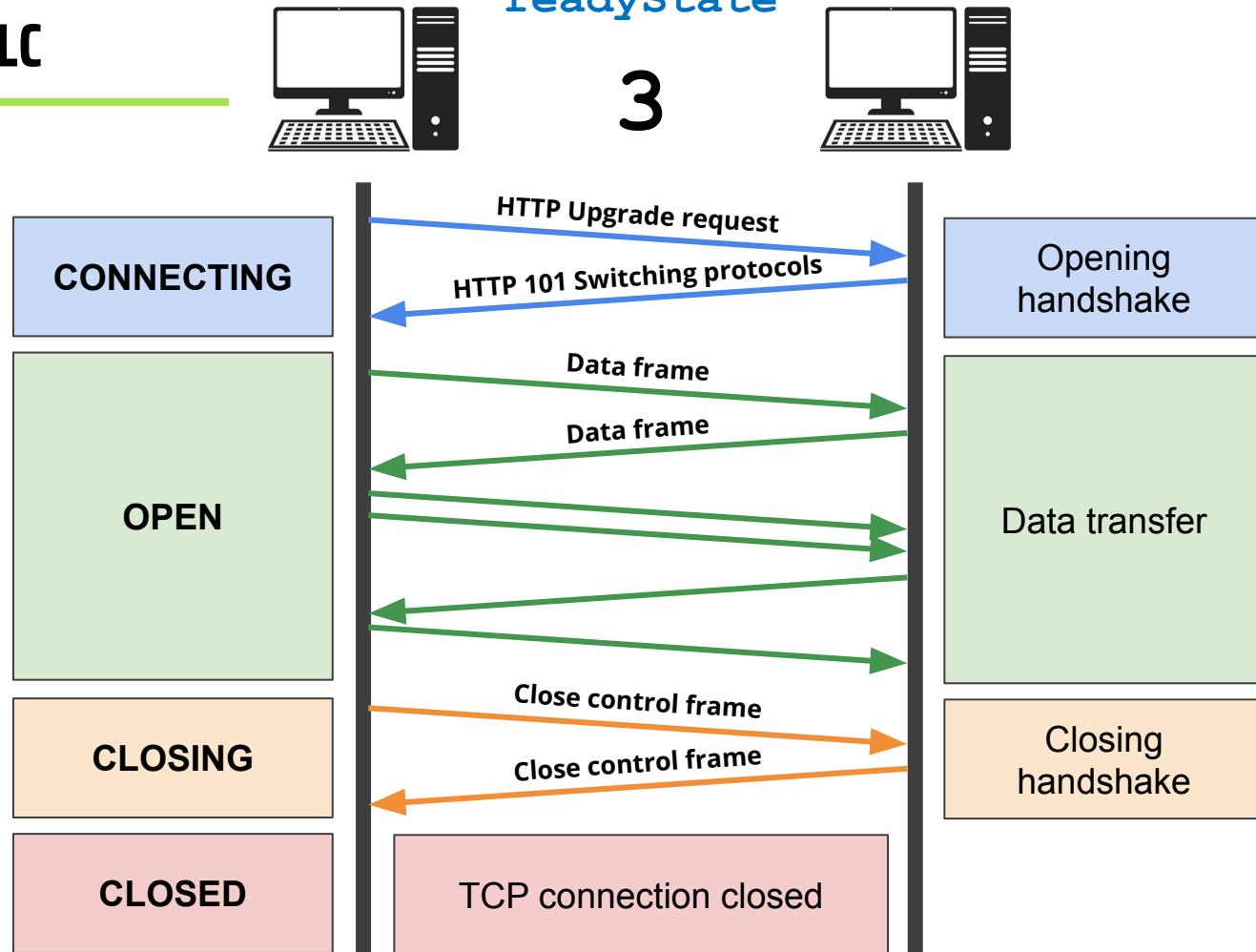
2



WebSocket LC

readyState

3





THE GAME
INSIDE
THE GAME.

Use-cases



Use-cases

- Sports feeds
- Trading and financial feeds
- Telemetry from devices
- Remote control
- Multiplayer games
- Collaborative editing of documents
- Data visualization
- Conferencing
- Dev tools

Benefits

- Wire traffic efficient
- No CORS
- Simple API
- Bi-directional
- Real-time
- Available in WebWorkers
- Widely supported in native devices, programming languages, etc.

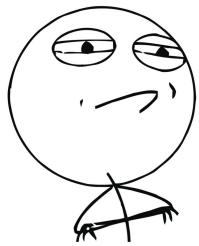
A black and white photograph of a baseball player in mid-pitch. He is wearing a Los Angeles Angels uniform, featuring the team's signature "A" logo on the back of his jersey. His right arm is extended forward, holding a baseball, while his left arm is bent at the elbow, providing balance. He is wearing a dark baseball cap and a light-colored wristband on his left wrist. The background is a soft-focus stadium or field.

Challenges

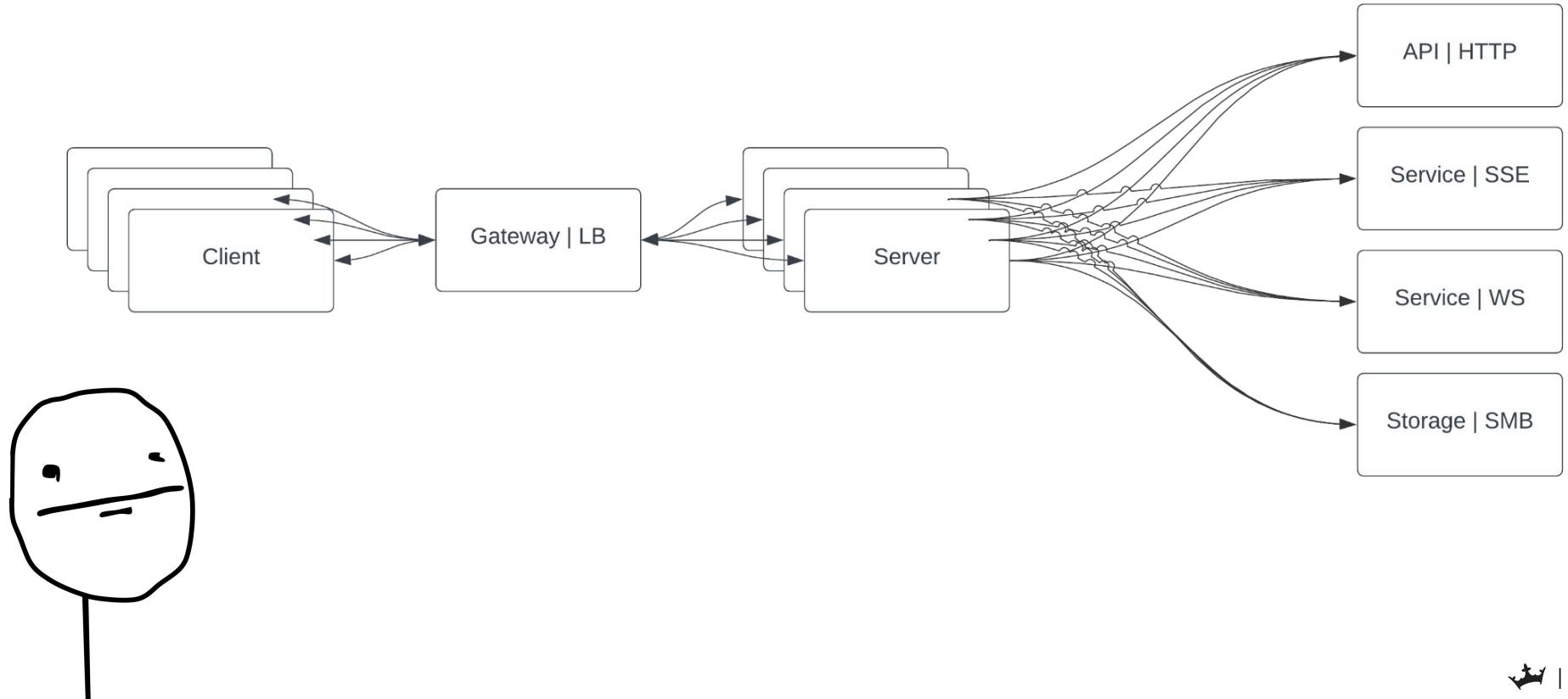
How tutorials present it?



CHALLENGE ACCEPTED

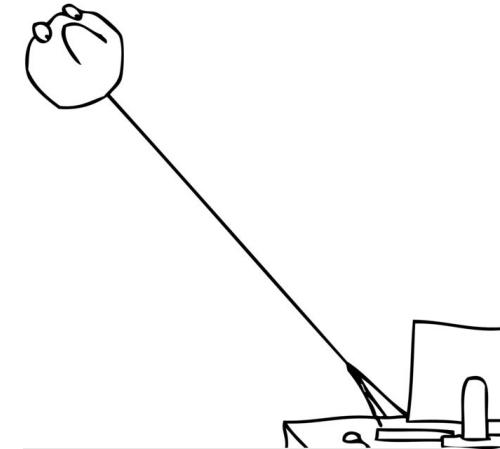


How it tends to become?



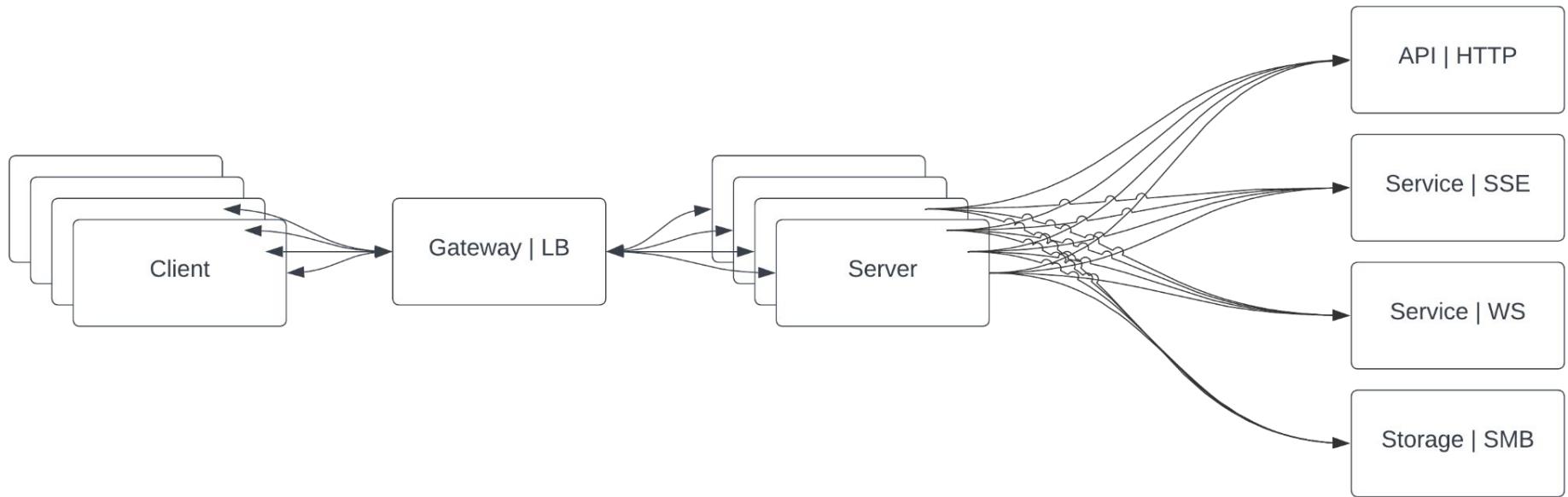
Some common challenges

- State management
 - Session stickiness
- Long lived connection
 - Load balancing
 - Disconnects
 - Heartbeat
- Application Security
 - WSS
 - Authentication
 - Authorization
 - CORS
- Backpressure
- Protocol efficiency
 - Size
 - Serialization/deserialization
- Deployment
- Network Security
 - Rate limiting
 - WAFs
 - DDoS
- APM



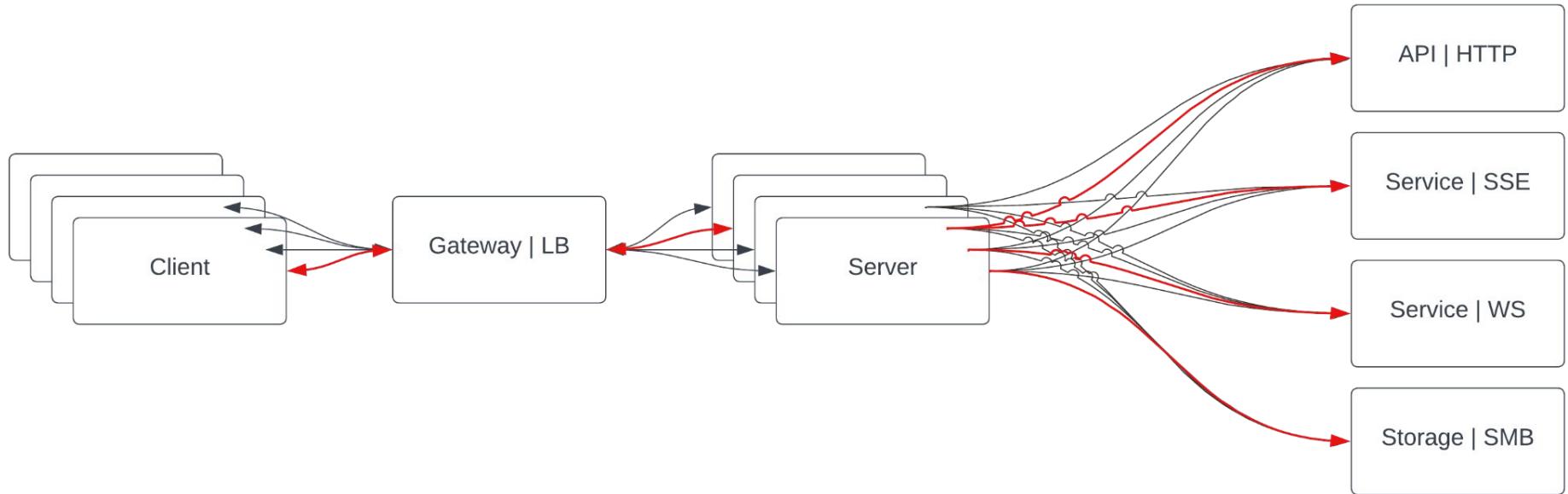
State management

State management



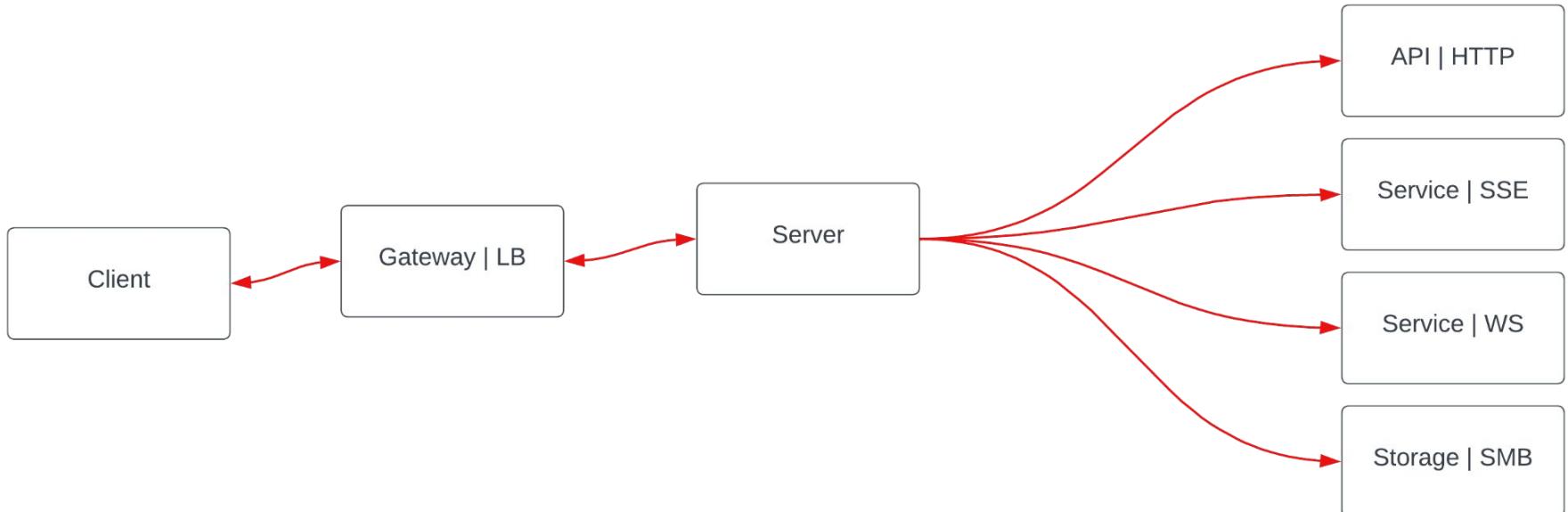
State management

- Session stickiness



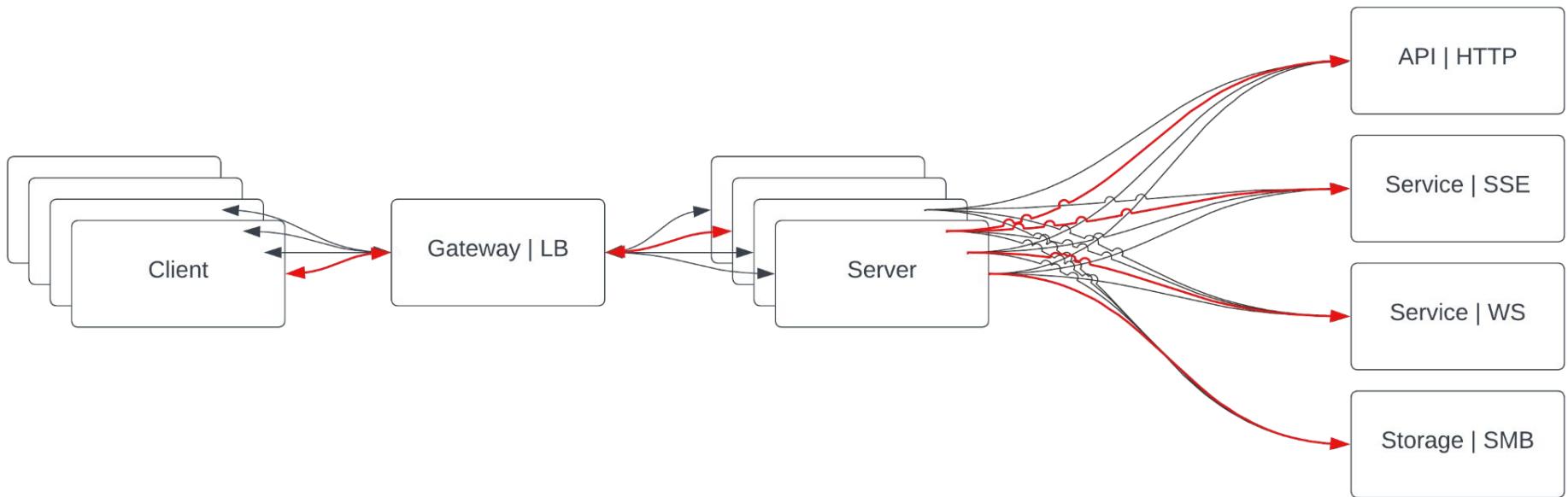
State management

- Session stickiness



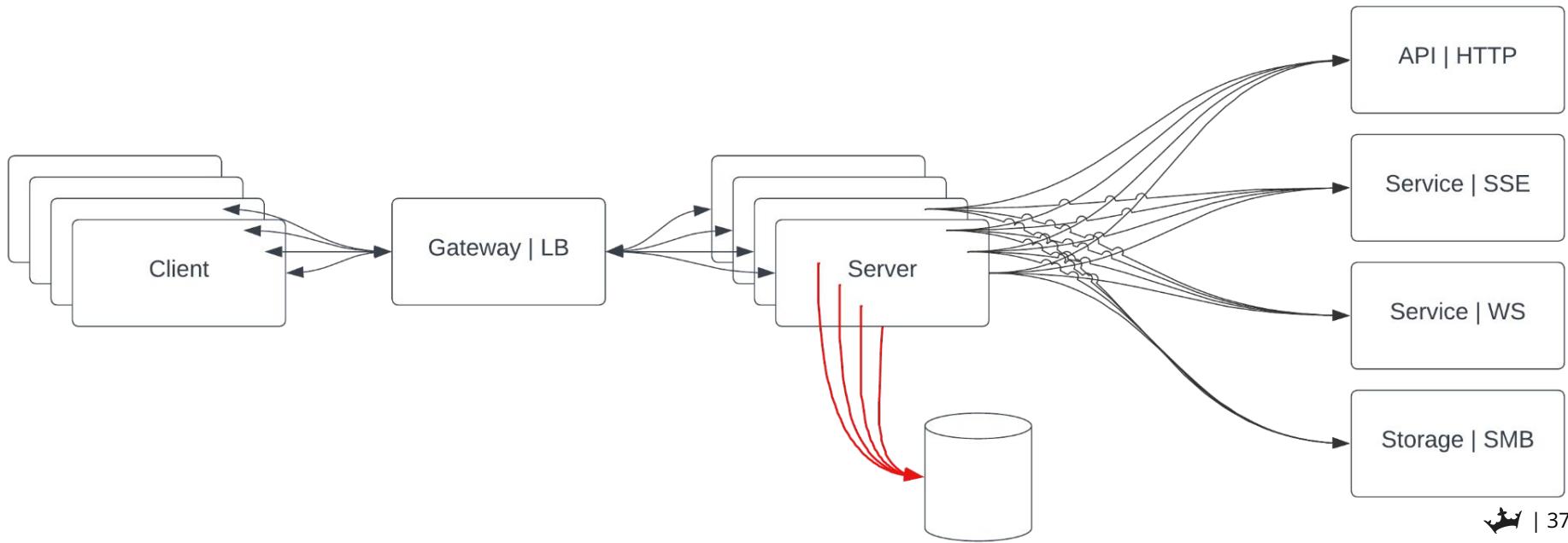
State management

- Session stickiness
- Persisting state in a store



State management

- Session stickiness
- Persisting state in a store

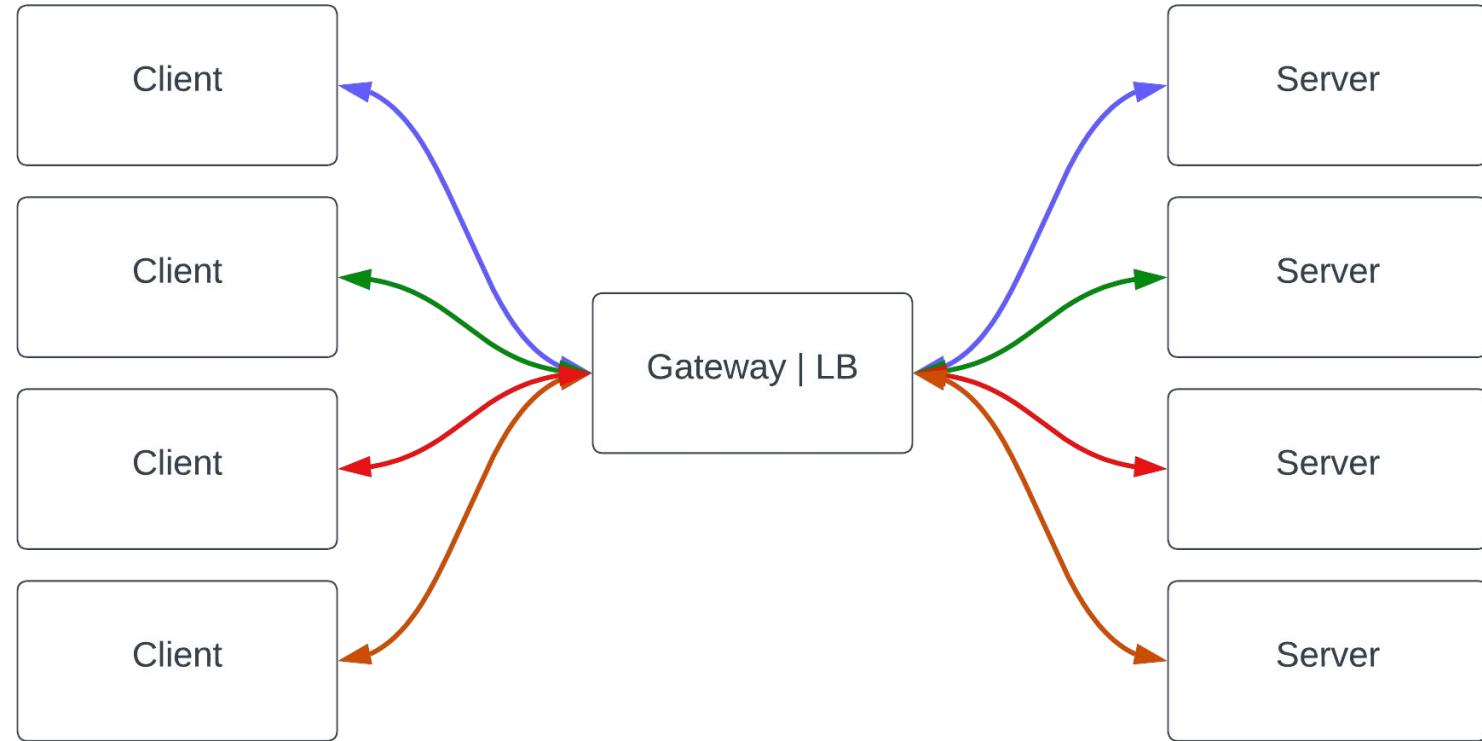


Load balancing long lived connections

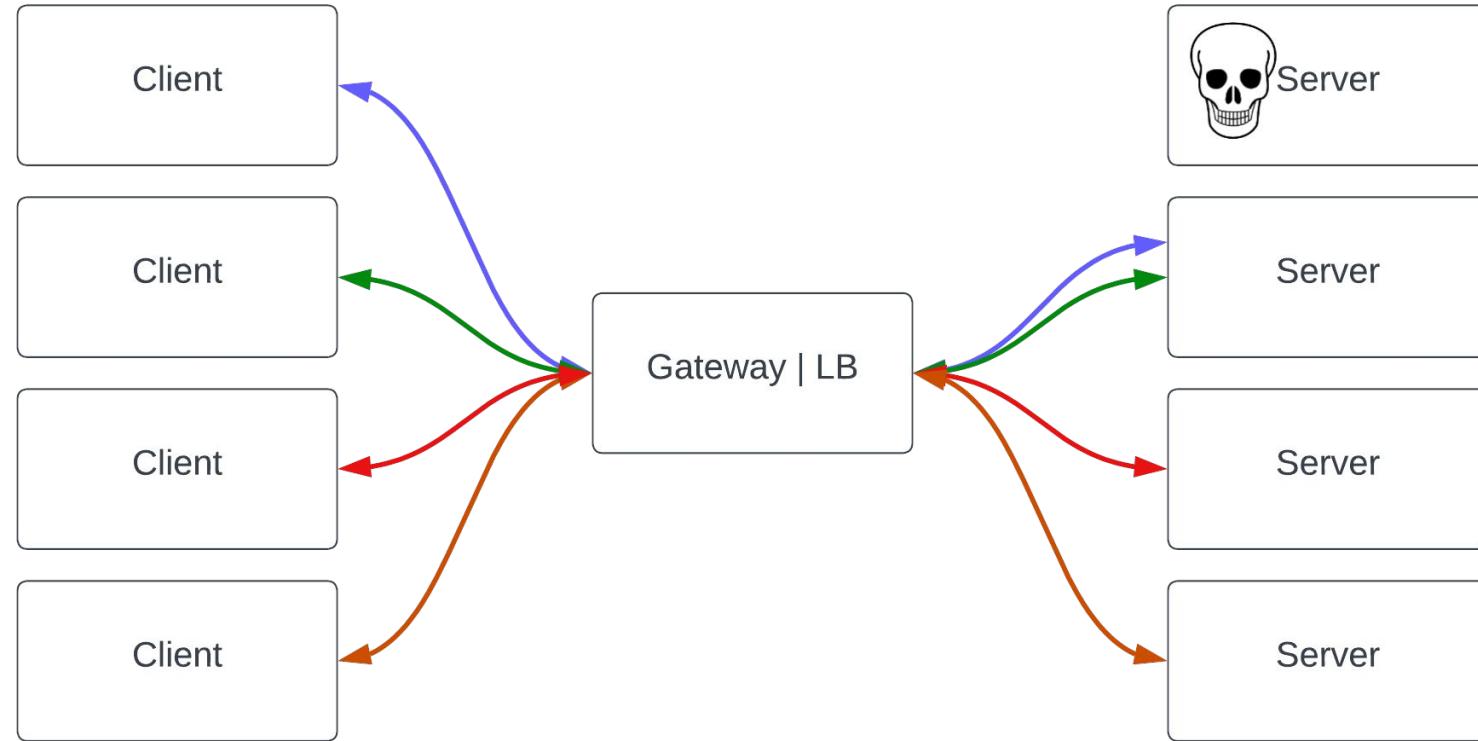
Long lived connections

- Load balancing
- Overloading the back-end
- Disconnects
- Deployment

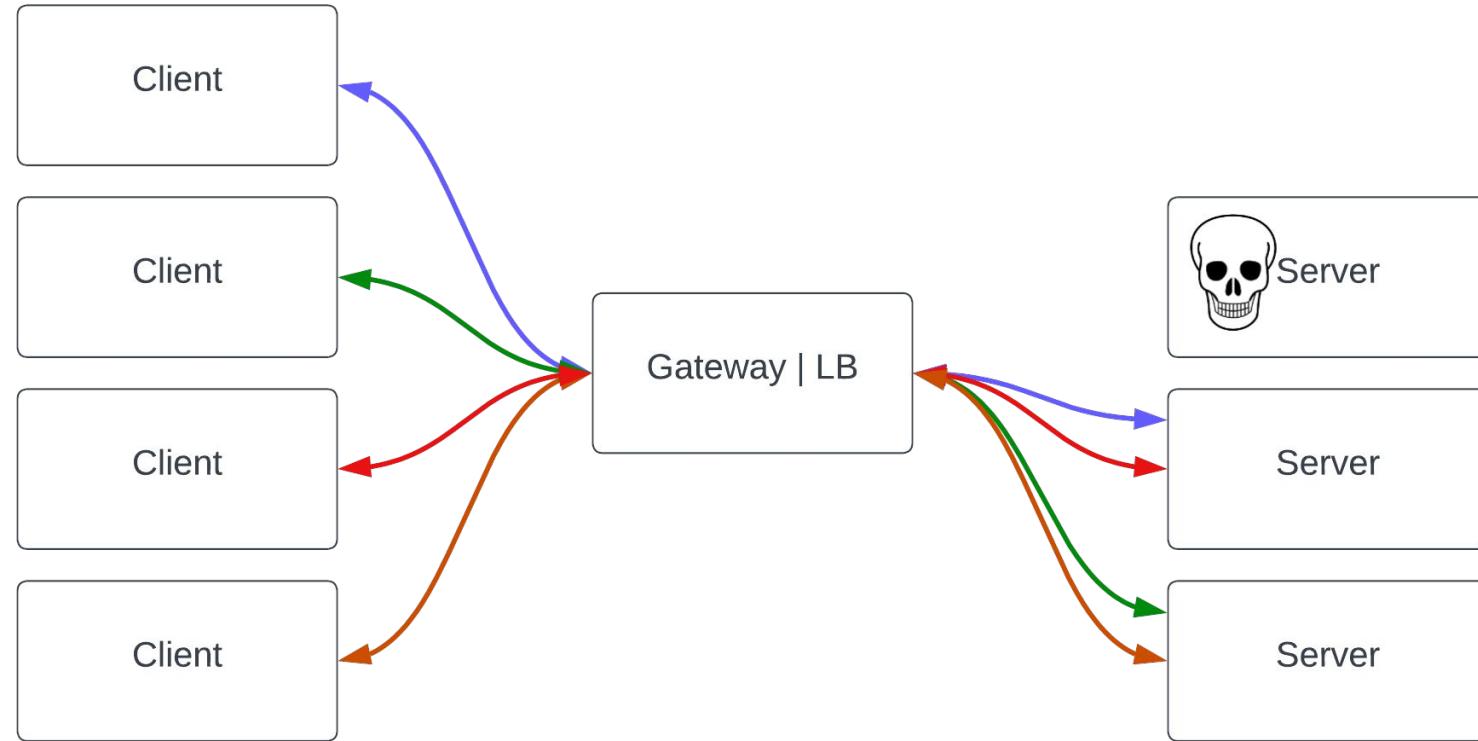
Load balancing



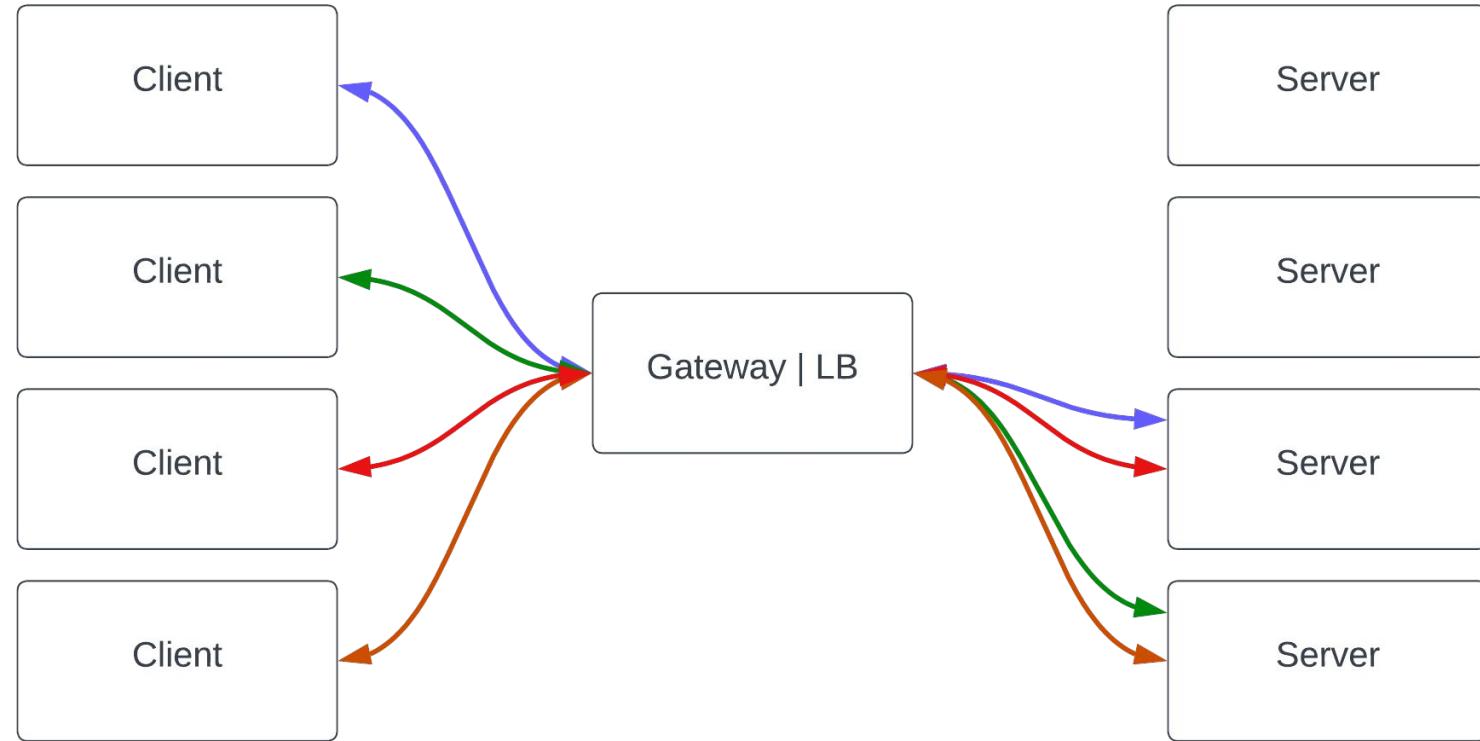
Load balancing



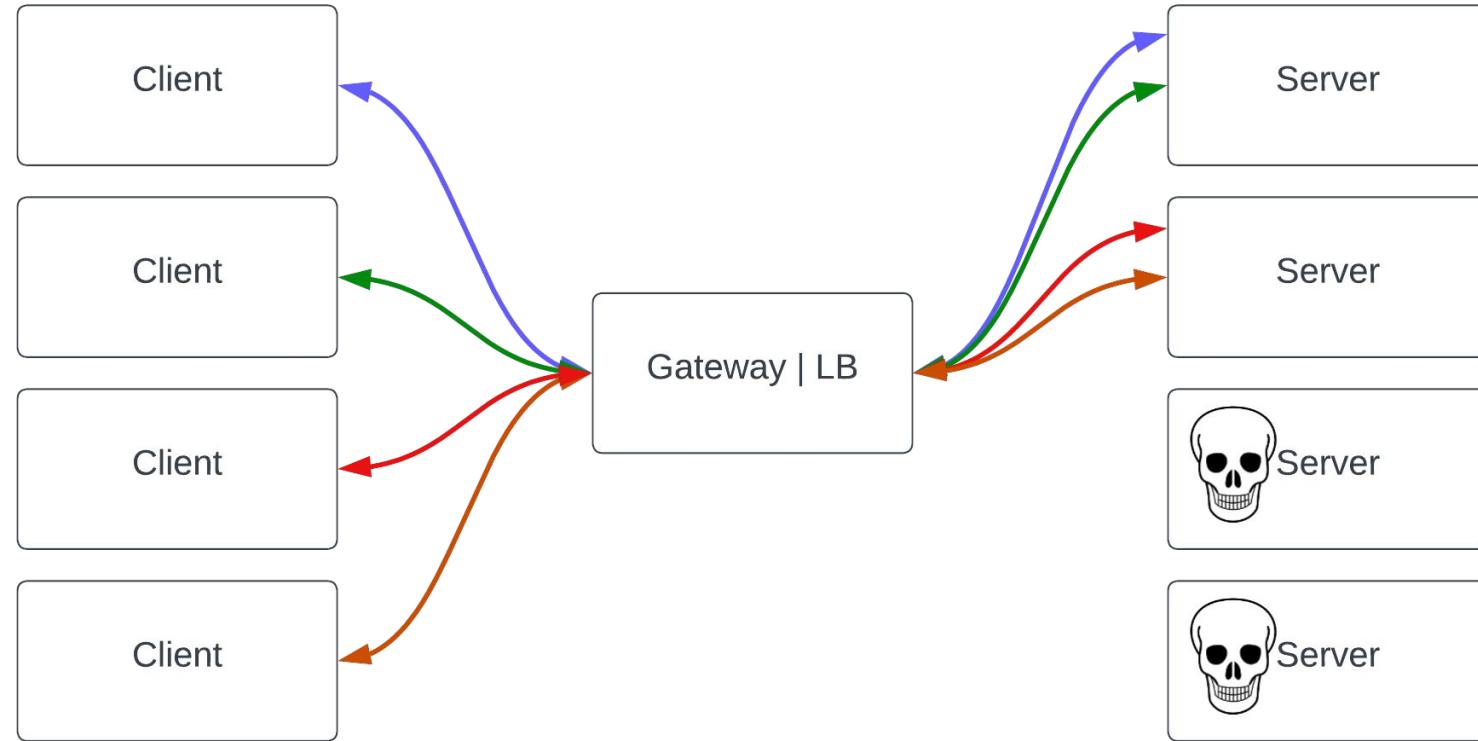
Load balancing



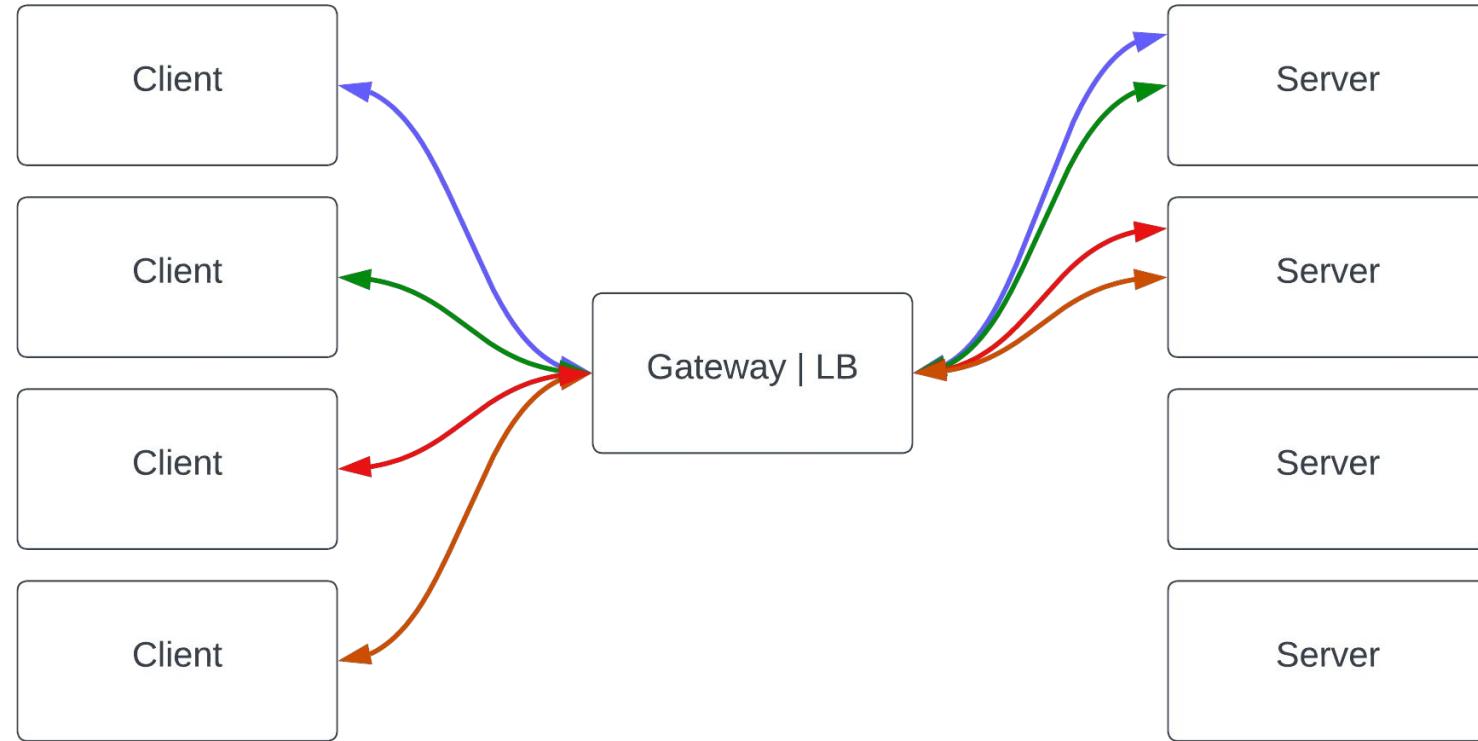
Load balancing



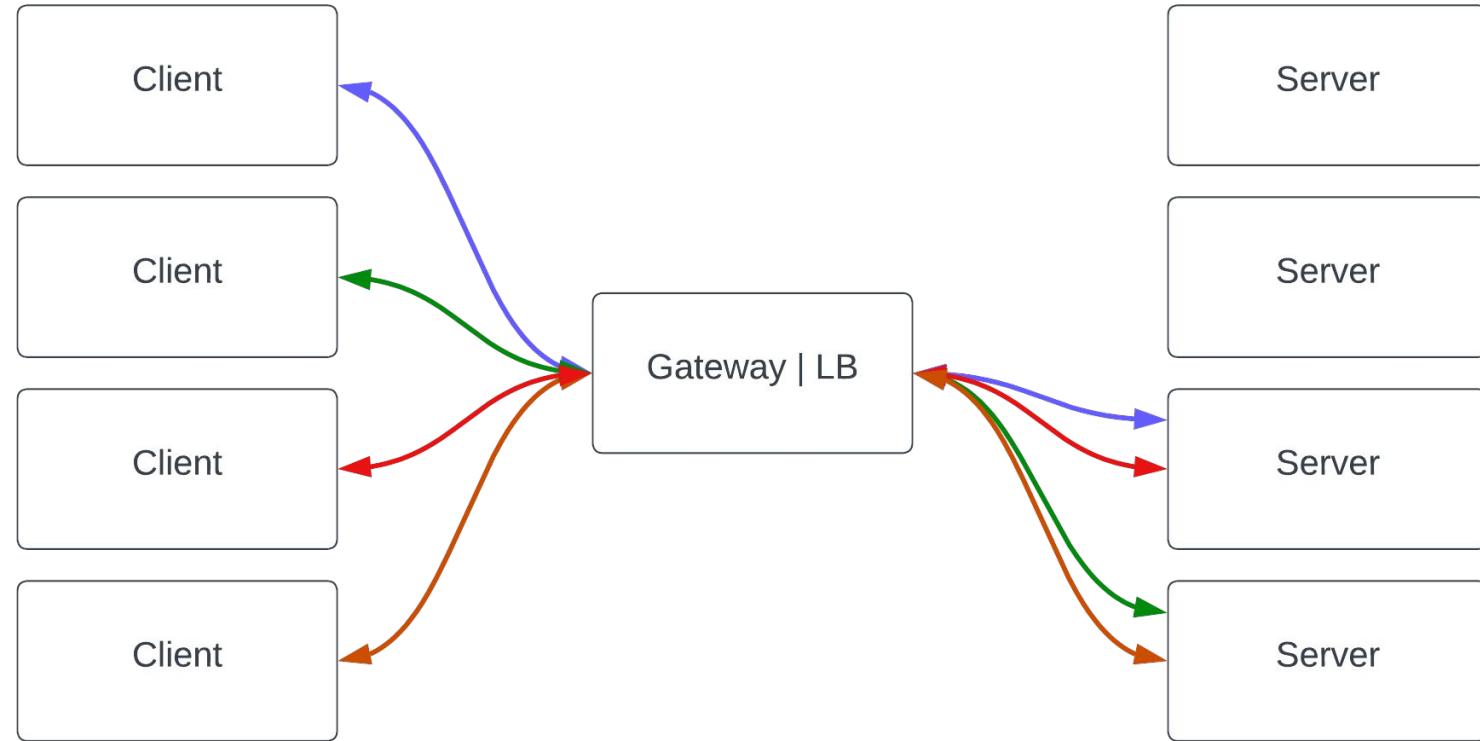
Load balancing



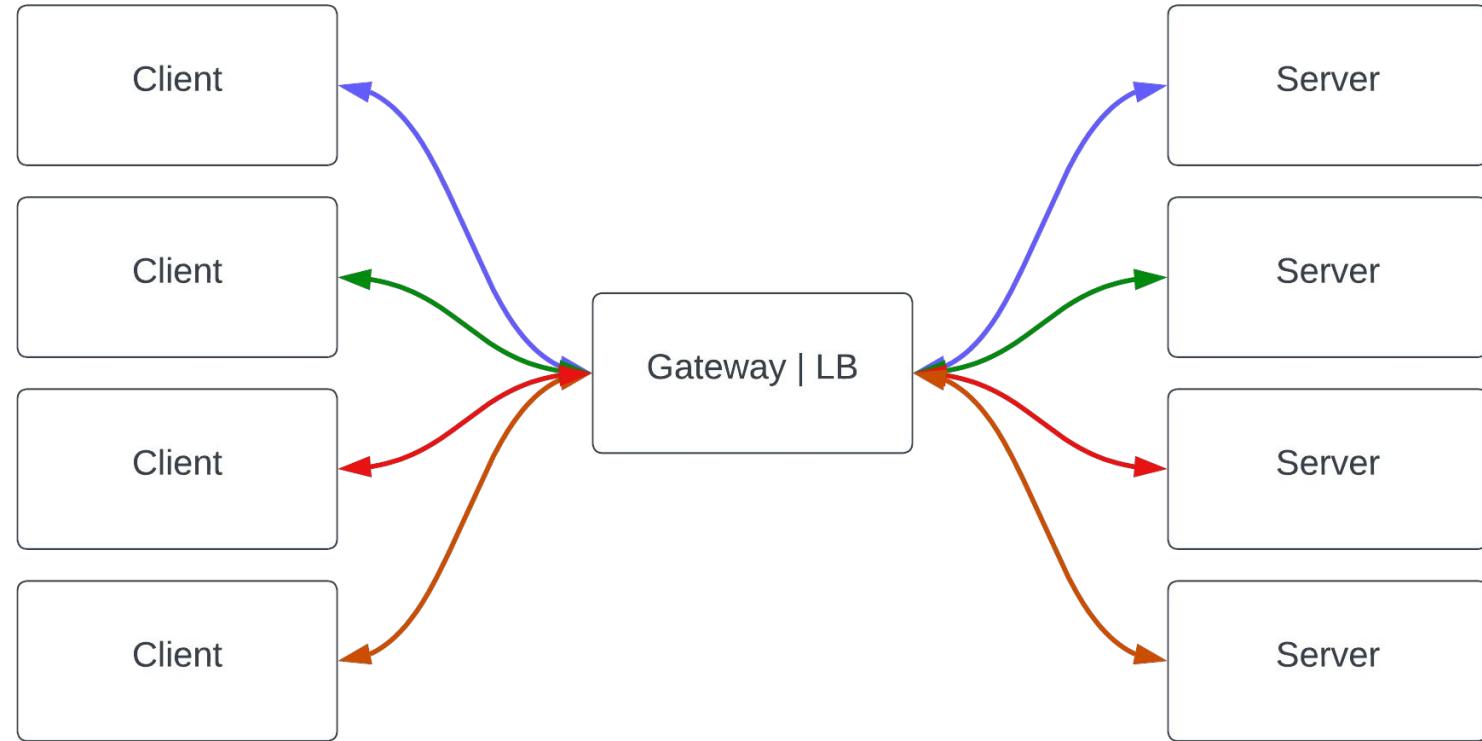
Load balancing



Load balancing

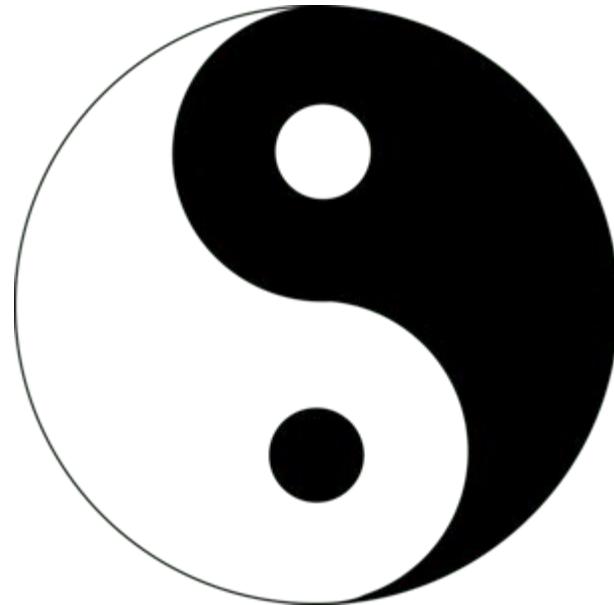


Load balancing



Load balancing

- **Gateway cutoff**
 - Client reconnection strategy
- Randomize connection lifetime
- Scheduled connection refresh
- Scale based on opened connections



Backpressure

Backpressure

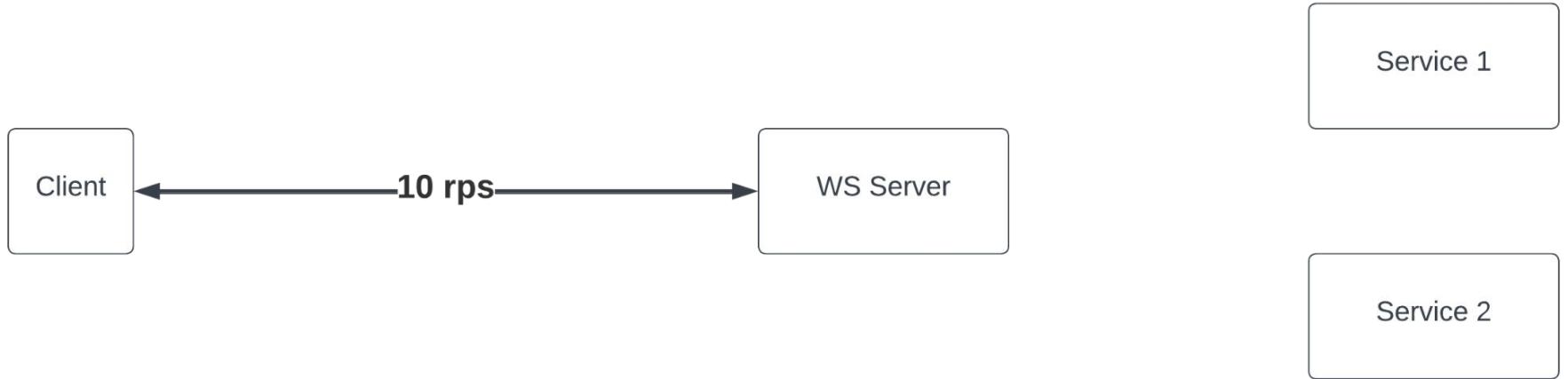
Client

WS Server

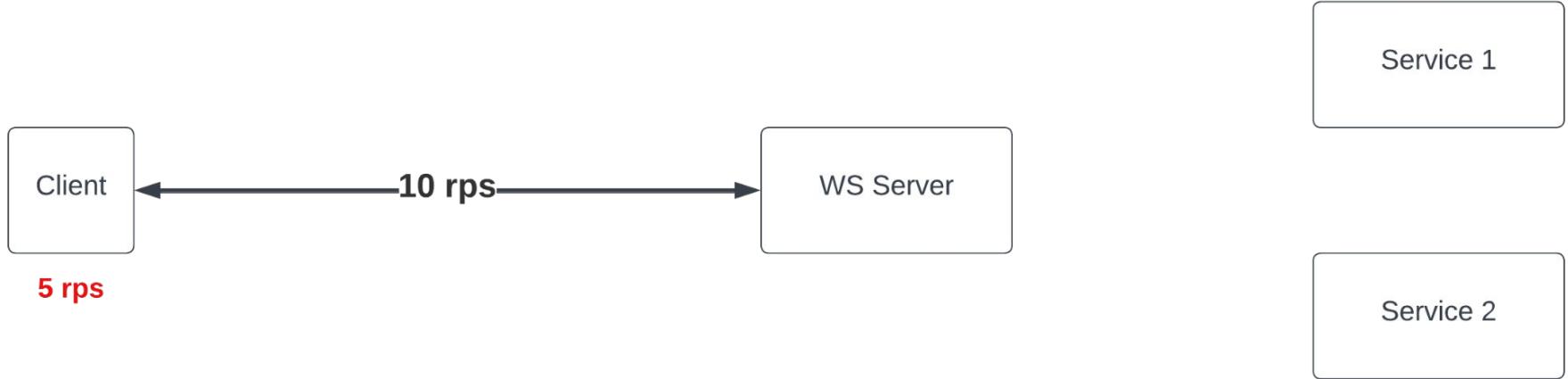
Service 1

Service 2

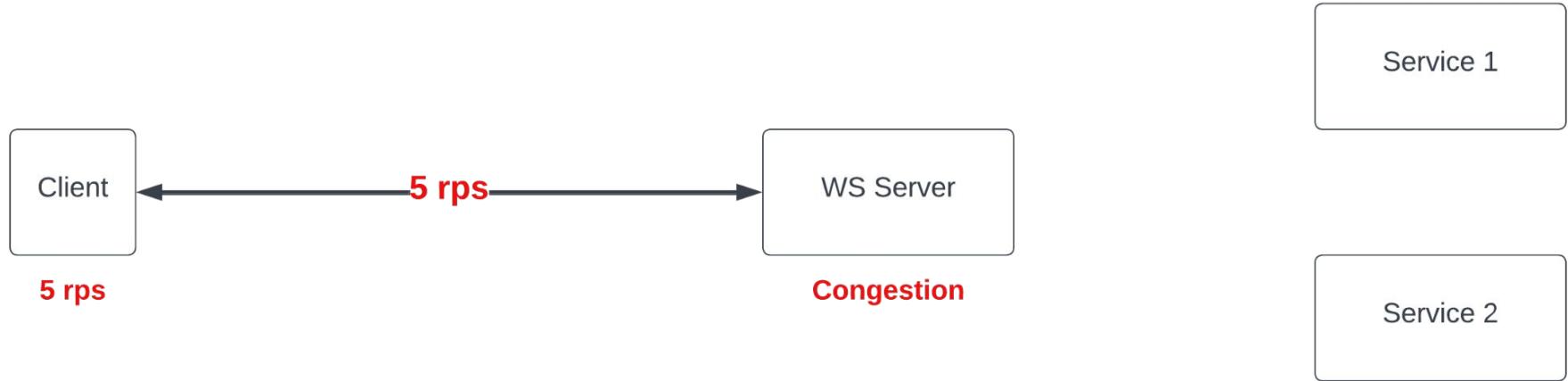
Backpressure



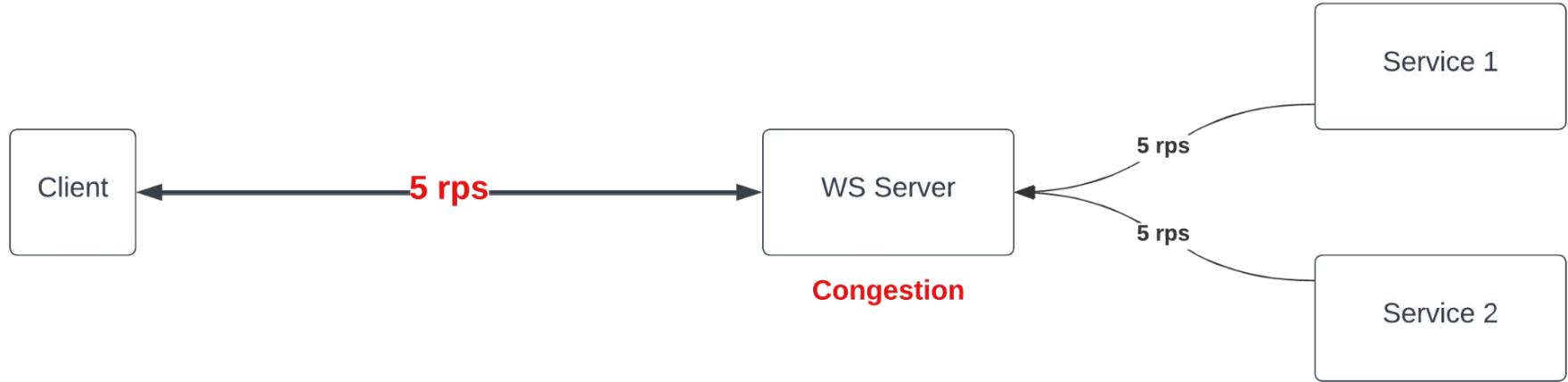
Backpressure



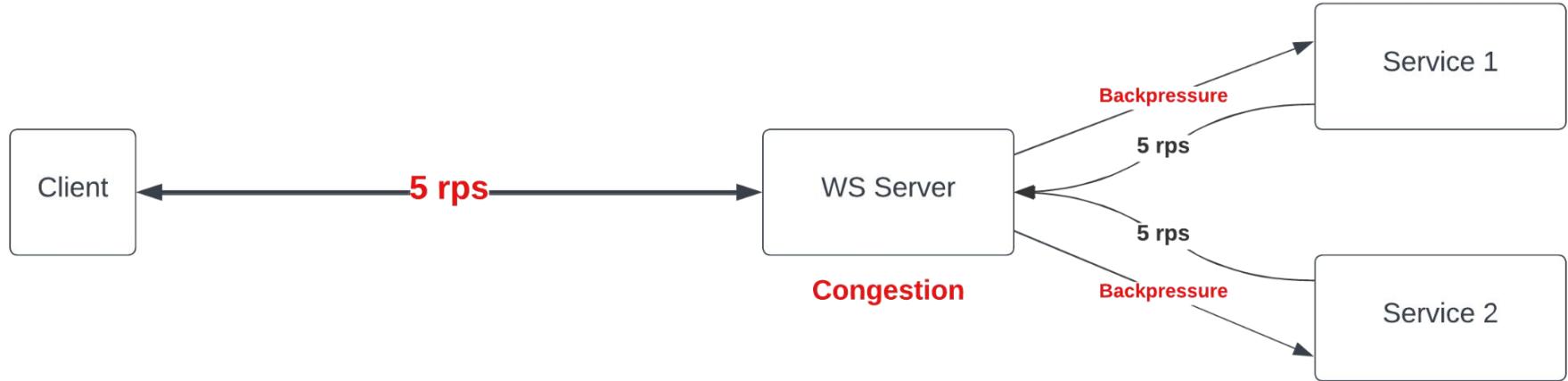
Backpressure



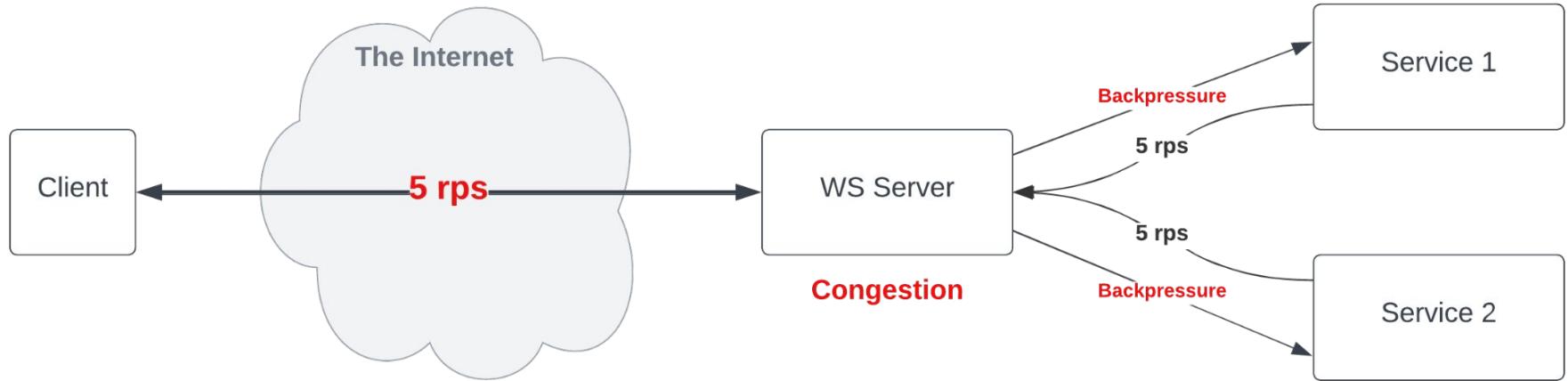
Backpressure



Backpressure

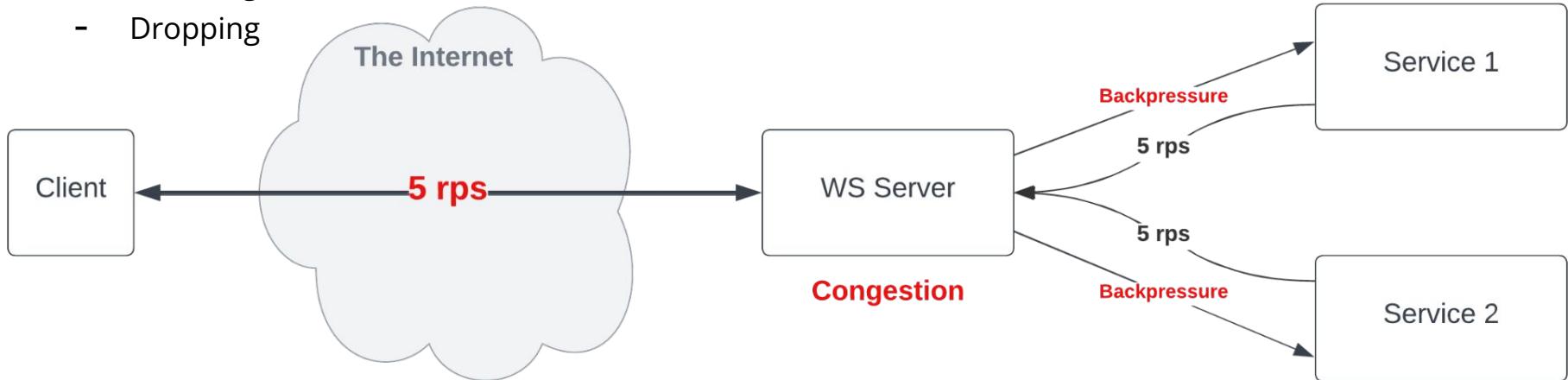


Backpressure



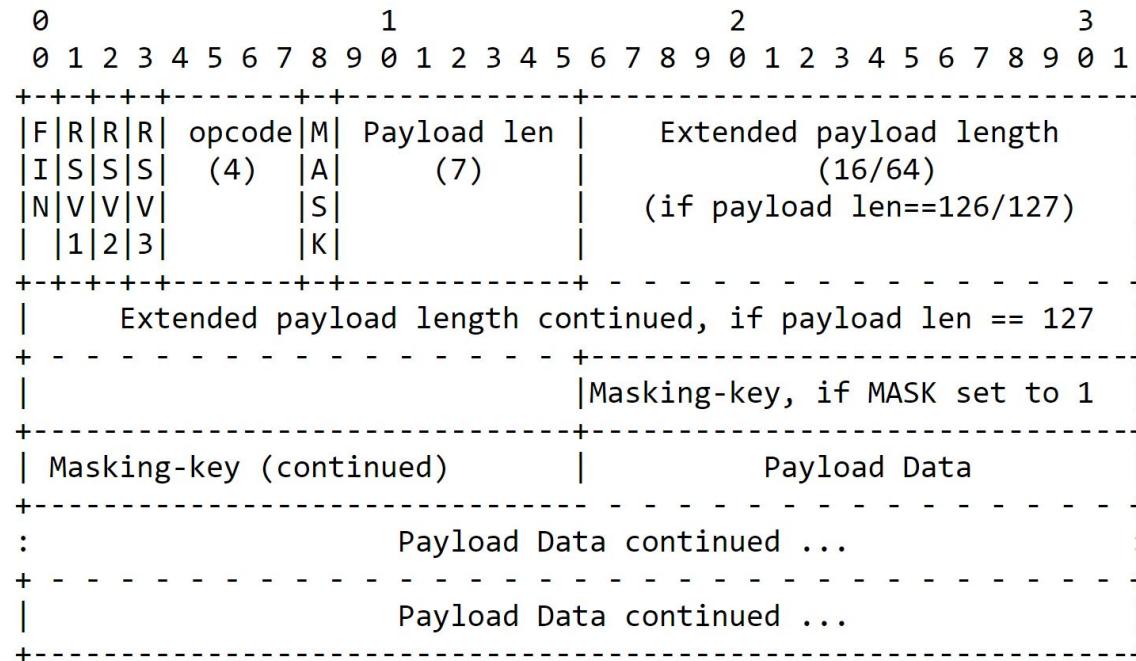
Backpressure

- Compression
- Compaction
 - Bulks
 - Sparse contracts
- Buffering
- Dropping



Protocol Efficiency

Protocol Efficiency



Protocol Efficiency

WebSocket Frame

Protocol Efficiency

Juicy JSON

Protocol Efficiency

Juicy JSON

Packet

Protocol Efficiency

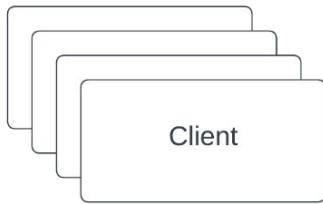


Security

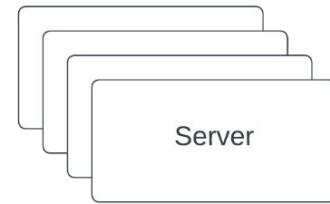
Security

- Authentication
- Authorisation
- User input

Authentication and Authorization



Gateway | LB



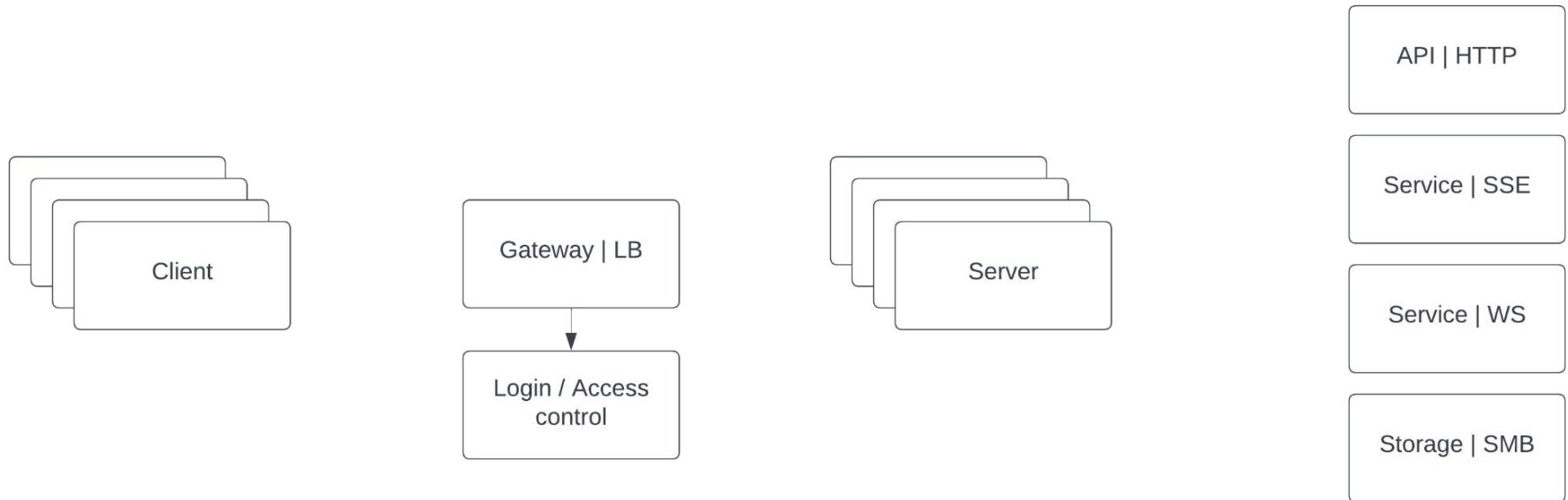
API | HTTP

Service | SSE

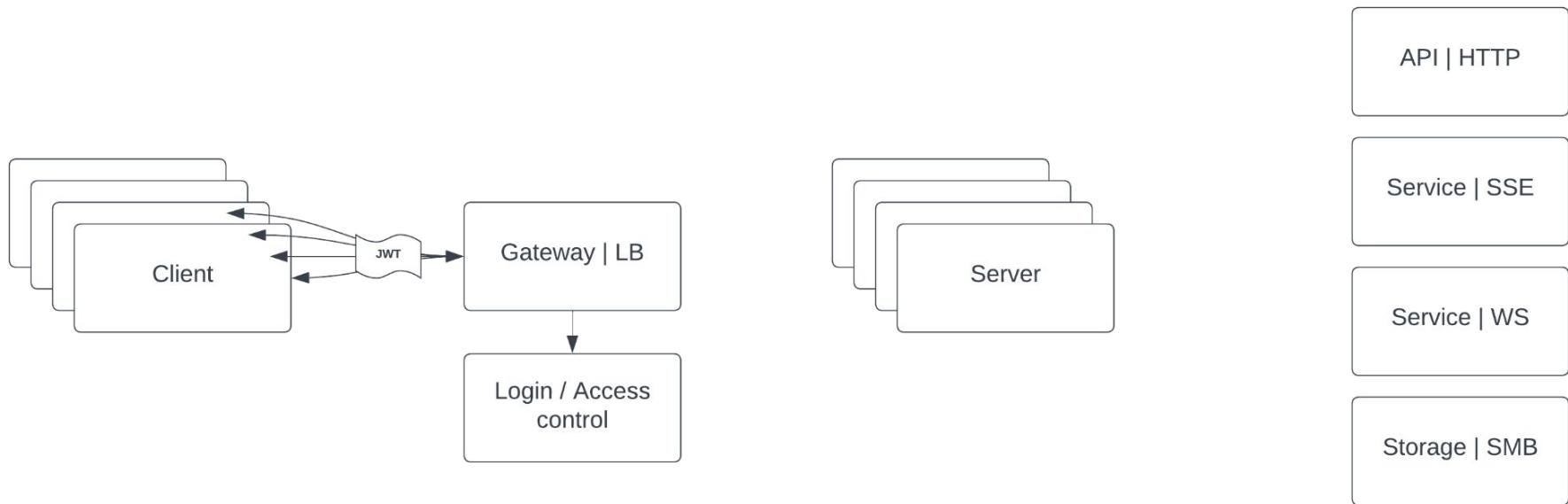
Service | WS

Storage | SMB

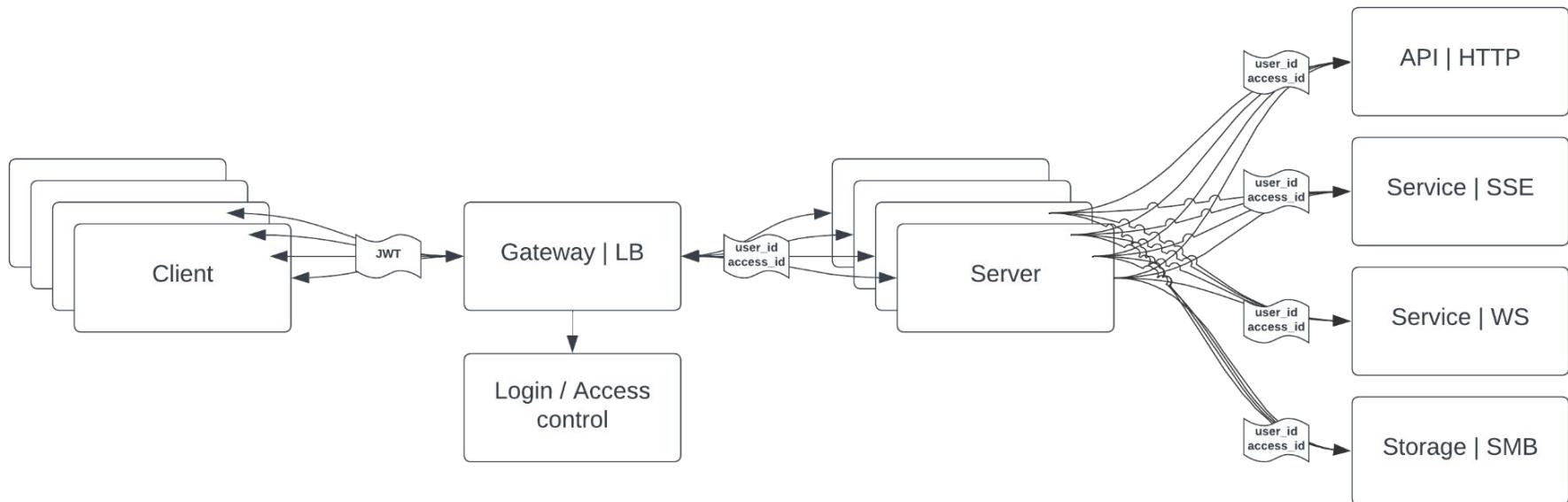
Authentication and Authorization



Authentication and Authorization

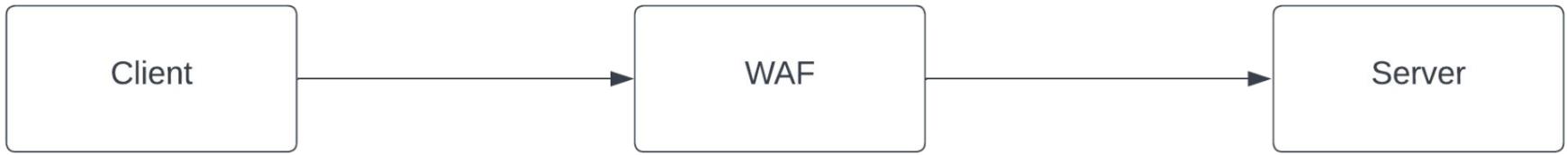


Authentication and Authorization

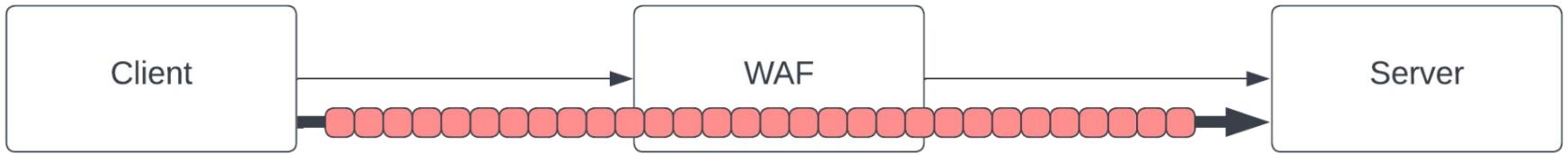


User input

User input



User input



Application Performance Monitoring

Application Performance Monitoring

- Amazing tooling for HTTP traffic
- Nothing works out of the box for WS
- Debugging



Honorable mentions (challenges edition)

- Head-of-line blocking
- No caching
- Connection fanout
- Sniffing
- CSWH

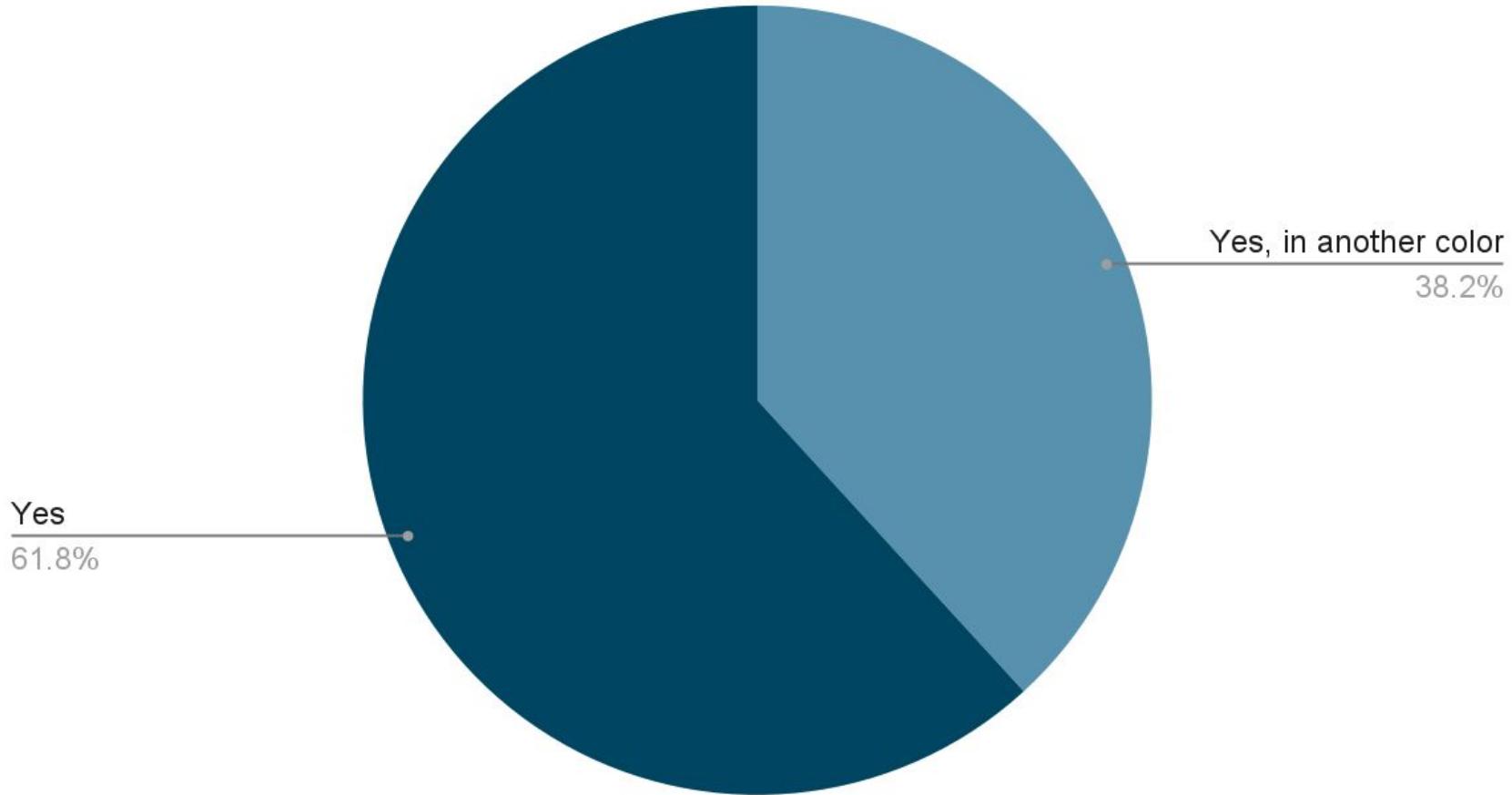


THE GAME
INSIDE
THE GAME.

Is it worth it?



Is it worth it?



Alternatives*

- Polling
- SSE
- WebRTC

Alternatives: Polling

Pros

- Downloading big amounts of data
- Works everywhere

Cons

- Headers overhead
- Maximizing latency
- Timeouts

Alternatives: SSE

Pros

- Read-only stream of data
- It's a protocol
- Automatic retry + managing deltas
- No header overhead

Cons

- It's a protocol
- Fixed direction of data flow from server to client
- No out of the box solution on how to change the subscription
- No headers
- Same challenges as with managing WS

Alternatives: WebRTC

Pros

- Polyfill for SSE is built on top of it
- Mix and match what you need
- UDP or TCP
- Packet loss tolerance
- Good for streaming (audio, video)

Cons

- Mix and match what you need
- Learning curve
- SCTP, SRTP, DTLS, ICE, STUN, TURN...



Key Takeaways



THE GAME
INSIDE
THE GAME.

Q&A

