

http2

ce qui va changer !

Julien Landuré
DevFest Paris, Février 2016
#DevFestParis #GDG

/me



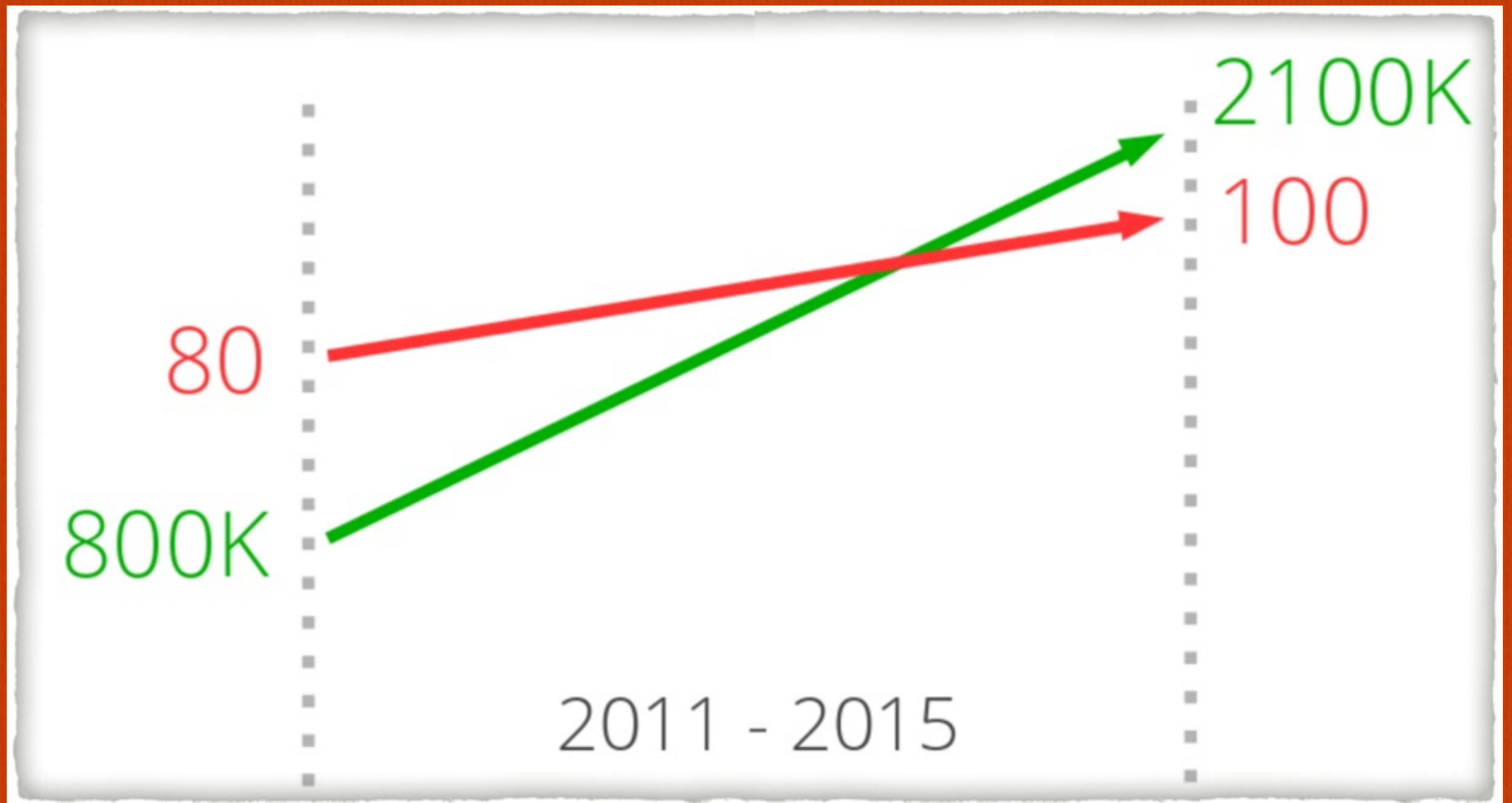
-
- ☐ **Consultant & formateur Web / Cloud / Docker**
 - ☐ **CTO Zenika Nantes**
 - ☐ **Fondateur du GDG Nantes**
 - ☐ **Organisateur du DevFest Nantes**
 - ☐ **@jlandure +JulienLanduré**

[http1...](http://1...)



Contexte

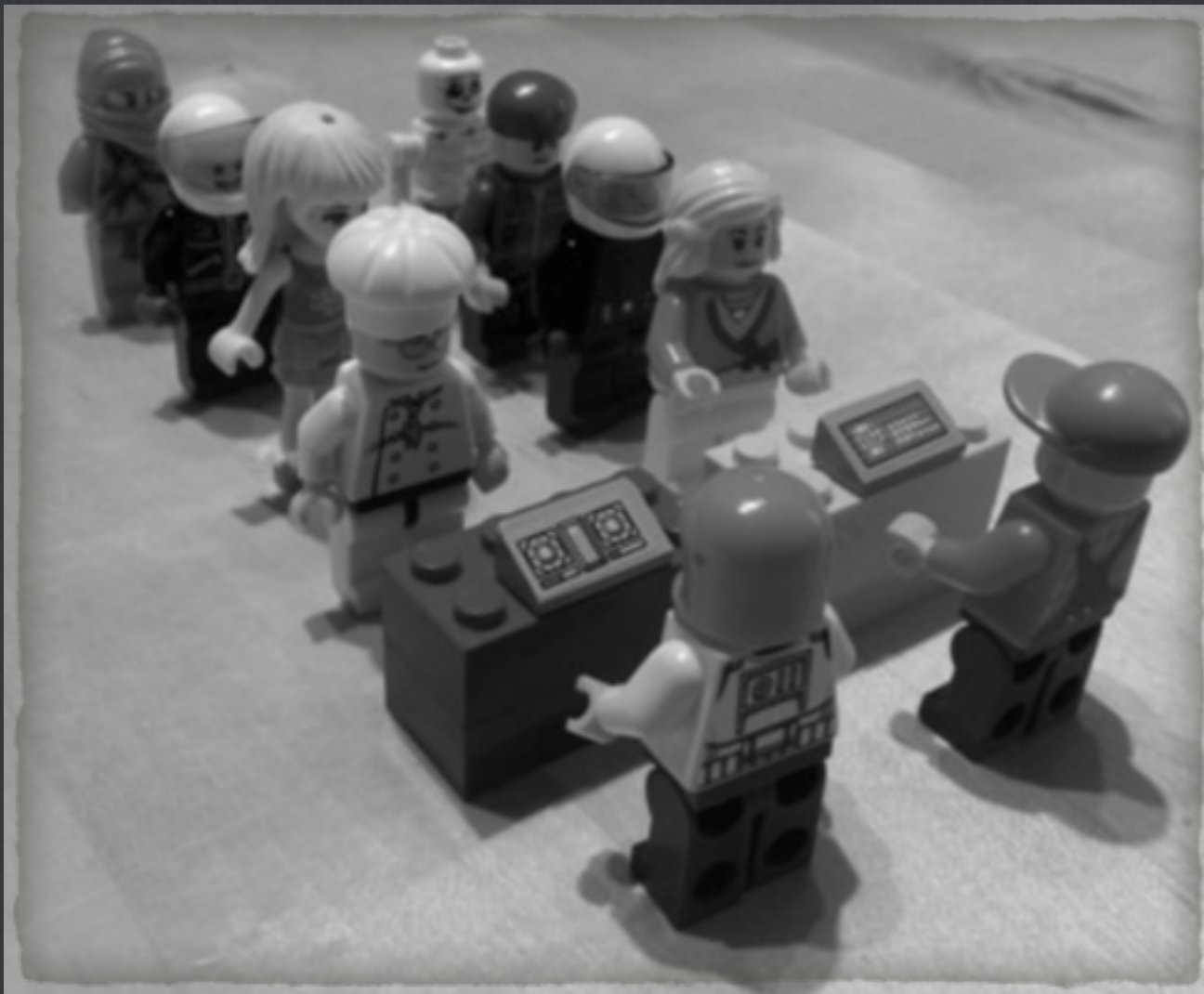
- ☐ **http omniprésent, base du web**
- ☐ **Protocole énorme (60 pages de RFC puis 176 pages)**
- ☐ **Une flopée d'options**
- ☐ **Le web est de plus en plus utilisé (mobile!)**



Volume de données

...et 40% du top 1000 des sites utilise un CDN

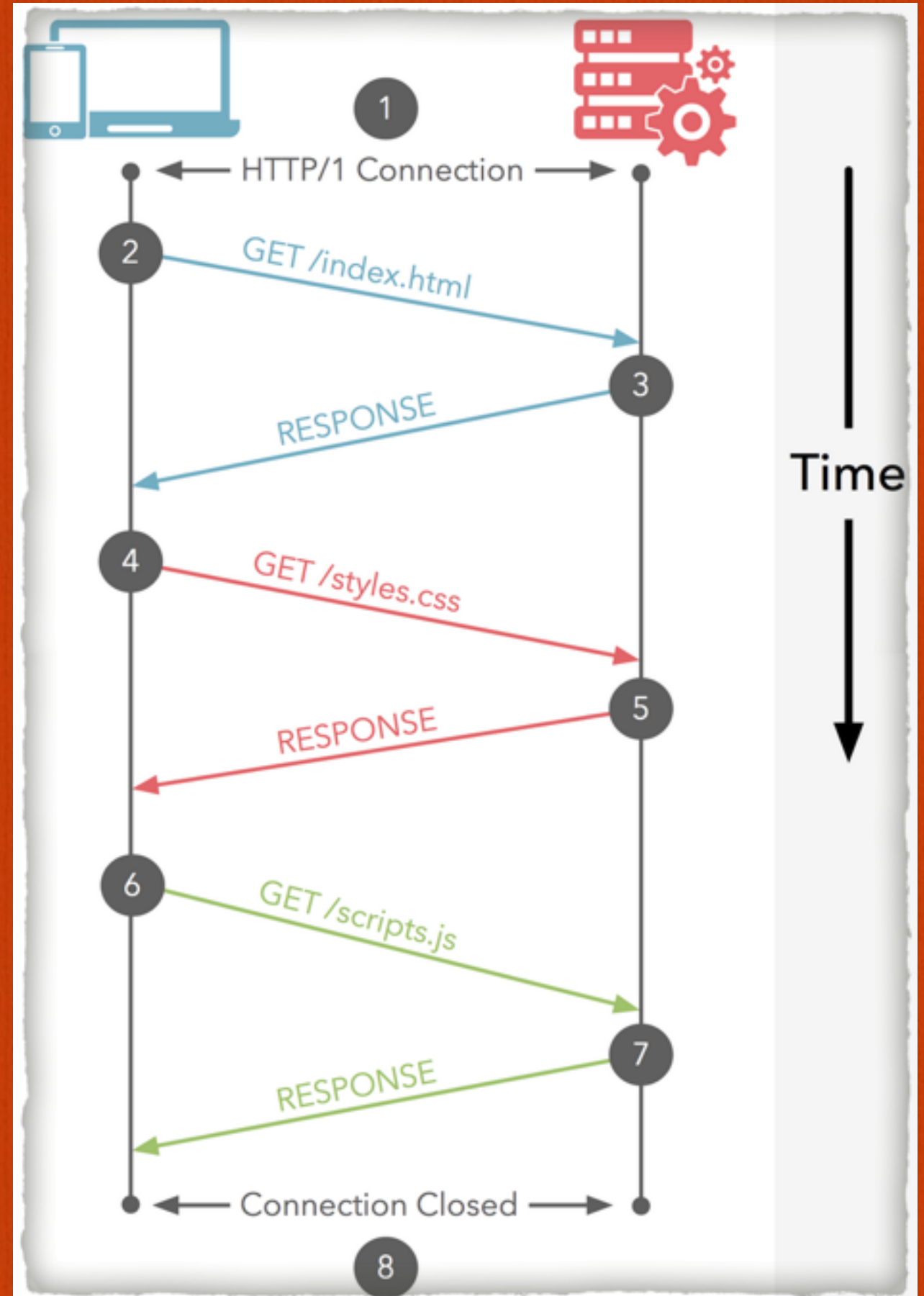
Un problème majeur



« Head of line blocking »

Les navigateurs ouvrent 6 connexions pour améliorer le temps de chargement...

La chargement http1





Les contournements

Utilisés par les top 1000 des sites visités

- ☐ **Image spriting**
- ☐ **Inlining (css, img en base64)**
- ☐ **Concaténation**
- ☐ **Domain sharding**

ONE PROTOCOL



TO RULE THEM ALL

http2 !

Objectif

Rendre le web meilleur

- ☐ **Améliorer le protocole tout en gardant TCP**
- ☐ **Être compatible avec http1, conserver les uris existantes**
- ☐ **Baisser la latence, moins d'aller retours**
- ☐ **Eviter de créer pleins de connexions vers le même hôte**
- ☐ **Charger les ressources rapidement**
- ☐ **En gardant un protocole ouvert et normalisé (sans version mineure)**

Historique

- ☐ **http1 nommé RFC 1945 sorti en 1996**
- ☐ **http1.1 nommé RFC 2616 sorti en 1999 (+ annexes)**
- ☐ **SPDY, protocole « Google » sorti en 2009**
- ☐ **http2 nommé RFC 7540 sorti le 15 mai 2015**

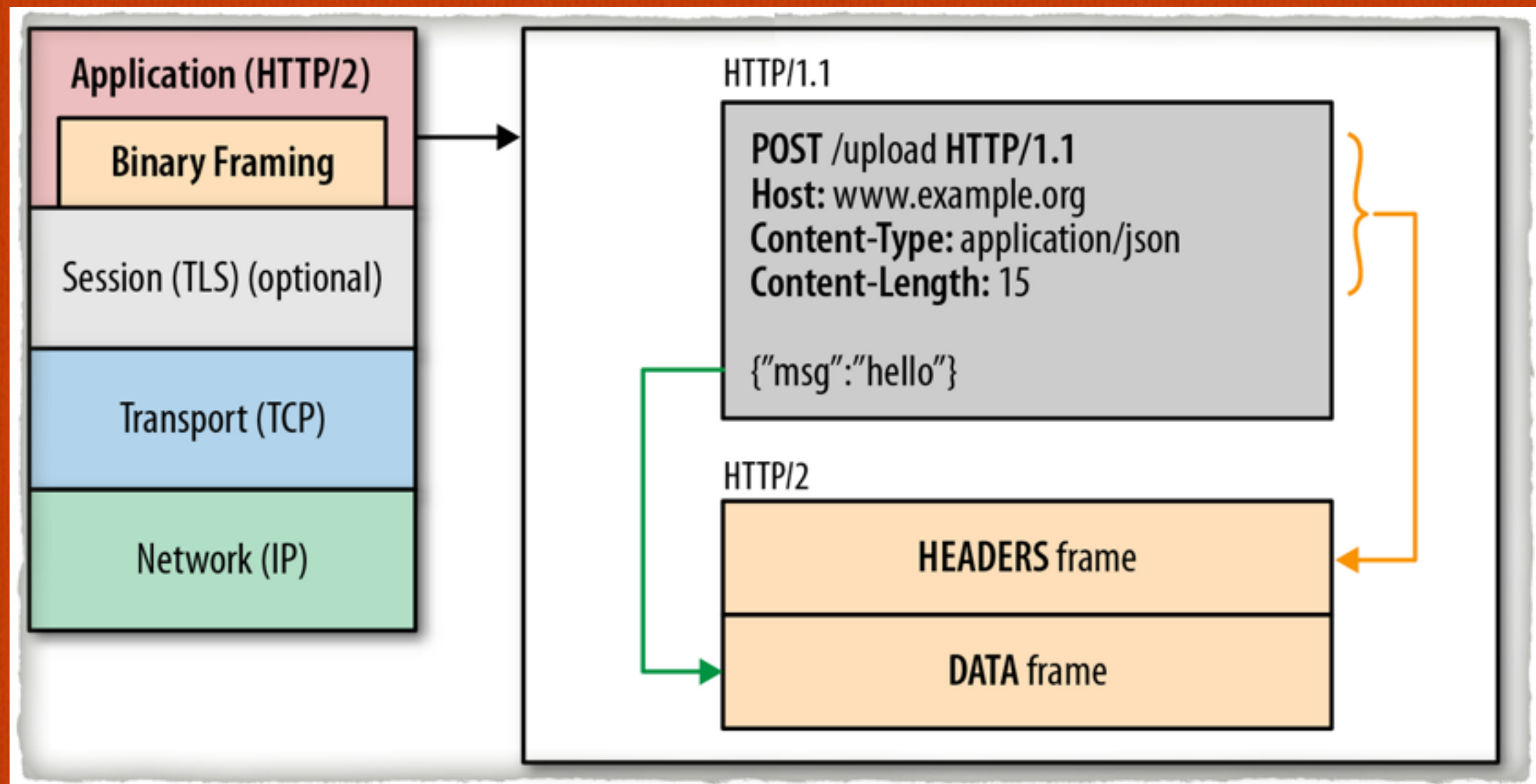


Dis-moi Jamy, c'est quoi http2 ?

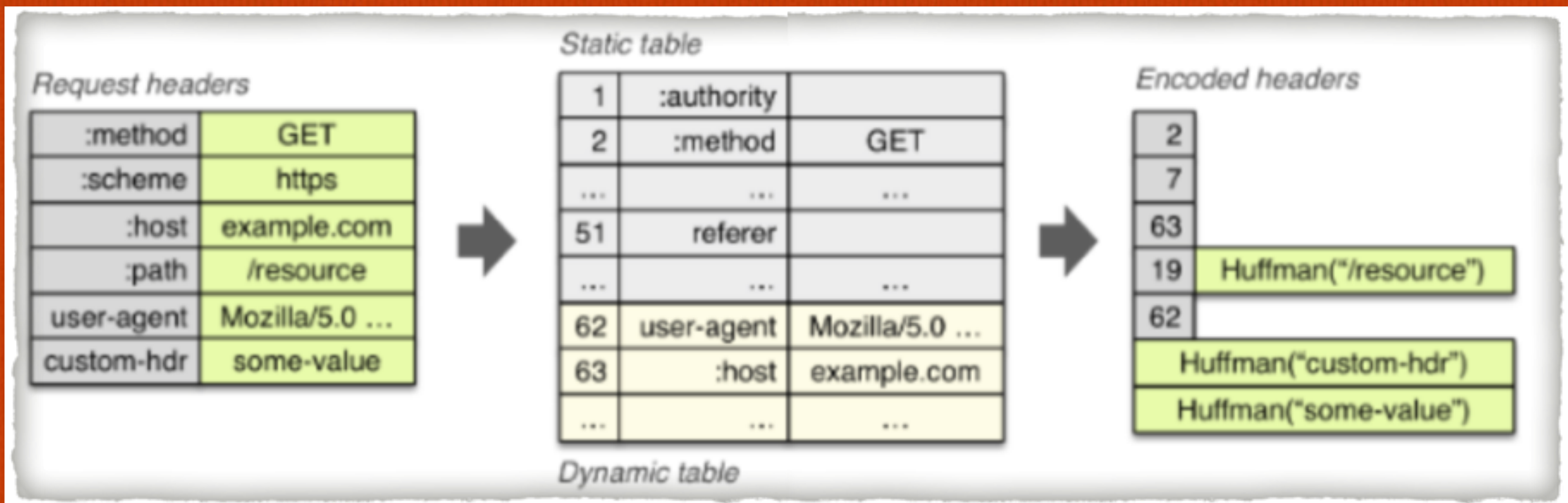
http2

- ☐ un protocole binaire, compressé avec des entêtes compressées
- ☐ une méthode pour passer de http1 vers http2
- ☐ prévue (surtout) pour fonctionner avec TLS
- ☐ utilise une seule connexion avec le serveur
- ☐ permet d'indiquer des priorités aux requêtes
- ☐ permet le multiplexage : plusieurs flux simultanés sur une même connexion
- ☐ permet le load-balancing

Les couches & http2



La compression HPACK





Le multiplexage en image



Upgrade to http2

```
GET /page HTTP/1.1
Host: server.example.com
Connection: Upgrade, HTTP2-Settings
Upgrade: h2c 1
HTTP2-Settings: (SETTINGS payload) 2
```

```
HTTP/1.1 200 OK 3
Content-length: 243
Content-type: text/html
```

(... HTTP/1.1 response ...)

(or)

```
HTTP/1.1 101 Switching Protocols 4
Connection: Upgrade
Upgrade: h2c
```

(... HTTP/2 response ...)

Notions avancées : Prioritization

- ☐ L'idée : donner des priorités aux ressources pour qu'elles soient téléchargées en premier
- ☐ Priorité définie de 1 à 256 et modifiable « on the fly »
- ☐ Très pratique pour charger la bonne image de fond si on scroll

Notions avancées : Server Push

- ☐ L'idée : pousser de la donnée avant que le client la trouve dans la page
- ☐ La ressource sera mise en cache et récupérer directement si le client la demande
- ☐ Très pratique pour envoyer directement les fichiers CSS, JS utilisés dans le « index.html »

```
1 link: </stylesheets/screen.css>; rel=preload; as=stylesheet
```

Notions avancées : Server Hints

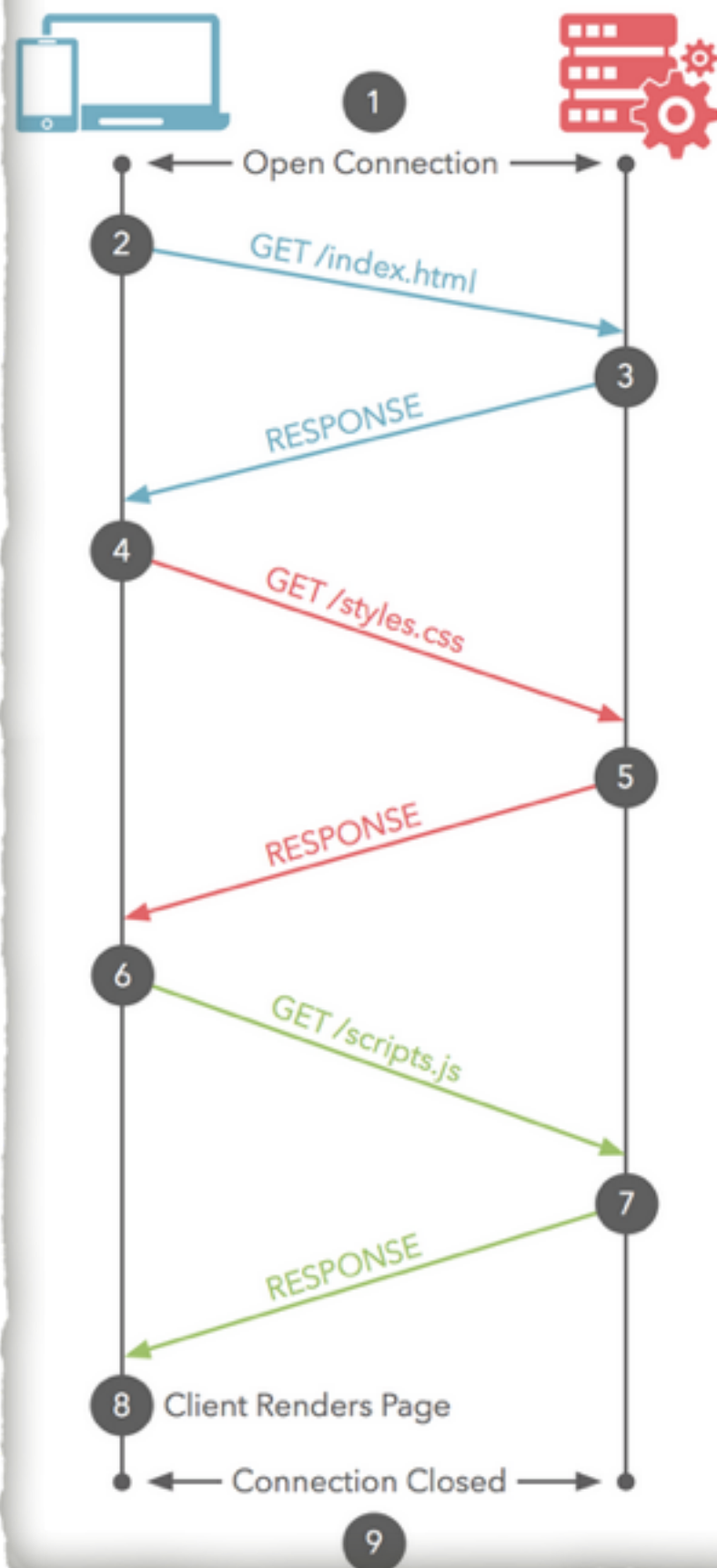
- ❑ L'idée : pousser les futures urls/ressources demandées sans que le client sache encore qu'il en ait besoin
- ❑ La ressource sera téléchargée quand le navigateur aura un peu de temps

Link: <<https://example.com/images/large-background.jpg>>; rel=prefetch

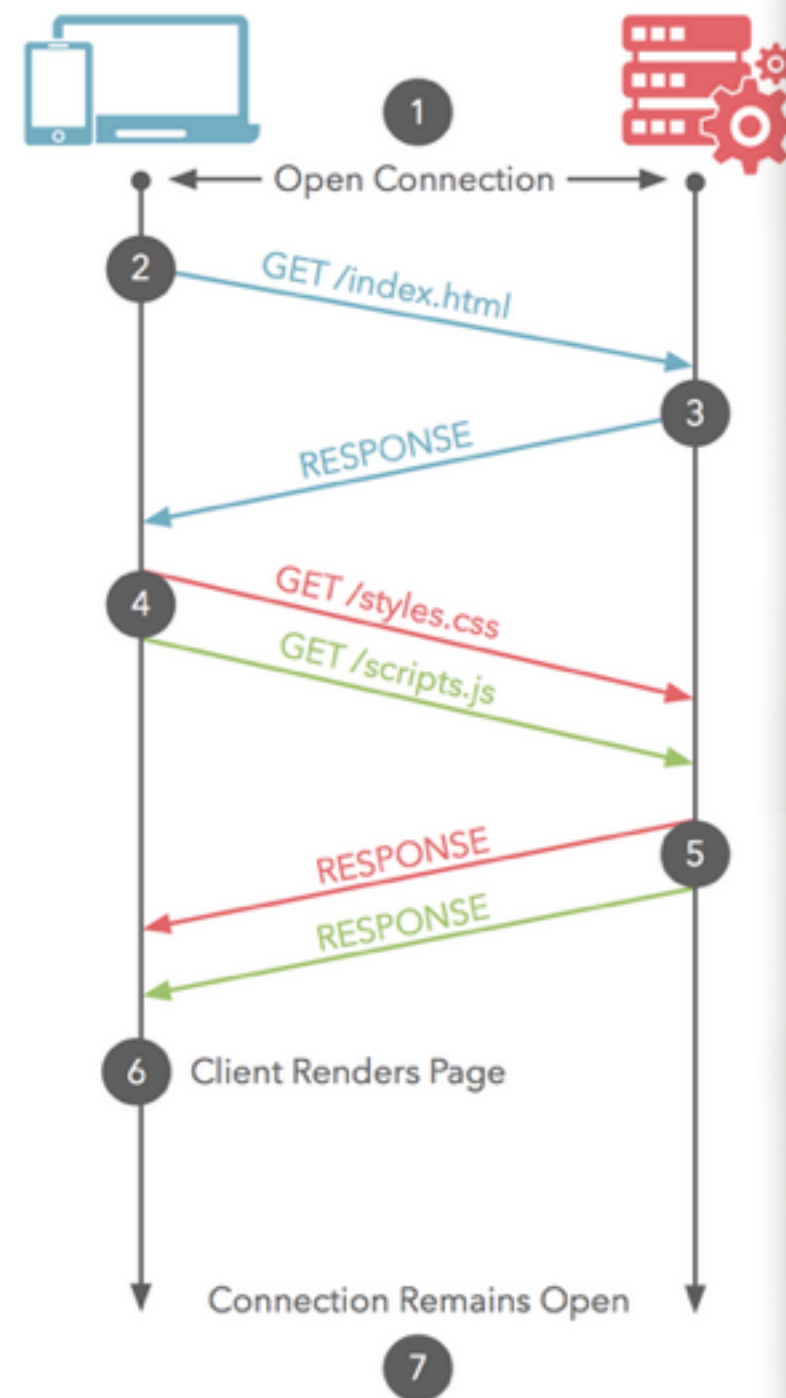


Qu'est ce qu'on gagne ?

HTTP/1.1 Baseline

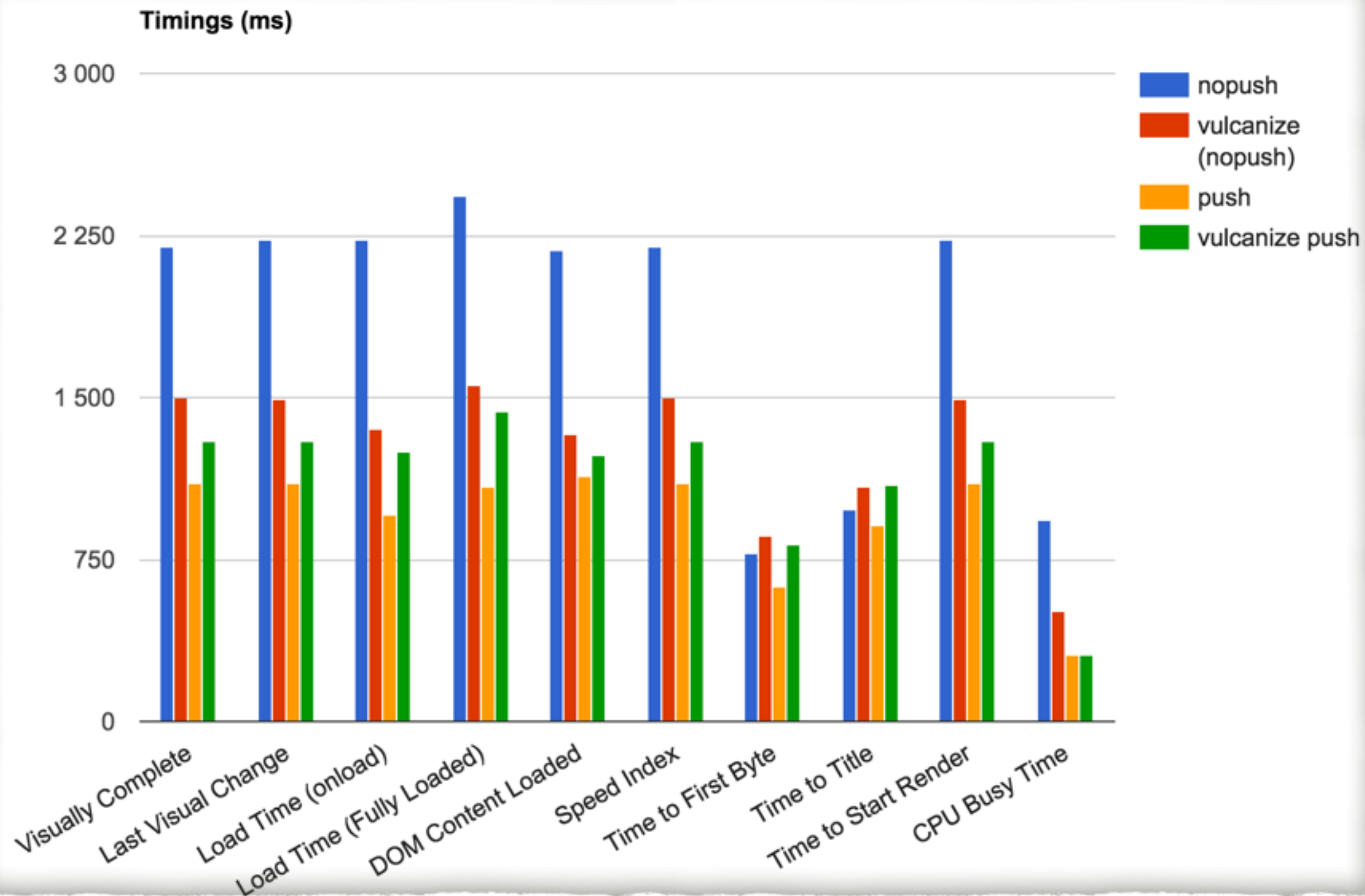


HTTP/2 Multiplexing



Et nos optimisations ?

- ✓ ~~Image spriting~~ : obligé de charger toutes les images
- ✓ ~~Inlining (css, img en base64)~~ : gain à faire des fichiers HTML légers et expirables facilement, facilité de mise à jour
- ✓ ~~Concaténation~~ : gain à faire des fichiers HTML légers et expirables
- ✓ ~~Domain sharding~~ : perte de temps dans la négociation DNS
- Mais attention, de gros fichiers permettent globalement une meilleure compression...



source : <https://http2-push.appspot.com/>

Démo ?

<https://http2.golang.org/gophertiles?latency=30>



HTTP2, HTTP2 EVERYWHERE



Et la suite ?

Ca va marcher ?

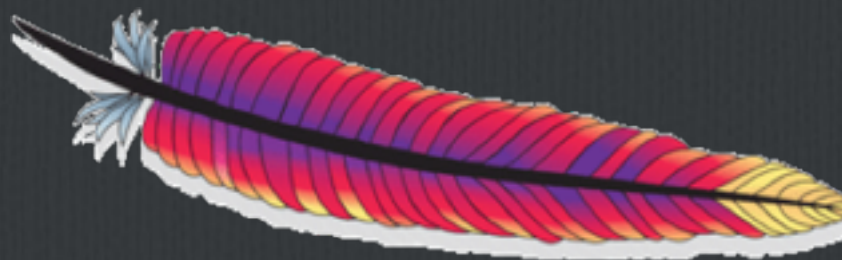
L'adoption est bonne

- Outils : Curl, Wireshark
- Langages : Go(lang), Python, PHP, NodeJS, Java etc.
- Serveurs : HA Proxy, Squid, Varnish, Nginx, H2O, Jetty, IIS, Apache

NGINX



VARNISH
CACHE



HAPROXY

jetty://

Ca va marcher ?



IE	Edge	Firefox	Chrome	Safari	Opera
8					
9			² 45		
10	² 12	² 42	² 46		
^{1 2} 11	² 13	² 43	² 47	^{2 3} 9	² 34
	² 14	² 44	² 48		² 35
		² 45	² 49		² 36
		² 46	² 50		

Accueil

Notifications

Messages



Recherchez sur Twitter



9 et 10
Novembre 2016

DEVFEST
2016
NANTES

<https://devfest.gdgnantes.com>

Nantes
Cité des Congrès

TWEETS
37

ABONNEMENTS
13

ABONNÉS
239

AIMÉS
28



Abonné

Elements Console Sources Network Timeline Profiles Resources Audits AngularJS

View: Preserve log ☒ Disable cache No throttling

Filter ☐ Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Other

Name	Status	Protocol	Type	Size	Time	Timeline - Start Time
☐ timeline.5e2177ca4c9b63db2e1348955033ccf08b...	200	http/1.1	script	99.9 KB	152 ms	
☐ trends?k=a258d0d802&pc=true&profileUserId=35...	200	h2	xhr	823 B	177 ms	
☐ media_timeline?for_photo_rail=true&last_note_ts=...	200	h2	xhr	878 B	663 ms	
☐ 686988431499472896?advertiser_name=Periscop...	200	h2	docu...	1.9 KB	170 ms	
☐ log.json?earned=false&event=impression&impress...	200	h2	xhr	137 B	148 ms	
☐ log.json?earned=false&event=impression&impress...	200	h2	xhr	89 B	149 ms	
☐ analytics.js	200	quic/1+spdy/3	script	10.8 KB	34 ms	

Ca va marcher ?

- ☐ Google avec SPDY a montré l'adoption d'un protocole qui a pris 1% des requêtes mondiales
- ☐ Il faudra toujours faire cohabiter http1 et http2... pour les vieux browser
- ☐ Conseil : utiliser https... mais attention : au revoir les proxy !
- ☐ Tentative sur UDP : Quic powered by Google

HTTP2 IS COMING

AND IT'S COOL!



Merci !
@jlandure
#DevFestParis #GDG

Ressources

- **HTTP2 explained**
- **HTTP/2 for a web faster by Benjamin Patch**
- **NGINX White paper on http2**
- **High Performance Browser Networking by Ilya Grigorik**
- **A simple performance comparison of https, spdy and http2**
- **http2 push tester**
- **http2 gophertiles**