INF5190

PROGRAMMATION WEB AVANCÉ

AUTOMNE 2019

JEAN-PHILIPPE CAISSY

CAISSY.JEAN-PHILIPPE@UQAM.CA

HTTPS://CAISSY.DEV/INF5190

DESCRIPTION DU COURS

Initier les étudiant-e-s aux ***méthodes avancées et aux bonnes pratiques*** de l'industrie pour la conception, le développement, la maintenance et le déploiement d'applications Web modernes.

PRÉALABLES

INF3080 - Bases de données

INF3190 - Introduction à la programmation Web

OBJECTIFS DU COURS

- Comprendre les fondements du développement logiciel d'applications Web
- Familiariser l'étudiant au cadre de développement web MVC (Model-View-Controller)
- Comprendre les différents composants d'une application web

OBJECTIFS DU COURS

- Connaître les différents formats de sérialisations
- Introduire l'étudiant-e aux différents systèmes de base de données
- Pouvoir concevoir des micro-services et l'interopérabilité de ceux-ci

OBJECTIFS DU COURS

- Différencier et comprendre l'utilisation des différents types d'authentifications et d'identifications
- Pouvoir maintenir et diagnostiquer les problèmes et exceptions d'une application web
- Pouvoir effectuer des tests de charges sur une application web
- Comprendre et effectuer le déploiement d'application web

TECHNOLOGIES VUES EN COURS

- Langage : Python, Django ou Flask (à confirmer)
- Stockage : MySQL, Redis, Memcached
- Environment de déploiement : Docker, Kubernetes, Google Cloud Platform (GCP)
- ... ?

CONTENU DU COURS

1. Développement d'applications web avec Python
2. Examen intra
3. Interopérabilité, déploiement et maintenance
4. Examen final
5. Remise du travail de session

CALENDRIER

https://github.com/jpcaissy/INF5190/blob/master/plan-de-cours.md#horaire

L'horaire est donné à titre indicatif. Le contenu de chacun des cours peut changer en cours de route.

ENSEIGNEMENT

- Cours magistraux (11 cours, 2 révisions et 2 examens)
- Laboratoires
- Travail de session

MODALITÉS D'ÉVALUATIONS

Type	Pondération	# cours	Date
Examen intra	35%	8e	23 octobre
Examen final	35%	15e	11 décembre
Projet de session	30%	15e	13 décembre 21h

Seuil de réussite :
60% pour l'ensemble des évaluations
50% pour les examens

EXAMEN

- Aucune documentation permise
- Compréhension générale de la matière
- Majorité réponse courte/choix de réponse

TRAVAIL DE SESSION

- Développement et déploiement d'une application web
- Se rapproche le plus possible d'une situation réelle en industrie
- Seul, ou en équipe de deux
- Aucun retard permis
    - Je prends la version disponible lors de l'échéancier

POINTS BONIS

Erreurs dans les exemples de codes?

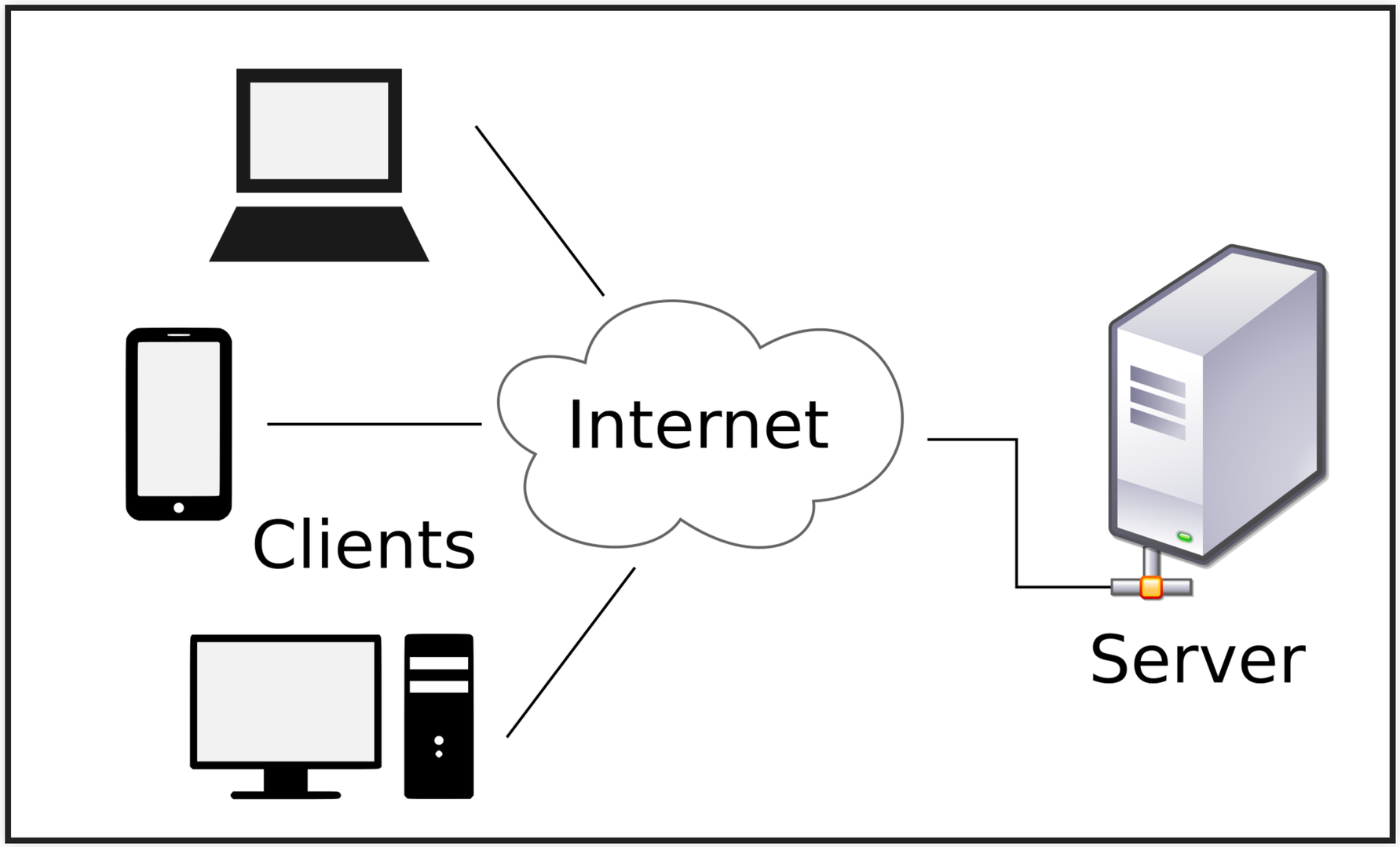
- Je prévois jusqu'à 5 points bonis aux examens (2.5 chaque) si vous trouvez des erreurs dans les exemples de codes que je donne
- Il suffit de créer une PR sur mon répo github du cours et d'y apporter les correcitfs
- Premier arrivé premier servis : c'est en fonction de l'heure à laquelle la PR a été créée
- Le montant des points bonis accordés sont à ma discretion

QUESTIONS?

DÉVELOPPEMENT D'APPLICATIONS WEB

RAPPELS DE BASE

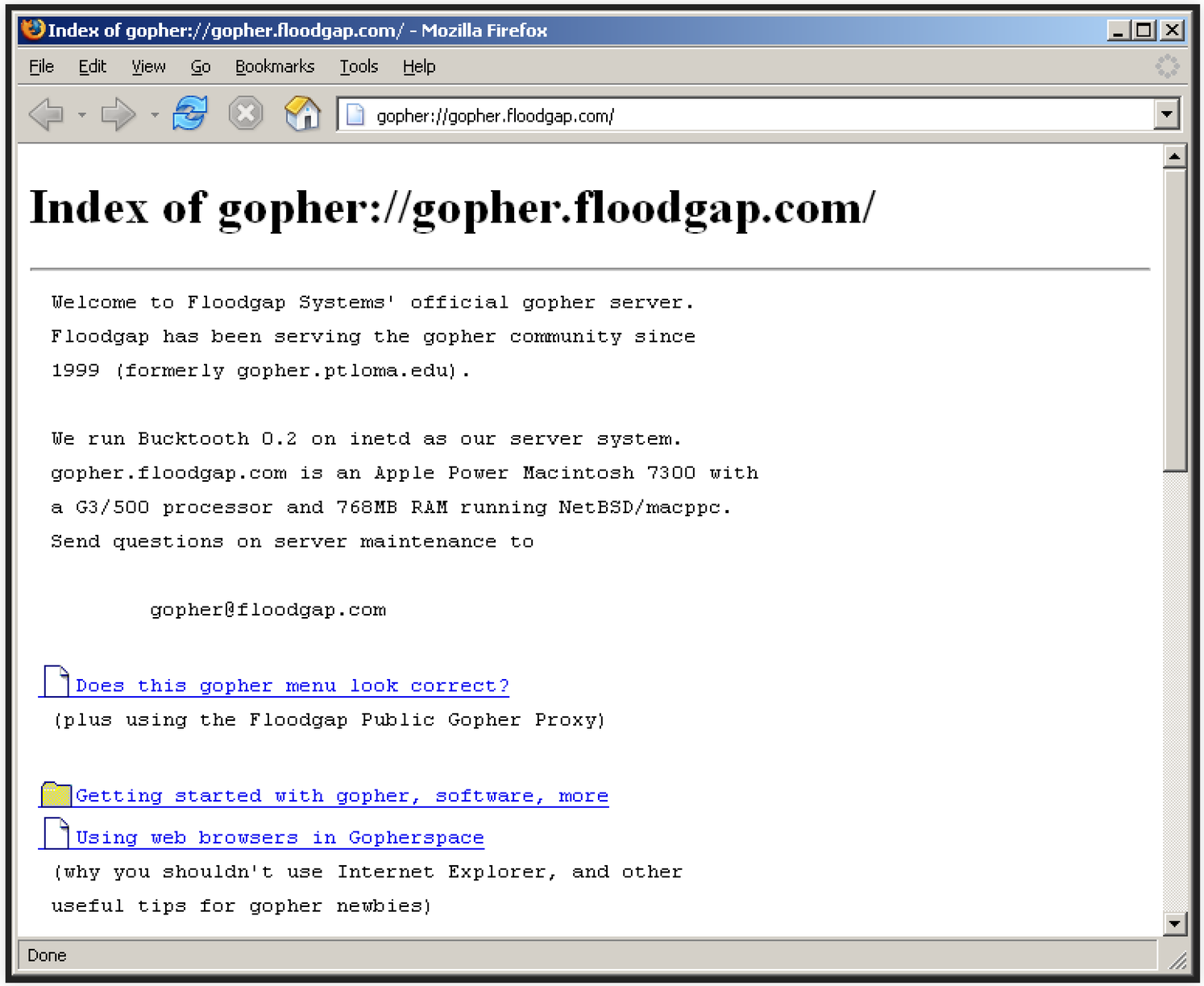
ARCHITECTURE CLIENT-SERVEUR



HISTORIQUE

- Besoin de distribuer et de chercher de l'informations et de documents sur IP
- Évolution des protocles de niveau 7 (couche applicative sur le modèle OSI)
  - Gopher (mi-1991)
  - HTTP 0.9 (1991)
  - HTTP 1.0 (1996)
  - HTTP 1.1 (1997)
  - HTTP 2 (2015)
  - HTTP 3 (2018)

GOPHER


Index of gopher://gopher.floodgap.com/ - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
gopher://gopher.floodgap.com/
Index of gopher://gopher.floodgap.com/
Welcome to Floodgap Systems' official gopher server.
Floodgap has been serving the gopher community since
1999 (formerly gopher.ptloma.edu).
We run Bucktooth 0.2 on inetd as our server system.
gopher.floodgap.com is an Apple Power Macintosh 7300 with
a G3/500 processor and 768MB RAM running NetBSD/macppc.
Send questions on server maintenance to
gopher@floodgap.com
Does this gopher menu look correct?
(plus using the Floodgap Public Gopher Proxy)
Getting started with gopher, software, more
Using web browsers in Gopherspace
(why you shouldn't use Internet Explorer, and other
useful tips for gopher newbies)
Done

GOPHER

- Forte hiérarchie des documents
- Interface textuelle avec menus
- Protocle texte simple
- Facile à utiliser et mettre en place

HTTP 0.9/1.0

- Première version du protocole HTTP
- Connexions **non** persistentes
- États non persistés (protocole *stateless*)
- Aucune compression
- Ne peux pas diviser la réponse du serveur (*chunk transfer*)
- Une seule adresse IP par hôte
- Protocole en format texte

HTTP 0.9/1.0

```
$ nc caissy.dev 80
GET / HTTP/1.0
```

```
HTTP/1.1 301 Moved Permanently
Server: nginx/1.14.2
Content-Type: text/html
Content-Length: 185
Connection: close
Location: https://caissy.dev/

<html>
<head><title>301 Moved Permanently</title></head>
<body bgcolor="white">
<center><h1>301 Moved Permanently</h1></center>
<hr><center>nginx/1.14.2</center>
</body>
</html>
```

HTTP 1.1

- Formalise plusieurs extensions de 1.0
- Ré-utilisation des connexions (*keep-alive*)
- Compressions supportés
- Permet de diviser la réponse du serveur (*chunk transfer*)
- Permet de reprendre le transfert (*byte range transfer*)
- Support multi-hôte (plusieurs domaines sur même IP)
- Rétro-compatible avec HTTP 1.0

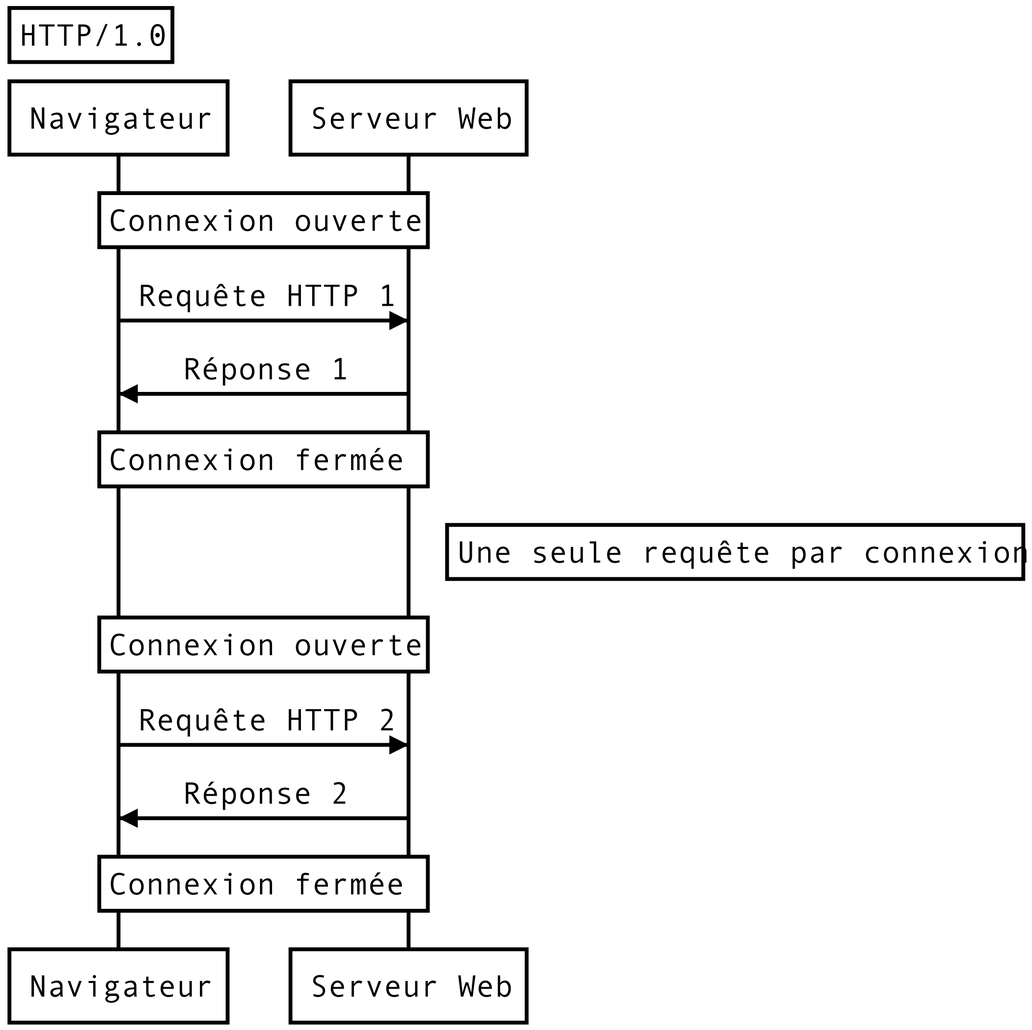
HTTP 1.1

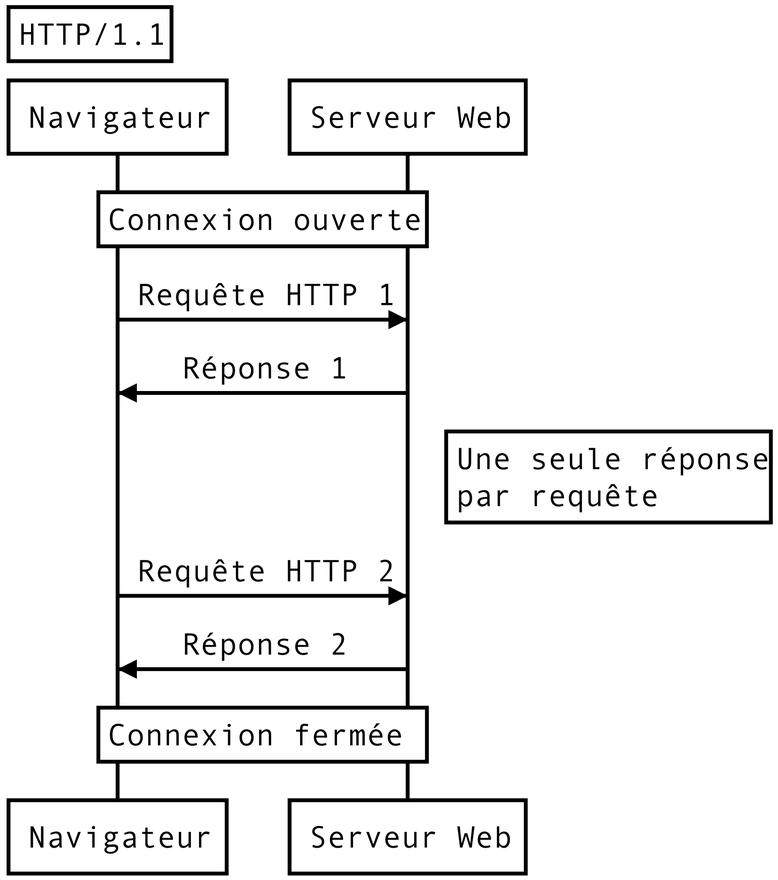
```
$ nc caissy.dev 80
GET /file.data HTTP/1.1
Host: caissy.dev
Range: bytes=125-165
```

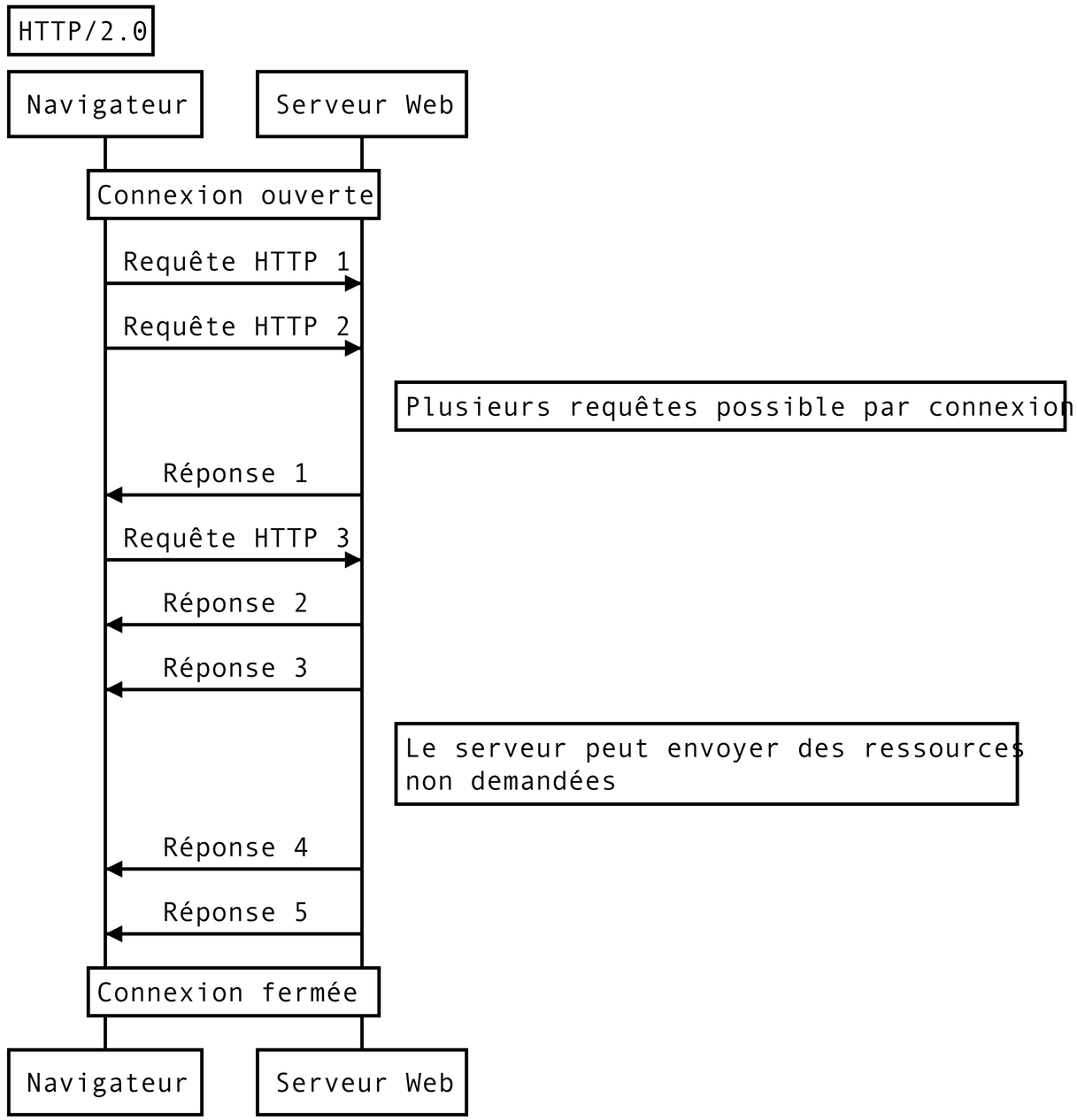
```
HTTP/1.1 206 Partial Content
Server: nginx/1.14.2
Content-Type: application/octet-stream
Content-Length: 41
Connection: keep-alive
Content-Range: bytes 125-165/8199
[...]

data        <------- début tronqué au 125e octet sur 8199 octe
Test data
Test data
Test data
Test d
^C          <------- besoin de fermer la connexion
```

HTTP 2.0

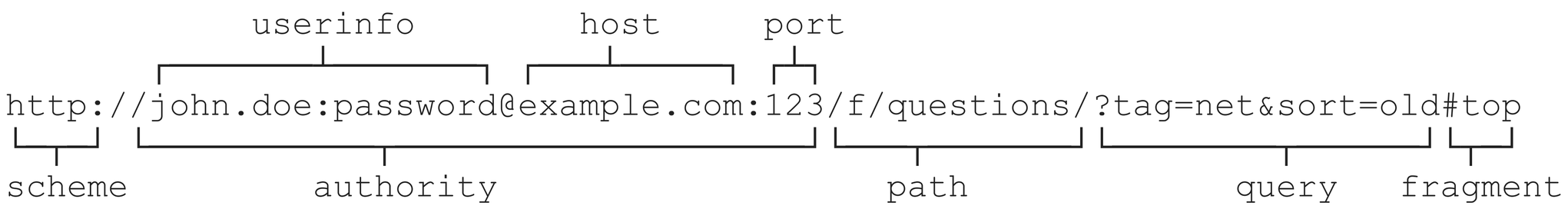
- Changements majeurs au protocole
- Développé initiallement par Google, sous le nom SPDY (Speedy)
- Objectif principal : augmenter la performance et la rapidité d'échange du protocole HTTP
- Compression des entêtes
- Multiplexage des requêtes sur la même connexion TCP
- Permet au serveur d'envoyer des ressources avant que le client le demande (*server push*)


HTTP/1.0
Navigateur
Serveur Web
Connexion ouverte
Requête HTTP 1
Réponse 1
Connexion fermée
Une seule requête par connexion
Connexion ouverte
Requête HTTP 2
Réponse 2
Connexion fermée
Navigateur
Serveur Web


HTTP/1.1
Navigateur
Serveur Web
Connexion ouverte
Requête HTTP 1
Réponse 1
Une seule réponse par requête
Requête HTTP 2
Réponse 2
Connexion fermée
Navigateur
Serveur Web


HTTP/2.0
Navigateur
Serveur Web
Connexion ouverte
Requête HTTP 1
Requête HTTP 2
Plusieurs requêtes possible par connexion
Réponse 1
Requête HTTP 3
Réponse 2
Réponse 3
Le serveur peut envoyer des ressources non demandées
Réponse 4
Réponse 5
Connexion fermée
Navigateur
Serveur Web

RESSOURCE D'UNE REQUÊTE HTTP



Source : Wikipedia contributors, "Hypertext Transfer Protocol," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=Hypertext_Transfer_Protocol&oldid=912162942 (accessed September 4, 2019).

REQUÊTE HTTP

MÉTHODE DE REQUÊTE

Nom	Description
GET	Obtenir une ressource. Ne devrait pas modifier une ressource.
HEAD	Similaire à GET, mais sans la réponse du serveur. Sert à récupérer les en-têtes (*headers*).
POST	Soumettre une entité à une ressource (ex: envoi de formulaire). Changement d'état possible.
PUT	Remplace la ressource existante avec le *payload* envoyé.
DELETE	Supprime une ressource spécifique.

D'autres méthodes existes : OPTIONS, TRACE, PATCH, CONNECT

SITES STATIQUES

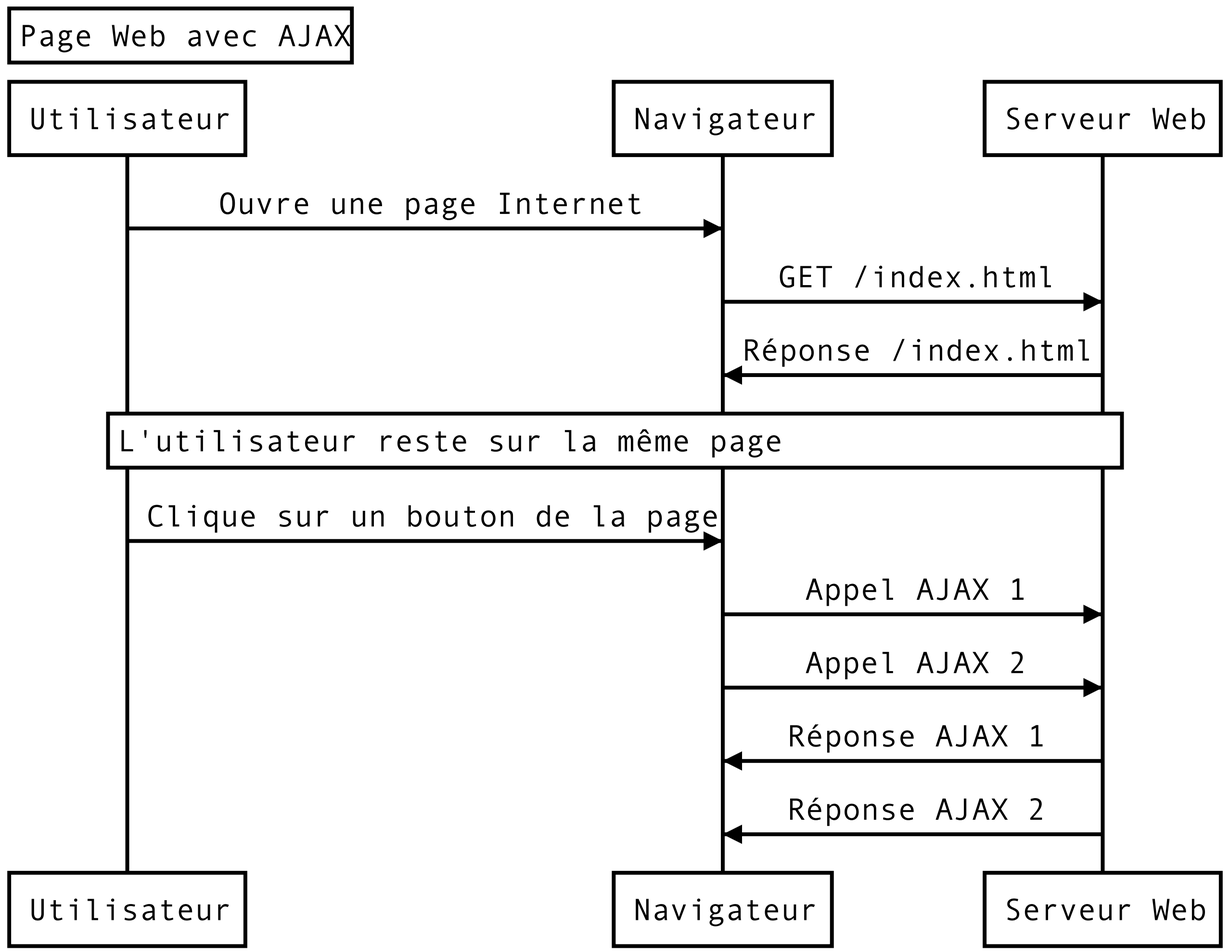
- Contenu d'une page web qui ne varie pas selon la requête
- Même contenu pour tous les visiteurs
- Souvent de simples fichiers exposés à travers un serveur Web
- Rapide, performant et plus sécuritaire que les sites non statiques
- Génération automatisé de contenu : Jekyll, Hugo, etc

PERSONALISATION DE SITE STATIQUE

- Plugins de navigateurs (Flash, Java applets, Silverlight, ActiveX)
  - Avantages : exécuté sur le navigateur
  - Désavantages : obsolète, non standardisé, vecteur d'attaque, peu de flexibilité
- Javascript côté client
  - Avantages : standardisé, simple, rapide, accessible sur les navigateurs majeurs
  - Désavantages : consomme beaucoup de ressources, vecteur d'attque

AJAX

- Envoie et réception de requêtes web asynchrone par le client (navigateur)
- Pas besoin de recharger la page Web
- Popularisé par Google dans les années 2005-2006
- Permet la création d'applications Web riche (Gmail, Slack, Google Docs, etc)
- API Javascript disponible dans le navigateur (XMLHttpRequest)


Page Web avec AJAX
Utilisateur
Navigateur
Serveur Web
Ouvre une page Internet
GET /index.html
Réponse /index.html
L'utilisateur reste sur la même page
Clique sur un bouton de la page
Appel AJAX 1
Appel AJAX 2
Réponse AJAX 1
Réponse AJAX 2
Utilisateur
Navigateur
Serveur Web

HTML5

- Version actuelle de HTML
- Étend, améliore et standardise le langage
- Nouveaux APIs: Canvas, Drag & Drop, WebRTC, WebStorage, etc