



JEECONF 2018

# BUILDING SCALABLE WEBSOCKET BACKEND

**KONSTANTIN SLISENKO**  
**LEAD SOFTWARE ENGINEER**

MAY 18-19, 2018



# Konstantin Slisenko

- Java Team Lead at EPAM
- I'm building trading software
- 8+ years with Java
- 3+ years with WebSocket



[github.com/kslisenko](https://github.com/kslisenko)



[kslisenko@gmail.com](mailto:kslisenko@gmail.com)

☆ FX CMD IND EQ ETF CSH

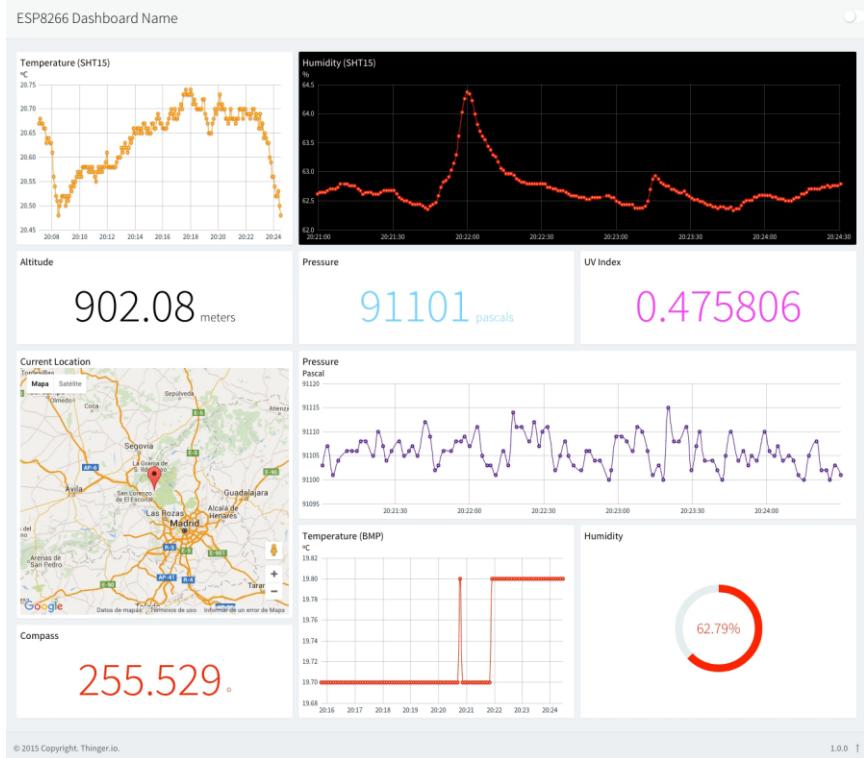
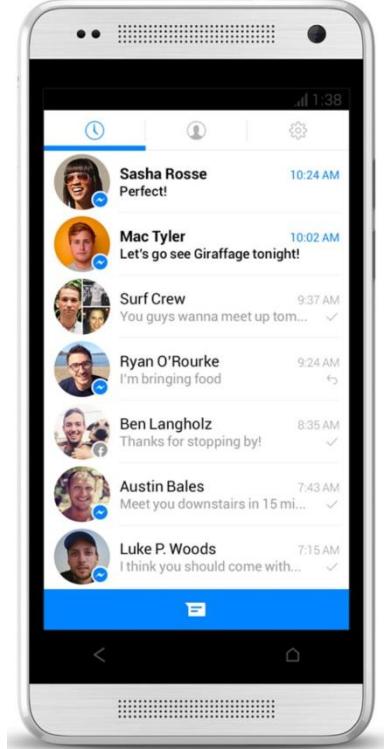
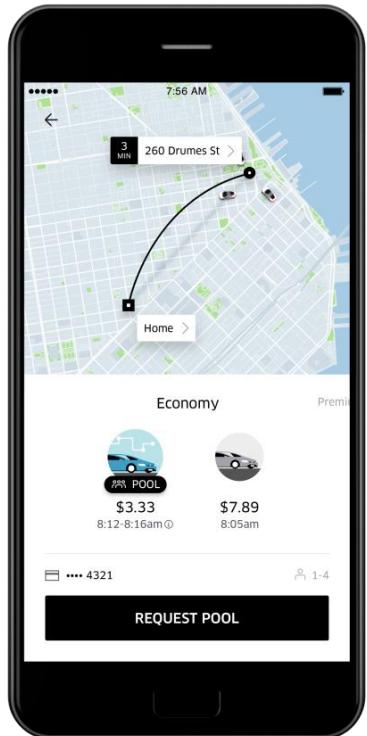
Search e.g. EURUSD

SYMBOL | CHANGE(..) BID ASK

	Sell	0.50	Buy	1.23239
1.23231		0.50		
Low: 1.22945	+0.14%		High: 1.23358	
300K	1.2323	1.23238		300K
800K	1.23229	1.2324		1.0M
1.0M	1.23226	1.23242		1.2M
4.0M	1.2322	1.2325		3.5M
4.0M	1.23212	1.2326		4.0M
▼ GBPUSD FX	+0.20%	1.39620	1.39646	
▲ USDJPY FX	-0.58%	105.664	105.681	
▼ AUDUSD FX	-0.13%	0.77789	0.77808	
▼ BITCOIN FX	-2.39%	8085.30	8182.23	
▼ ETHEREUM F	-2.65%	590.453	604.518	
▼ US500 IND	-0.09%	2754.8	2755.5	
▲ DE30 IND	+0.09%	12393.1	12394.0	
▼ UK100 IND	-1.12%	7060.4	7062.1	


+ EURUSD (H1) × OIL (H1) × GOLD (H1) × DE30 (H1) ×
Open positions □ × Pending orders History Account summary

POSITION	TYPE	VOLUME	OPEN TIME	OPEN PRICE	SL	TP	EXPIRATION DA...	MARKET PRICE	COMMISSION	SWAP	PROFIT	CLOSE ▾
----------	------	--------	-----------	------------	----	----	------------------	--------------	------------	------	--------	---------



# WHAT IS WEBSOCKET



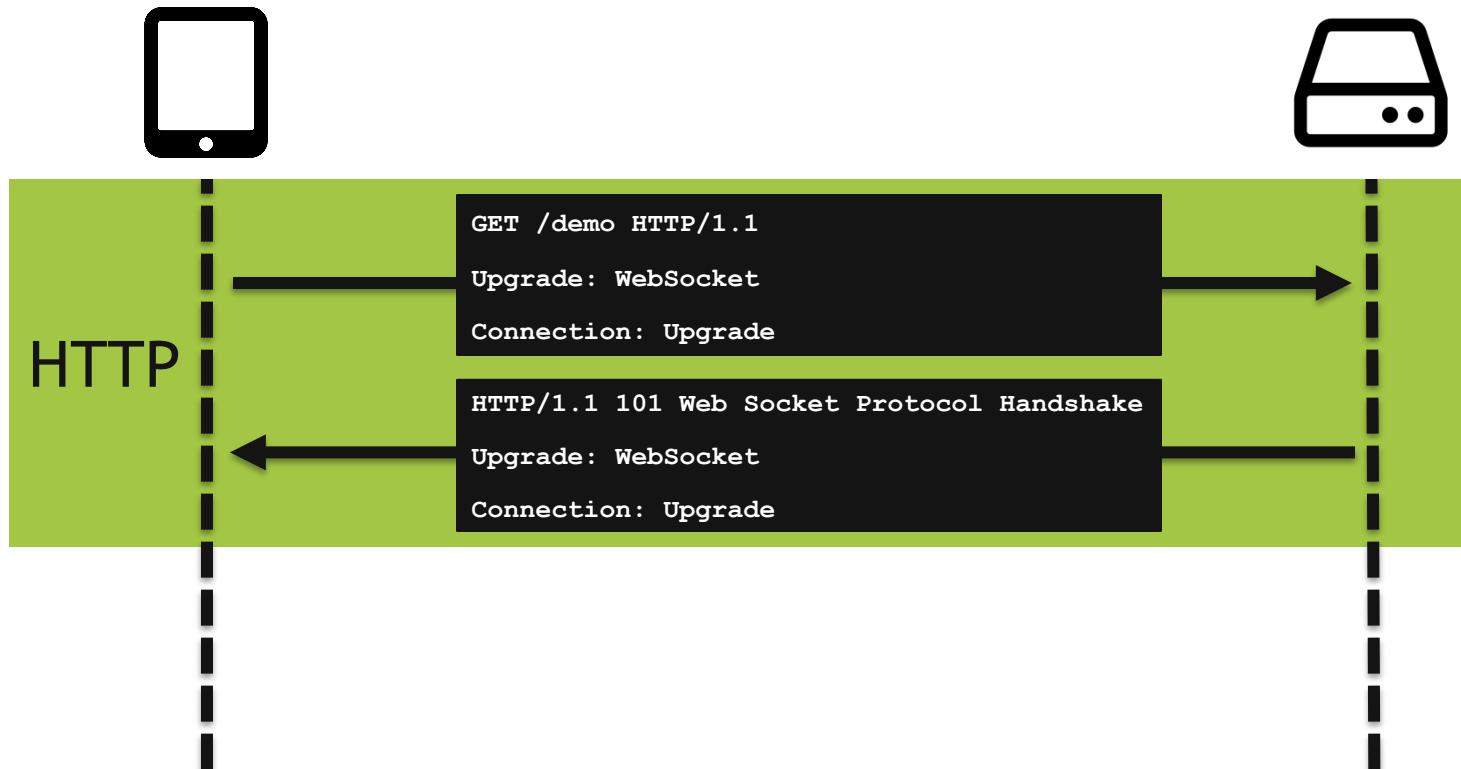
standard: [tools.ietf.org/html/rfc6455](https://tools.ietf.org/html/rfc6455)

# WHAT IS WEBSOCKET



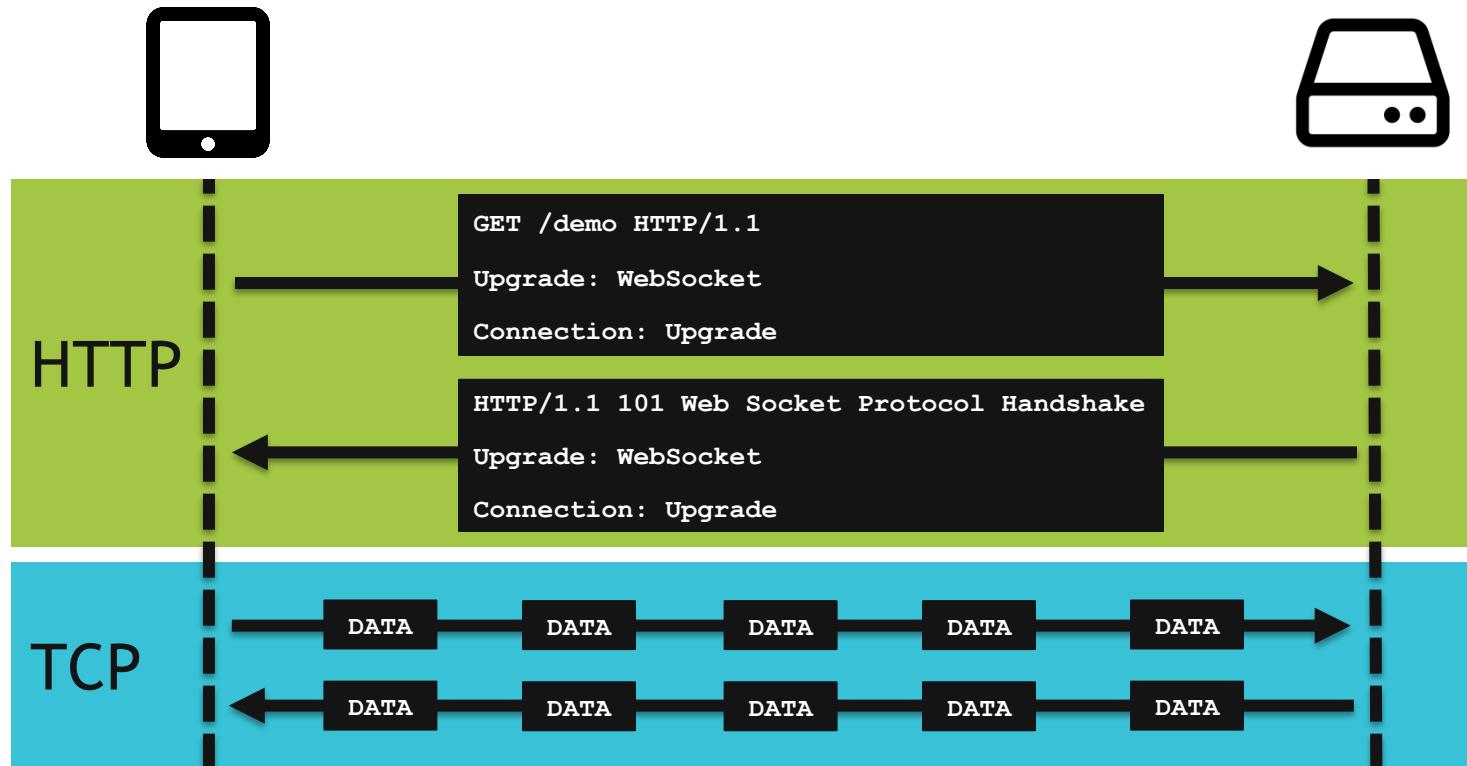
standard: [tools.ietf.org/html/rfc6455](https://tools.ietf.org/html/rfc6455)

# WHAT IS WEBSOCKET



standard: [tools.ietf.org/html/rfc6455](http://tools.ietf.org/html/rfc6455)

# WHAT IS WEBSOCKET



standard: [tools.ietf.org/html/rfc6455](http://tools.ietf.org/html/rfc6455)

## WEBSOCKET PROVIDES

---

- Permanent connection



## WEBSOCKET PROVIDES

---

- Permanent connection
- Bidirectional communication



## WEBSOCKET PROVIDES

---

- Permanent connection
- Bidirectional communication
- Sending text or binary data



## WEBSOCKET PROVIDES

---

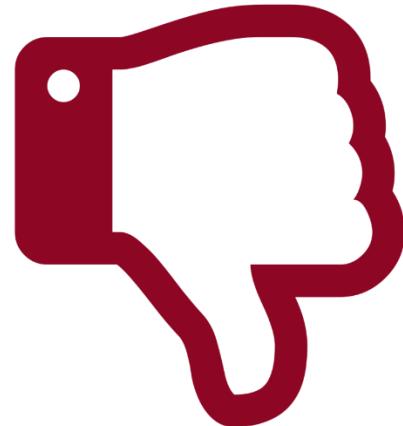
- Permanent connection
- Bidirectional communication
- Sending text or binary data
- Guaranteed delivery (TCP/IP)



## WEBSOCKET DOESN'T PROVIDE

---

- API and application protocol



## WEBSOCKET DOESN'T PROVIDE

---

- API and application protocol
- Network failure handling\*



\*Only basic handling

## WEBSOCKET DOESN'T PROVIDE

---

- API and application protocol
- Network failure handling\*
- Network slowness handling\*



\*Only basic handling

## WEBSOCKET DOESN'T PROVIDE

---

- API and application protocol
- Network failure handling\*
- Network slowness handling\*
- Bandwidth/frequency limitation



\*Only basic handling

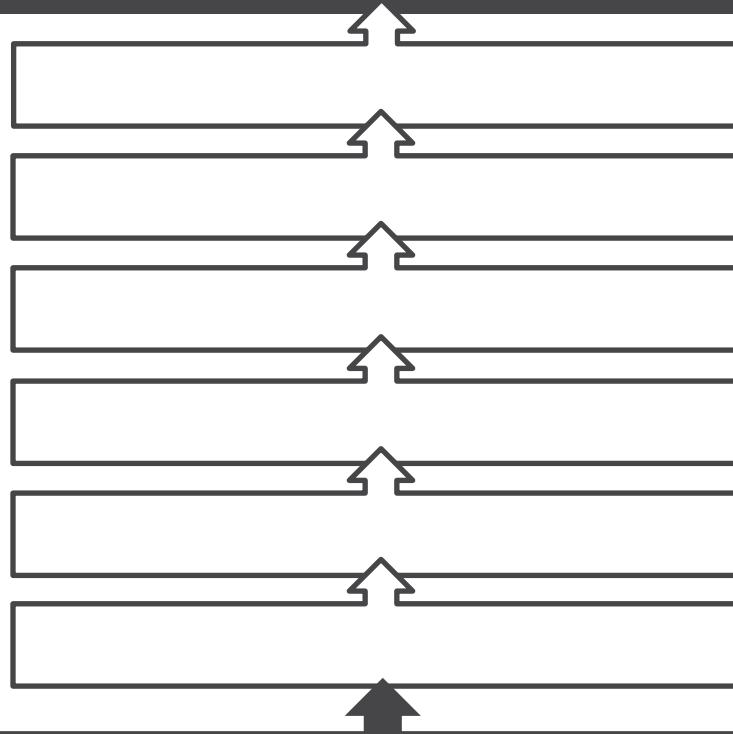


ONE DOES NOT SIMPLY

BUILD HIGH PERFORMANCE APPLICATION  
USING ONLY STANDARD WEB SOCKET FEATURES



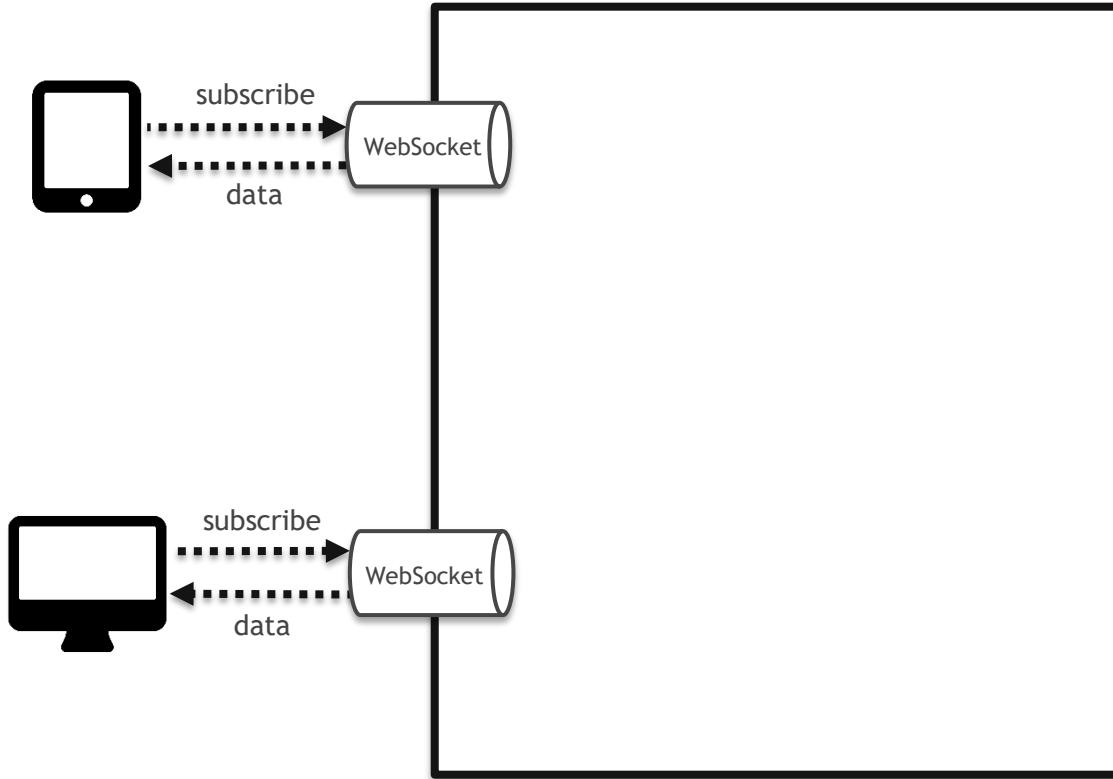
# High performance WebSocket backend

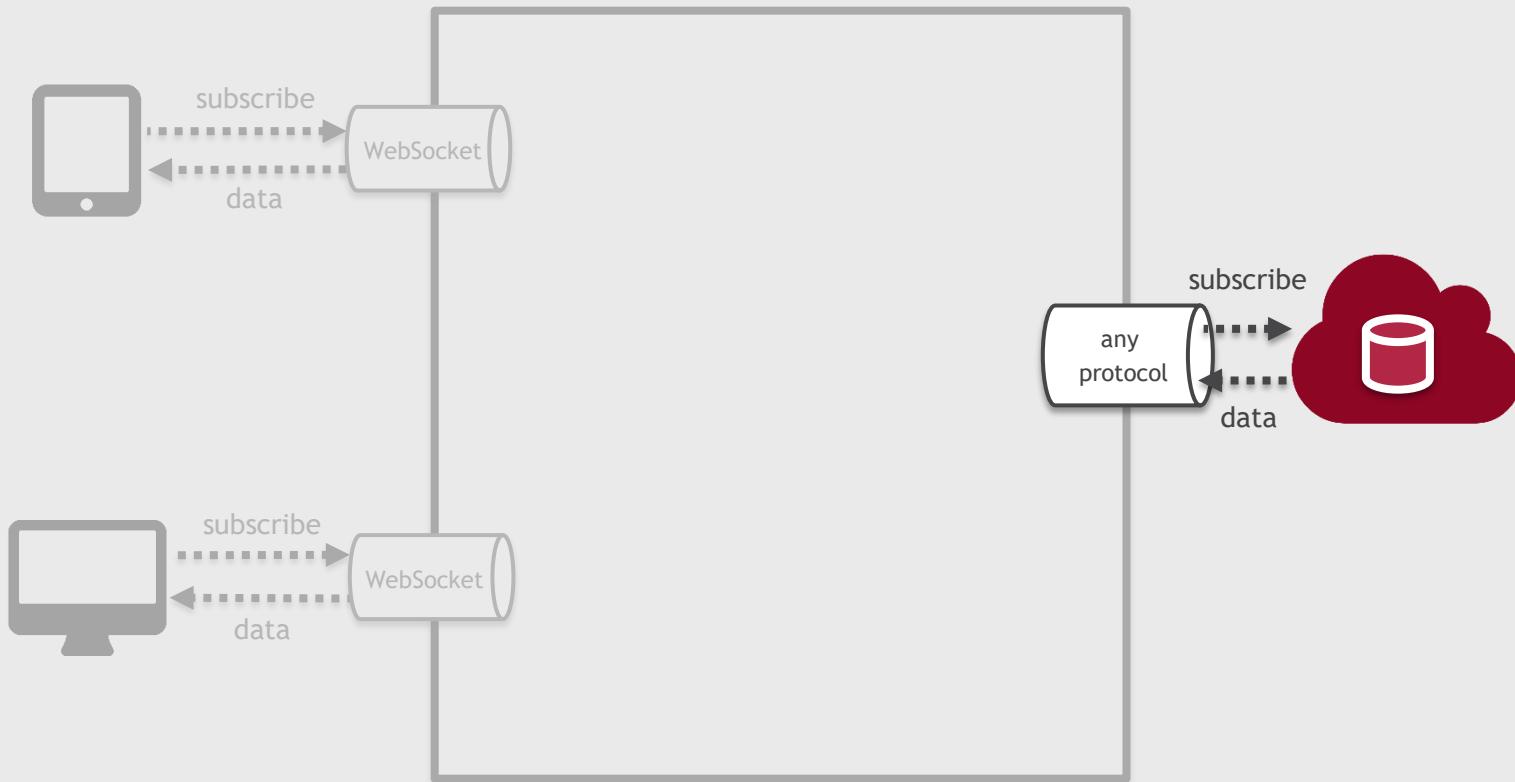


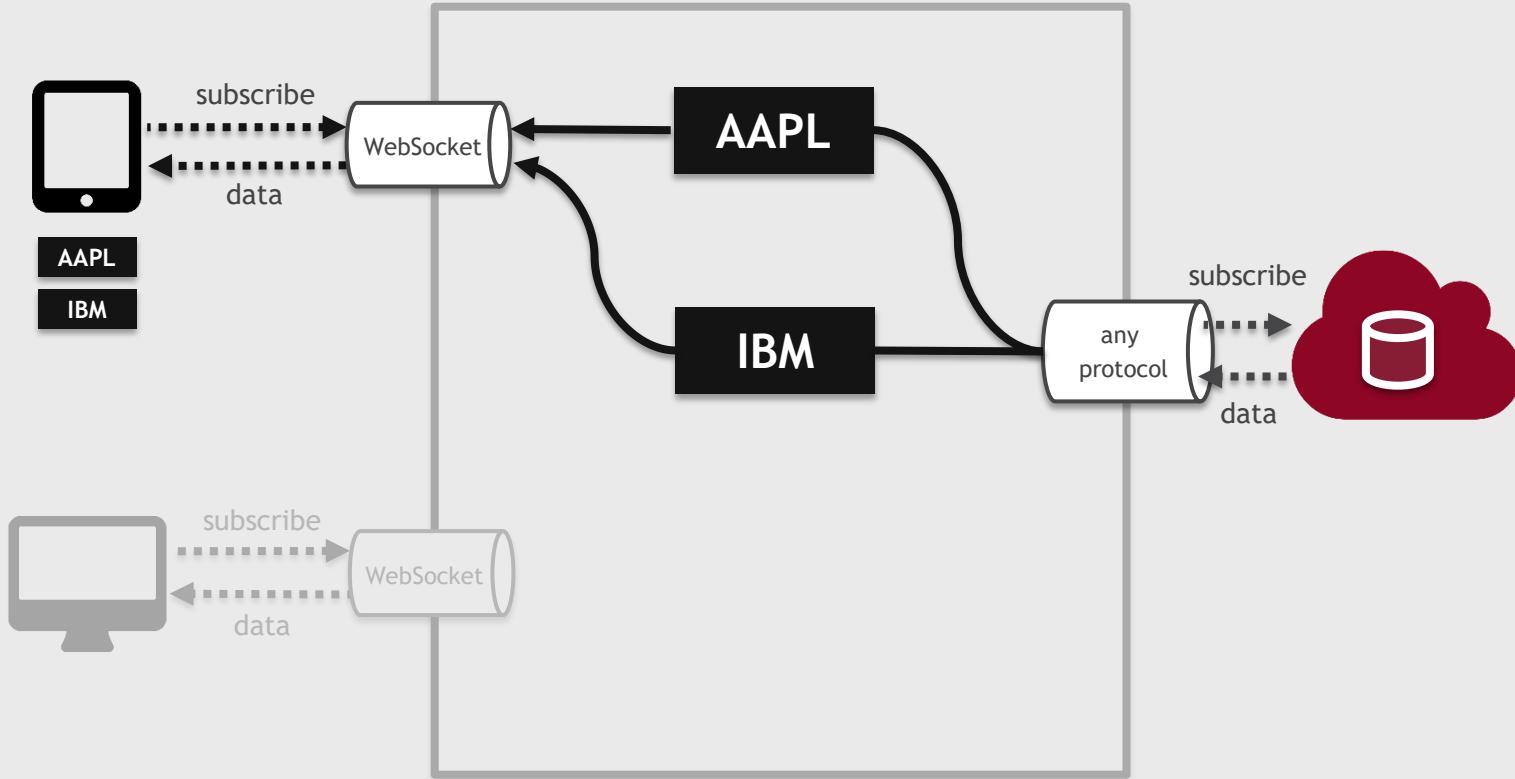
Basic WebSocket backend

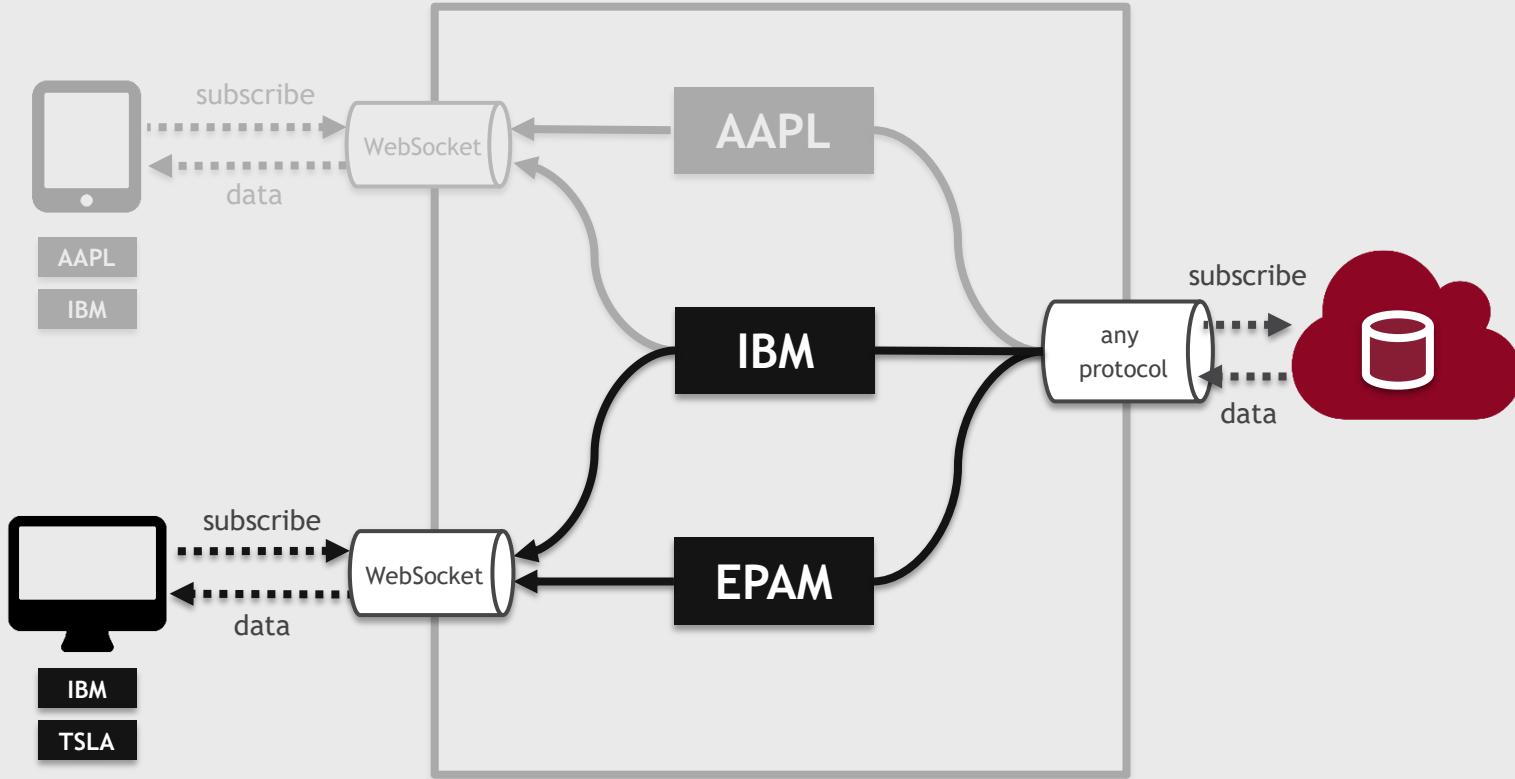


# BASIC WEBSOCKET BACKEND









A close-up photograph of a person's hands typing on a laptop keyboard. The laptop is open, showing its screen and keyboard. A white computer mouse sits to the right of the keyboard. The person is wearing a green and white checkered shirt. In the top left corner of the image, there is a small white rectangular button with the text "LIVE DEMO" in black capital letters.

LIVE DEMO

# BASIC WEBSOCKET BACKEND

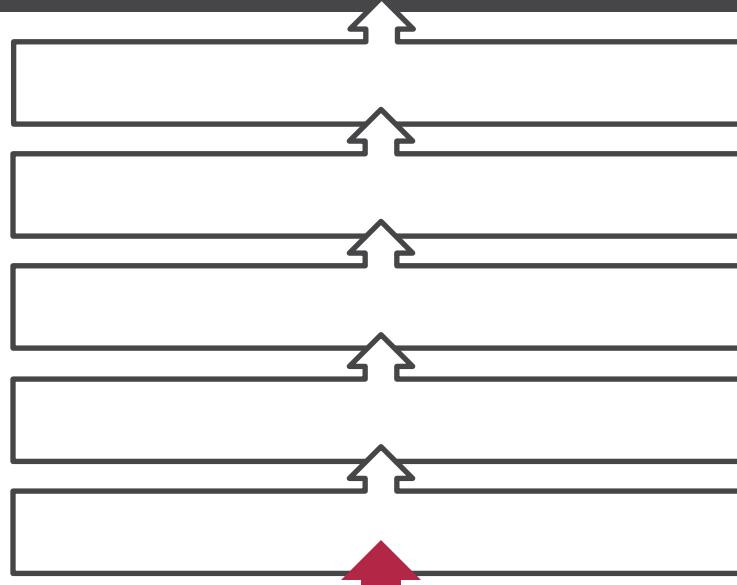


# TOO MUCH DATA?

SEND LESS DATA!



# High performance WebSocket backend



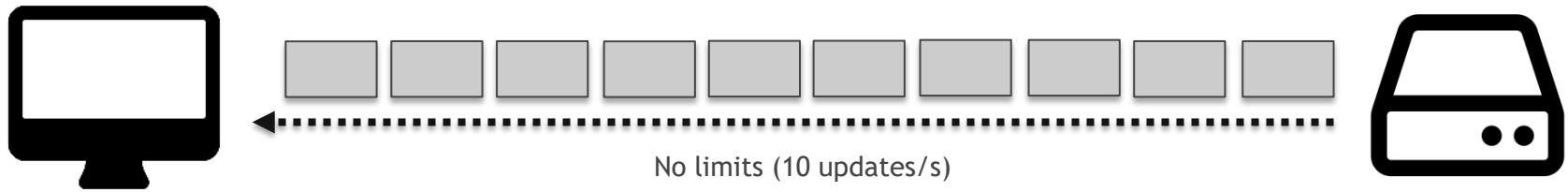
Max frequency limit



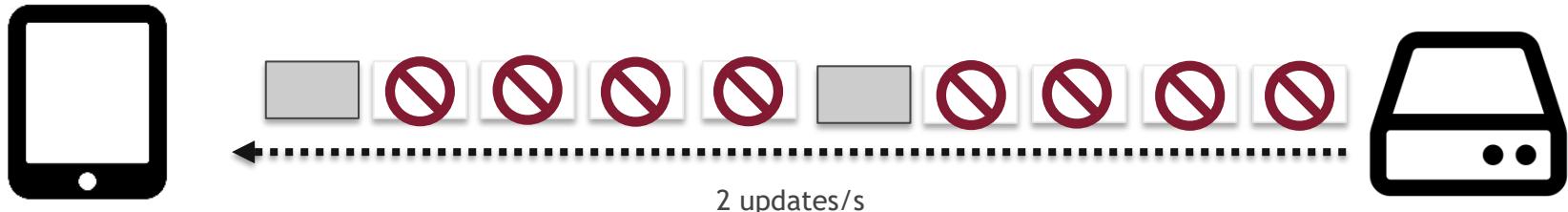
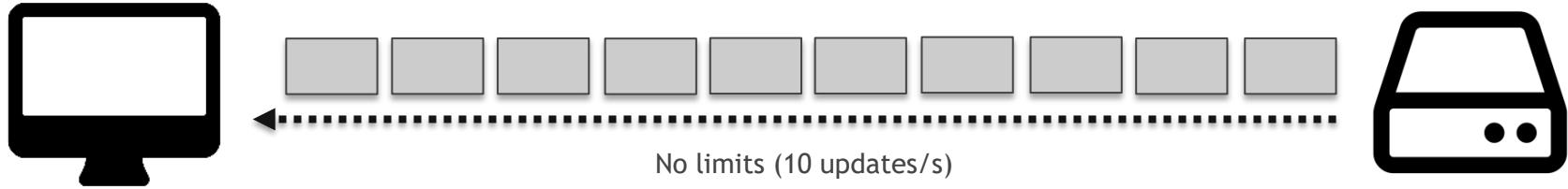
## Basic WebSocket backend

**HOW DO WE  
SEND LESS DATA?**

# RESAMPLING



# RESAMPLING



# RESAMPLING + CONFLATION



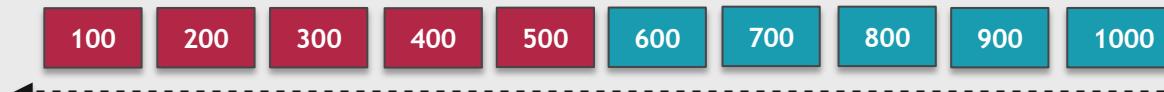
No limits (10 updates/s)



2 updates/s (last available)



# RESAMPLING + CONFLATION



No limits (10 updates/s)

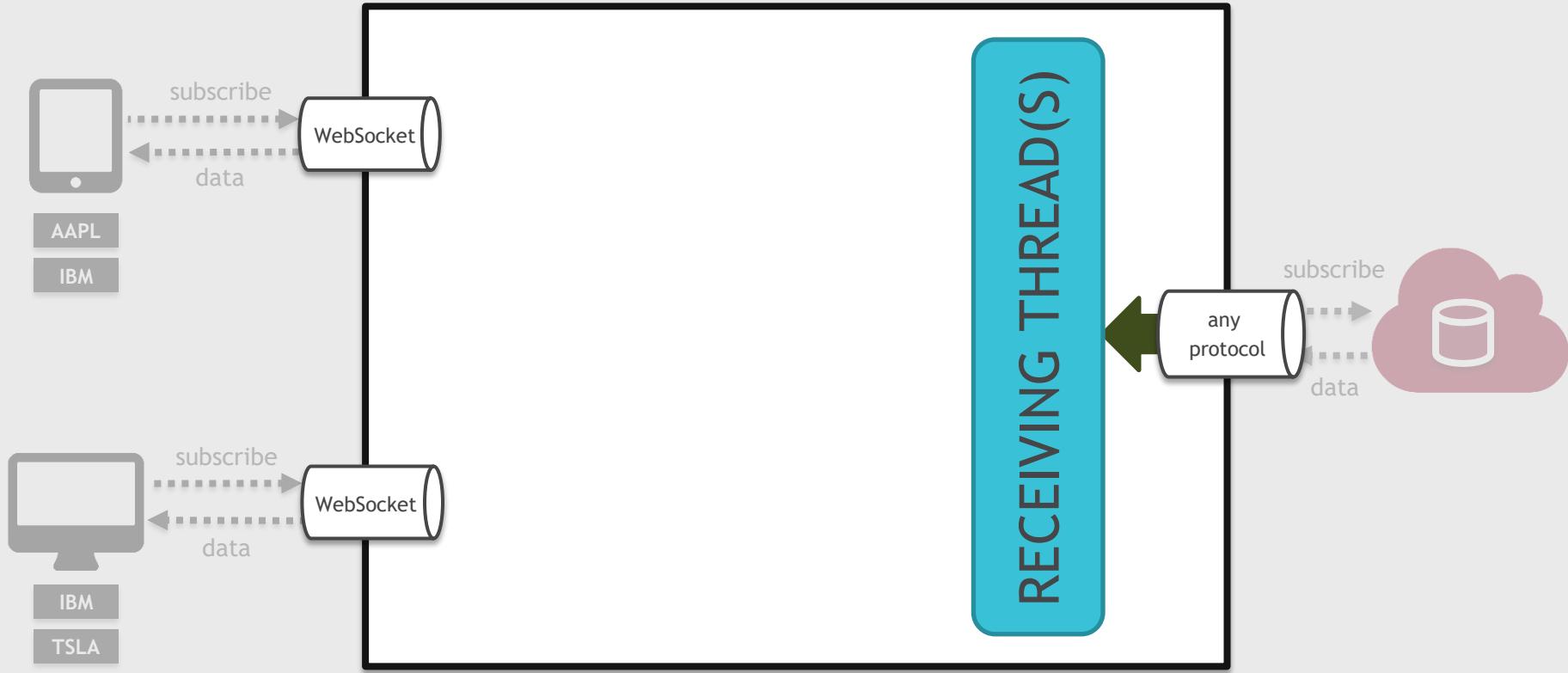


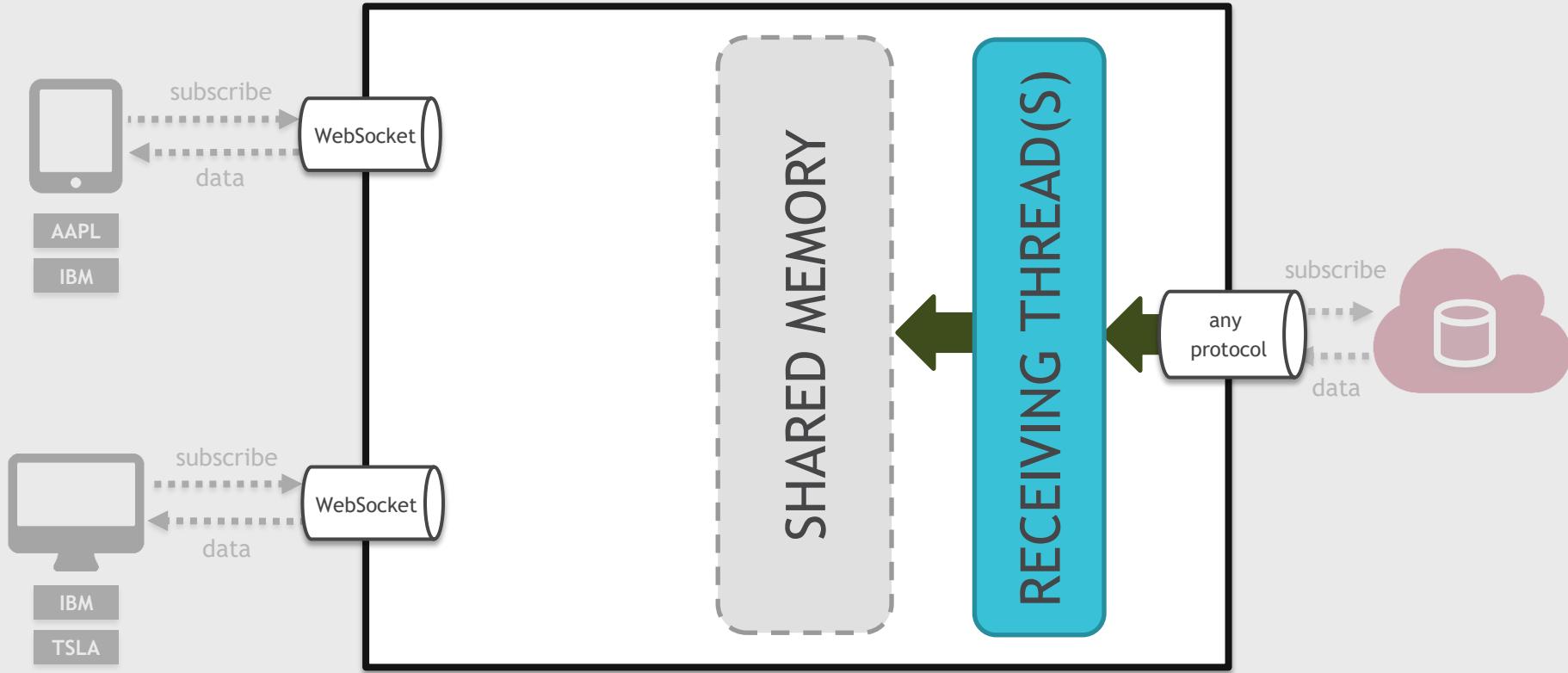
$$\frac{100 + 200 + 300 + 400 + 500}{5} = 300$$

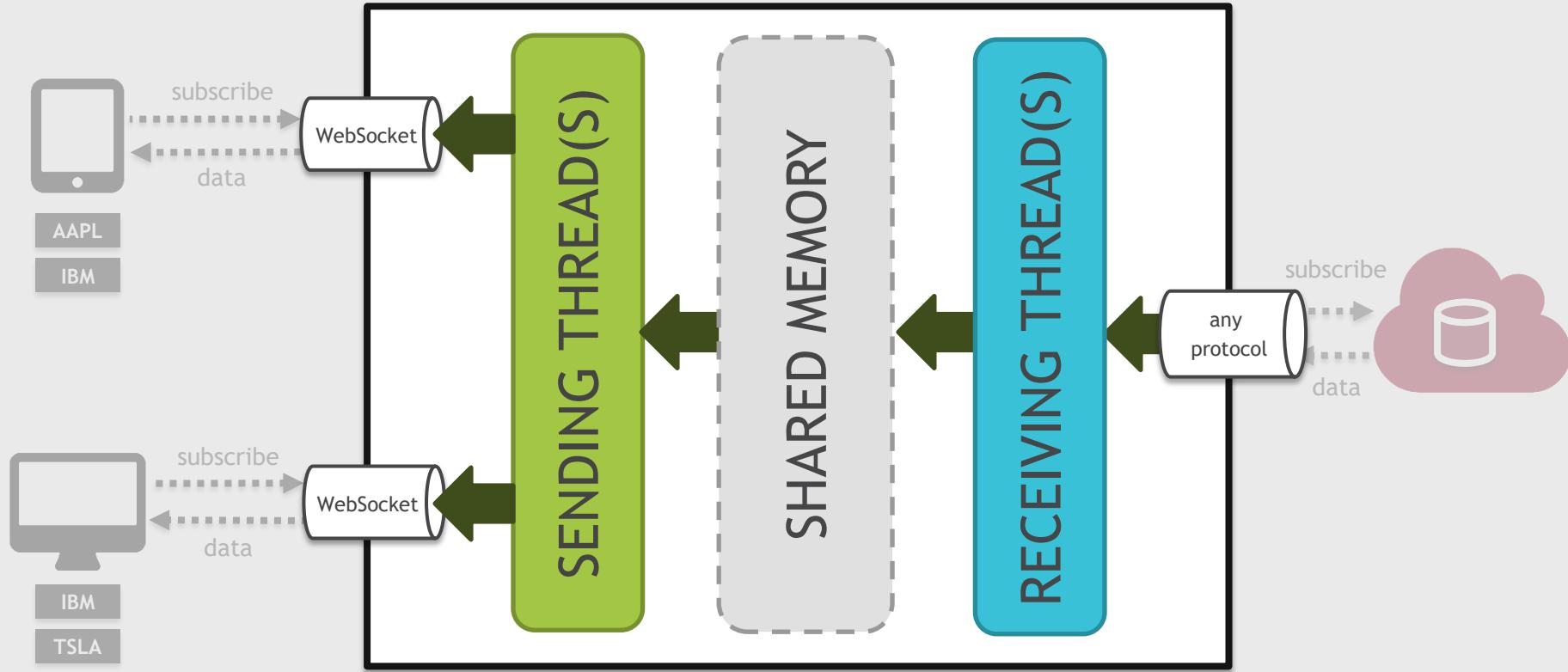
$$\frac{600 + 700 + 800 + 900 + 1000}{5} = 800$$

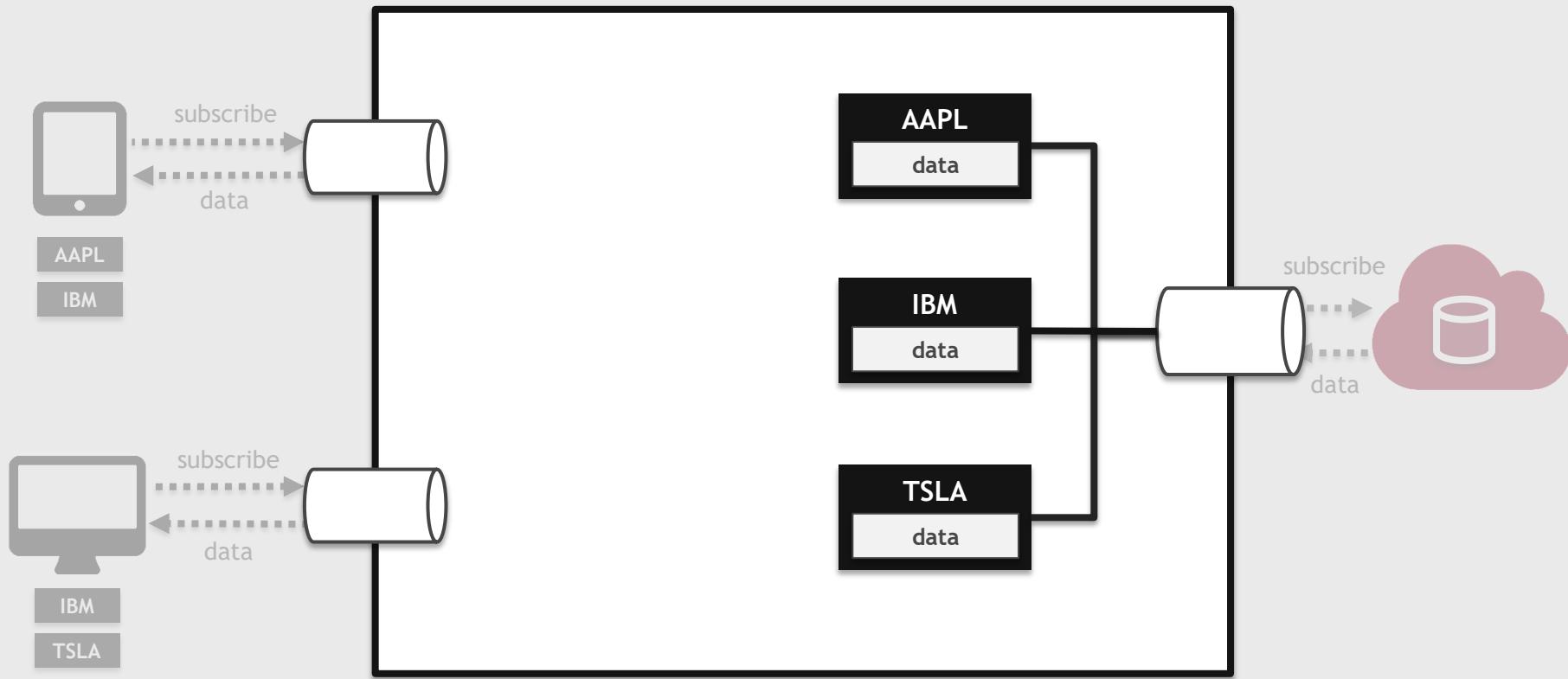


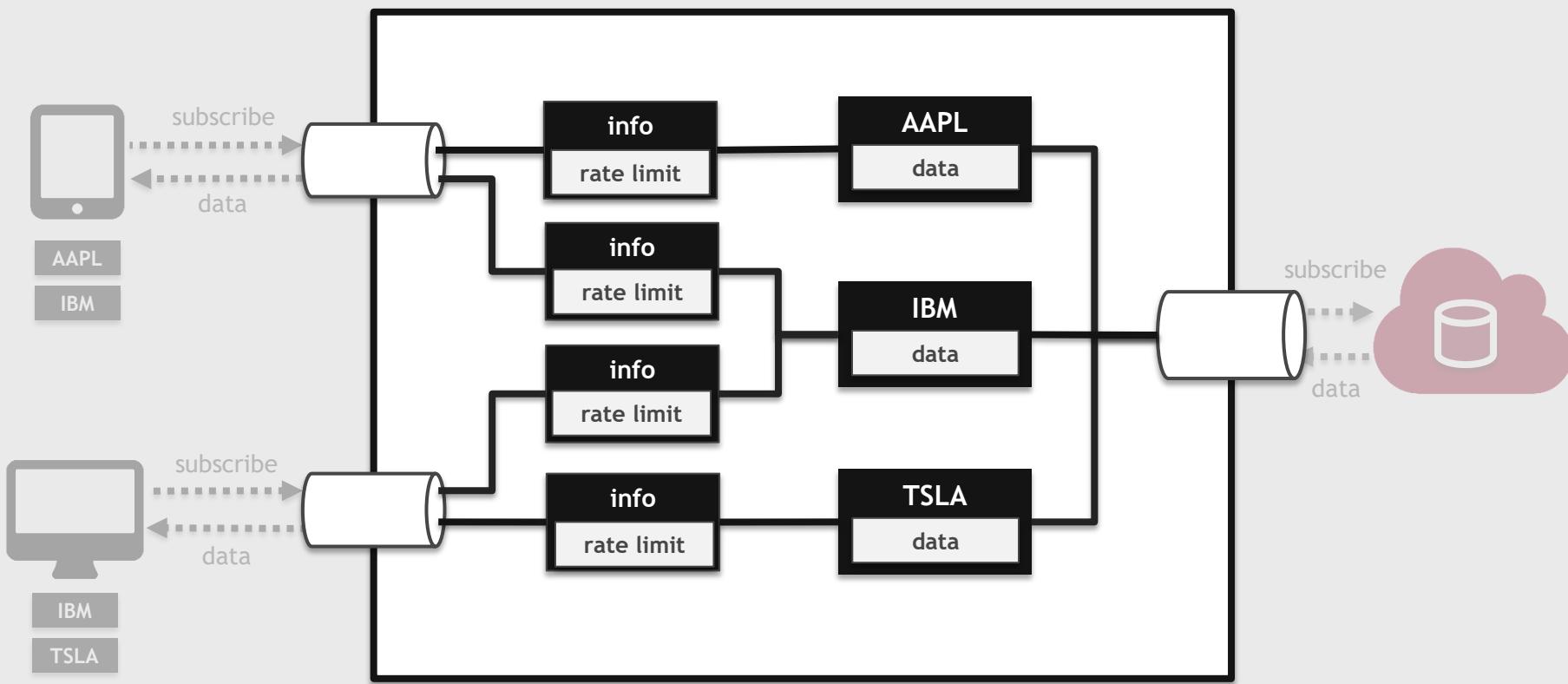
2 updates/s (average)









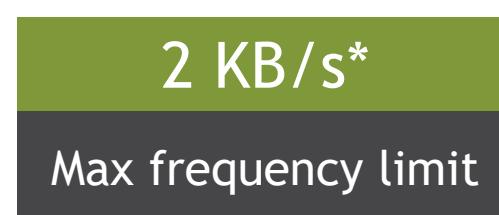
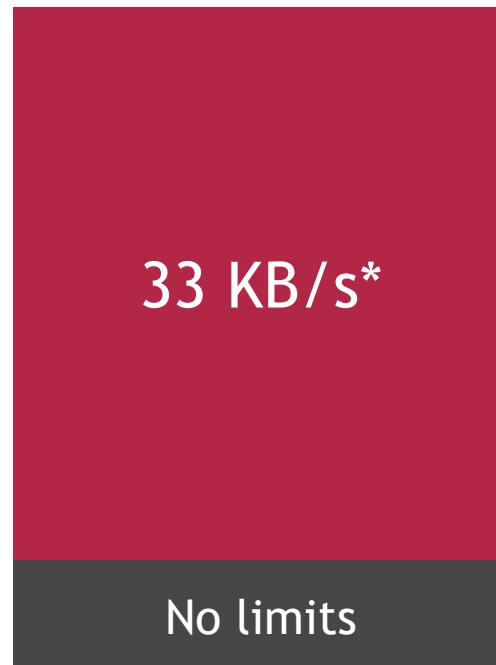


A close-up photograph of a person's hands typing on a laptop keyboard. The person is wearing a green and white checkered shirt. A pair of glasses rests on the laptop screen. The background is dark.

LIVE DEMO

# WEBSOCKET FREQUENCY LIMIT

# MAX FREQUENCY LIMIT



\* for demo application including gzip compression

# WHAT ELSE, CAP?



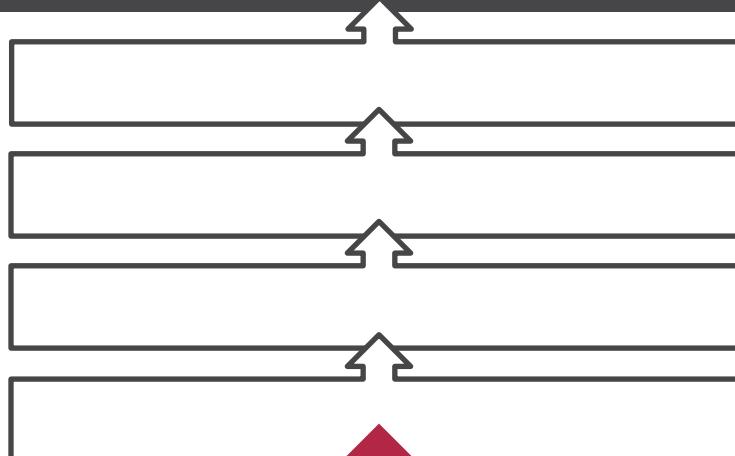
# WHAT ELSE, CAP?



## SEND EVEN LESS DATA!



# High performance WebSocket backend



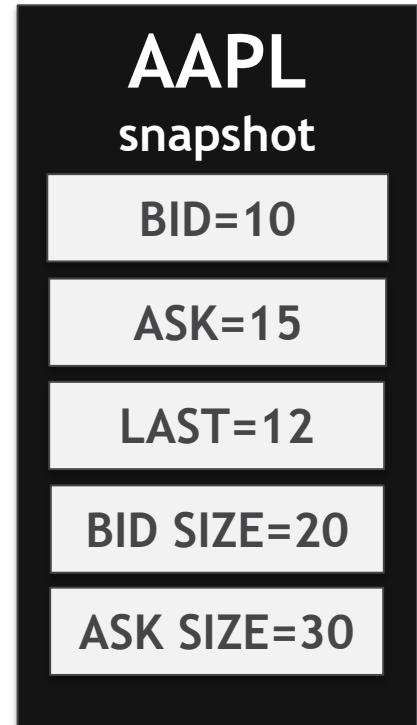
Schema

Max frequency limit

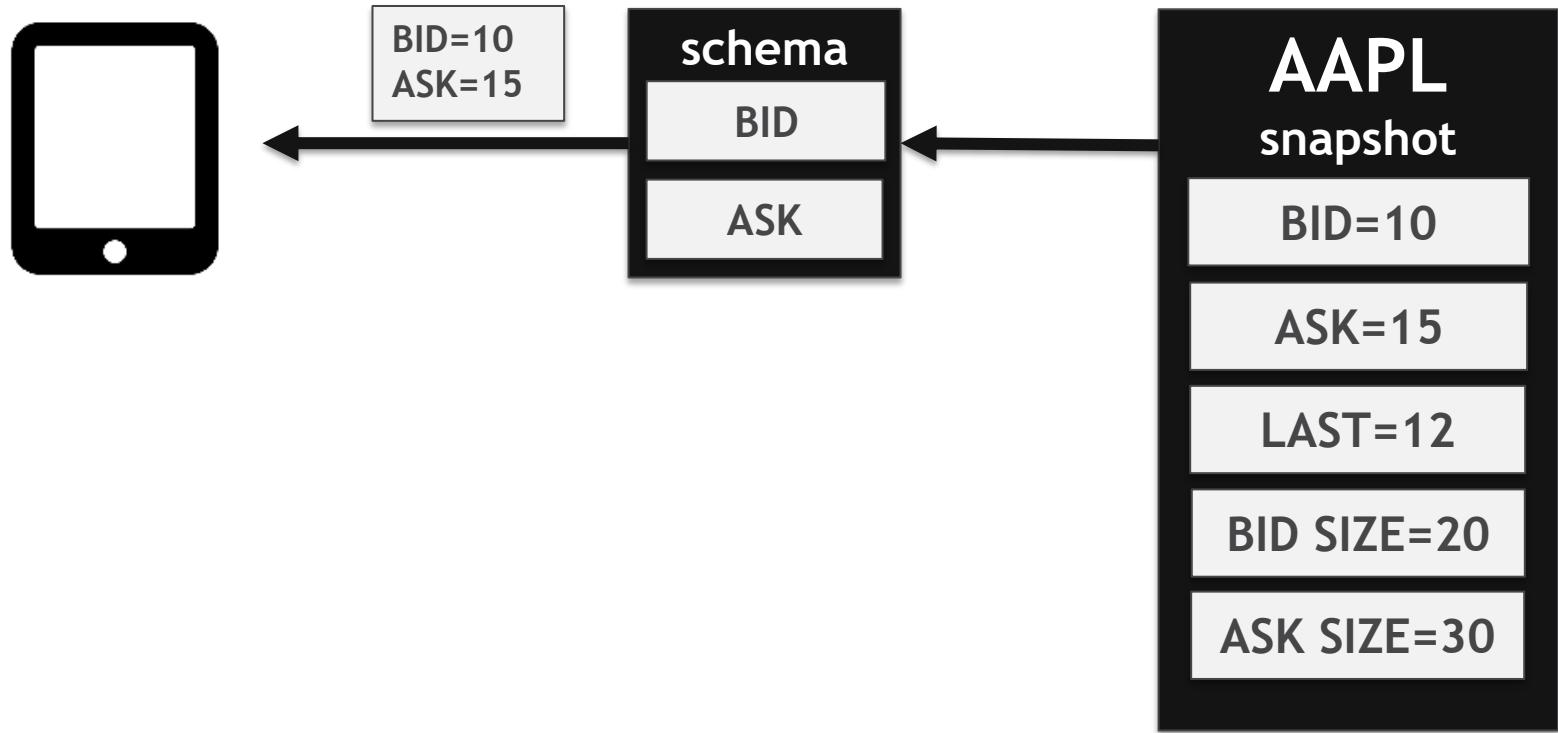


## Basic WebSocket backend

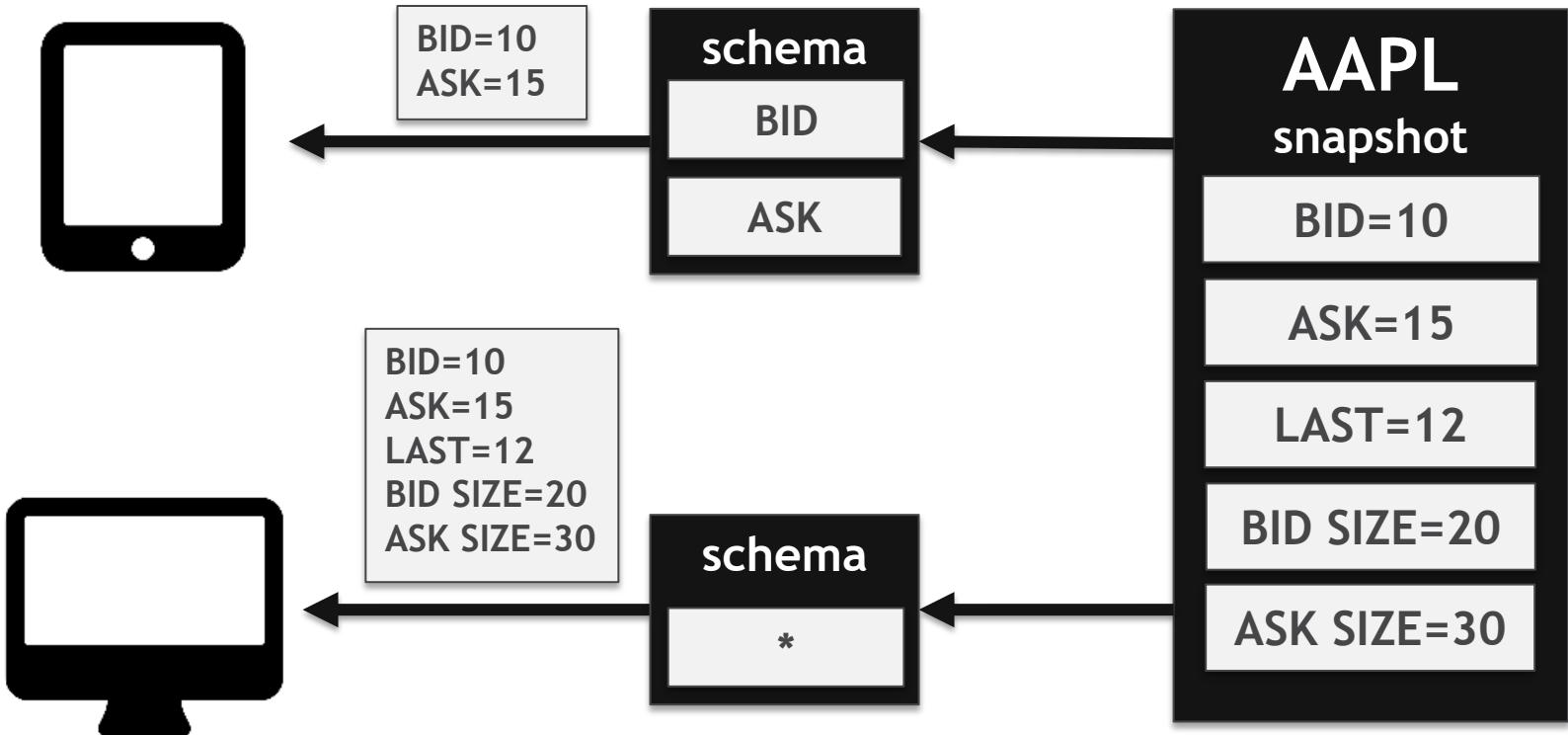
# SCHEMA



# SCHEMA

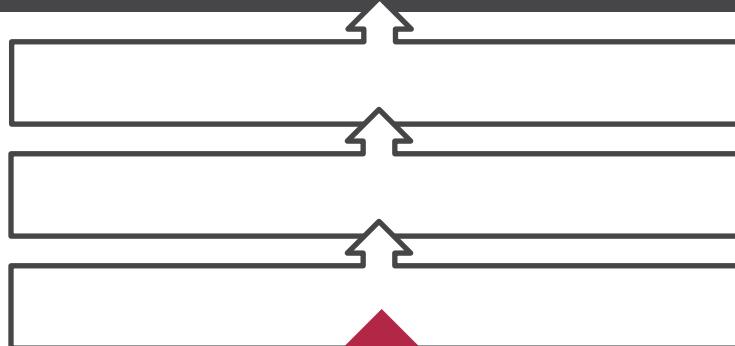


# SCHEMA





# High performance WebSocket backend



Delta delivery

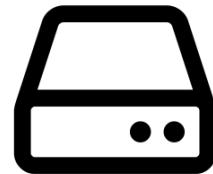
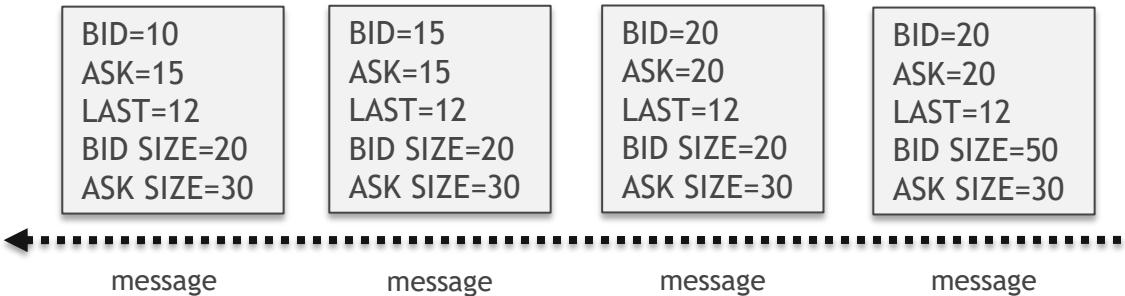
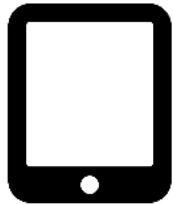
Schema

Max frequency limit

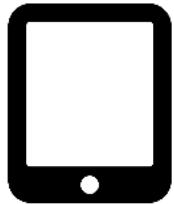


## Basic WebSocket backend

# DELTA DELIVERY



# DELTA DELIVERY



BID=10  
ASK=15  
LAST=12  
BID SIZE=20  
ASK SIZE=30

BID=15  
ASK=15  
LAST=12  
BID SIZE=20  
ASK SIZE=30

BID=20  
ASK=20  
LAST=12  
BID SIZE=20  
ASK SIZE=30

BID=20  
ASK=20  
LAST=12  
BID SIZE=50  
ASK SIZE=30



message

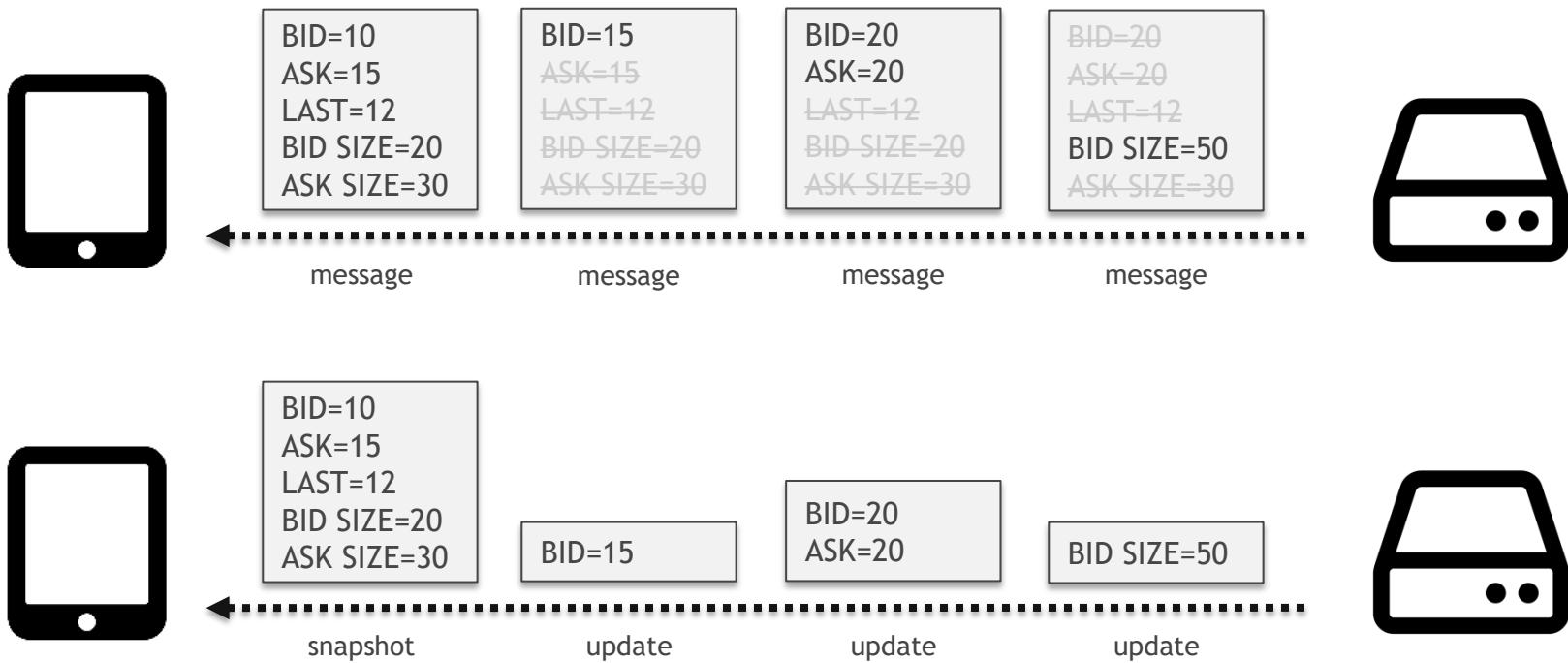
message

message

message

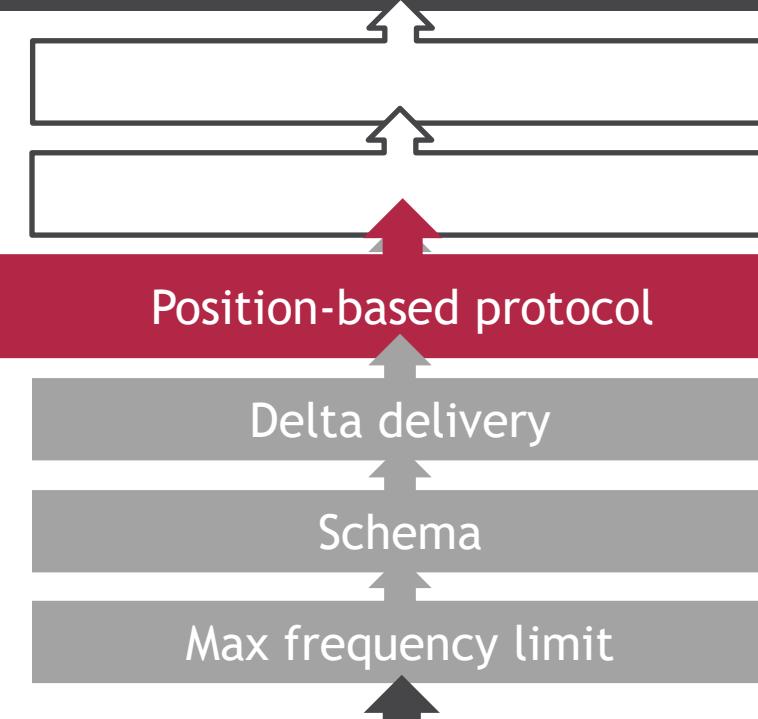


# DELTA DELIVERY





# High performance WebSocket backend



Basic WebSocket backend

```
{  
    symbol: AAPL,  
    bid: 10,  
    ask: 12,  
    last: 11,  
    bidSize: 100,  
    askSize: 200  
}
```

? bytes

```
AAPL|10|12|11|100|200
```

? bytes

```
{  
    symbol: AAPL,  
    bid: 10,  
    ask: 12,  
    last: 11,  
    bidSize: 100,  
    askSize: 200  
}
```

85 bytes

```
AAPL|10|12|11|100|200
```

21 bytes

```
{  
    symbol: AAPL,  
    bid: 10,  
    ask: 12,  
    last: 11,  
    bidSize: 100,  
    askSize: 200  
}
```

46 bytes

```
AAPL || 12 || 200
```

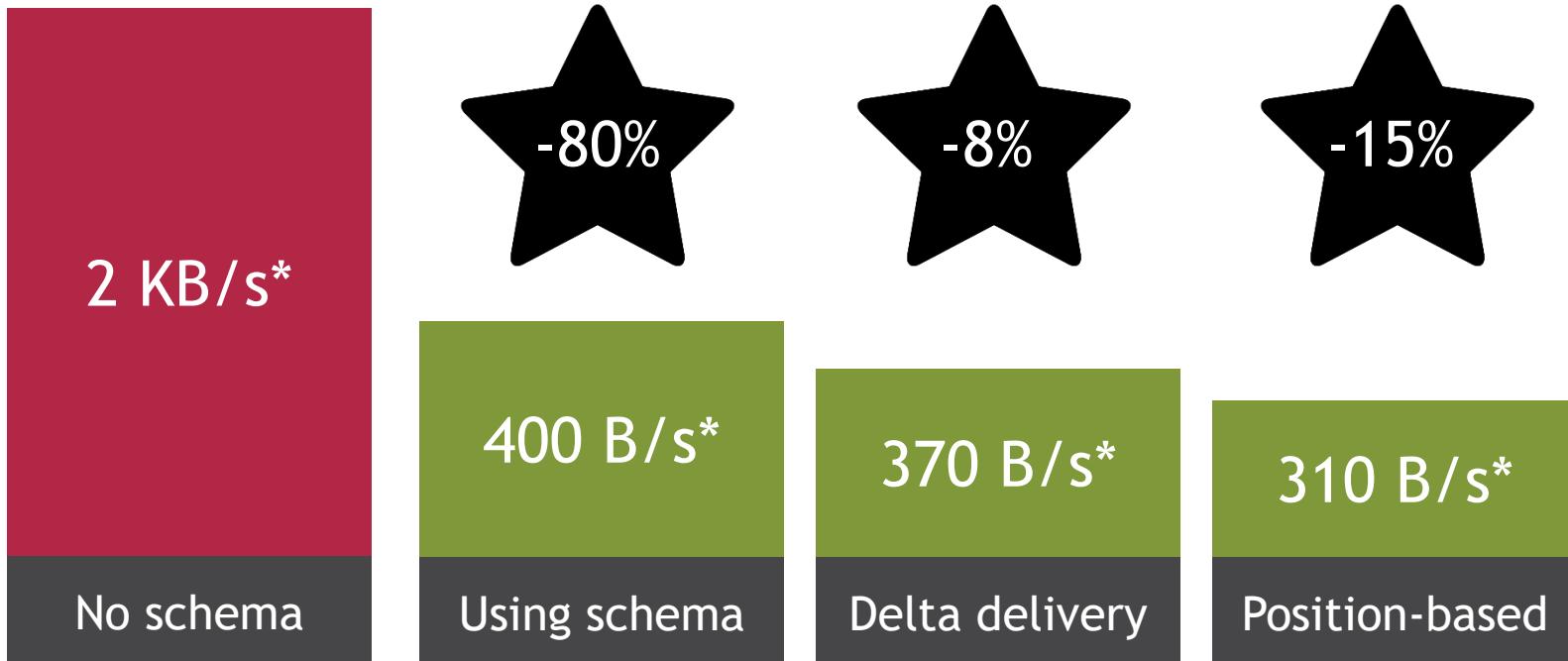
14 bytes

A close-up photograph of a person's hands typing on a laptop keyboard. The person is wearing a green and white checkered shirt. A pair of glasses rests on the laptop screen. In the foreground, there is a solid blue rectangular overlay. On the left side of this overlay, the words "LIVE DEMO" are written in white, bold, uppercase letters. On the right side, the words "BANDWIDTH OPTIMIZATION" are written in large, white, bold, uppercase letters.

LIVE DEMO

BANDWIDTH  
OPTIMIZATION

# SCHEMA + DELTA DELIVERY + POSITION-BASED PROTOCOL



\* for demo application including gzip compression

A scene from Star Trek: The Next Generation. Captain Jean-Luc Picard, wearing his red uniform with a black collar, stands in the center of the bridge. He has his arms outstretched wide and is smiling warmly at the camera. In the background, Lieutenant Worf (in a yellow uniform) and Counselor Deanna Troi (in a purple dress) are seated in their respective chairs. The bridge is filled with various control panels and monitors.

**GOOD JOB!**



# High performance WebSocket backend



Dynamic bandwidth control



Position-based protocol



Delta delivery



Schema

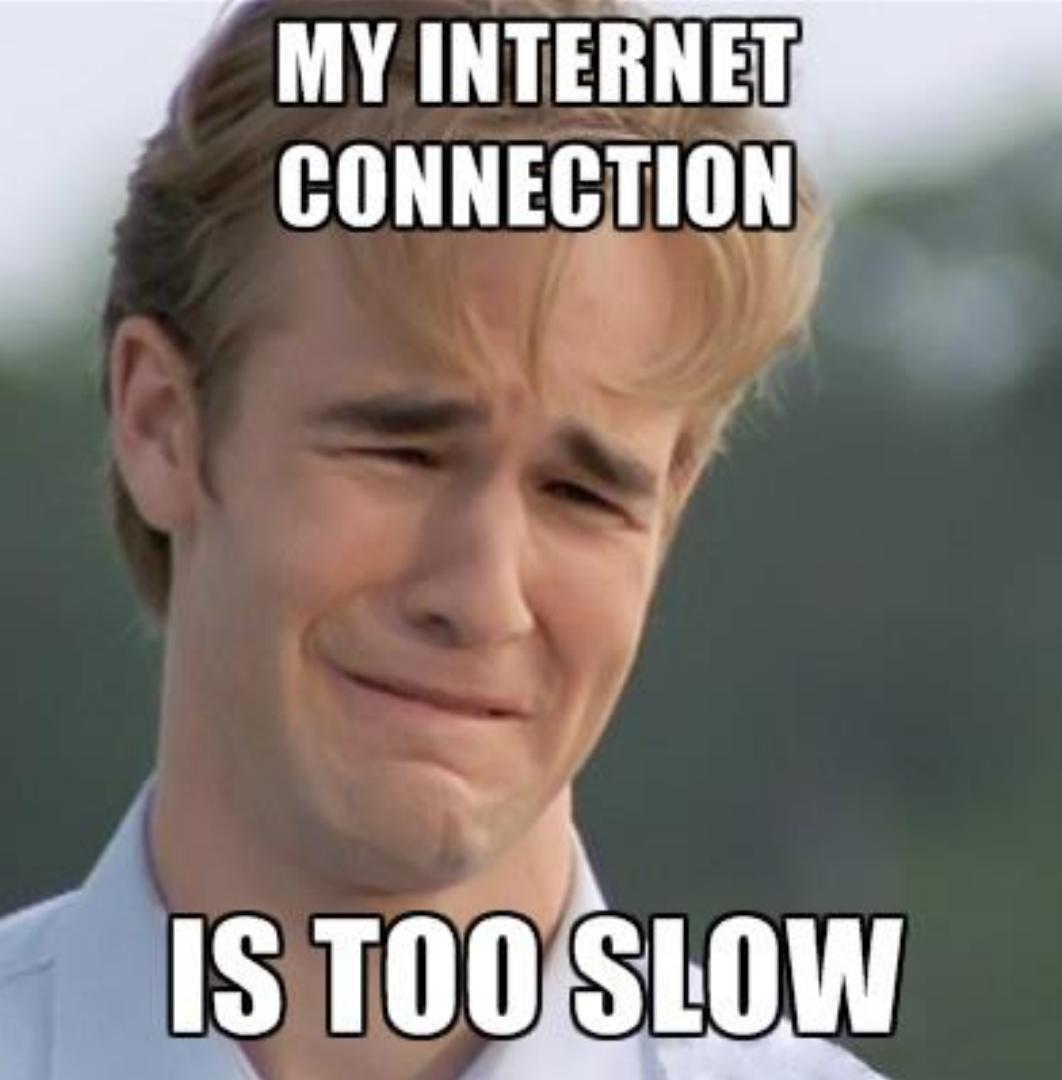


Max frequency limit



## Basic WebSocket backend





MY INTERNET  
CONNECTION

IS TOO SLOW

# DYNAMIC BANDWIDTH CONTROL



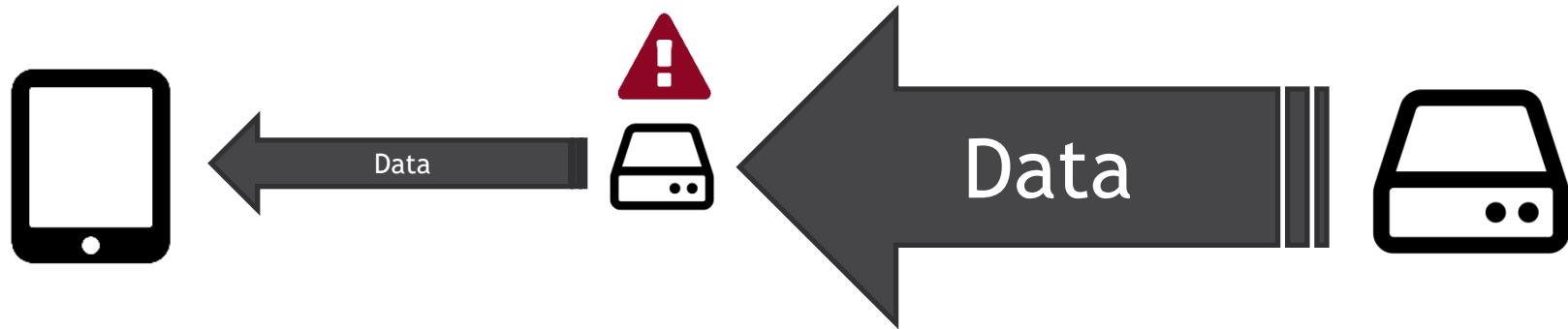
# DYNAMIC BANDWIDTH CONTROL



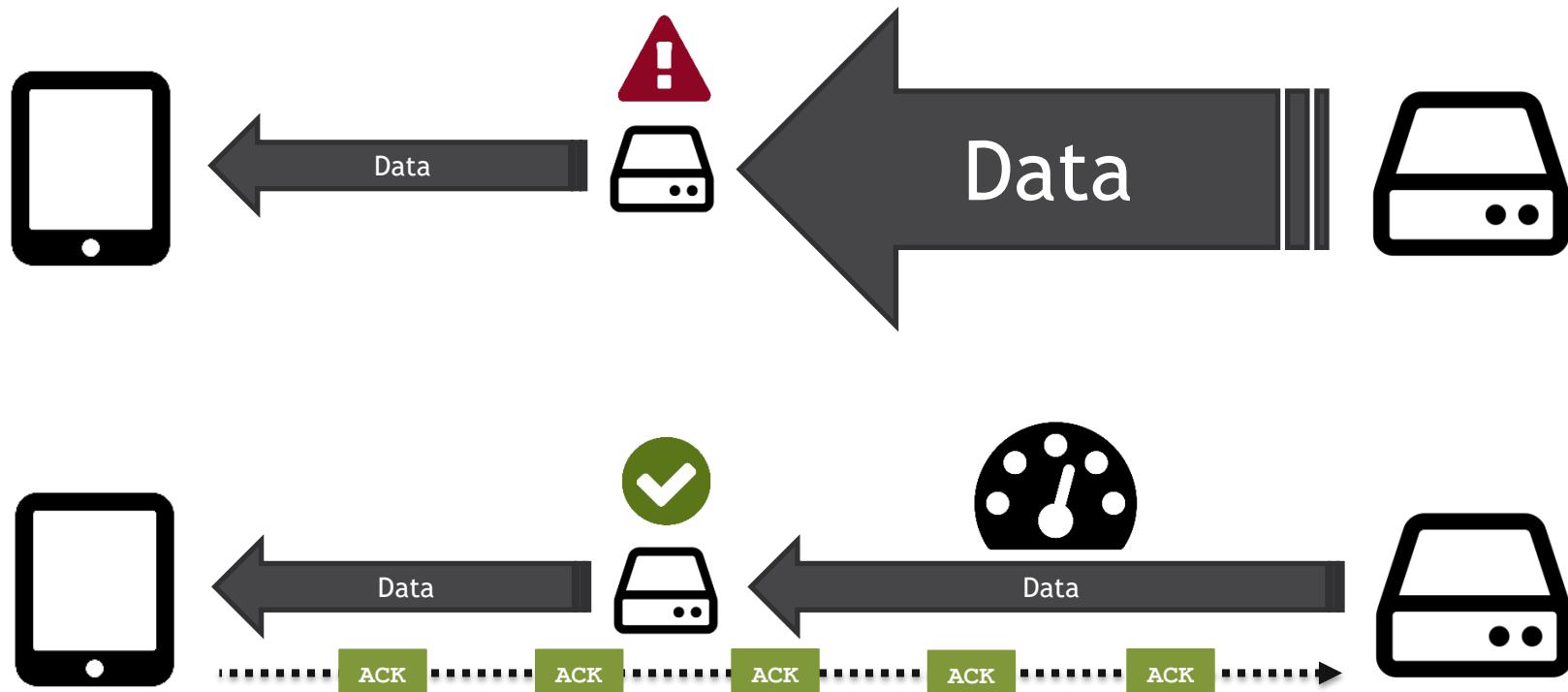
# DYNAMIC BANDWIDTH CONTROL



# DYNAMIC BANDWIDTH CONTROL



# DYNAMIC BANDWIDTH CONTROL

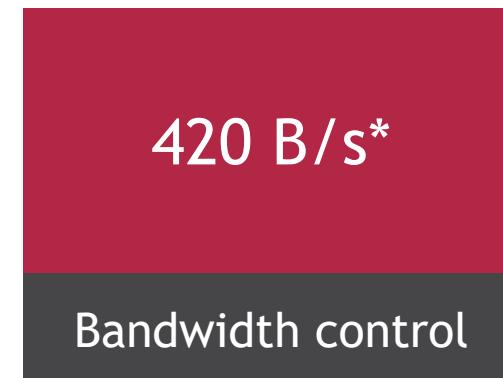
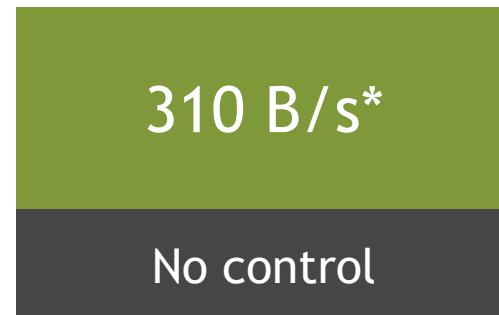


A close-up photograph of a person's hands typing on a laptop keyboard. The person is wearing a green and white checkered shirt. A pair of glasses rests on the laptop screen. The background is dark.

LIVE DEMO

# DYNAMIC BANDWIDTH CONTROL

# DYNAMIC BANDWIDTH CONTROL



\* for demo application including gzip compression



# High performance WebSocket backend

Bandwidth limit per user

Dynamic bandwidth control

Position-based protocol

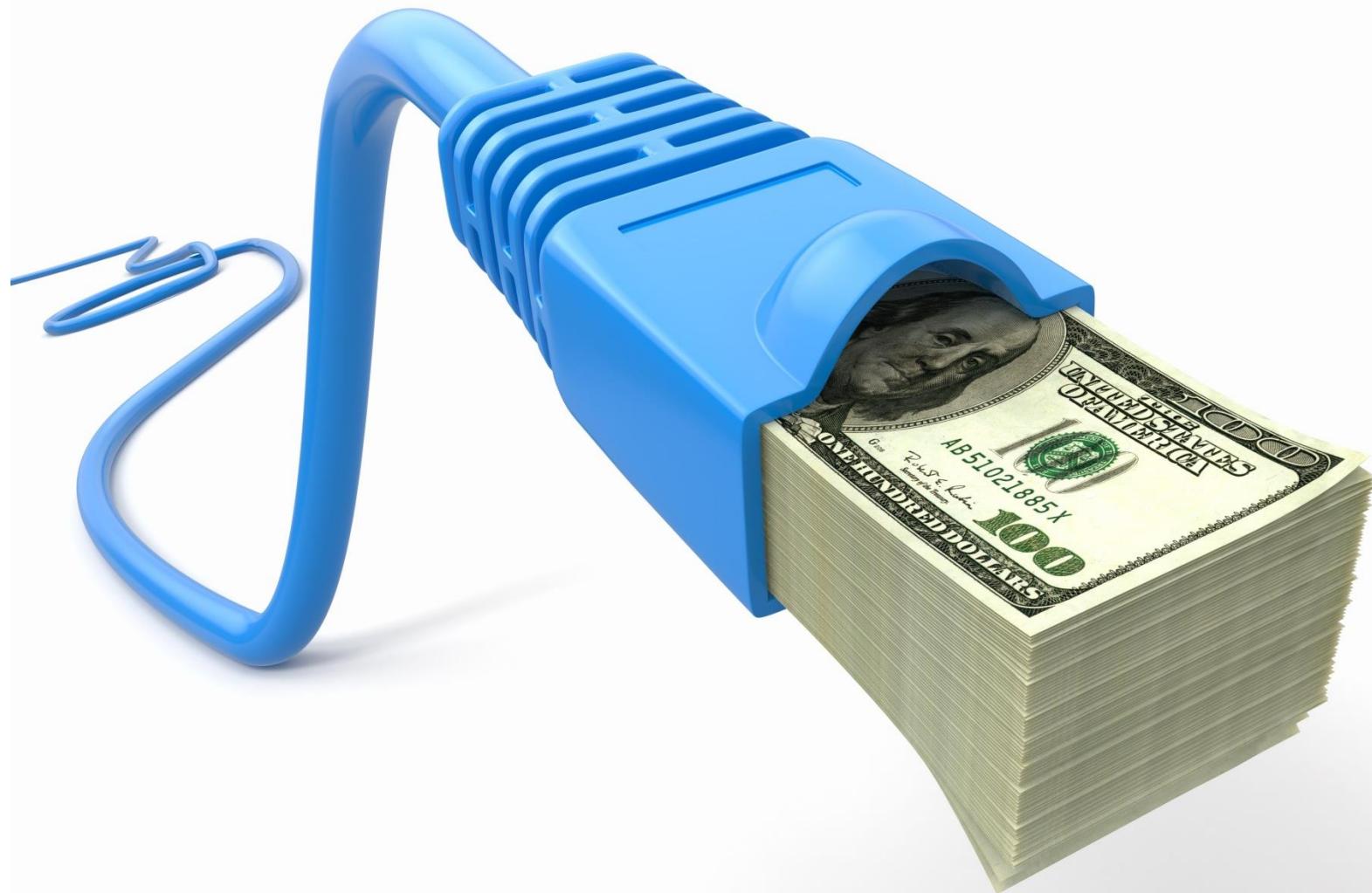
Delta delivery

Schema

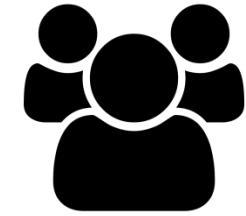
Max frequency limit



## Basic WebSocket backend



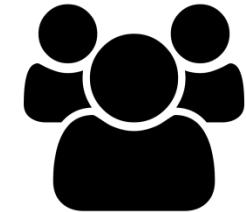
# BANDWIDTH LIMIT PER USER



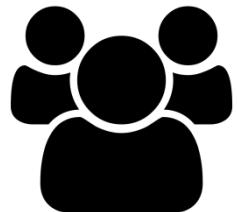
paying users



# BANDWIDTH LIMIT PER USER



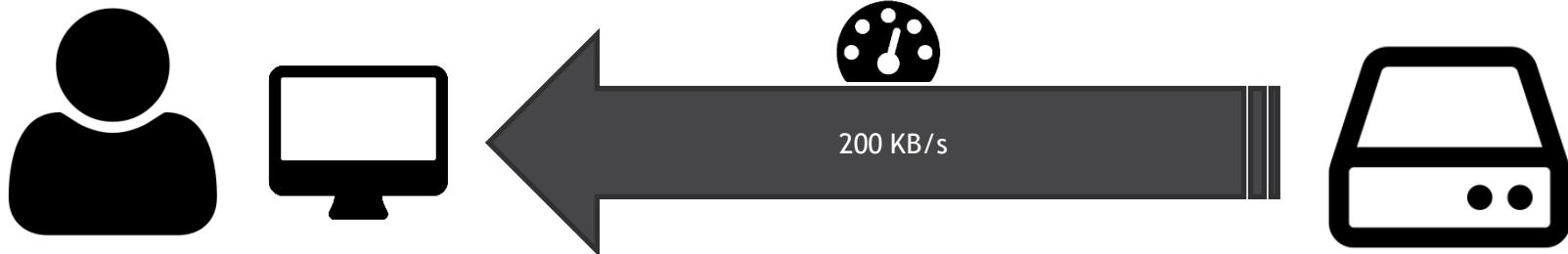
paying users



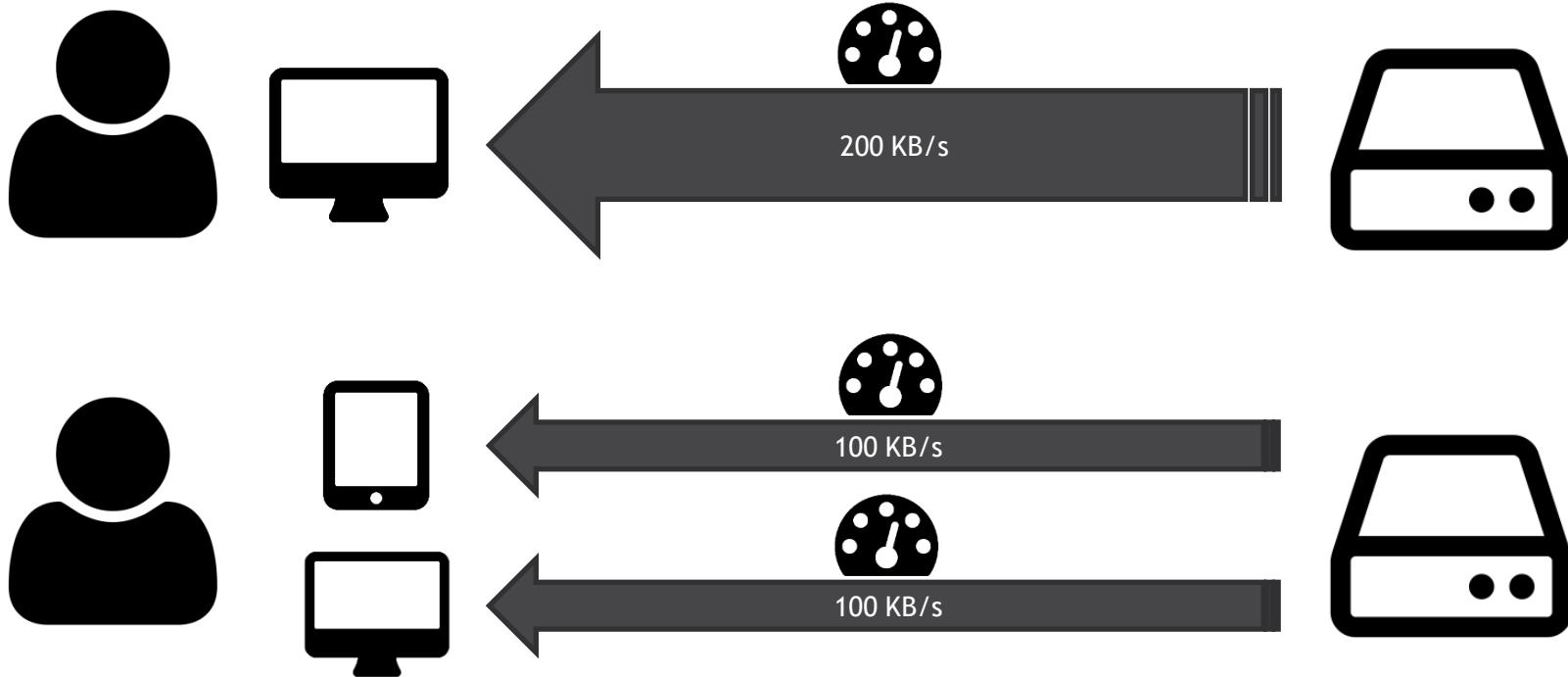
free users



# BANDWIDTH LIMIT PER USER



# BANDWIDTH LIMIT PER USER



A close-up photograph of a person's hands typing on a laptop keyboard. The laptop is open, showing its screen and keyboard. A white computer mouse sits to the right of the keyboard. The person is wearing a green and white checkered shirt. In the top left corner of the image, there is a small white rectangular button with the text "LIVE DEMO" in black capital letters.

LIVE DEMO

# BANDWIDTH LIMIT PER USER

**ARE WE READY  
FOR PRODUCTION?**



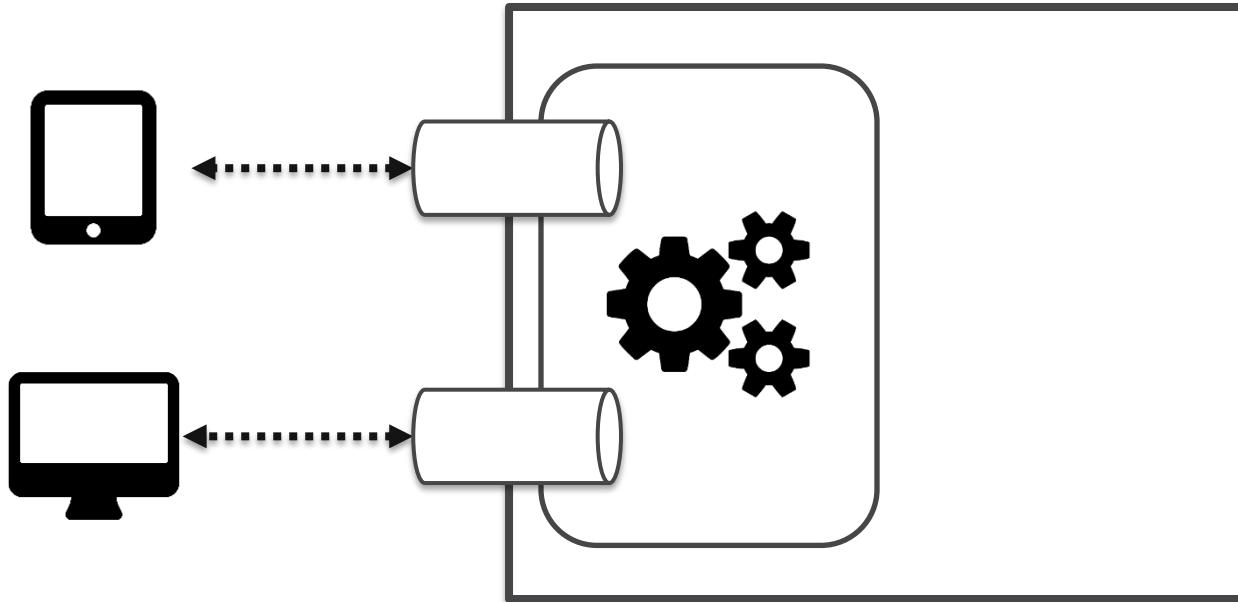


*POWERED BY*

<http://lightstreamer.com>

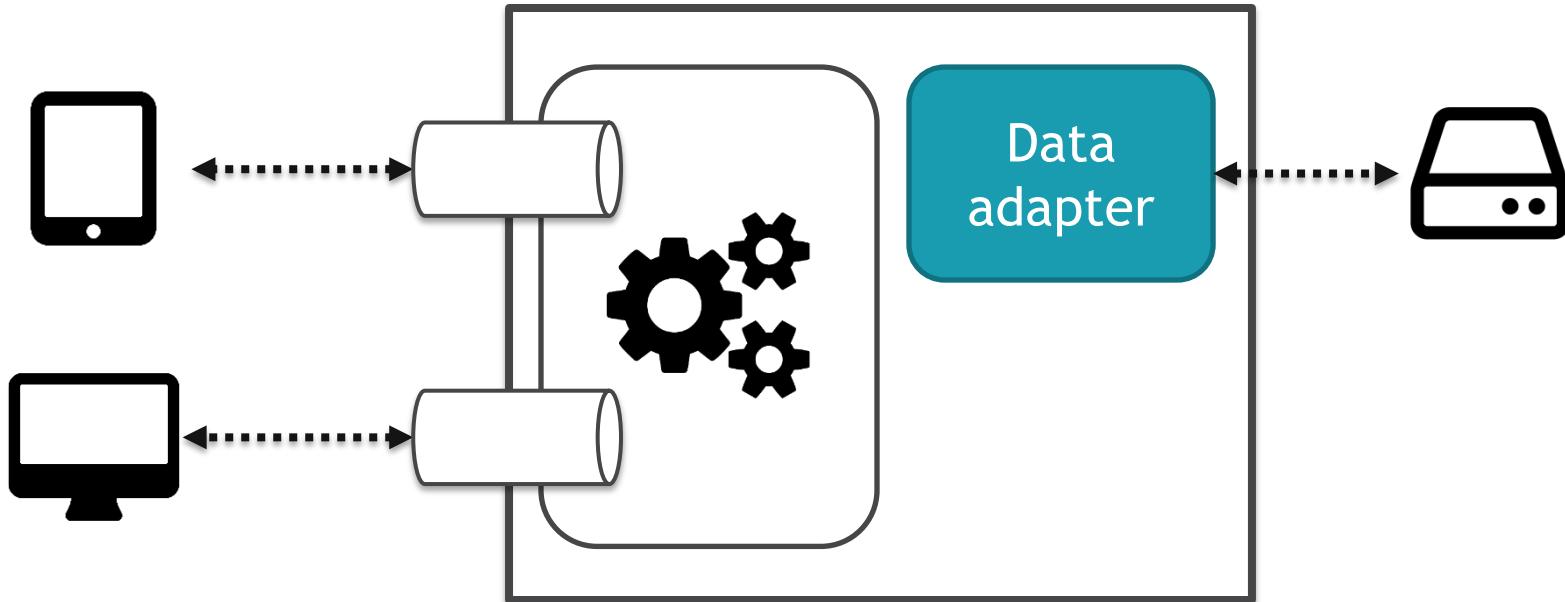
# LIGHTSTREAMER SERVER

POWERED BY

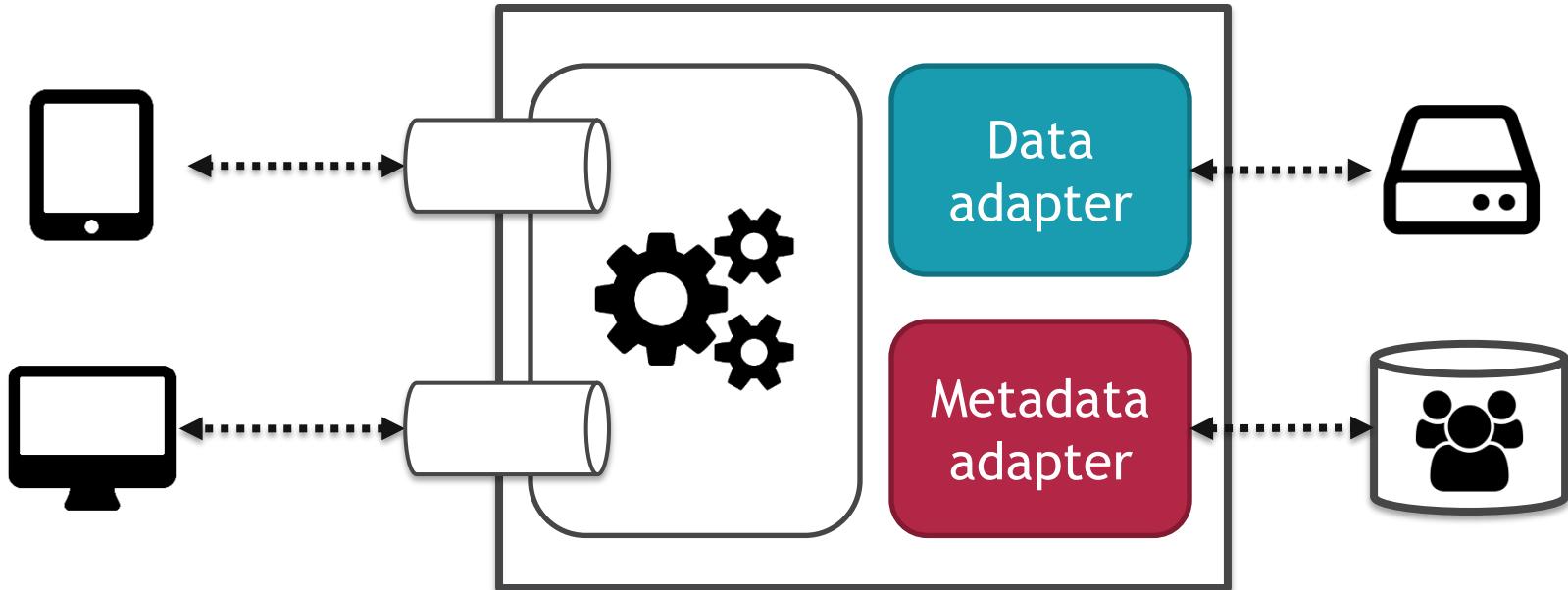


# LIGHTSTREAMER SERVER

POWERED BY

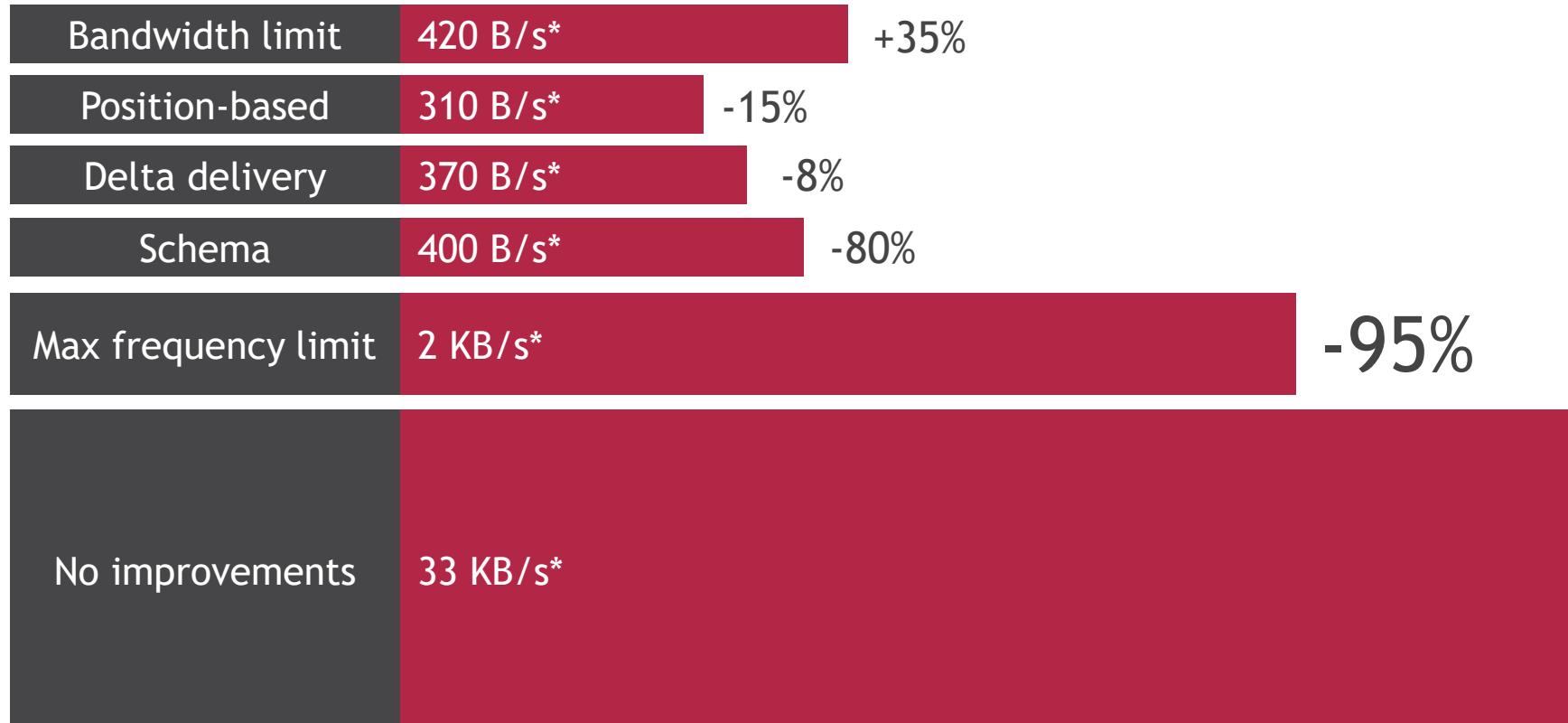


# LIGHTSTREAMER SERVER

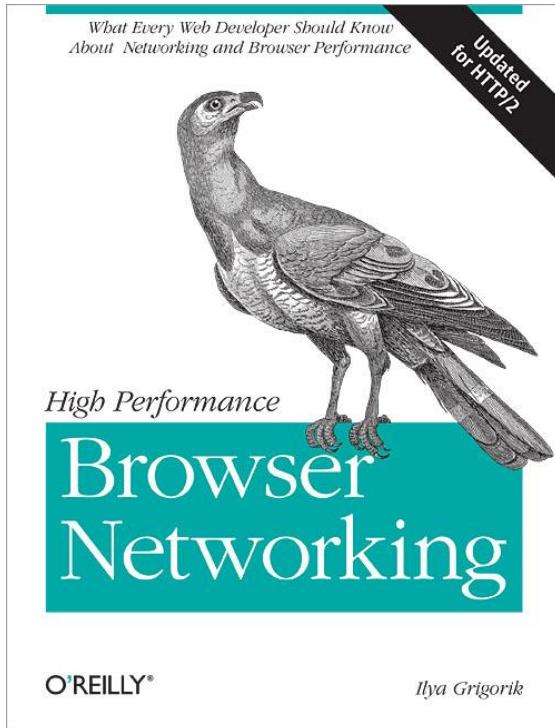


# CONCLUSION

# BANDWIDTH OPTIMIZATION\*



\* for demo application including gzip compression



## High Performance Browser Networking

What every web developer should know about networking and web performance

Free reading: <https://hpbn.co/>

By Ilya Grigorik

Publisher: O'Reilly Media

Release Date: September 2013

POWERED BY



<http://lightstreamer.com/docs>

POWERED BY



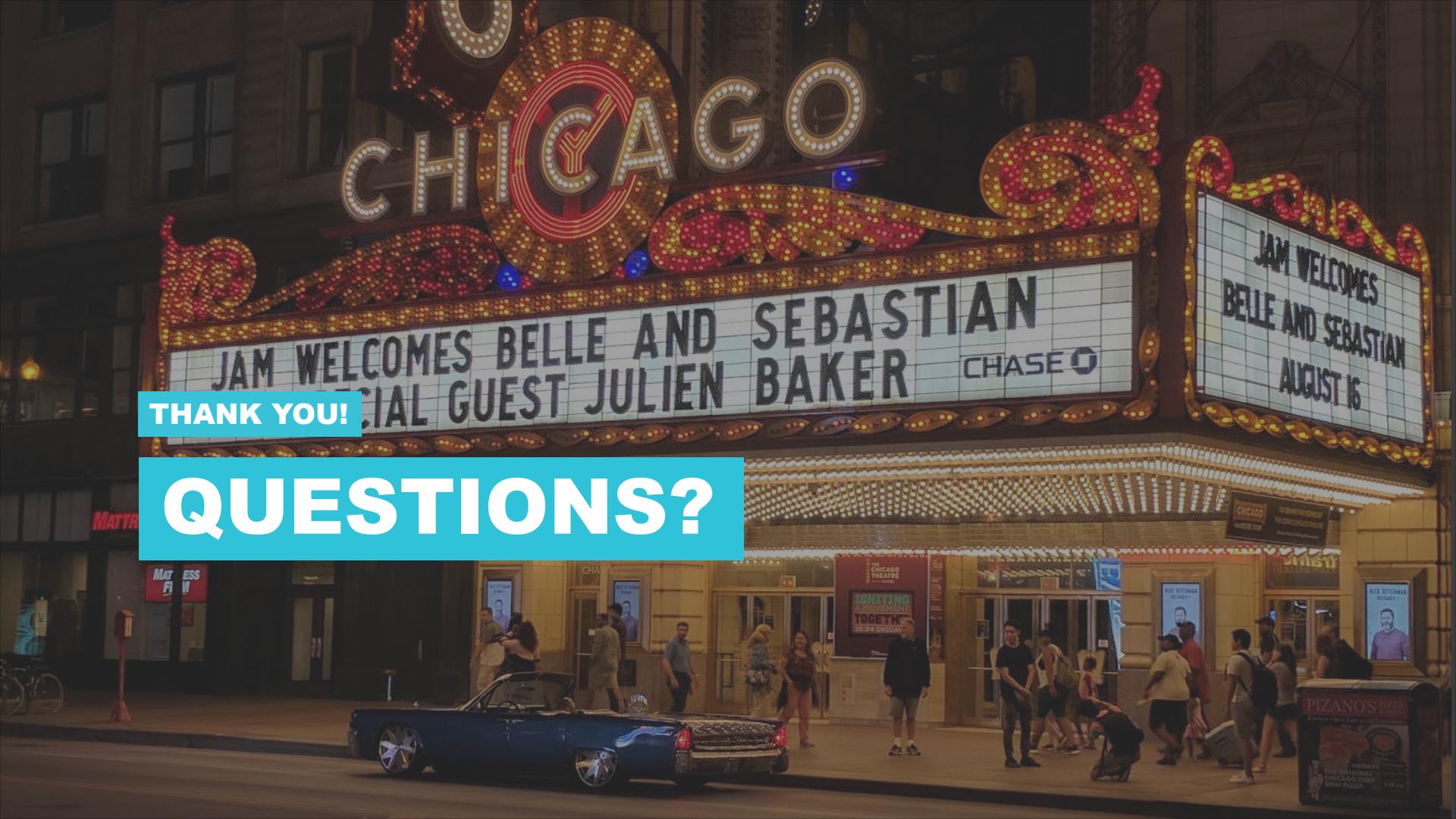
# More Than Just WebSockets for Real-Time Multiplayer Games and Collaboration

On YouTube and Slideshare  
By Alessandro Alinone (CEO of Lightstreamer)

## TAKE AWAYS

---

1. Improve user-experience
2. Reduce bandwidth
3. Handle network failures

The background image shows the famous Chicago Theater marquee at night. The main sign reads "CHICAGO" in large, illuminated letters, with "JAM WELCOMES BELLE AND SEBASTIAN" and "SPECIAL GUEST JULIEN BAKER" below it. To the right, a smaller marquee displays "JAM WELCOMES BELLE AND SEBASTIAN AUGUST 16". A blue rectangular overlay contains the text "THANK YOU!" and "QUESTIONS?".

THANK YOU!

QUESTIONS?



# GitHub



[github.com/kslisenko/streaming](https://github.com/kslisenko/streaming)