goto; copenhagen

GOTO Copenhagen 2019
Conference Nov. 18 - 20

HTTP/3 is next generation HTTP
Is it QUIC enough?

Daniel Stenberg

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2019
Conference Nov. 18 - 20

Click ‘Rate Session’ to rate session and ask questions.

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2019
Conference Nov. 18 - 20

Please

Remember to rate this session

Thank you!

Follow us @gotocph

goto; copenhagen

GOTO Copenhagen 2019
Conference Nov. 18 - 20

Did you **remember to rate** the previous session ?

Follow us @gotocph

Daniel Stenberg

https://daniel.haxx.se @bagder

curl://

Daniel Stenberg

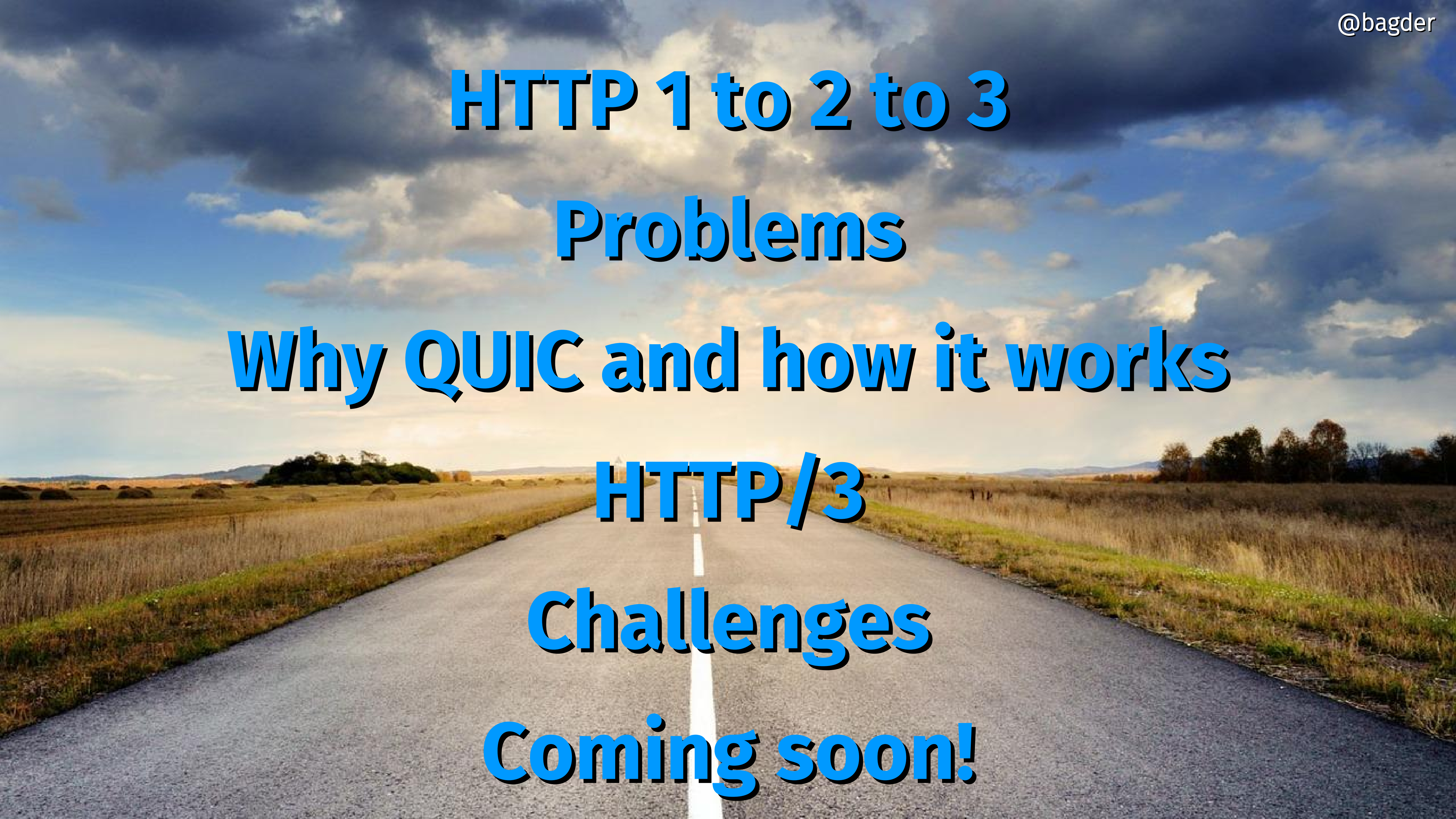
@bagder

wolfSSL

Daniel Stenberg

@bagder

I E T F®



HTTP 1 to 2 to 3

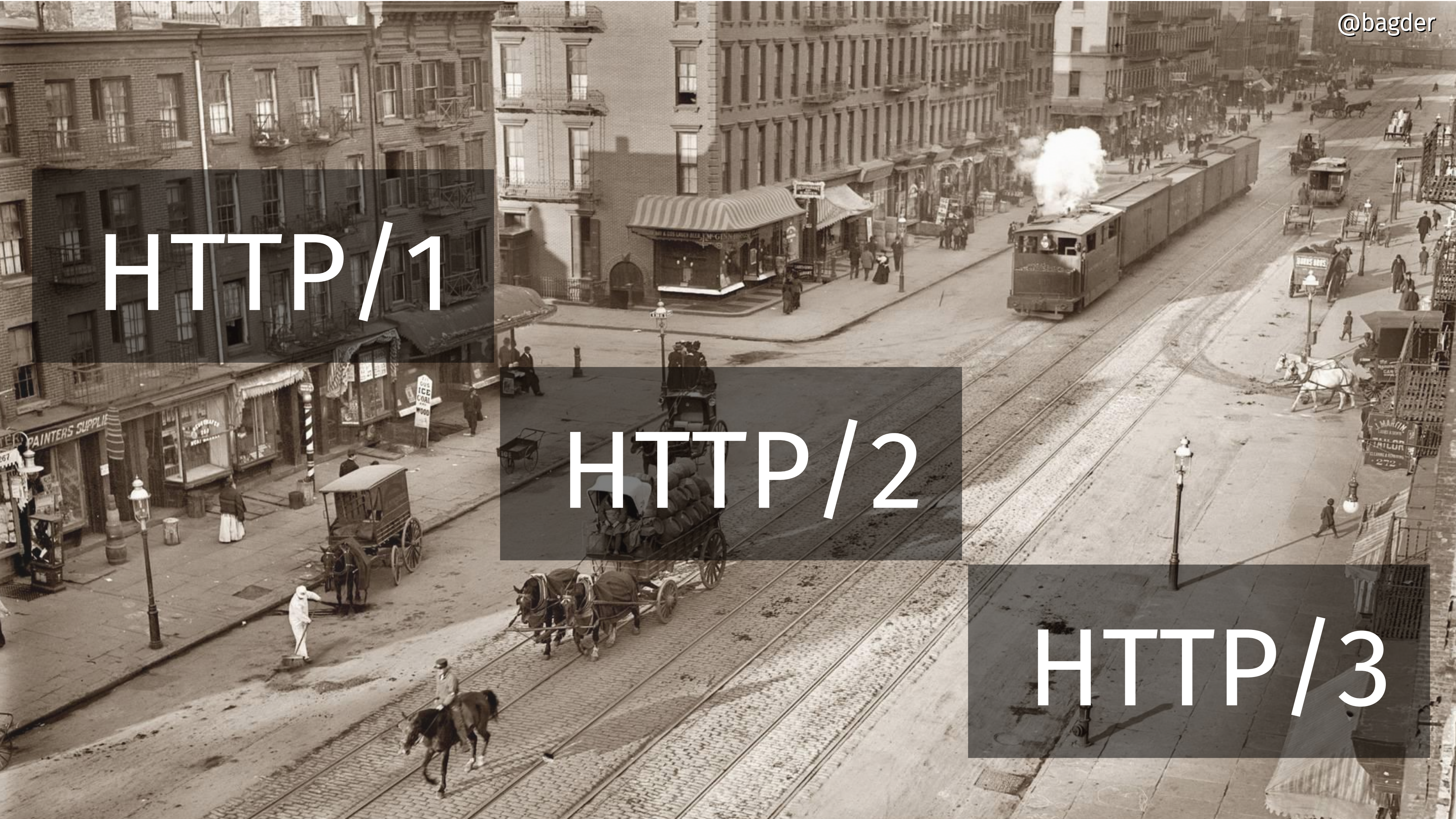
Problems

Why QUIC and how it works

HTTP/3

Challenges

Coming soon!



HTTP/1

HTTP/2

HTTP/3

Under the hood

```
GET / HTTP/1.1
Host: www.example.com
Accept: */*
User-Agent: HTTP-eats-the-world/2019
```

```
HTTP/1.1 200 OK
Date: Thu, 09 Nov 2018 14:49:00 GMT
Server: my-favorite v3
Last-Modified: Tue, 13 Jun 2000 12:10:00 GMT
Content-Length: 12345
Set-Cookie: this-is-simple=yeah-really;
Content-Type: text/html

[content]
```

HTTP started done over TCP

TCP

TCP/IP works over IP

Establishes a “connection”

3-way handshake

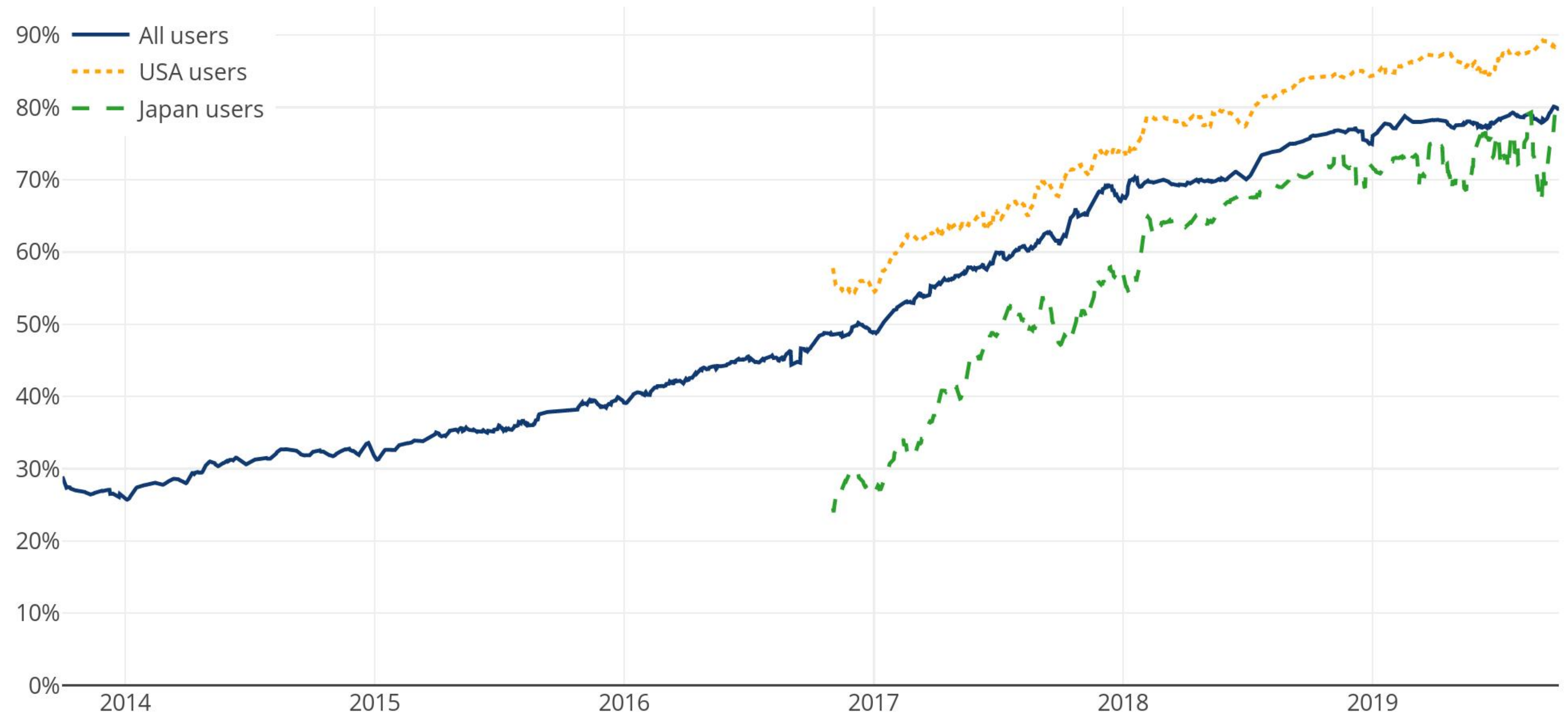
Resends lost packages

Delivers a byte stream

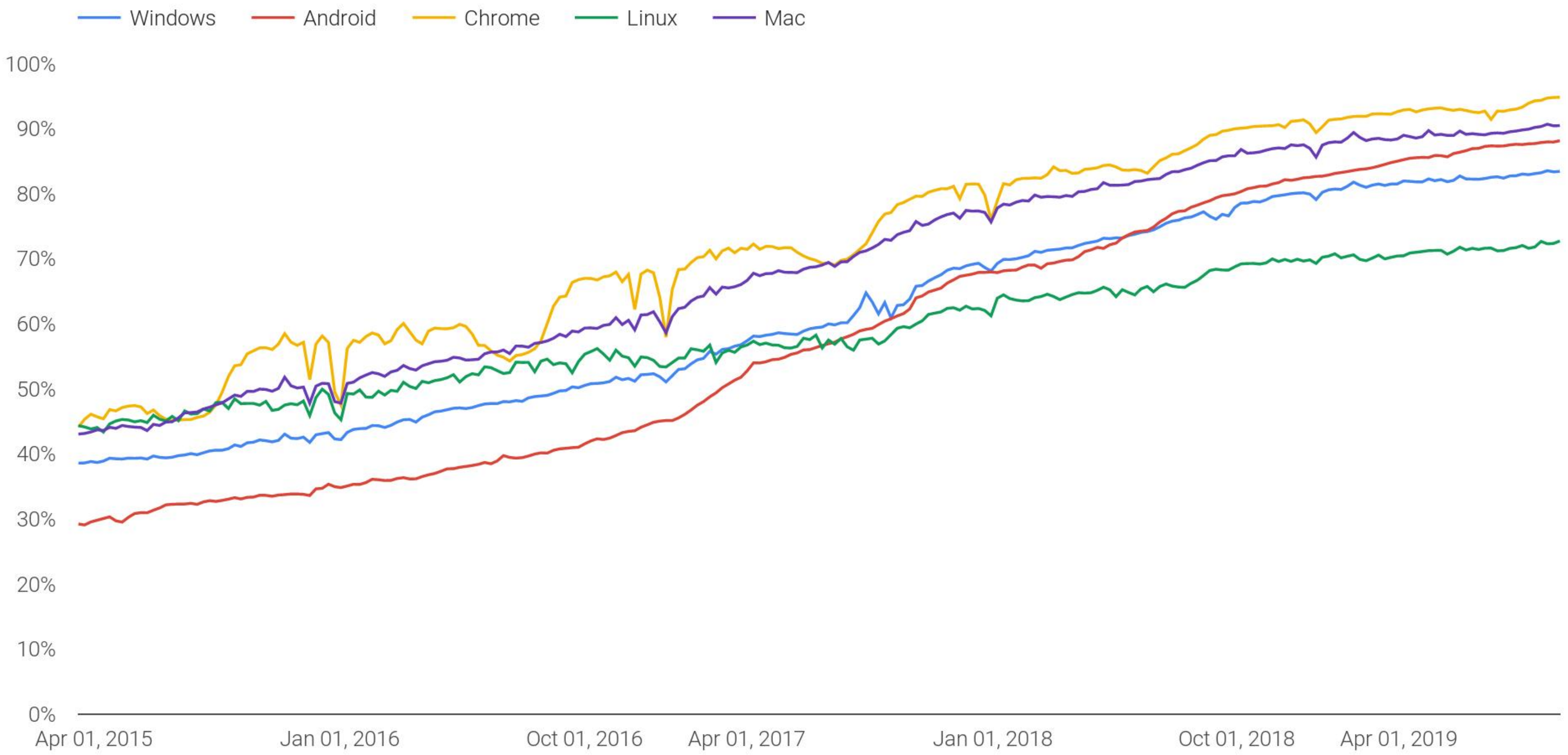
Clear text

HTTPS means TCP + TLS + HTTP

Percentage of Web Pages Loaded by Firefox Using HTTPS



Percentage of pages loaded over HTTPS in Chrome by platform



TLS

TLS is done over TCP for HTTP/1 or 2

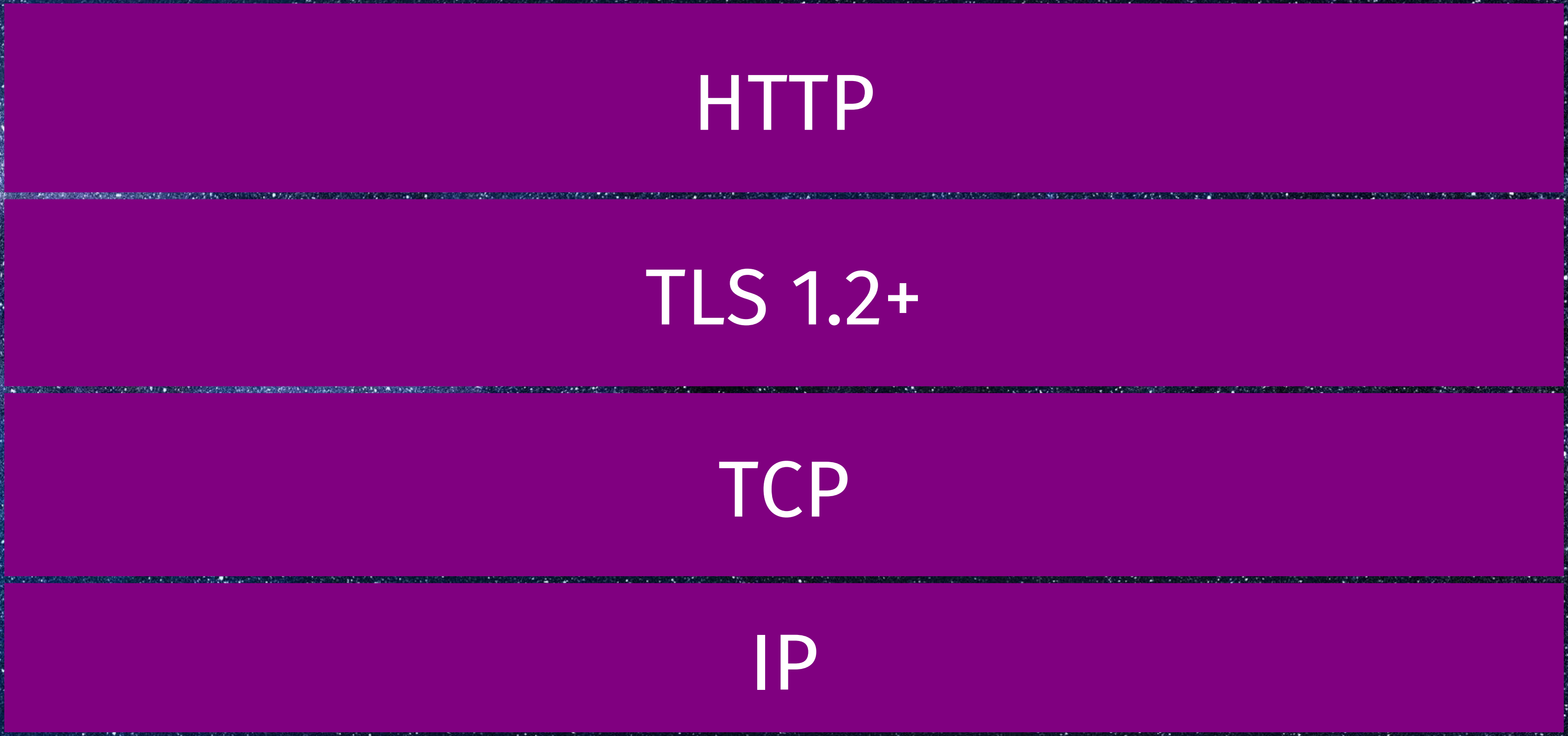
Transport Layer Security

Additional handshake

Privacy and security

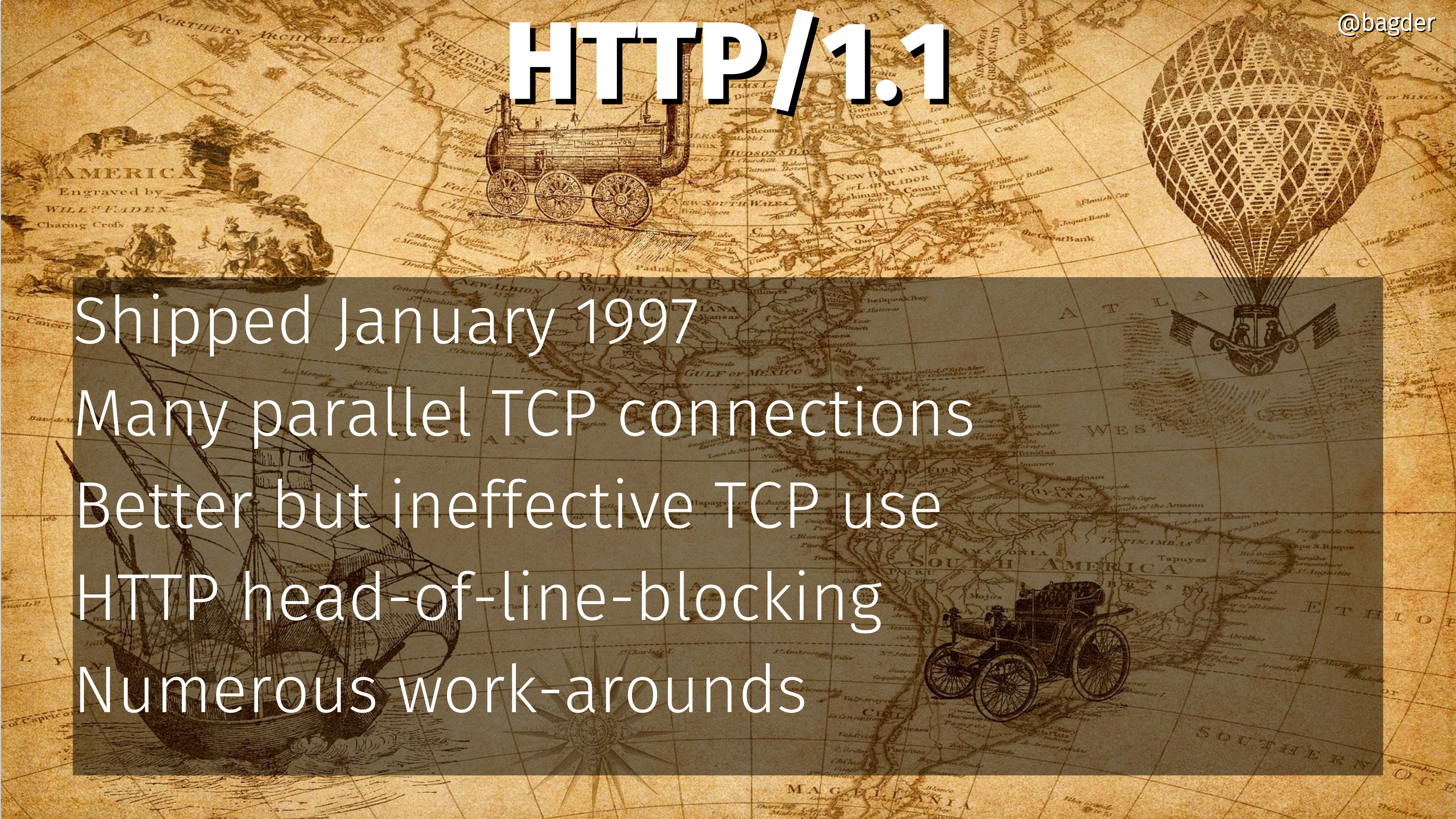
@bagder

Classic HTTP Network Stack



HTTP over TCP

HTTP/1.1



Shipped January 1997

Many parallel TCP connections

Better but ineffective TCP use

HTTP head-of-line-blocking

Numerous work-arounds

HTTP/2

Shipped May 2015

Uses single connection per host

Many parallel *streams*

TCP head-of-line-blocking

Ossification

Internet is full of boxes

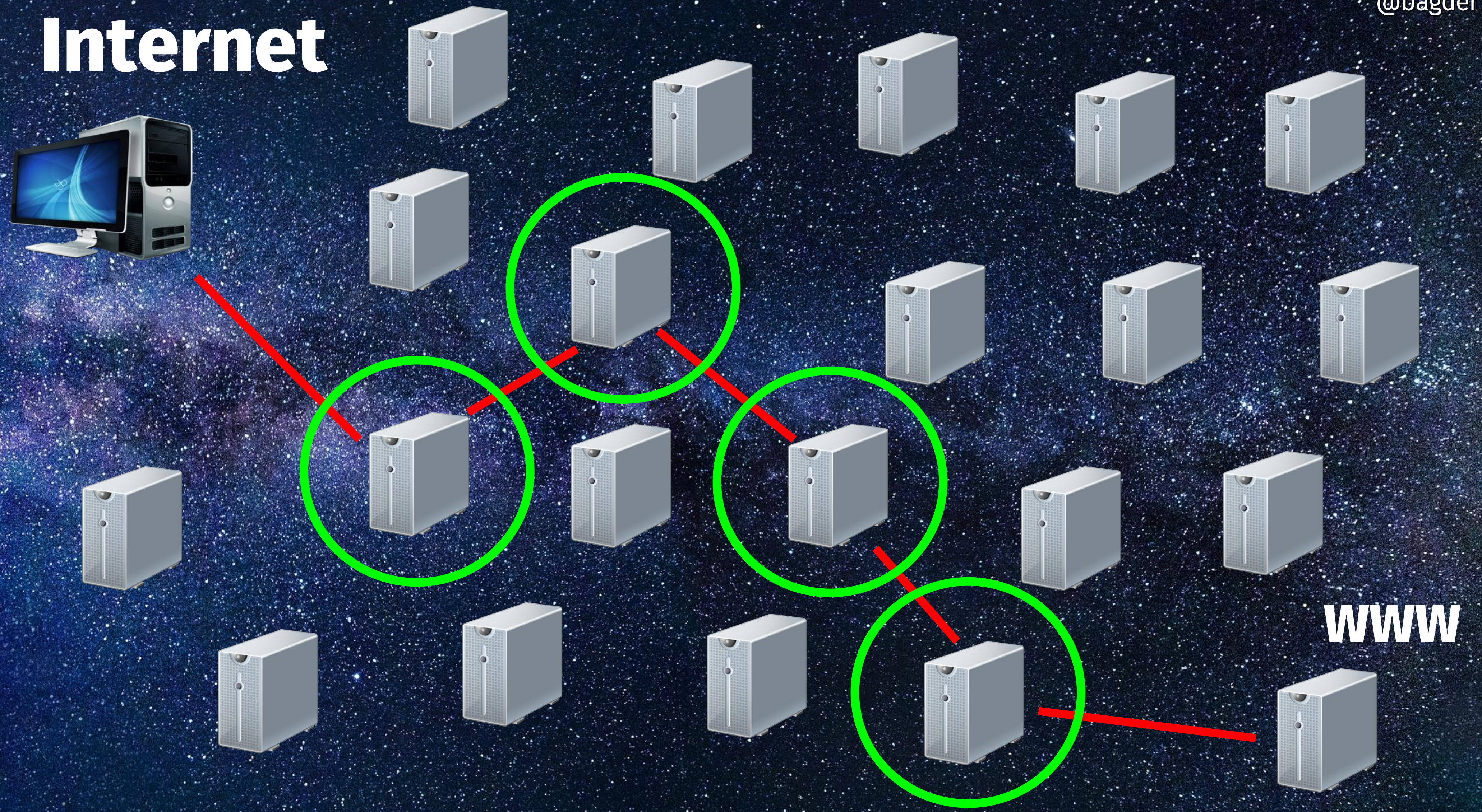
Routers, gateways, firewalls, load balancers, NATs...

Boxes run software to handle network data

Middle-boxes work on existing protocols

Upgrade much slower than edges

Internet



Ossification casualties

HTTP/2 in clear text

TCP improvements like TFO

TCP/UDP replacements

HTTP brotli

Future innovations

... unless encrypted

Improvement in spite of ossification

QUIC

@bagder

f5®

LITESPEED

fastly

Microsoft

traffic server™

Akamai

moz://a

Google

CLOUDFLARE®

A new transport protocol

Built on experiences by Google QUIC

Google deployed “http2 frames over UDP”-QUIC in 2013

Widely used client

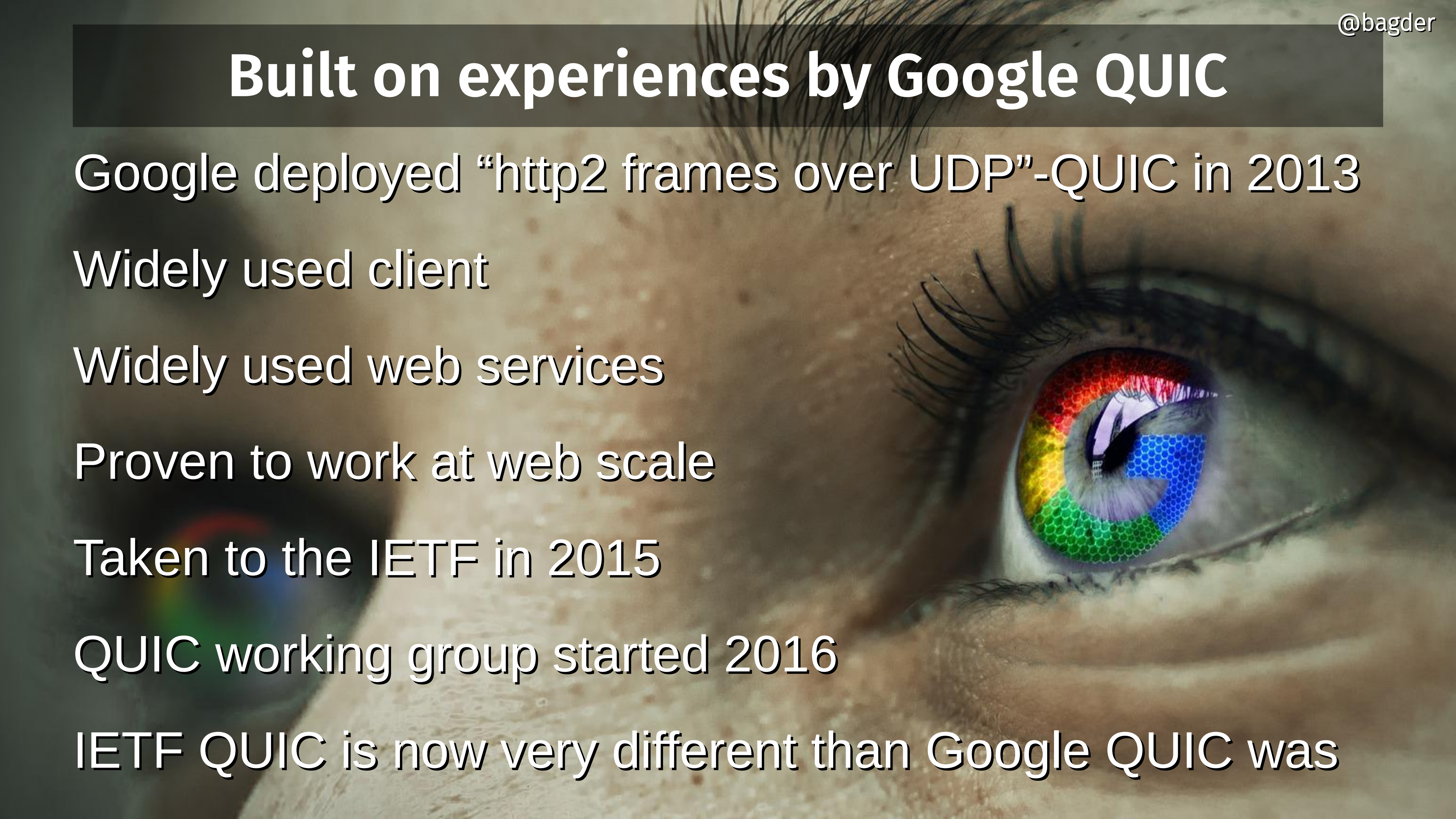
Widely used web services

Proven to work at web scale

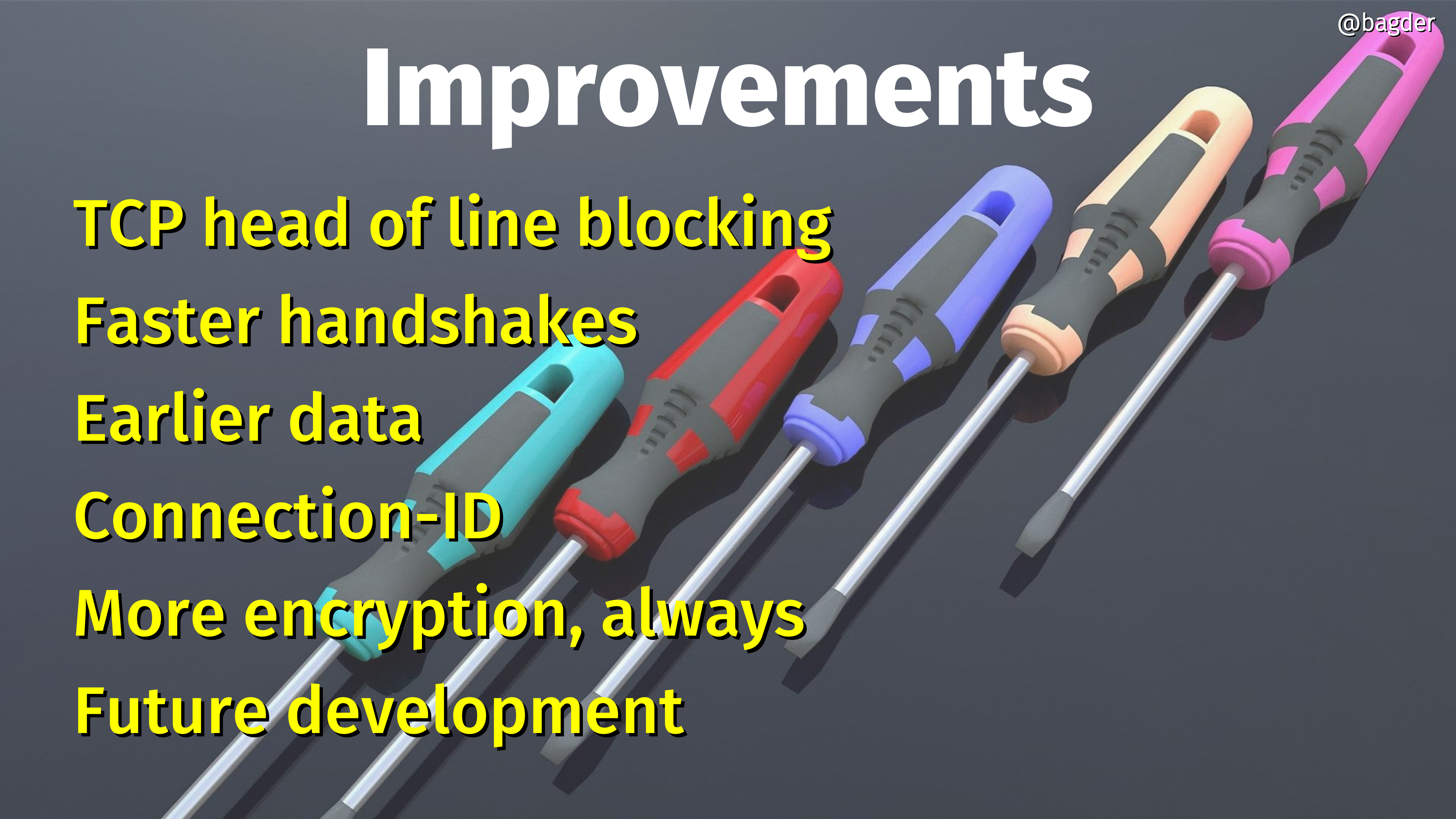
Taken to the IETF in 2015

QUIC working group started 2016

IETF QUIC is now very different than Google QUIC was



Improvements



TCP head of line blocking

Faster handshakes

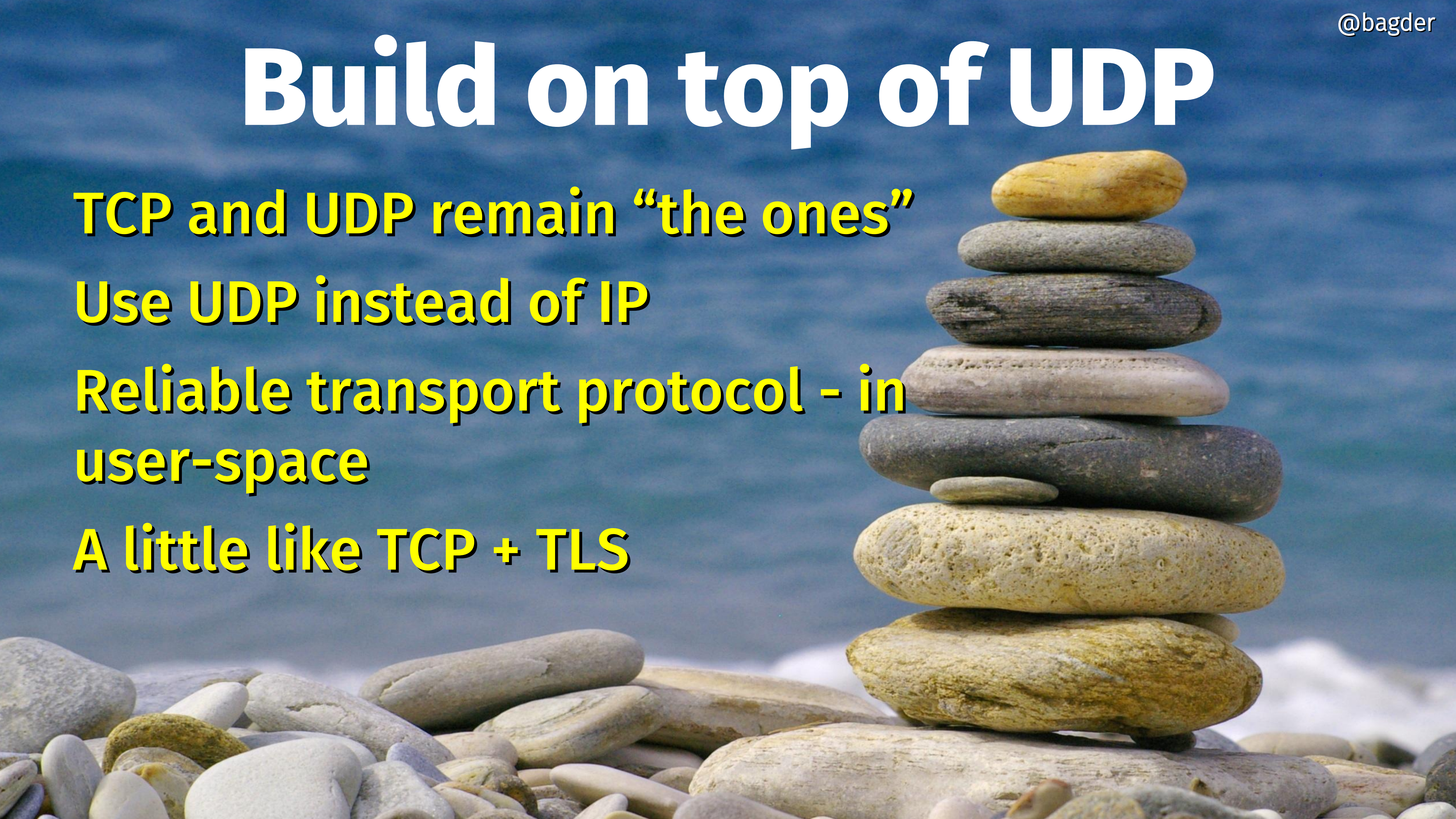
Earlier data

Connection-ID

More encryption, always

Future development

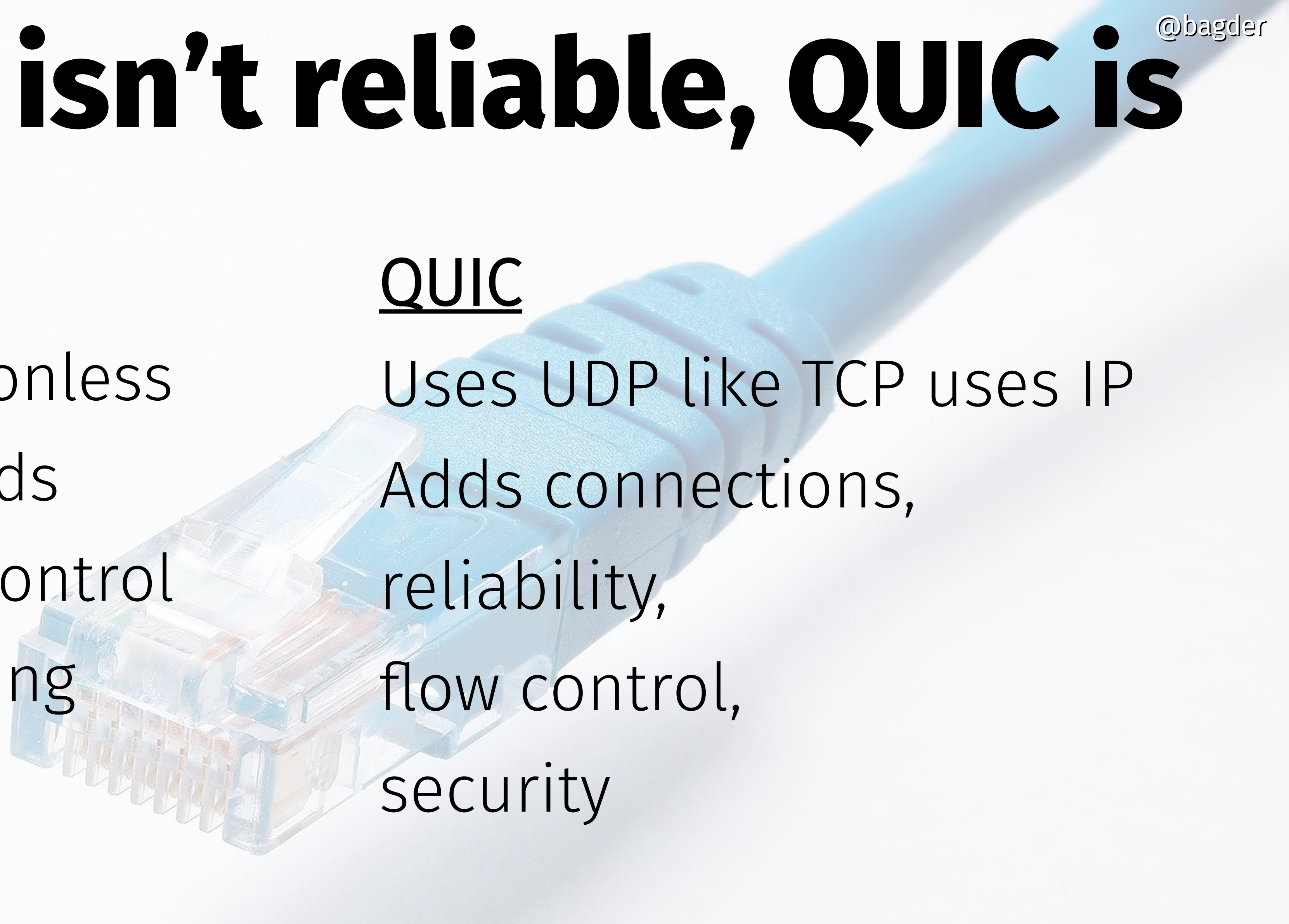
Build on top of UDP



- TCP and UDP remain "the ones"
- Use UDP instead of IP
- Reliable transport protocol - in user-space
- A little like TCP + TLS

@bagder

UDP isn't reliable, QUIC is



UDP

Connectionless

No resends

No flow control

No ordering

QUIC

Uses UDP like TCP uses IP

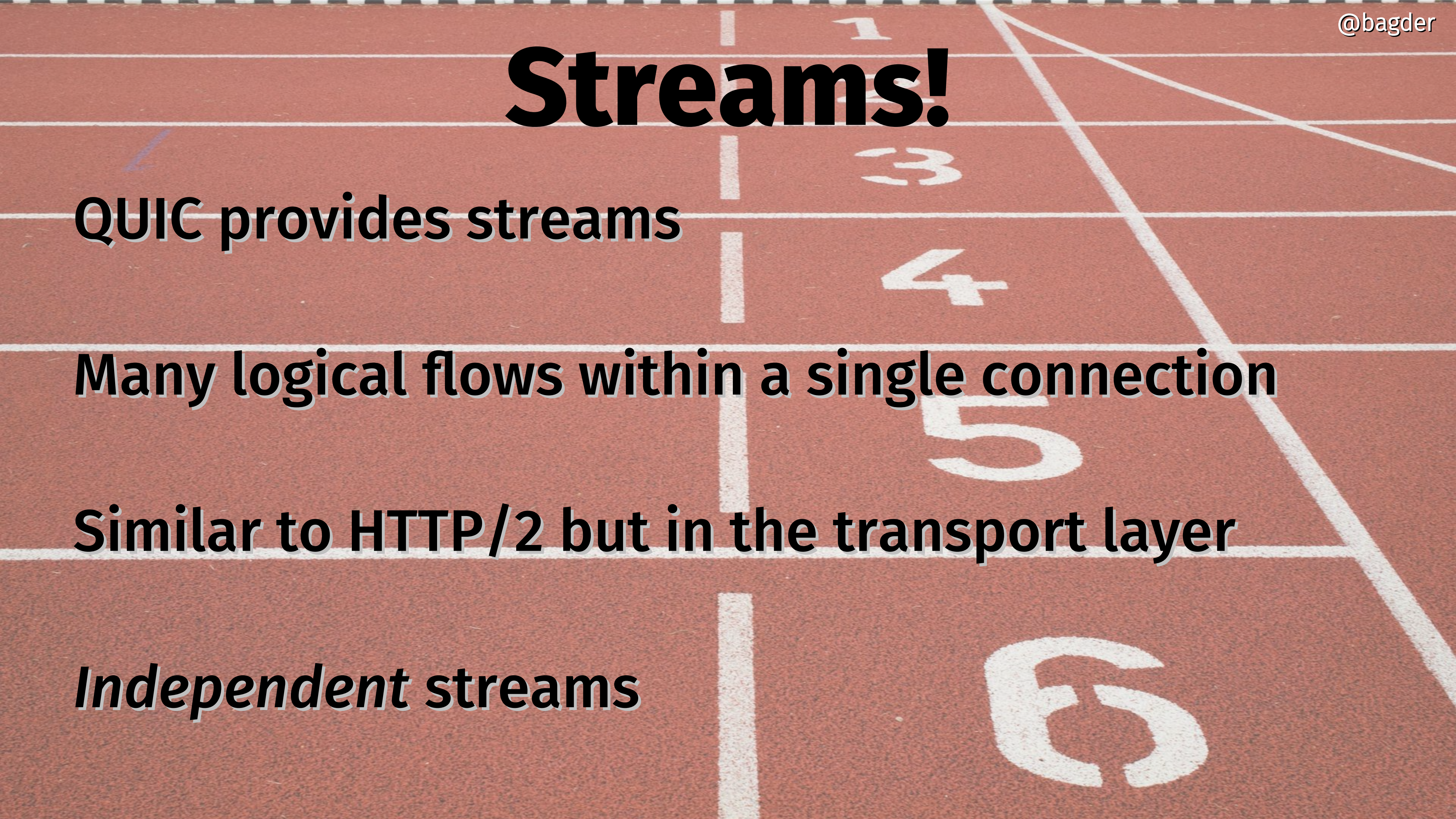
Adds connections,

reliability,

flow control,

security

@bagder



Streams!

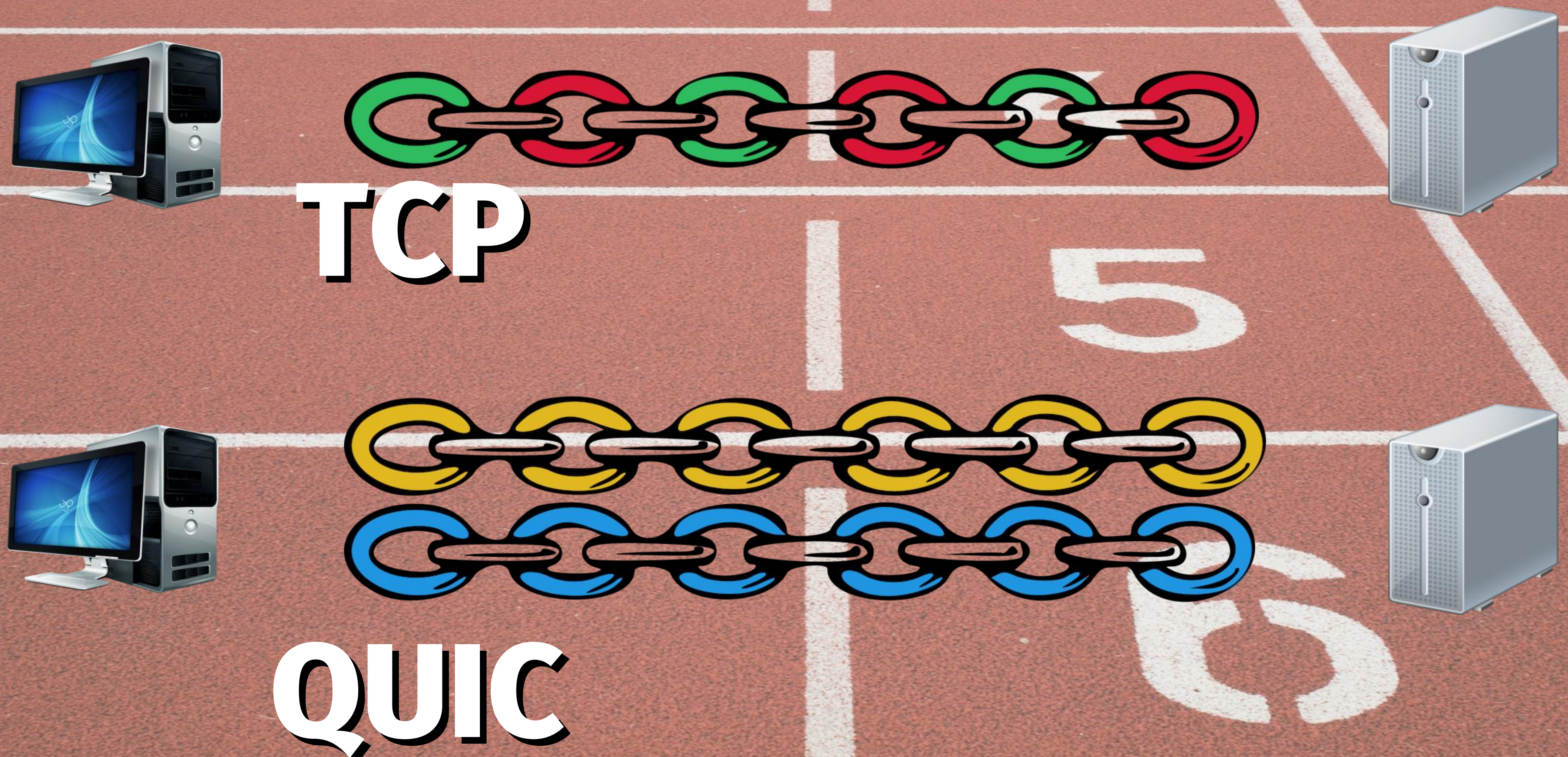
QUIC provides streams

Many logical flows within a single connection

Similar to HTTP/2 but in the transport layer

Independent streams

Independent streams



Application protocols over QUIC

Streams for free

Could be “any protocol”

HTTP worked on as the first

Others are planned to follow

HTTP/3 = HTTP over QUIC

HTTP – same but different

Request

- method + path
- headers
- body

Response

- response code
- headers
- body

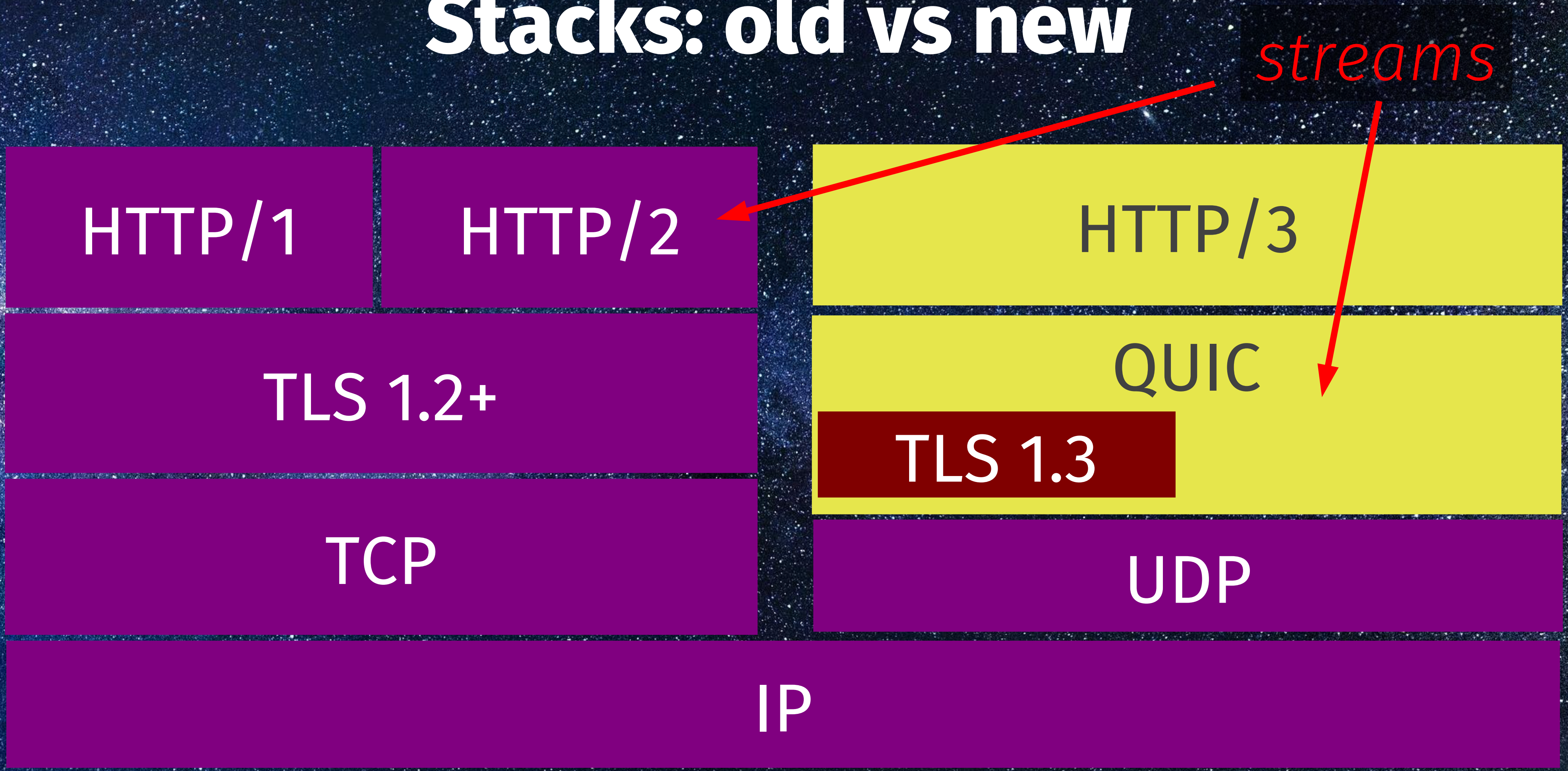
HTTP – same but different

HTTP/1 – in ASCII over TCP

HTTP/2 – binary multiplexed over TCP

HTTP/3 – binary over multiplexed QUIC

Stacks: old vs new



@bagder

HTTP feature comparison

	HTTP/2	HTTP/3
Transport	TCP	QUIC
Streams	HTTP/2	QUIC
Clear-text version	Yes	No
Independent streams	No	Yes
Header compression	HPACK	QPACK
Server push	Yes	Yes
Early data	In theory	Yes
0-RTT Handshake	No	Yes
Prioritization	Messy	Changes

HTTP/3 is faster

(Thanks to QUIC)

Faster handshakes

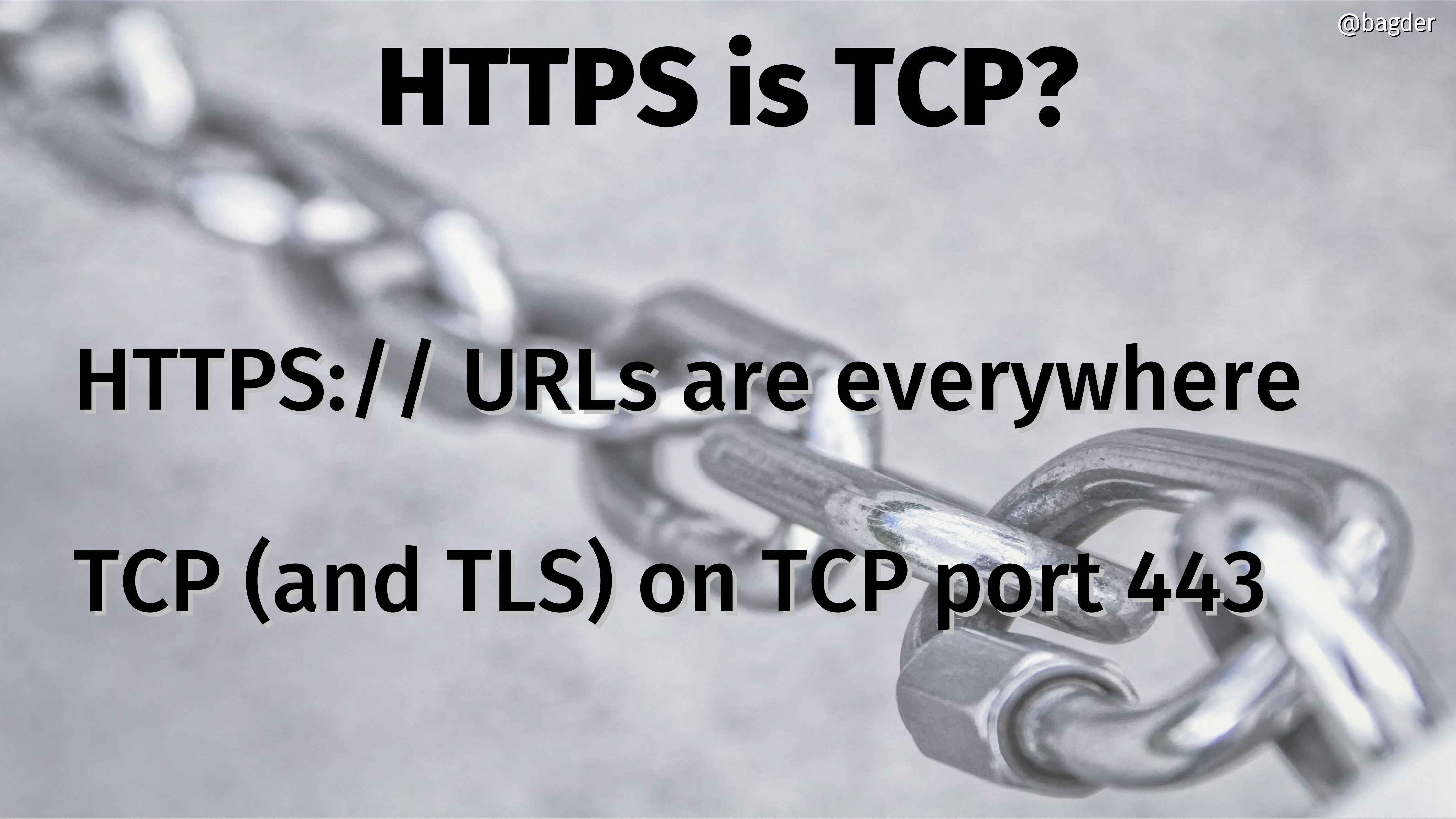
Early data that works

The independent streams

By how much remains to be measured!

@bagder

HTTPS is TCP?



HTTPS:// URLs are everywhere

TCP (and TLS) on TCP port 443

This service - over there!

The `Alt-Svc:` response header

Another host, protocol or port number is the same "origin"

This site also runs on HTTP/3 "over there", for the next NNNN seconds

Race connection attempts?

Might be faster

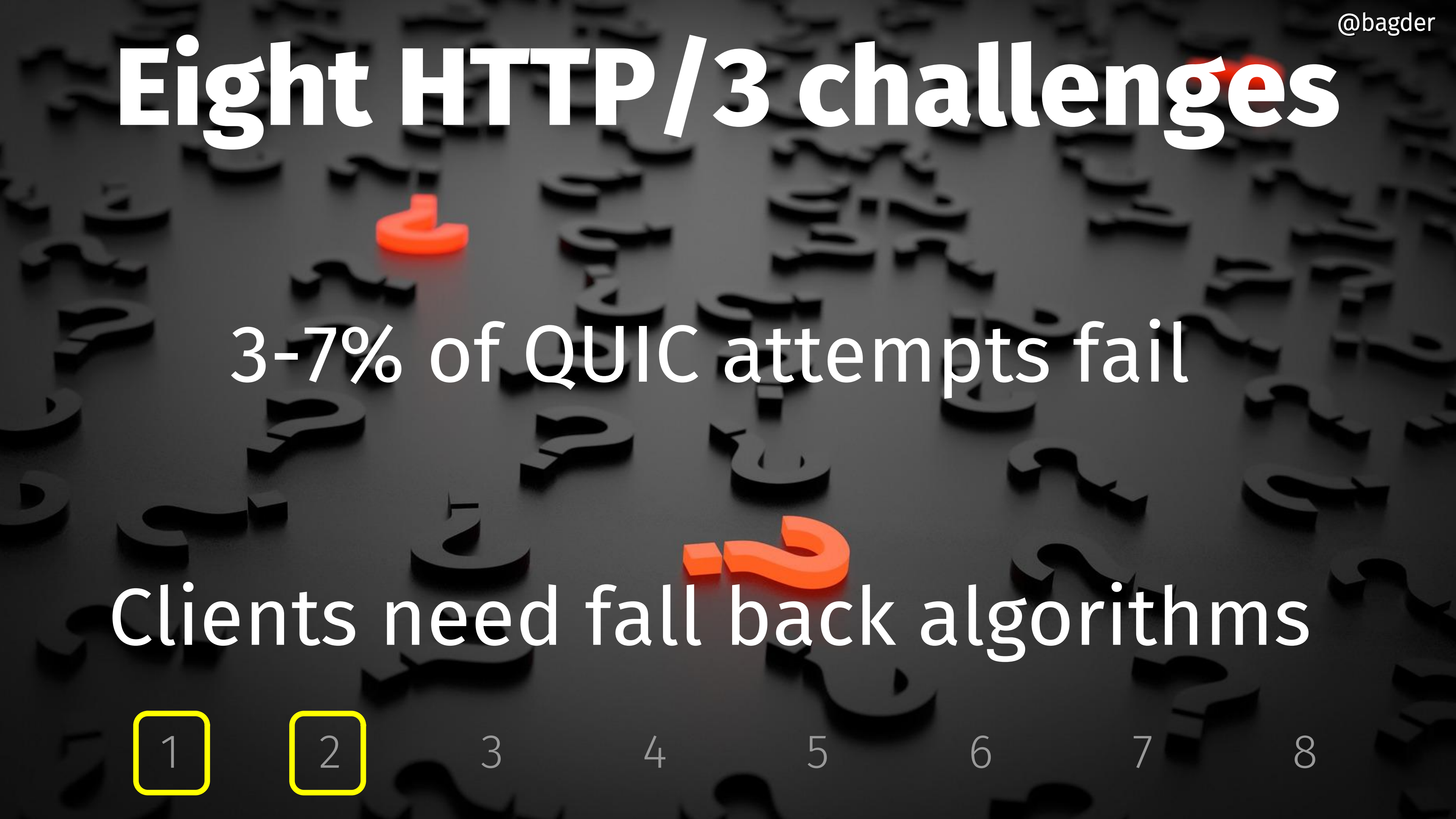
Needed occasionally anyway

QUIC connections verify the host cert

HTTPSSVC

Will HTTP/3 deliver?

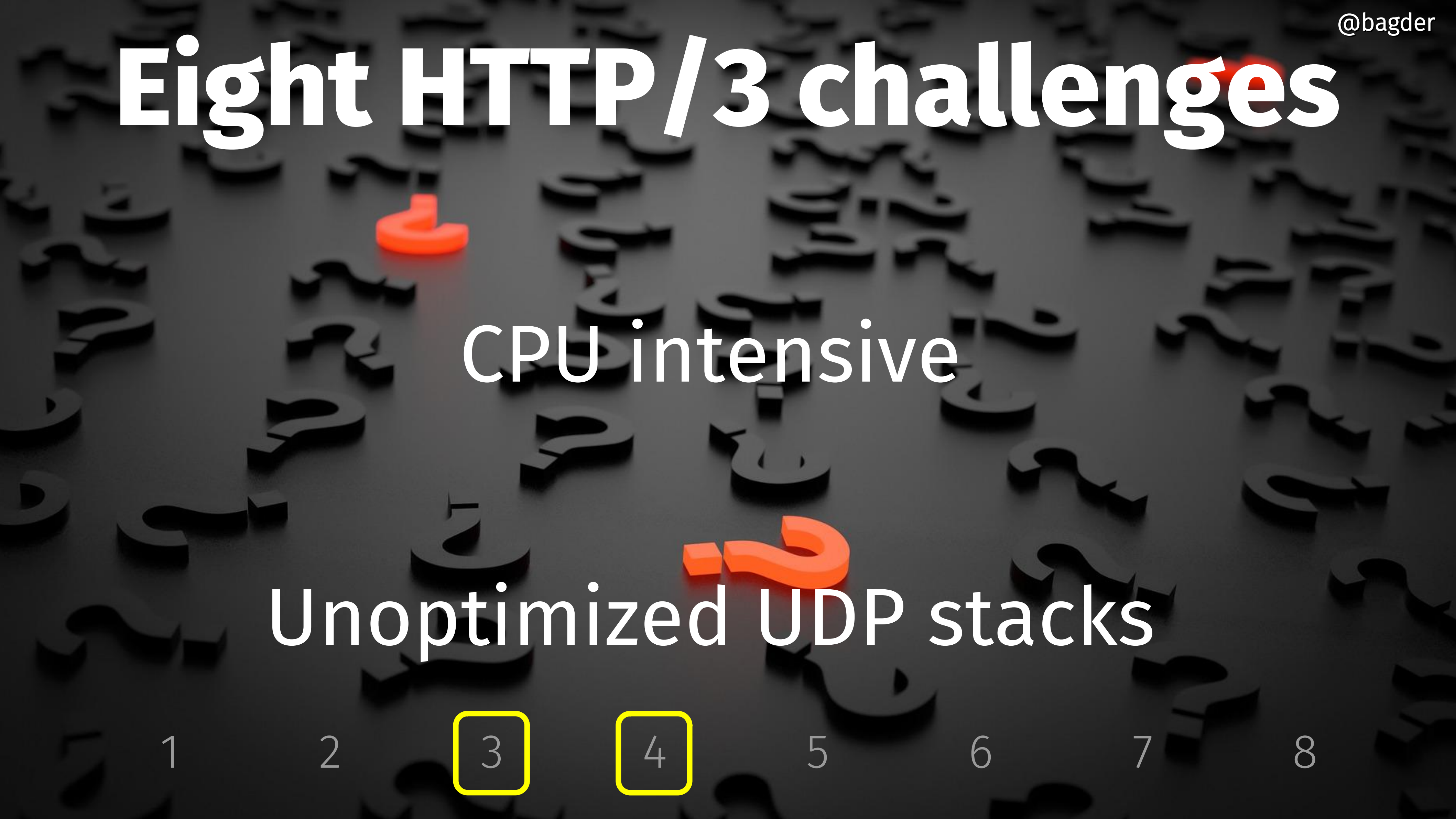
Eight HTTP/3 challenges



3-7% of QUIC attempts fail

Clients need fall back algorithms

$\boxed{1}$ $\boxed{2}$ 3 4 5 6 7 8



@bagder

Eight HTTP/3 challenges

CPU intensive

Unoptimized UDP stacks

1 2 $\boxed{3}$ $\boxed{4}$ 5 6 7 8

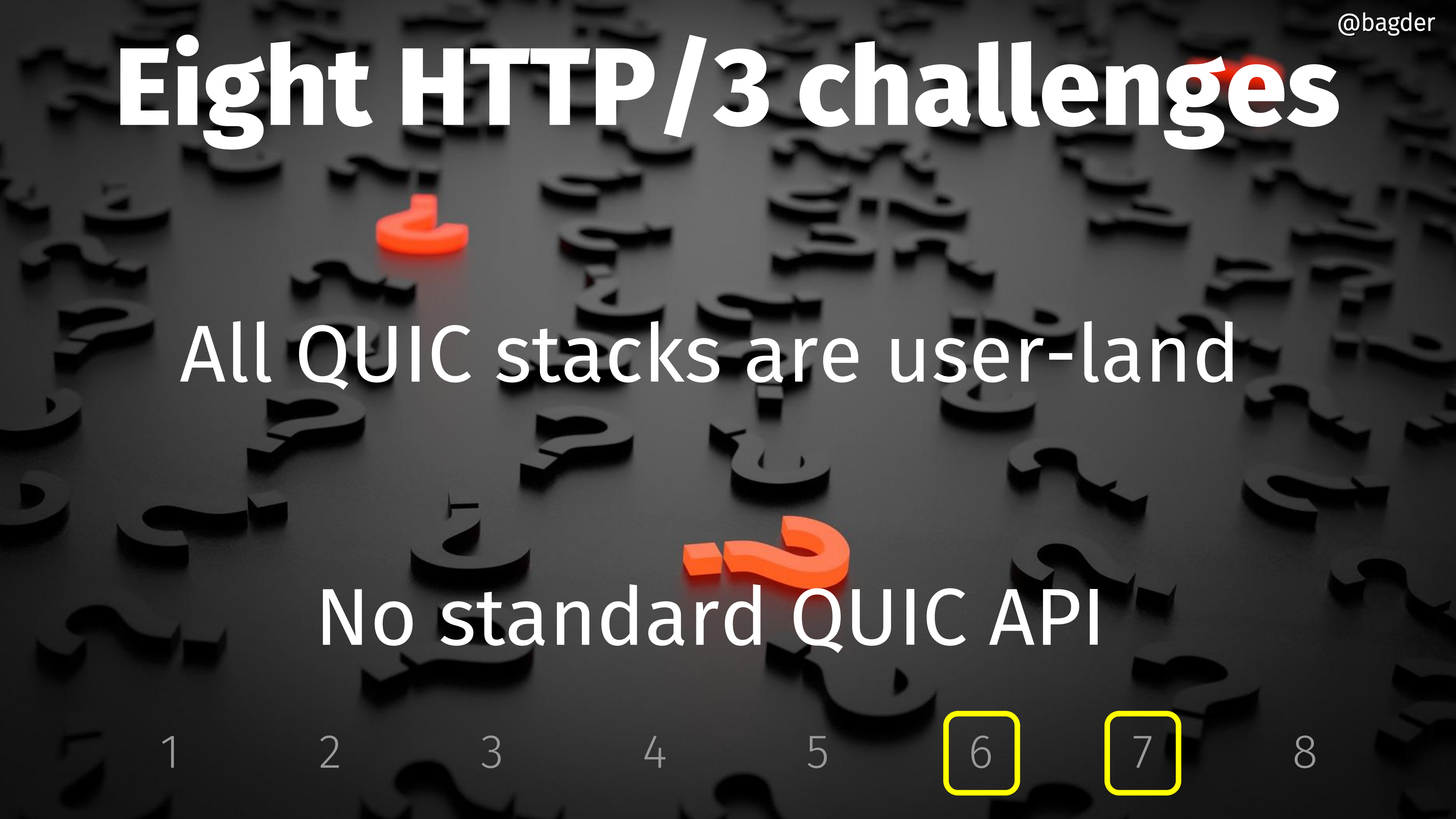
Eight HTTP/3 challenges

“Funny” TLS layer

1 2 3 4 $\boxed{5}$ 6 7 8

@bagder

Eight HTTP/3 challenges



All QUIC stacks are user-land

No standard QUIC API

1 2 3 4 5 $\boxed{6}$ $\boxed{7}$ 8

@bagder

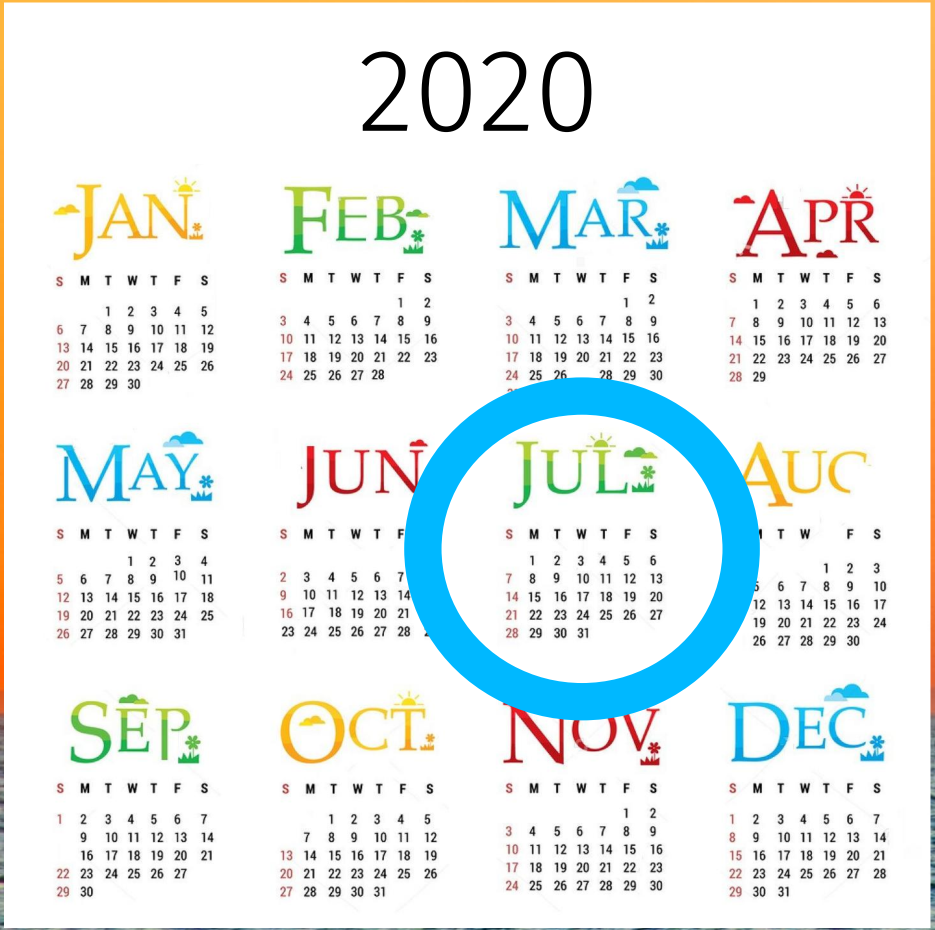
Eight HTTP/3 challenges

Lack of tooling

1 2 3 4 5 6 7 $\boxed{8}$

@bagder

Ship date



2020

JAN

S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

FEB

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28		

MAR

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26		28	29	30

APR

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29					

MAY

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

JUN

S	M	T	W	T	F	
2	3	4	5	6	7	
9	10	11	12	13	14	
16	17	18	19	20	21	
23	24	25	26	27	28	

JUL

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

AUG

	T	W		F	S
			1	2	3
5	6	7	8	9	10
12	13	14	15	16	17
19	20	21	22	23	24
26	27	28	29	30	

SEP

S	M	T	W	T	F	S
1	2	3	4	5	6	7
	9	10	11	12	13	14
	16	17	18	19	20	21
22	23	24	25	26	27	
29	30					

OCT

S	M	T	W	T	F	S
		1	2	3	4	5
	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

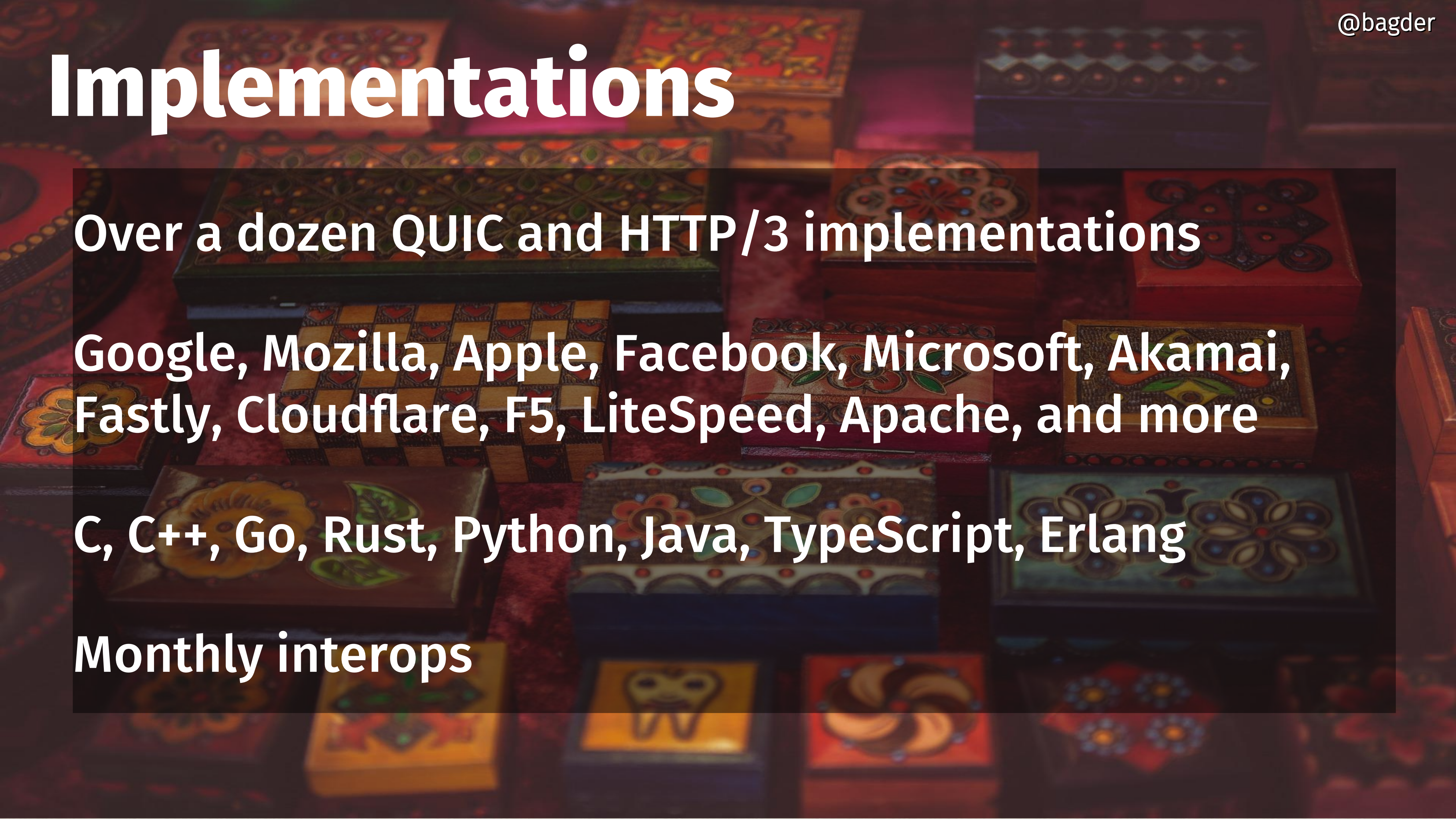
NOV

S	M	T	W	T	F	S
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

DEC

S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Implementations



Over a dozen QUIC and HTTP/3 implementations

Google, Mozilla, Apple, Facebook, Microsoft, Akamai, Fastly, Cloudflare, F5, LiteSpeed, Apache, and more

C, C++, Go, Rust, Python, Java, TypeScript, Erlang

Monthly interops

Implementation Status

😀

curl

Chrome and Edge Canary

Firefox Nightly

Caddy

ngx_quic + quiche

☹️

No Safari

No Apache nor IIS

OpenSSL PR #8797

HTTP/3 in curl

Experimental h3-24 works!

Alt-svc support is there

Based on ngtcp2 and QUICHE

Fallback is tricky

Try it!

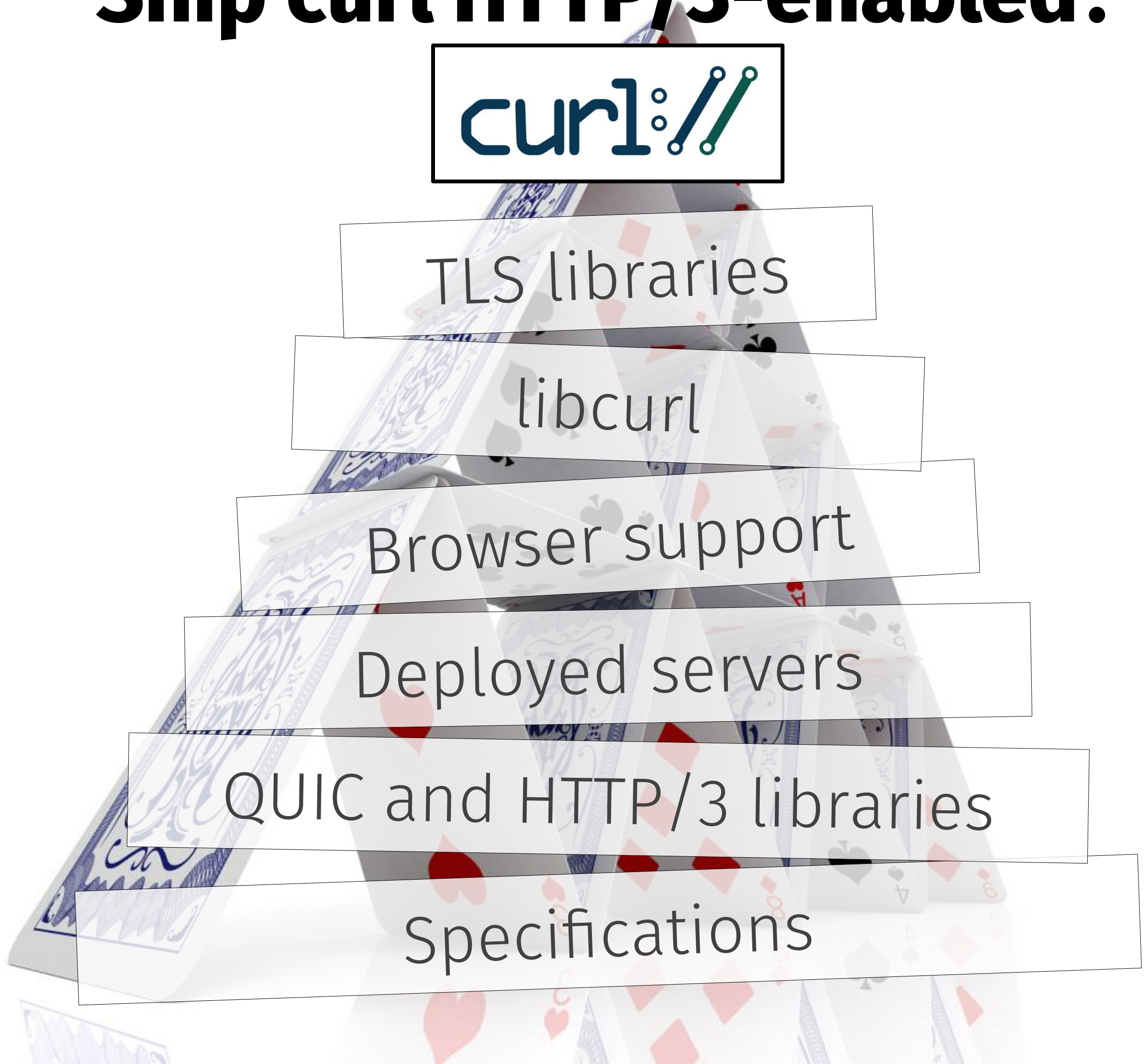
@bagder

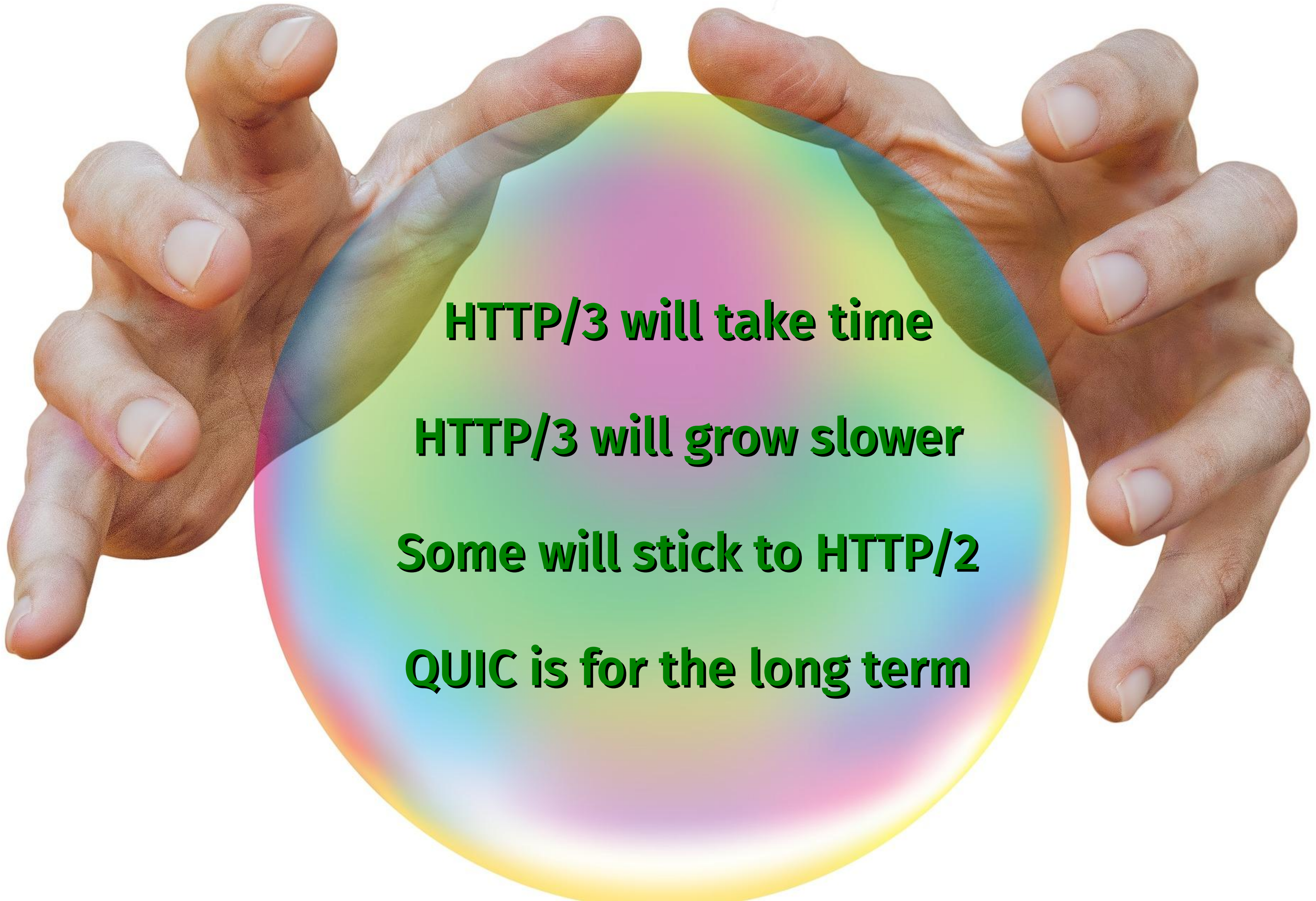
curl HTTP/3 command line

```
$ curl --http3 --head https://example.com/
HTTP/3 200
date: Wed, 09 Oct 2019 11:16:06 GMT
content-type: text/html
content-length: 106072
set-cookie: cfduid=d8bc7e716b30f10766; expires=Thu, 08-
Oct-20 11:16:06 GMT; path=/; domain=example.com;
alt-svc: h3-24=":443"; ma=86400
```

Ship curl HTTP/3-enabled?

curl://





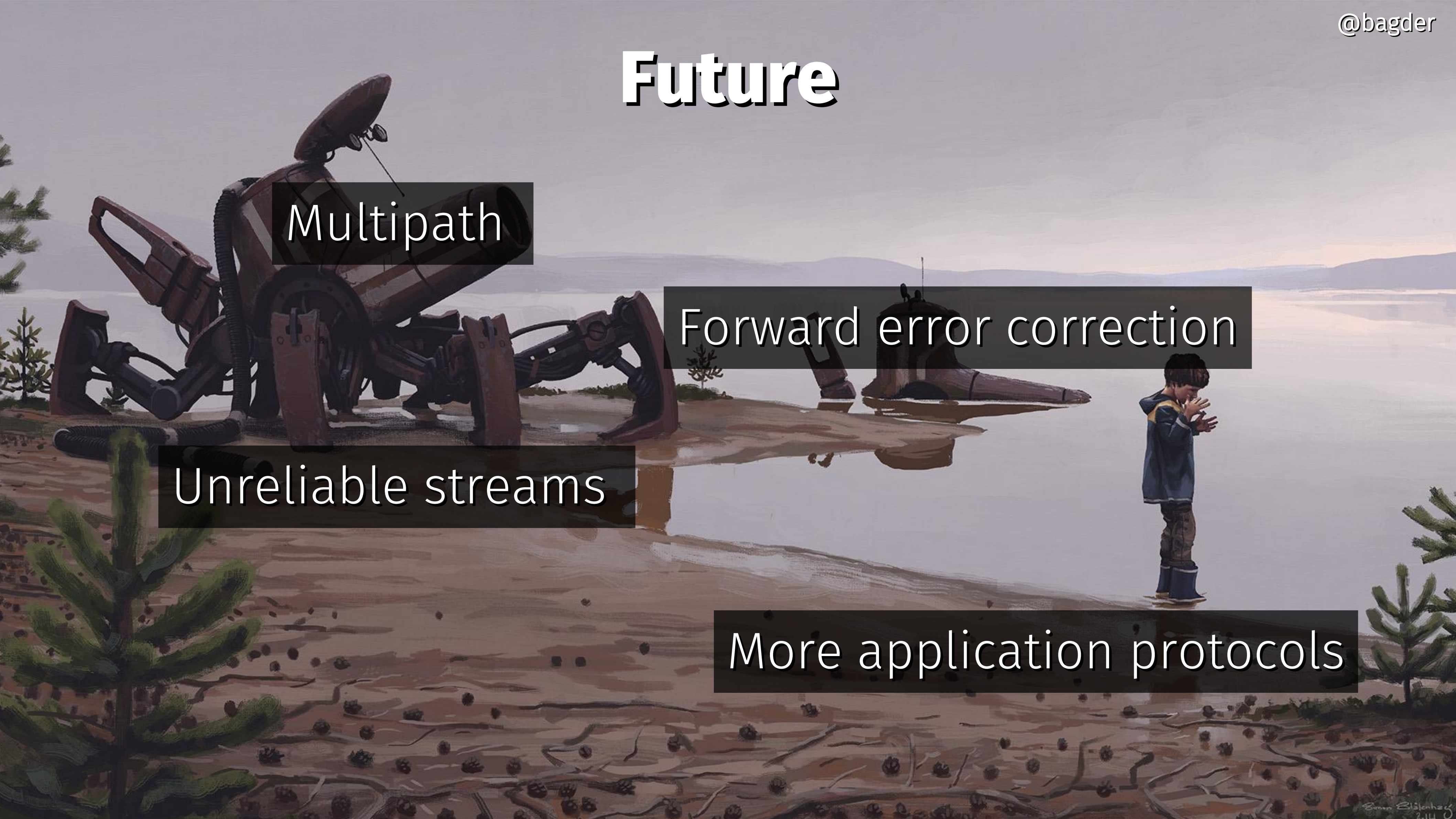
HTTP/3 will take time

HTTP/3 will grow slower

Some will stick to HTTP/2

QUIC is for the long term

Future



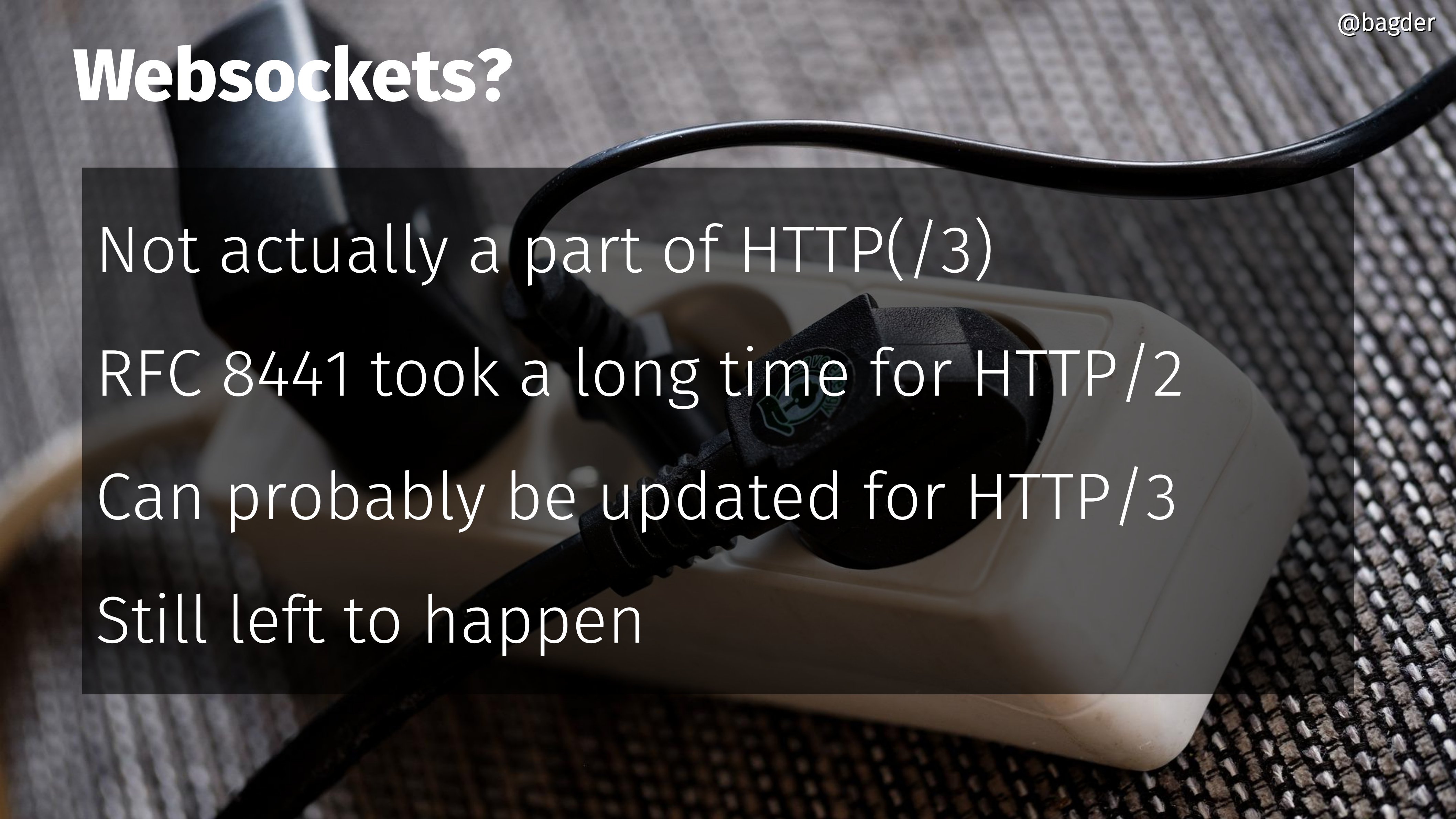
Multipath

Forward error correction

Unreliable streams

More application protocols

Websockets?



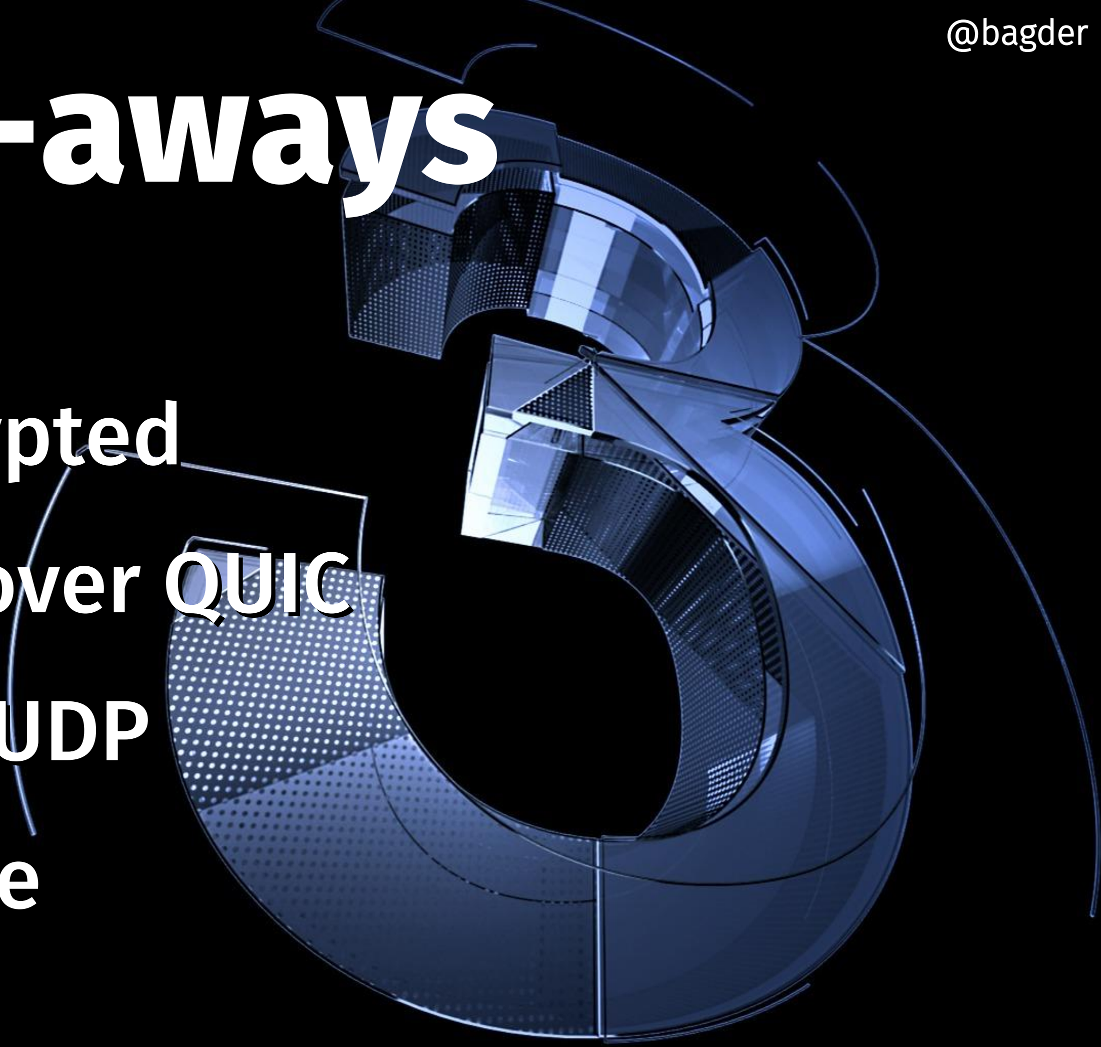
Not actually a part of HTTP(/3)

RFC 8441 took a long time for HTTP/2

Can probably be updated for HTTP/3

Still left to happen

Take-aways



HTTP/3 is coming soon

HTTP/3 is always encrypted

Similar to HTTP/2 but over QUIC

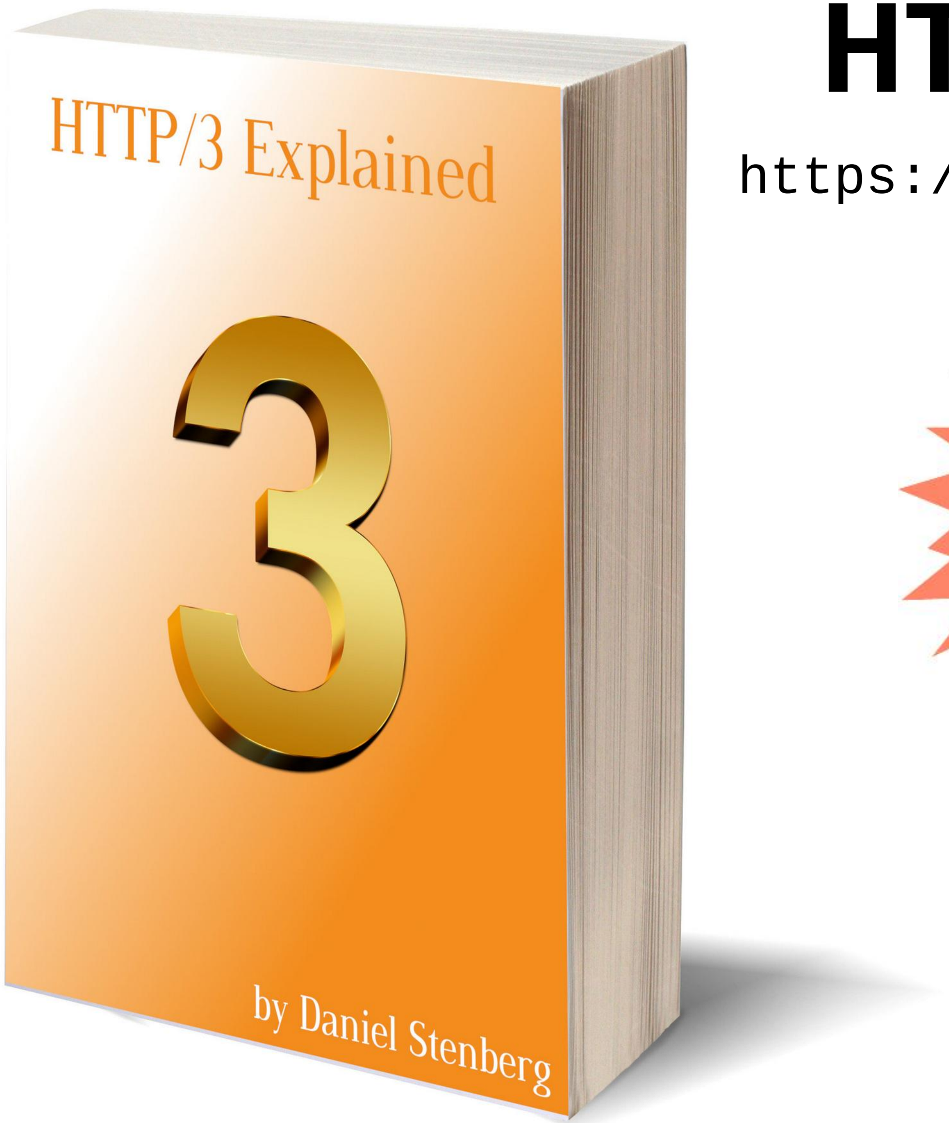
QUIC is transport over UDP

Challenges to overcome

Early/mid 2020?

HTTP/3 Explained

```
https://daniel.haxx.se/http3-explained
```



Thank you!

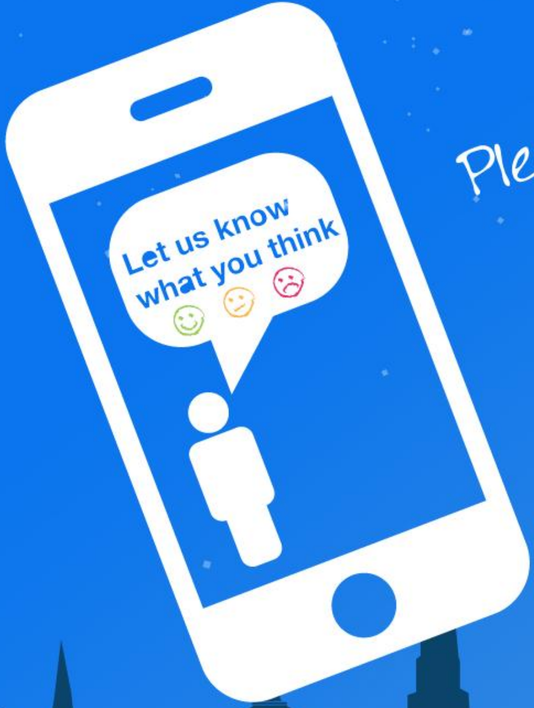
Questions?

Daniel Stenberg
@bagder
https://daniel.haxx.se/

goto; copenhagen

GOTO Copenhagen 2019

Conference Nov. 18 - 20



Please

Remember to rate this session

Thank you!

Follow us @gotocph

License

QUIC and HTTP/3 links

QUIC drafts: https://quicwg.github.io/

HTTPS stats Firefox: https://letsencrypt.org/stats/#percent-pageloads

HTTPS stats Chrome: https://transparencyreport.google.com/https/overview?hl=en

Images: http://www.simonstalenhag.se/ and https://pixabay.com/

HTTP/3 Explained: https://http3-explained.haxx.se/

QUIC implementations: https://github.com/quicwg/base-drafts/wiki/Implementations

HTTPSSVC: https://tools.ietf.org/html/draft-nygren-dnsop-svcb-httpssvc-00

Build curl with HTTP/3: https://github.com/curl/curl/blob/master/docs/HTTP3.md