

Evolving REST for an IoT World

Todd L. Montgomery
@toddlmontgomery

New York 2014

Tutorials: June 9-10
Conference: June 11-13

QCon

INTERNATIONAL
SOFTWARE DEVELOPMENT
CONFERENCE

www.qconnewyork.com



Representational State Transfer

pro·to·col *noun* \ˈprō-tə-,kəl, -,kōl, -,käl, -kəl\

...

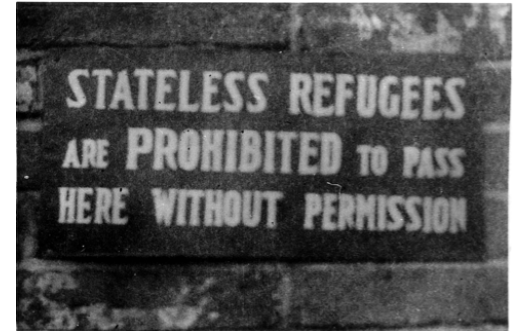
3 b : a set of conventions governing the *treatment* and especially the *formatting* of data in an electronic communications system <network *protocols*>

...

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*>



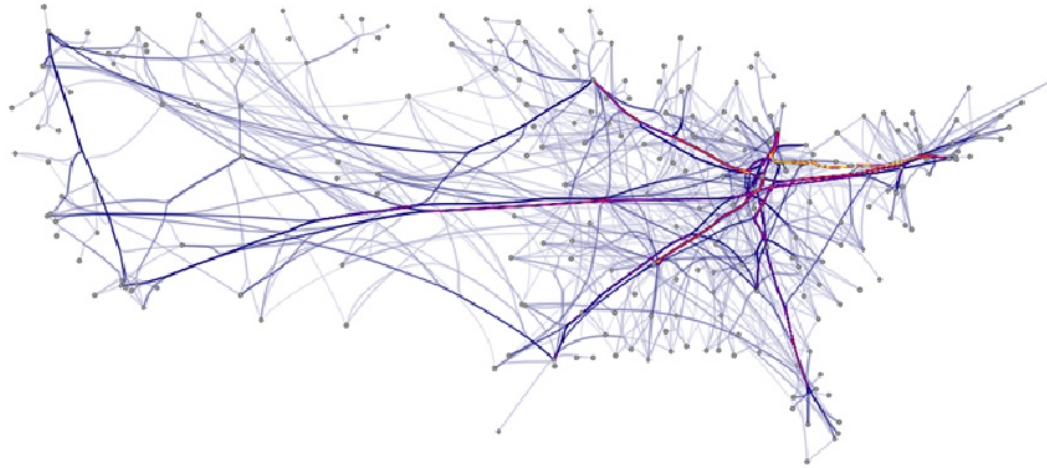
Client - Server



Stateless

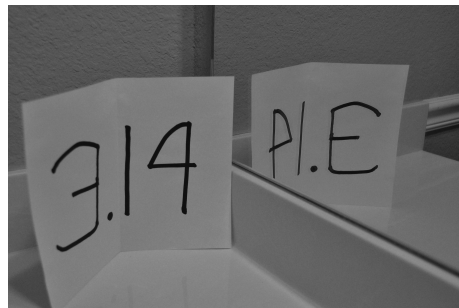


Cacheable



Uniform Interface
Hypermedia, Resources,
URIs

Layered

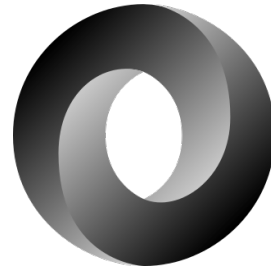


Hmmm...

REST Ecosystem

grows those URLs
CURL

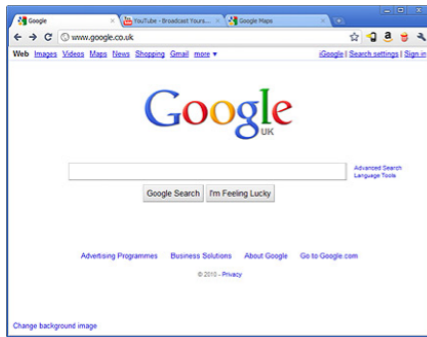
Tools - CLI



JSON



HTTP/1.1, TCP,
[TLS/SSL], IP



Browser

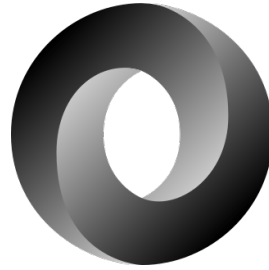
**Fast, Easy
Integration**



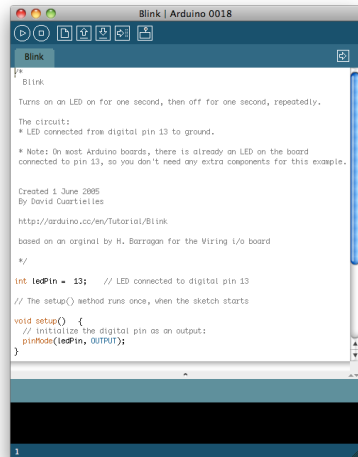
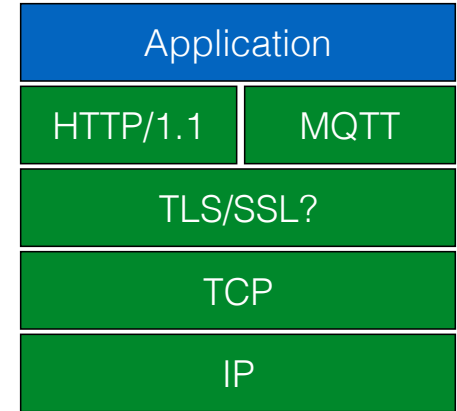
IoT/loE Ecosystem



Boards & Kits

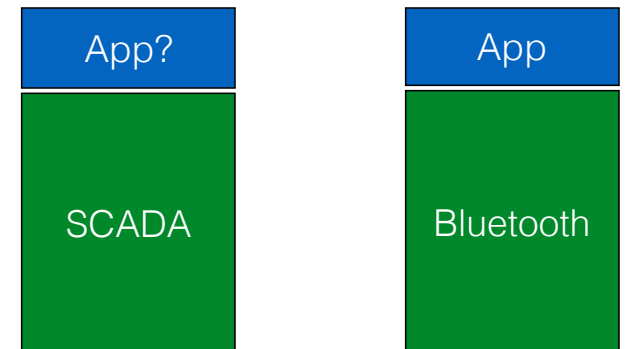


JSON
??



Environments

**Evolving
Rapidly**



Multiple
Stacks

Publish/Subscribe

Request/Response

Request/Response

Communication Patterns

Streaming

"Ingest"

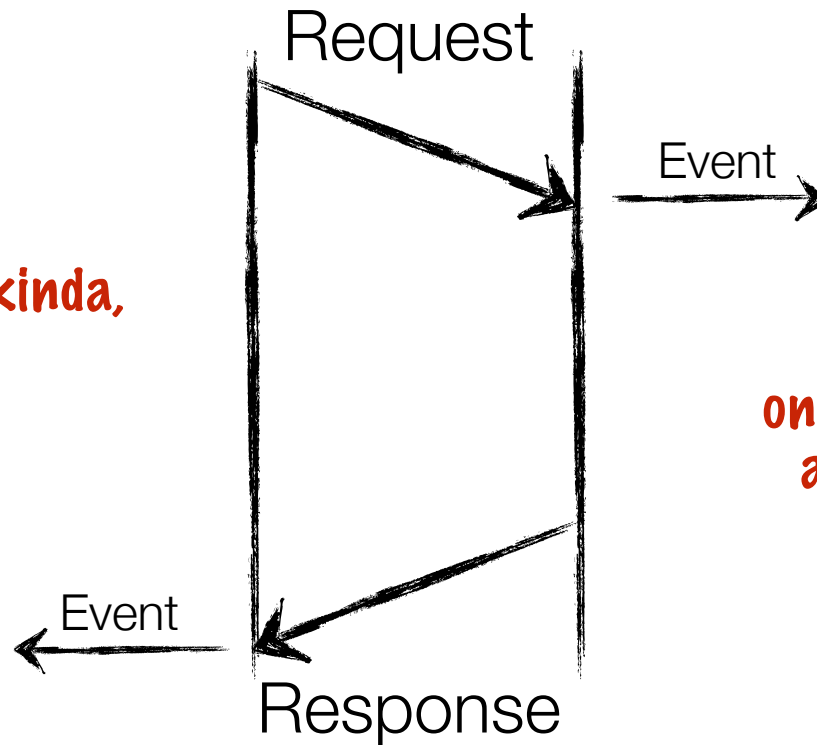
History & Evolution

HTTP

RFC 2068, 2616, ..., 7230-7240

June 2014

**Bi-Directional... kinda,
but...**



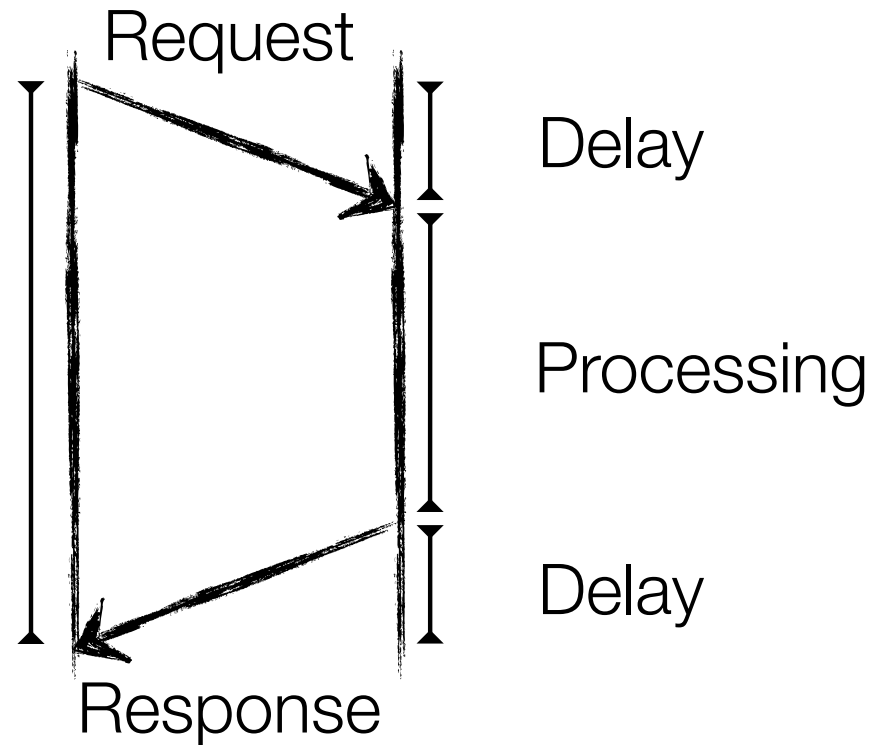
**... only
one direction
at-a-time**

**Synchronous
Request/Response**

HTTP

**What happens
here while
waiting?**

...Nothing...



Stop-and-Wait

Head-Of-Line Blocking



Latency Sensitivity



Mobile

“OK” Bandwidth + Long RTT + High Loss Rate + No Effective HTTP Pipelining



Truly Awful User Experiences

Asynchronous Request / Response

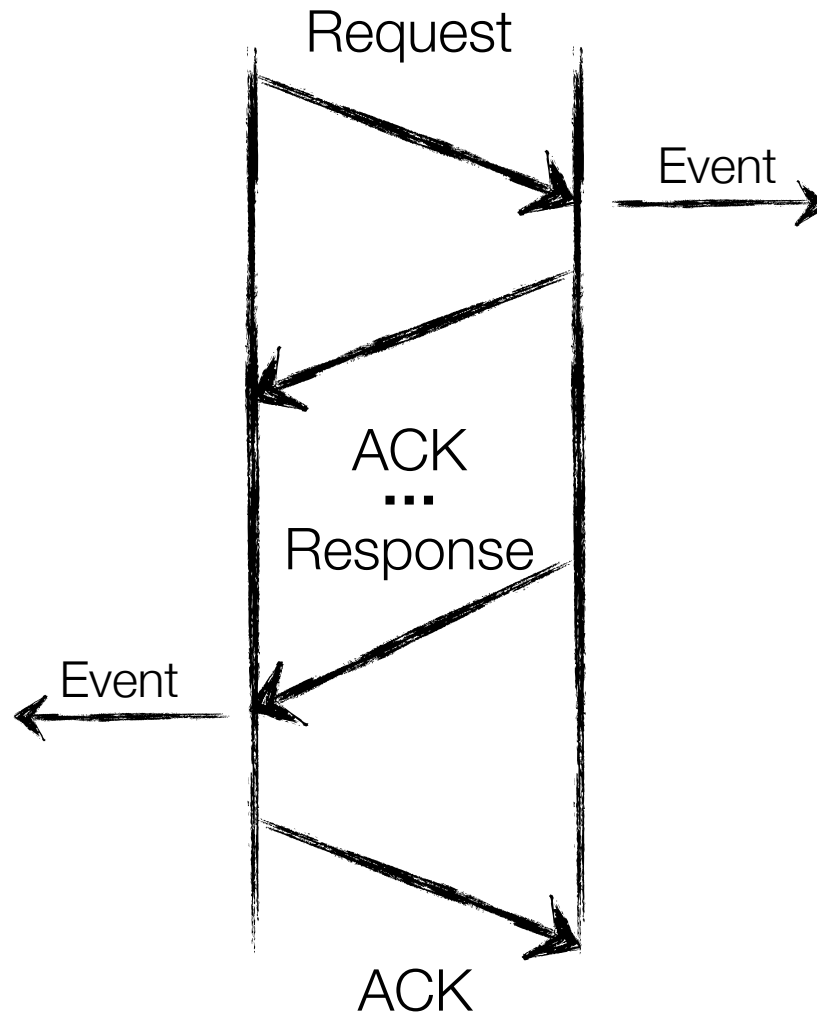
Unlock More Reactive Patterns!

Web Services

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

No, *seriously*,
lots of these!!

**Sync
Response**



**Sync
Request**

But... Async Request/Response... kinda

Thankfully, Locked within the
Enterprise...

Mostly...

*Just because you **could** use
HTTP, doesn't mean you **should**...*

*“Yeah, yeah, but your scientists
were so preoccupied with whether
or not they **could** that they didn't
stop to think if they **should**.”*

— Jurassic Park

Philosophy of some REST APIs

HTCPCP

RFC 2324, Extended by RFC 7168



"there is a strong, dark, rich requirement for a protocol designed espressoly [sic] for the brewing of coffee"

[\[Docs\]](#) [\[txt|pdf\]](#) [\[Errata\]](#)

Updated by: [7168](#)

Network Working Group
Request for Comments: 2324
Category: Informational

INFORMATIONAL

Errata Exist

L. Masinter

1 April 1998

Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0)

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document describes HTCPCP, a protocol for controlling, monitoring, and diagnosing coffee pots.

1. Rationale and Scope



BREW

WHEN

418 I'm a teapot

*"This has a serious purpose – it identifies many of the ways in which HTTP has been extended inappropriately."
— Larry Masinter, author
<http://larry.masinter.net/>*

Tooling

Addressing

Easy firewall traversal

Why is HTTP used?

Works with Anything

Simple, Flexible, Familiar

Publish/Subscribe

Request/Response

Request/Response

Communication Patterns

Streaming

"Ingest"

**Security
(Challenge)**

**User State
Query**



Request



Response



**Keep-Alive
or Watchdog**

**Support
(UI/Device)**

Persistent connections help a LOT!

Well designed protocols help a LOT MORE!

Stay out of High
Energy State!

Battery Life

Many simultaneous connections hurt!

Using the wrong protocol with the wrong pattern hurts A LOT!

The Wrong Patterns Hurt a LOT!

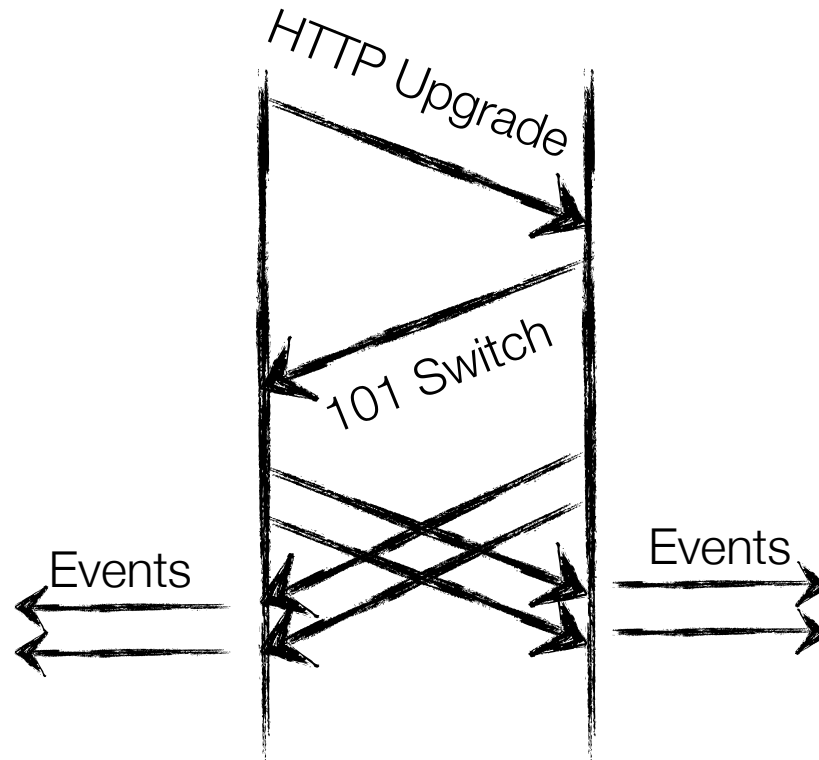
New Protocols & Standards

WebSocket RFC 6455

**Really a
Transport
Protocol**

**Async
Request/
Response**

Streaming

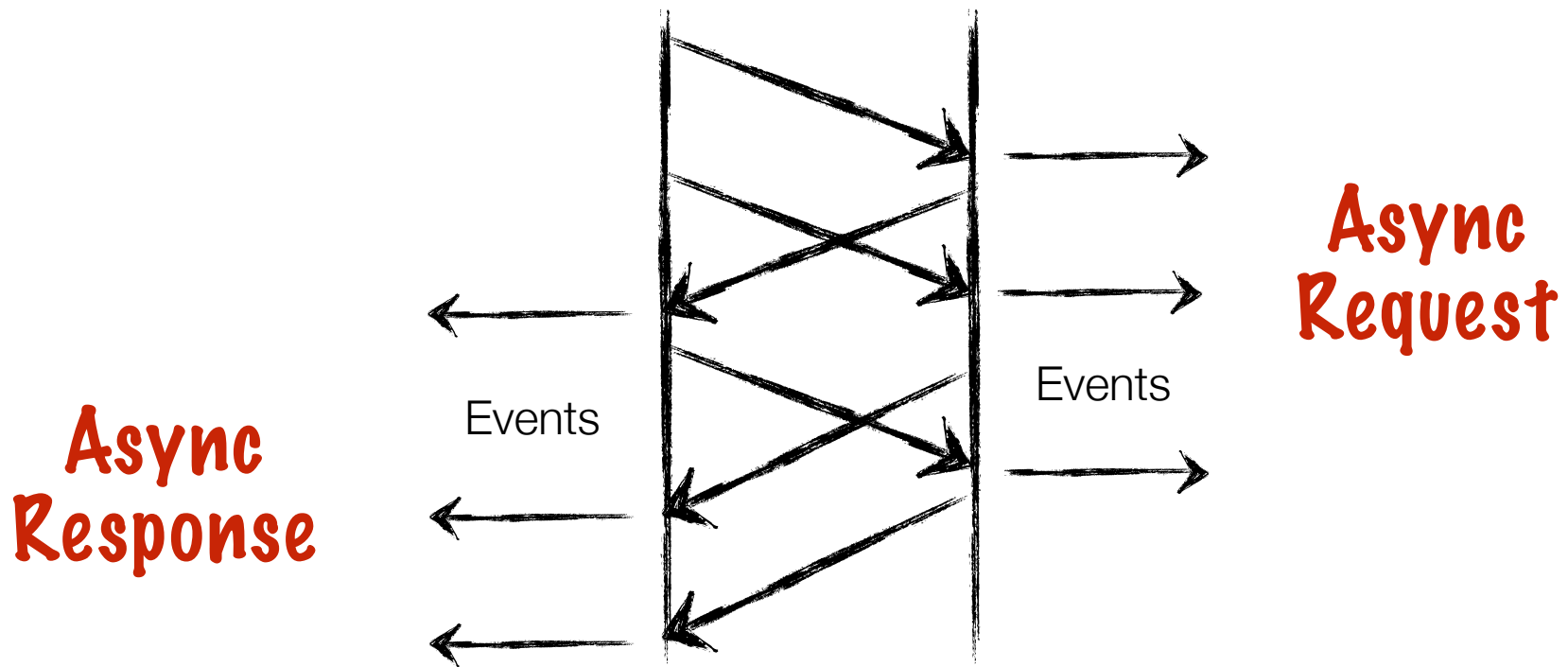


Ingest

**Full Duplex, Asynchronous
"TCP over the Web"**

SPDY & HTTP/2

IETF Drafts



Async Request/Response
Multiple Streams
Efficient Headers (HPACK)
Binary Encoding

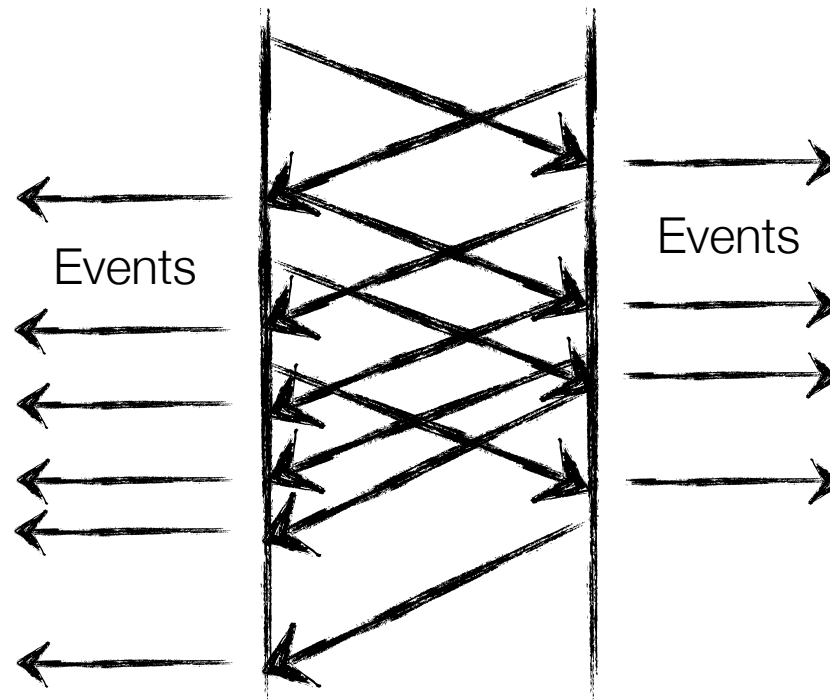
WebSocket over HTTP/2 IETF Draft

**Async
Response**

**Async
Request**

Streaming

Ingest



**Full Duplex, Asynchronous
with Multiple Channels/Streams**

Broker-Based

**Runs over TCP
or WebSocket (v3.1.1)**

MQ Telemetry Transport (MQTT)

OASIS Standard

MQTT-SN for non-TCP/IP

**Lightweight
Publish/Subscribe
Messaging Transport**

**Standardized HTTP
Mapping**

**Runs over UDP, DTLS,
or WebSocket**

Constrained Application Protocol (CoAP)

IETF CoRE WG (Constrained RESTful Environments)

**Resource Discovery,
Linking, etc.**

**Request/Response
(either direction),
Publish/Subscribe**

Sustain REST Principles

Easily Parsed

Standards-Based

Requirements

Easy to Implement

Flexible - Easily Extended

Efficient Handling of Data/Metadata

HTTP/2

WebSocket + MQTT

Possible Game Plan(s)

Combining IoT & REST

WebSocket + CoAP

WebSocket + HPACK

**Nothing Optional,
TLS, HPACK, etc.**

**More complex
than HTTP/1.1**

HTTP/2

Familiar Primitives

**Ecosystems:
REST Yes,
IoT No**

**HTTP Mapping?
WebSocket can adapt**

**Ecosystems:
IoT Yes,
REST No (w/o WS)**

WebSocket + MQTT

**Some Guaranteed
Messaging Semantics**

**Enables Many
Patterns**

**HPACK handles
method + headers**

**Not a Standard,
but made of
Standards**

WebSocket + HPACK

**Use header for
Stream ID**

**HPACK is
(subjectively)
complex**

HTTP Mapping

**Not Broker-based,
Peer-to-Peer**

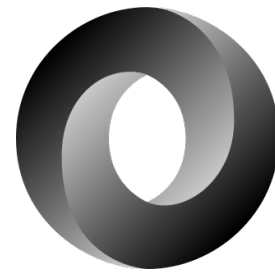
WebSocket + CoAP

**Ecosystems:
REST Yes,
IoT Yes**

**No
Guaranteed
Messaging**

One More Thing...

JSON





Thing 1

Not a human

Does not need to be
human readable



Thing 2

Also, ...not a human



Binary Encoding

Concise Binary Object Representation (COBR)

FIX / Simple Binary Encoding (SBE)

HPACK (Part of HTTP/2)

<http://tools.ietf.org/html/rfc7049>

<https://github.com/real-logic/simple-binary-encoding>

Questions?

- Kaazing <http://www.kaazing.com>
- Slideshare <http://www.slideshare.com/toddleemontgomery>
- Twitter @toddlmontgomery

Thank You!