fastly

The Smartest CDN on the Planet

Streaming Media and API Acceleration @ Fastly

Fastly Overview

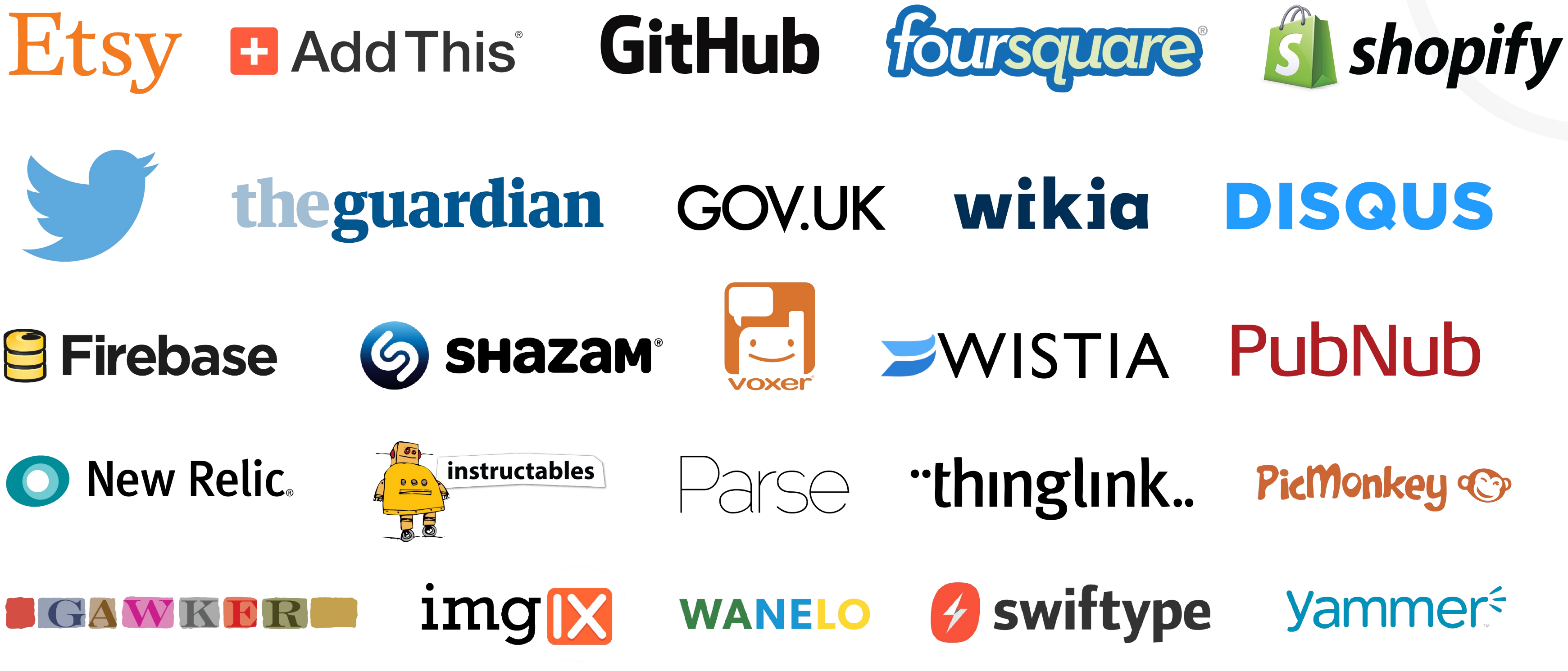
Not Invented Here

- Artur Bergman at Wikia, annoyed with 90s CDN
- Decides to build his own using Varnish
- 90s CDN goes down, his own takes the load without problems
- Purging is very useful for Wiki pages
- No existing CDNs support instant purging
- Starts Fastly and builds CDN out to support multiple customers

Fastly Overview

Fastly	Delivering content since 2008
Funding	$12M
Management Team	Artur Bergman, CEO Tyler McMullen, CTO Bill Kaufmann, COO
Team includes Veterans of	Akamai, Google, Edgecast, Limelight, Digital Ocean, etc…
Headquarters	San Francisco, CA
Investors	August Capital, Battery Ventures, Amplify Partners & OATV
VC Order	Battery, August, Amplify and OATV

Customer Logos



What's New?

Single Page Apps

Benchmarking the New Front End (Velocity 2013)
By Emily Nakashima and Rachel Meyers (ModCloth)
@eanakashima & @rachelmyers
http://bit.ly/ostrichandyak (slides at the bottom)

Single Page Apps

- The Ostrich

Single Page Apps

- The Ostrich
- Single page with lots of Javascript
- All data through API
- Only replace parts of page that need replacing
- Performance hard to measure

So How Now Brown Cow?

So How Now Brown Cow?

Or “What good is a CDN now?”

So How Now Brown Cow?

Or “What good is a CDN now?”

- “Dynamic Site Acceleration”

So How Now Brown Cow?

Or “What good is a CDN now?”

- Connection: Keep-Alive

Speed of Light

299 792 458 m/s

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

real
ronto
Boston
New York
Philadelphia
Ashburn
Charlotte
nta
Dublin
London
Amsterdam
Paris
Madrid
Marseille
Zurich
Milan
Frankfurt
Prague
Berlin
Copenh
St
Vi
B

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

Berlin → Amsterdam → London → New York → Ashburn

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

Berlin → Amsterdam → London → New York → Ashburn

577 + 357 + 5576 + 350 = 6860 km

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

Berlin → Amsterdam → London → New York → Ashburn

577 + 357 + 5576 + 350 = 6860 km

34.3 milliseconds one way

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

Berlin → Amsterdam → London → New York → Ashburn

577 + 357 + 5576 + 350 = 6860 km

34.3 milliseconds one way, ~70ms roundtrip

fastly

Speed of Light

299 792 458 m/s

~200 000 000 m/s in silica (SiO_2)

Berlin → Amsterdam → London → New York → Ashburn

577 + 357 + 5576 + 350 = 6860 km

34.3 milliseconds one way, ~70ms roundtrip

Reality ~90-120ms

TCP

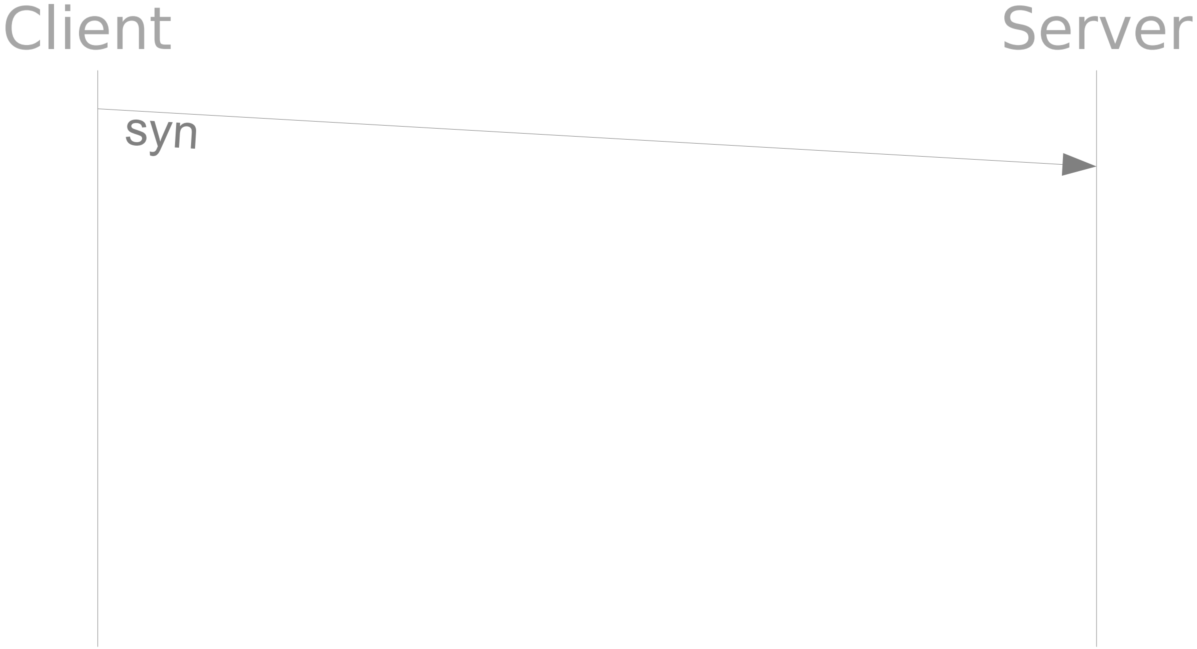
3-way handshake

Client

Server

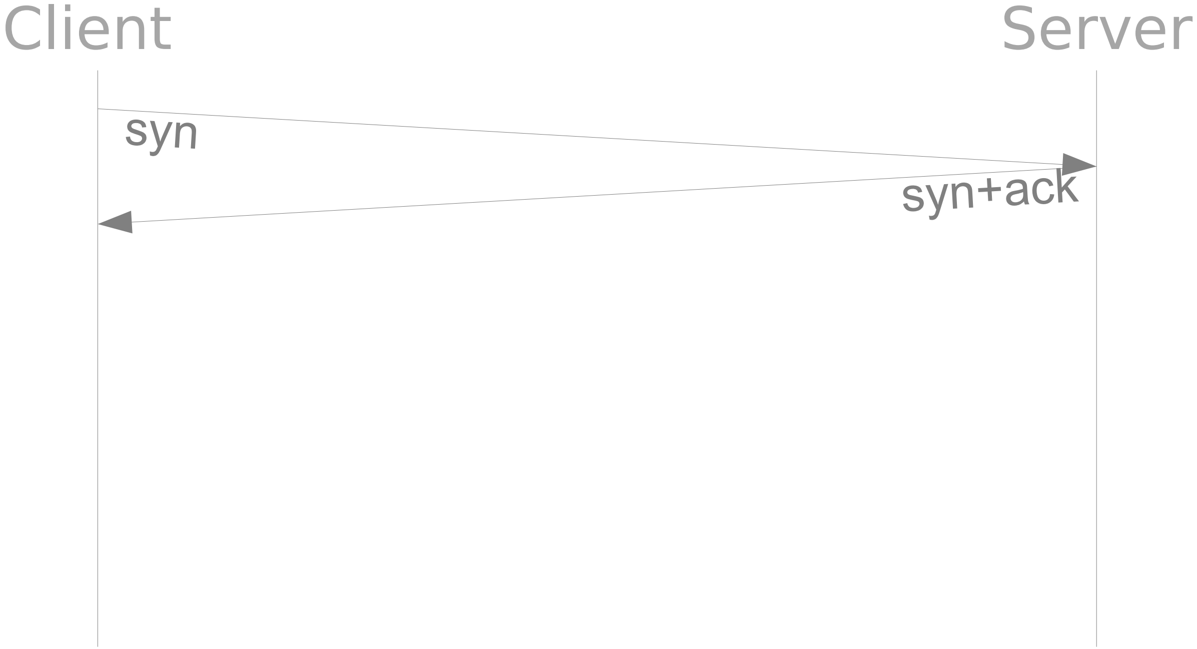
TCP

3-way handshake



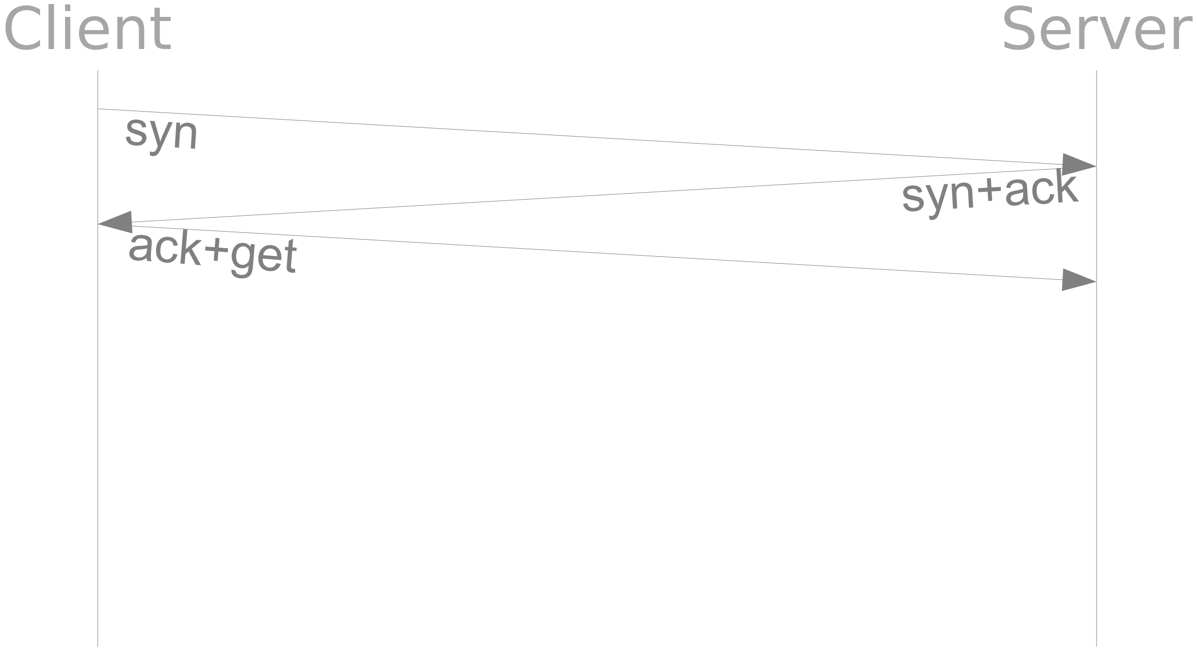
TCP

3-way handshake



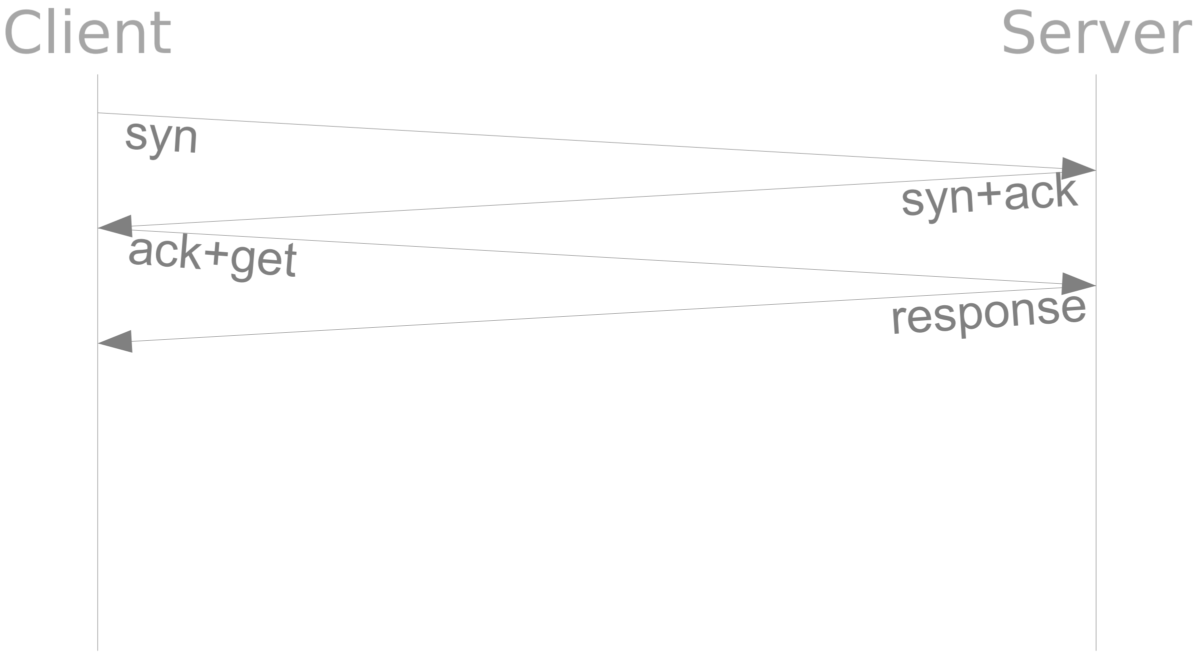
TCP

3-way handshake



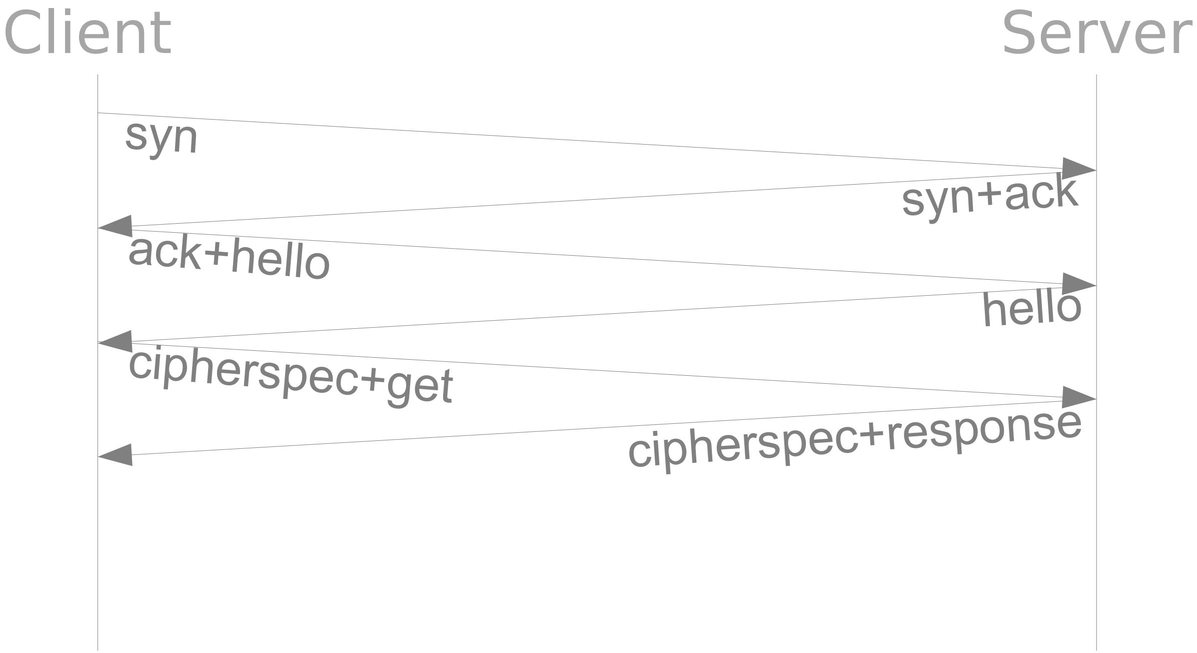
TCP

3-way handshake



TCP+TLS

3-way handshake + TLS handshake



Back and forth

3 roundtrips @ ~100ms each

300ms for one HTTP request

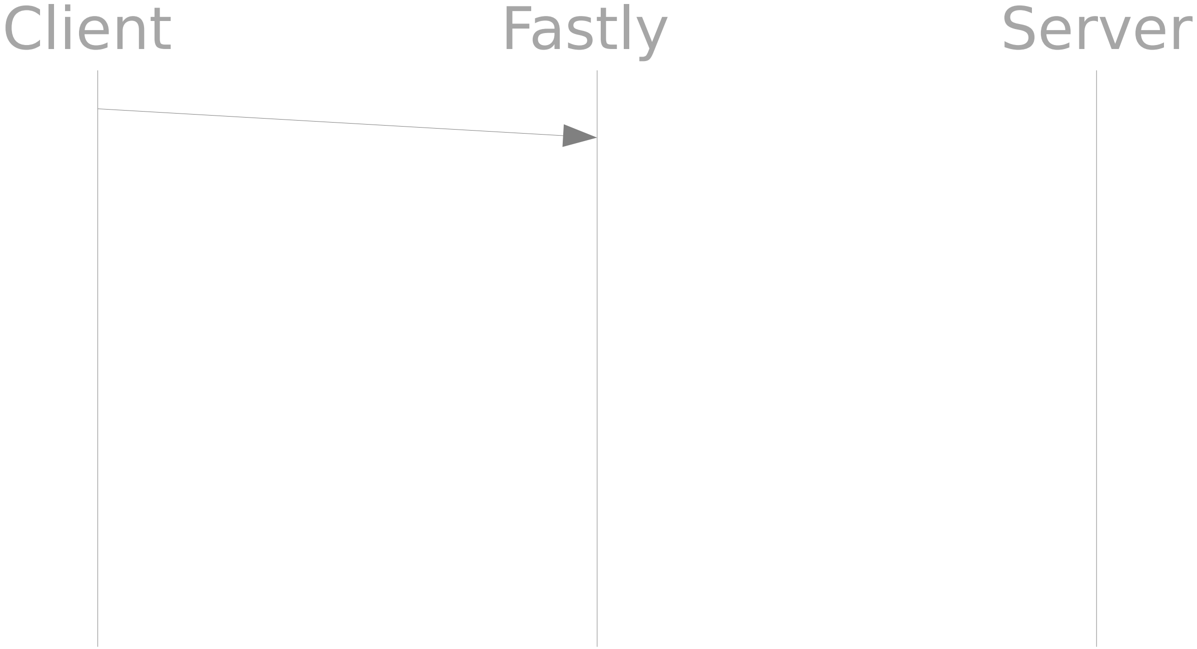
TCP+TLS+Fastly

3-way handshake + TLS handshake

Client Fastly Server

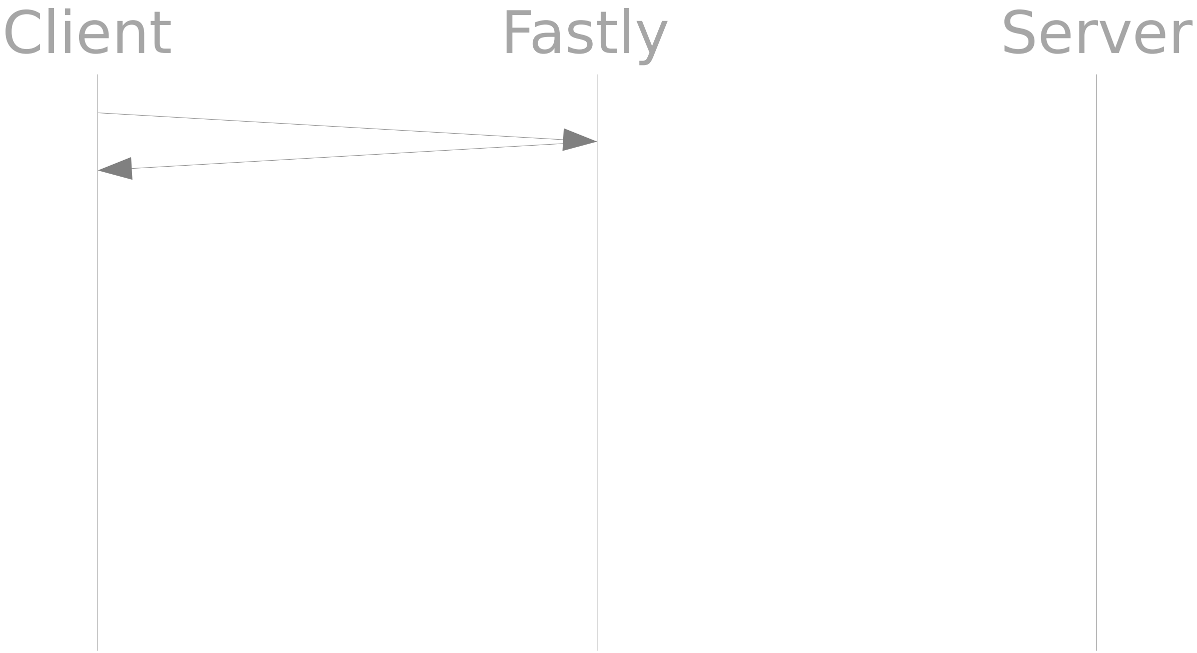
TCP+TLS+Fastly

3-way handshake + TLS handshake



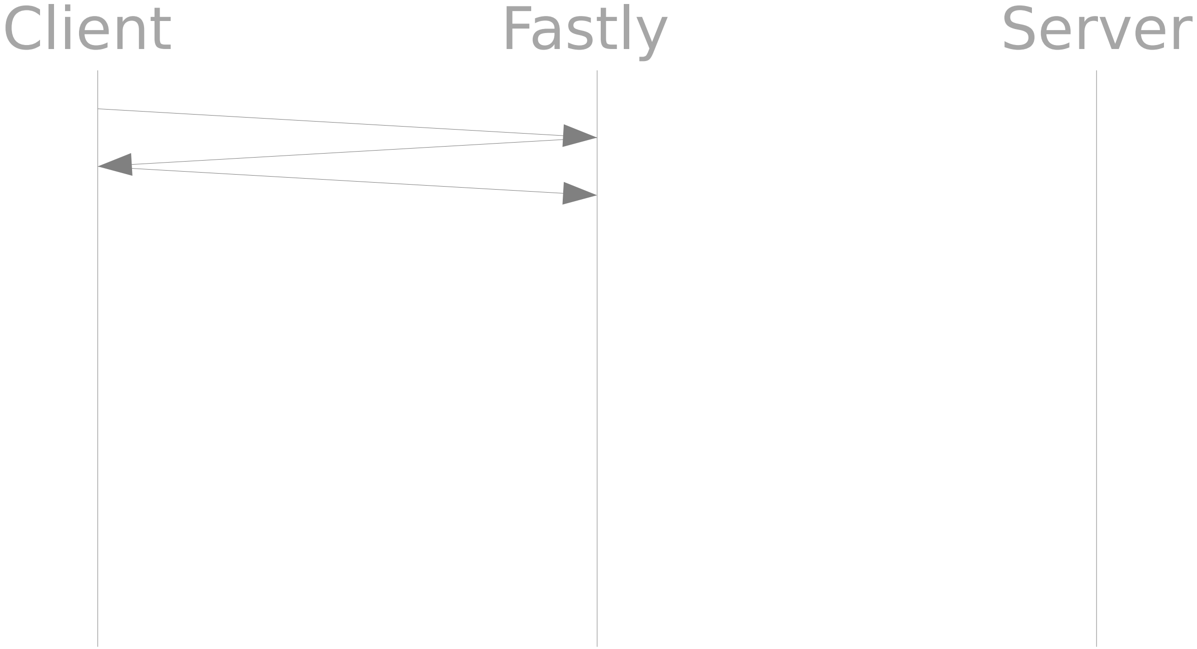
TCP+TLS+Fastly

3-way handshake + TLS handshake



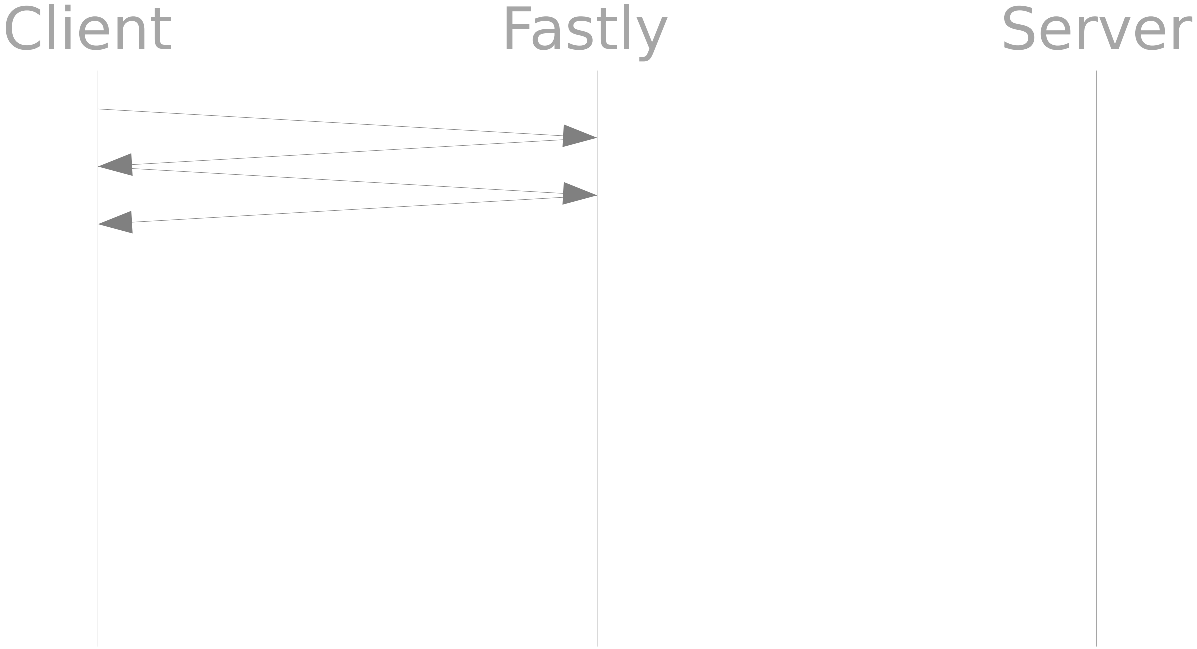
TCP+TLS+Fastly

3-way handshake + TLS handshake



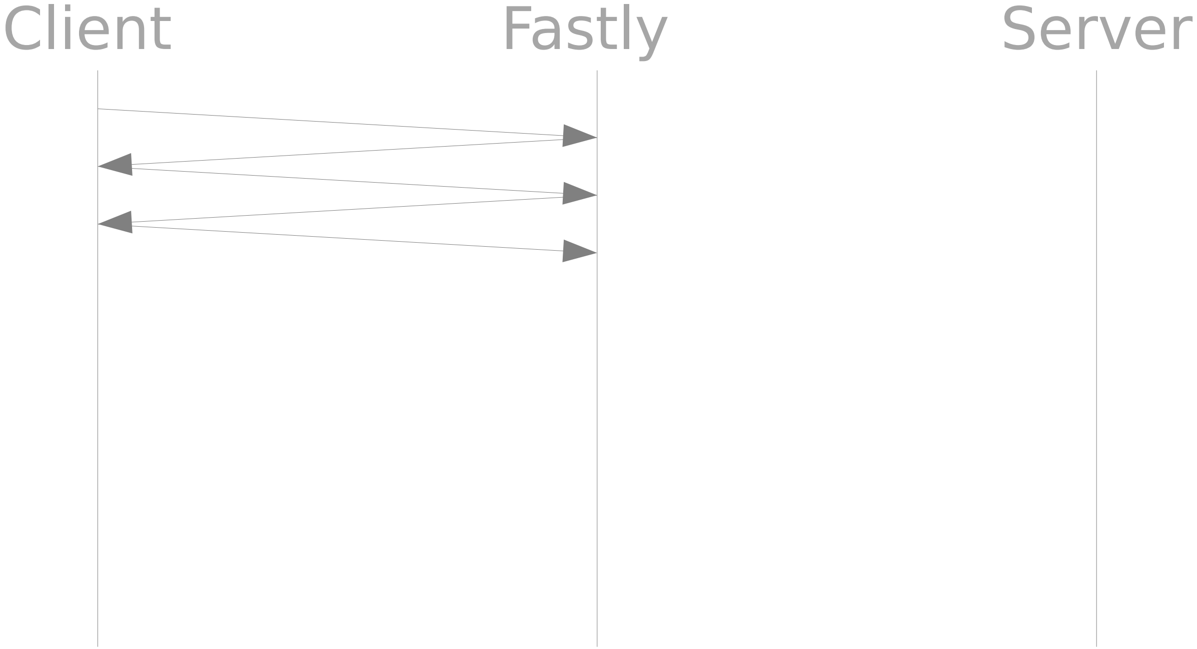
TCP+TLS+Fastly

3-way handshake + TLS handshake



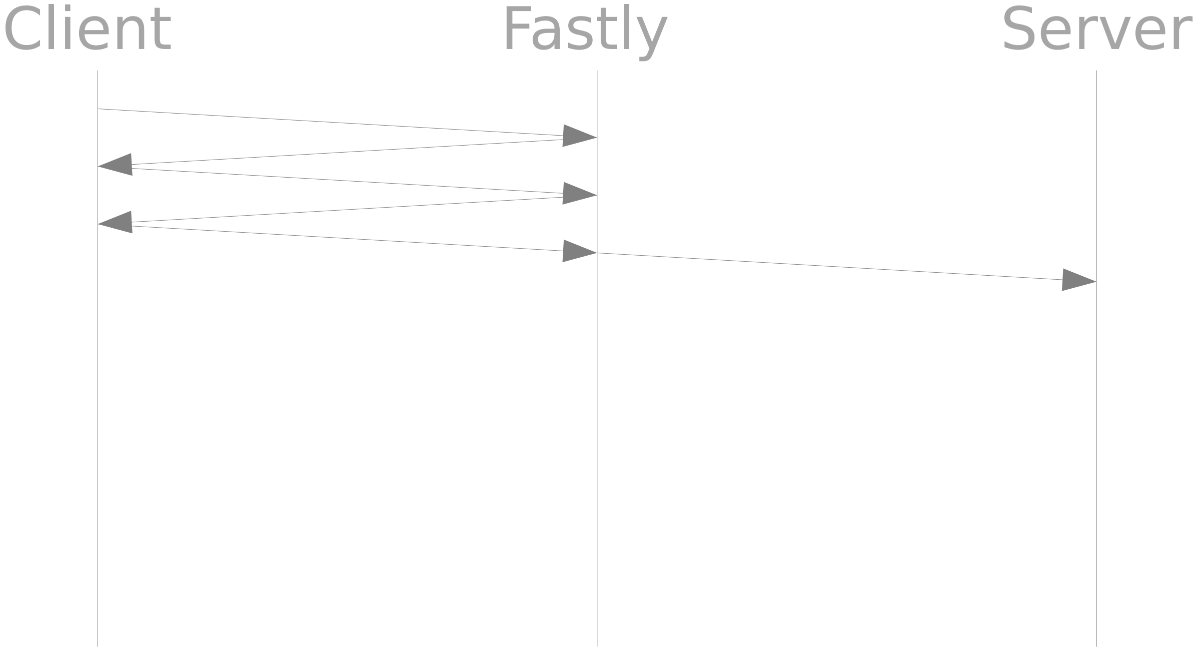
TCP+TLS+Fastly

3-way handshake + TLS handshake



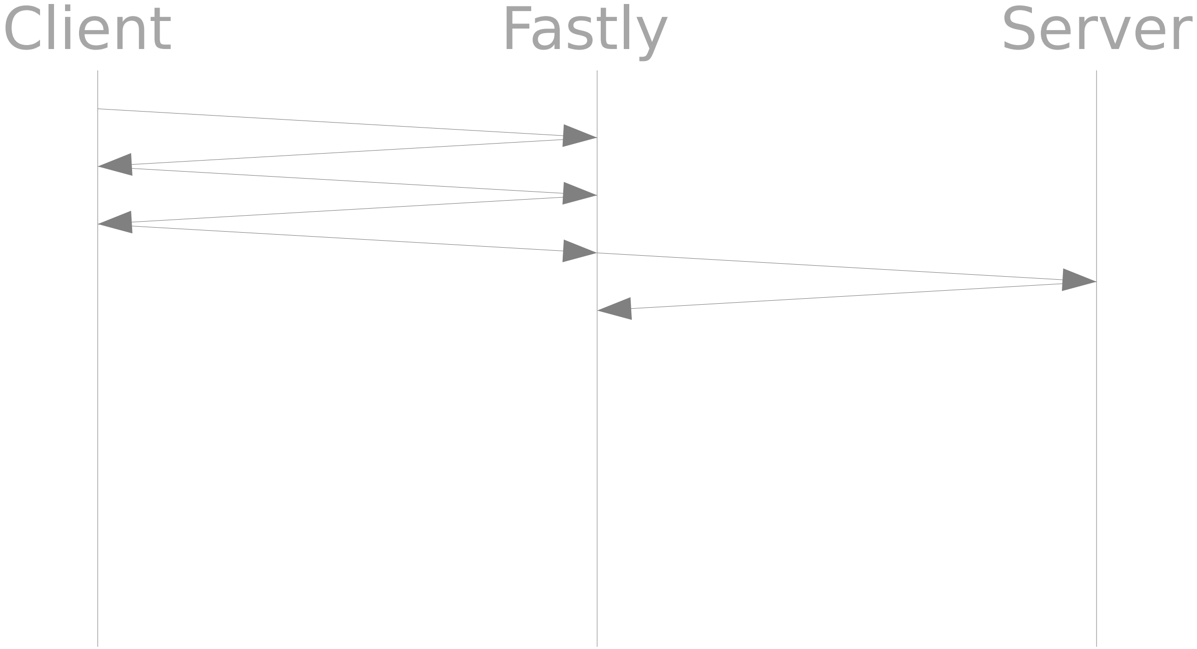
TCP+TLS+Fastly

3-way handshake + TLS handshake



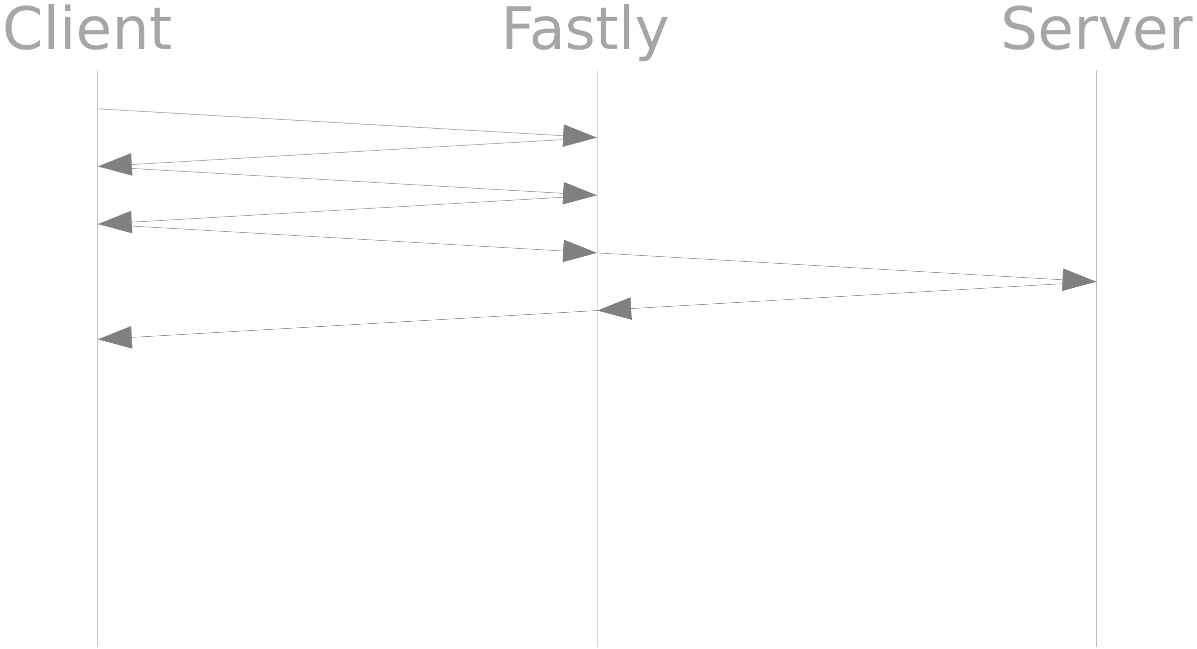
TCP+TLS+Fastly

3-way handshake + TLS handshake



TCP+TLS+Fastly

3-way handshake + TLS handshake



Back and forth

3 roundtrips @ ~20ms each

1 roundtrip @ ~80ms

140ms for one HTTP request

So How Now Brown Cow?

Or “What good is a CDN now?”

- Connection: Keep-Alive

So How Now Brown Cow?

Or “What good is a CDN now?”

- Connection: Keep-Alive
- Cache your API

Caching your API

- Short TTLs help
- Long TTL + Purge
- Use GET for idempotent requests
- Use req.postbody (Fastly)

Short TTLs help

raw.github.com

- <3
- Insane spikes
- Many 10s of Gbits/sec 40-80 x baseline spikes
- Lasting for 2-3 seconds
- Even short TTL completely offloads origin
- 5s TTL
- Longer TTL for “abusers”
- 80+% hitratio

Long TTL + Purge

- Just like regular web pages
- Use Surrogate-Keys (secondary hash) with Fastly
- Use regex bans with vanilla Varnish
- Get 90-95% hit ratio

Idempotence

Idempotence (/ˌaɪdɨmˈpoʊtəns/ eye-dəm-poh-təns) is the property of certain operations in mathematics and computer science, that can be applied multiple times without changing the result beyond the initial application.

Idempotence

Same request over and over makes no difference?

Idempotence

Same request over and over makes no difference?

Cacheable!

Idempotence

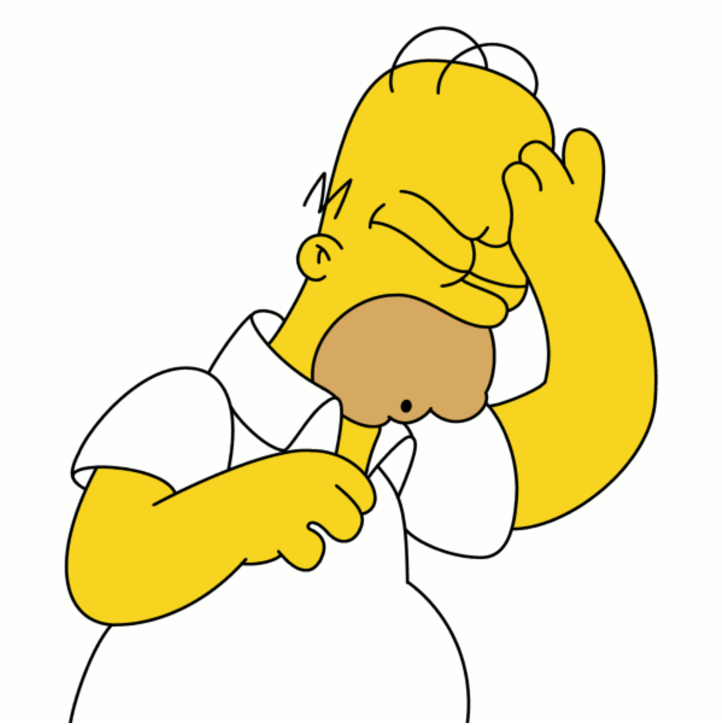
- Use GET for idempotent API requests
- Use POST for other API requests

Caching POST requests

“All API requests must be POST requests”

Caching POST requests

“All API requests must be POST requests”



Caching POST requests

```
sub vcl_recv {
…
  if (req.request == "POST"
      && req.postbody ~ "(^|&)action=list(&|$)") {
    return (lookup);
  }
…
}
```

Caching POST requests

```
sub vcl_hash {
  # Sorry about the 2.1 syntax
  set req.hash += req.url;
  set req.hash += req.http.host;
  if (req.request == "POST" && req.postbody) {
    set req.hash += req.postbody;
  }
  return (hash);
}
```

Caching POST requests

Requirements:

Caching POST requests

Requirements:

- Content-Type: application/x-www-form-urlencoded

Caching POST requests

Requirements:

- Content-Type: application/x-www-form-urlencoded

  action=list&category=xmas%20sweaters

Caching POST requests

Requirements:

- Content-Type: application/x-www-form-urlencoded

  action=list&category=xmas%20sweaters

- Max 2048 bytes request body

Video Streaming

“Whatever you do,
stay away from Video”

Old school

RTMP:

- Developed by Adobe for Flash
- One connection, one big stream of data
- Nightmare to deal with

New school

HLS (HTTP Live Streaming)

New school

HLS (HTTP Live Streaming)

HDS (HTTP Dynamic Streaming)

New school

HLS (HTTP Live Streaming)

HDS (HTTP Dynamic Streaming)

MPEG-DASH (Dynamic Adaptive Streaming over HTTP)

New school

HLS (**HTTP** Live Streaming)

HDS (**HTTP** Dynamic Streaming)

MPEG-DASH (Dynamic Adaptive Streaming over **HTTP**)

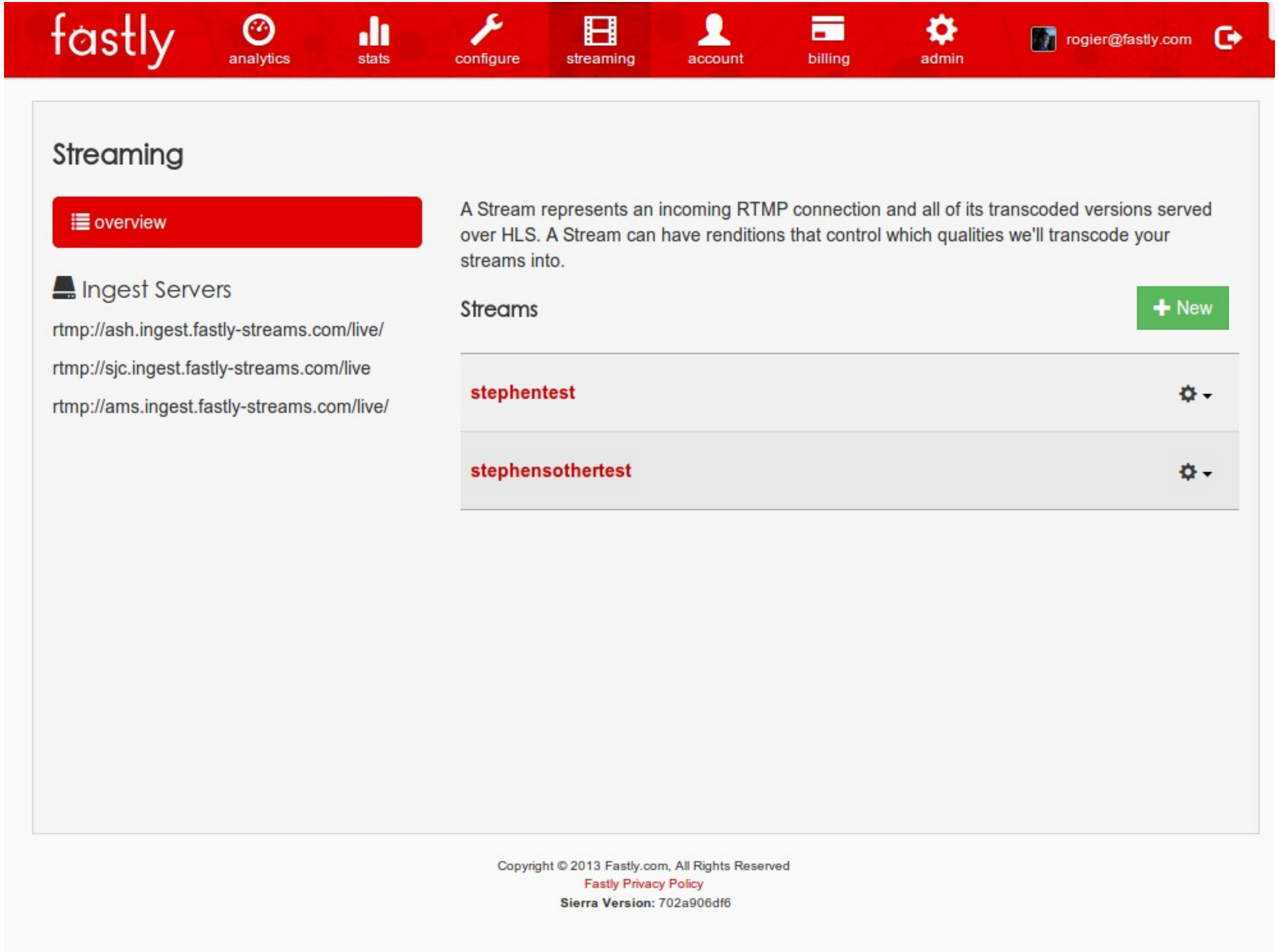
New school

- Chop stream up into chunks
- Serve each chunk over HTTP
- Serve playlist/manifest files over HTTP to list the chunks

Who is good at HTTP?

fastly

ShamWow!



Streaming

Ingest Servers

rtmp://ash.ingest.fastly-streams.com/live/

rtmp://sjc.ingest.fastly-streams.com/live

rtmp://ams.ingest.fastly-streams.com/live/

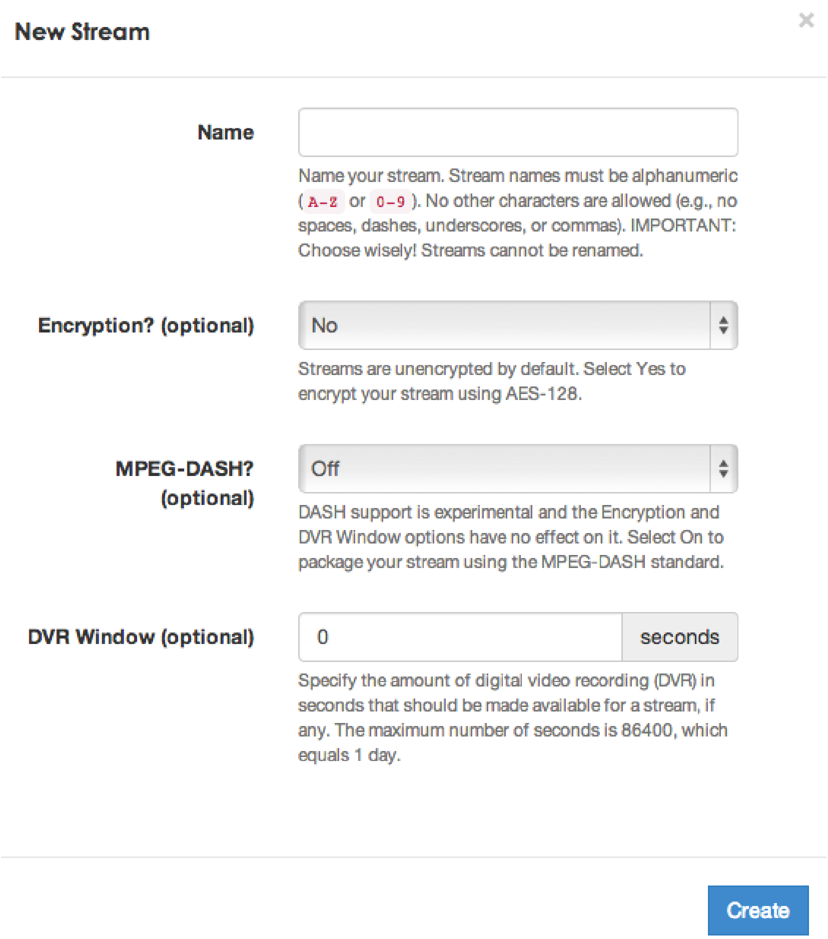
A Stream represents an incoming RTMP connection and all of its transcoded versions served over HLS. A Stream can have renditions that control which qualities we'll transcode your streams into.

Streams

stephentest

stephensothertest

Copyright © 2013 Fastly.com, All Rights Reserved
Fastly Privacy Policy
Sierra Version: 702a906df6



New Stream

Name

Name your stream. Stream names must be alphanumeric (`A-Z` or `0-9`). No other characters are allowed (e.g., no spaces, dashes, underscores, or commas). IMPORTANT: Choose wisely! Streams cannot be renamed.

Encryption? (optional): No

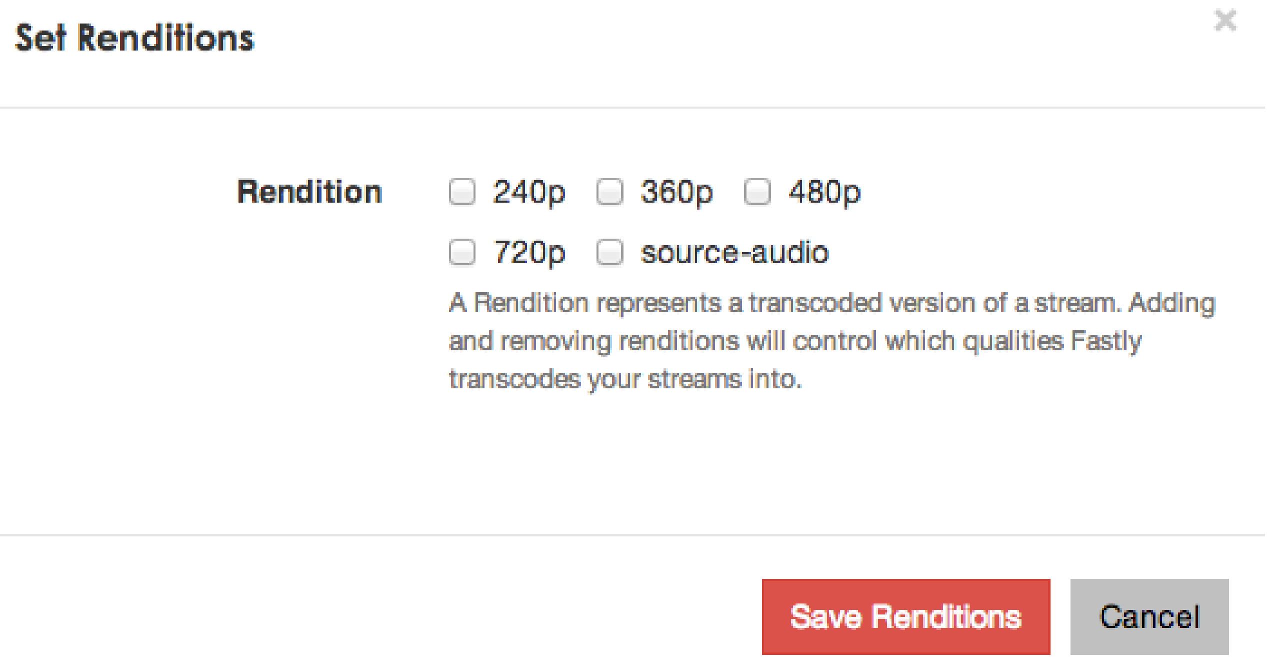
Streams are unencrypted by default. Select Yes to encrypt your stream using AES-128.

MPEG-DASH? (optional): Off

DASH support is experimental and the Encryption and DVR Window options have no effect on it. Select On to package your stream using the MPEG-DASH standard.

DVR Window (optional): 0 seconds

Specify the amount of digital video recording (DVR) in seconds that should be made available for a stream, if any. The maximum number of seconds is 86400, which equals 1 day.



Set Renditions

Rendition □ 240p □ 360p □ 480p □ 720p □ source-audio

A Rendition represents a transcoded version of a stream. Adding and removing renditions will control which qualities Fastly transcodes your streams into.

Questions?

- Please argue with Artur
- Sign up at www.fastly.com/signup
- Cache everything, control everything, beat the speed of light

Thank you

- Rogier Mulhuijzen
- DocWilco
- @drwilco
- Sign up at www.fastly.com/signup