DO YOU REALLY NEED WEBSOCKETS

Stanko Krtalic Rusendic

github.com/stankec
@monorkin
hey@stanko.io
stanko.io

01

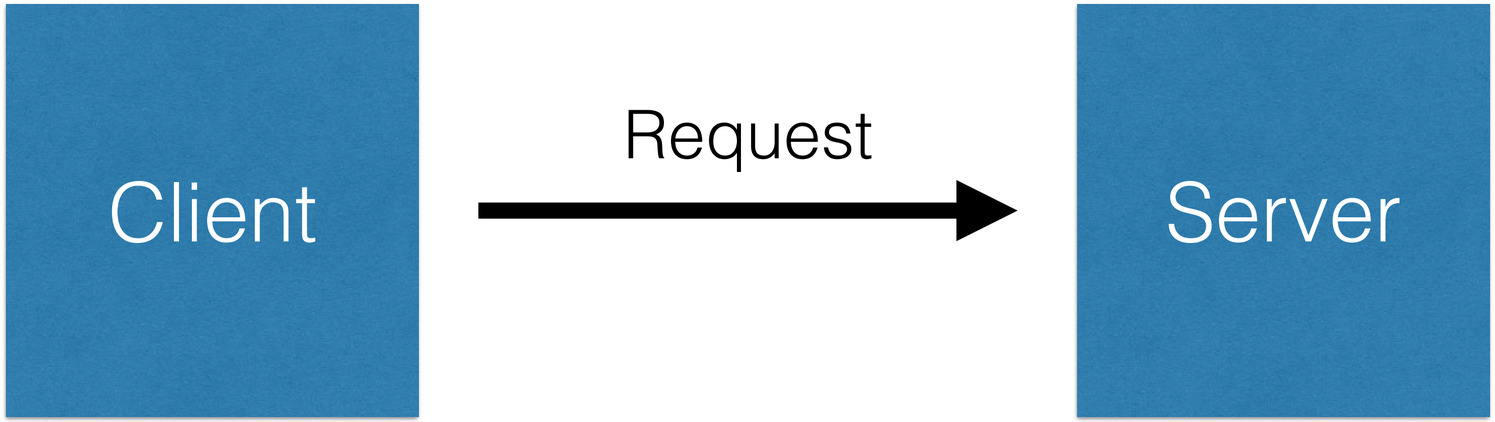
THE REAL TIME WEB

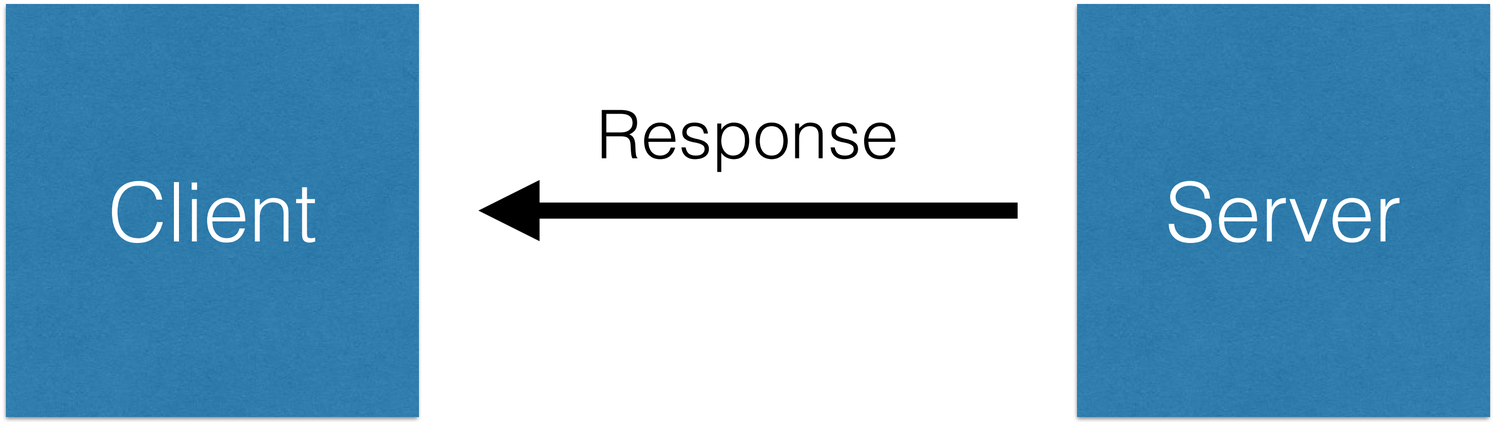
WebSocket are cool

Asynchronous communication

Client

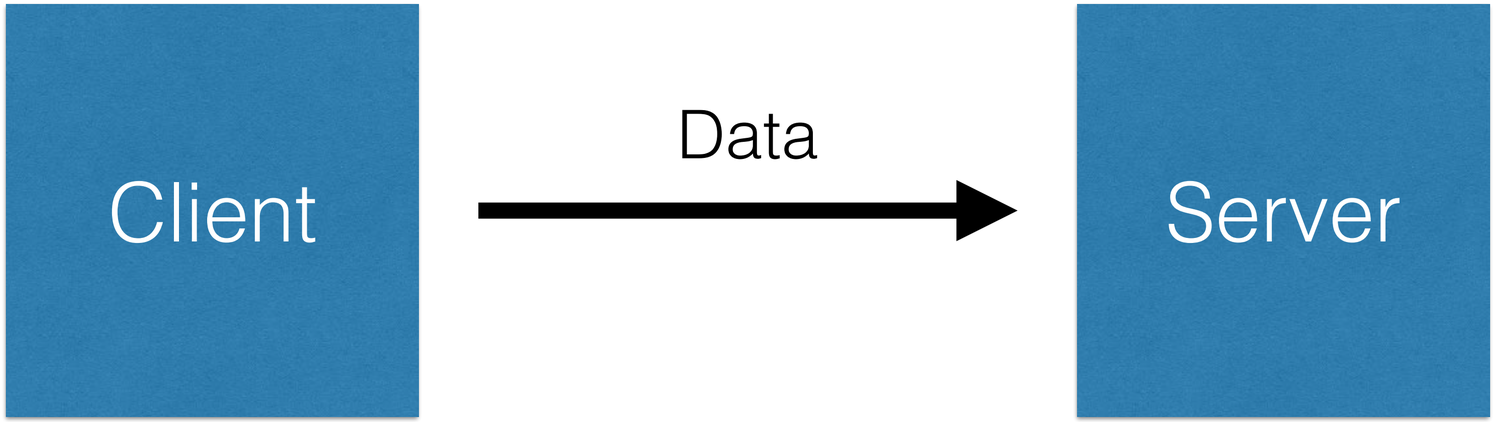
Server


Client
Request
Server


Client
Response
Server

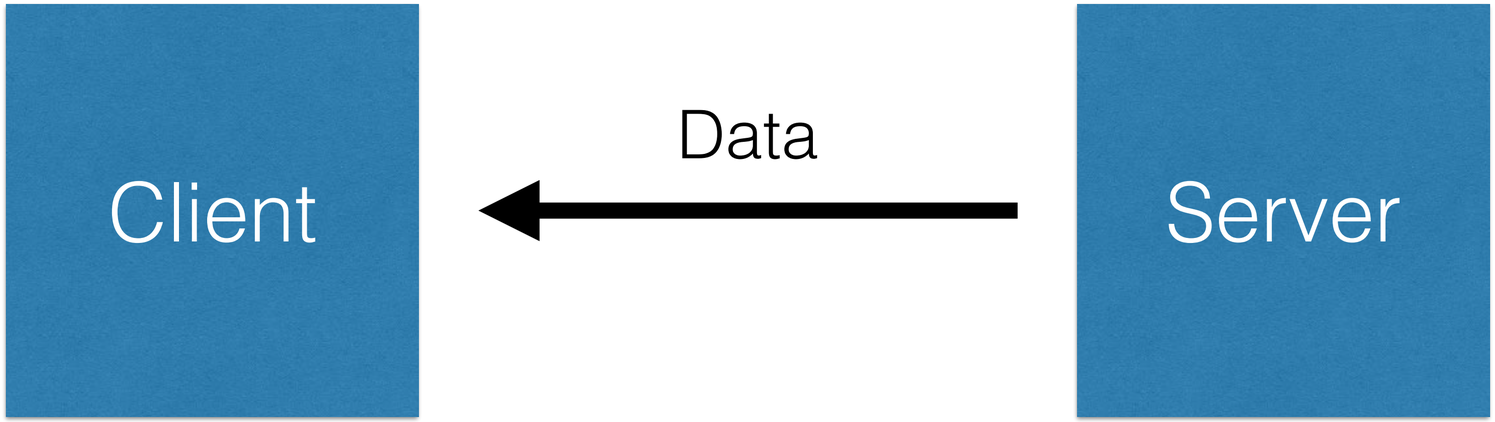
Client

Server


Client
Data
Server

Client

Server


Client
Data
Server

Full-duplex

Realtime client-server communication

Introduced in 2011

RFC 6455

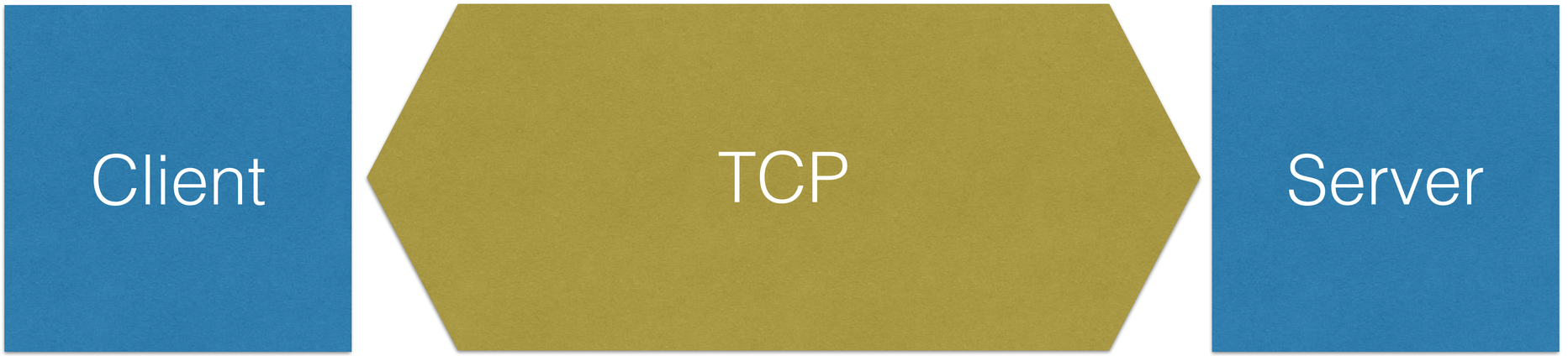
The protocol is…

…interesting

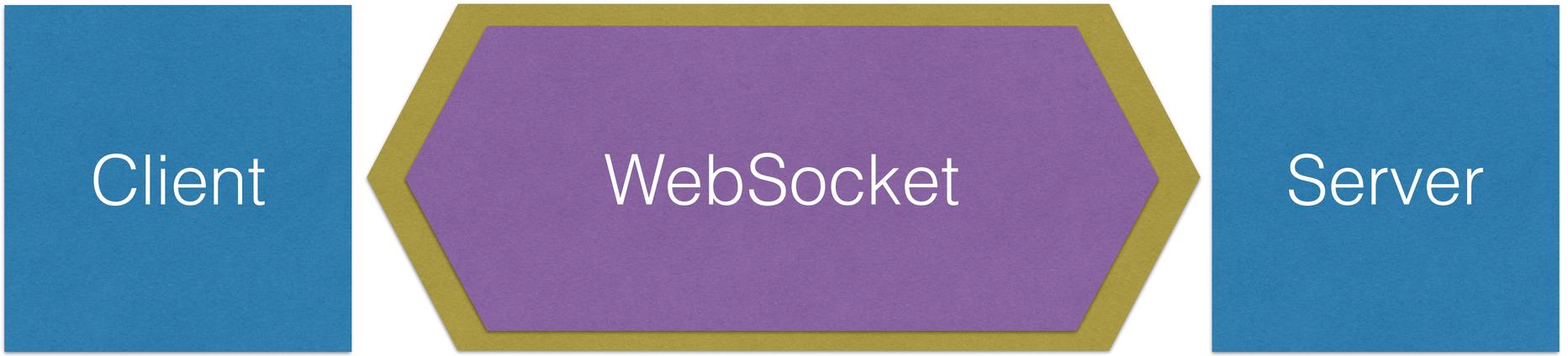
Each WebSocket starts as a HTTP request

But ends up as a WebSocket?


Client
Server


Client
TCP
Server


Client
HTTP Upgrade
Server


Client
WebSocket
Server

Part of HTML5 spec

```
1  // Create WebSocket connection.
2  const socket = new WebSocket('ws://localhost:8080');
3
4  // Connection opened
5  socket.addEventListener('open', function (event) {
6      socket.send('Hello Server!');
7  });
8
9  // Listen for messages
10 socket.addEventListener('message', function (event) {
11     console.log('Message from server ', event.data);
12 });
```

websocket-demo.js hosted with ❤ by GitHub view raw

WebSocket JS API demo from https://developer.mozilla.org/en-US/docs/Web/API/WebSocket

Up to 1024 connections

Non standard proxying/LB

Dropped client detection

No reconnection handling

What are they useful for?

Voice chat

Video chat

Games

Real-time **client to server** communication

02

OVERKILL SOLUTION

Separate controls flow from the HTTP server

More complexity

Alternatives?

03

SERVER SENT EVENTS

Introduced in 2006

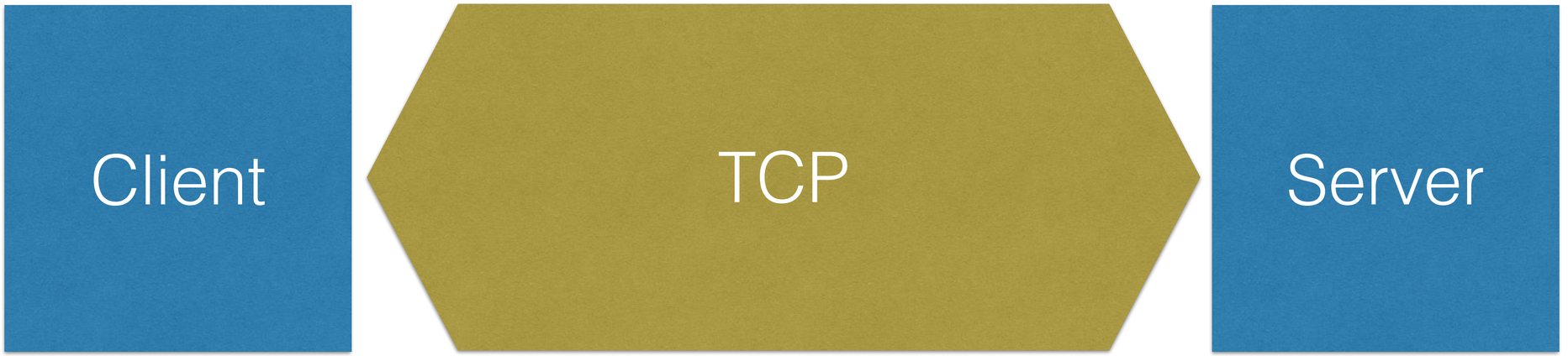
WHATWG Web Applications 1.0

Only Server to Client communication

HTTP based

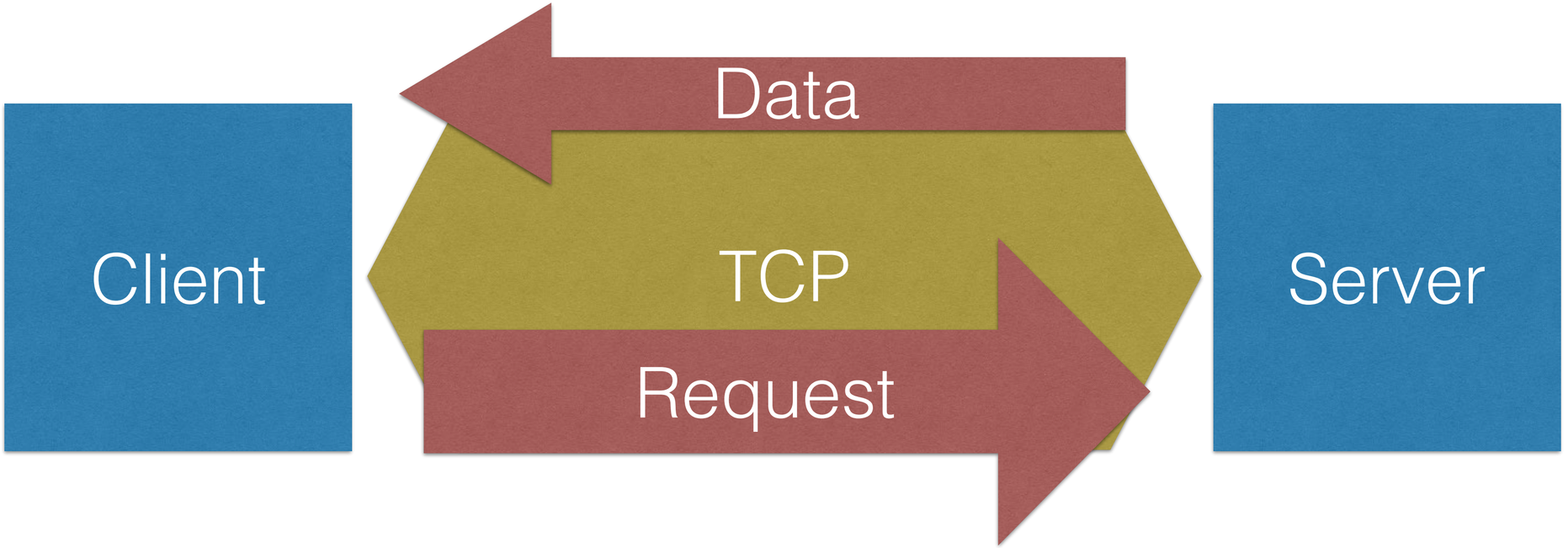
Relies on HTTP streaming

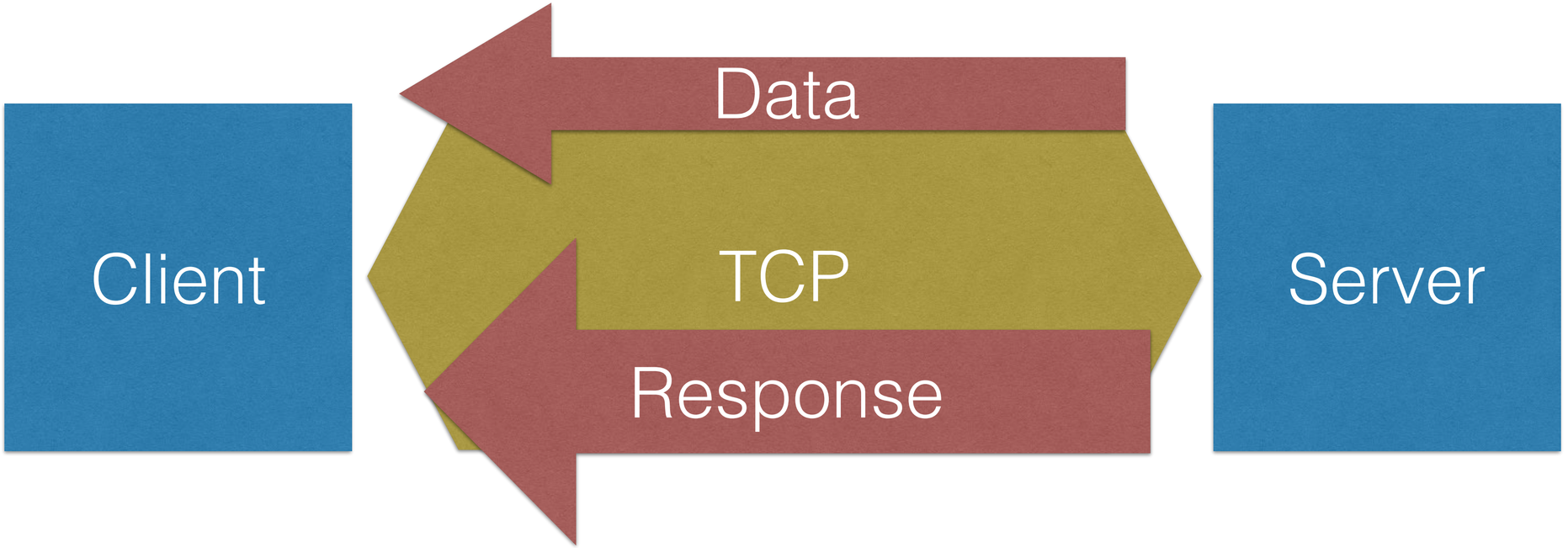

Client
Server


Client
TCP
Server


Client
HTTP Request
Server


Client
Data
Server


Client
Data
TCP
Request
Server


Client
Data
TCP
Response
Server

Not supported on IE / Edge

Subject to the ~6 connection limit

Automatic reconnection handling

No dropped client detection

Benefits?

Same controls flow

Simple protocol

What are they useful for?

Text chat
Notifications

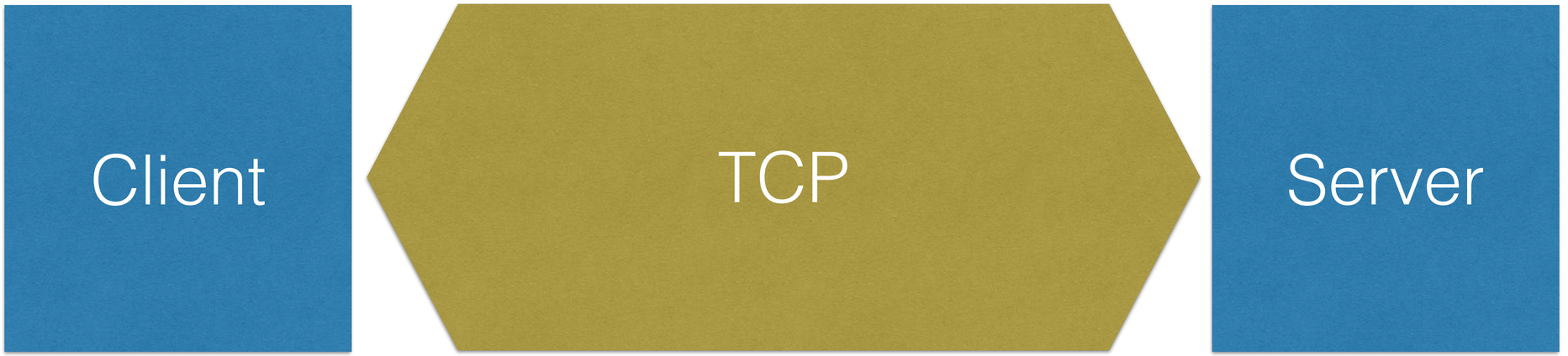
Real-time **server to client** communication

04 LONG POLLING

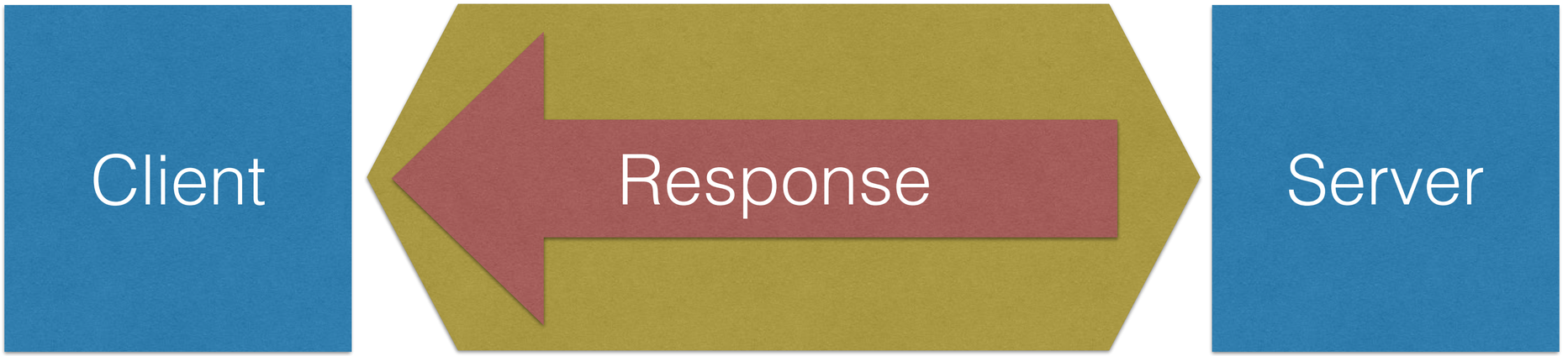
Pure HTTP

The server delays the response


Client
Server


Client
TCP
Server


Client
HTTP Request
Server


Client
Response
Server


Client
HTTP Request
Server

Subject to the ~6 connection limit

Blackout handling

Benefits?

Supported everywhere

Same controls flow

Better for large numbers of users

What are they useful for?

Same as SSE

05

COMPARISON

	WebSockets	Server Sent Events	Long Polling
Number of parallel connections from Browser	1024	~6 per domain	~6 per domain
Load Balancing and Proxying	Non-Standard / Complicated	Standard / Easy	Standard / Easy
Supported on all browsers	Yes (90%)	No (84% - not on IE and Edge)	Yes (100%)
Dropped Client Detection	Yes	No	No
Reconnection Handling	No	Yes	No

06

CONCLUSION

QUESTIONS

 @monorkin

 hi@stanko.io

 github.com/stankec

blog.stanko.io/343aed40aa9b