



Computer Network

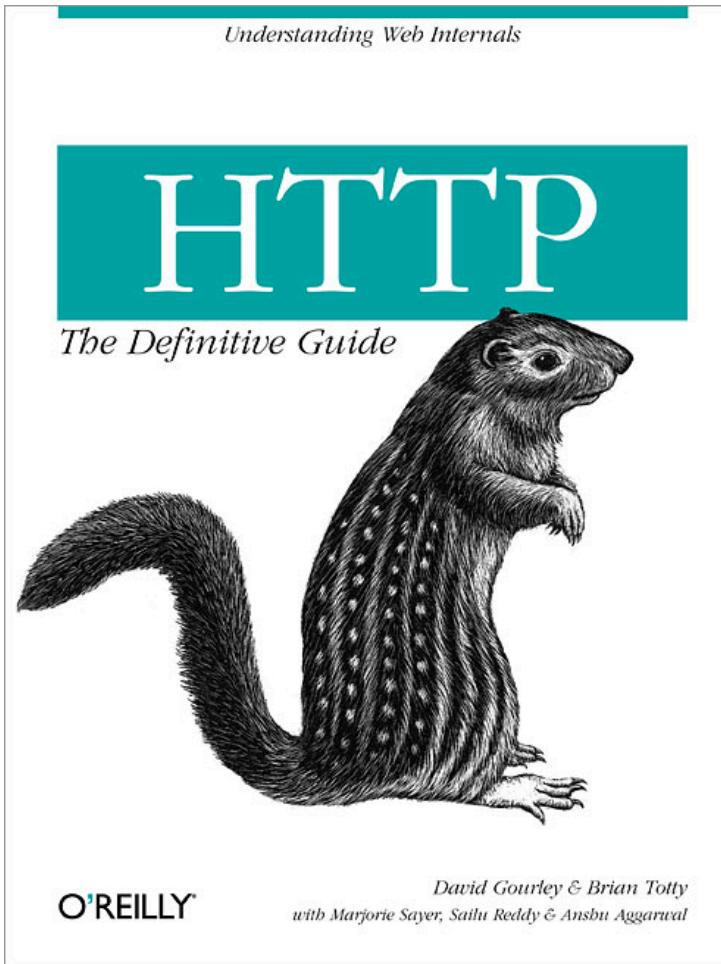
Lecture 10 HTTP & SIP

2023. 05. 11

Sungwon Lee
Department of Software Convergence

Textbook of HTTP Chapter

“HTTP는 책을 읽어 볼만 합니다. 본 챕터는 아래 책을 교재로 합니다”



HTTP: The Definitive Guide

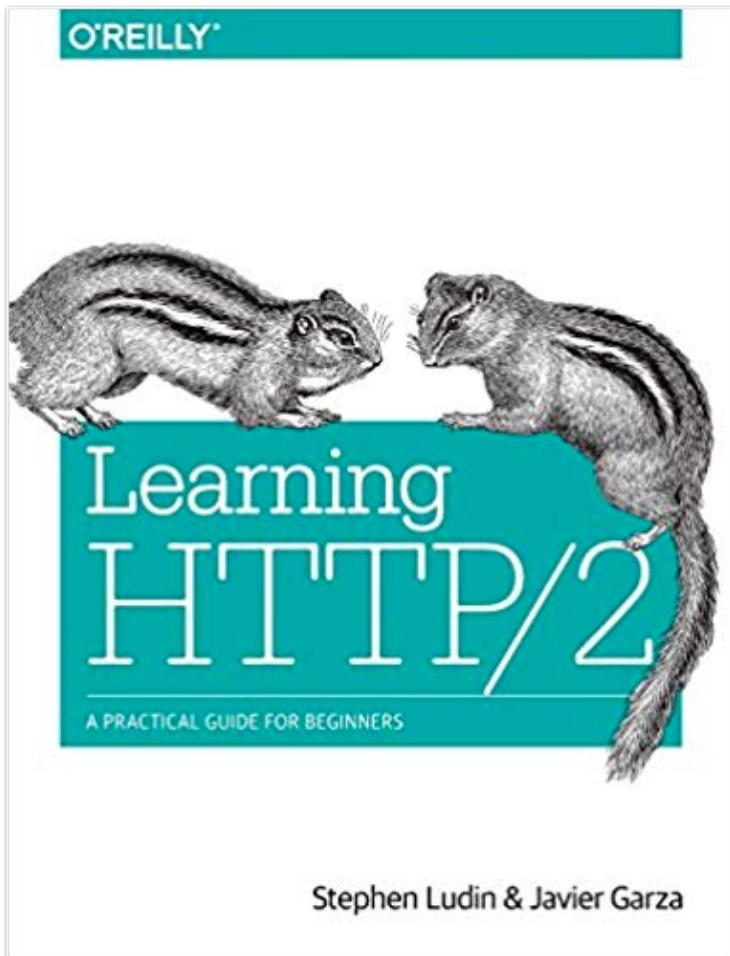
By Brian Totty, David
Gourley, Marjorie
Sayer, Anshu Aggarwal, Sailu
Reddy

Publisher: O'Reilly Media

Release Date: June 2009

Textbook of HTTP Chapter

“HTTP는 2.0이 확산 중이고, 3.0으로 넘어가고 있습니다”



Learning HTTP/2: A Practical
Guide for Beginners

By Stephen Ludin, and Javier
Garza.

Publisher: O'Reilly Media

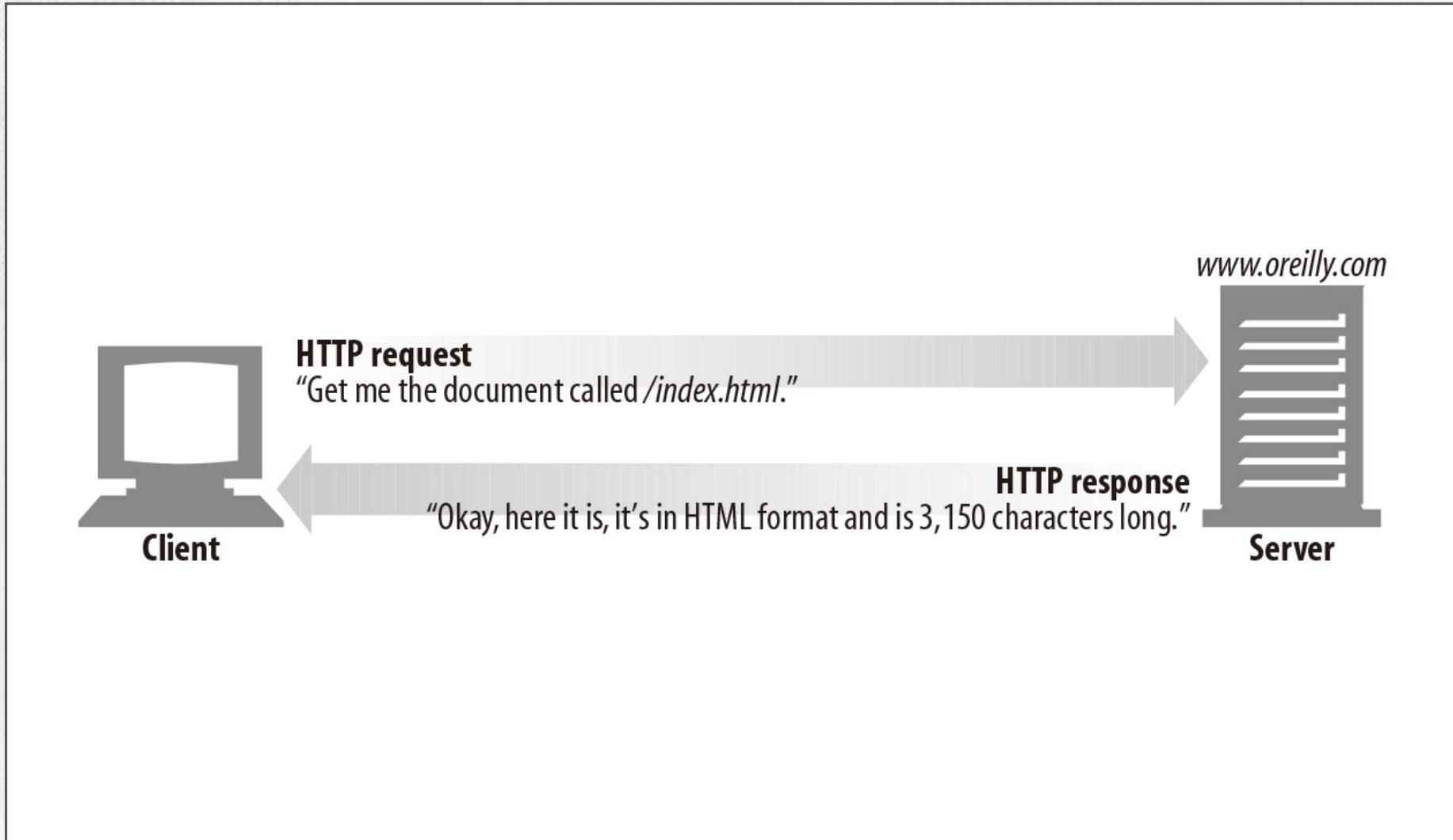
Release Date: June 2017

Contents

- **HTTP/1.1**
- HTTP/2
- HTTP/3
- SIP
- Advanced Web Technologies

Web Client and Servers

“Basic components of the WWW”

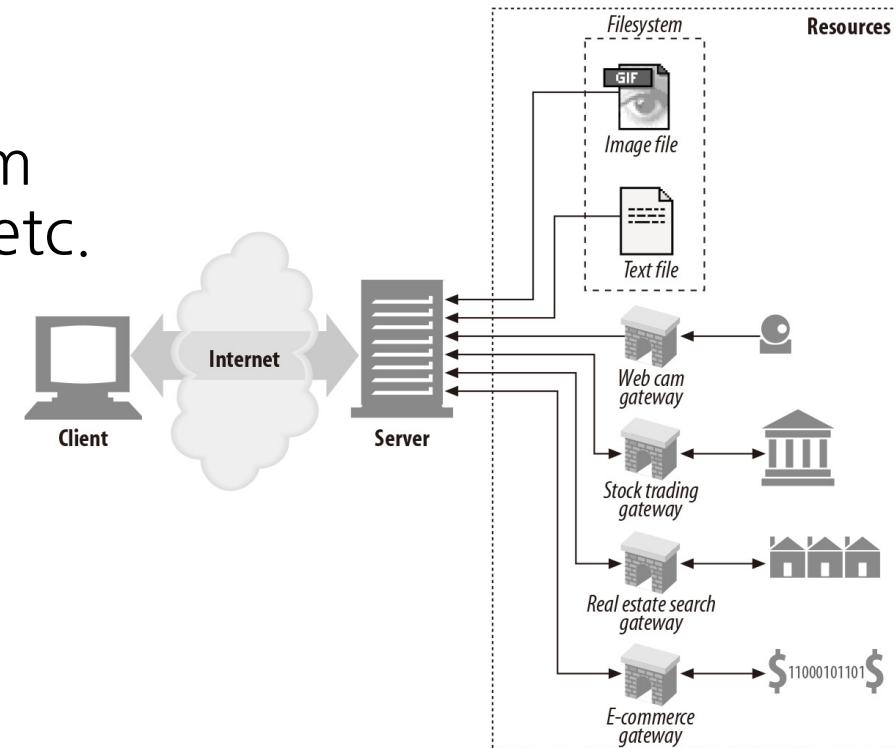


Resources

“A resource is any kind of content source”

- Static content
 - File system
 - MS word file, Image file, **Video Clip**, etc.

- Dynamic content
 - Generated by SW program
 - Live image, Trade stocks, etc.

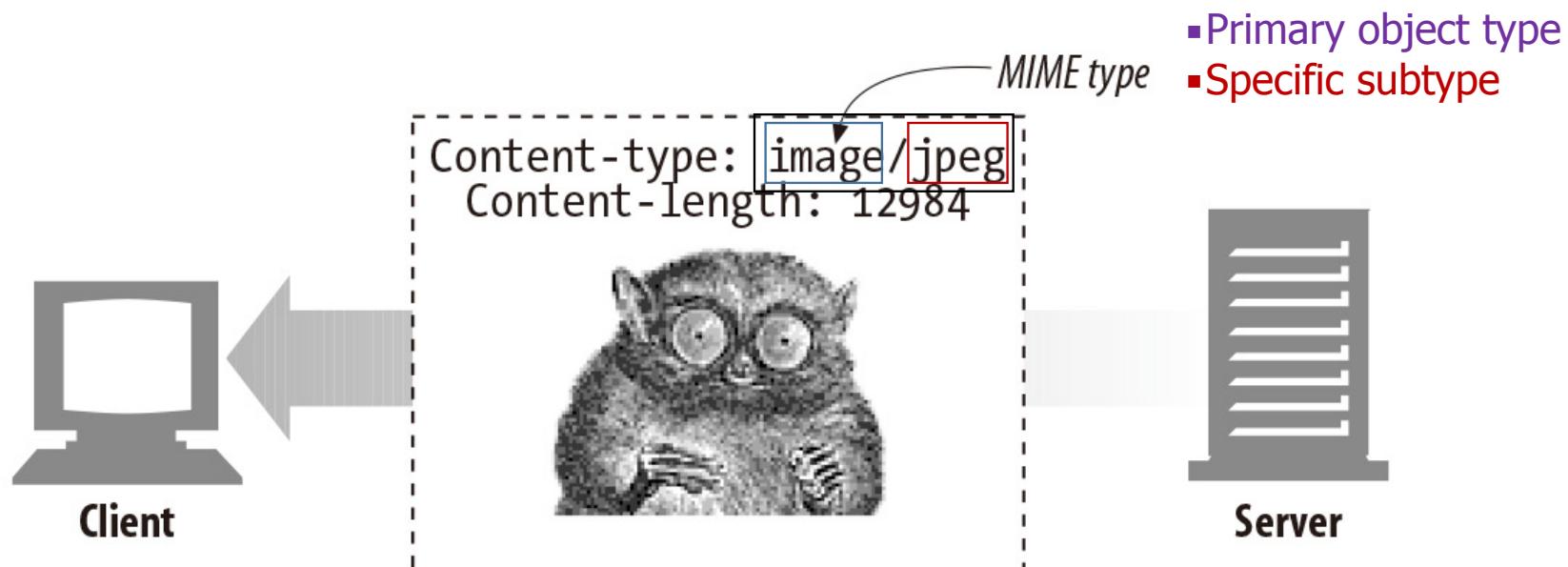


Resources

“A resource is any kind of content source”

□ Media Types

- MIME (Multipurpose Internet Mail Extensions)
- Textual label



Resources

“A resource is any kind of content source”

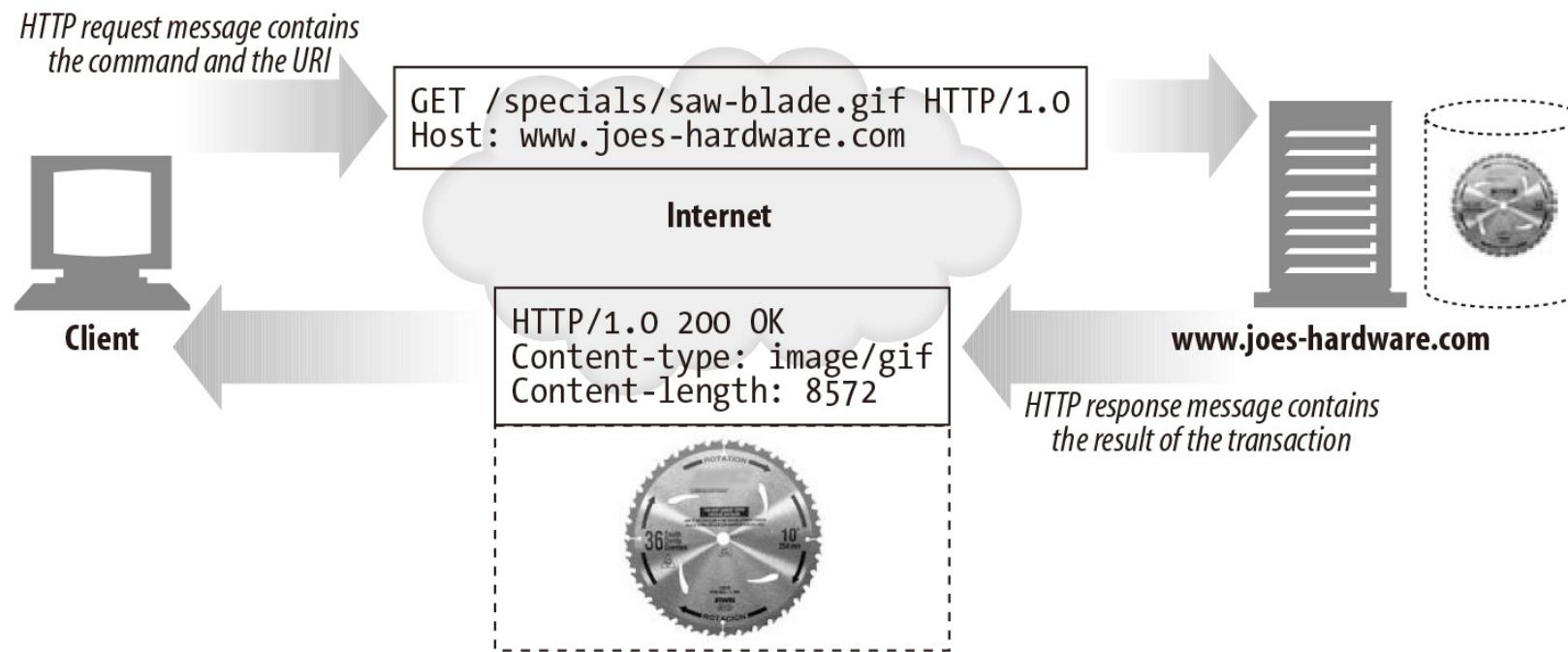
□ URI (Uniform Resource Identifier)

- Server resource name
 - ✓ Addresses of the Internet
 - ✓ Uniquely identifying
 - ✓ Locating information resource
- URL and URN

Transactions

“Requests and Responses between HTTP Client/Server”

- HTTP Messages
 - Request
 - Response



Transactions

“Requests and Responses between HTTP Client/Server”

- HTTP Methods
 - Supports several request commands
 - Server what action to perform

HTTP method	Description
GET	Send named resource from the server to the client.
PUT	Store data from client into a named server resource.
DELETE	Delete the named resource from a server.
POST	Send client data into a server gateway application.
HEAD	Send just the HTTP headers from the response for the named resource.

Transactions

“Requests and Responses between HTTP Client/Server”

- Status Codes
 - HTTP response contain status code
 - Three-digit numeric code
 - Explanatory textual “reason phrase”

HTTP status code	Description
200	OK. Document returned correctly.
302	Redirect. Go someplace else to get the resource.
404	Not Found. Can't find this resource.

Messages

“Requests and Responses between HTTP Client/Server”

- ❑ Request and Response
 - No other kinds of HTTP Messages
- ❑ Line-oriented sequences of characters
- ❑ Easy for humans to read and write

(a) Request message

```
GET /test/hi-there.txt HTTP/1.0
Accept: text/*
Accept-Language: en,fr
```

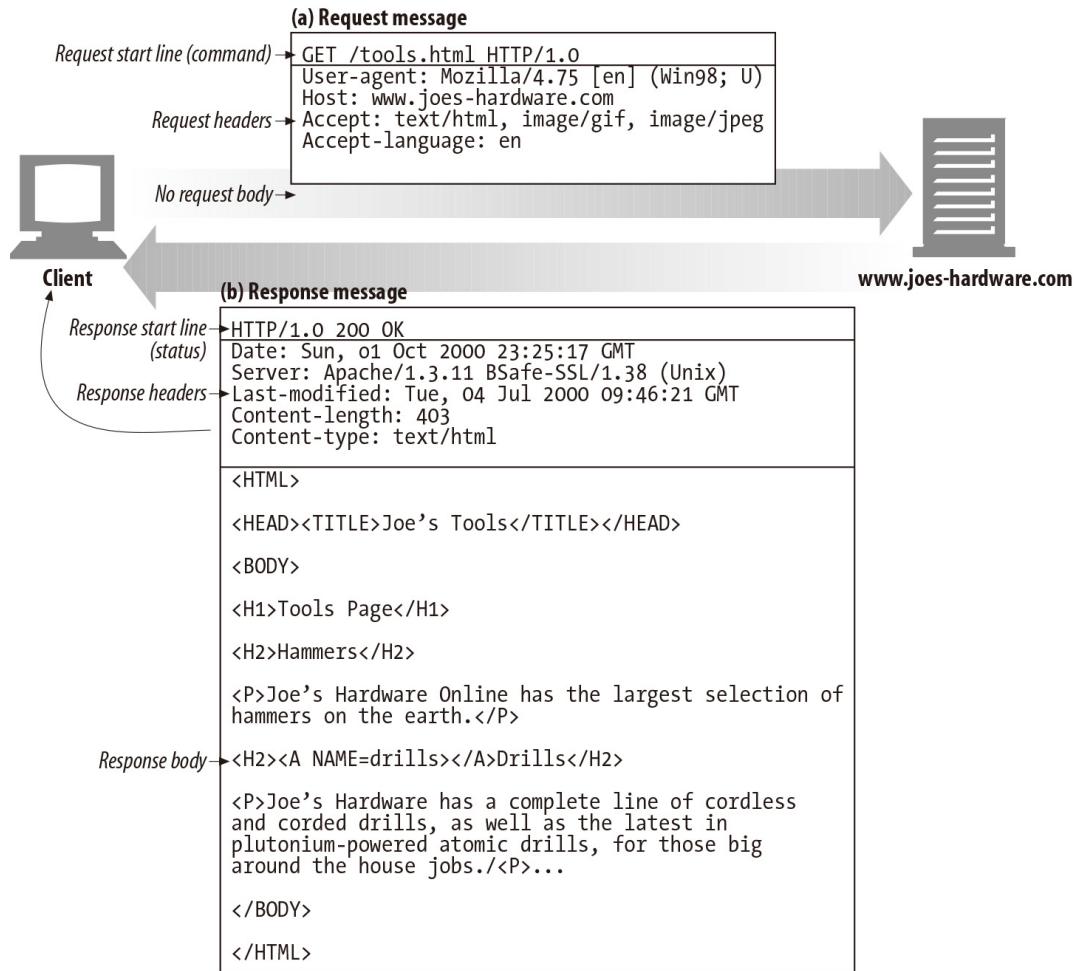
(b) Response message

<i>Start line</i>	HTTP/1.0 200 OK
<i>Headers</i>	Content-type: text/plain Content-length: 19
<i>Body</i>	Hi! I'm a message!

Messages

“Requests and Responses between HTTP Client/Server”

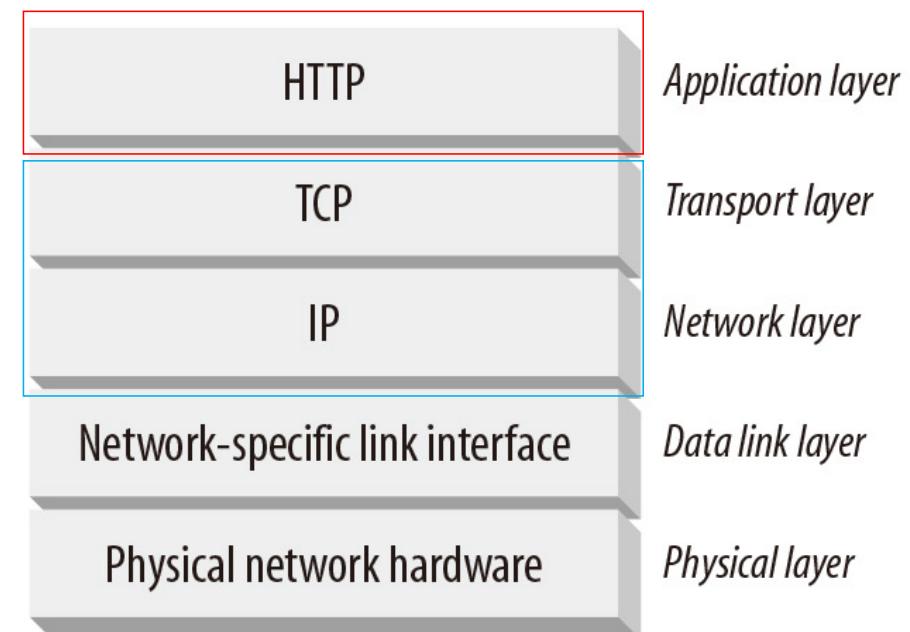
□ Example



Connections

“Transport layer under HTTP”

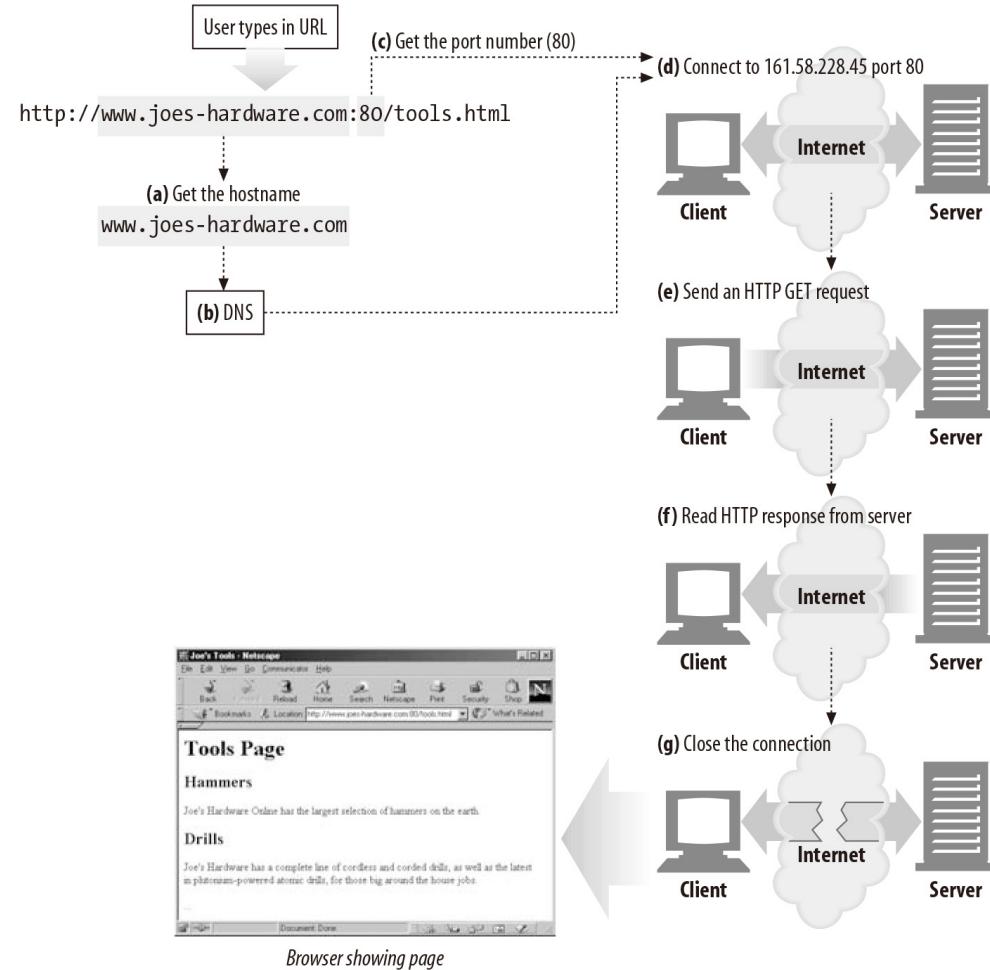
- TCP/IP
 - Reliable Data Transfer
 - In Order Transfer
 - Connection Oriented
 - Bidirectional Byte Stream
 - Flow control
 - Congestion control



Connections

“Transport layer under HTTP”

□ Connections Process



Methods

“Case study of each methods using example”

□ GET

- The most common method
- Ask a server to send a resource

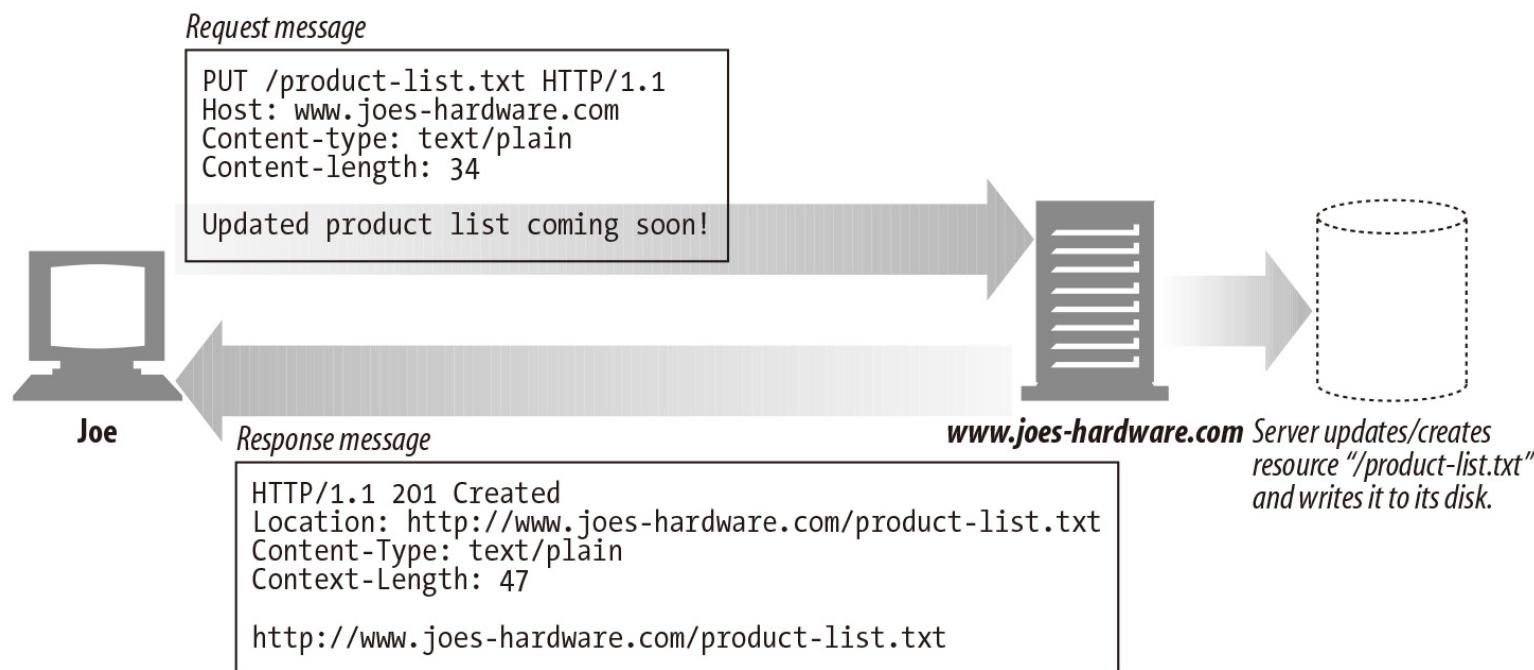


Methods

“Case study of each methods using example”

□ PUT

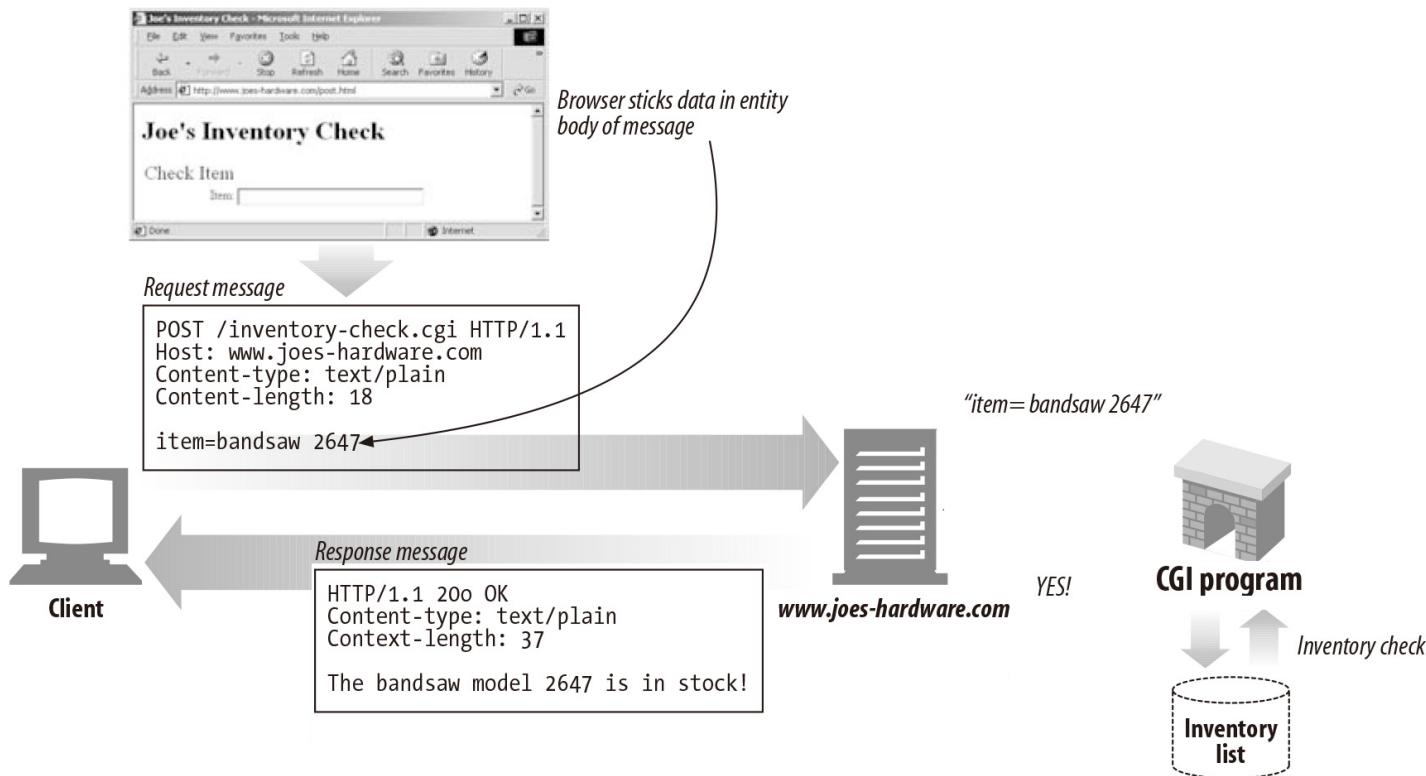
- Writes documents to a server
- Many server require password before perform a PUT



Methods

“Case study of each methods using example”

- POST
 - Send input data to the server

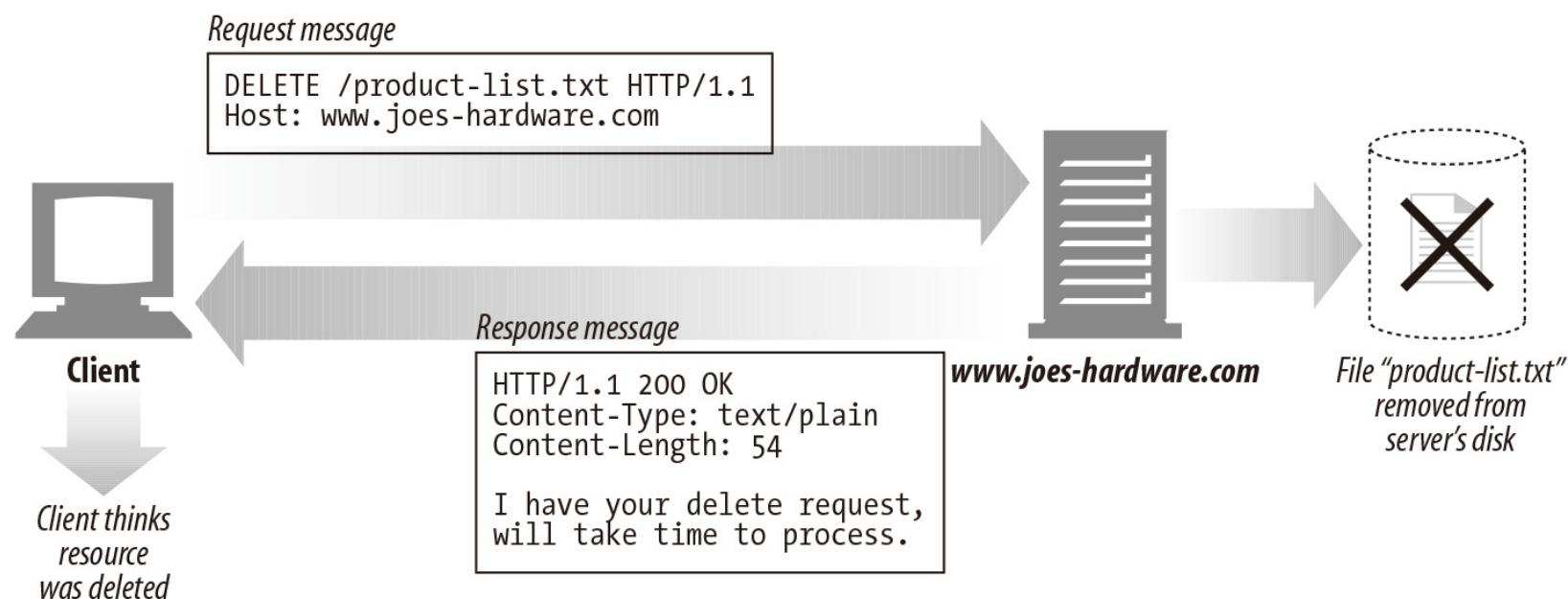


Methods

“Case study of each methods using example”

□ DELETE

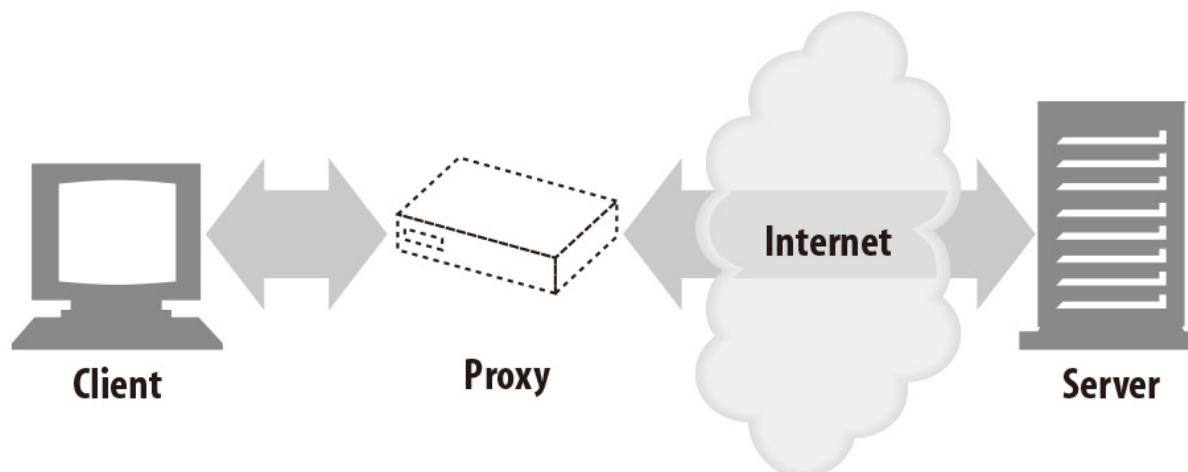
- Ask the server to delete the resource specified by request URL



Architectural Components of the Web

"HTTP Networking"

- Proxy
 - HTTP intermediaries
 - Sit between clients and servers
 - Web Security
 - Application Integration
 - Performance Optimization



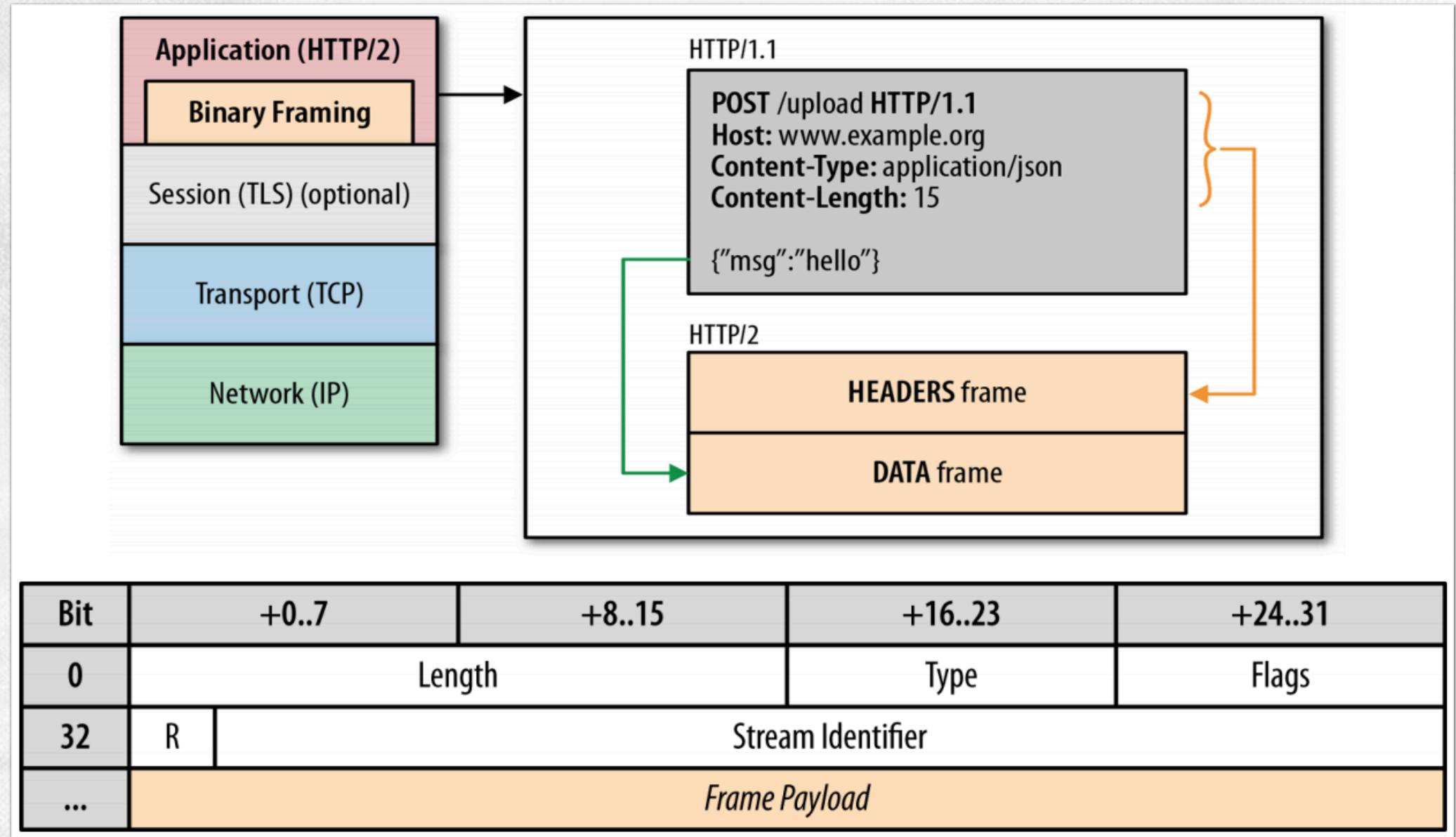
Contents

- HTTP/1.1
- **HTTP/2**
- HTTP/3
- SIP
- Advanced Web Technologies

바이너리 프로토콜	<ul style="list-style-type: none"> - 텍스트가 아닌 Binary 프레임으로 구성 → 파싱이 더 빠르고, 오류 발생 가능성이 낮음
Multiplexing	<ul style="list-style-type: none"> - 하나의 TCP Connection내에서 다수의 Stream을 생성 - 하나의 요청이 지연되면 나머지 응답이 늦어지는 HTTP Pipelining과는 달리 각각의 요청/응답을 독립적으로 처리 → 다수의 요청/응답을 동시에 처리 가능
헤더 압축	<ul style="list-style-type: none"> - 반복적으로 사용되는 헤더를 헤더 테이블내의 인덱스로 표기 → 헤더 크기를 80% 정도 줄임
우선순위 설정	<ul style="list-style-type: none"> - Stream별로 우선 순위를 지정 → 중요한 리소스의 처리 지연을 방지
Server Push	<ul style="list-style-type: none"> - 클라이언트가 요청하지 않아도 필요가 예상되는 리소스를 서버가 미리 전송

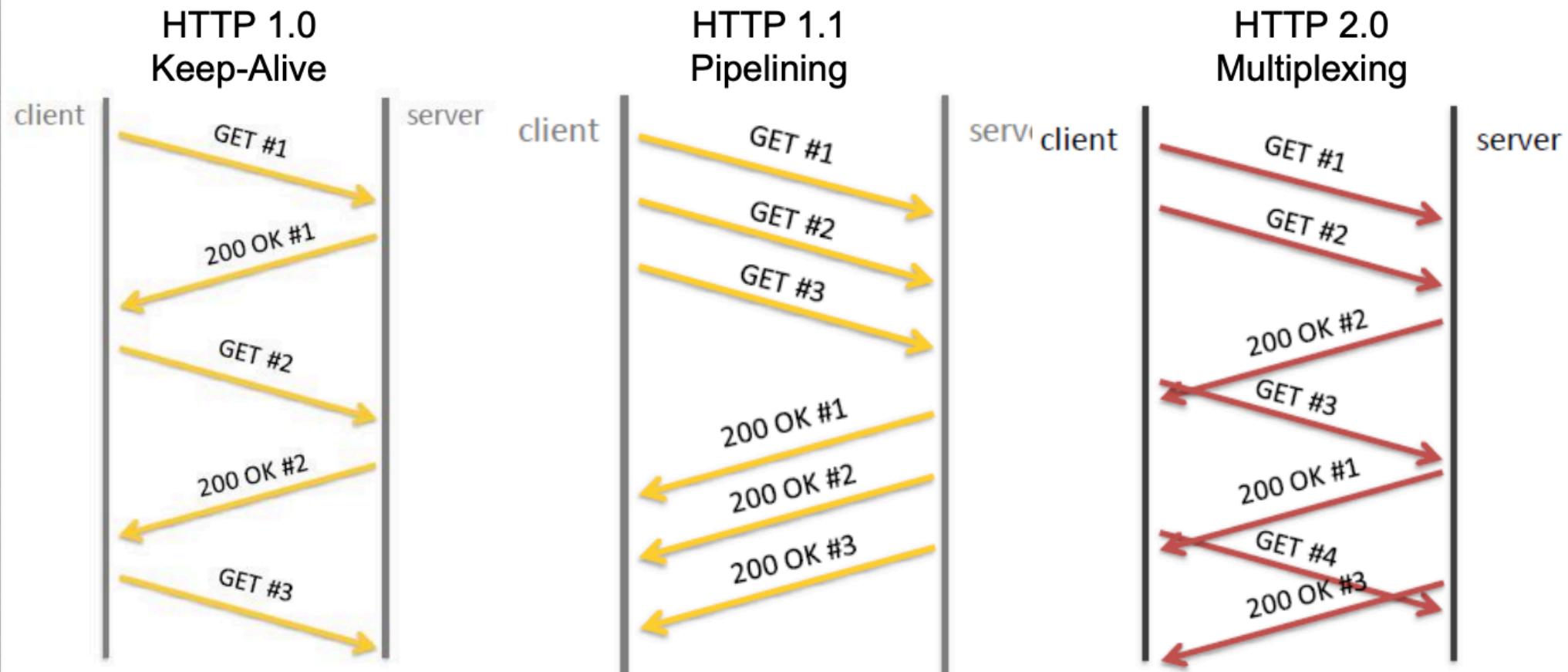
HTTP 2.0

Binary protocol

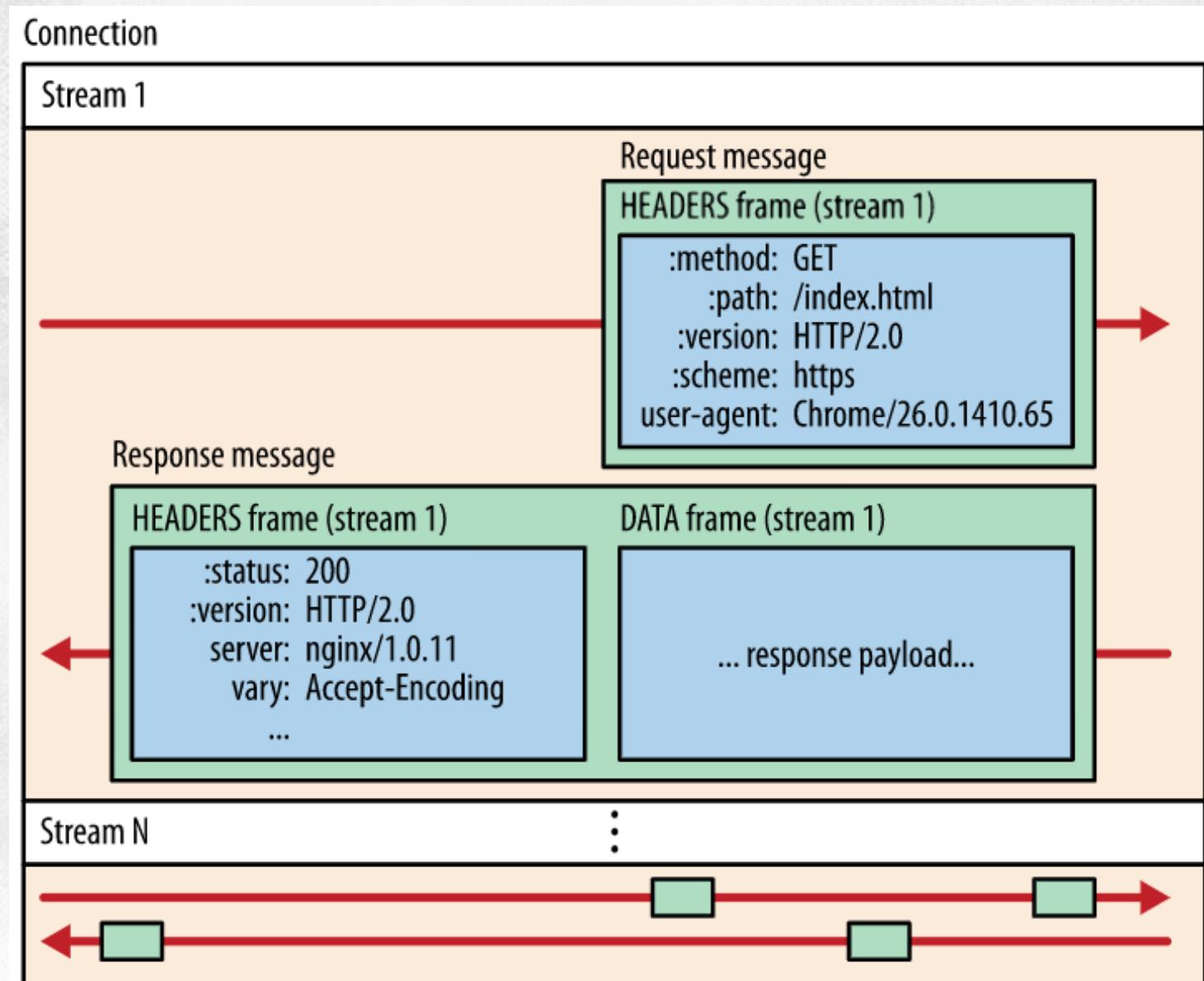


Multiplexing

- 프레임마다 관련된 요청 순서 번호(Stream #)를 기재
- 요청 순서에 상관없이 데이터 전송 가능
- 하나의 TCP 세션 사용



HTTP 2.0 Multiplexing

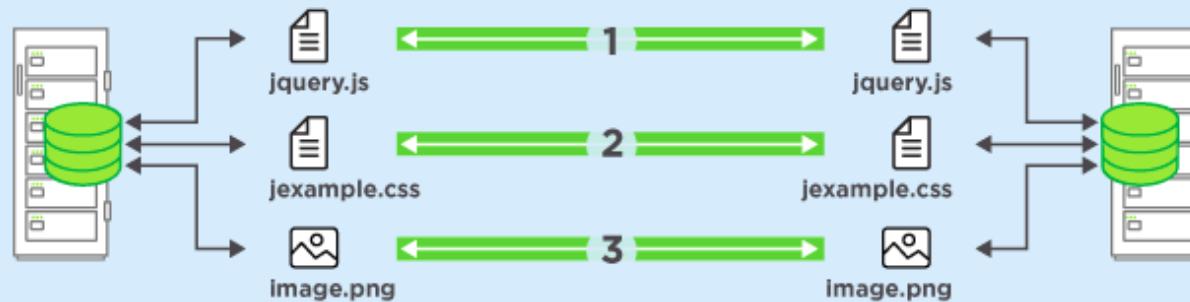


HTTP 2.0

Multiplexing

HTTP 1.1

3 TCP CONNECTIONS



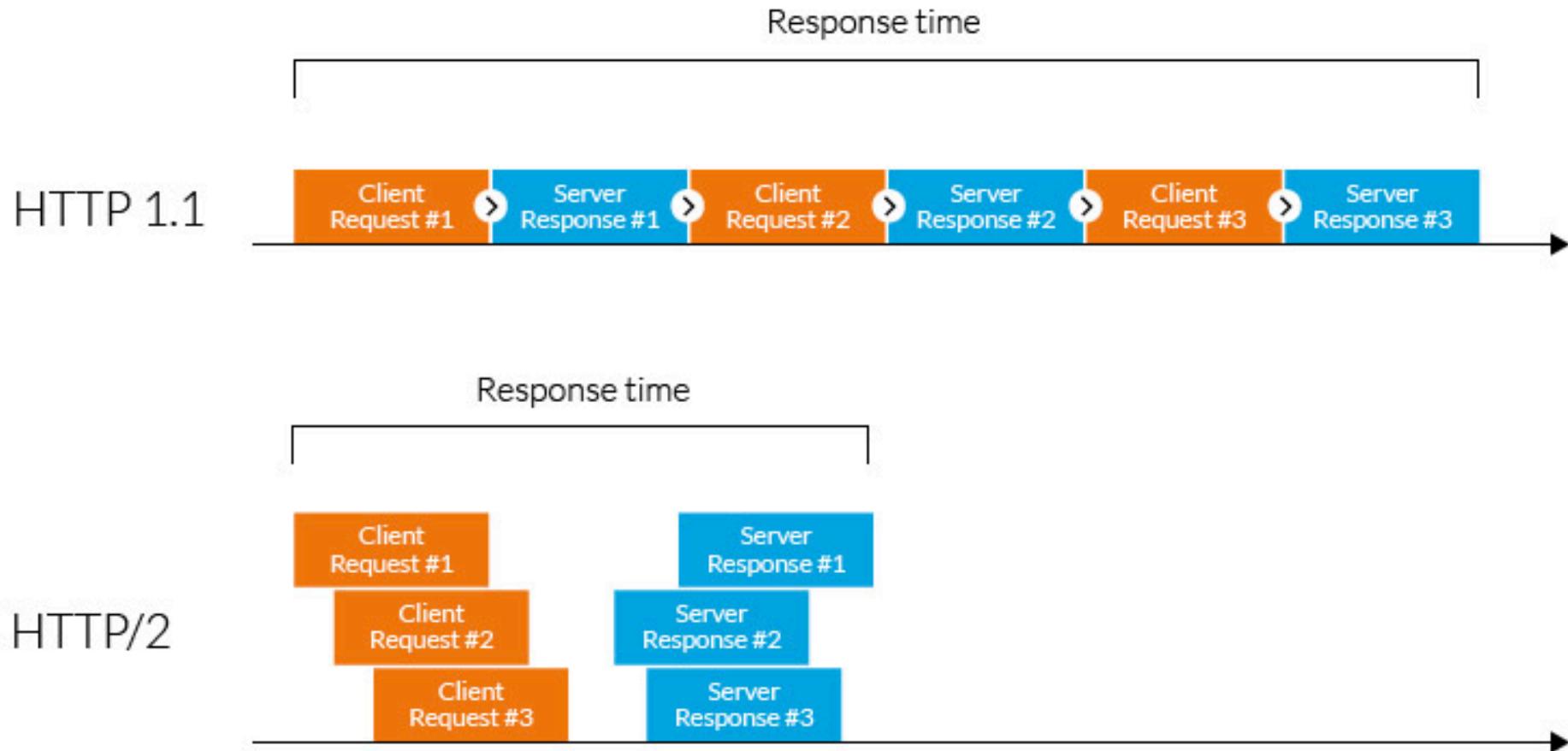
HTTP/2

1 TCP CONNECTION



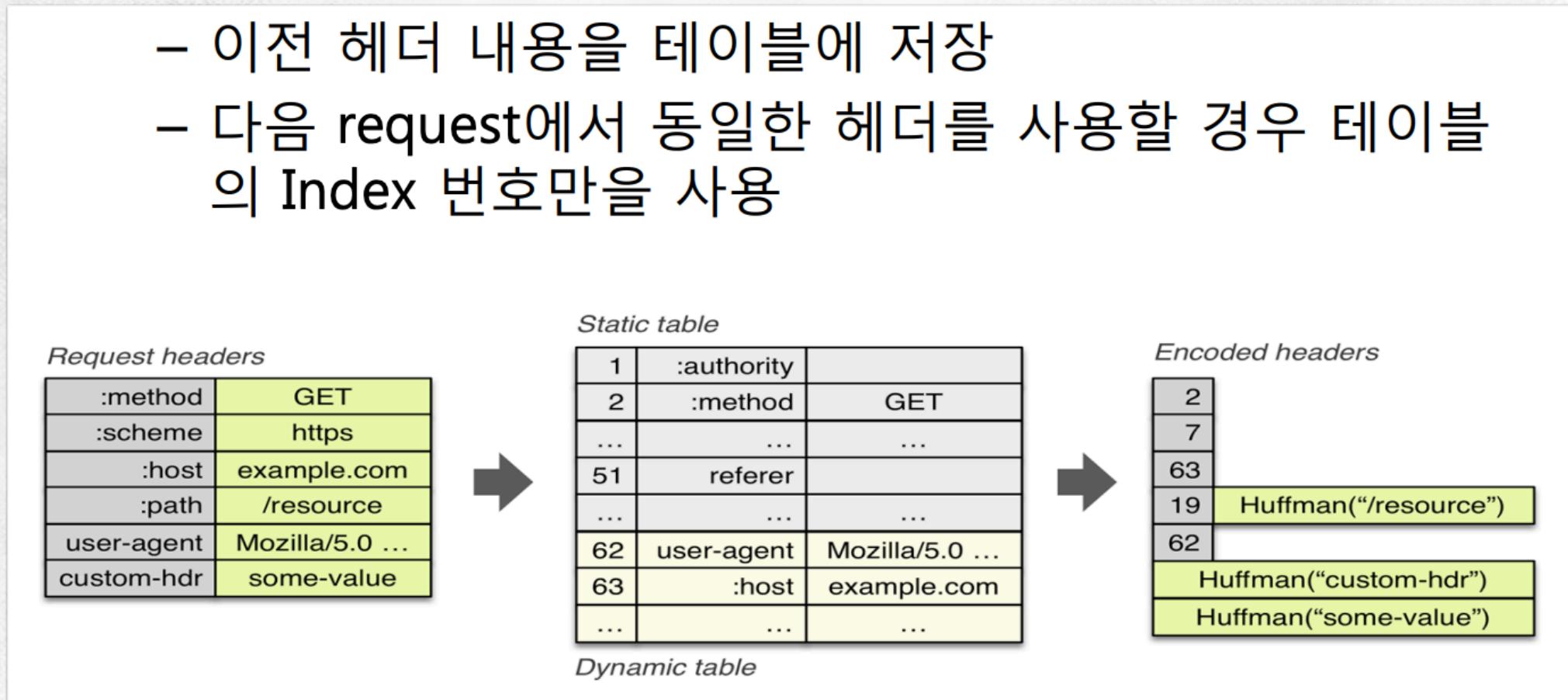
HTTP 2.0

Multiplexing



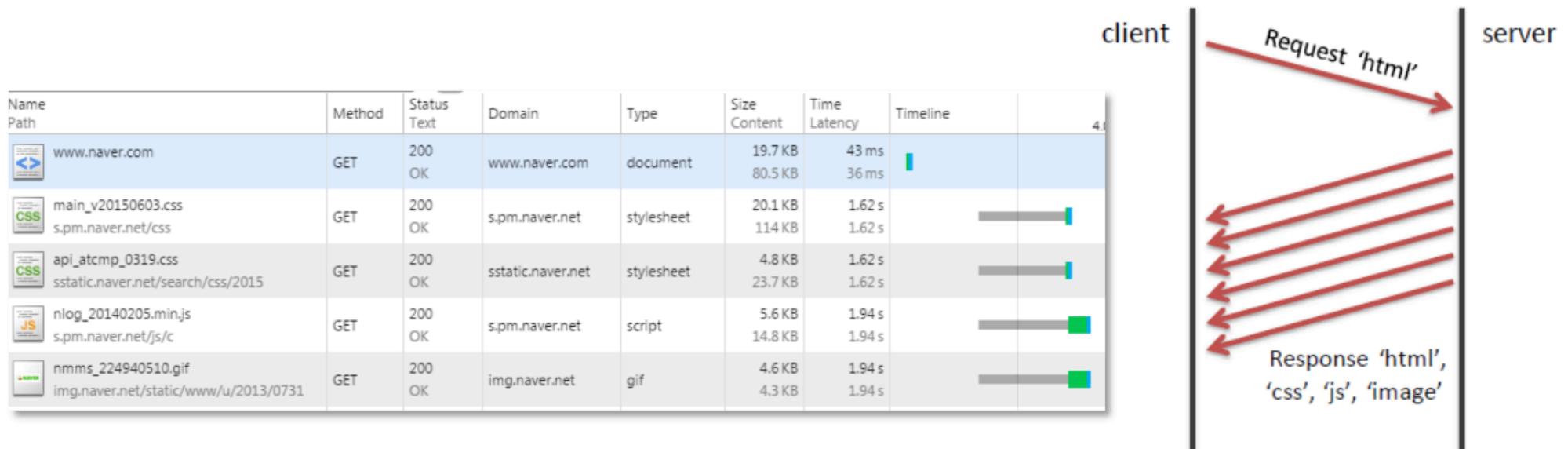
Header Compression

- 이전 헤더 내용을 테이블에 저장
- 다음 request에서 동일한 헤더를 사용할 경우 테이블의 Index 번호만을 사용



Server Push

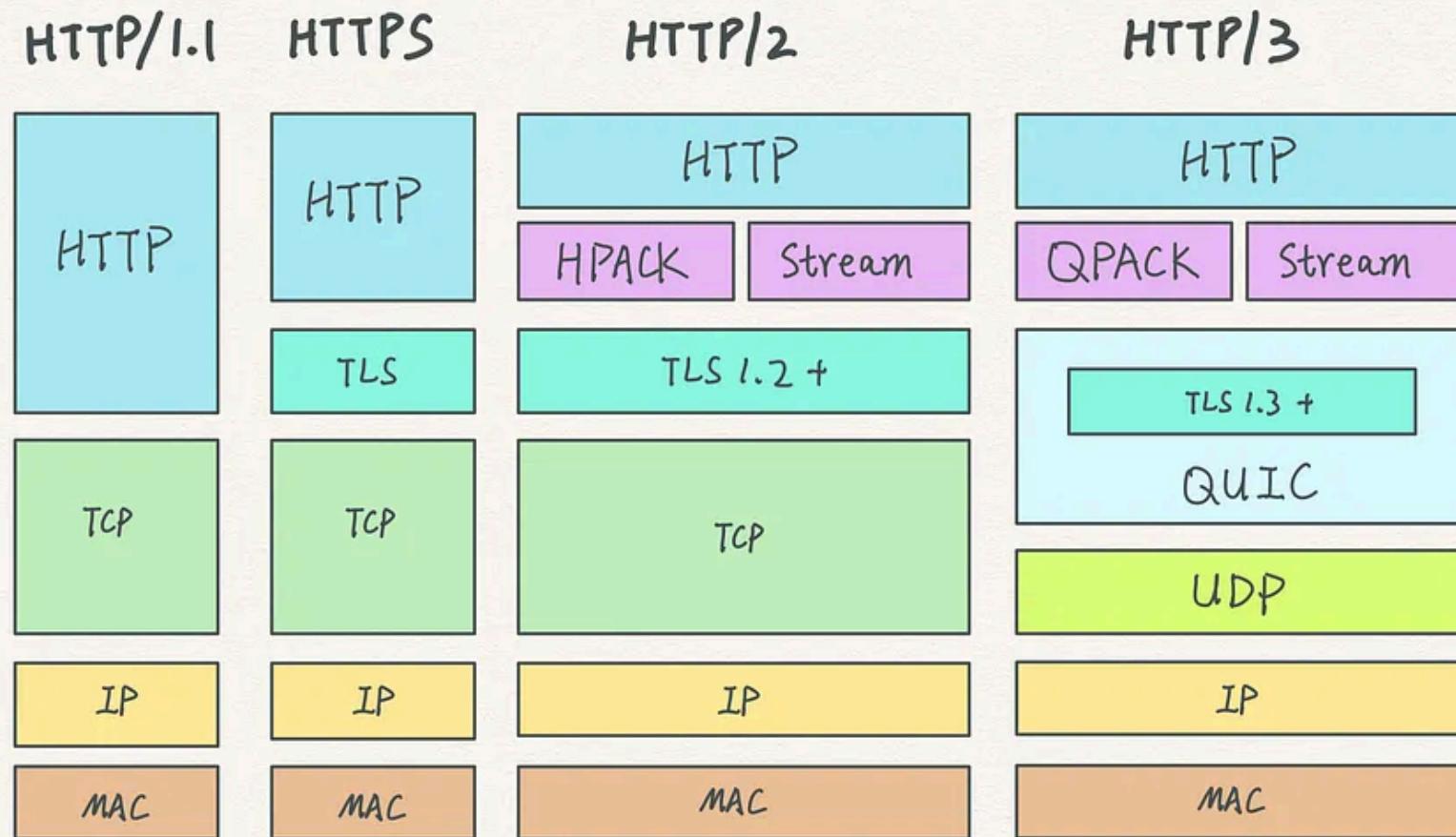
- Client가 요청하기 전에 필요가 예상되는 리소스를 Server가 먼저 전송
- Server 쪽에서의 추가적인 프로그래밍이 필요



Contents

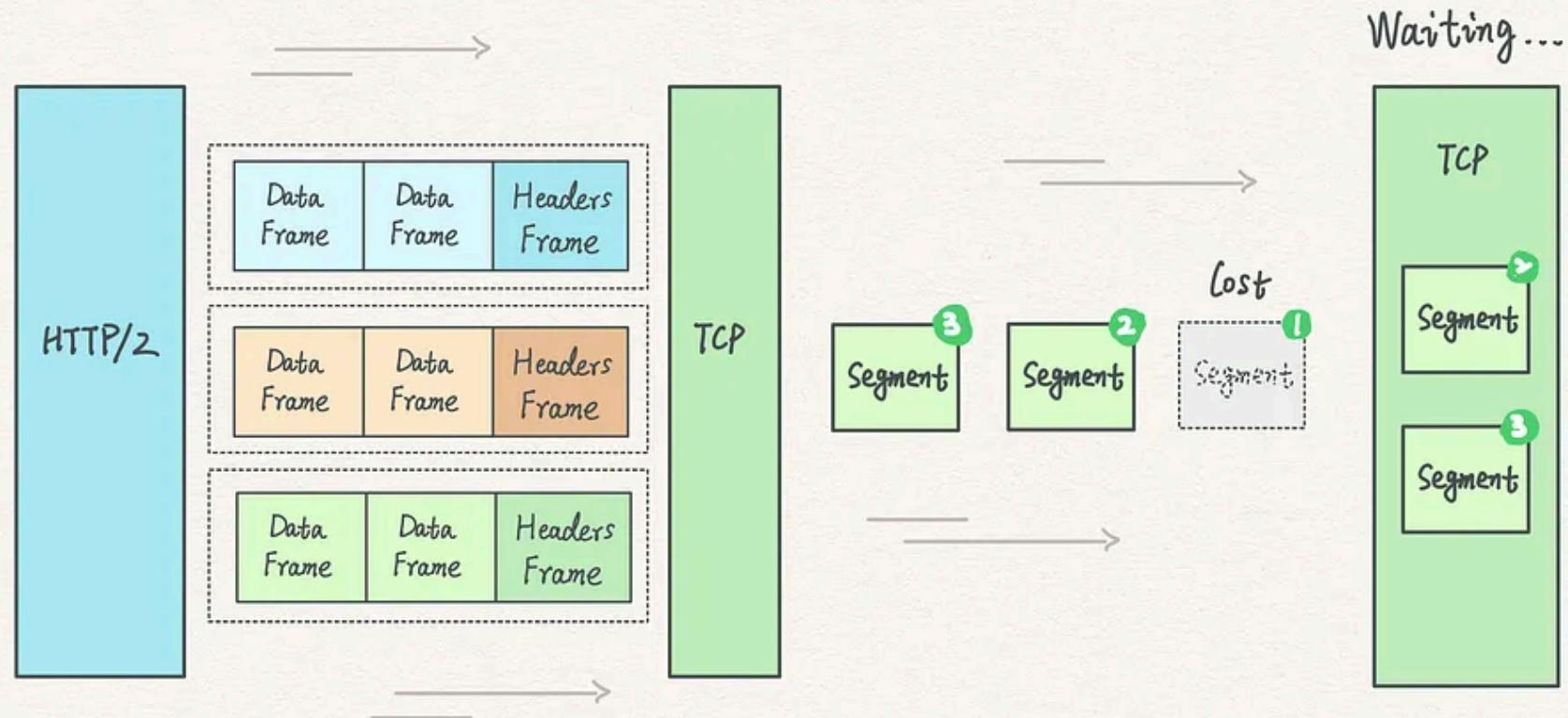
- HTTP/1.1
- HTTP/2
- **HTTP/3**
- SIP
- Advanced Web Technologies

UDP based NEW Secure Transport Protocol



Why UDP?

Head-of-Line Blocking on HTTP/2



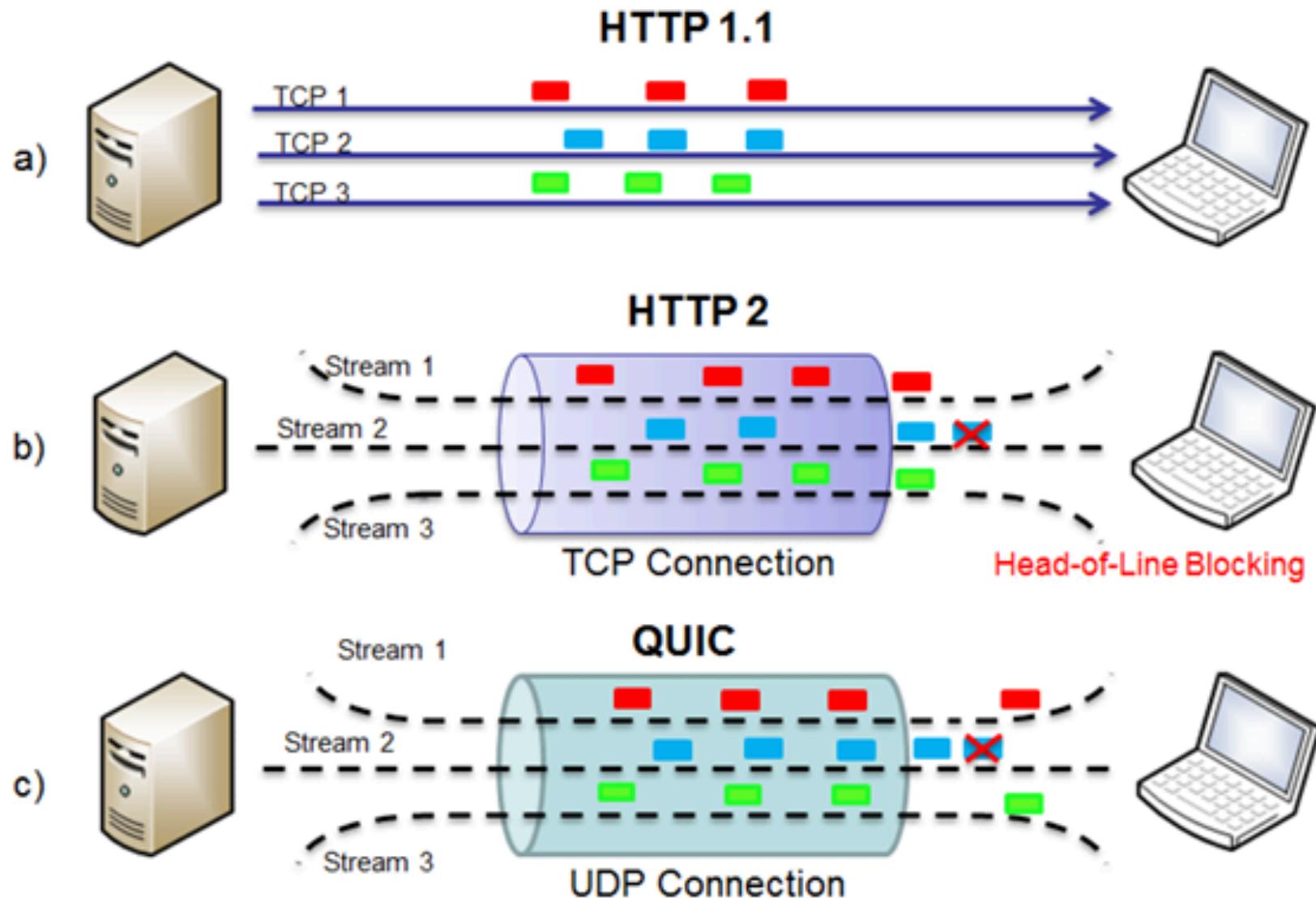
● QUIC 연혁

- ◆ QUIC은 TCP를 대체하는 범용 목적의 전송 계층 통신 프로토콜로서, 구글의 짐 로스킨드가 처음 설계하였고, 2012년 구현 및 적용되었으며, 2013년 실험 확대로서 공개 발표됨
- ◆ 국제 인터넷 표준화 기구에 기술되었고, 2021년 5월 IETF RFC9000(+RFC9001/9002/8989)으로 정식 표준화 됨
- ◆ 이미 구글 크롬에서부터 구글 서버에 이르는 모든 연결의 절반 이상에 사용되고 있으며, 크롬/MS 엣지/모질라 파이어폭스/사파리에서 지원함

● QUIC 이름

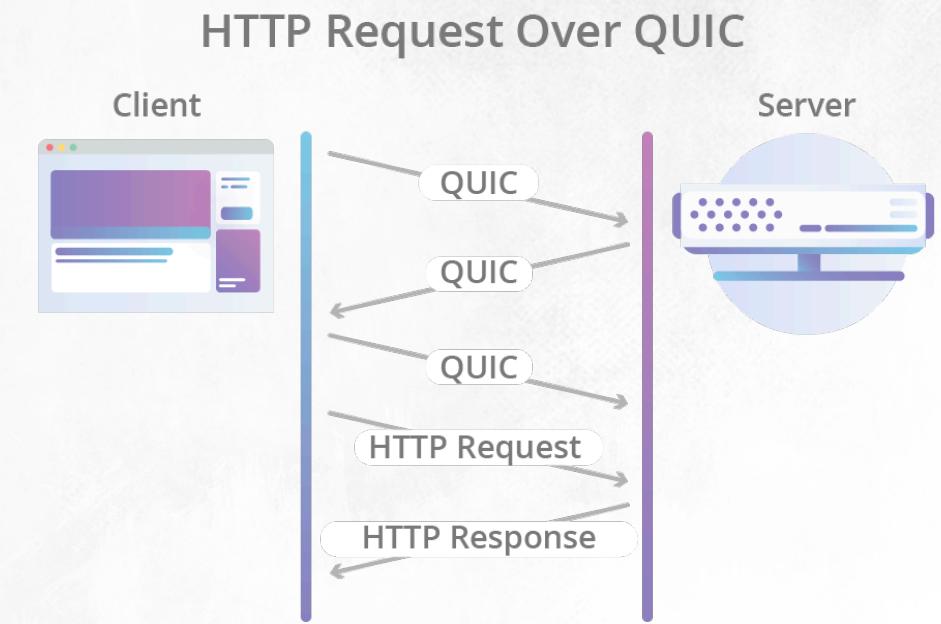
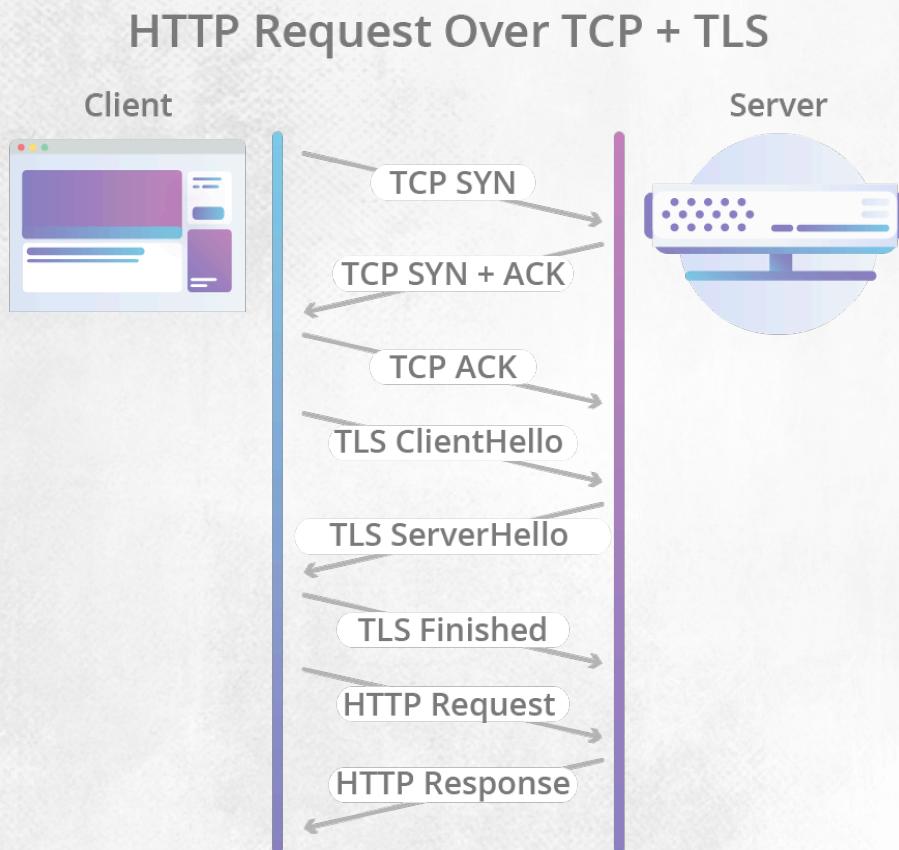
- ◆ 당초 구글은 "Quick UDP Internet Connections"로 제안했으나, IETF에서 'QUIC'을 고유 명사로 지정함

Stream Multiplexing



Reduced {Signaling} Latency

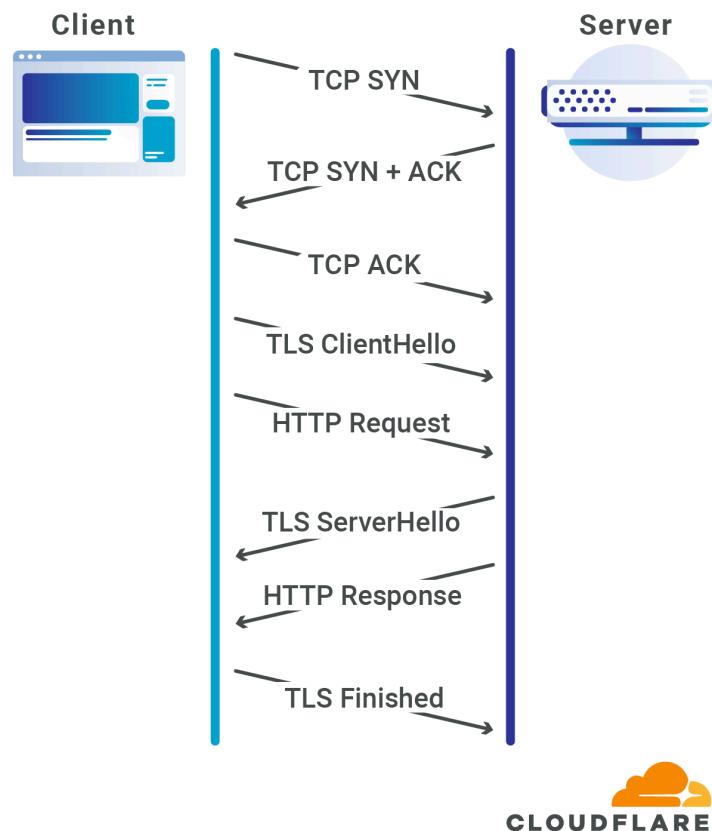
- 일반적인 TCP/TLS vs QUIC



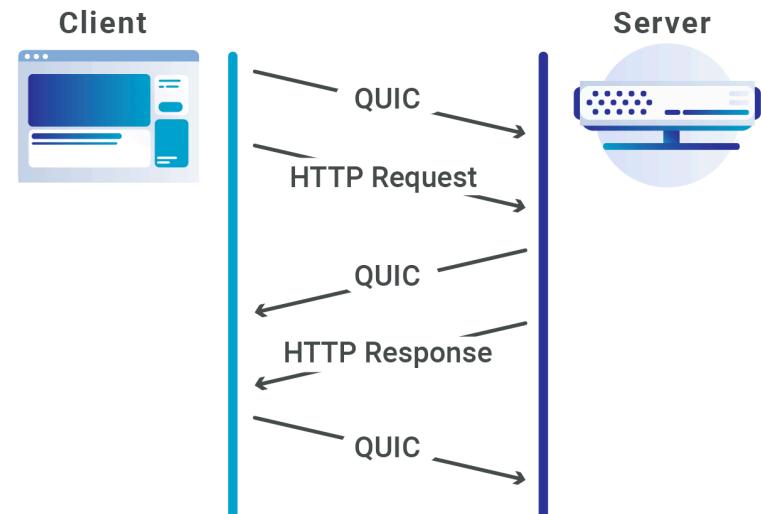
Reduced {Signaling} Latency

● 가속화된 TCP/TLS vs QUIC

HTTP Request over TCP+TLS (with 0-RTT)



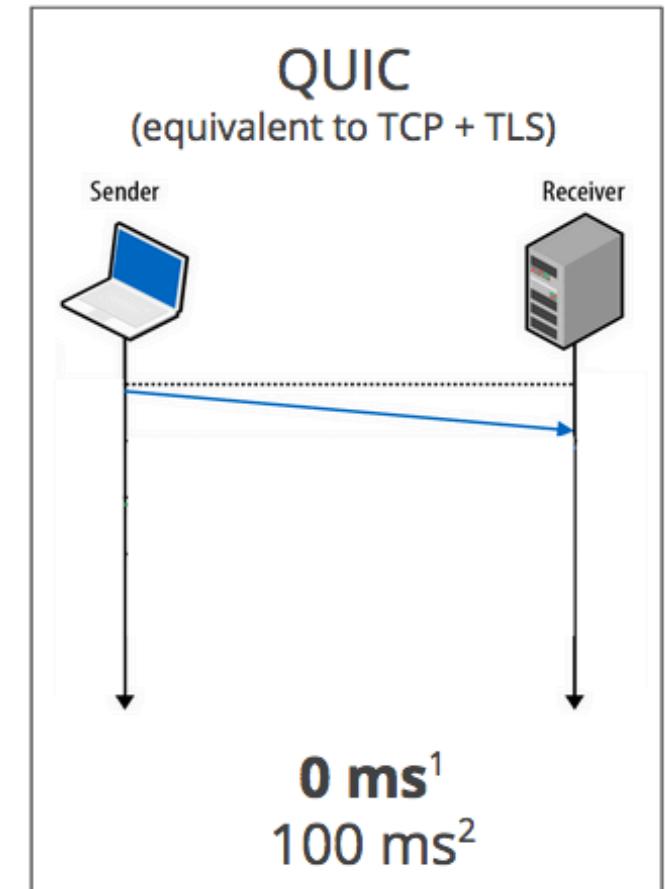
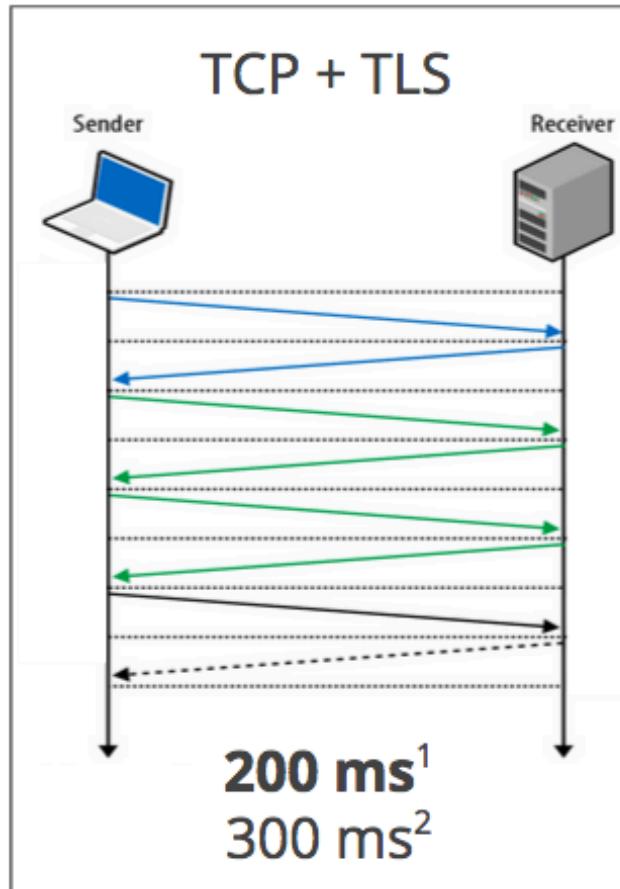
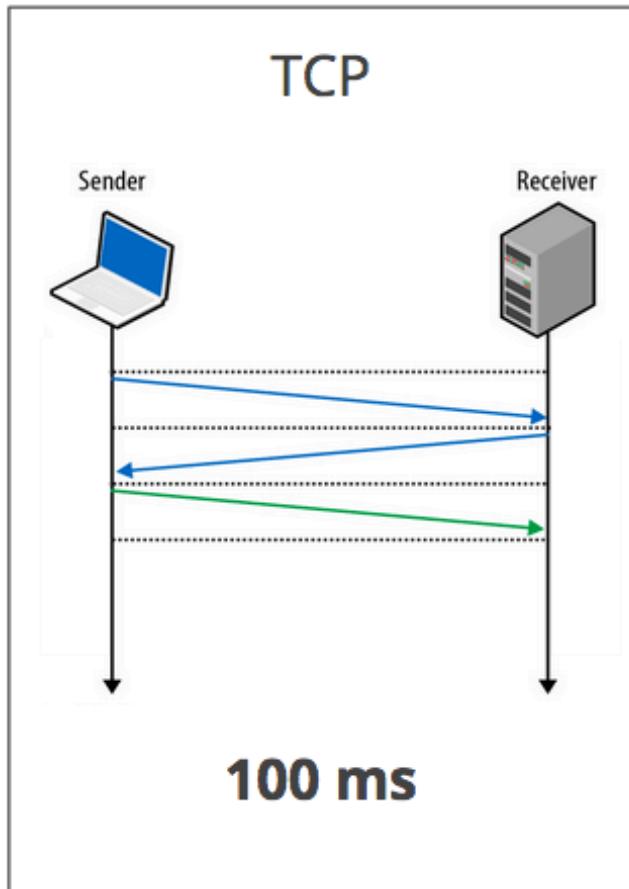
HTTP Request over QUIC (with 0-RTT)



Reduced {Signaling} Latency

- Result

Zero RTT Connection Establishment



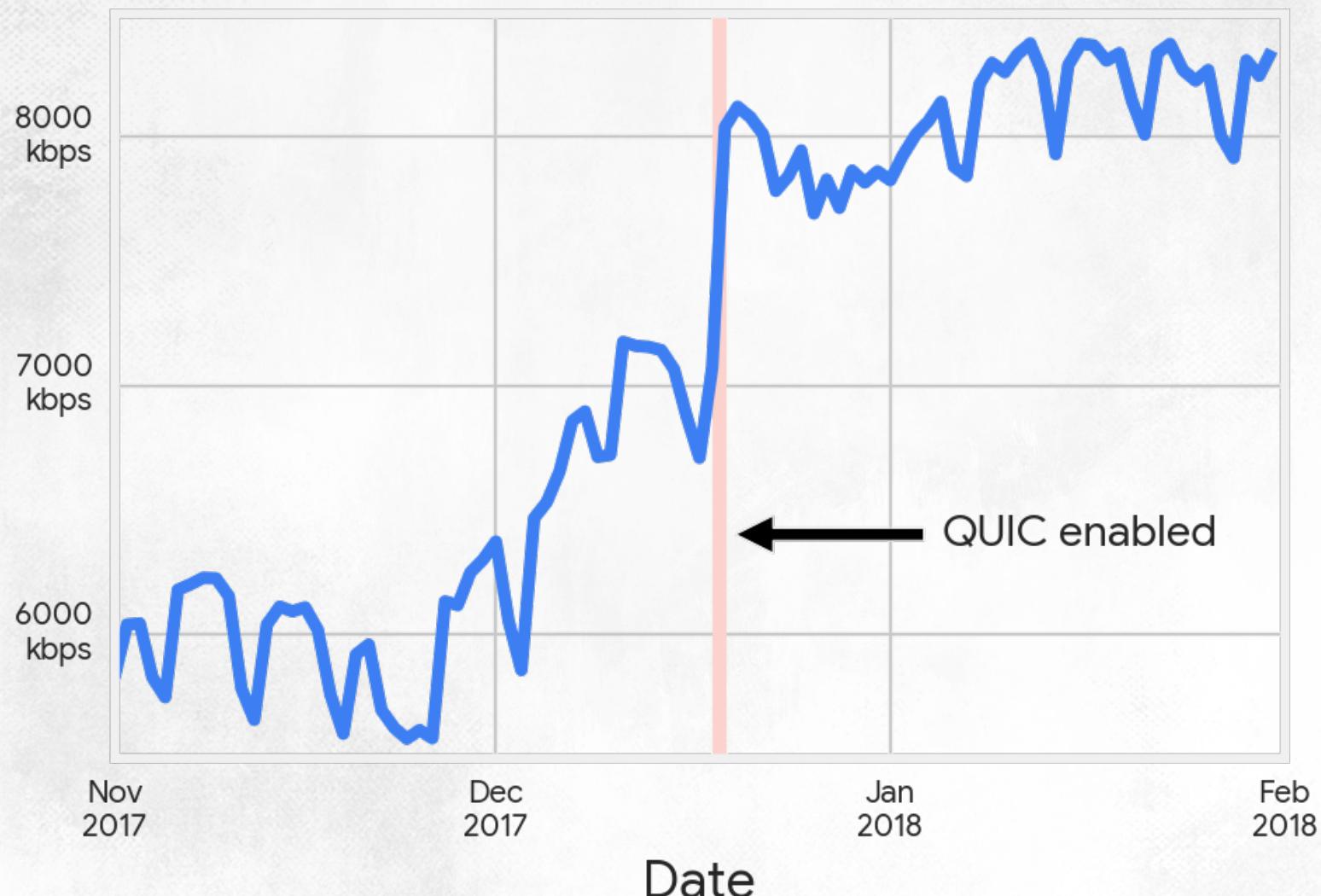
1. Repeat connection
2. Never talked to server before

HTTP/3 is ‘HTTP over QUIC’

- HTTP is the First & Primary service of QUIC
- HTTP/2 vs HTTP/3 (HTTP over QUIC)
 - In QUIC the streams are provided by the transport itself, while in HTTP/2 the streams were done within the HTTP layer
 - QUIC streams are slightly different than HTTP/2 streams (thus, header compression protocol is also slightly different)

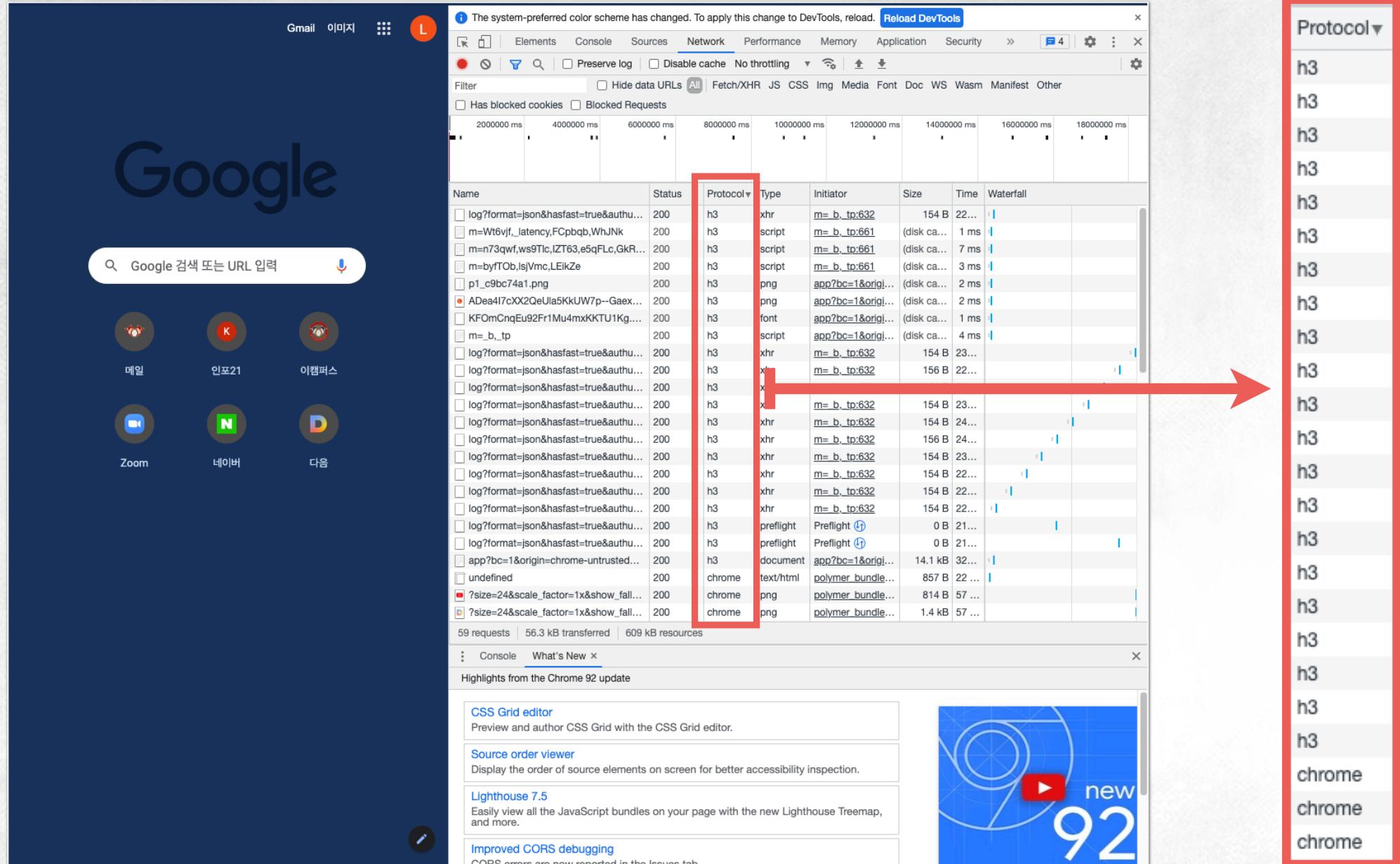
Finally, Enhanced Performance

Cloud CDN throughput (50th percentile)



QUIC & HTTP/3 현황

Google



QUIC & HTTP/3 현황

Google's YouTube

ZDNET Korea 뉴스 이슈진단+ 지스타2022 반도체 교육 컨퍼런스 칼럼 · 연재 포토 · 영상

NG

최고의 게이밍 성능을 Arm에서
Arm의 새로운 GPU는
최고의 성능과 효율성을 제공합니다.

더 알아보기

네이버, HTTP/3 도입..."음영지역서도 안정적 검색 가능"

● 이용자에게 빠르고 안정적인 검색 환경 제공 목표

인터넷 | 입력 : 2022/11/15 10:55

김성현 기자 | 기자 페이지 구독 기자의 다른기사 보기

[웨비나] 11월 22일 Slack 추가기능 데모 및 무신사 피플 슬

네이버 검색 HTTP/3 도입

HTTP/3

HTTP/2

Contents

- HTTP/1.1
- HTTP/2
- HTTP/3
- SIP
- Advanced Web Technologies

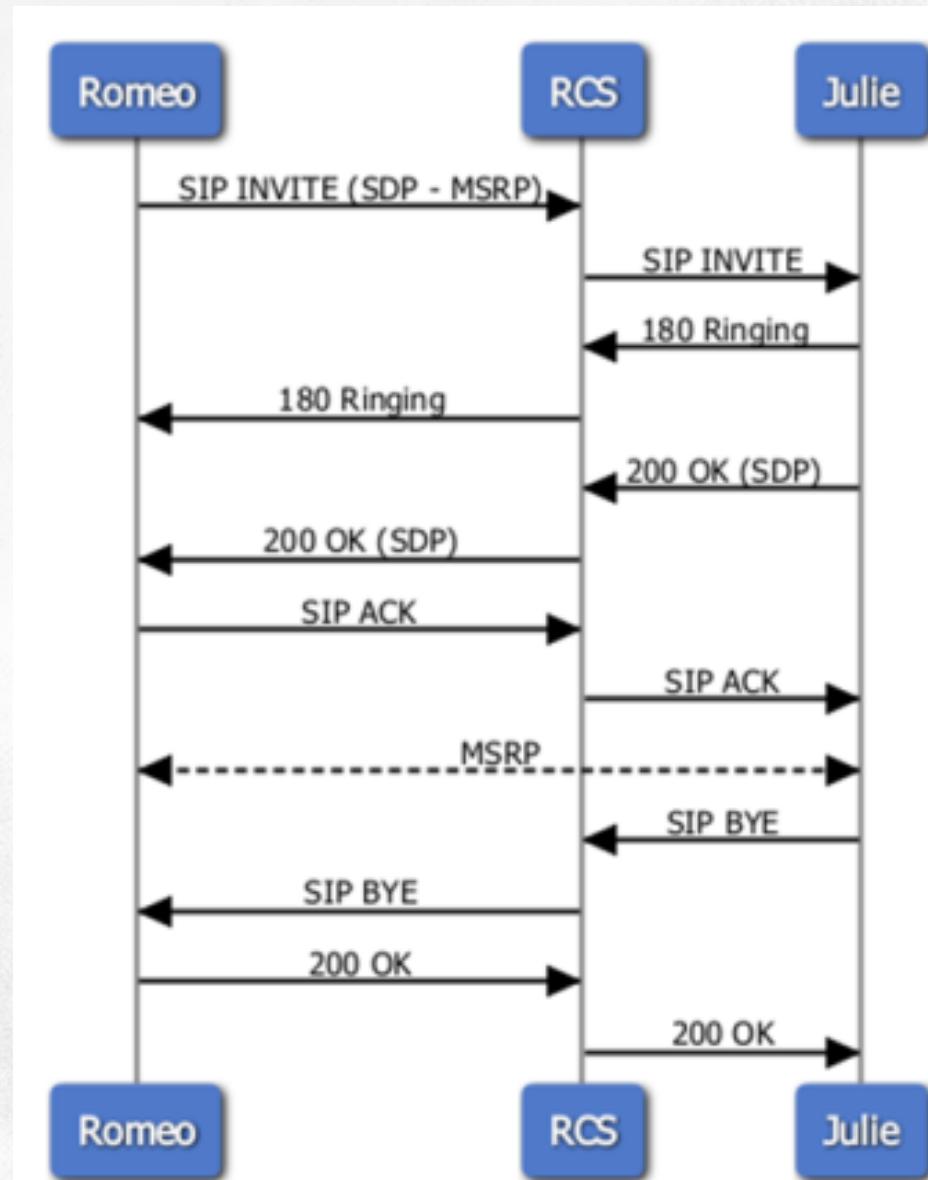
Session Initiation Protocol

Concept

- SIP is an IETF signaling protocol designed for creating, modifying, and terminating media sessions with one or more participants
- SIP is independent of the media. Media is transported via the Real-Time Protocol (RTP)
- SIP is heavily inspired by HTTP in that it is a text-based, request-response protocol

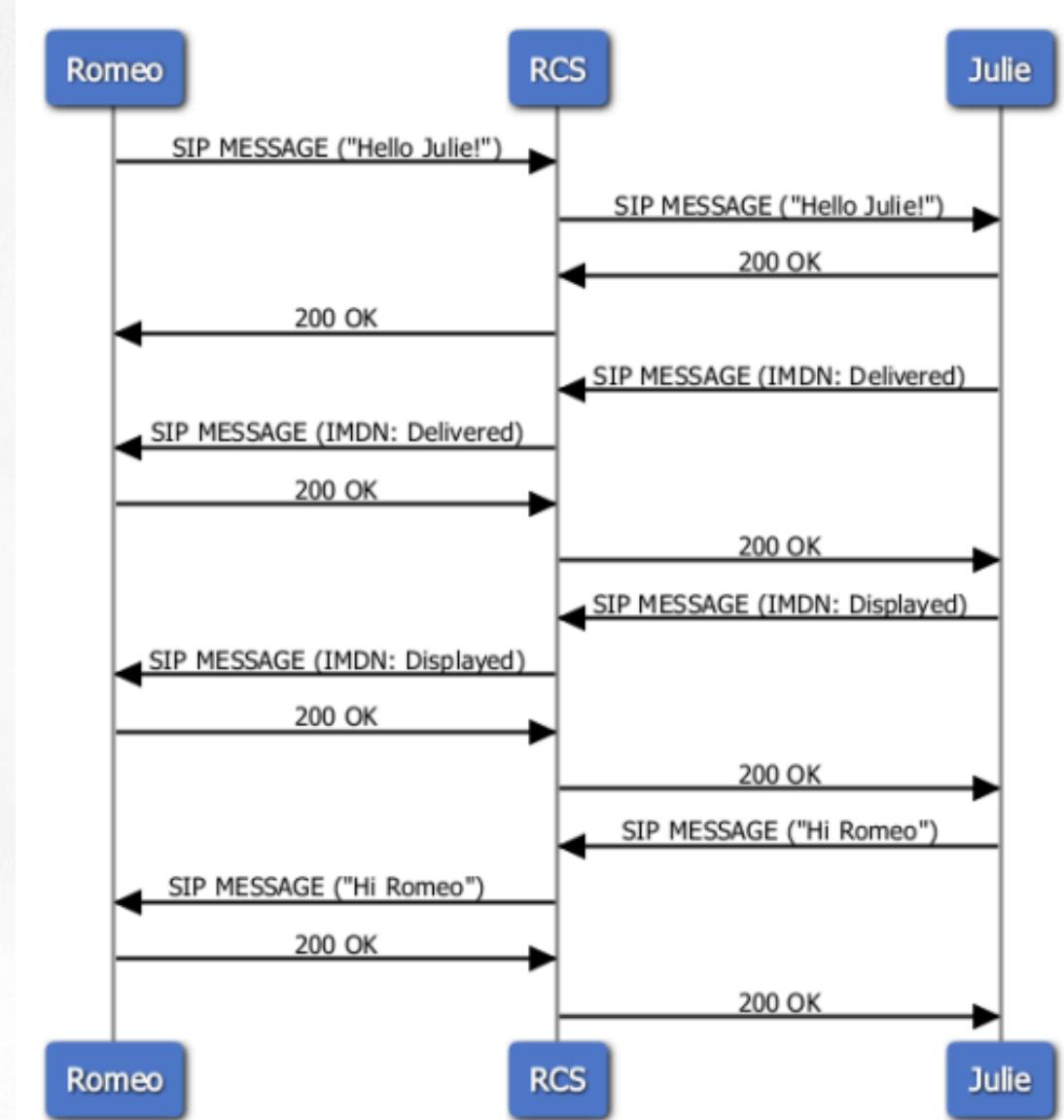
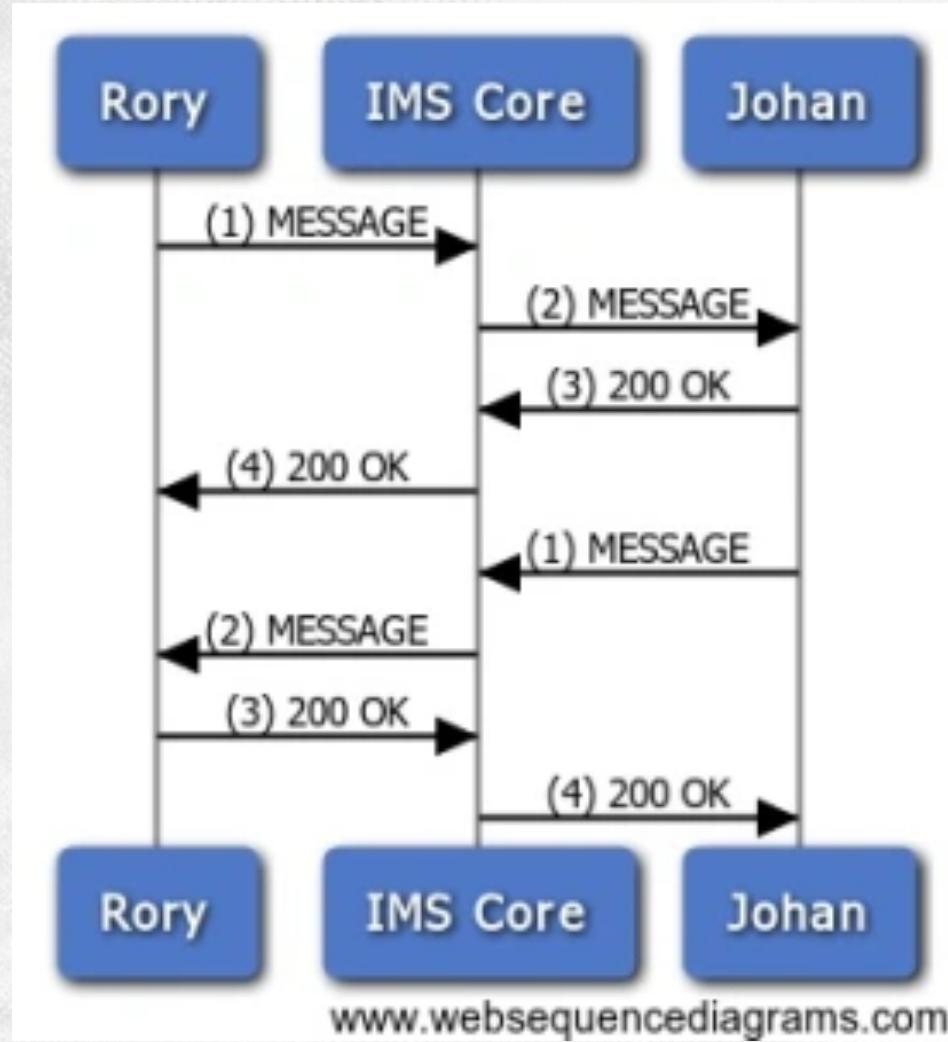
Session Initiation Protocol

Connection Establishment & Release



Session Initiation Protocol

Instant Messaging



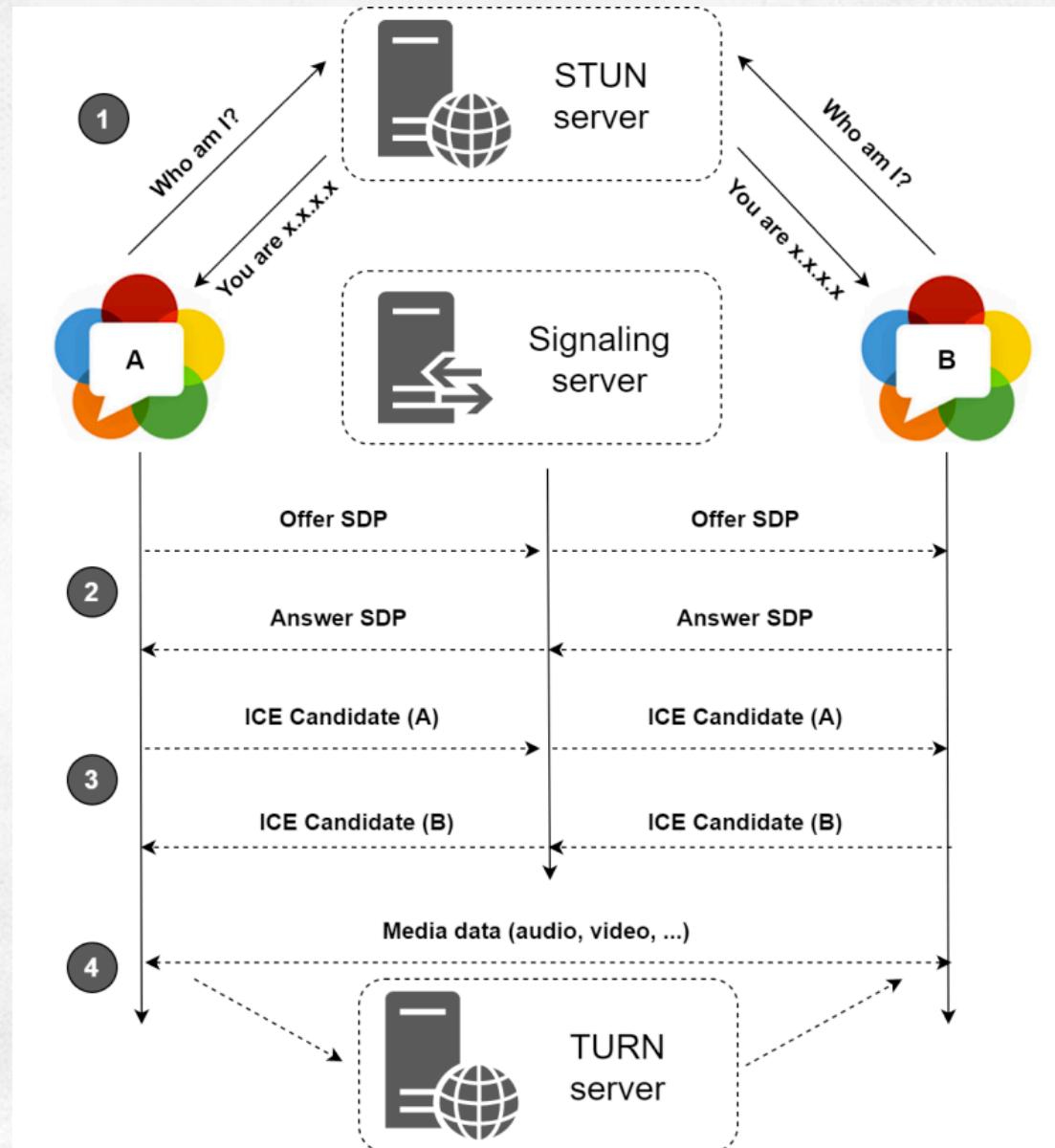
Contents

- HTTP 1.x
- HTTP 2.0
- HTTP 3.0
- SIP
- Advanced Web Technologies

WebRTC Concept

- WebRTC (Web Real-Time Communication) is a free, open-source project that provides web browsers and mobile applications with real-time communication (RTC) via simple application programming interfaces (APIs).
- It allows audio and video communication to work inside web pages by allowing direct P2P (peer-to-peer, server-less) communication, eliminating the need to install plugins or download native apps.
- Supported by Google, Microsoft, Mozilla, and Opera, WebRTC is being standardized through the World Wide Web Consortium (W3C) and the Internet Engineering Task Force (IETF).





Mozilla foundation's Mixed Reality

[HARDWARE](#)[GAMES](#)[TOP PICKS](#)[VR DESIGN](#)[GUEST ARTICLES](#)[NEWSLETTER](#)[MORE](#)

Mozilla Releases Firefox-based ‘WebXR Viewer’ App for Browser-based AR on iOS

By Ben Lang - Jun 8, 2020 3

Image courtesy Mozilla

Mozilla foundation's Mixed Reality



C/C++ over Web Browser

- Designed by Mozilla foundation
- First appeared at 21 March 2013
- Supported by Firefox, Chrome and *Edge (future roadmap)*

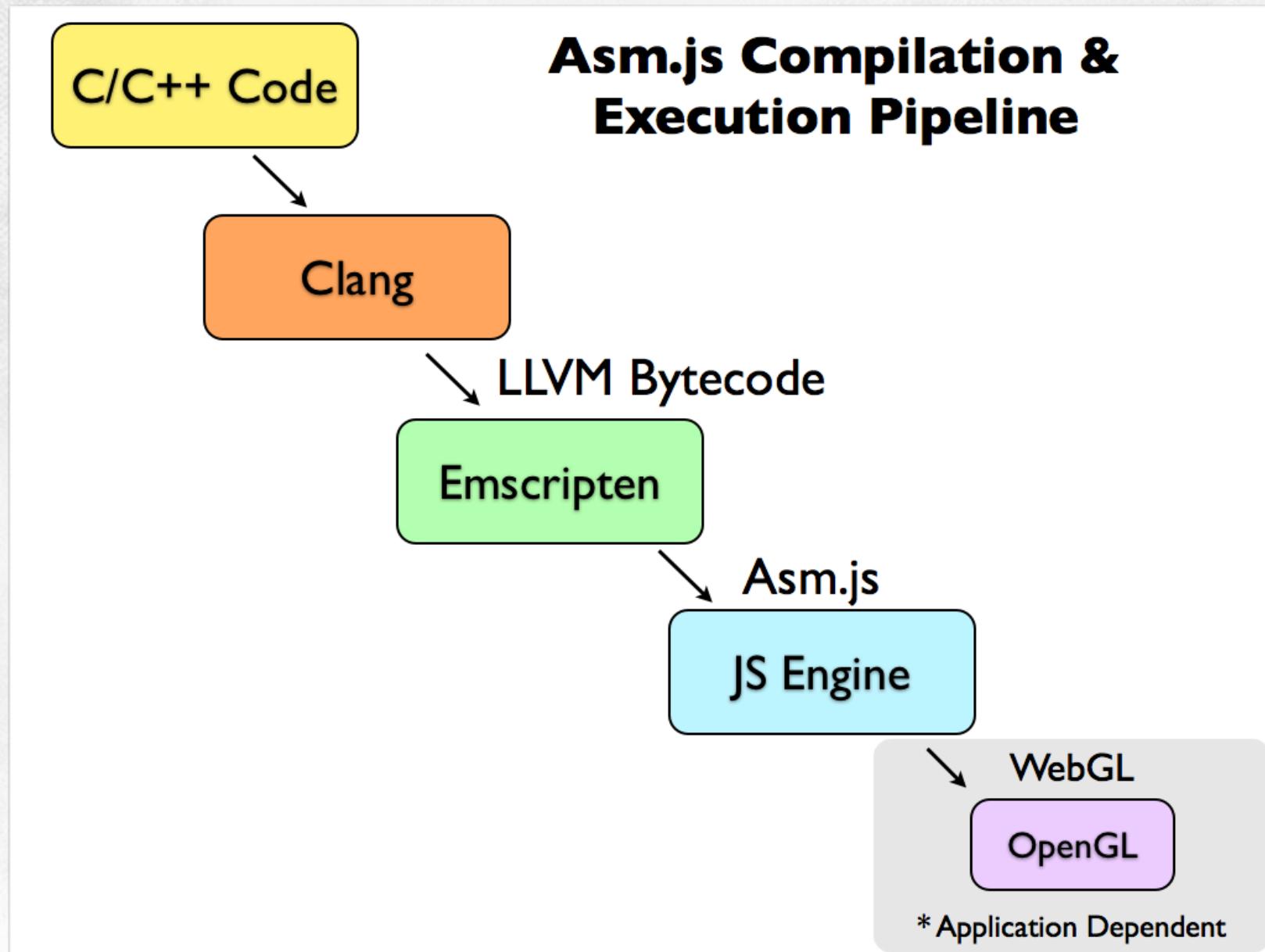
asm.js

an extraordinarily optimizable, low-level subset of JavaScript

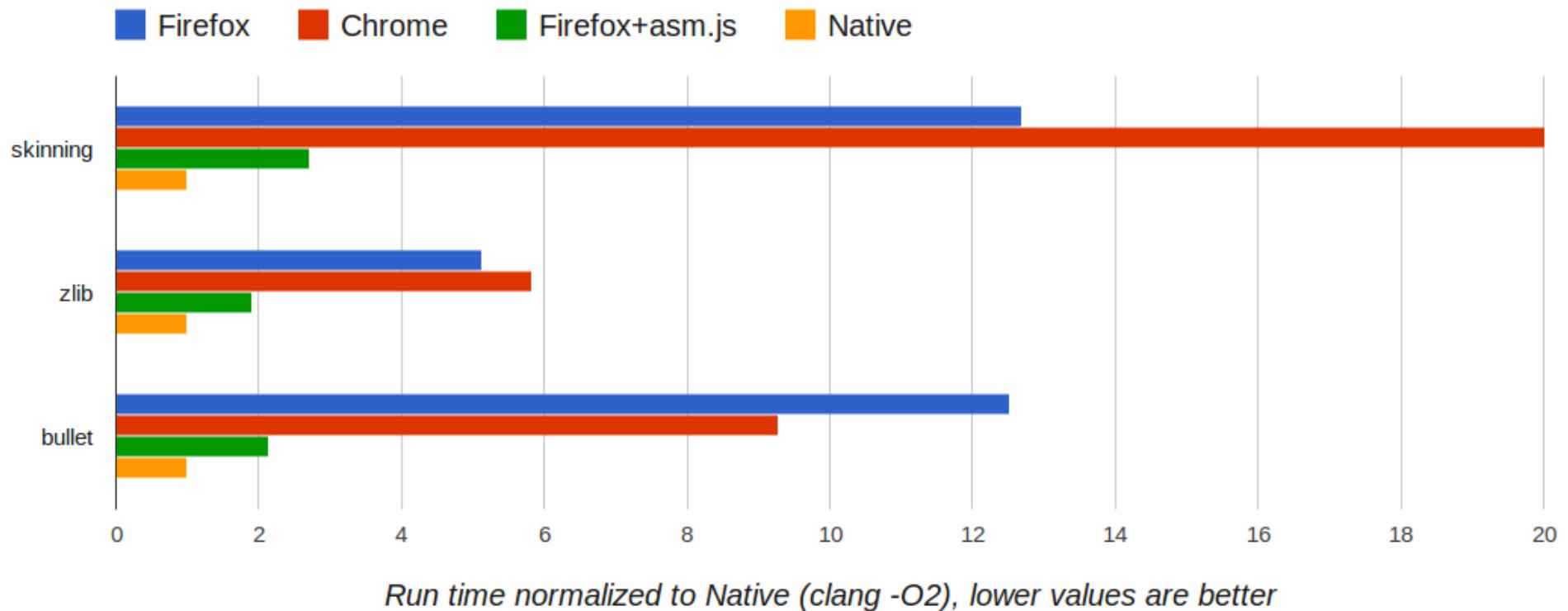
[specification](#) (work in progress)

[frequently asked questions](#)

[big web app? compile it!](#) – slide deck by [@kripken](#)



Performance



Example

For example, given the following C code:

```
int f(int i) {
    return i + 1;
}
```

Emscripten would output the following JS code:

```
function f(i) {
    i = i|0;
    return (i + 1)|0;
}
```

Here is another example to calculate the length of a string:

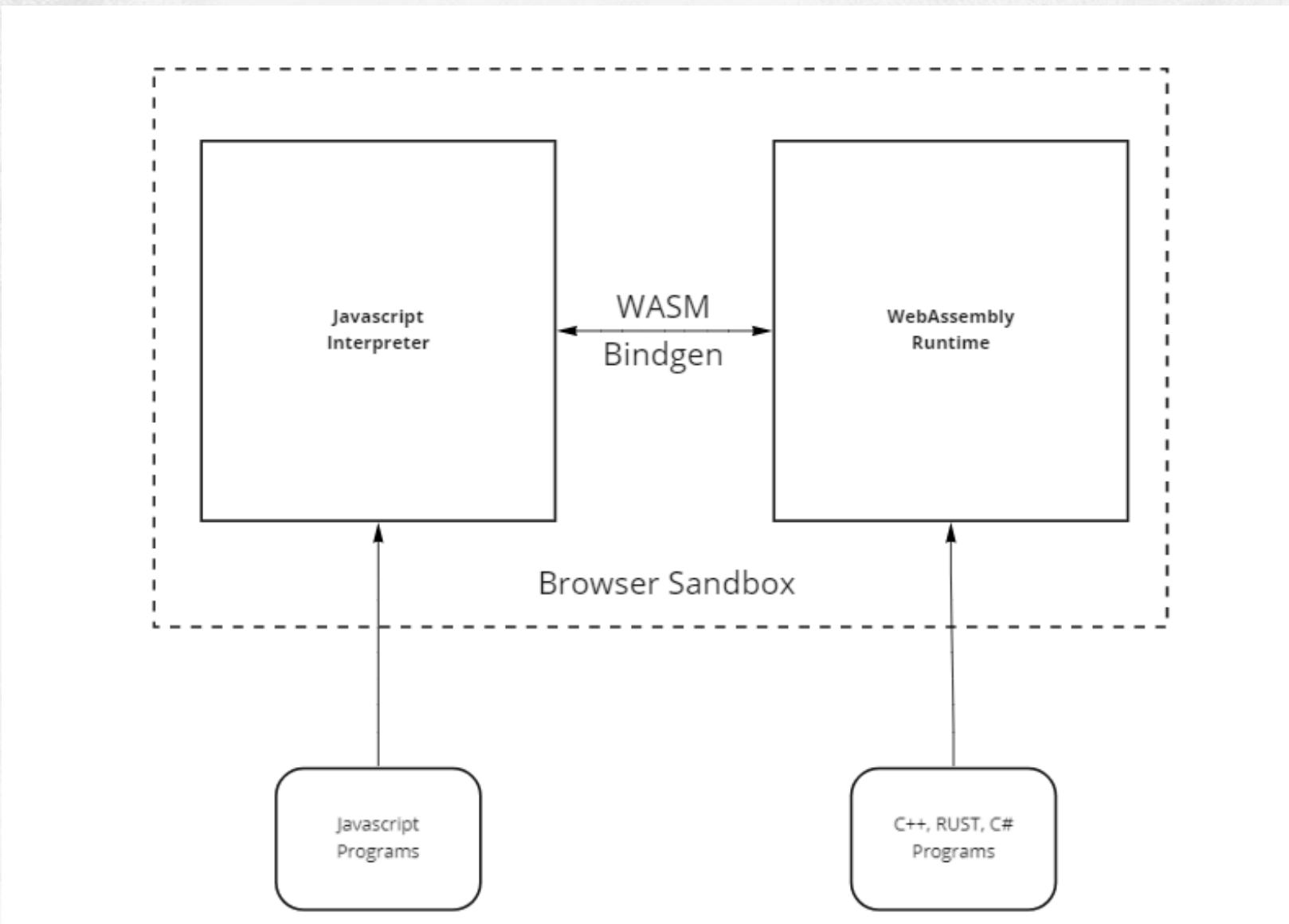
```
size_t strlen(char *ptr) {
    char *curr = ptr;
    while (*curr != 0) {
        curr++;
    }
    return (curr - ptr);
}
```

This would result in the following asm.js code:

```
function strlen(ptr) {
    ptr = ptr|0;
    var curr = 0;
    curr = ptr;
    while (MEM8[curr]|0 != 0) {
        curr = (curr + 1)|0;
    }
    return (curr - ptr)|0;
}
```

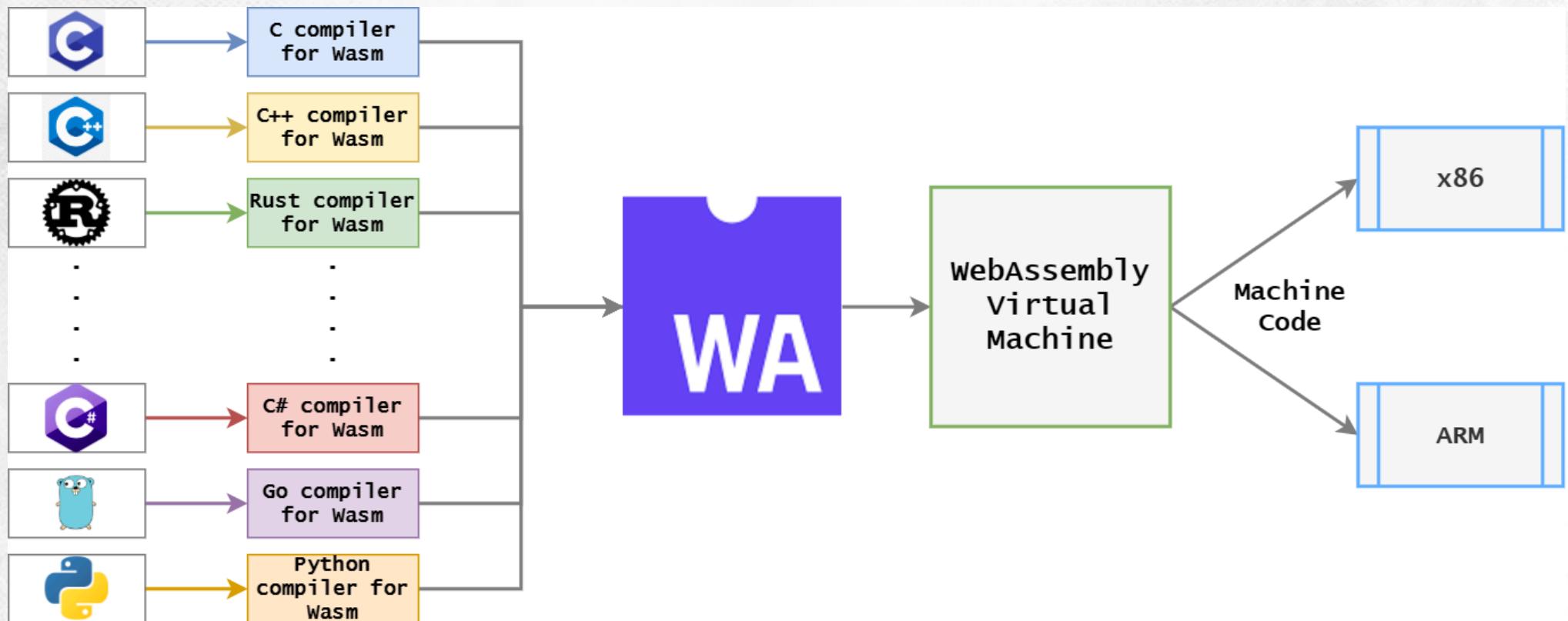
WebAssembly

Original Idea



WebAssembly

Finally, Lightweight VM for all environments



WebAssembly

Rush into Datacenter

Solomon Hykes / [@shykes@hachyderm.io](https://twitter.com/shykes)

@solomonstre · [Follow](#)

If WASM+WASI existed in 2008, we wouldn't have needed to created Docker. That's how important it is. Webassembly on the server is the future of computing. A standardized system interface was the missing link. Let's hope WASI is up to the task!

Lin Clark @linclark

WebAssembly running outside the web has a huge future. And that future gets one giant leap closer today with...

📢 Announcing WASI: A system interface for running WebAssembly outside the web (and inside it too)

[hacks.mozilla.org/2019/03/standa...](https://hacks.mozilla.org/2019/03/standalone-wasi/)

5:39 AM · Mar 28, 2019

[Read the full conversation on Twitter](#)

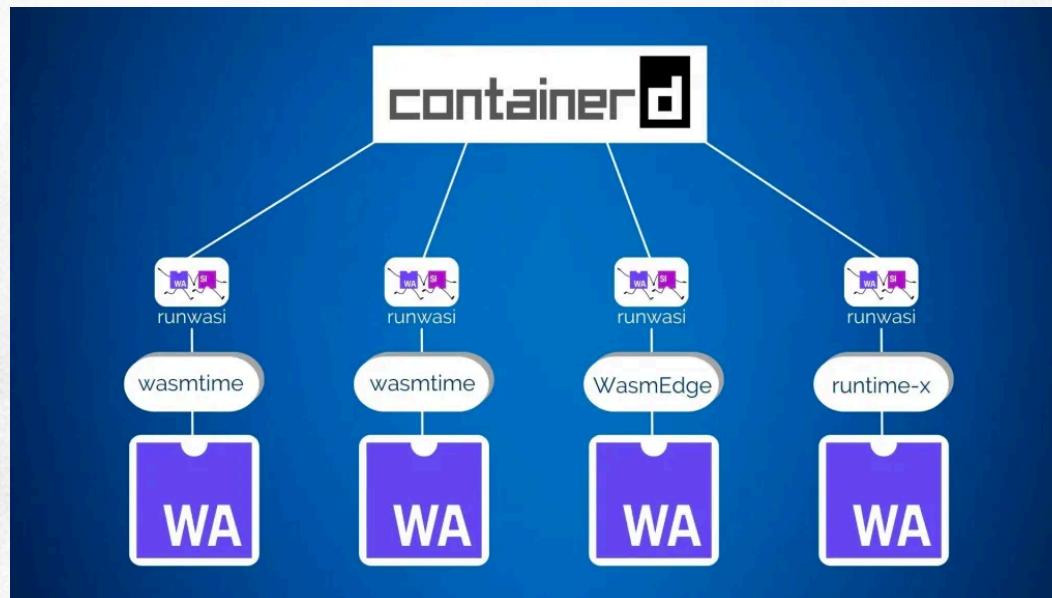
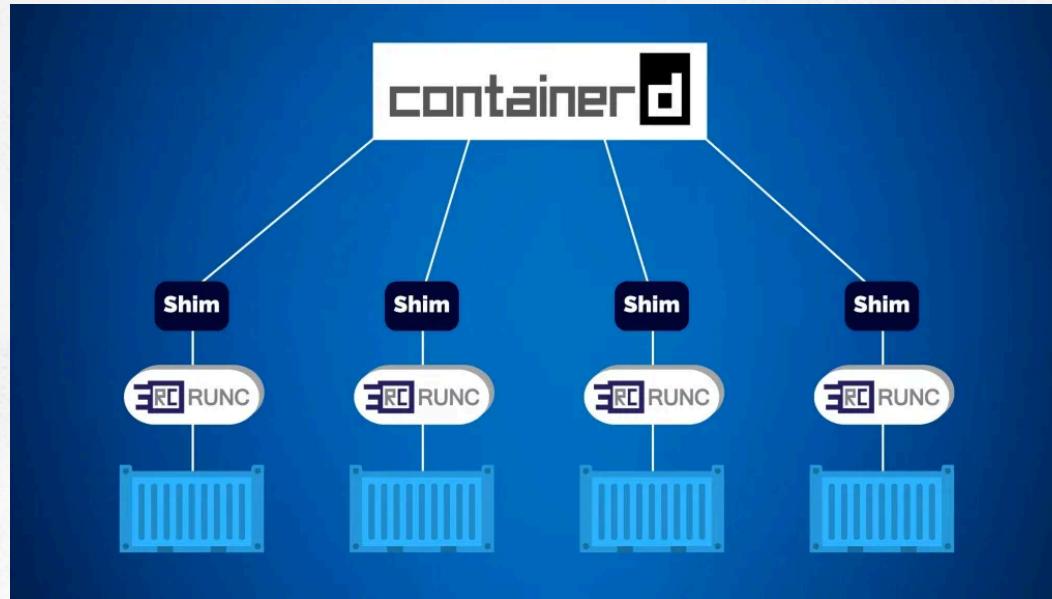
2.2K Reply Copy link

[Read 32 replies](#)



WebAssembly

Rush into Datacenter



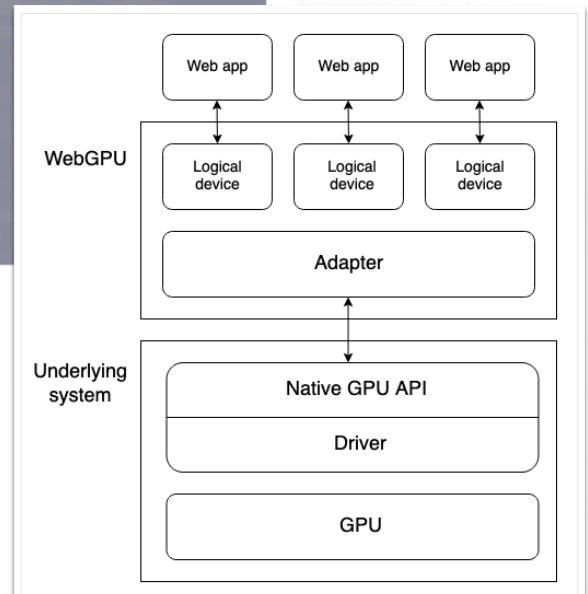
Web with Native GPU



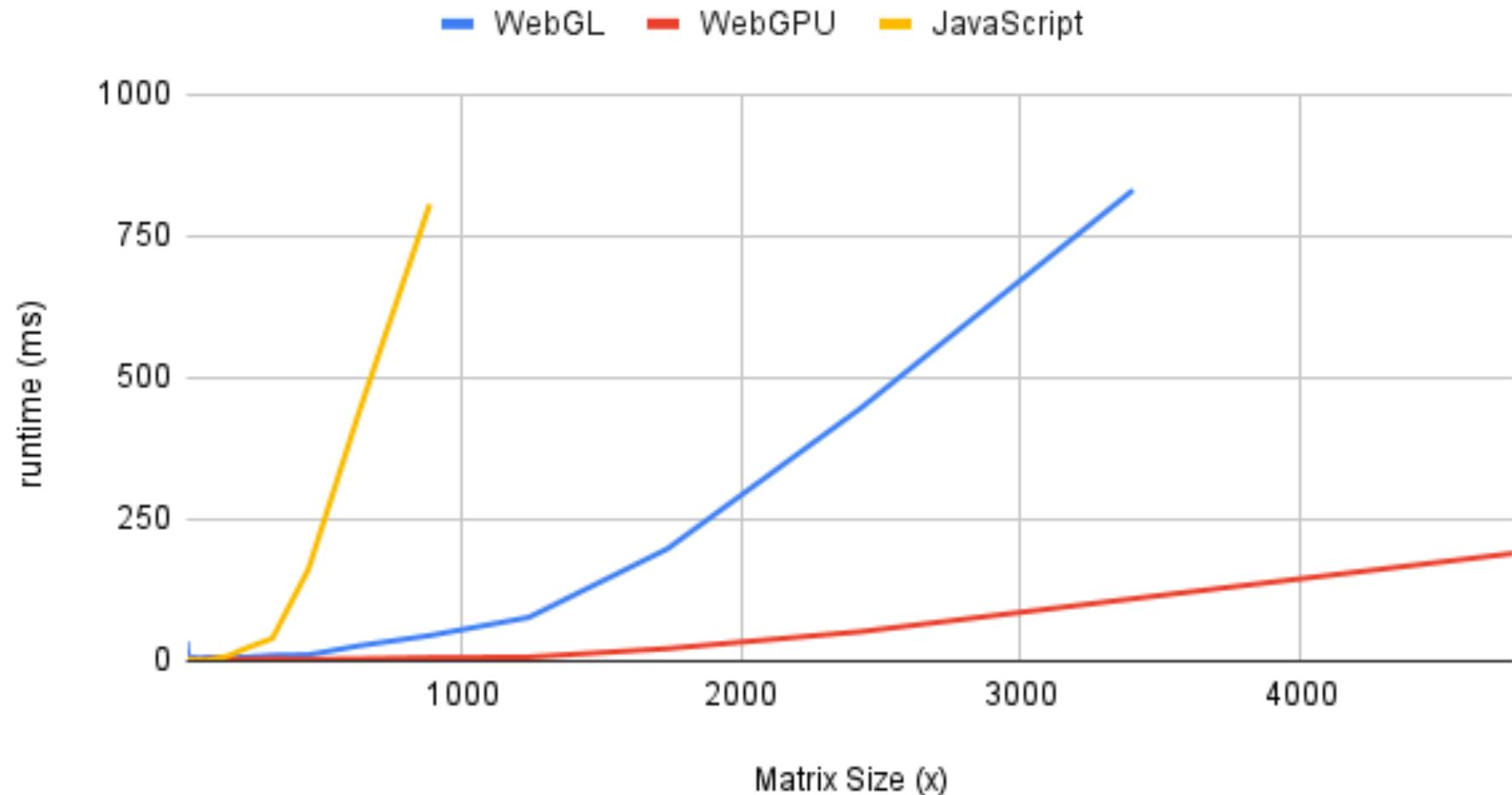
Chrome ships WebGPU

After years of development, the Chrome team ships WebGPU which allows high-performance 3D graphics and data-parallel computation on the web.

Published on Thursday, April 6, 2023



Matrices multiplication benchmark

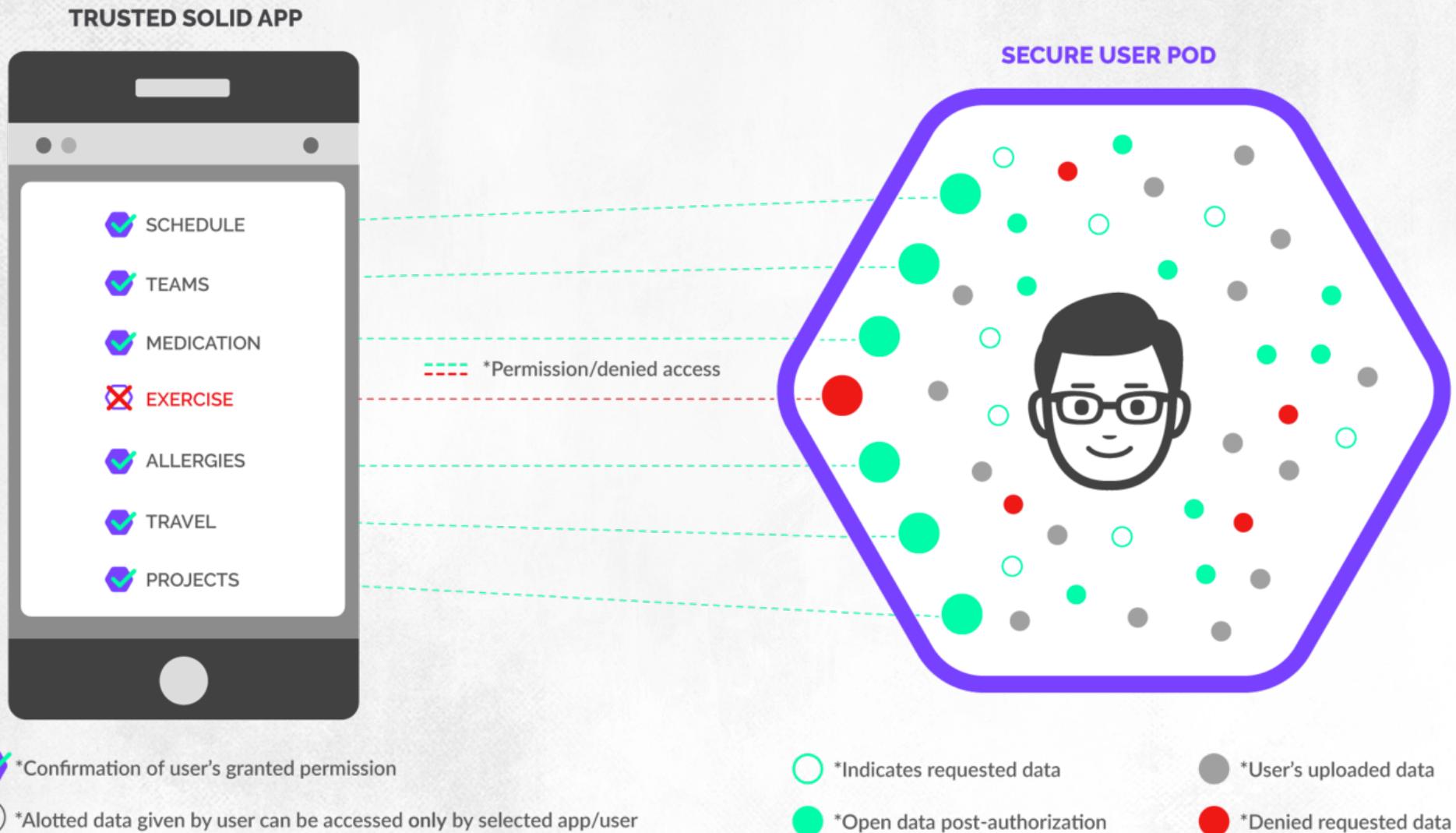


New Internet : Solid Concept

- Solid is an exciting **new project led by Prof. Tim Berners-Lee, inventor of the World Wide Web**, taking place at MIT.
- The project aims to radically change the way Web applications work today, resulting in true data ownership as well as improved privacy.
- Solid (derived from "social linked data") is a proposed set of conventions and tools for building decentralized social applications based on Linked Data principles.
- Solid is modular and extensible and it relies as much as possible on existing W3C standards and protocols.



New Internet : Solid Idea





Thank you