StoneTor

WebSockets, Reactive APIs, & Microservices

Todd L. Montgomery
@toddlmontgomery

WebSocket!!!

Reactive!!!

OMG! Buzzwords!

Microservices!!!

WebSocket!!!

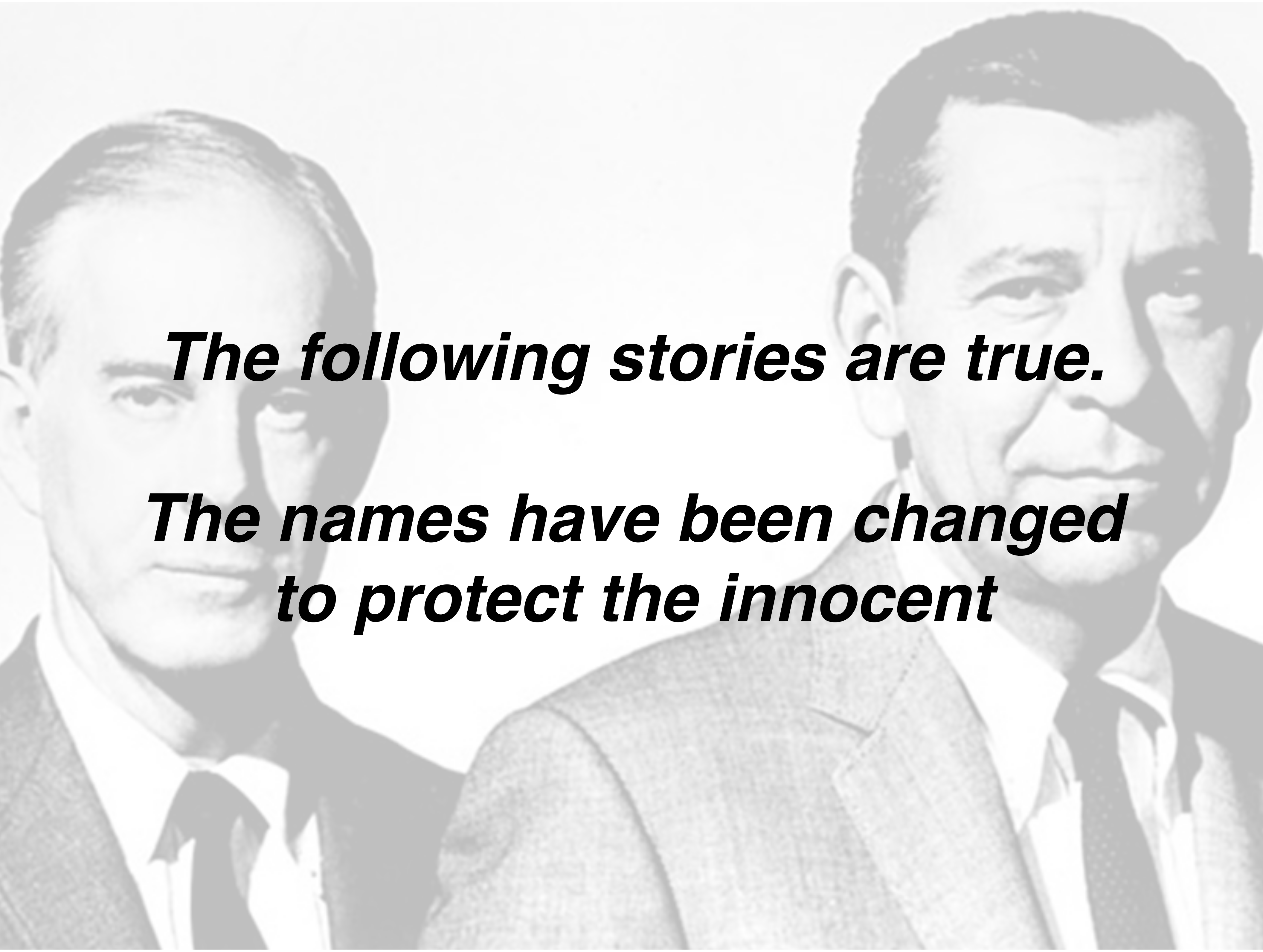
Reactive!!!

HYPE!!!11!

Microservices!!!

Takeaways

- ✓ ***Many old (new) techniques***
- ✓ ***Many new technologies***
- ✓ ***Constant evolution***
- ✓ ***Being used to great effect***



The following stories are true.

The names have been changed to protect the innocent

A Story in 3 Parts…

- ✓ ***The Case of WebSocket?***
- ✓ ***Touch of Reactive API?***
- ✓ ***The Big Microservice?***

WebSocket

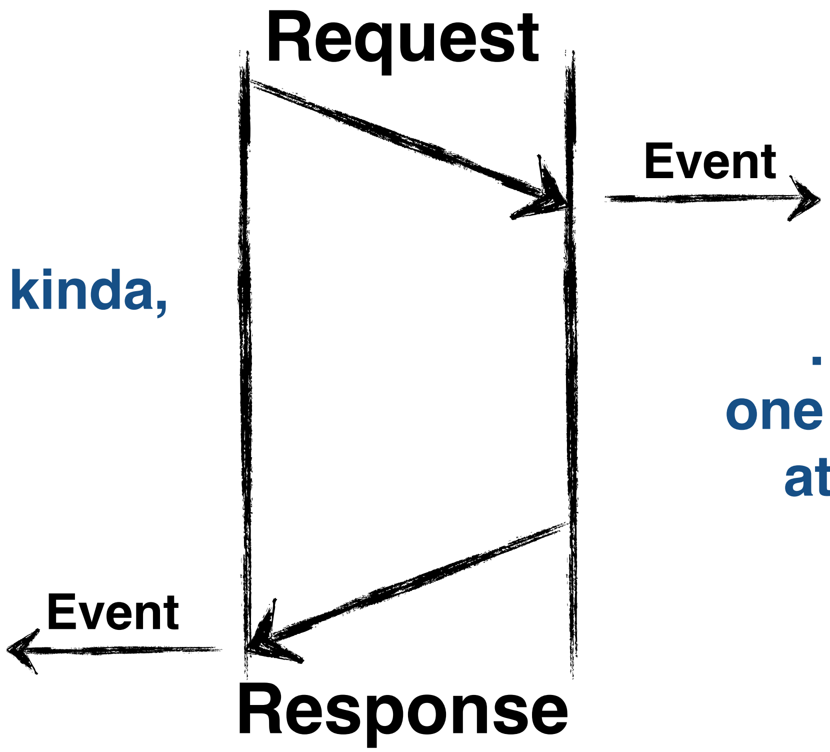
Request-Response is not enough

It's never been enough!

Where we have come from…

HTTP 1.1
RFC 2068, 2616, …, 7230-7240

June 2014



Bi-Directional… kinda, but…

… only one direction at-a-time

Synchronous Request/Response

- ✓ ***ASCII Encoded***
- ✓ ***Very Synchronous***
- ✓ ***Many TCP Connections***
- ✓ ***Request / Response Focus***

“Everything changes and nothing stands still”

— Heraclitus of Ephesus

Changing Needs

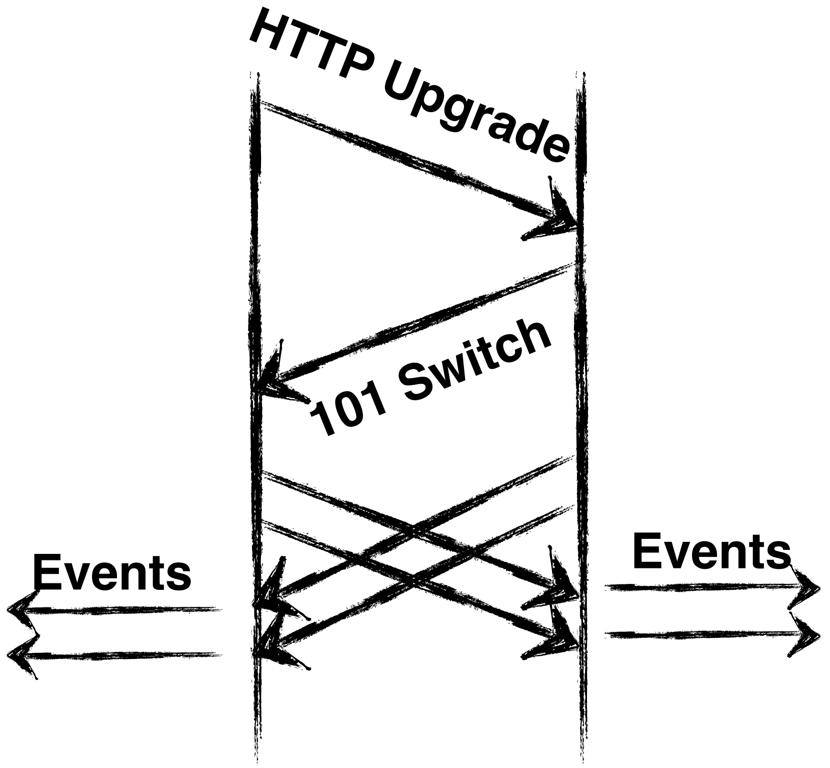
Changing Needs

- ✓ ***What about data feeds?***
- ✓ ***What about interactivity?***
- ✓ ***What about …***

WebSocket
RFC 6455

Really a Transport Protocol

Async Request/ Response

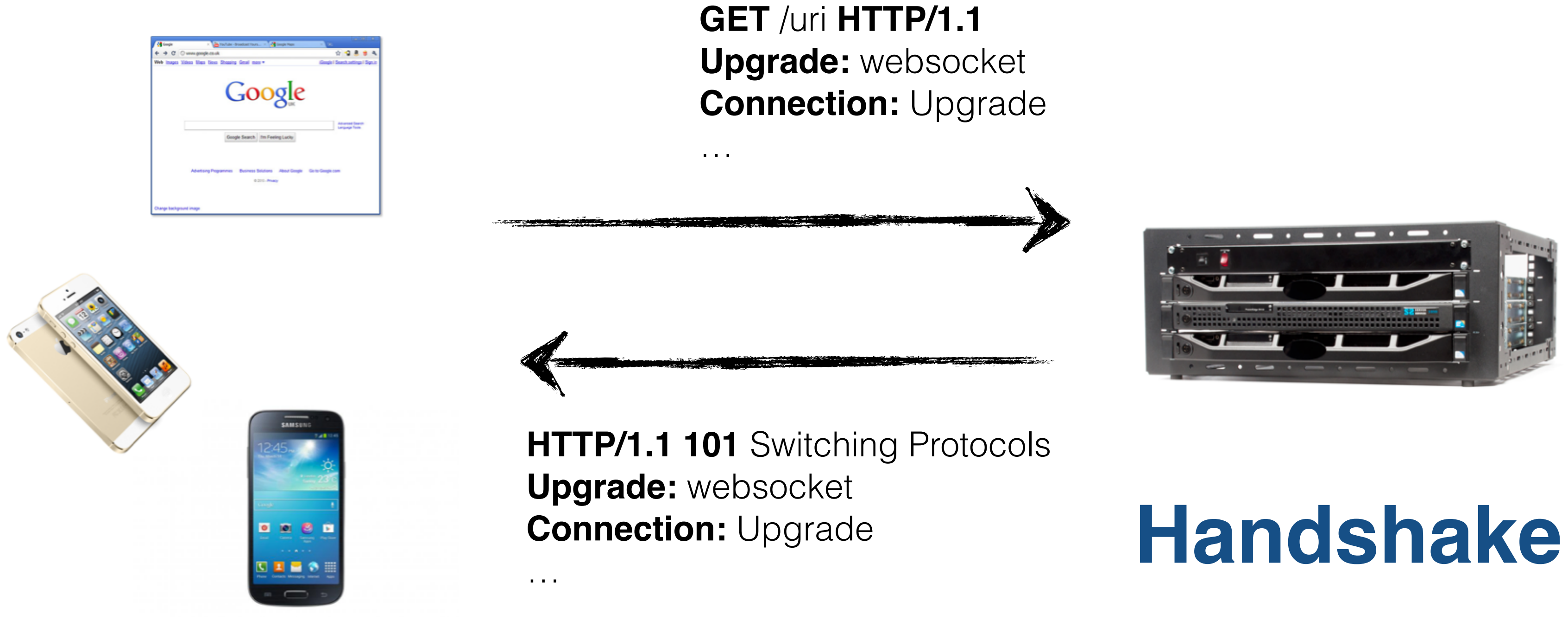


Streaming

Ingest

Full Duplex, Asynchronous
"TCP over the Web"

https://tools.ietf.org/html/rfc6455

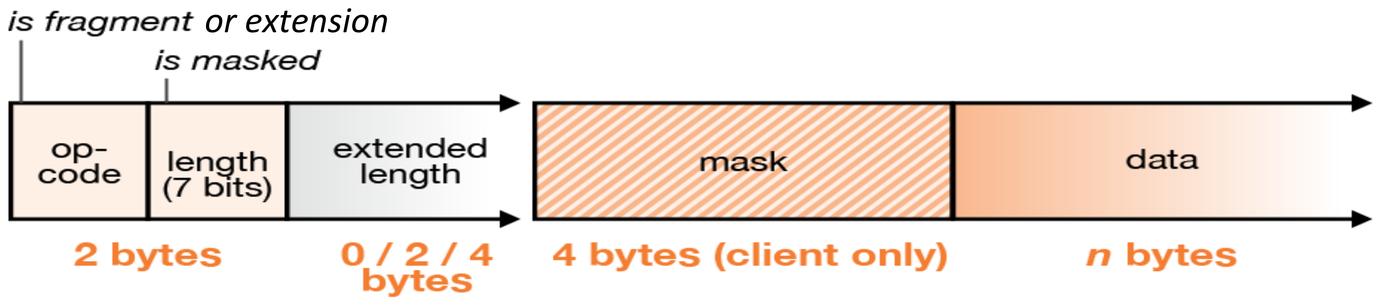


GET /uri **HTTP/1.1**
Upgrade: websocket
Connection: Upgrade
…

HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
…

Handshake

Simple Framing



Challenges?

Challenges

- ✓ ***Hostile Intermediaries***
- ✓ ***Load Balancing***
- ✓ ***TLS Termination***

But who really uses WebSocket?

Telemetry
(Live Data Feeds)

Interactivity (Live Data Feed + Execution)

Responsiveness (Async UIs)

Reactive APIs

The Lure of Complexity

The Need for Simplicity

Today

Asynchronous is the norm

Composition is hard

ReactiveX

http://reactivex.io/

Observables

JavaScript

- ✓ ***RxJS***
- ✓ ***ECMAScript Observables***

https://github.com/ReactiveX/RxJS

https://github.com/zenparsing/es-observable

Challenges?

Challenges

- ✓ ***Non-Blocking Back Pressure***
- ✓ ***Heterogeneous Connectivity***

Dealing with Back Pressure

- ✓ ***ReactiveStreams***
- ✓ ***RxJava 2.0***

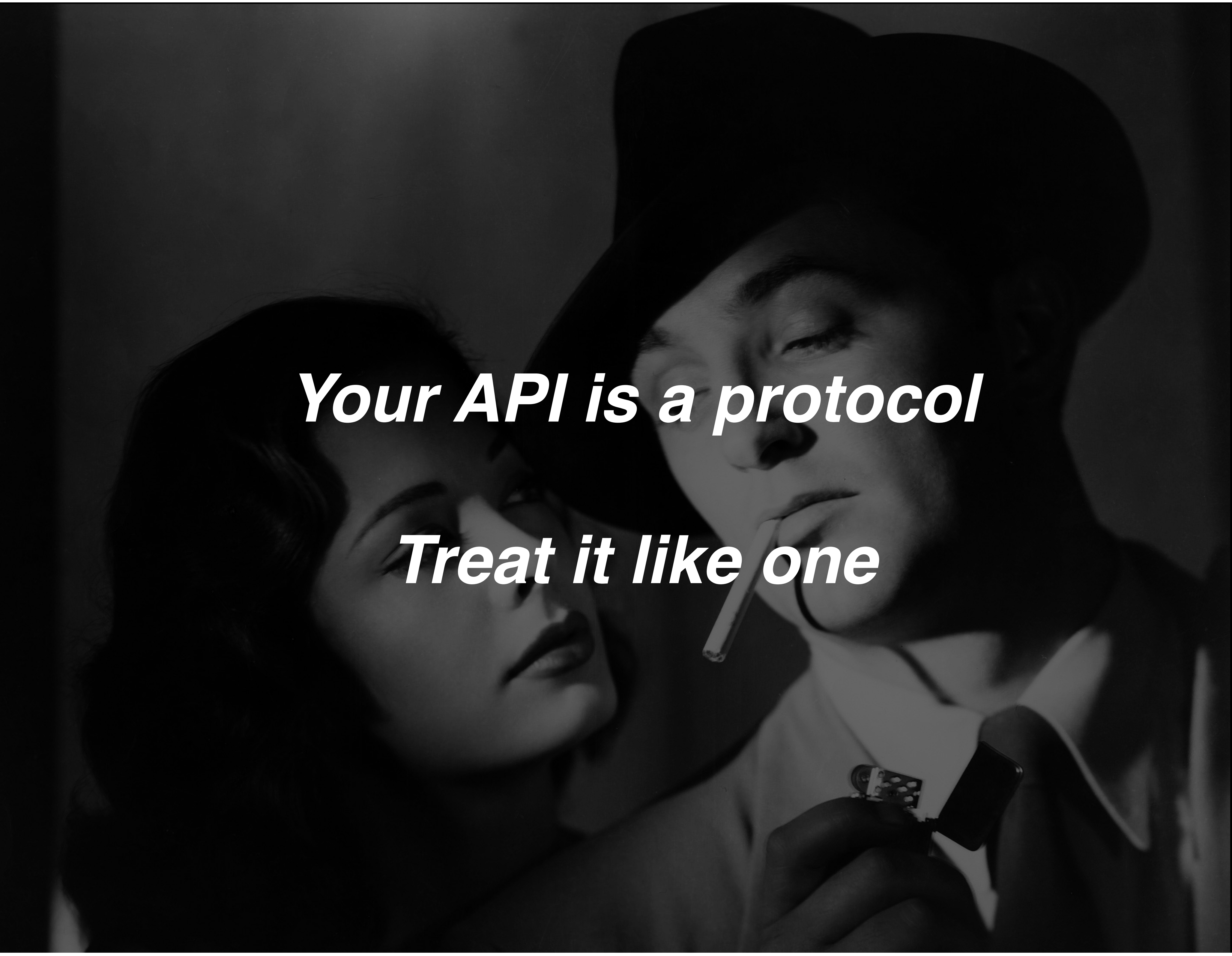
http://www.reactive-streams.org/

But who really uses Rx?

Responsiveness (Async UIs)

Interactivity (Live Data Feeds + Execution)

But, language constructs are not the main story



Your API is a protocol

Treat it like one

pro·to·col ***noun*** \ˈprō-tə-ˌkȯl, -ˌkōl, -ˌkäl, -kəl\

. . .

3 b : a set of conventions governing the *treatment* and especially the *formatting* of data in an electronic communications system <network *protocols*>

. . .

3 a : a code prescribing strict adherence to correct etiquette and precedence (as in diplomatic exchange and in the military services) <a breach of *protocol*>

Rx Heterogenous Connectivity

✓ ***ReactiveSocket***

http://reactivesocket.io/

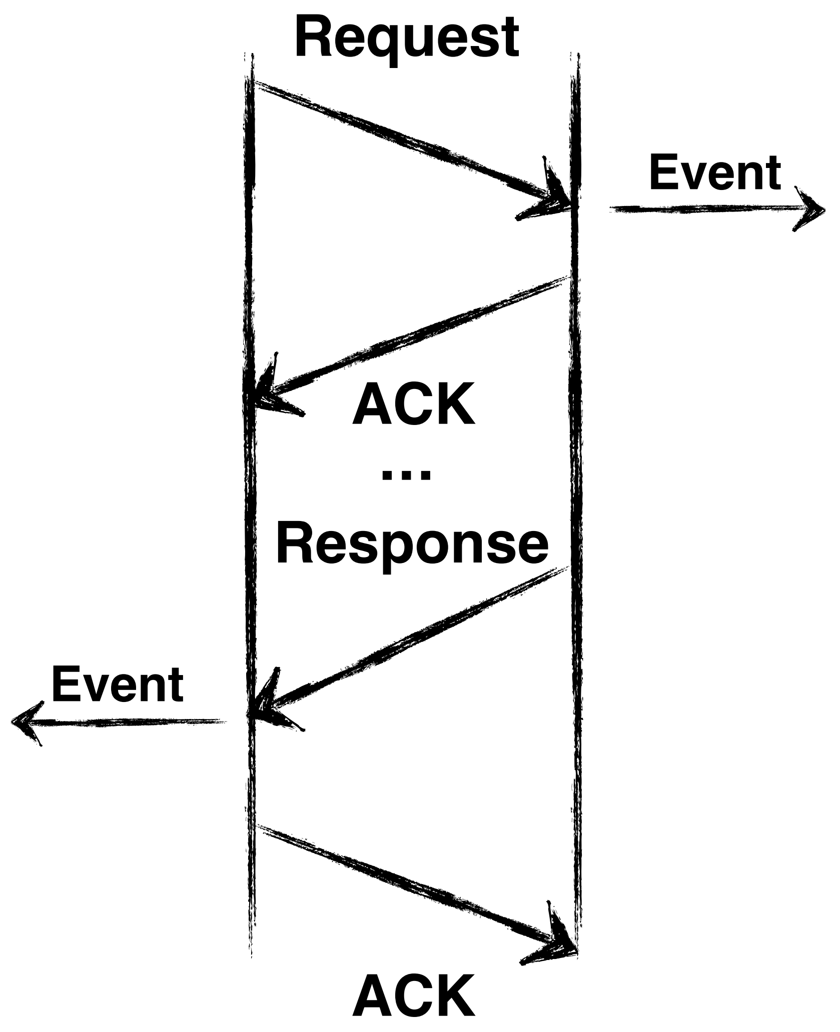
Not so long ago…

Web Services…

Web Services

http://en.wikipedia.org/wiki/List_of_web_service_specifications

No, *seriously*, lots of these!!

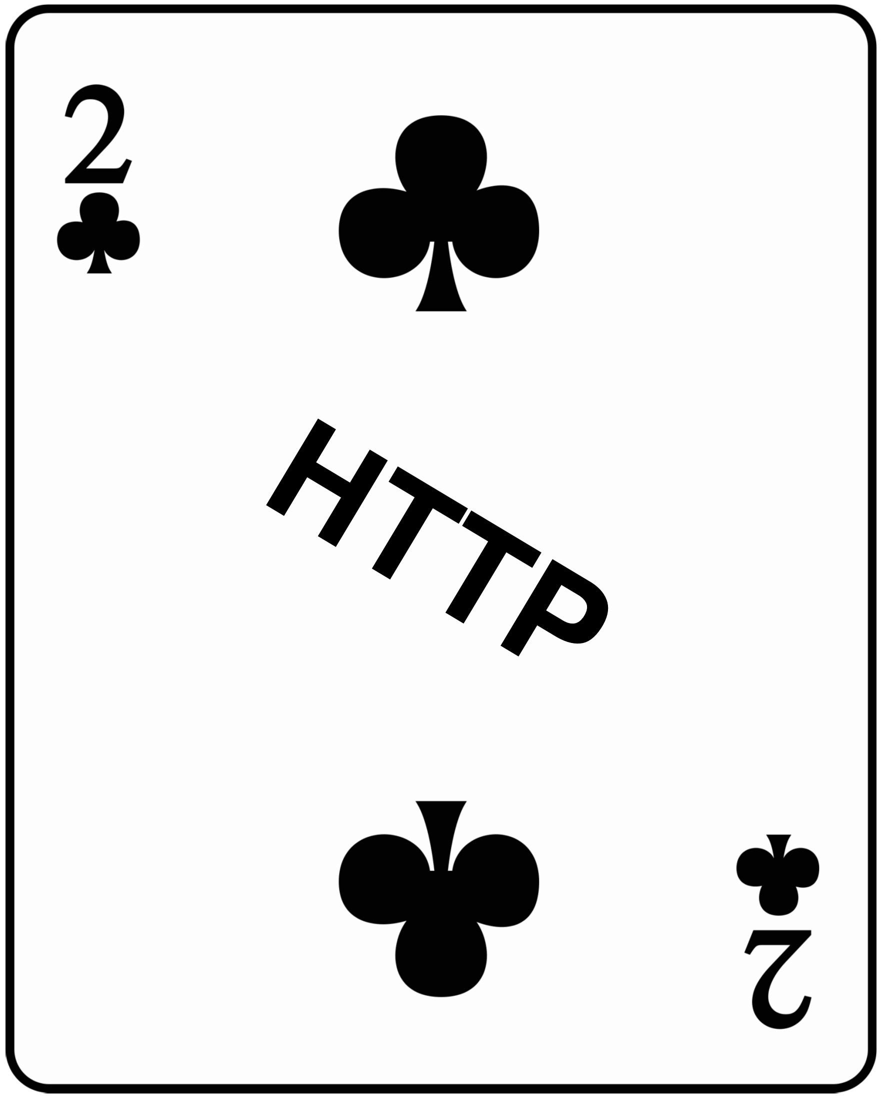


Sync Request

Sync Response

But… Async Request/Response… kinda

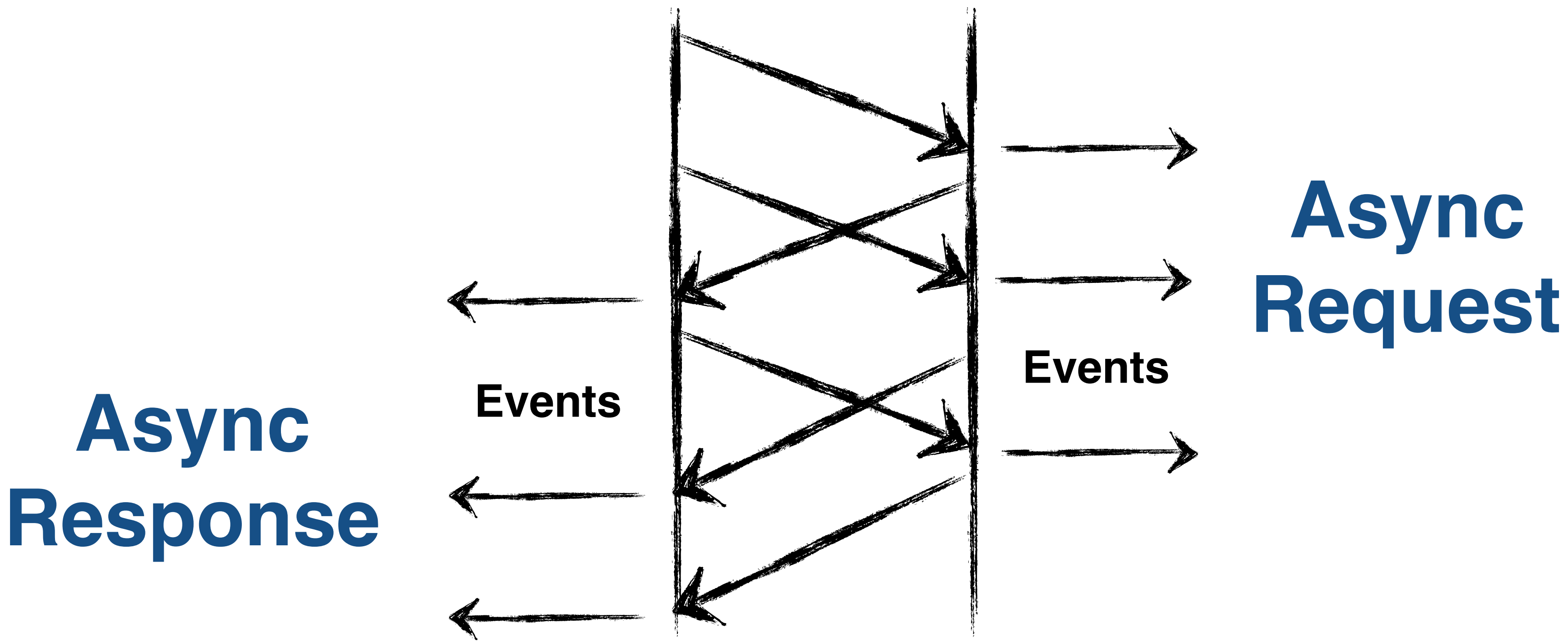
Then this happened


2
HTTP
2

There is emerging implementation experience and interest in a protocol that retains the semantics of HTTP without the legacy of HTTP/1.x message framing and syntax, which have been identified as hampering performance and encouraging misuse of the underlying transport.

— IETF httpbis Charter
http://datatracker.ietf.org/wg/httpbis/charter/

SPDY & HTTP/2
RFC 7540



Async Request/Response Streaming (Server Push)

https://tools.ietf.org/html/rfc7540

- ✓ ***Persistent Connection***
- ✓ ***Binary Encoding***
- ✓ ***Multiple Streams***
- ✓ ***Efficient Headers (HPACK)***
- ✓ ***Server Push***

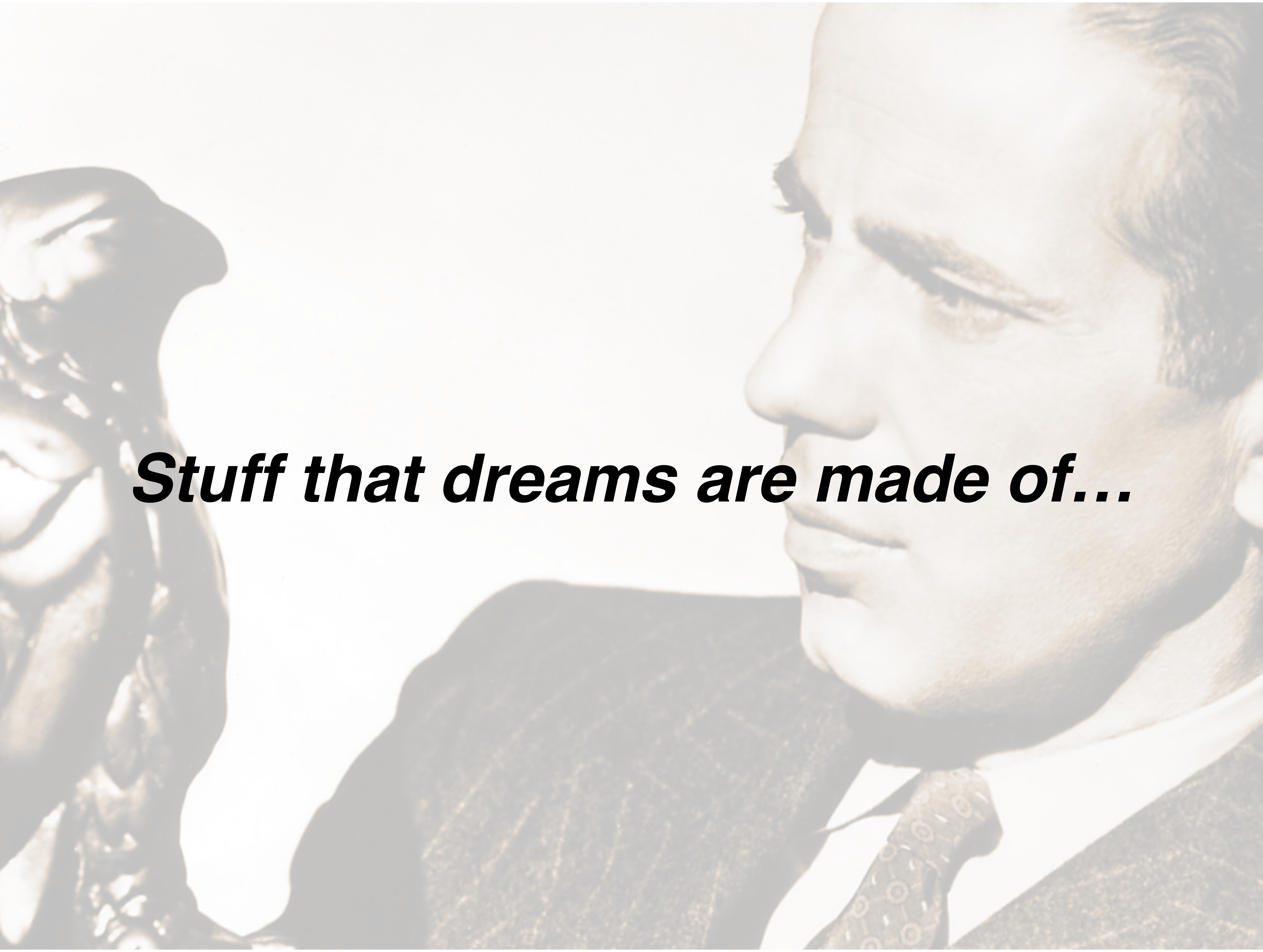
HTTP/2 & APIs

- ✓ ***Framing***
- ✓ ***Streams***
- ✓ ***Settings***

For APIs,

HTTP/2 = Interesting Times Ahead

Microservices



Stuff that dreams are made of…

Soo much to say, but

at the very core…

Moving Faster

- ✓ ***Service Independence***
- ✓ ***Fast Service Evolution***
- ✓ ***Service Isolation***
- ✓ ***Independent Deployability***

Component Decoupling

Asynchronous Binary Boundary

Serverless… Lambda

Protocols can and do couple

Protocol? Coupling?

Protocol Coupling

- ✓ ***Version Dependence***
- ✓ ***Response Dependence***
- ✓ ***Insufficient Encapsulation***
- ✓ ***3rd Party Service Dependence***
- ✓ ***Message Layout (Encoding)***

Message Layout

- ✓ ***Object Serialization or Not***
- ✓ ***To Schema or Not***
- ✓ ***Efficiency?***

- ✓ ***SBE (Simple Binary Encoding)***

An old argument

An old argument

The Ultra-Thick Client

vs

The Under-Over-Specified Protocol

But who really uses Microservices?

REALLY?

Responsiveness (Async UIs)

Constant Deployment + Versions (Execution)

Takeaways

- ✓ ***Many old (new) techniques***
- ✓ ***Many new technologies***
- ✓ ***Constant evolution***
- ✓ ***Being used to great effect***

Questions?

- *http://ietf.org/*
- *http://www.reactive-streams.org/*
- *http://reactivesocket.io/*
- *https://github.com/real-logic/Aeron*
- *https://github.com/real-logic/simple-binary-encoding*
- GitHub @tmontgomery
- Twitter @toddlmontgomery

Thank You!