EVENTS AND SOCKET.IO

Building real-time software

GRACE HOPPER
ACADEMY

```
var userTweets = new EventEmitter();

// Elsewhere in the program . . .
userTweets.on('newTweet', function (tweet) {
    console.log(tweet);
});

// Elsewhere in the program . . .
userTweets.emit('newTweet', {
    text: 'Check out this fruit I ate'
});
```

GRACE HOPPER
ACADEMY

```
var userTweets = new EventEmitter();

// Elsewhere in the program . . .
userTweets.on('newTweet', function (tweet) {
    console.log(tweet);
});

// Elsewhere in the program . . .
userTweets.emit('newTweet', {
    text: 'Check out this fruit I ate'
});
```

```
var userTweets = new EventEmitter();

// Elsewhere in the program . . .
userTweets.on('newTweet', function (tweet) {
    console.log(tweet);
});

// Elsewhere in the program . . .
userTweets.emit('newTweet', {
    text: 'Check out this fruit I ate'
});
```

```
var userTweets = new EventEmitter();

// Elsewhere in the program . . .
userTweets.on('newTweet', function (tweet) {
    console.log(tweet);
});

// Elsewhere in the program . . .
userTweets.emit('newTweet', {
    text: 'Check out this fruit I ate'
});
```

```
var userTweets = new EventEmitter();

// Elsewhere in the program . . .
userTweets.on('newTweet', function (tweet) {
    console.log(tweet);
});

// Elsewhere in the program . . .
userTweets.emit('newTweet', {
    text: 'Check out this fruit I ate'
});
```

GRACE HOPPER
ACADEMY

EVENT EMITTERS

- Objects that can “emit” specific events with a payload to any amount of registered listeners
- An instance of the “observer/observable” a.k.a “pub/sub” pattern
- Feels at-home in an *event*-driven environment

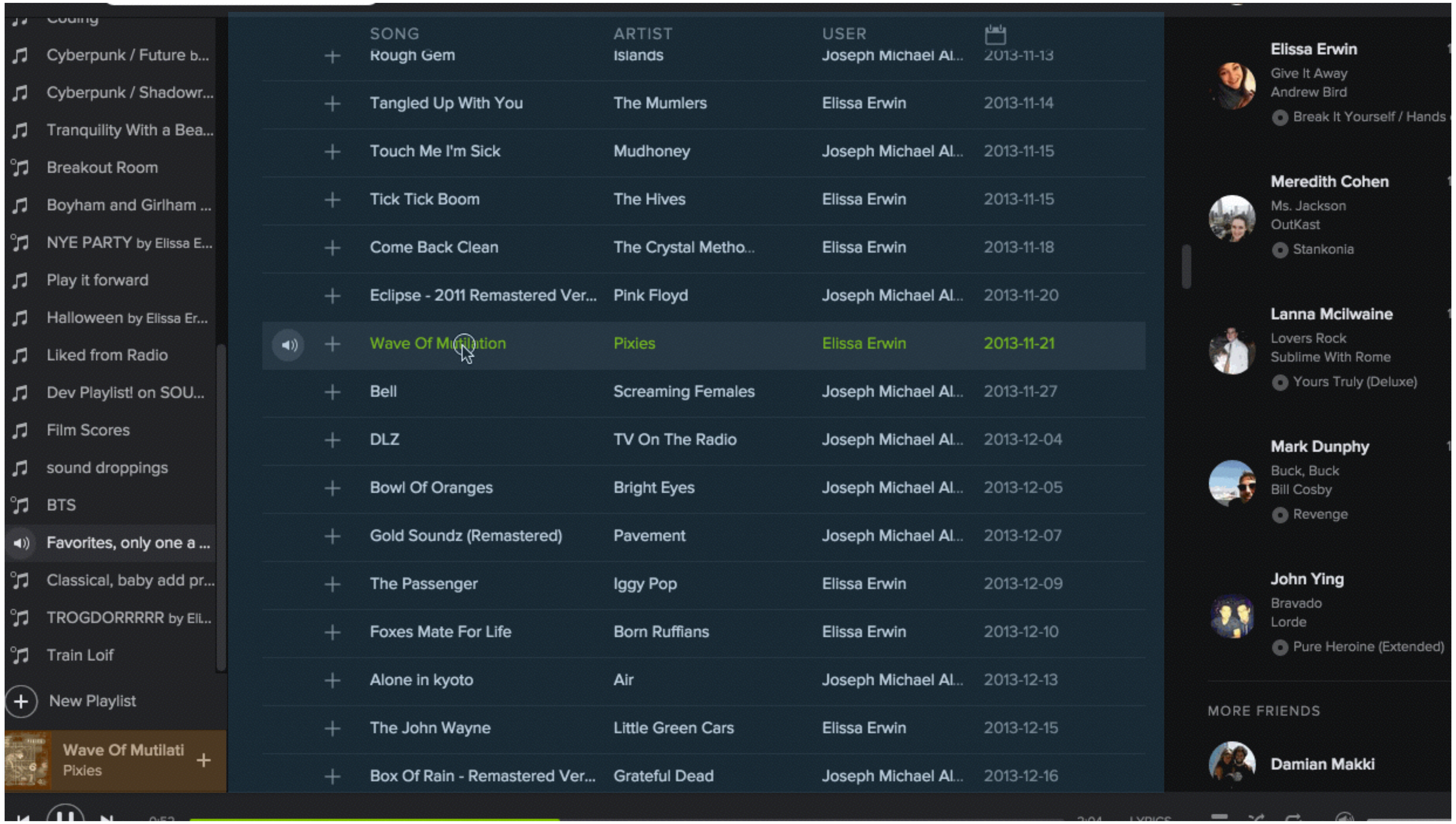
PRACTICAL USES

- Connect two decoupled parts of an application

```
var currentTrack = new EventEmitter();
```

```
currentTrack.emit('changeTrack', newTrack);
```

```
currentTrack.on('changeTrack', function (newTrack) {
    // Display new track!
});
```



Cyberpunk / Future b...
Cyberpunk / Shadowr...
Tranquility With a Bea...
Breakout Room
Boyham and Girlham ...
NYE PARTY by Elissa E...
Play it forward
Halloween by Elissa Er...
Liked from Radio
Dev Playlist! on SOU...
Film Scores
sound droppings
BTS
Favorites, only one a ...
Classical, baby add pr...
TROGDORRRRR by Eli...
Train Loif
New Playlist
Wave Of Mutilati
Pixies

SONG	ARTIST	USER	
Rough Gem	Islands	Joseph Michael Al...	2013-11-13
Tangled Up With You	The Mumlers	Elissa Erwin	2013-11-14
Touch Me I'm Sick	Mudhoney	Joseph Michael Al...	2013-11-15
Tick Tick Boom	The Hives	Elissa Erwin	2013-11-15
Come Back Clean	The Crystal Metho...	Elissa Erwin	2013-11-18
Eclipse - 2011 Remastered Ver...	Pink Floyd	Joseph Michael Al...	2013-11-20
Wave Of Mutilation	Pixies	Elissa Erwin	2013-11-21
Bell	Screaming Females	Joseph Michael Al...	2013-11-27
DLZ	TV On The Radio	Joseph Michael Al...	2013-12-04
Bowl Of Oranges	Bright Eyes	Joseph Michael Al...	2013-12-05
Gold Soundz (Remastered)	Pavement	Joseph Michael Al...	2013-12-07
The Passenger	Iggy Pop	Elissa Erwin	2013-12-09
Foxes Mate For Life	Born Ruffians	Elissa Erwin	2013-12-10
Alone in kyoto	Air	Joseph Michael Al...	2013-12-13
The John Wayne	Little Green Cars	Elissa Erwin	2013-12-15
Box Of Rain - Remastered Ver...	Grateful Dead	Joseph Michael Al...	2013-12-16

Elissa Erwin
Give It Away
Andrew Bird
Break It Yourself / Hands

Meredith Cohen
Ms. Jackson
OutKast
Stankonia

Lanna Mcilwaine
Lovers Rock
Sublime With Rome
Yours Truly (Deluxe)

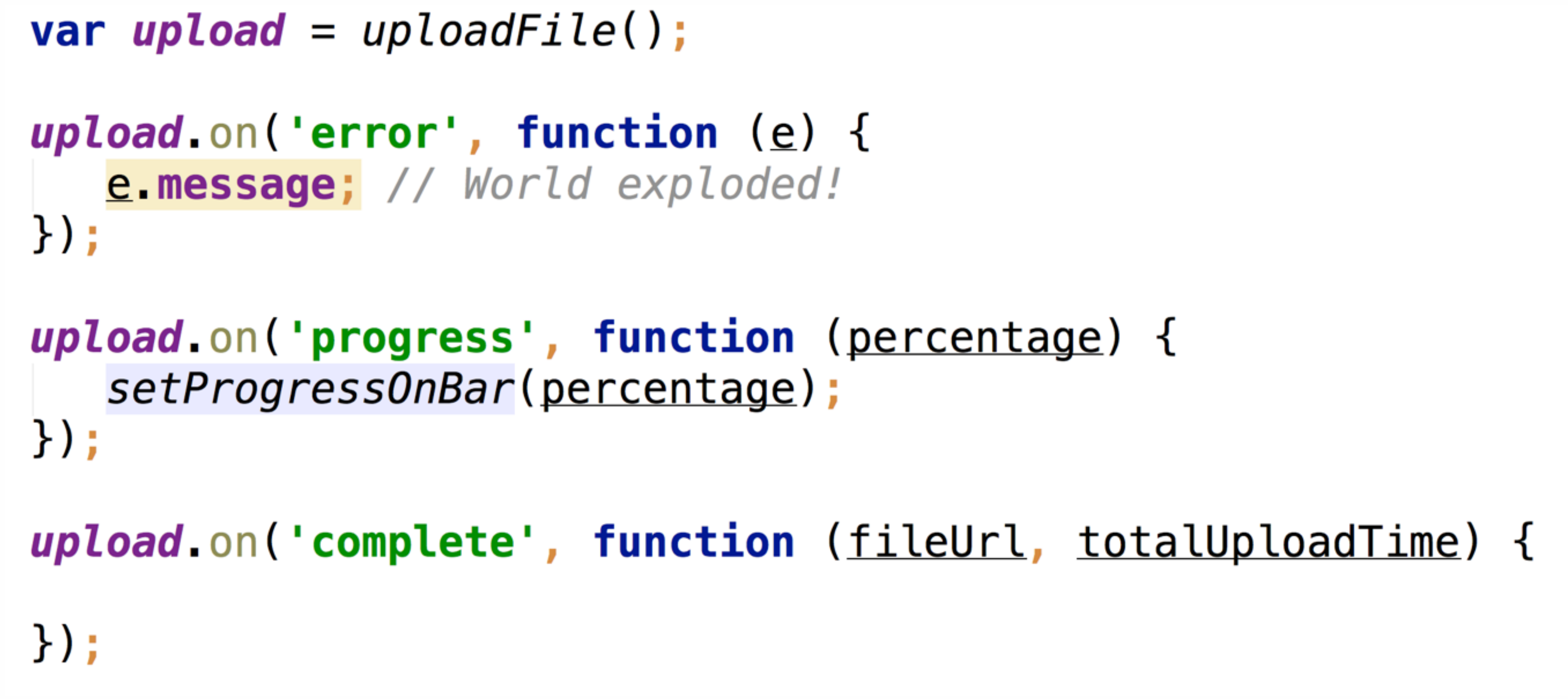
Mark Dunphy
Buck, Buck
Bill Cosby
Revenge

John Ying
Bravado
Lorde
Pure Heroine (Extended)

MORE FRIENDS
Damian Makki

PRACTICAL USES

- Represent multiple asynchronous events on a single entity.



```
var upload = uploadFile();

upload.on('error', function (e) {
    e.message; // World exploded!
});

upload.on('progress', function (percentage) {
    setProgressOnBar(percentage);
});

upload.on('complete', function (fileUrl, totalUploadTime) {

});
```

ALL OVER NODE

- server.on(‘request’)
- request.on(‘data’) / request.on(‘end’)
- process.stdin.on(‘data’)
- mongoose.on(‘connection’)
- Streams

HTTP, PART 2

Sequels are always worse than the original

WHAT WE KNOW ABOUT HTTP

- A client makes a “request” to a server
- Server receives this “request” and generates a “response”
- One request, one response: them’s the rules
- Requests can include a body (payload)
- Responses can include a body (payload)

The New York Times

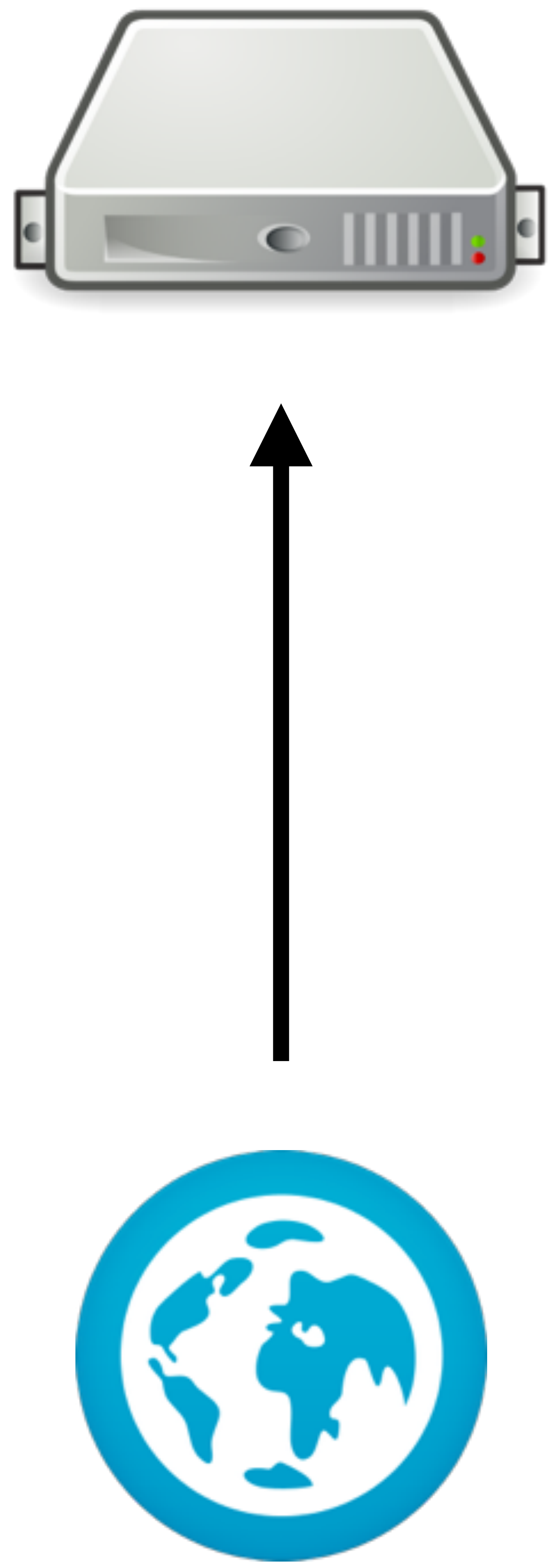
2014

FIFA WORLD CUP
Brasil

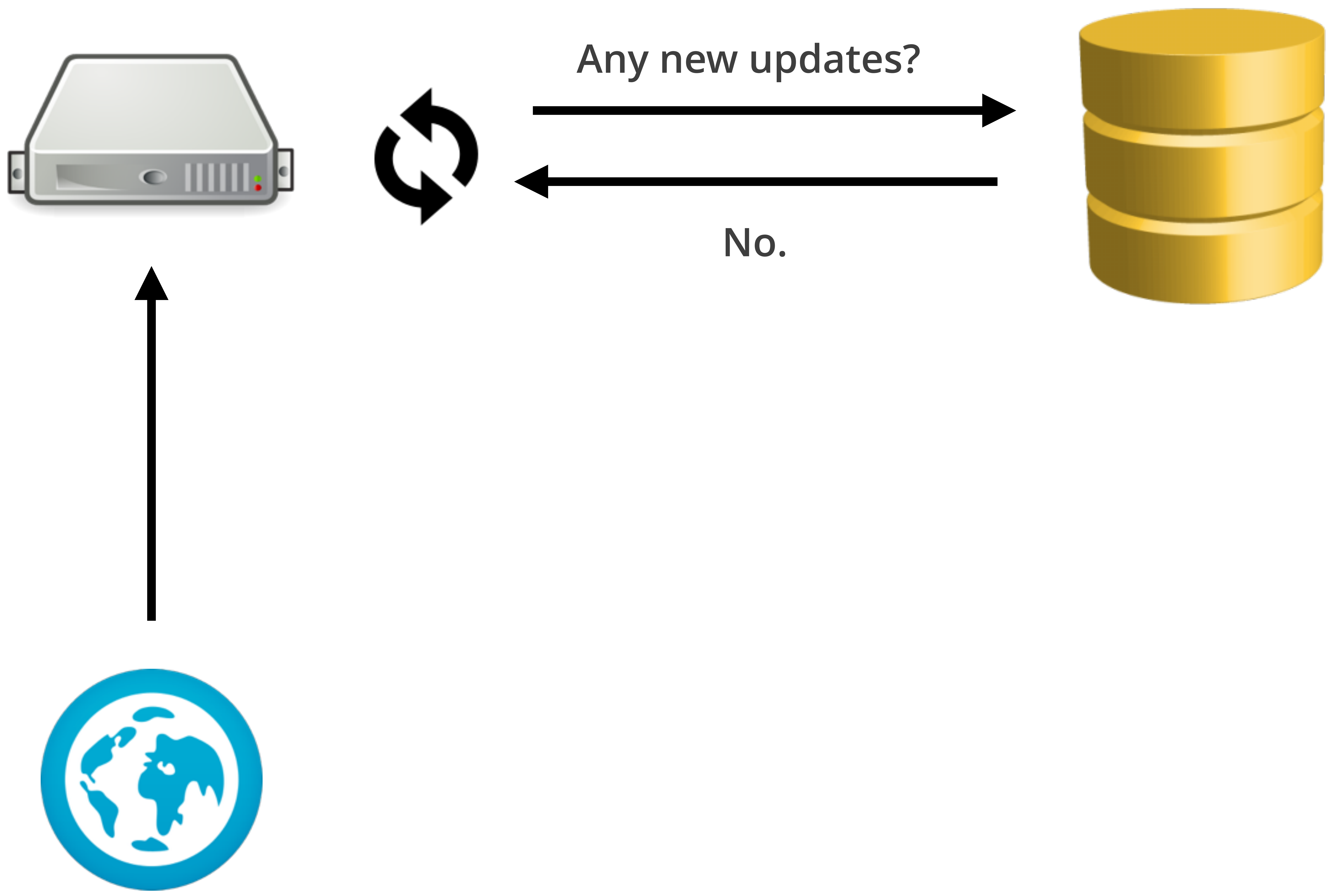
LIVE WORLD CUP COVERAGE

- A user visits a web page
- This web page has a live updating list of game coverage (“events”) provided by New York Times commentator (“Brazil receives yellow card”/“Germany scores goal”)
- When the event line is submitted by the commentator, it should immediately display to the user

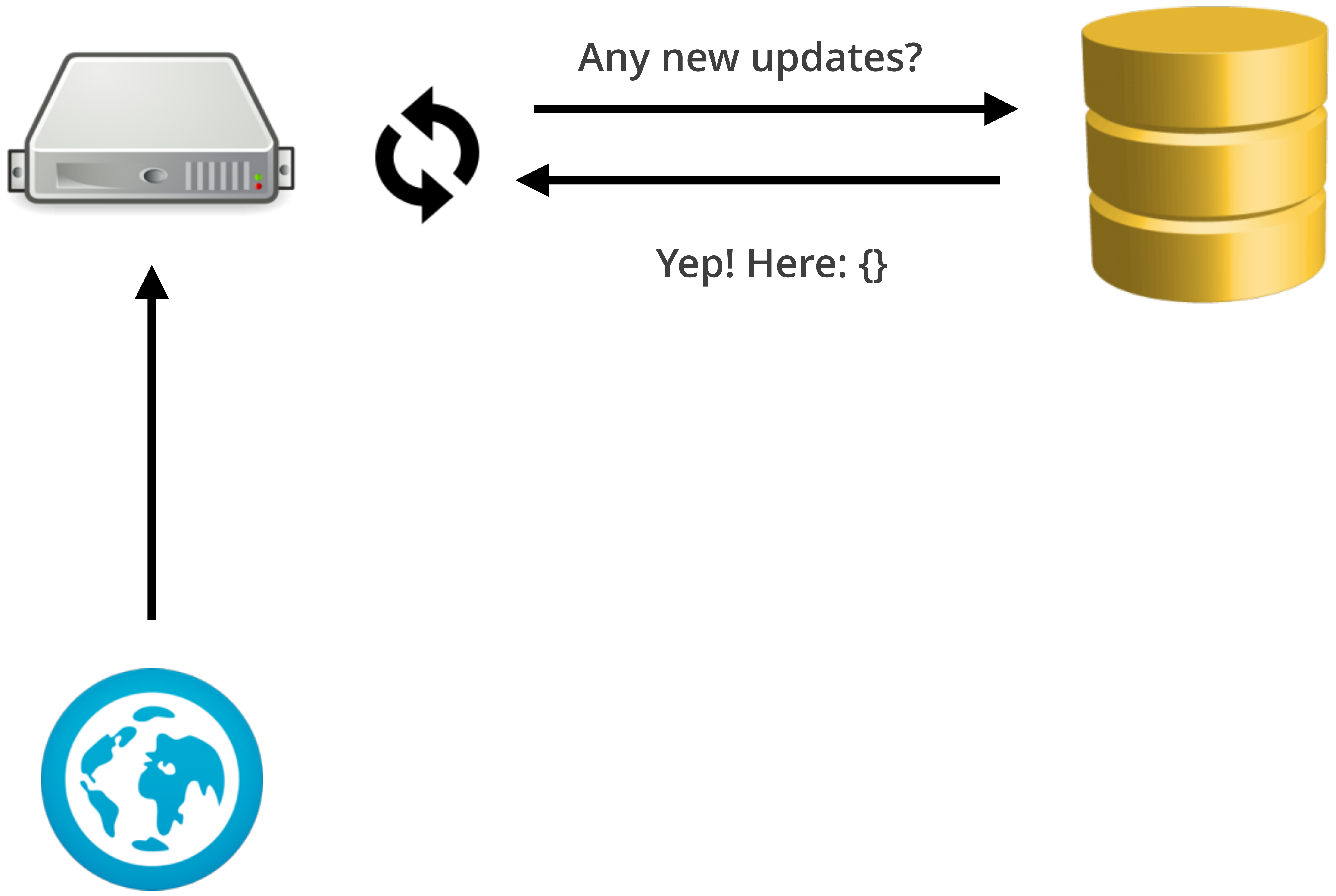
HTTP LONG POLLING



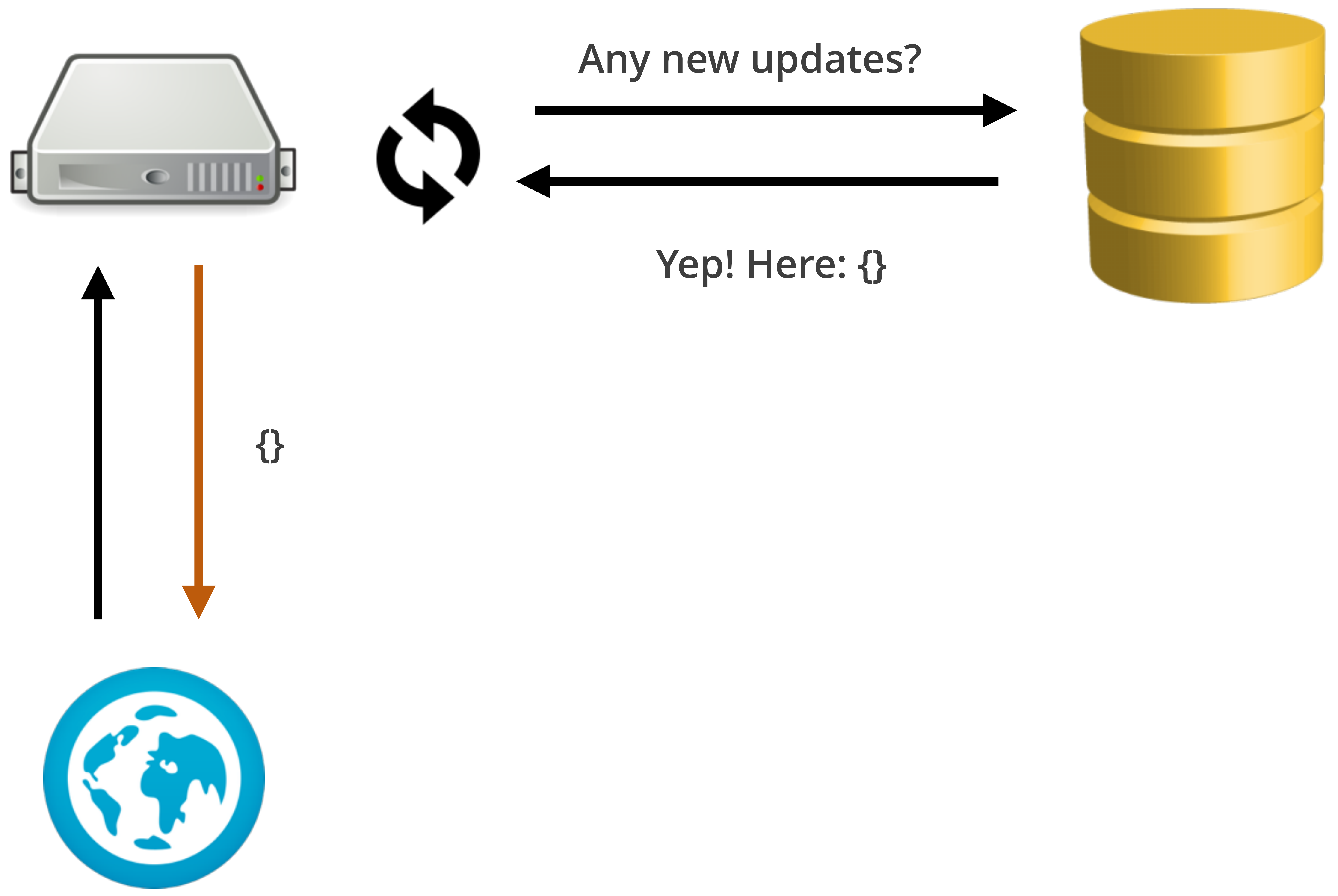
HTTP LONG POLLING



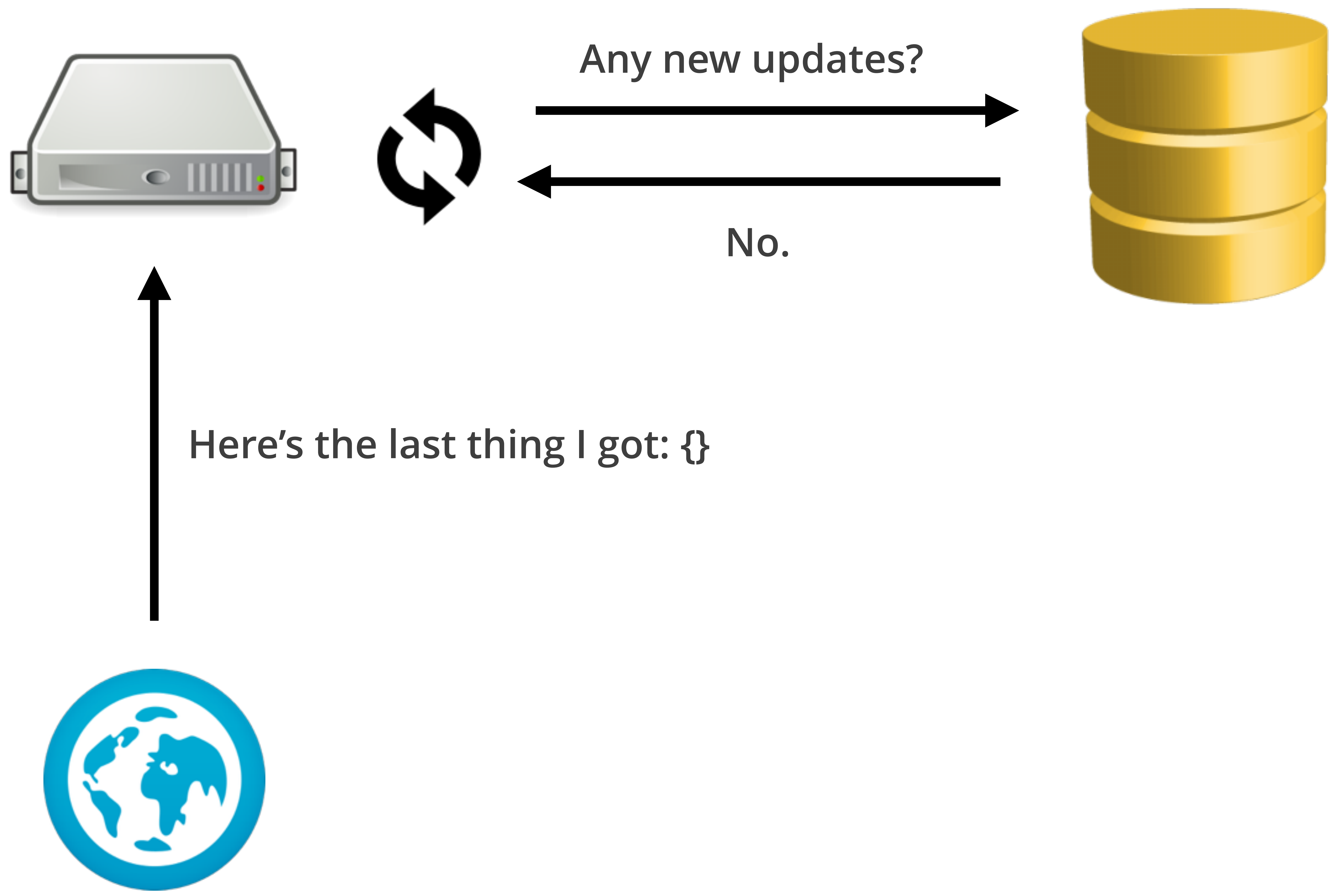
HTTP LONG POLLING



HTTP LONG POLLING



HTTP LONG POLLING



HTTP IS A REQUEST/RESPONSE PROTOCOL

- Clients must send a *request* before the server can issue a *response*
- There is no way for the server to *push* data to the client without an outstanding request
- No live updates without long polling 😢

TCP

Transmission Control Protocol

TCP

- **Protocol: standardized way that computers communicate with one another**
- **Establishes a reliable, duplex connection between two machines that persists over time**
  - **Reliable:** All your data gets there in the order you sent it
    - (or you know that it didn't)
  - **Duplex:** Either end of the connection can send or receive bits
  - **Persistent:** The connection lasts until one side ends it
- **TCP is a *transport* layer protocol**

INTERNET PROTOCOL SUITE — LAYERS WITHIN LAYERS

(opening into more layers)

4: Application

Data format, “actions”, “requests”

- HTTP — The web
- TLS / SSL — Encryption
- DNS — Domain name lookup
- SMTP — E-mail

3: Transport

Ports, reliable connections: packet ordering, retries, flow control

- TCP (Reliable & slow)
- UDP (Unreliable. Used for realtime video & games)
- ICMP (Ping, IP errors)

2: Internet

Addressing & Routing

- IPv4
- IPv6
- IPSec (IP-level packet encryption)

1: Link

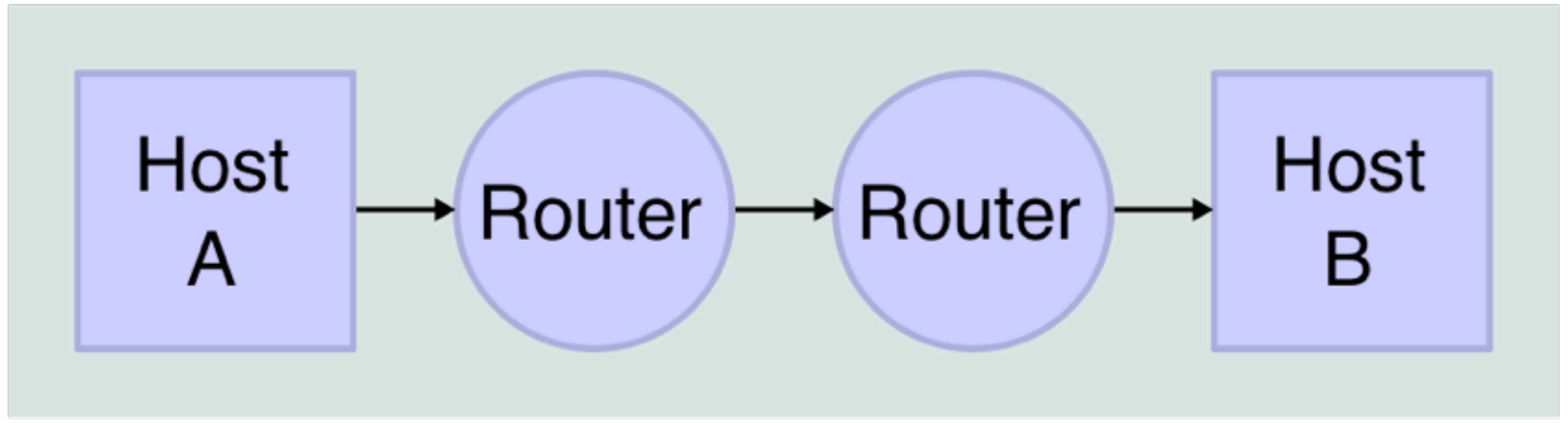
What physical media are we using?

Who can talk when?

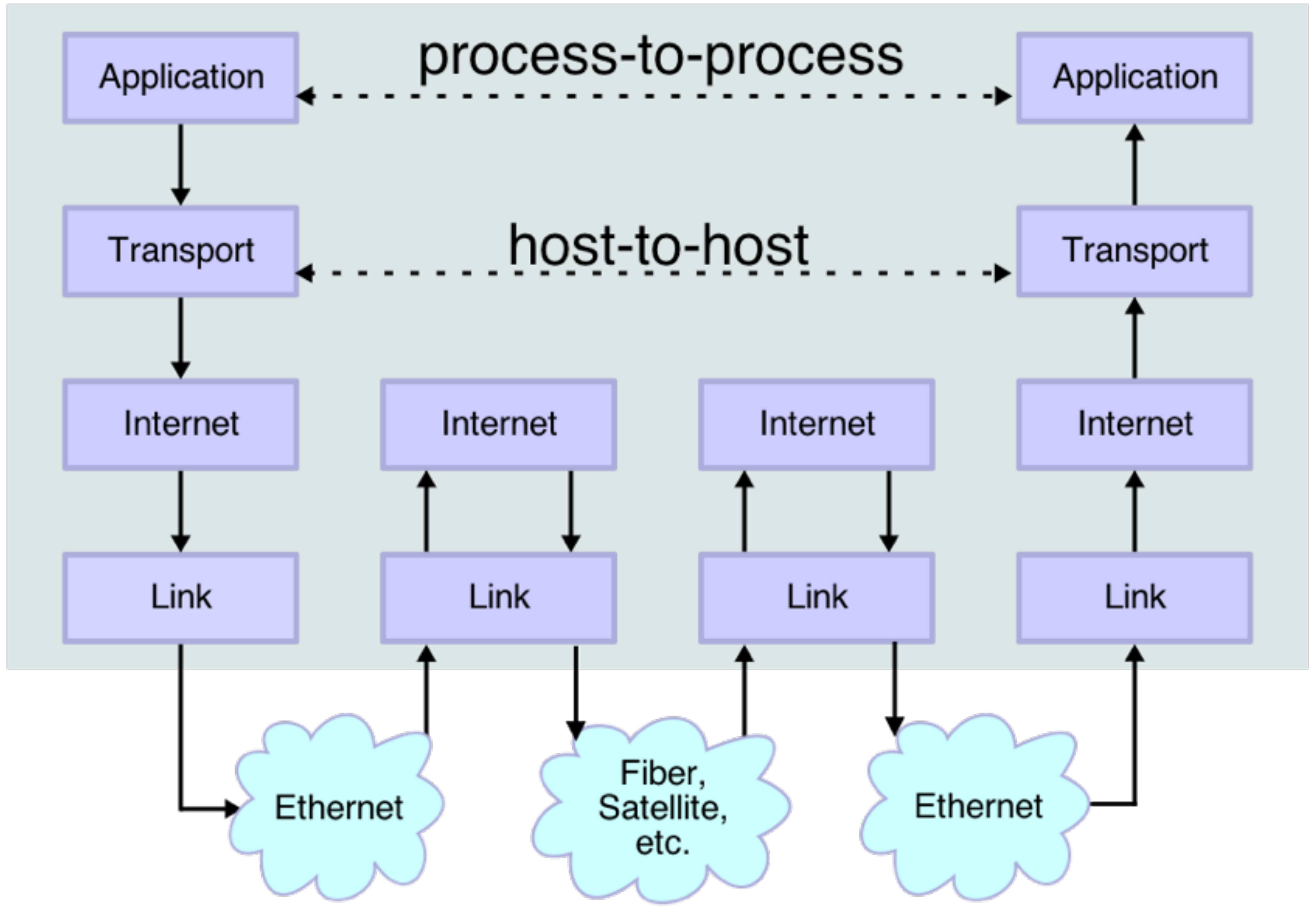
- Ethernet (over 1000-BaseT, 1000-BaseSX, etc…)
- WiFi 802.11*{a{c,d,f,h,i,j,q,x,y},b,g,n}*

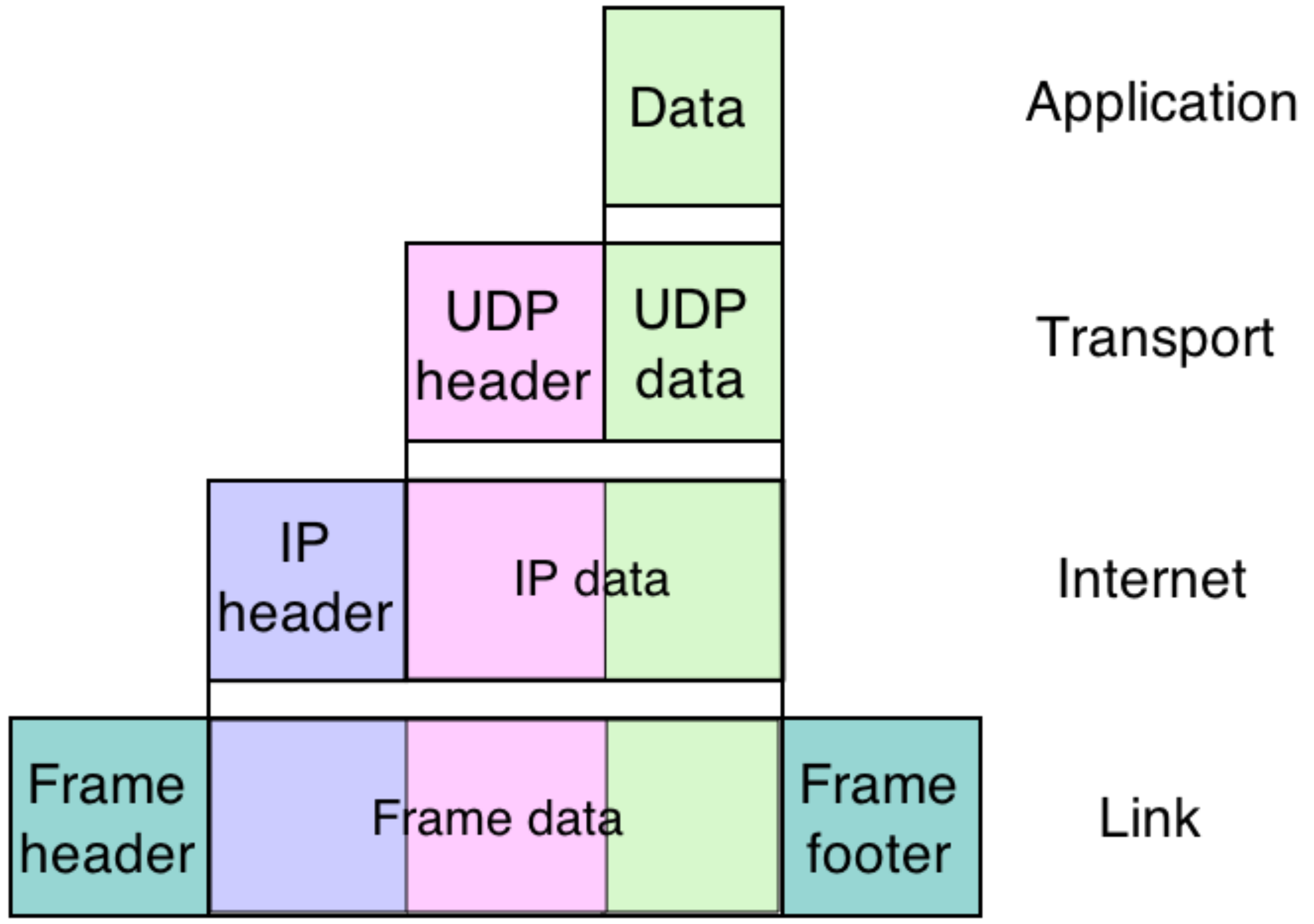
GRACE HOPPER ACADEMY

Network Topology



Data Flow

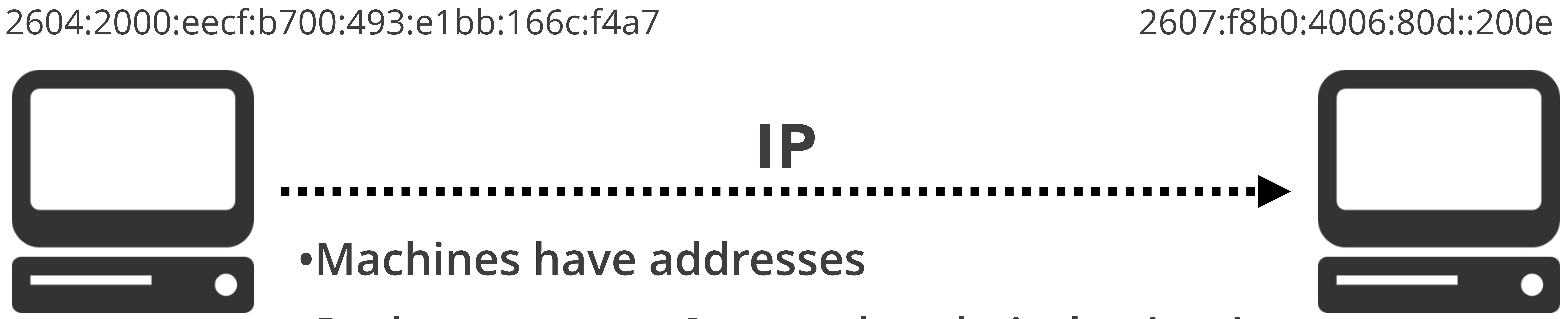




These are IPv6

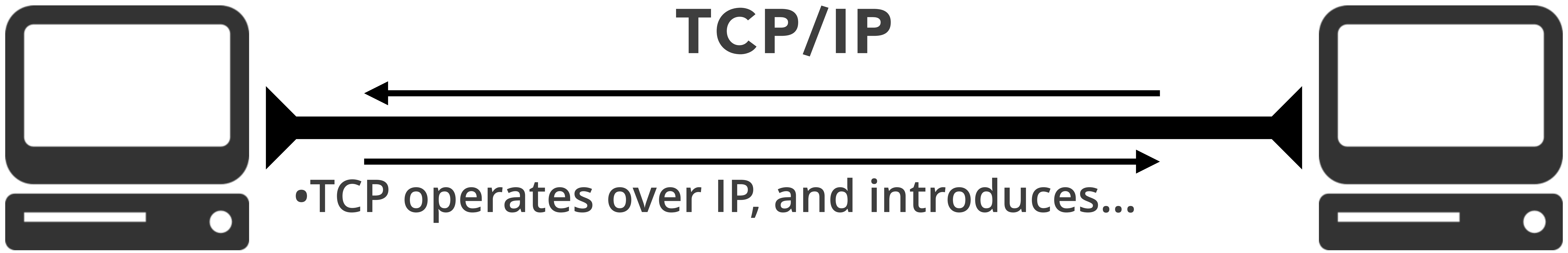
They're really long so there can be more of them than with IPv4.

(See, we ran out of IPv4 addresses. So we made IPv6 addresses big enough to assign an address to every atom in the universe every nanosecond.)



- Machines have addresses
- Packets are sent & routed to their destination
- Do they get there? In what order? Nobody knows!
- What process was this message meant for? Unclear!
- Unreliable, connectionless

GRACE HOPPER ACADEMY



- TCP operates over IP, and introduces...
- Ports: to figure out which process gets the packet
- Connections: to figure out packet ordering & loss
- Retries & flow control: to deal with packet loss
- Reliable connection that persists over time

TCP AND HTTP

- HTTP is an application layer protocol
- It (usually) operates over TCP, (usually) on port 80
  - But “HTTP only presumes a reliable transport; any protocol that provides such guarantees can be used” — HTTP 1.1 Spec
  - HTTPS, for instance, operates over TLS on port 443
- Implements the idea of a “session”, which establishes a TCP socket for the client to make requests and the server to issue responses

CLIENT OPENS A TCP CONNECTION TO SERVER

let’s synchronize!
ok, let’s synchronize!
ok!

GRACE HOPPER ACADEMY

TCP CONNECTION IS ESTABLISHED

GRACE HOPPER ACADEMY

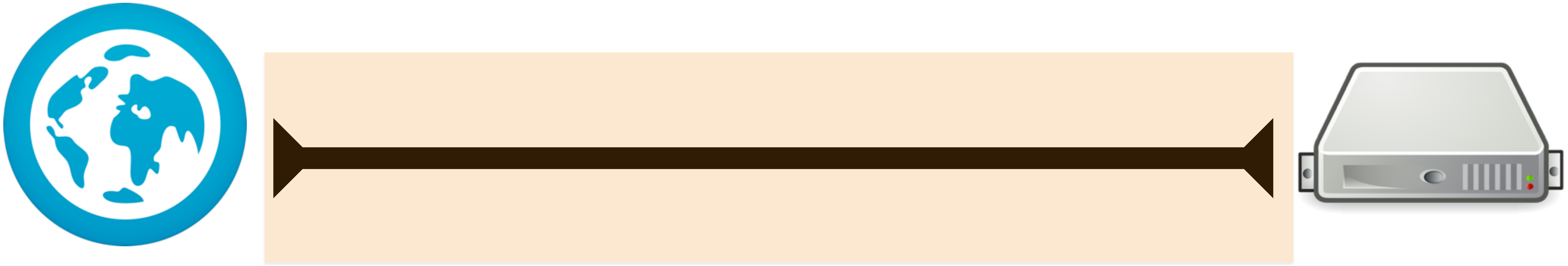
CLIENT SENDS A REQUEST

(over the connection)

SERVER SENDS A RESPONSE

(over the connection)

TCP CONNECTION STAYS OPEN



GRACE HOPPER ACADEMY

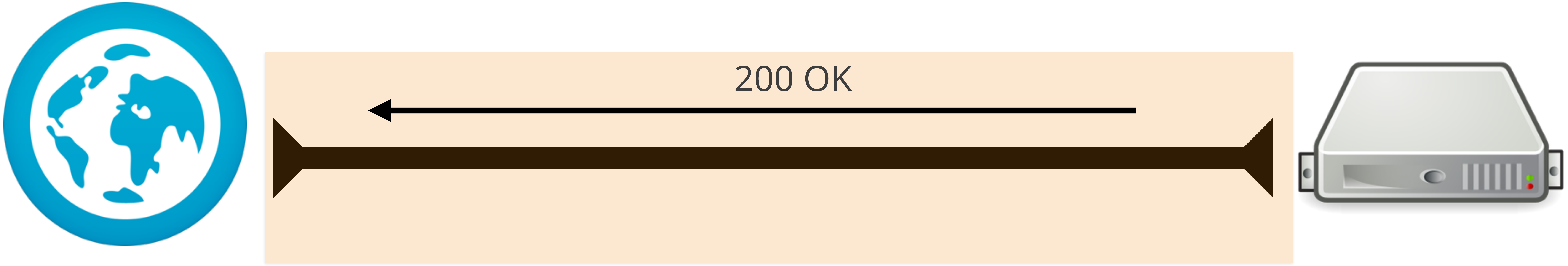
CLIENT SENDS MORE REQUESTS

(over the same connection)

GET /Flow_control HTTP/1.1

SERVER SENDS MORE RESPONSES

(over the same connection)



EVENTUALLY, YOU CLOSE THE TAB

OR YOU DON'T SAY ANYTHING FOR A WHILE AND THE SERVER TIMES OUT

AND ONE OF YOU ENDS THE CONNECTION

HTTP 1.1 REQUEST / RESPONSE CYCLE

- Client sends a request
- Server sends a response
- Server can't "push" more data to the client unless the client makes another request
  - ...Even though there's this tasty TCP connection just sitting around

WEBSOCKETS AND SOCKET.IO

GRACE HOPPER ACADEMY

WEBSOCKET

- Application-layer protocol
- Message-based
- Either the client or server can choose to send a message at any time
    - No “requests” or “responses” unless *you* design the protocol for them
- Allows for awesome real-time software
- So how do you open a WebSocket?

WEBSOCKETS START WITH HTTP

Client says:

GET /chat HTTP/1.1

Host: server.example.com

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Sec-WebSocket-Protocol: chat, superchat

Sec-WebSocket-Version: 13

Origin: http://example.com

Server replies:

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

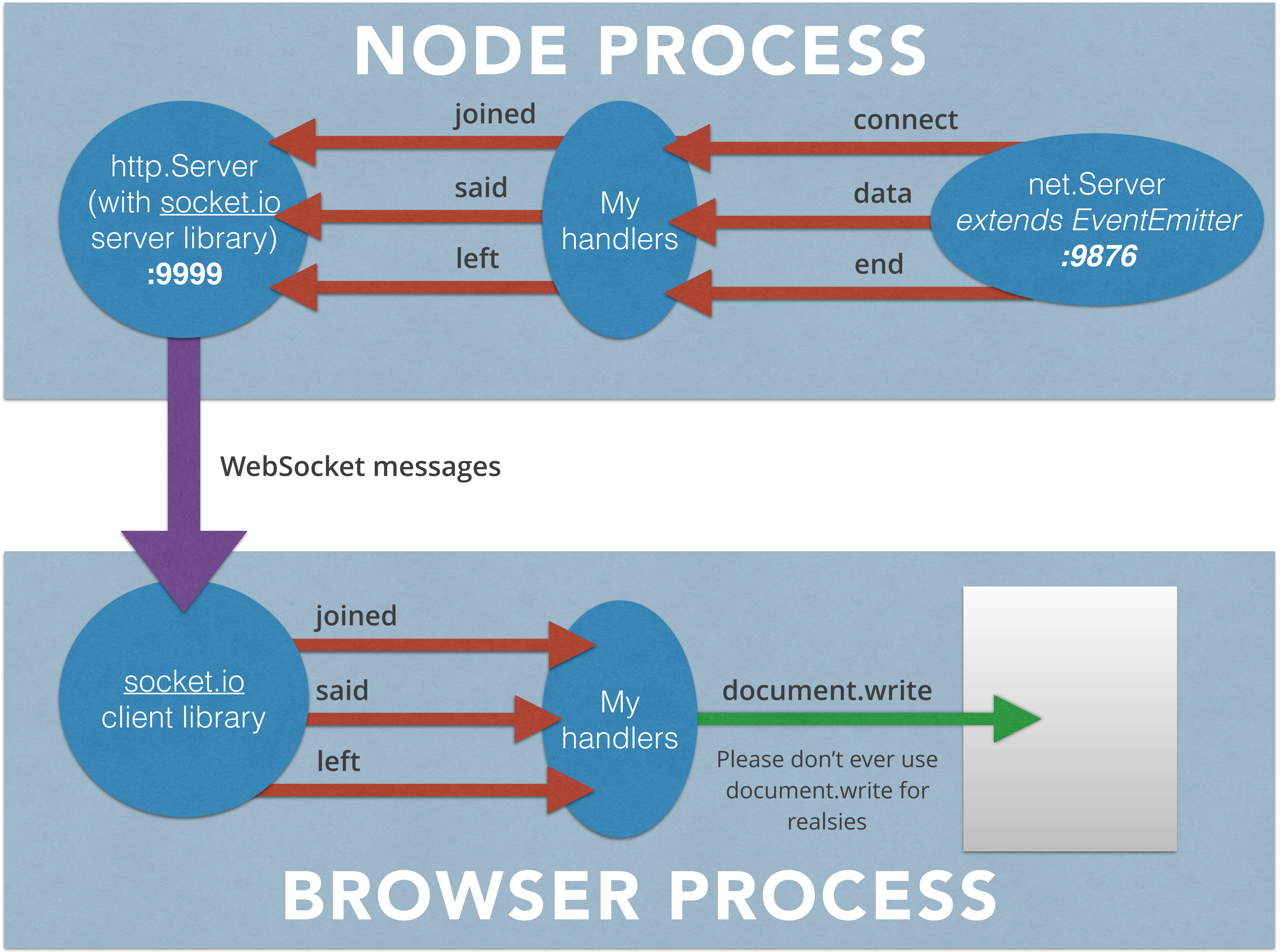
Sec-WebSocket-Accept: HSmrc0sMlYUkAGmm5OPpG2HaGWk=

Sec-WebSocket-Protocol: chat

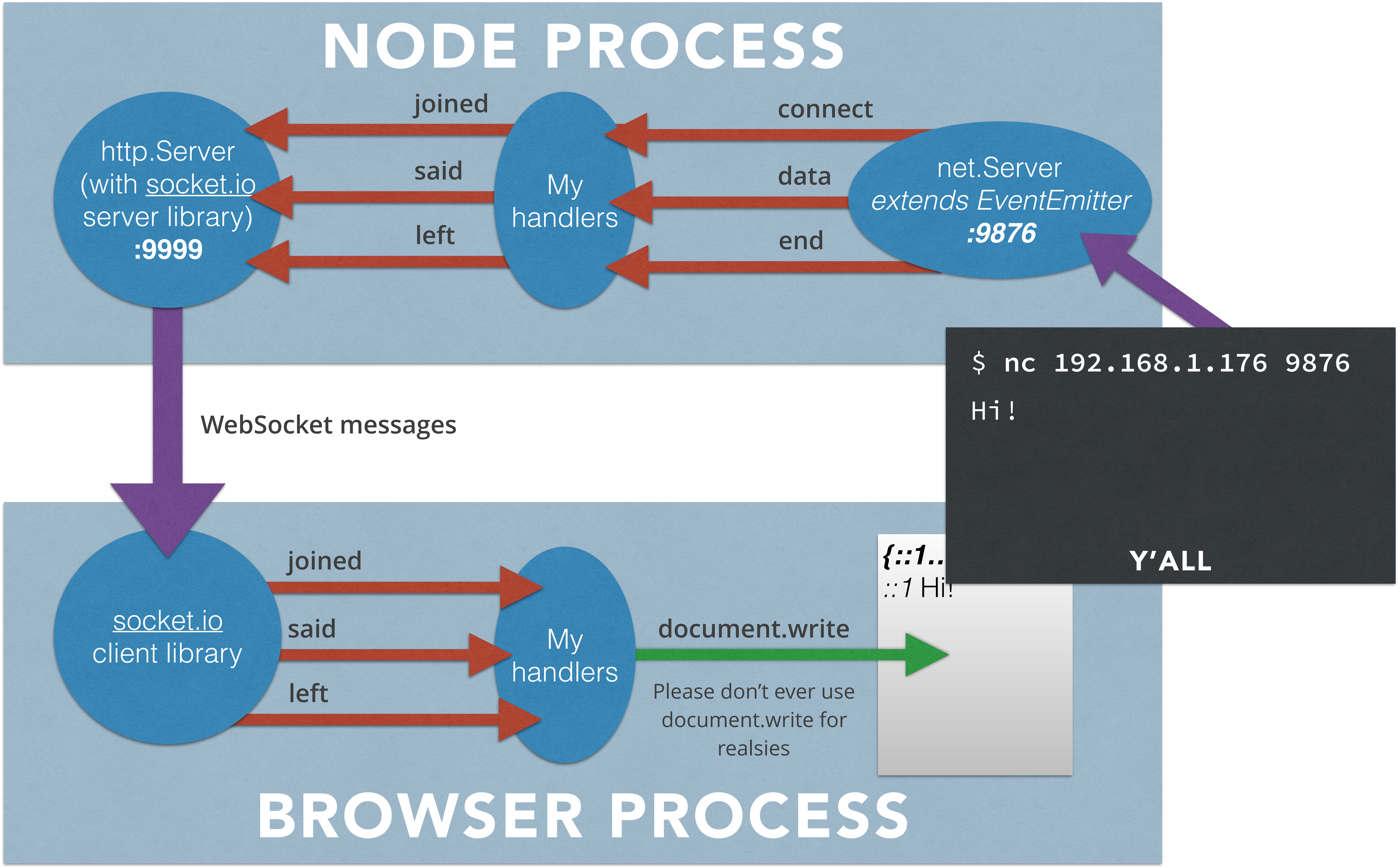
And now WebSocket has taken over the connection.

SOCKET.IO

- You don’t have to implement that
- Socket.IO is a duet of libraries (one for server-side [node.js] and one for client-side [the browser])
- Abstracts the complex implementation of websockets for easy use
- Extensively uses `EventEmitters`
  - `EventEmitters` are a good fit for a message-based protocol



GRACE HOPPER ACADEMY



```
$ nc 192.168.1.176 9876
Hi!
```

GRACE HOPPER ACADEMY

USE CASES

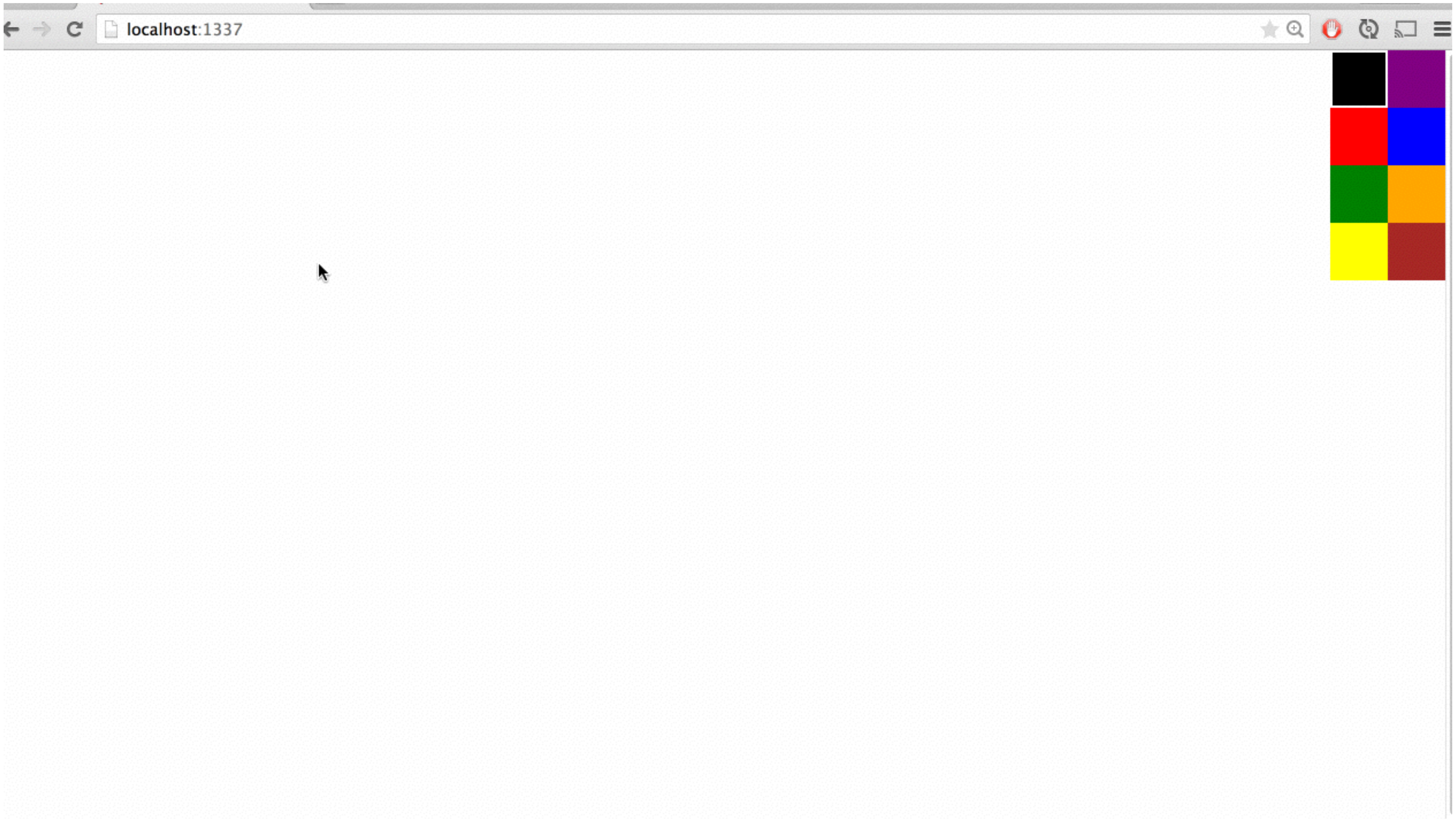
- Networked enabled games
- Chat applications
- Collaborative applications
- Any “real-time” software

DRAWBACKS

- The server now *must* hold on to the connection
- Connections are expensive (they require memory within the operating system)
- If a socket sits dormant for a long time, it’s wasting server resources.
    - You could fix this in your app, though! You have the power!

OTHER SOCKET.IO NOTES

- Documentation leaves a lot to be desired
- Automatically uses fallbacks for different capabilities and environments (long polling, Flash)
- Has “rooms” and “namespaces” for socket organization
- Can “broadcast” to all sockets within a “room”



GRACE HOPPER
ACADEMY