Daniel Stenberg
GOTOpia
November 2020

HTTP/3

is next generation HTTP

Is it

QUIC

enough?

Daniel Stenberg

https://daniel.haxx.se **@bagder**

curl://

Daniel Stenberg

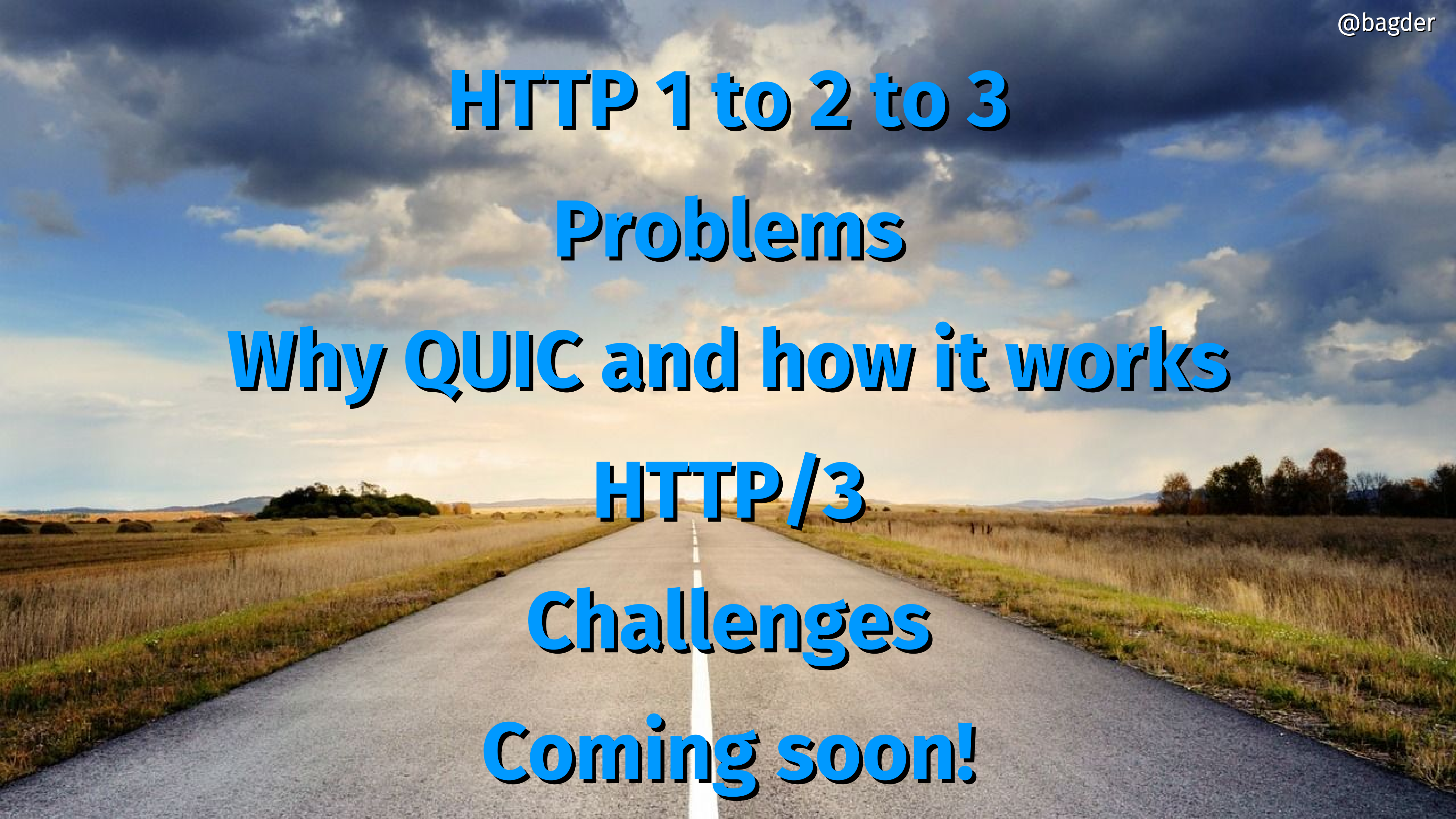
@bagder

wolfSSL

Daniel Stenberg

@bagder

IETF®



HTTP 1 to 2 to 3

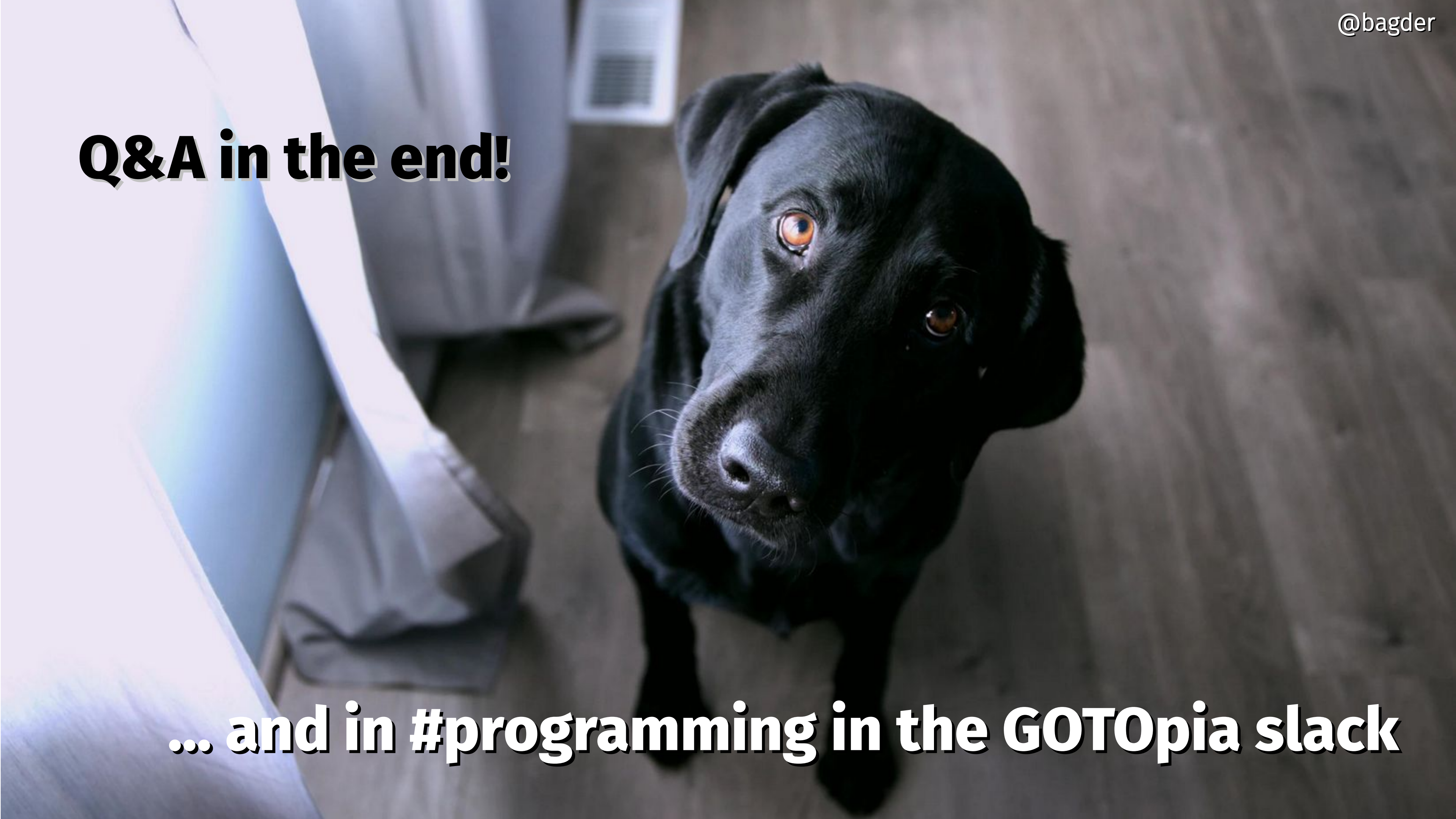
Problems

Why QUIC and how it works

HTTP/3

Challenges

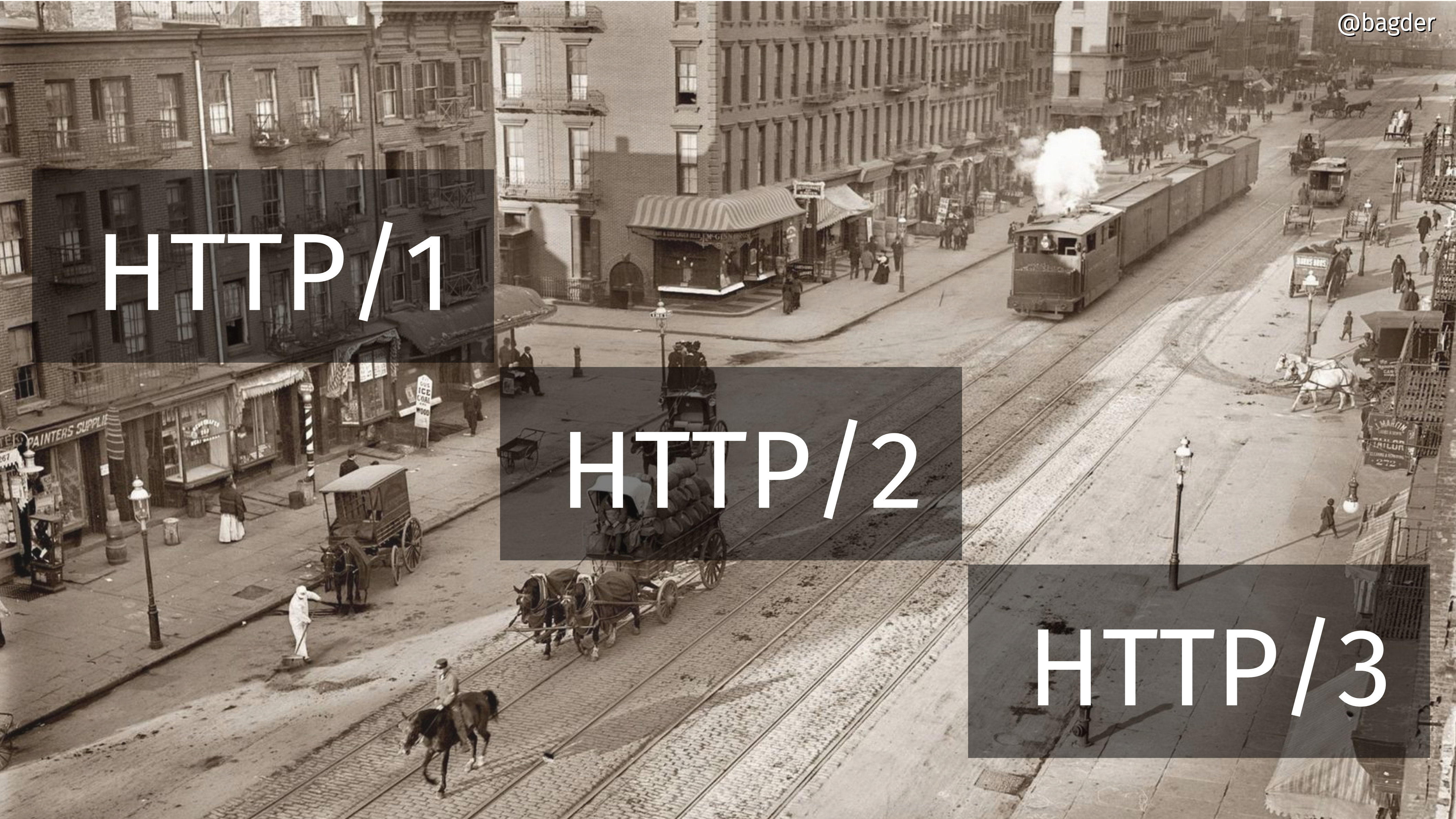
Coming soon!



Q&A in the end!

... and in #programming in the GOTOpia slack

@bagder



@bagder

Under the hood

```
GET / HTTP/1.1
Host: www.example.com
Accept: */*
User-Agent: HTTP-eats-the-world/2020
```

```
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 is transport over IP

Establishes a “connection”

3-way handshake

Resends lost packages

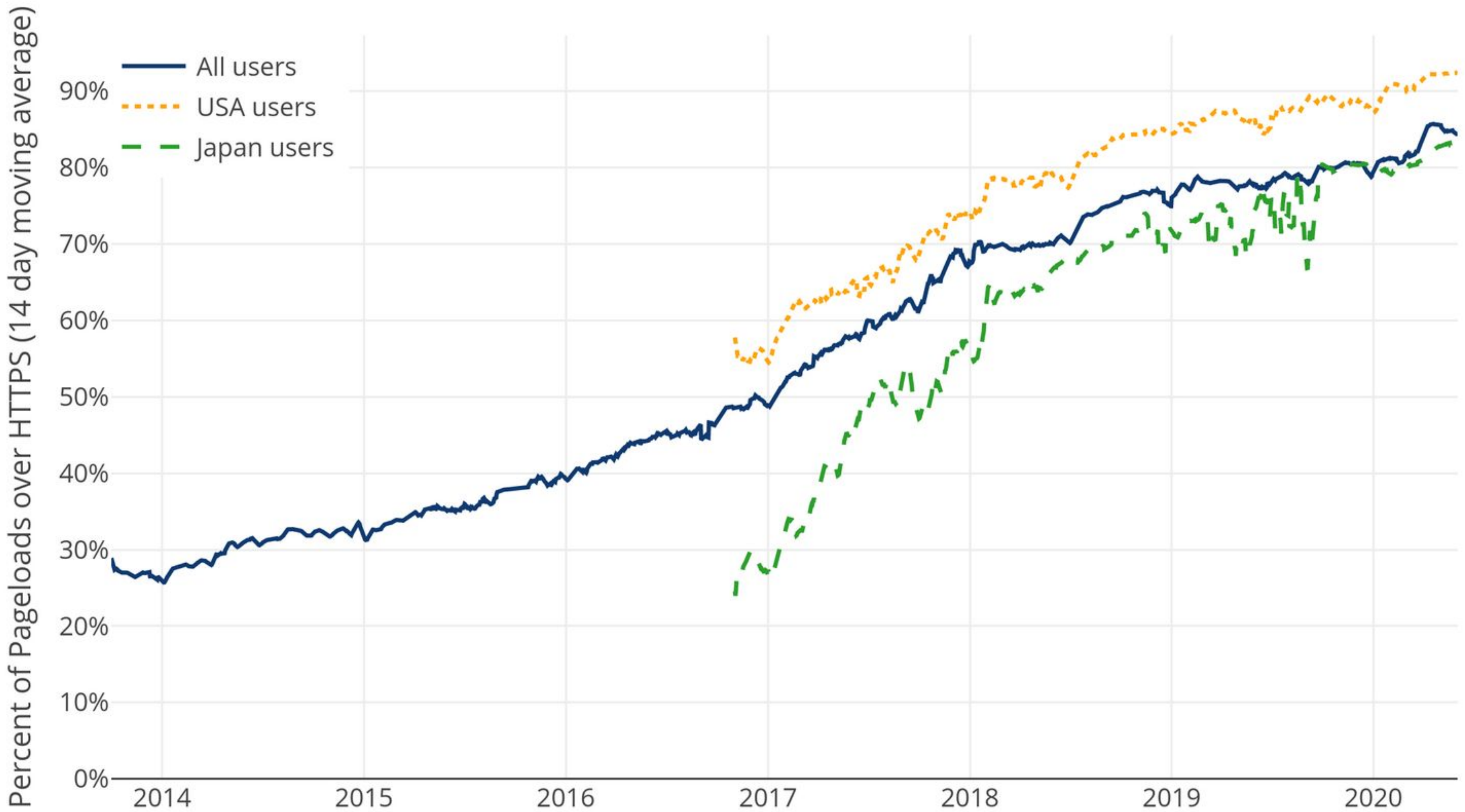
A reliable byte stream

Clear text

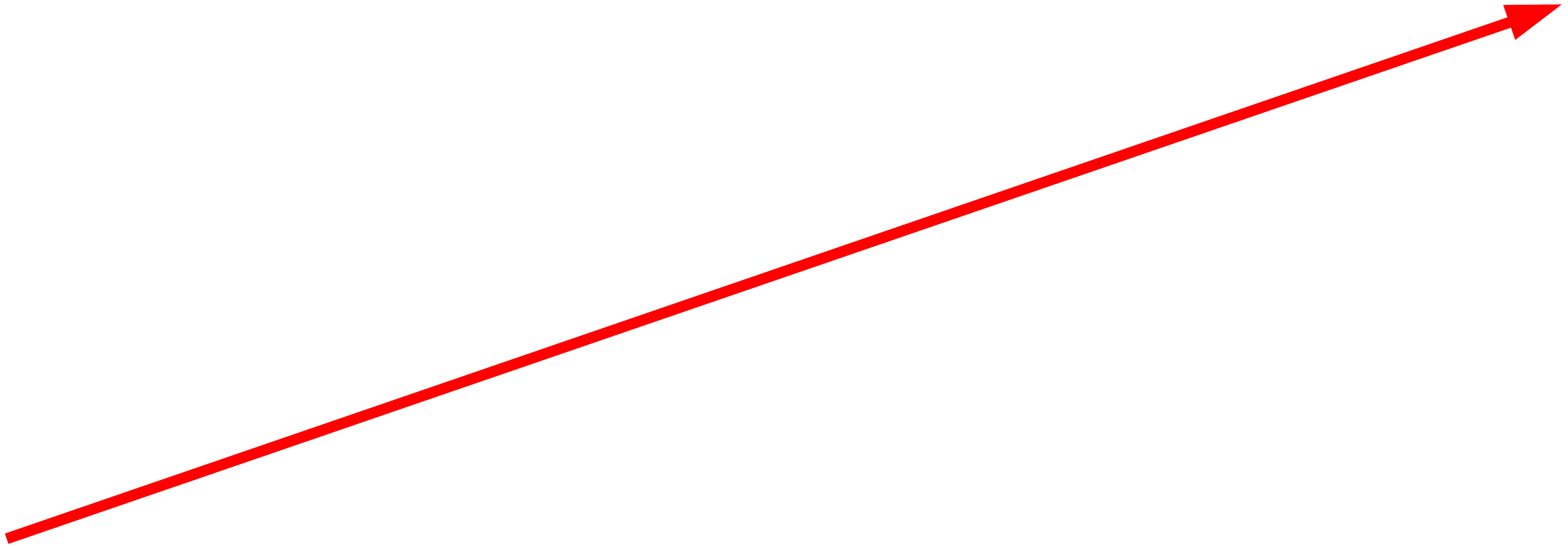
HTTPS means TCP + TLS + HTTP

@bagder

Web pages over HTTPS in Firefox



Web pages over HTTPS



TLS

TLS is done over TCP for HTTP/1 or 2

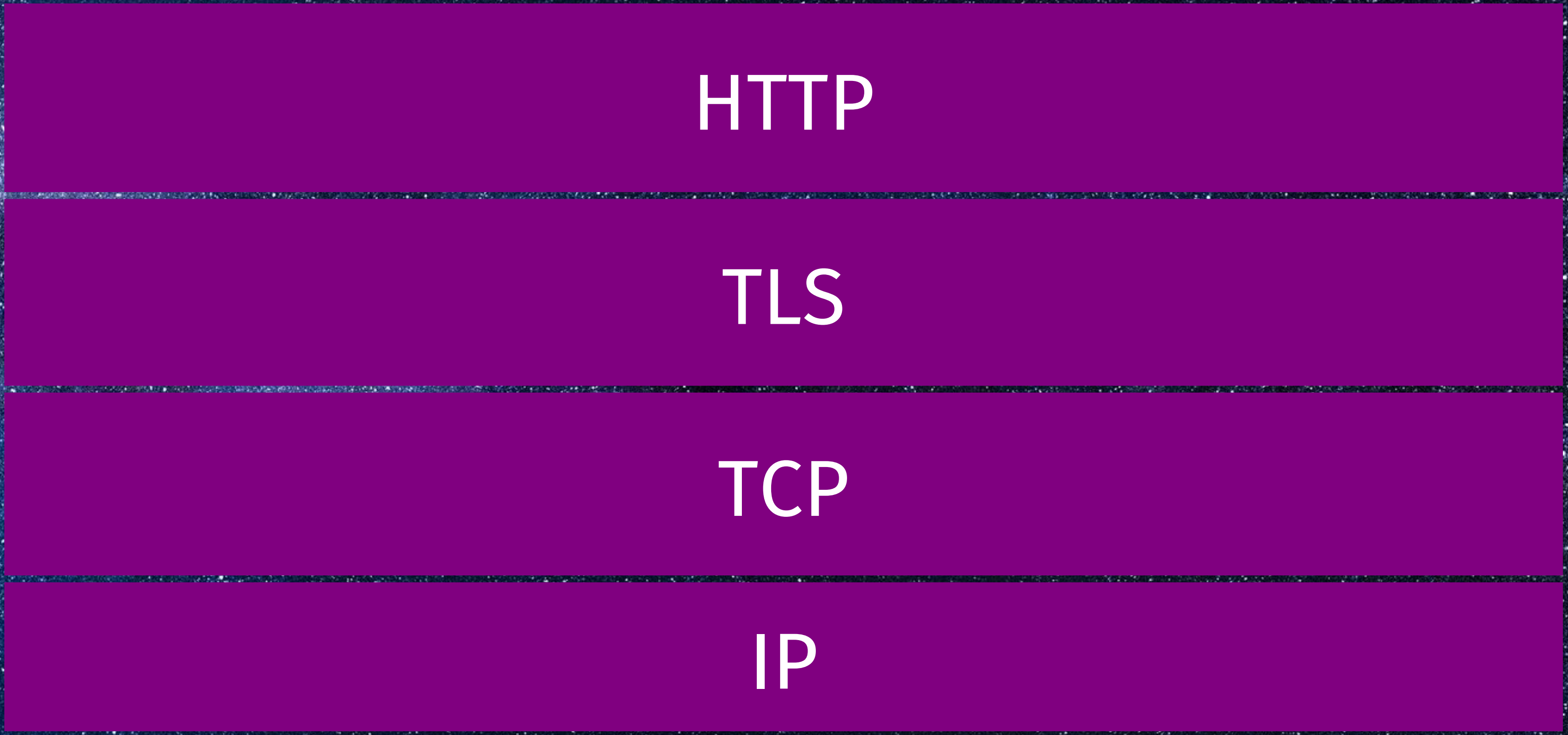
Transport Layer Security

Additional handshake

Privacy and security

@bagder

Classic HTTPS stack



HTTP over TCP

HTTP/1.1

@bagder

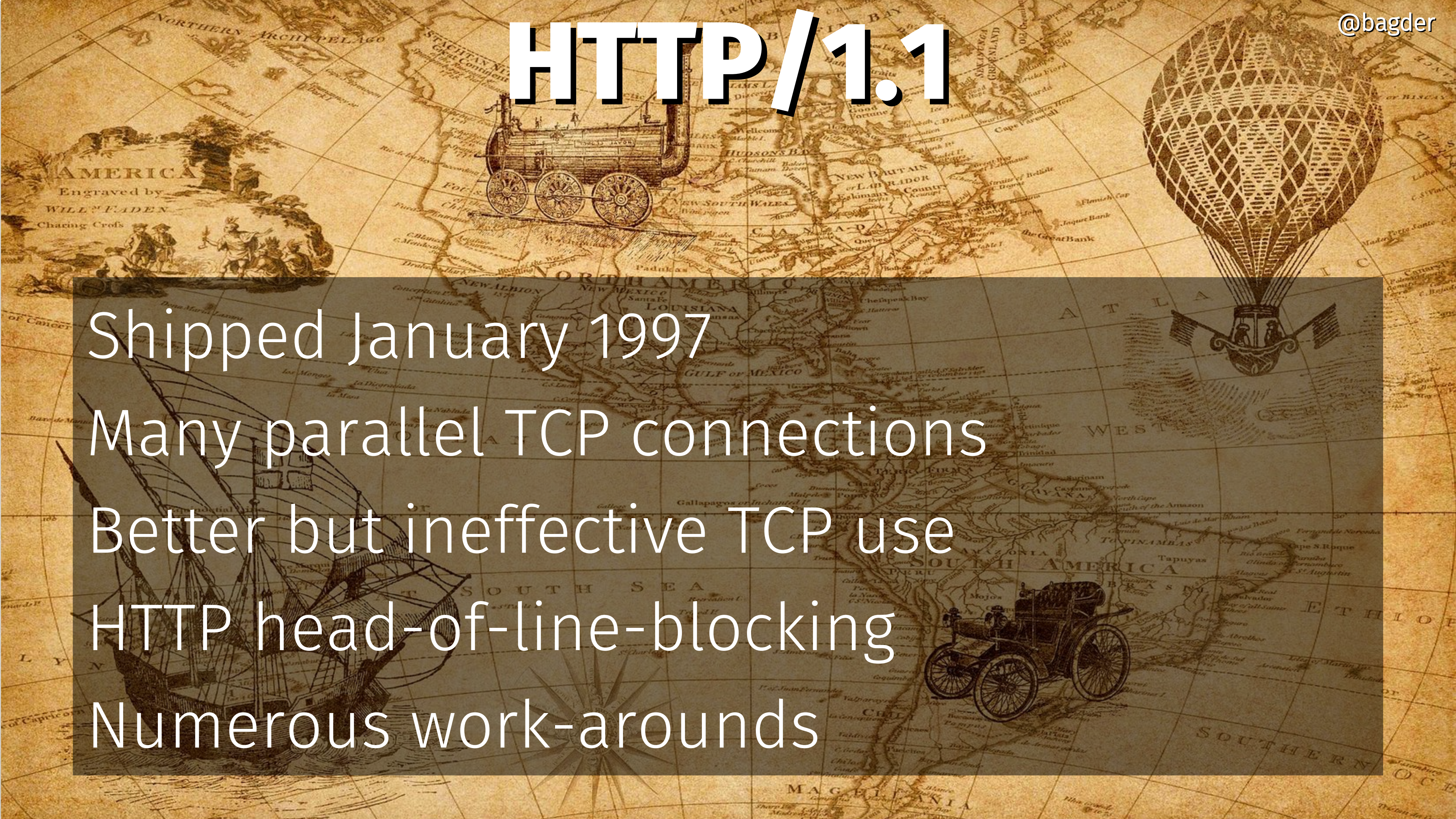
Shipped January 1997

Many parallel TCP connections

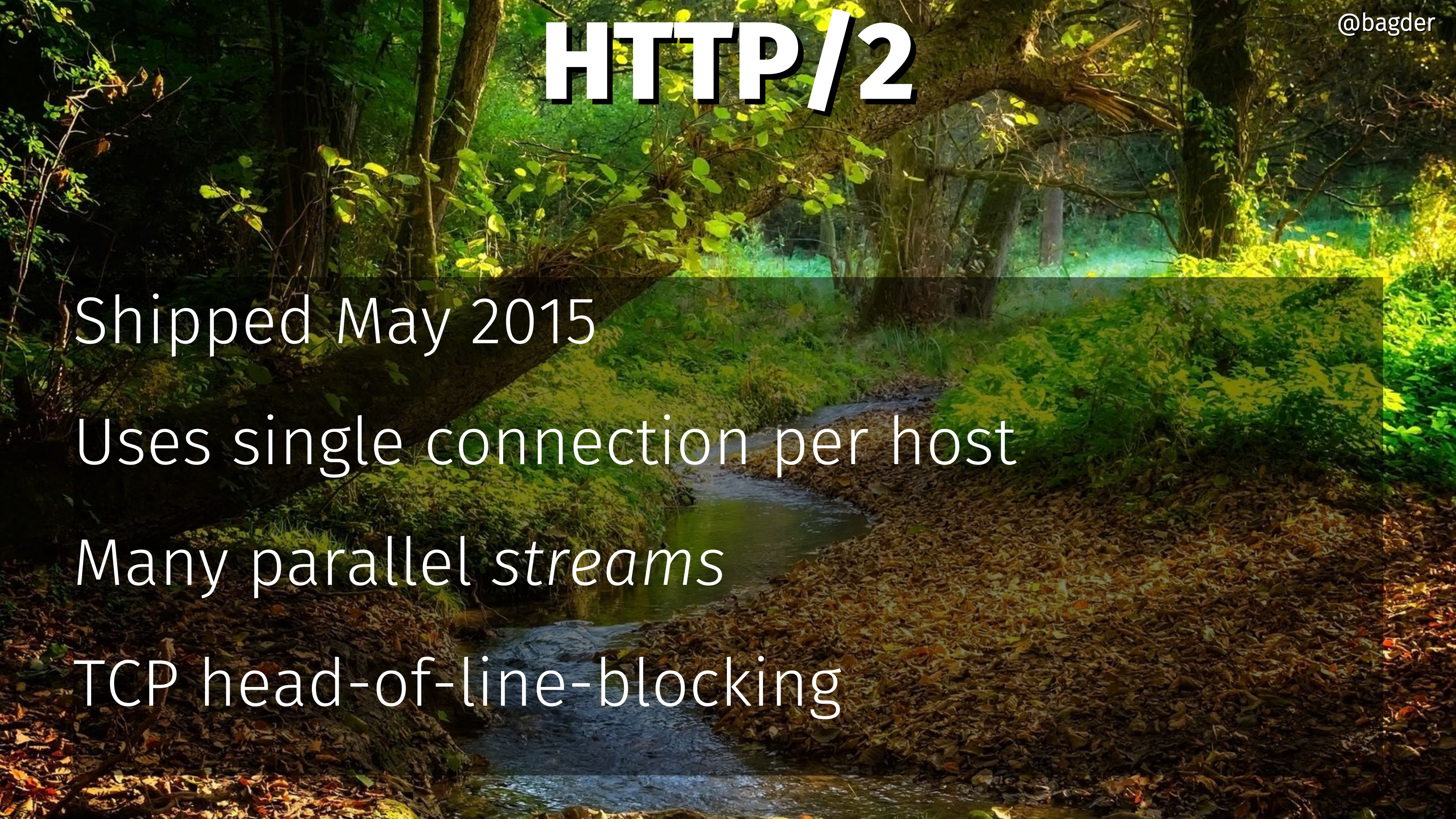
Better but ineffective TCP use

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

@bagder

Internet is full of boxes

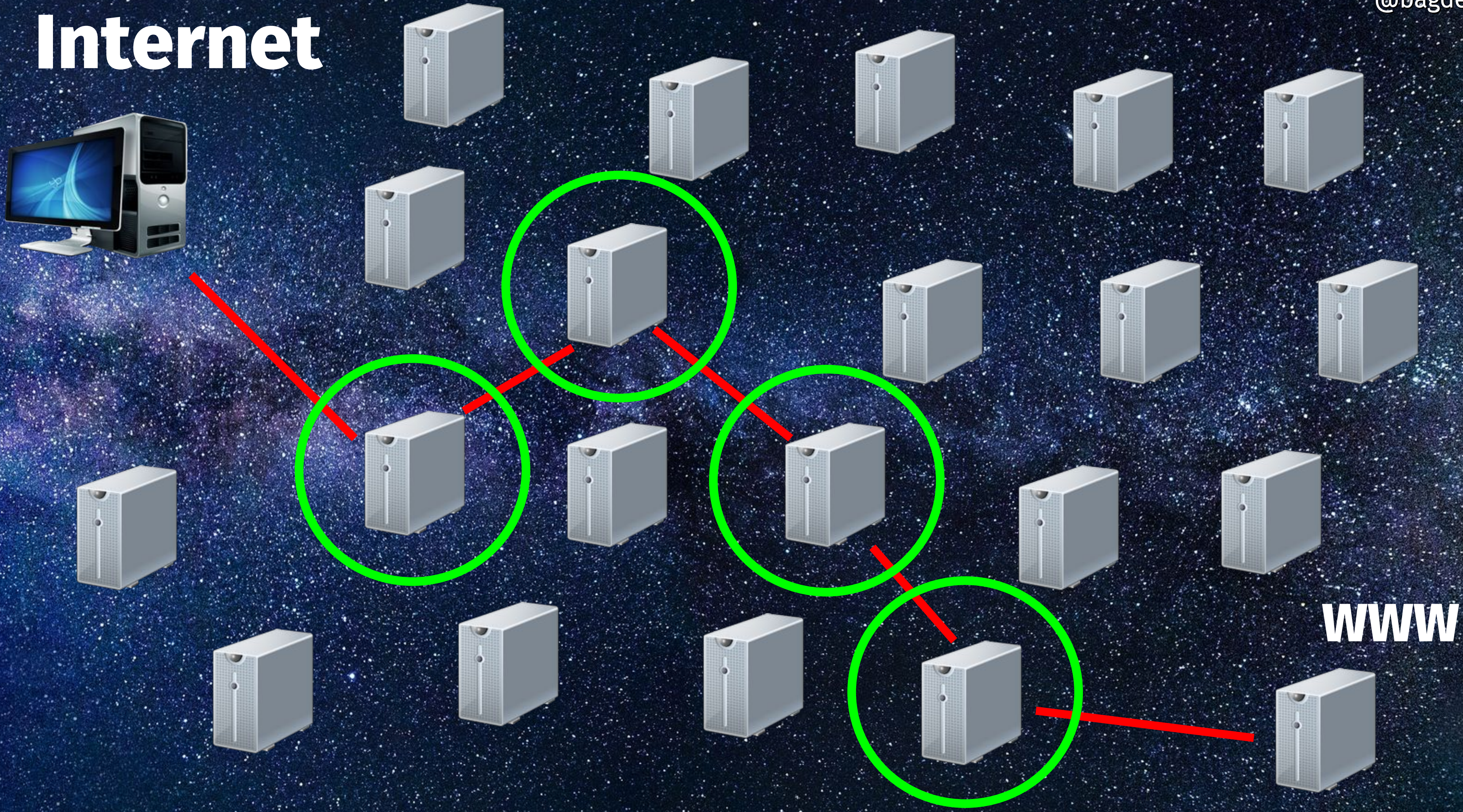
Routers, gateways, firewalls, load balancers, NATs...

Boxes run software to handle network data

Middle-boxes work on existing protocols

Upgrade much slower than edges

Internet



Middle-boxes prevent

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

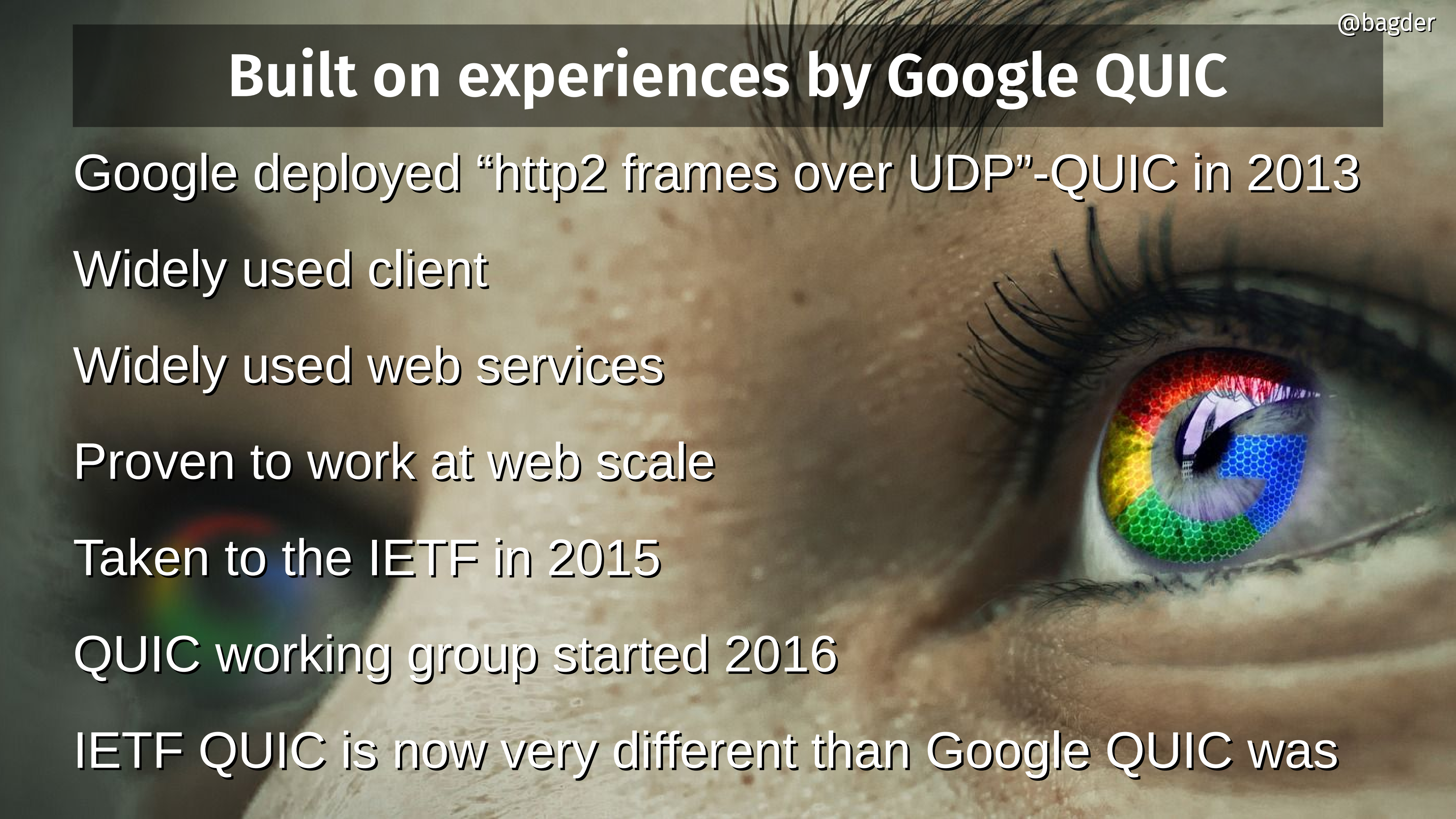
moz://a

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

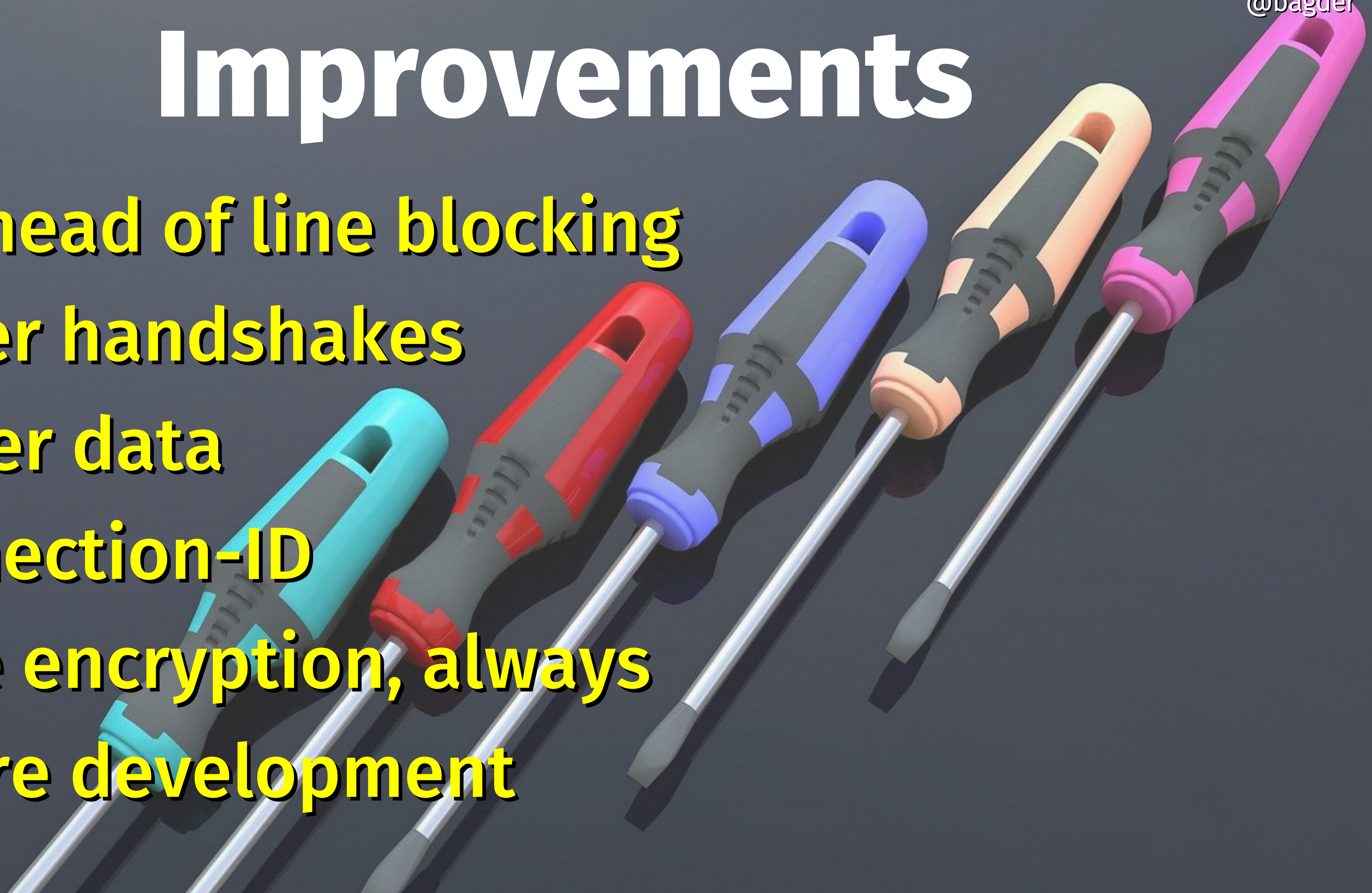
Proven to work at web scale

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

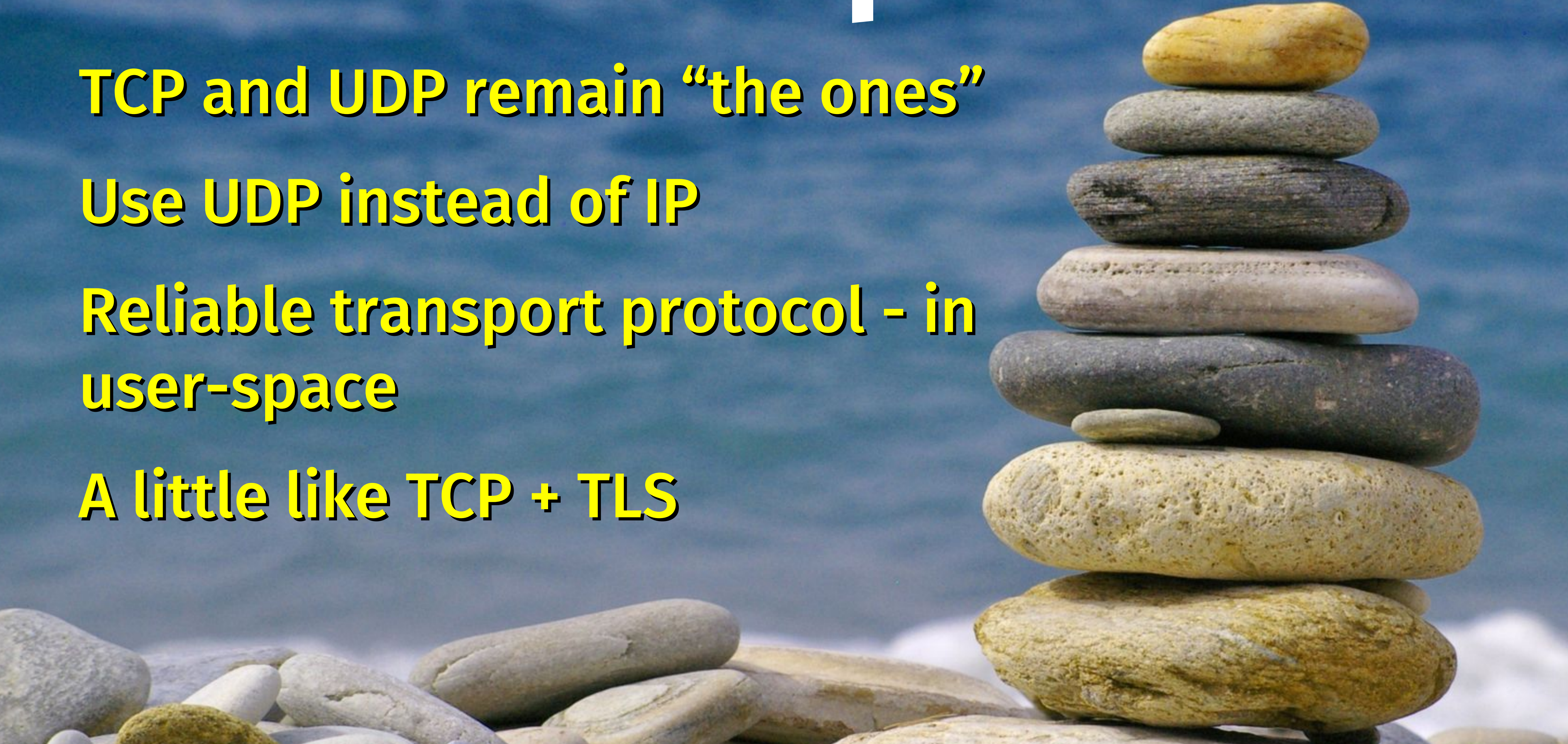
Earlier data

Connection-ID

More encryption, always

Future development

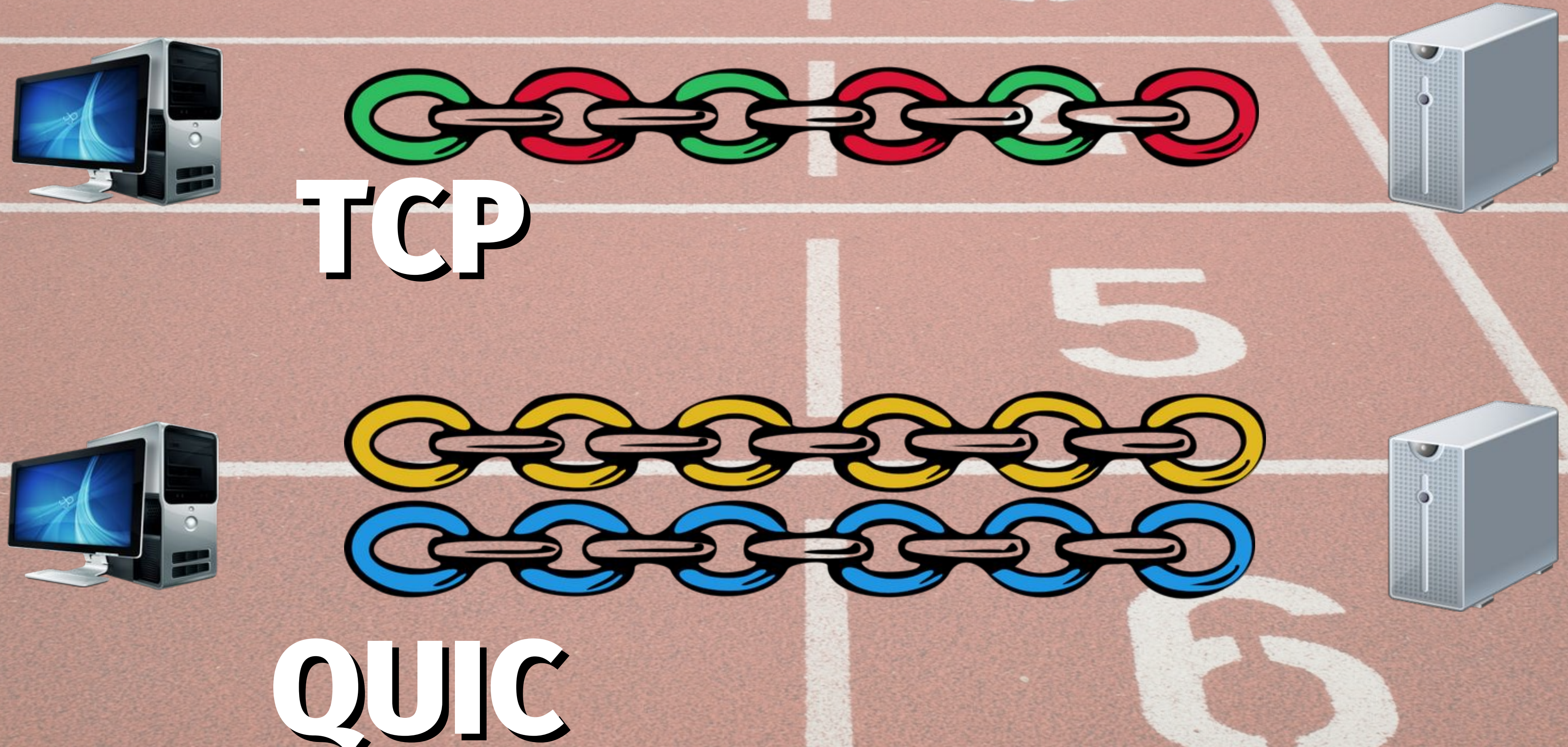
Build on top of UDP

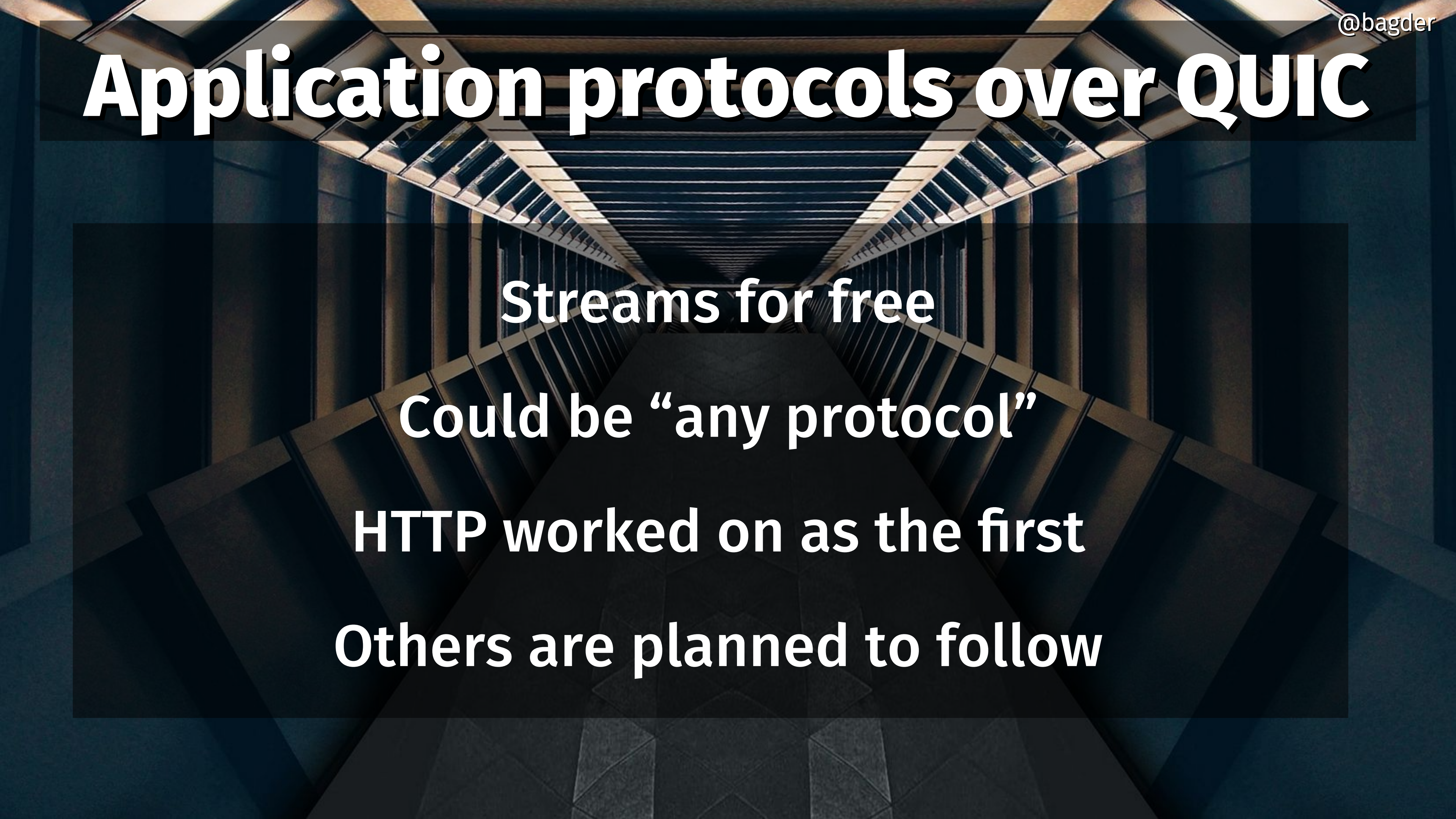


- TCP and UDP remain “the ones”
- Use UDP instead of IP
- Reliable transport protocol - in user-space
- A little like TCP + TLS

@bagder

Independent streams





@bagder

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

@bagder

HTTP – same but different

Request

- method + path
- headers
- body

Response

- response code
- headers
- body

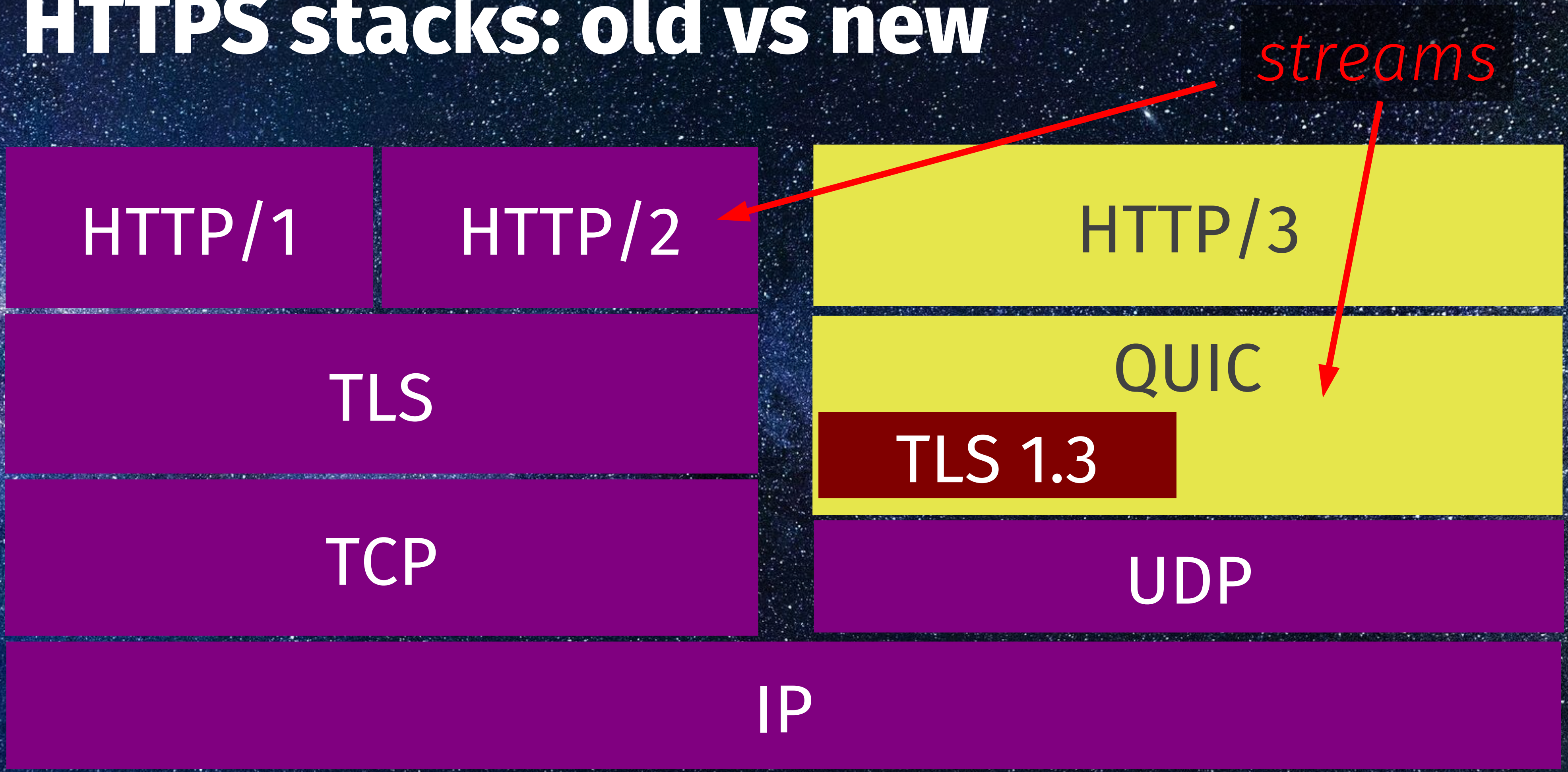
HTTP – same but different

HTTP/1 – in ASCII over TCP

HTTP/2 – binary multiplexed over TCP

HTTP/3 – binary over multiplexed QUIC

HTTPS stacks: old vs new



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

Improved loss-recovery

Early data that works

The independent streams

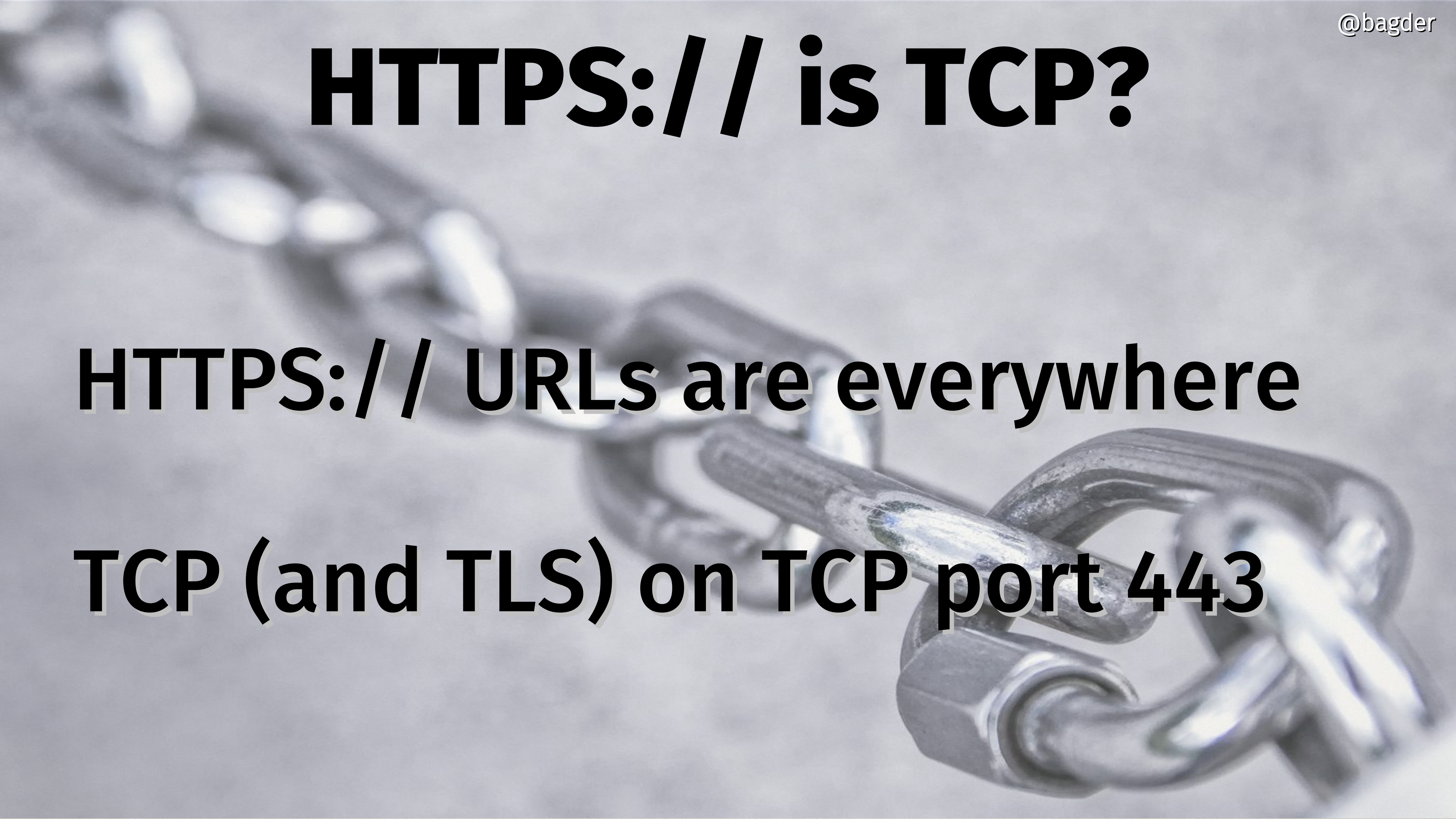
By how much remains to be measured!

What they say

"We've found that IETF QUIC significantly outperforms HTTP over TLS 1.3 over TCP." / Google

"QUIC and HTTP/3 generally outperform TCP and HTTP/2, which in turn outperform TCP and HTTP/1.1." / Facebook

HTTPS:// is TCP?



HTTPS:// URLs are everywhere

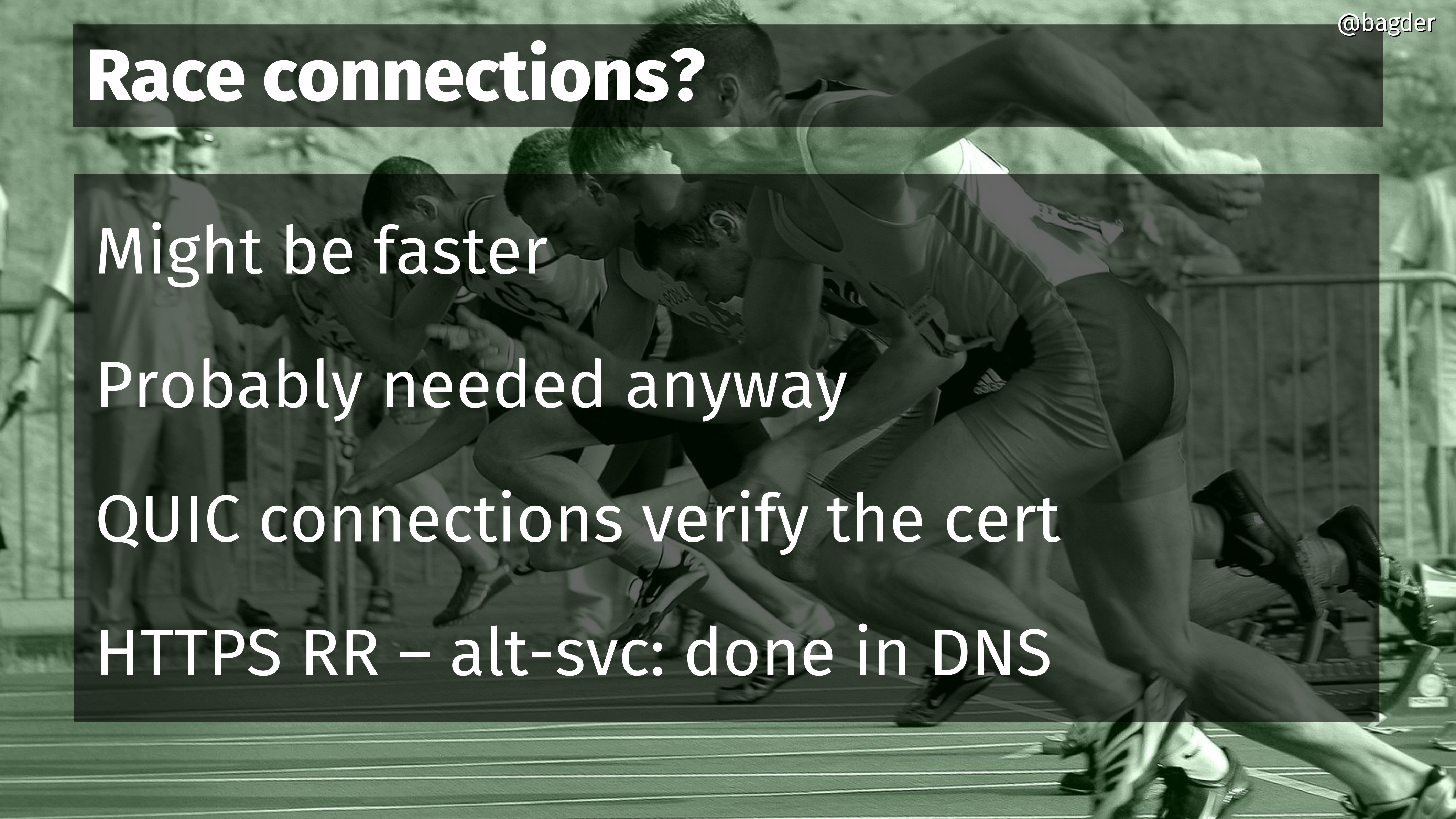
TCP (and TLS) on TCP port 443

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



@bagder

Race connections?

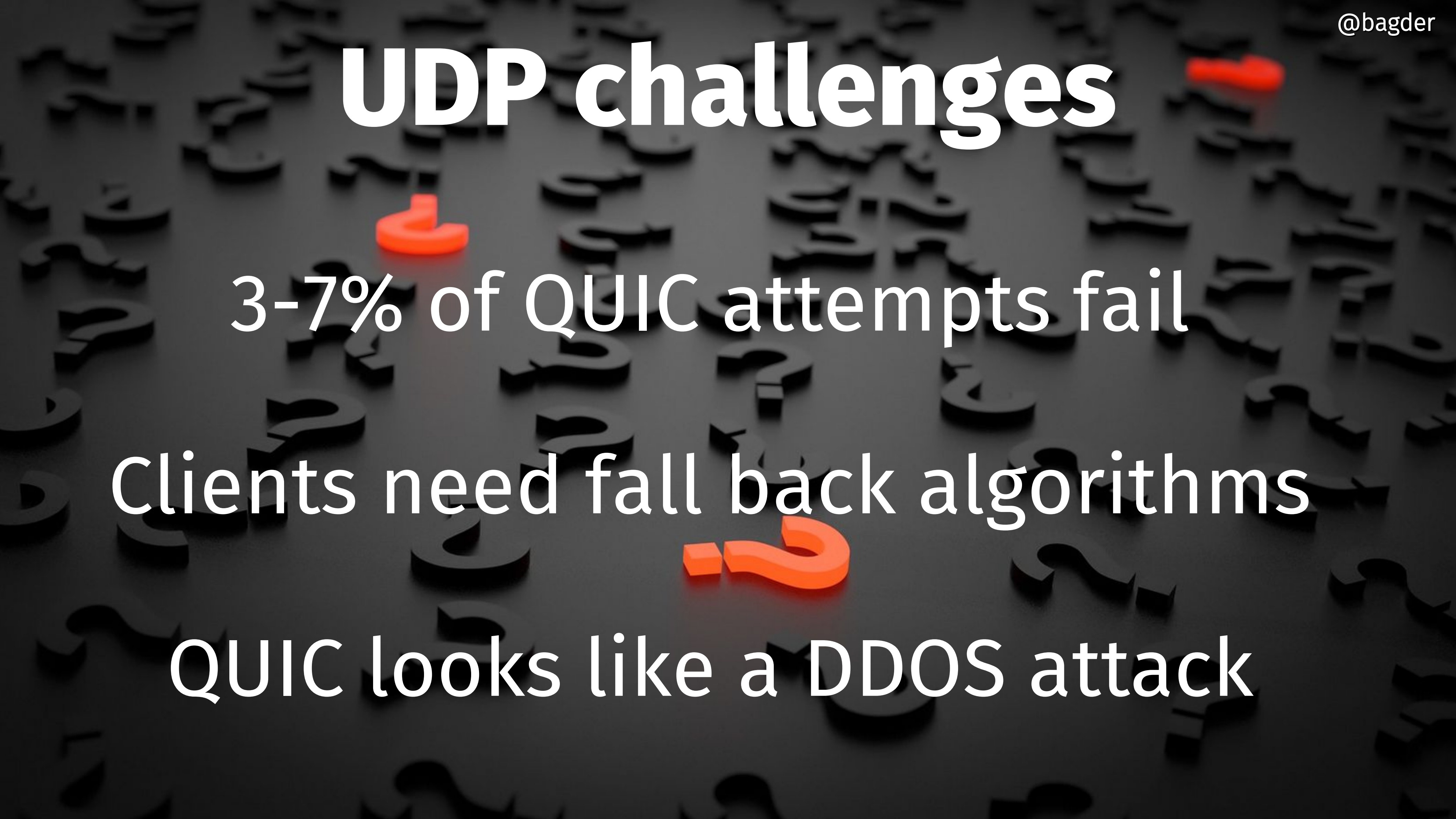
Might be faster

Probably needed anyway

QUIC connections verify the cert

HTTPS RR – alt-svc: done in DNS

Will HTTP/3 deliver?



UDP challenges

3-7% of QUIC attempts fail

Clients need fall back algorithms

QUIC looks like a DDOS attack

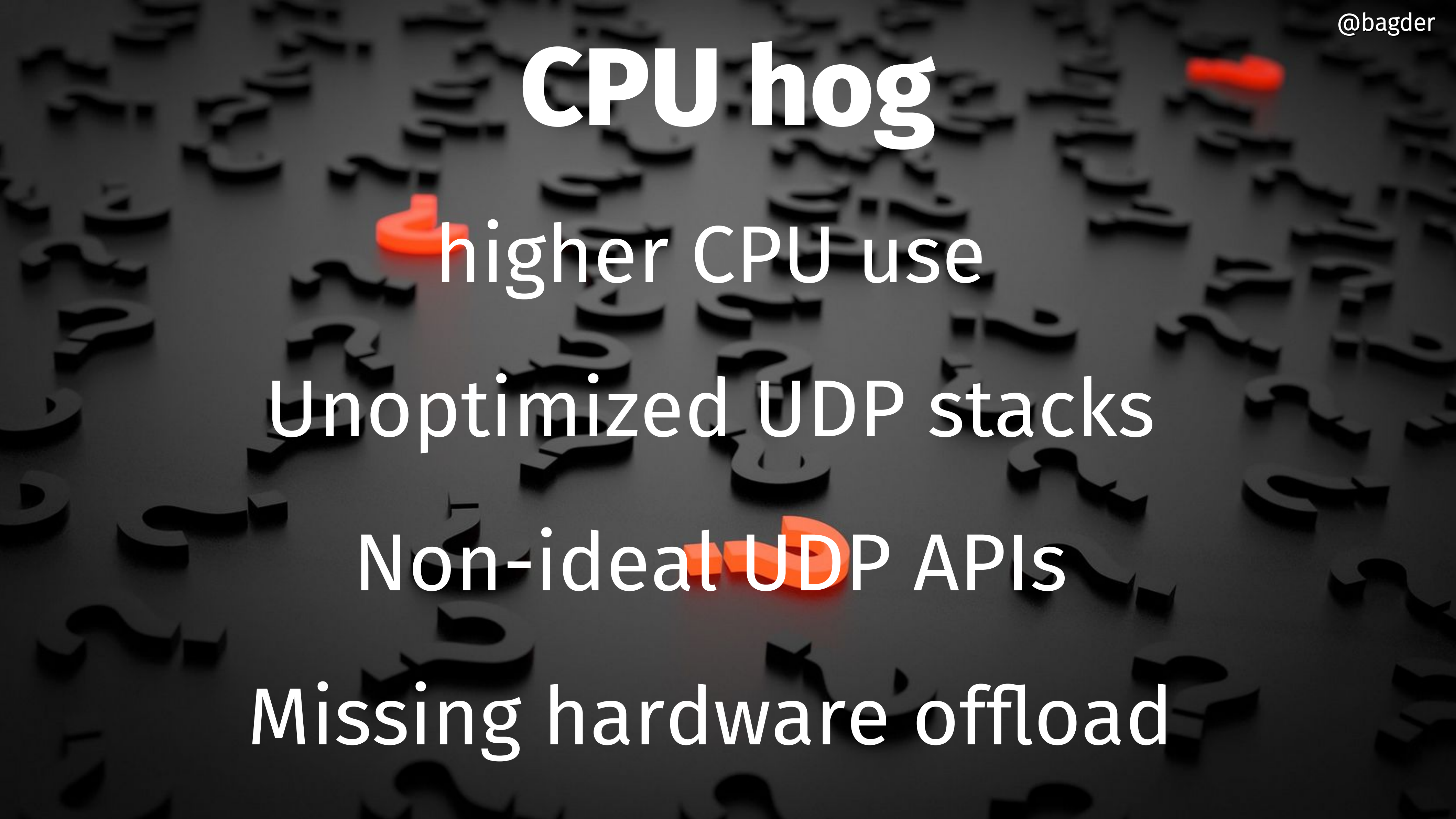
CPU hog

higher CPU use

Unoptimized UDP stacks

Non-ideal UDP APIs

Missing hardware offload



The TLS situation (1/3)

TLS was made for TCP

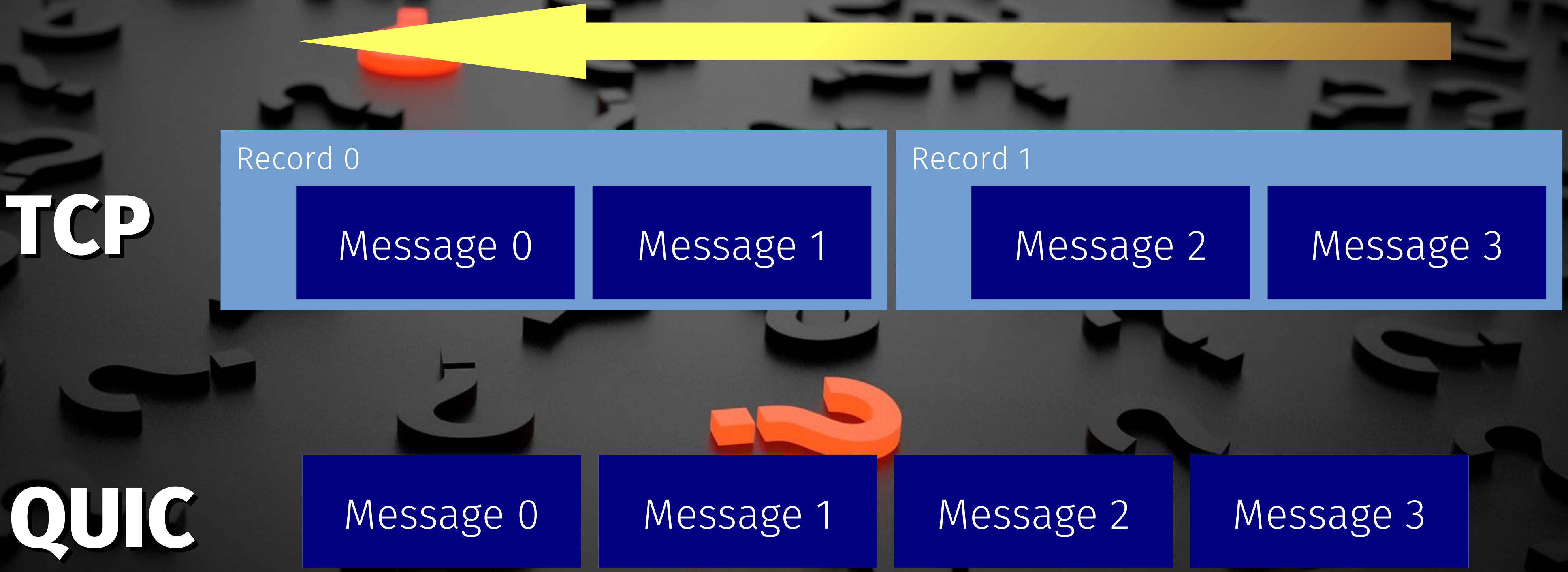
TLS is sent over TCP as *records* containing individual *messages*

QUIC uses TLS *messages*

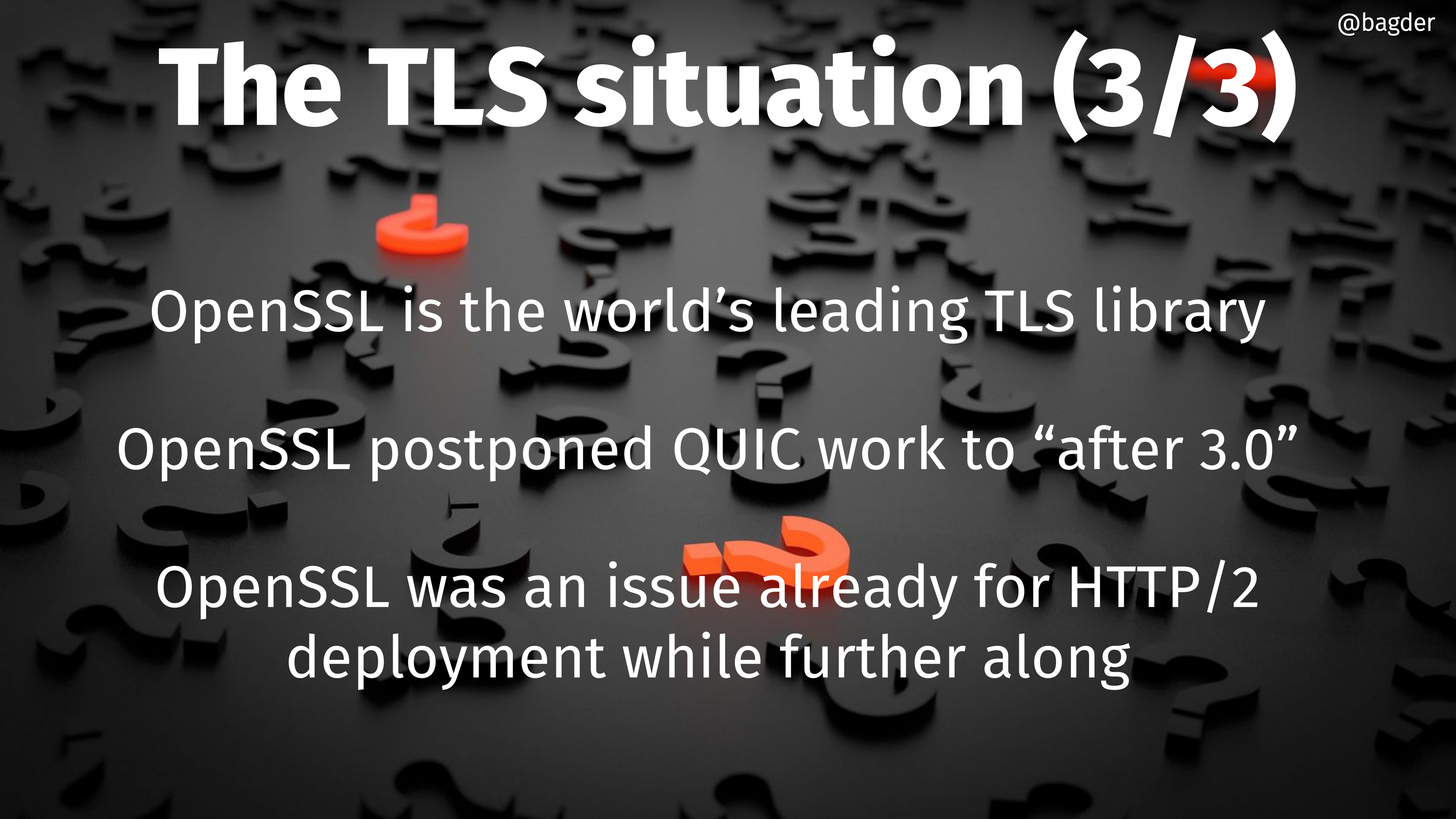
No TLS library support(ed) TLS messages

QUIC also needs additional secrets

The TLS situation (2/3)



The TLS situation (3/3)

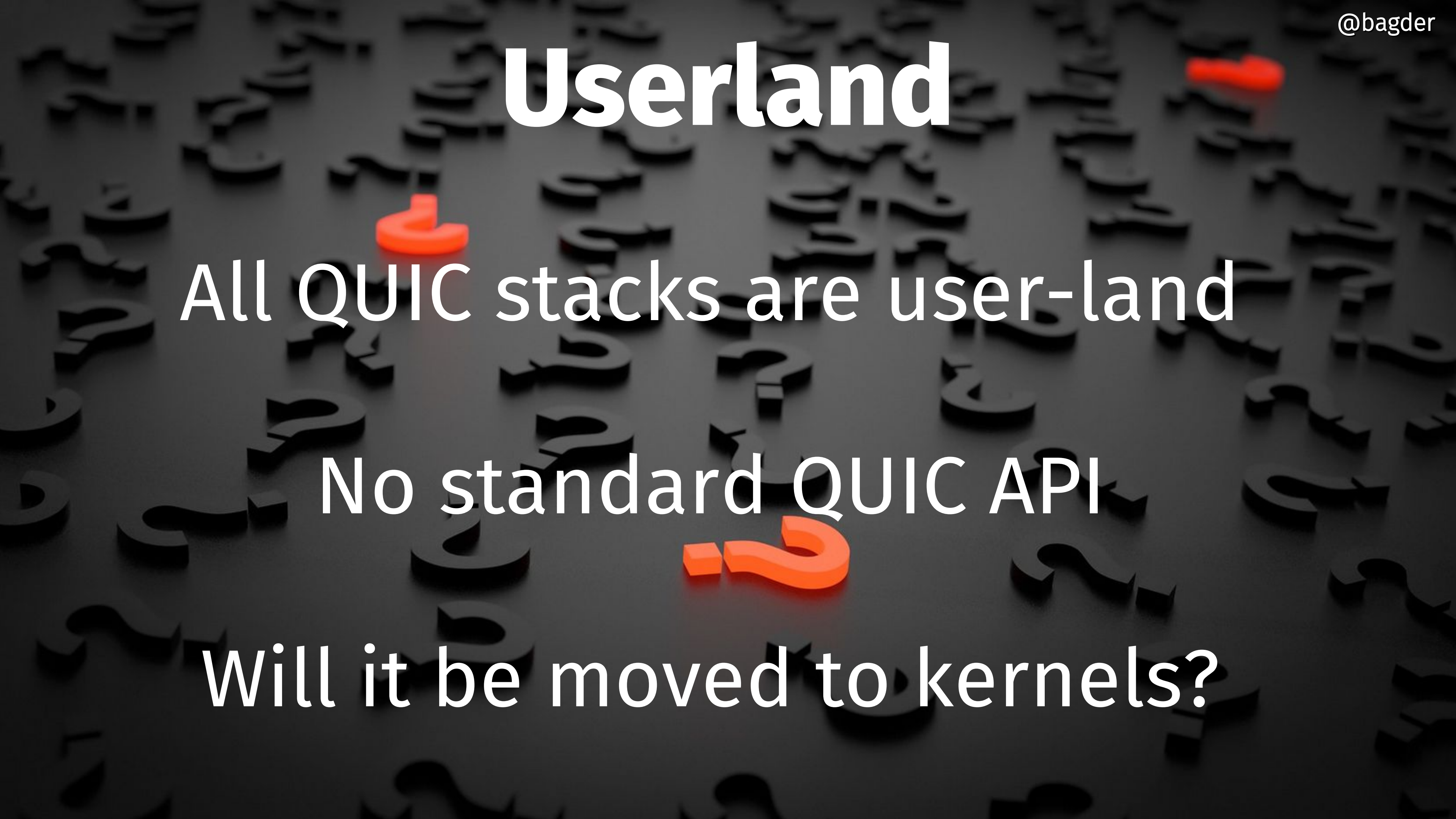


OpenSSL is the world's leading TLS library

OpenSSL postponed QUIC work to "after 3.0"

OpenSSL was an issue already for HTTP/2 deployment while further along

@bagder



Userland

All QUIC stacks are user-land

No standard QUIC API

Will it be moved to kernels?

Tooling

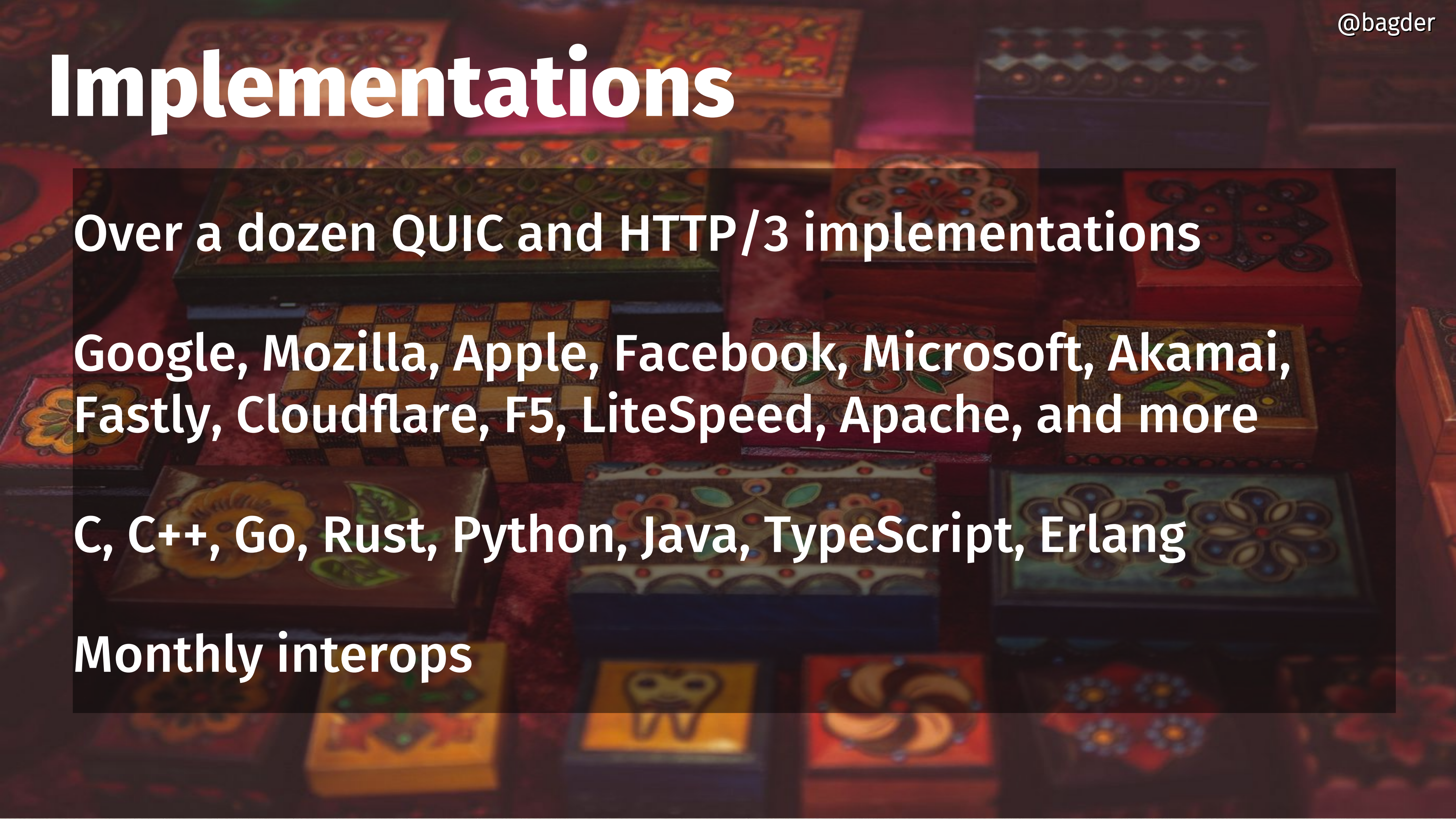
Needs new tooling

Hooray for WIRESHARK

qlog & qvis

Ship date

Early 2021?



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

@bagder

HTTP/3 Implementation Status

curl

Chrome and Edge Canary, Firefox Nightly, Safari 14 Beta

Caddy and LiteSpeed

NGINX “tech preview”

nginx-patch + quiche

BoringSSL and GnuTLS

Wireshark

No Apache httpd

No IIS

No OpenSSL (PR #8797)

Browsers doing HTTP/3

```
--enable-quic
--quic-version=h3-29
```

```
about:config
network.http.http3.enabled
```

Settings > Advanced > Experimental WebKit Features > HTTP3

CAN

Sites on HTTP/3 – *right now!*

Facebook
Instagram
Google
Youtube
Cloudflare

https://bagder.github.io/HTTP3-test/

HTTP/3 in curl://

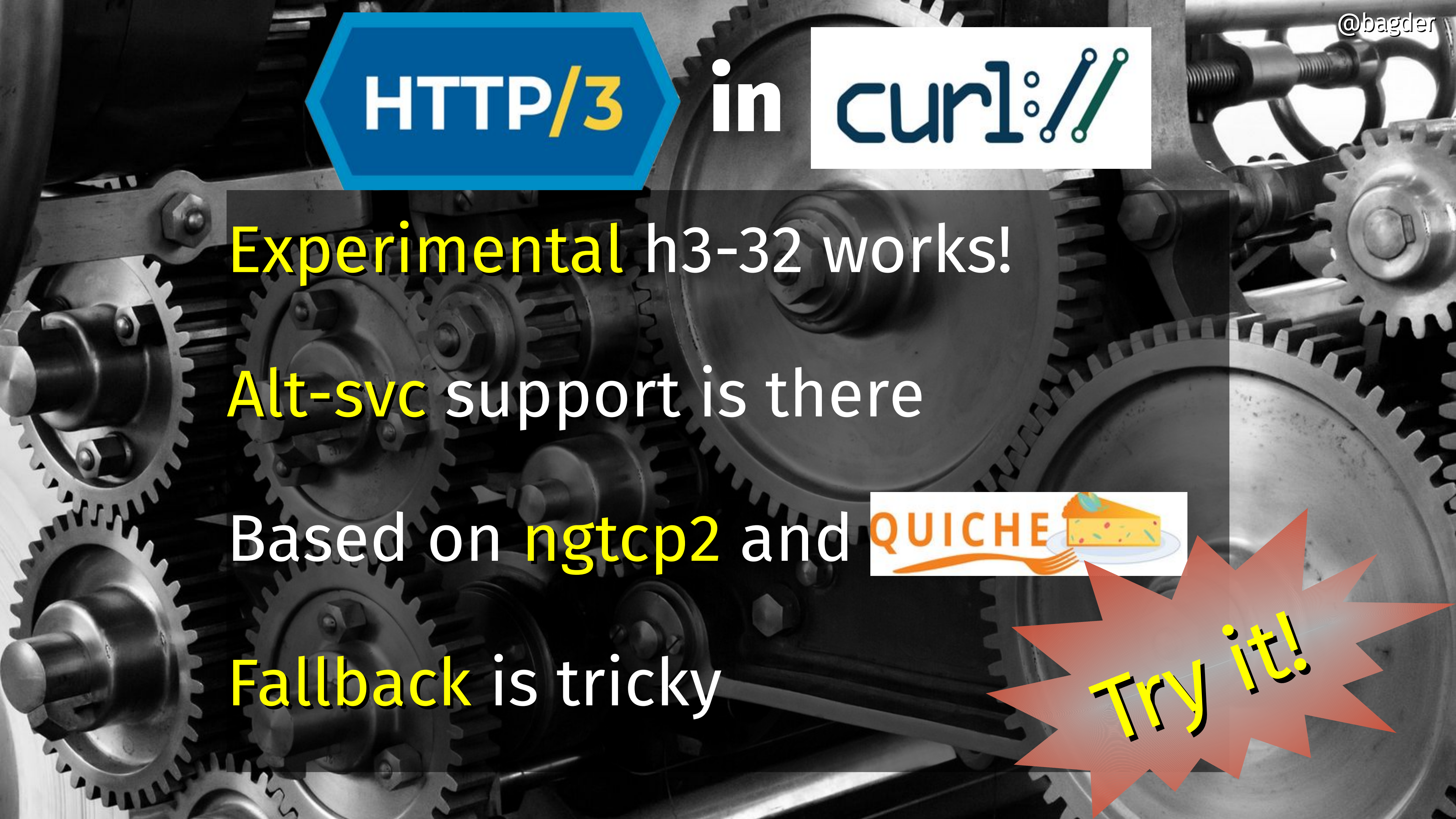
Experimental h3-32 works!

Alt-svc support is there

Based on ngtcp2 and QUICHE

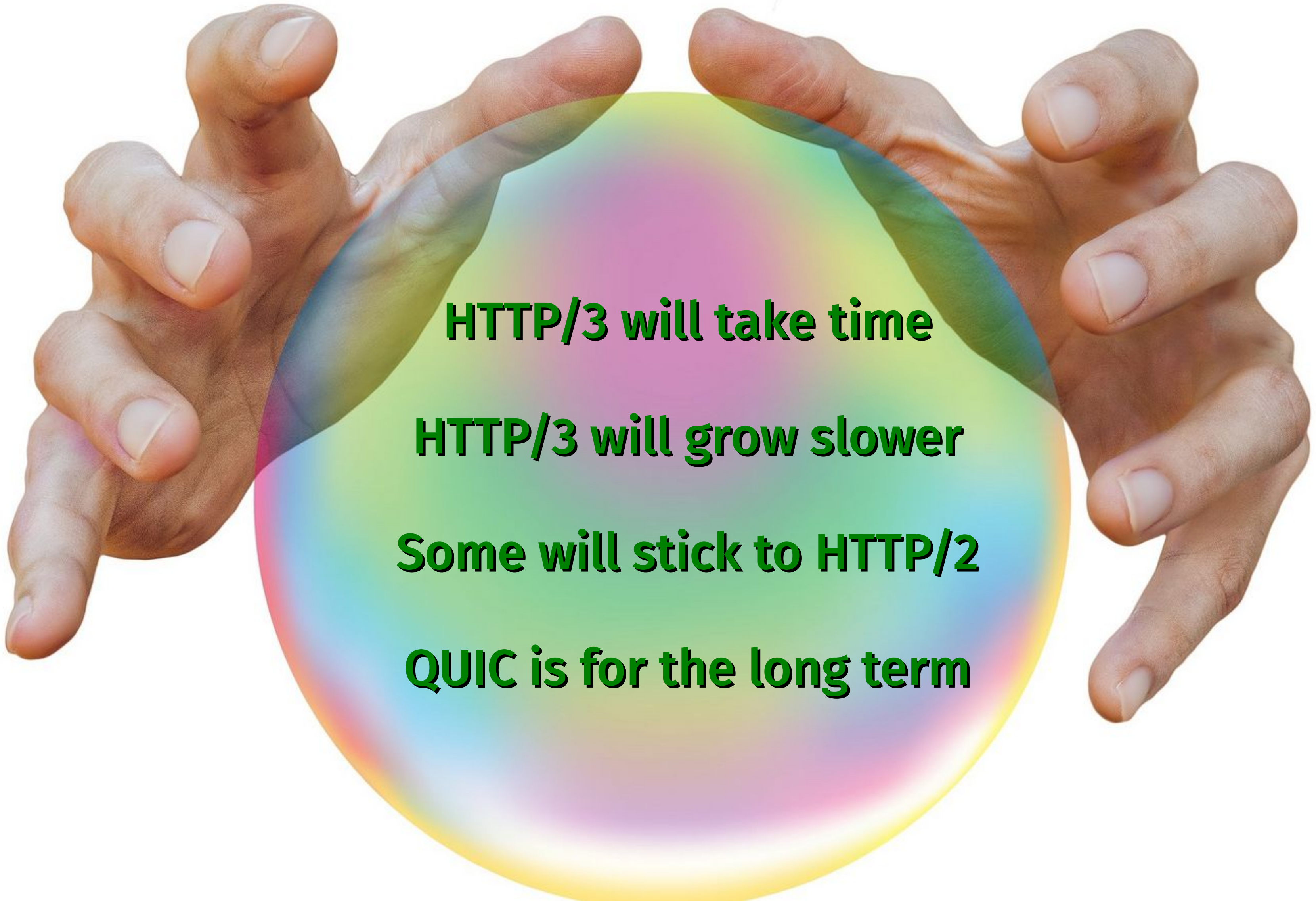
Fallback is tricky

Try it!



curl HTTP/3 command line

```
$ curl --http3 https://example.com/
HTTP/3 200
date: Wed, 09 Oct 2019 11:16:06 GMT
content-type: text/html
content-length: 10602
set-cookie: crazy=d8bc7e7; expires=Thu, 08-Oct-22
11:16:06 GMT; path=/; domain=example.com;
alt-svc: h3-32=":443"; ma=86400
```



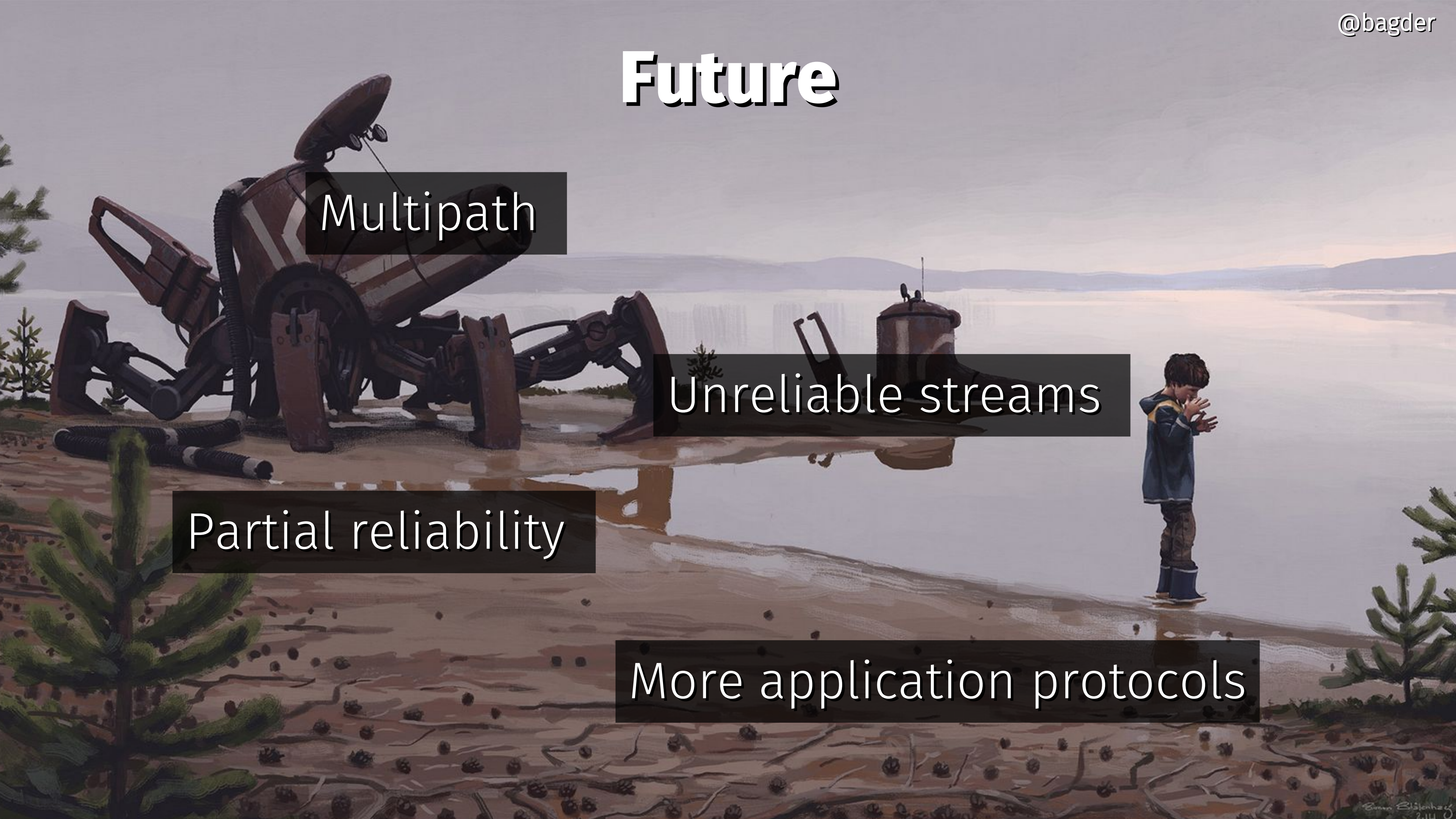
HTTP/3 will take time

HTTP/3 will grow slower

Some will stick to HTTP/2

QUIC is for the long term

Future



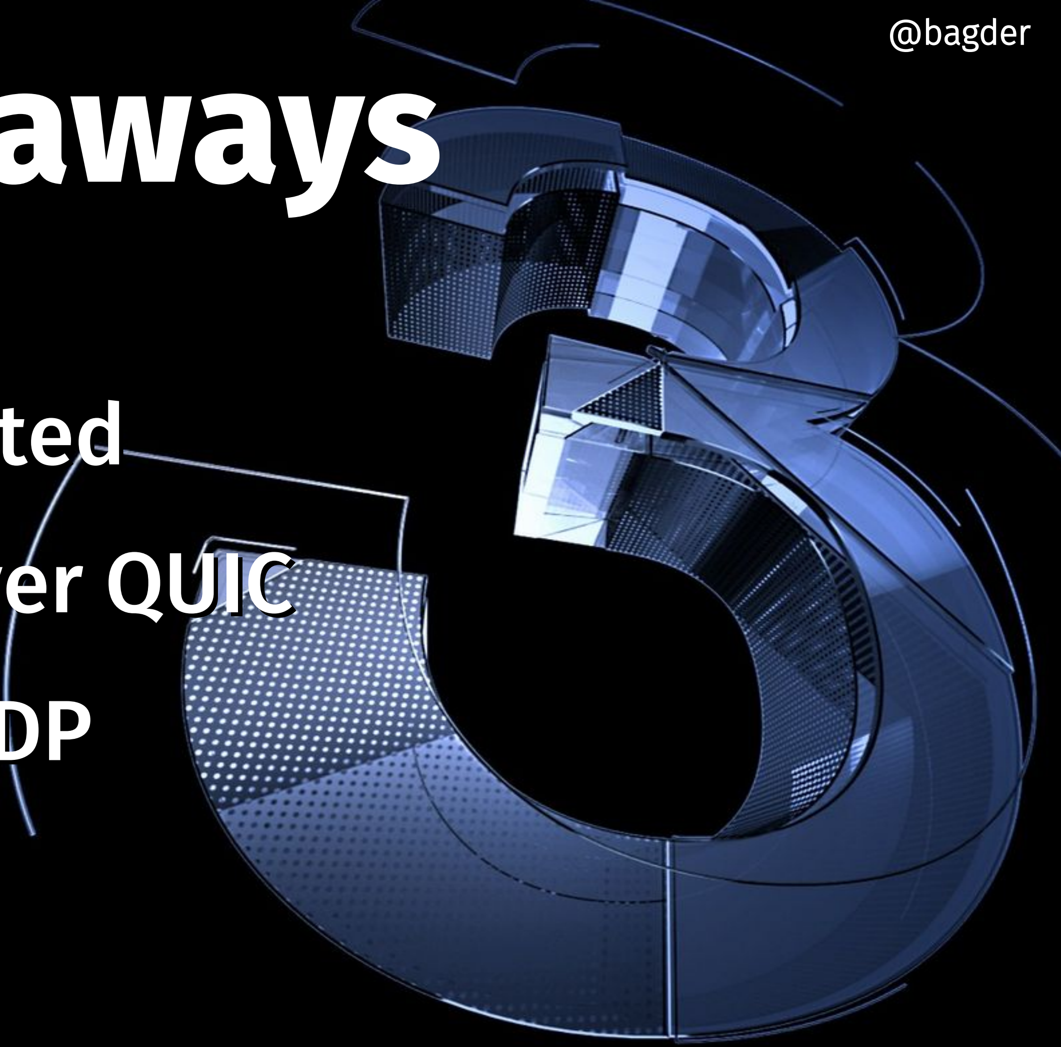
Multipath

Unreliable streams

Partial reliability

More application protocols

Take-aways



HTTP/3 is coming

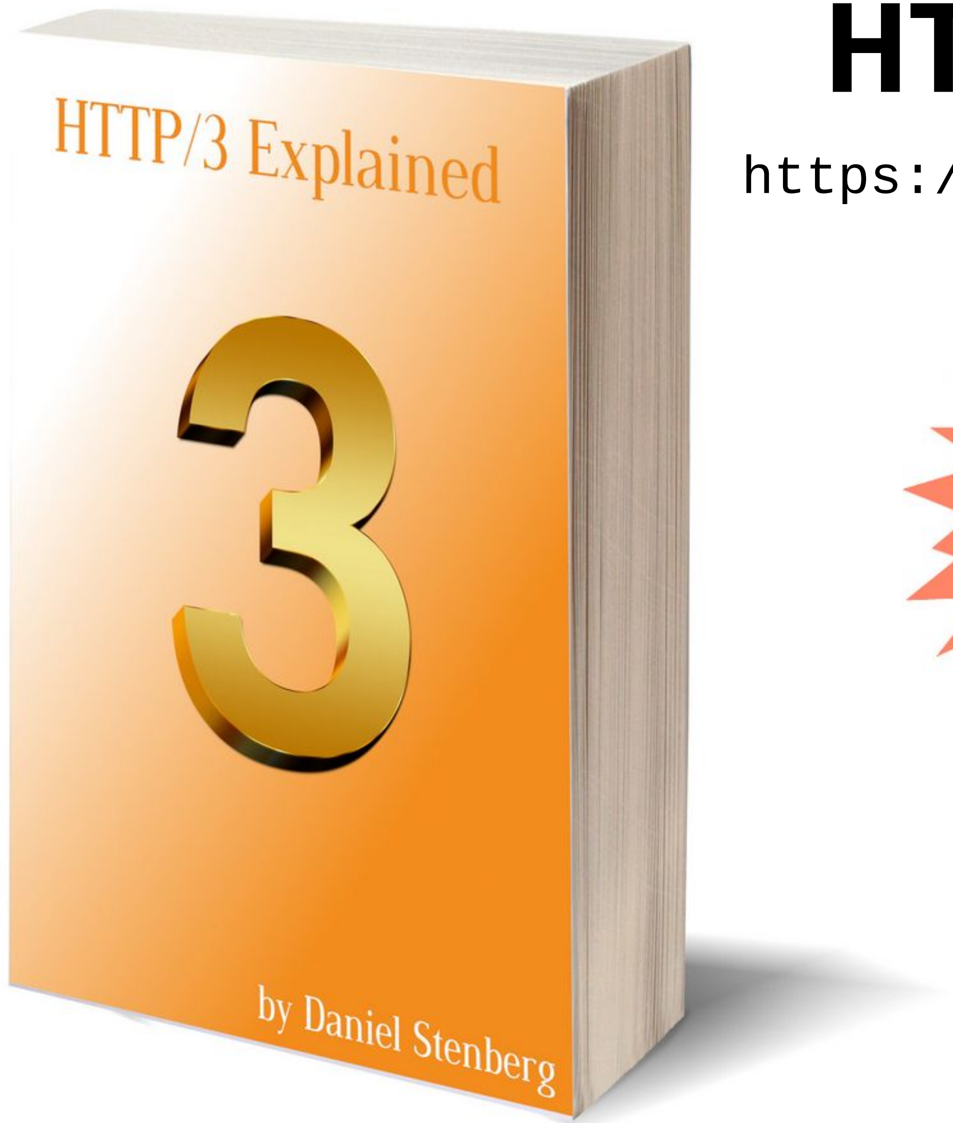
HTTP/3 is always encrypted

Similar to HTTP/2 but over QUIC

QUIC is transport over UDP

Challenges to overcome

Early 2021?



HTTP/3 Explained

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

Thank you!

Questions?

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

@bagder

License

Links to data and more info

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

DATAGRAM: https://tools.ietf.org/html/draft-pauly-quic-datagram-05

QUIC multipath: https://tools.ietf.org/html/draft-deconinck-quic-multipath-03

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

Chrome is deploying HTTP/3 and IETF QUIC: https://blog.chromium.org/2020/10/chrome-is-deploying-http3-and-ietf-quic.html

How Facebook is bringing QUIC to billions: https://engineering.fb.com/2020/10/21/networking-traffic/how-facebook-is-bringing-quic-to-billions/

Web Transport: https://w3c.github.io/webtransport/

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

Nginx + quiche: https://github.com/cloudflare/quiche/tree/master/extras/nginx

HTTPSSVC: https://tools.ietf.org/html/draft-ietf-dnsop-svcb-httpssvc-03

qlog: https://github.com/quiclog/internet-drafts

qvis: https://qvis.edm.uhasselt.be

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