ENSTA BRETAGNE

Couches applicatives
14/10, suite

La 7ème couche…

Application
Présentation
Session
Transport
Réseau
Liaison
Physique

Applications :
HTTP, FTP, IMAP, SMTP,
VoIP, SIP, proprietary Skype, XMPP,
Telnet, SSH, DNS, …

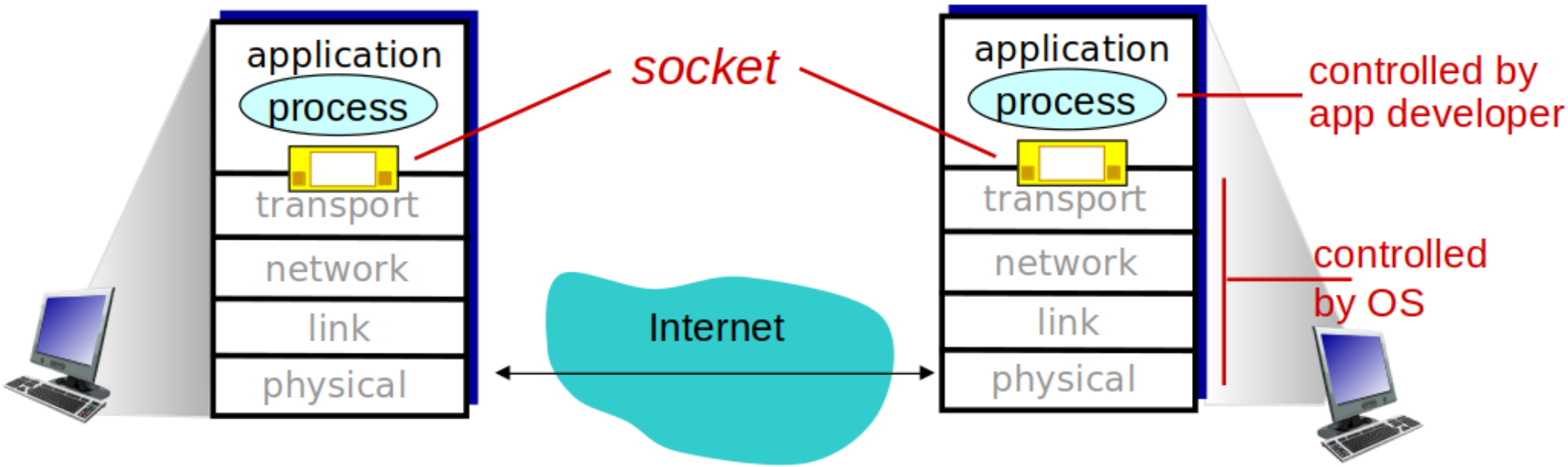
Pas forcément de normes pour les protocoles applicatifs

- Chaque application définit sa nomenclature
- Données reçues : suite d'octets

Focus du cours :

- DNS : le service de résolution des noms de domaine
- HTTP: le protocole pour votre navigateur internet

Sockets et couche transport



Sockets

Le développeur écrit le code du programme exécuté comme un processus, utilisant l'interface de socket avec la couche transport.

Programme de ce cours

- Le service DNS
- HTTP
- Les protocoles pour les emails
- Conclusion

Définition incrémentale au cours de ce cours :
Mais #$$!!:/¨^^… c'est quoi, le web ?

Le service DNS

Le service DNS

DNS : Domain Name System

Résolution d'un nom de domaine en adresse IP.

Exemple :

```
google.fr.          287     IN   A   216.58.215.35
google.com.         300     IN   A   216.58.215.46
www.google.com.     60      IN   A   216.58.211.164
```

Qui répond à ces questions ? Le serveur DNS

Quelle est son IP ? Il faut la connaître…

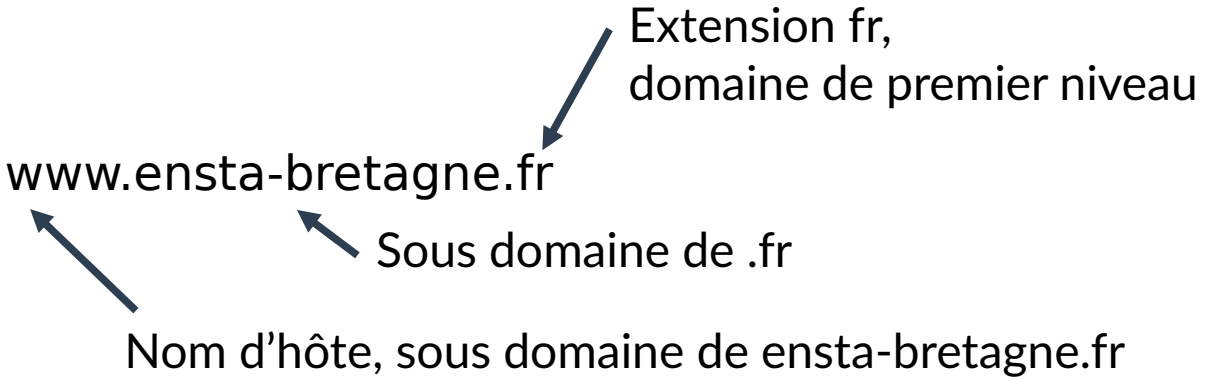
Serveur DNS bien connu : 8.8.8.8 (dns.google.com => Démo)

Noms de domaines

Sert d'identifiant pour identifier des ensembles d'ordinateurs.

Exemple : ensta-bretagne.fr
Domaine de premier niveau : .fr
Domaine de second niveau : ensta-bretagne.fr
Sous domaine : partage.ensta-bretagne.fr
Elément : caractères, chiffres, trait d'union : 63 max

Les domaines de premier niveau sont gérés par :
- AFNIC (.fr, .paris, .bzh, ...)
- Verisign (.com, .net, ...)
- U.S. military (.mil, ...)



Noms de domaine de premier niveau

Trois grandes familles :

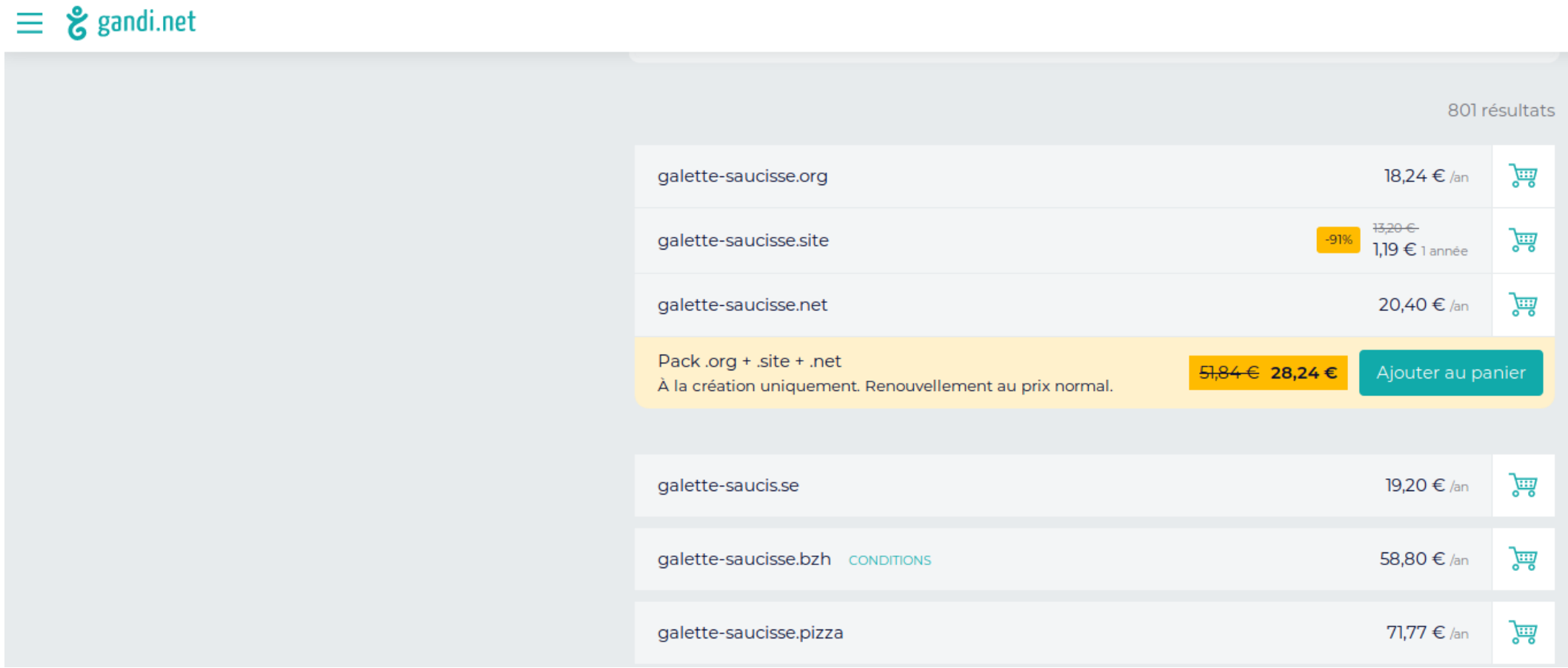
- Les nationaux : .fr, .ca, ...

- Les génériques :
  - Ouverts : .com, .net : attachés à une signification mais sans contrainte particulière
  - Parrainés : .org, .edu, .gov, .mil
  - Restreints : .name, .nom.fr, .pro

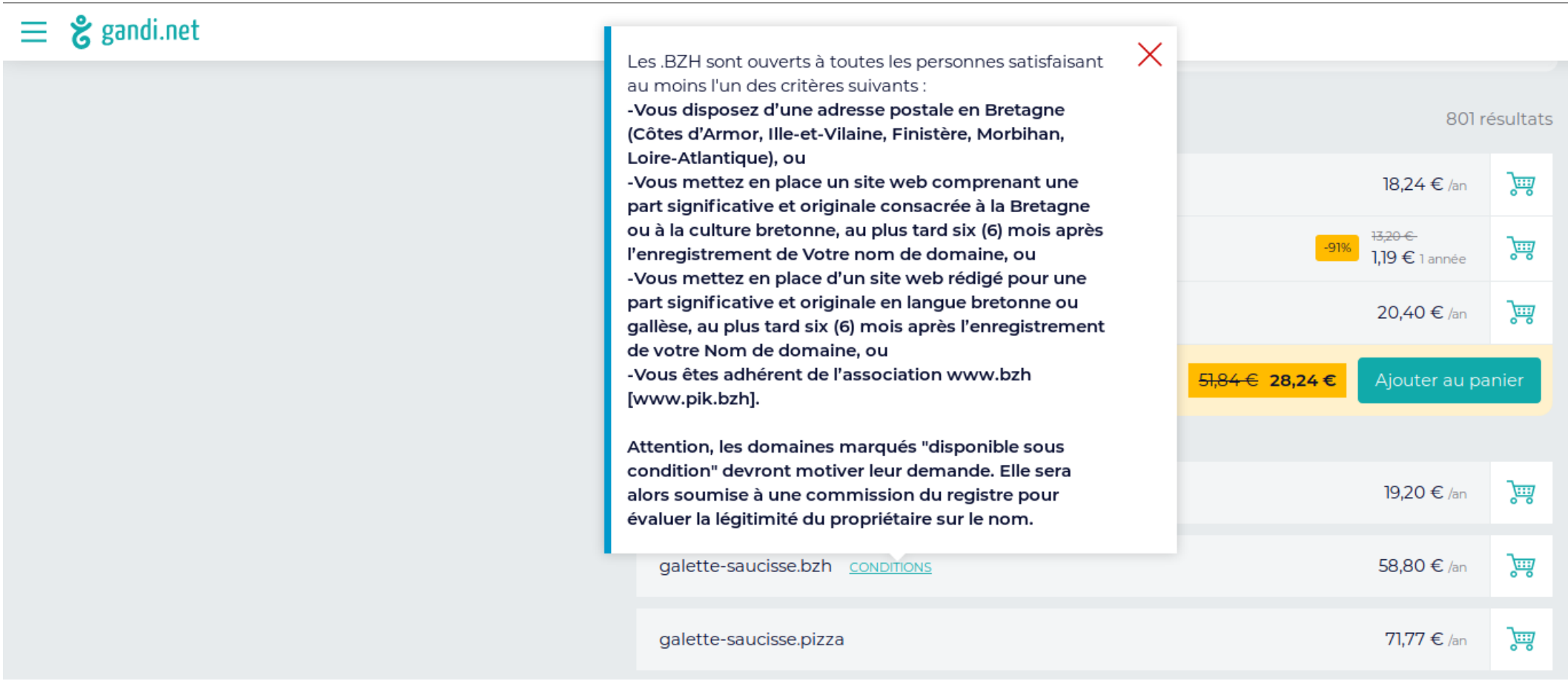
- Les réservés : .example, .invalid, .localhost, .test

- Les nouveaux (2000 à 2012) : .paris, .bzh, .pizza, .alsace, .science

Exemple I


gandi.net
801 résultats
galette-saucisse.org
18,24 € /an
galette-saucisse.site
-91%
13,20 €
1,19 € 1 année
galette-saucisse.net
20,40 € /an
Pack .org + .site + .net
À la création uniquement. Renouvellement au prix normal.
51,84 € 28,24 €
Ajouter au panier
galette-saucis.se
19,20 € /an
galette-saucisse.bzh
CONDITIONS
58,80 € /an
galette-saucisse.pizza
71,77 € /an

Exemple II


gandi.net
Les .BZH sont ouverts à toutes les personnes satisfaisant au moins l'un des critères suivants :
-Vous disposez d'une adresse postale en Bretagne (Côtes d'Armor, Ille-et-Vilaine, Finistère, Morbihan, Loire-Atlantique), ou
-Vous mettez en place un site web comprenant une part significative et originale consacrée à la Bretagne ou à la culture bretonne, au plus tard six (6) mois après l'enregistrement de Votre nom de domaine, ou
-Vous mettez en place d'un site web rédigé pour une part significative et originale en langue bretonne ou gallèse, au plus tard six (6) mois après l'enregistrement de votre Nom de domaine, ou
-Vous êtes adhérent de l'association www.bzh [www.pik.bzh].
Attention, les domaines marqués "disponible sous condition" devront motiver leur demande. Elle sera alors soumise à une commission du registre pour évaluer la légitimité du propriétaire sur le nom.
801 résultats
18,24 € /an
-91%
13,20 €
1,19 € 1 année
20,40 € /an
51,84 € 28,24 €
Ajouter au panier
19,20 € /an
galette-saucisse.bzh
CONDITIONS
58,80 € /an
galette-saucisse.pizza
71,77 € /an

Architecture

DNS : domain name system

- Implémenté de manière hiérarchique
- Impossible à faire en centralisé :
  - Evite le problème du « single point of failure »
  - Evite la surcharge en terme de trafic
  - Mais nécessite plusieurs serveurs et de la réplication

Architecture

DNS : domain name system

- Implémenté de manière hiérarchique
- Impossible à faire en centralisé :
  - Évite le problème du « single point of failure »
  - Évite la surcharge en terme de trafic
  - Mais nécessite plusieurs serveurs et de la réplication

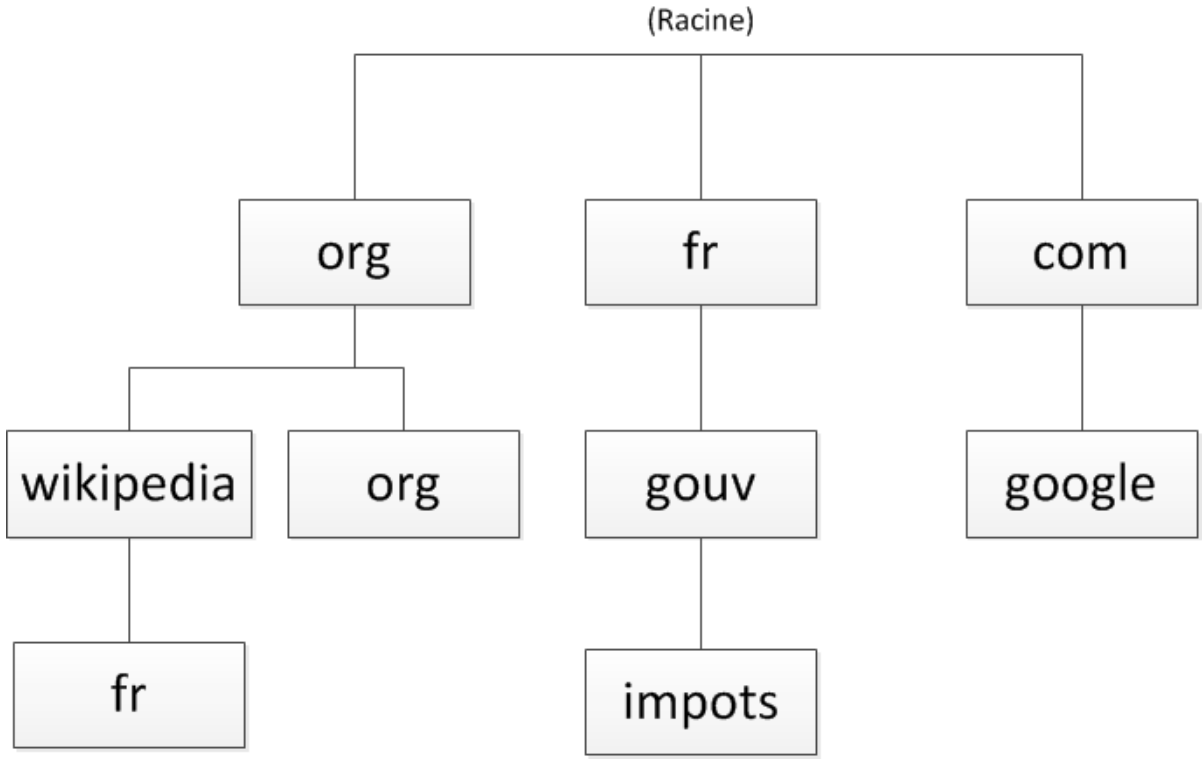
Conséquence : résoudre un nom de domaine ou sous-domaine nécessite plusieurs serveurs DNS.

FQDN

FQDN : fully qualified domain name = domaine absolu désignant un nœud jusqu'à la racine

Par convention, se termine par un point.

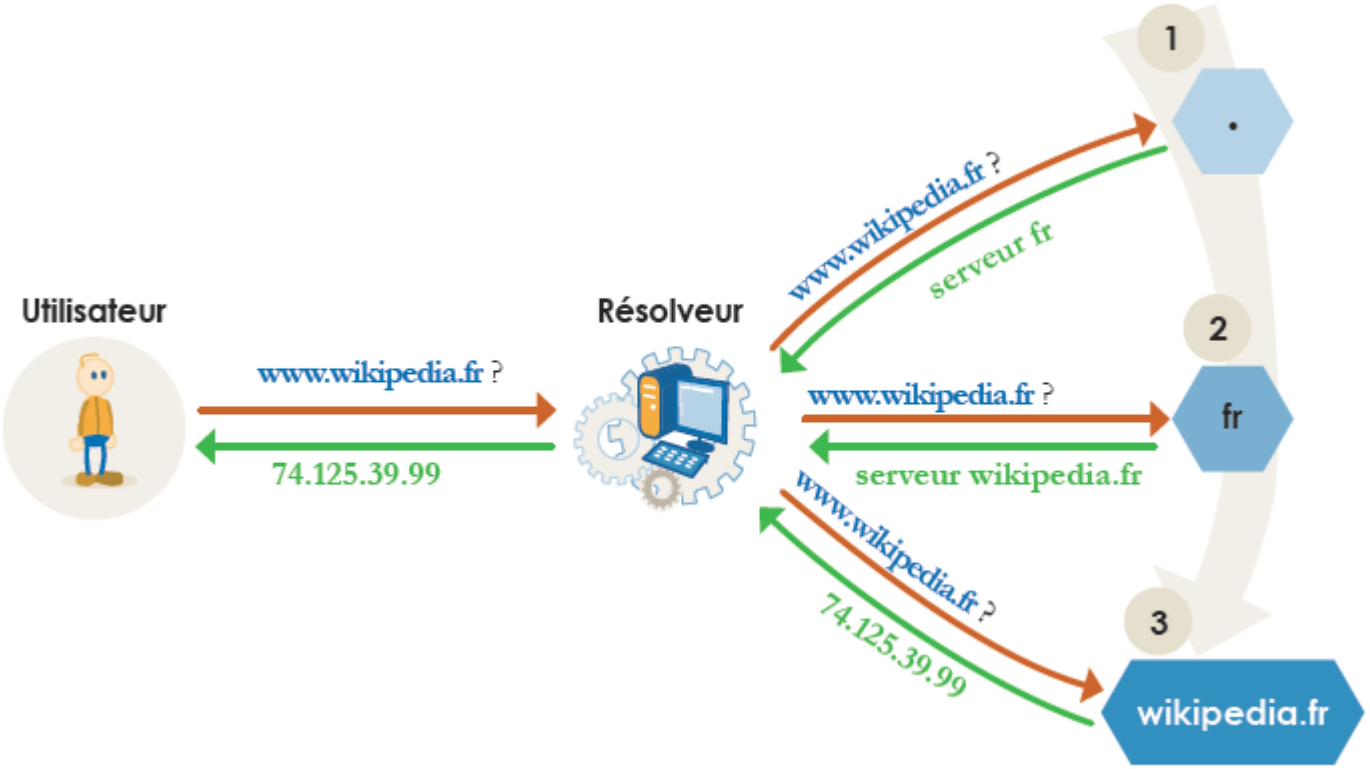
Exemple :
impots.gouv.fr.
fr.wikipedia.org.



La résolution d'un FQDN I

La résolution utilise la hiérarchie des serveurs DNS

Un exemple simplifié :



AFNIC et repris par Dominique Revuz (UPEM)

La résolution d'un FQDN II

Un exemple plus complexe :

La machine du réseau local de l'entreprise toto souhaite joindre : www.microsoft.com.

Les différents serveurs DNS

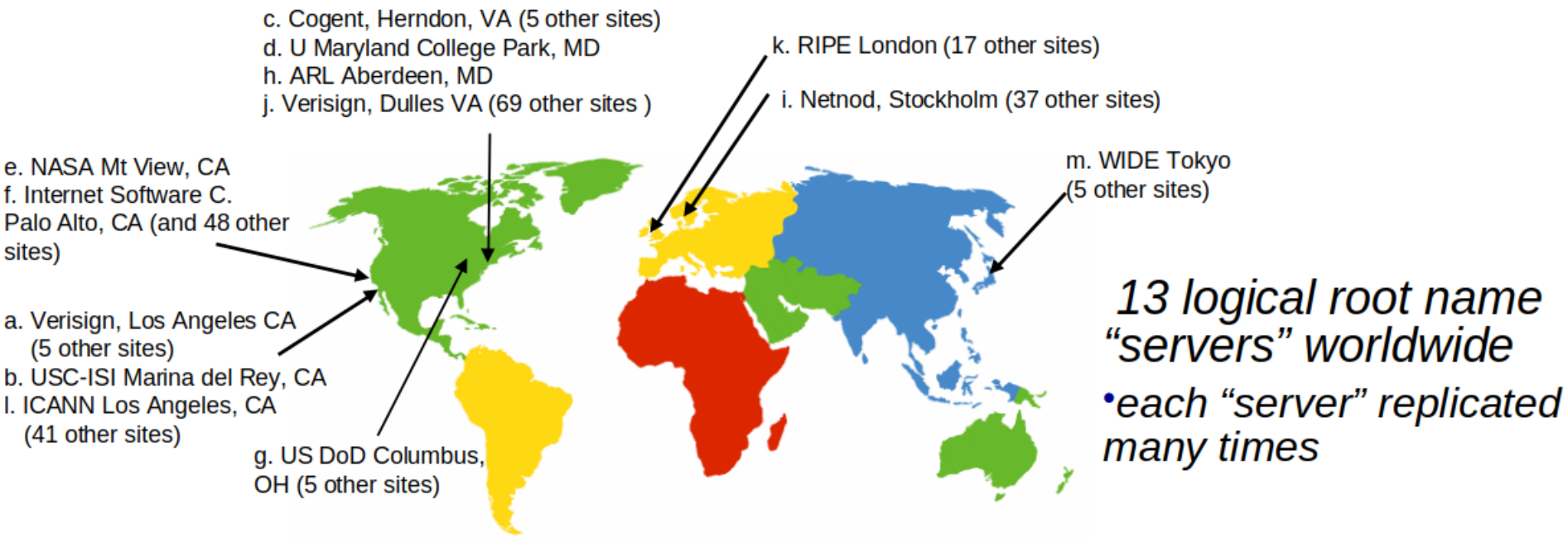
- S1 est le serveur DNS de l'entreprise toto. Il a autorité sur toto.fr.
- S2 est le serveur DNS de l'entreprise microsoft. Il a autorité sur microsoft.com.
- S3 est le serveur DNS ayant autorité sur .fr.
- S4 est le serveur DNS ayant autorité sur .com.
- S5 est un serveur racine d'Internet qui a autorité sur .

La résolution d’un FQDN III

Schéma d'une résolution

- La machine voulant faire la résolution interroge S1.
- S1 ne connait pas l'adresse IP de www.microsoft.com. Il interroge S3, qui n'a pas la réponse mais qui fournit à S1 l'adresse IP de S5.
- S1 interroge S5 qui n'a pas la réponse mais qui fournit à S1 l'adresse IP de S4.
- S1 interroge S4 qui n'a pas la réponse mais qui fournit à S1 l'adresse IP de S2.
- S1 interroge S2 qui a autorité sur microsoft.com et qui connait donc l'adresse IP de la machine www.microsoft.com. S2 envoie à S1 cette adresse IP.
- S1 fournit la réponse à la machine de départ.

Root name servers



DNS root servers

Avec un routage des requêtes en « anycast », i.e. chaque requête est routée vers l'une des répliques de ces 13 serveurs.

DNS local et cache + Démo !

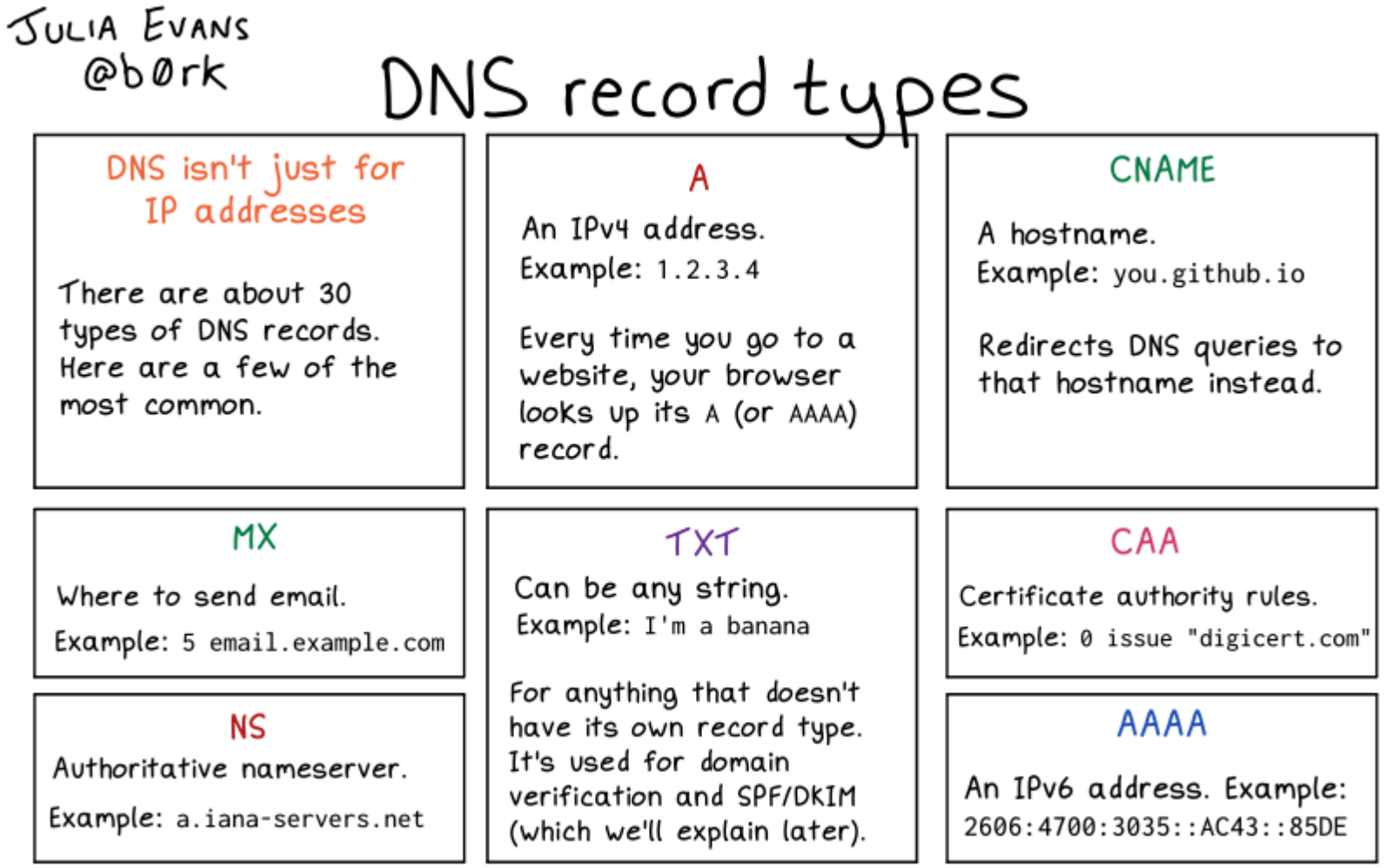
En pratique, de nombreux domaines sont en cache dans les serveurs DNS, en particulier dans un serveur DNS local :

- Le serveur DNS de votre fournisseur d'accès
- Le serveur DNS de votre école

Exemple :

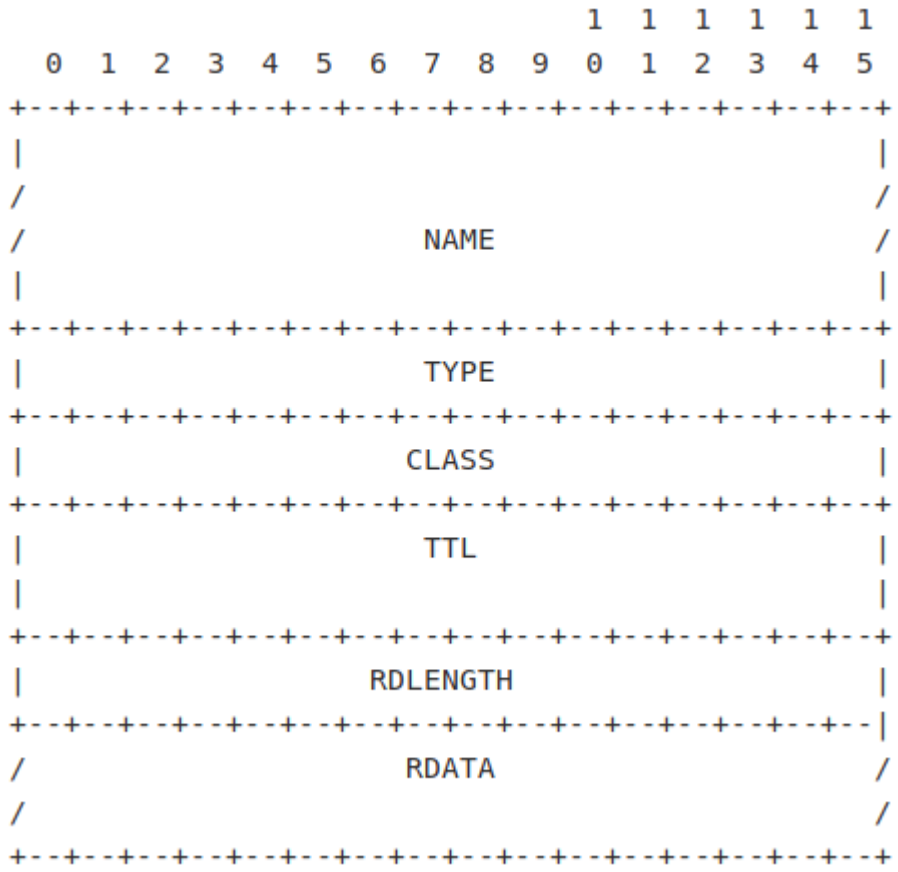
```
dig ensta-bretagne.fr

dig ensta-bretagne.fr # Attention au TTL


dig MX ensta-bretagne.fr # Pour les mails

# autres commandes plus tard :)
```

Les types d'enregistrements DNS



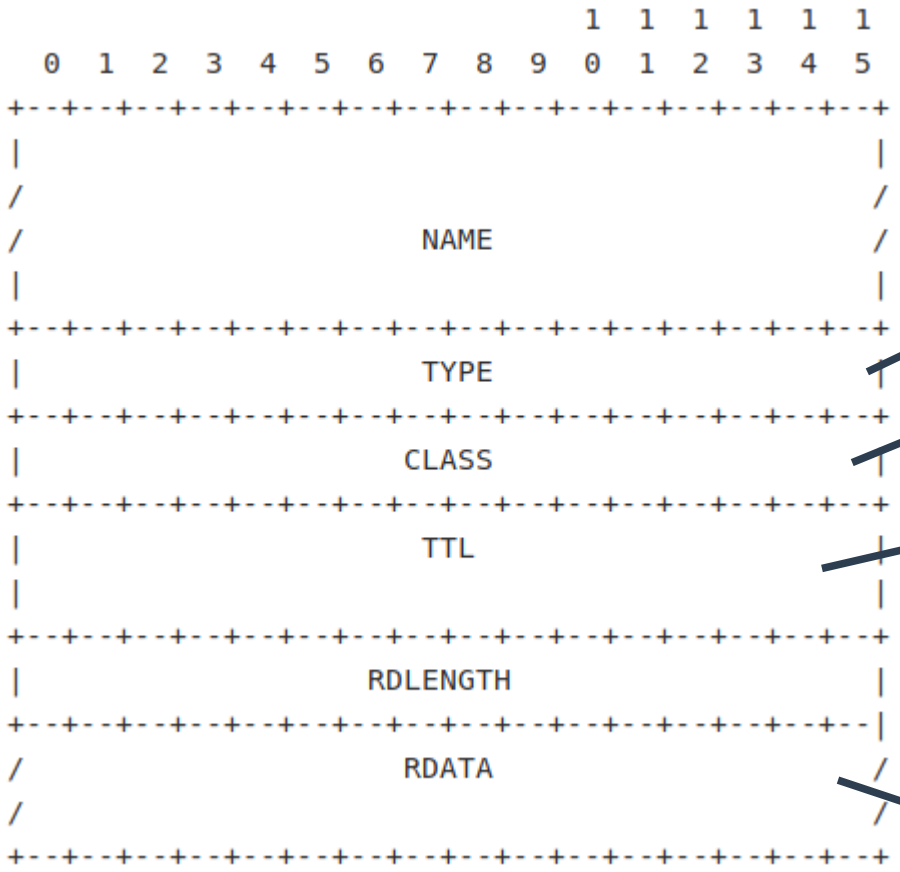
Les types d'enregistrements DNS

La RFC 1035 définit de manière générale un enregistrement DNS :



Les types d'enregistrements DNS

La RFC 1035 définit de manière générale un enregistrement DNS :



Types

Classes (peu utilisé et ésotérique)
En général : IN Internet
CH Chaos, HS Hesiod

Durée de validité de l'enregistrement pendant lequel il est peut être mis en cache.

Données supplémentaires suivant le type

Le type TXT illustré avec SPF

Type TXT : permettre d'associer du texte arbitraire avec un FQDN

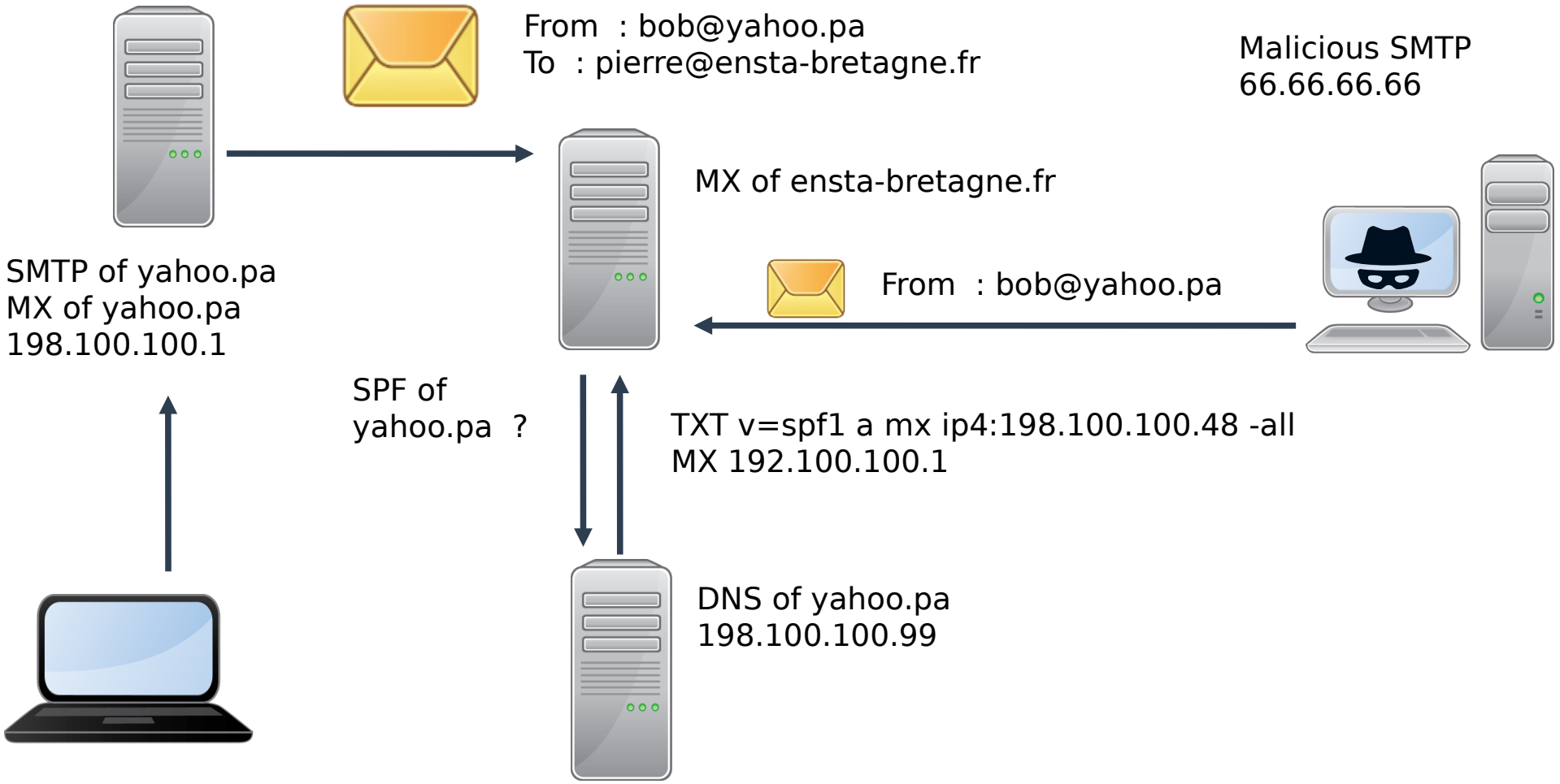
```
example.com    IN    TXT    "Reserved for use in documentation"
```

On peut donc l'utiliser à des tas d'usages particuliers.

Exemple :
SPF : déclarer les serveurs autorisés à transmettre un email avec un From : correspondant au nom de domaine.

```
v=spf1 a mx ip4:198.100.100.48 -all
```

SPF : se protéger du SPAM



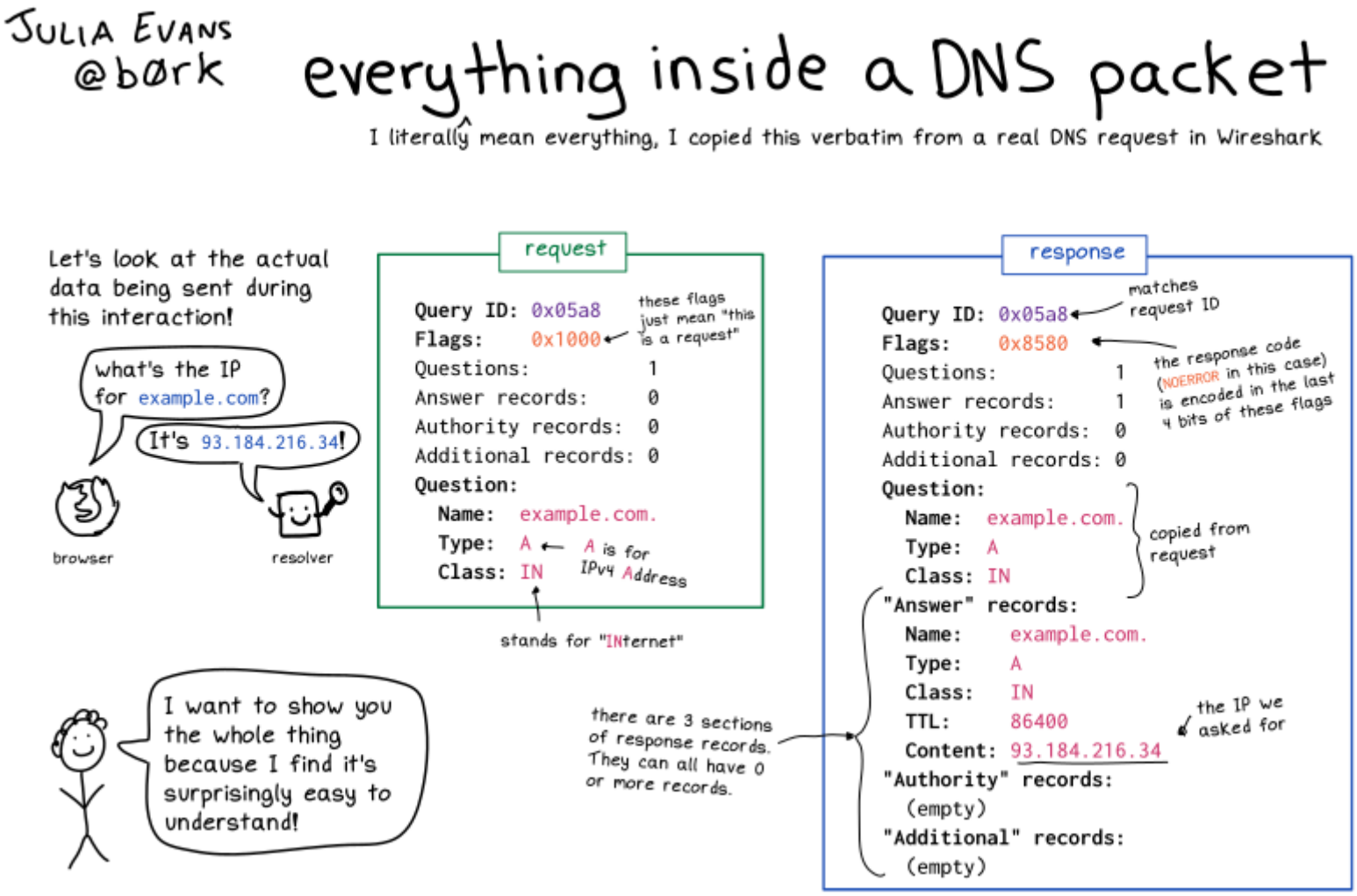
La résolution inverse

On peut aussi tenter de réaliser la résolution inverse d'une adresse IP, pour retrouver le FQDN associé :

```
dig -x 8.8.8.8
```

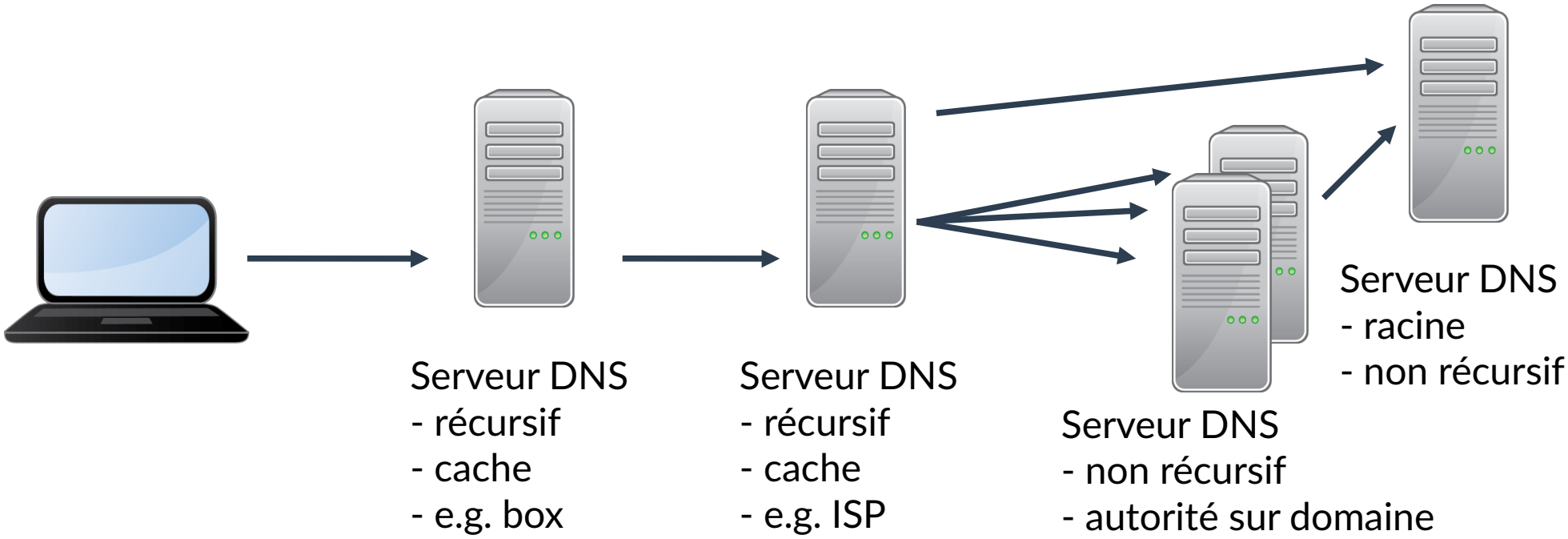
Démo

Protocole DNS : résumé de (l'excellente) Julia Evans

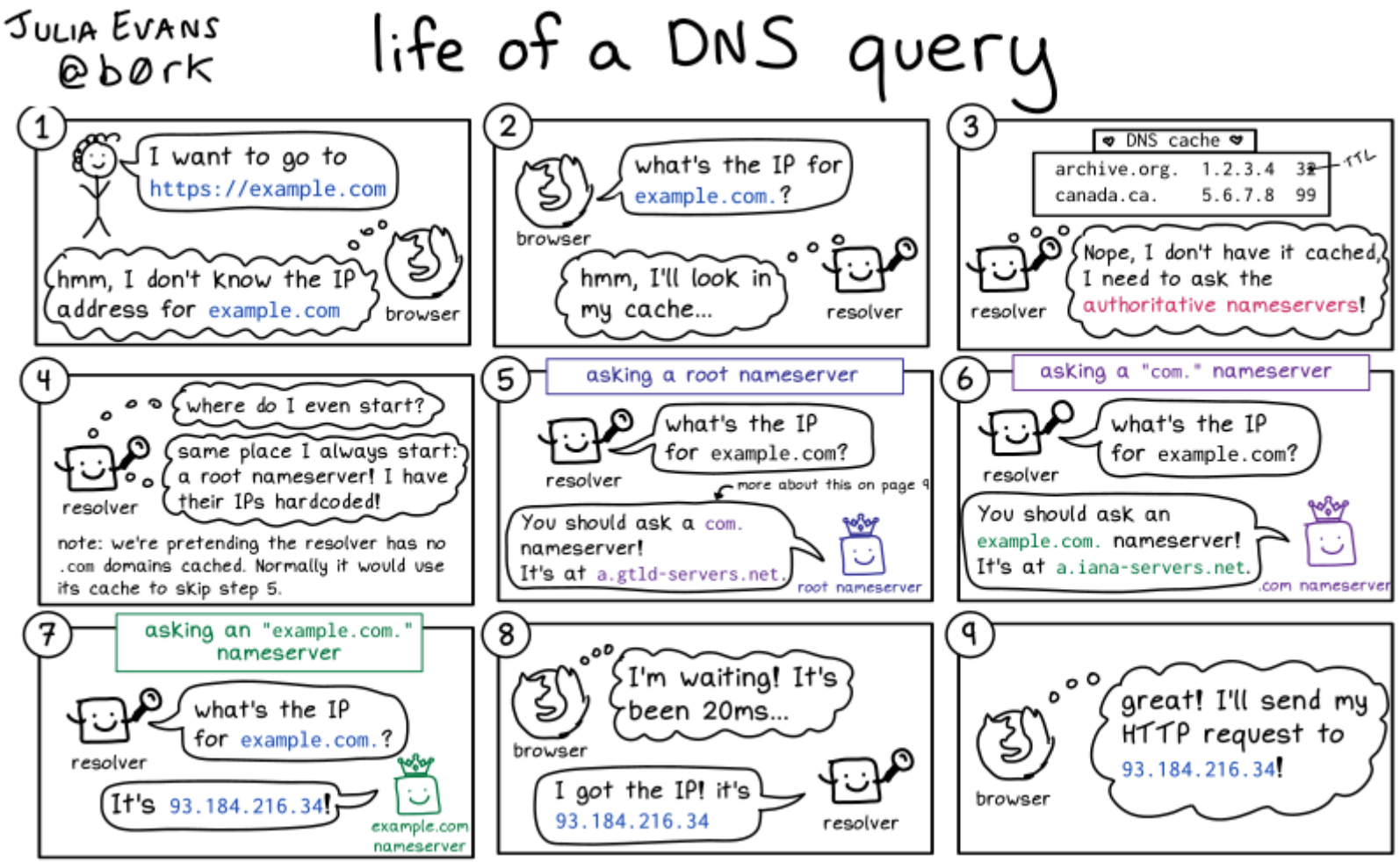


Architecture des serveurs DNS

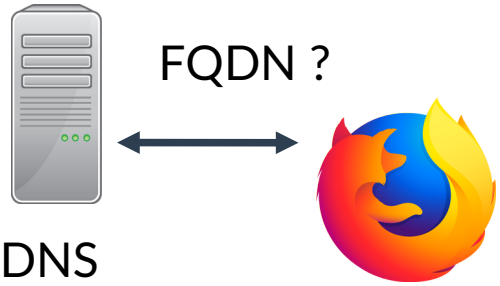
Le système des *Autoritative DNS server* est suffisant pour faire marcher la résolution de noms de domaines : si la récursion est présente on peut toujours remonter à la racine. Mais l'utilisation de cache et de TTL améliore les performances.



En résumé + exercice !



Def incrémentale : c'est quoi le web ?



HTTP

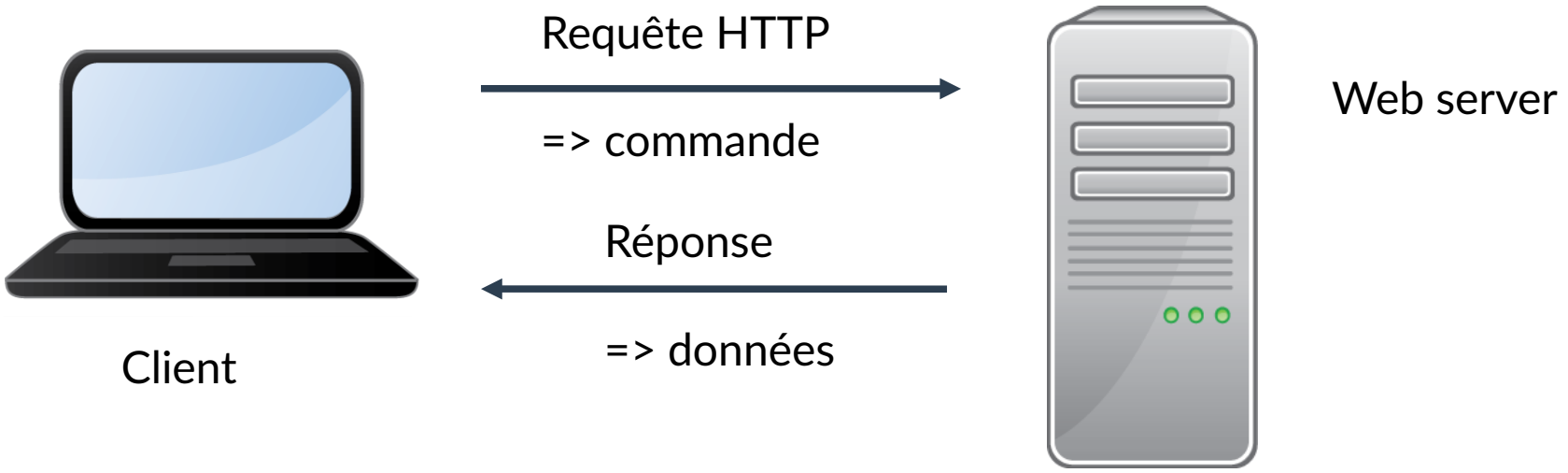
Le web et HTTP

HTTP pour HyperText Transfert Protocol est inventé en 1990, en même temps que les noms de domaine, le langage HTML. Il permet, historiquement, de faire communiquer un navigateur web avec un web serveur.

Le protocole est numéroté et défini par plusieurs RFC :

- HTTP/0.9 (invention par Tim Berners-Lee)
- HTTP/1.0 (96): RFC 1945
- HTTP/1.1 (97) : RFC 2068 2616
- HTTP/2.0 (12) : en travaux…
- HTTP/1.1 (14) : RFC 7230 à 7237

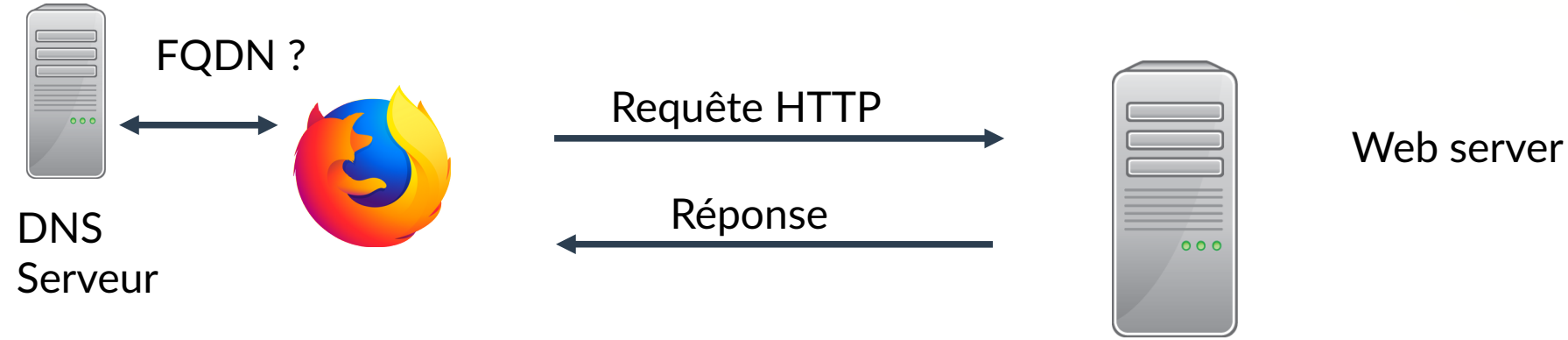
Caractéristiques



Le protocole utilise:

- TCP pour le transport
- Le port 80 du serveur (443 pour HTTPS)
- La notion de commande pour décrire une action
- L'indépendance des requêtes (stateless)
- Un format « textuel » (full text)

Def incrémentale : c'est quoi le web ?



HTTP : format des messages

Valable pour une requête ou une réponse :

```
HTTP-message   = start-line
                 *( header-field CRLF )
                 CRLF
                 [ message-body ]
```

Une première ligne obligatoire (commande ou réponse)

- Un certain nombre d'entêtes
- Une ligne vide
- Un corps de message (payload)

Les commandes

Très utilisées :

- GET : demande d'une ressource cible au serveur
- HEAD : idem mais sans la payload
- POST : envoie des données au serveur (payload)

Déjà beaucoup moins :

- PUT : mettre à jour la cible avec les données
- DELETE : détruire la cible

Jamais entendu parler... :

- CONNECT, OPTIONS, TRACE, PATCH

Le code de statut de la réponse

HTTP status codes

Every HTTP response has a ★status code★.

There are 50ish status codes but these are the most common ones in real life:

200 OK

} 2xxs mean ★ success ★

301 Moved Permanently
302 Found
temporary redirect
304 Not Modified
the client already has the latest version, "redirect" to that

} 3xxs aren't errors, just redirects to somewhere else

400 Bad Request
403 Forbidden
API key/OAuth/something needed
404 Not Found
we all know this one :)
429 Too Many Requests
you're being rate limited

} 4xx errors are generally the client's fault: it made some kind of invalid request

500 Internal Server Error
the server code has an error
503 Service Unavailable
could mean nginx (or whatever proxy) couldn't connect to the server
504 Gateway Timeout
the server was too slow to respond

} 5xx errors generally mean something's wrong with the server.

14

Julia Evans
@b0rk

Les codes de réponse importants

A part 200, 404, 500…

301 : déplacé de manière permanente

308 : redirection permanente : le navigateur suit la nouvelle URL

Les codes de réponse importants

A part 200, 404, 500…

301 : déplacé de manière permanente

308 : redirection permanente : le navigateur suit la nouvelle URL

Une erreur particulière…

Les codes de réponse importants

A part 200, 404, 500…

301 : déplacé de manière permanente

308 : redirection permanente : le navigateur suit la nouvelle URL

418 : I’m a teapot (je suis une théière)

- le serveur refuse de préparer un café

- RFC 2324, section 2.3.2 : HTCPCP/1.0

Hyper Text Coffee Pot Control Protocol :

Any attempt to brew coffee with a teapot should result in the error code "418 I'm a teapot". The resulting entity body MAY be short and stout.

Les codes de réponse importants

A part 200, 404, 500…

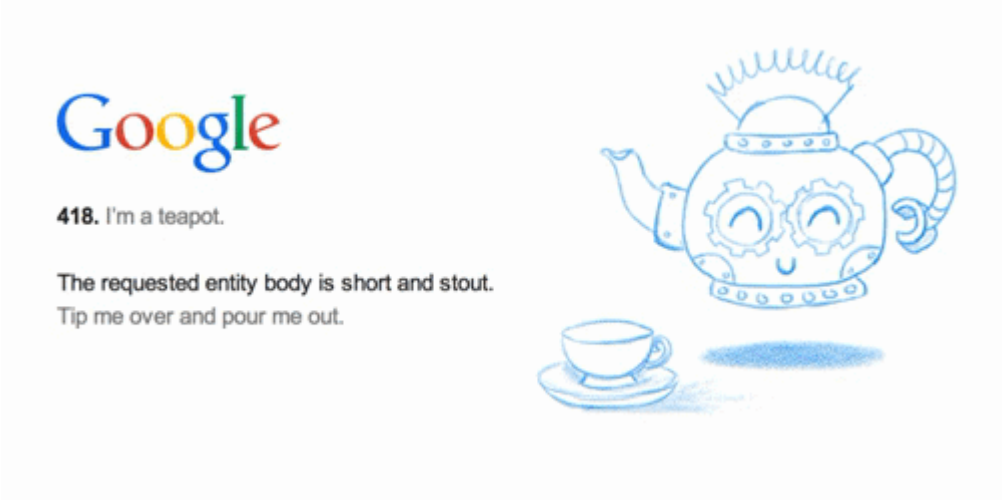
301 : déplacé de manière permanente

308 : redirection permanente : le navigateur suit la nouvelle URL

418 : I'm a teapot (je suis une théière)

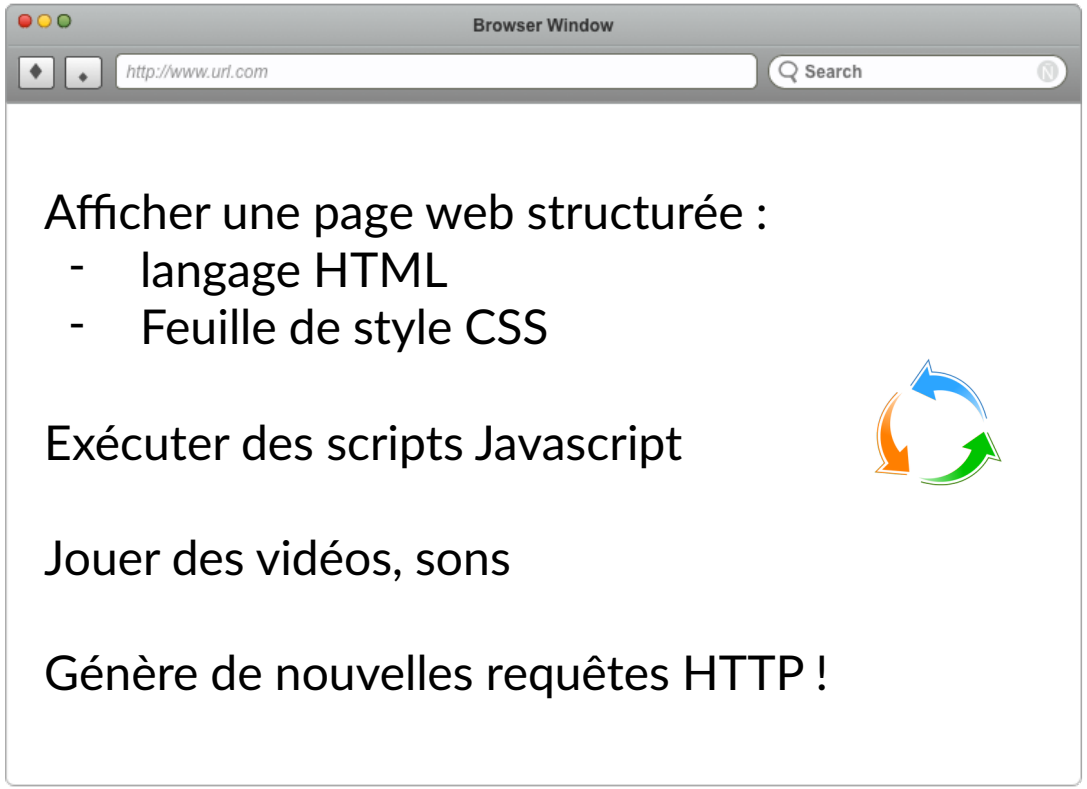
- le serveur refuse de préparer un café

- RFC 2324, section 2.3.2 : HTCPCP/1.0



1er avril 1998 à l'IETF, ça a été conservé depuis…

Rôle du navigateur

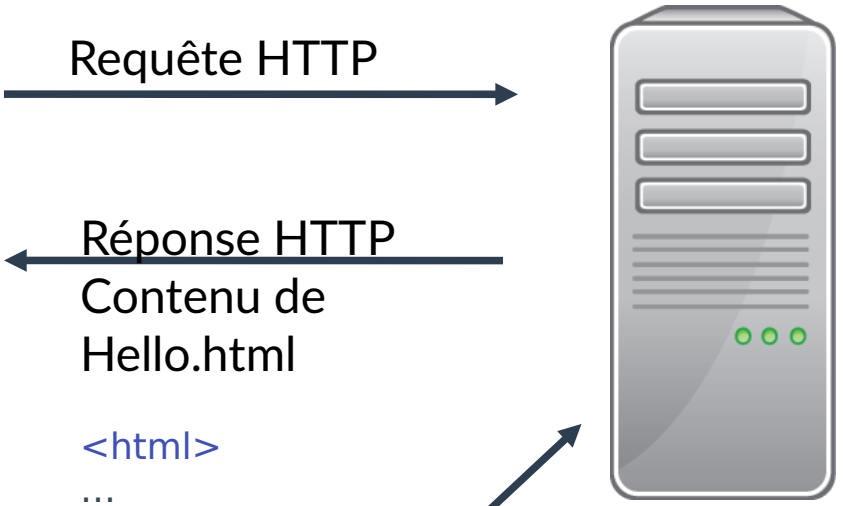


Afficher une page web structurée :
- langage HTML
- Feuille de style CSS

Exécuter des scripts Javascript

Jouer des vidéos, sons

Génère de nouvelles requêtes HTTP !



Formulaire HTML

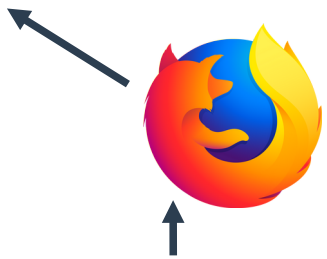
First name:

Mickey

Last name:

Mouse

Submit



POST /post HTTP/1.1
Host: postman-echo.com
Content-Length: 31
...

firstname=Mickey&lastname=Mouse

Requête HTTP POST

```
<form action="https://postman-echo.com/post" method="POST">
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>
```

PUT / DELETE

Créer / Mettre à jour / Détruire une ressource côté serveur

- Format identique à du POST
- Cible une ressource et la créé
- La cible est l'url de la ressource à créer (il faut l'avoir déterminer en avance)
- Le serveur crée la cible et renvoie 201 ou 204

En pratique, POST est souvent utilisé à mauvais escient, à la place de PUT.

```
> GET /clients/123
< 404
> PUT /clients/123
> toto
< 201
> GET /clients/123
< 200
< toto
> PUT /clients/123
> toto
< 204
> GET /clients/123
< 200
< toto
> PUT /clients/123
> poum
< 204
> GET /clients/123
< 200
```

Def incrémentale : c'est quoi le web ?

FQDN ?

DNS
Serveur

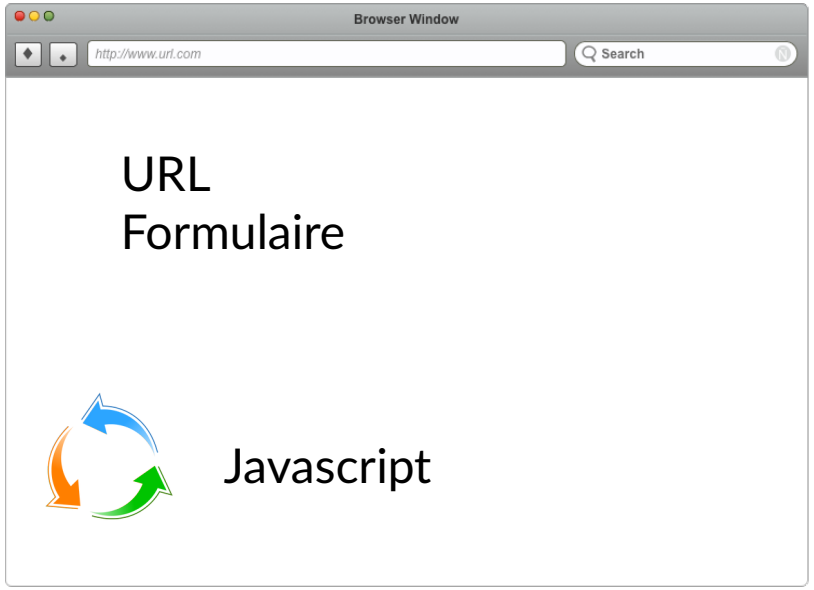
Requête HTTP

Réponse

Web server

GET
POST
PUT DELETE

traitement



URL
Formulaire

Javascript

Cookie

Initialisation de la session : le serveur ajoute le header :

Set-Cookie : name=value ; cookie-av

Avec cookie-av = "Comment" "=" value

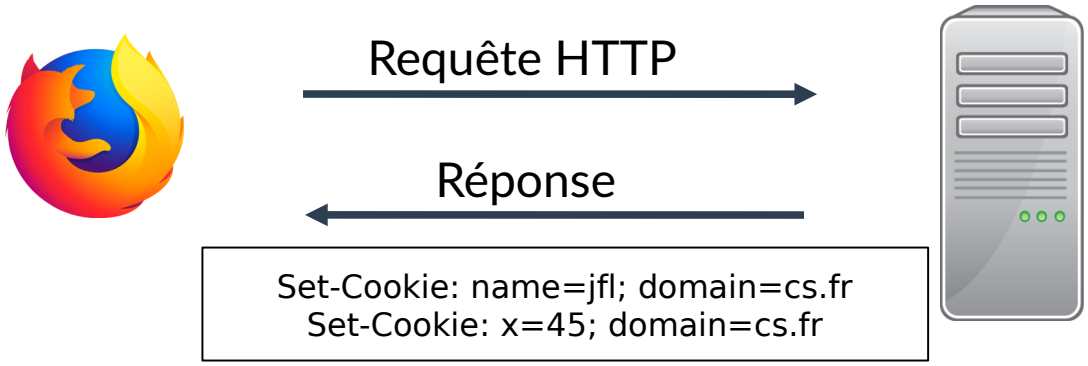
| "Domain" "=" value

| "Max-Age" "=" value

| "Path" "=" value

| "Secure"

| "Version" "=" 1*DIGIT



Domain : nom de domaine de validité, commençant par un point.

Max-Age (ou Expires avec une date) : durée de validité

Path : sous ensemble d'url ou le cookie est valide

Secure : indique au client d'utiliser un canal sécurisé… (cf. slide suivant)

Version : numéro de version correspond à la RFC.

Cookie et sécurité

Un cookie est utilisé :

- Pour maintenir l'authentification d'un utilisateur
- Pour « tracer » un utilisateur (publicité)
- Pour du paramétrage (langue, etc.)

Cookie et sécurité

Un cookie est utilisé :

- Pour maintenir l'authentification d'un utilisateur
- Pour « tracer » un utilisateur (publicité)
- Pour du paramétrage (langue, etc.)

Or, HTTP n'est pas chiffré mais il existe HTTPS pour sa version sécurisée.

Cookie et sécurité

Un cookie est utilisé :

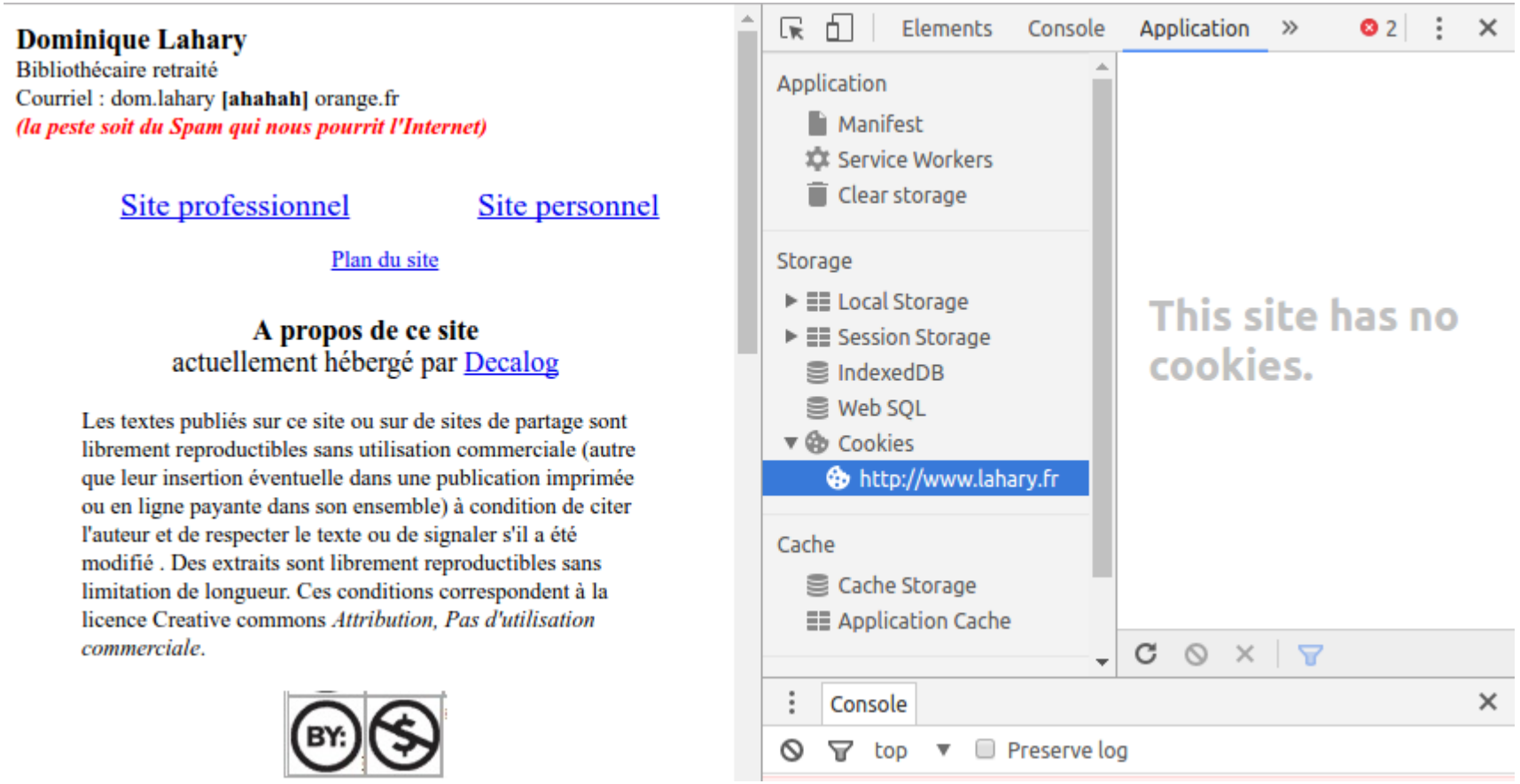
- Pour maintenir l'authentification d'un utilisateur
- Pour « tracer » un utilisateur (publicité)
- Pour du paramétrage (langue, etc.)

Or, HTTP n'est pas chiffré mais il existe HTTPS pour sa version sécurisée.

Pour un cookie d'authentification, l'attribut « Secure » d'un cookie demande au client de NE PAS renvoyer ce cookie en HTTP (ce qui serait grave d'un point de vue sécurité).

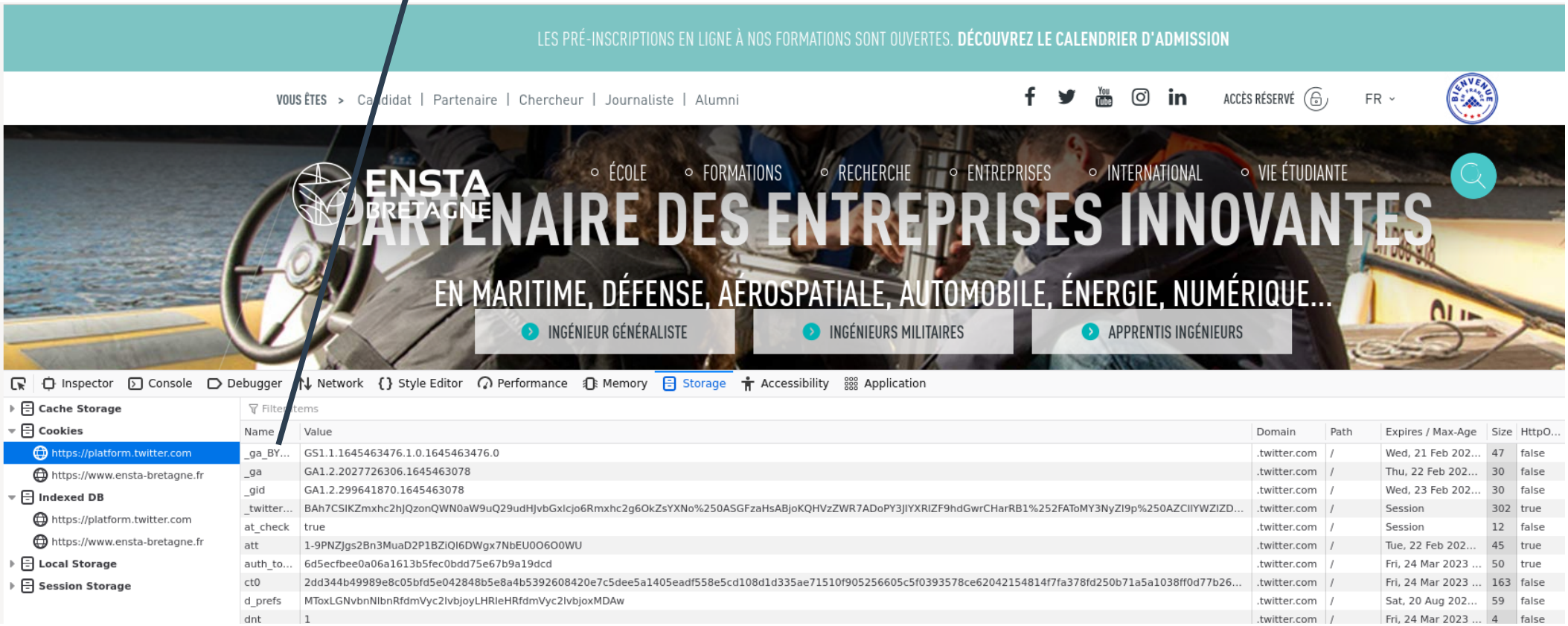
Pour un cookie autre, par exemple de paramétrage, ce ne serait pas grave de le faire.

Exemple : pas de cookie

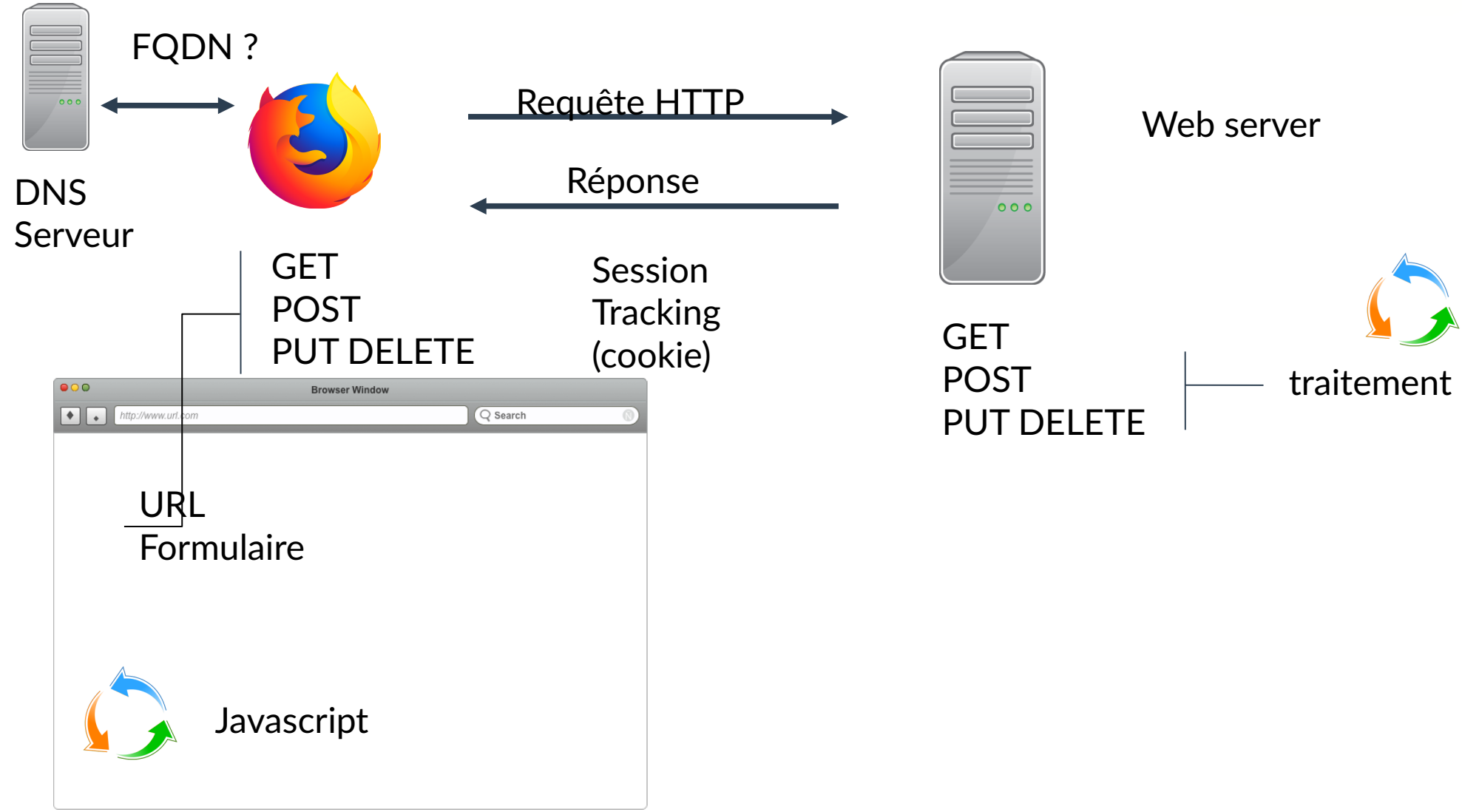


Exemple : miam miam cookies !

Cookie **_ga**
Google Analytics :)



Def incrémentale : c'est quoi le web ?



Les Emails

E-Mail

Message, au format défini par la RFC 2822.

- au format texte, avec jeu de caractères ASCII
- sinon au format MIME cf. RFC 2045 à 2049
- des lignes de textes délimitées par CRLF (13+10)
- des headers, un body

On dirait du HTTP...

From: JFL (bob@gmail.com)

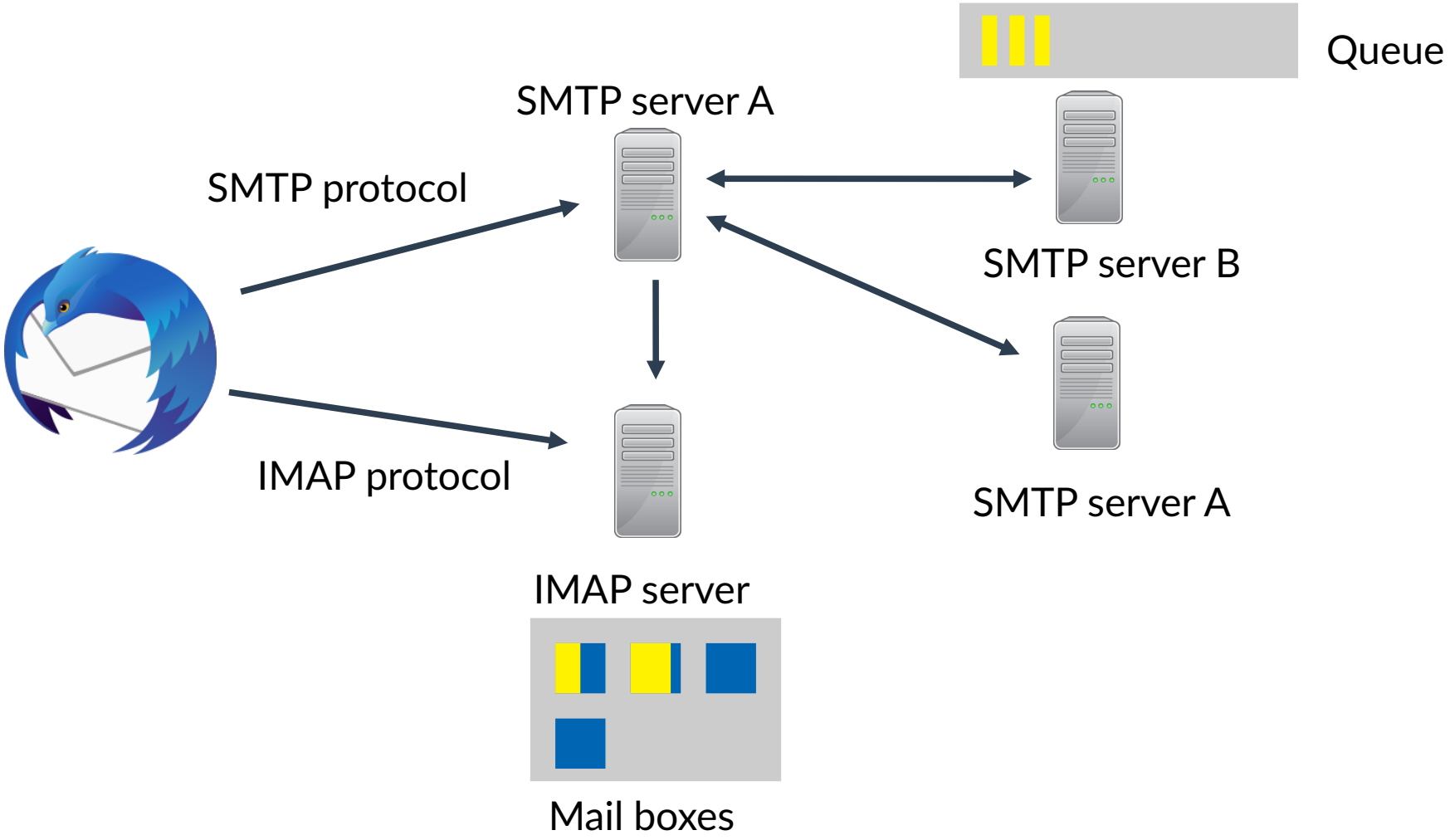
Subject: this course rocks !

Date: September 23, 2018 3:30:58 PM PDT

To: user@example.com

Bla bla...

Architecture



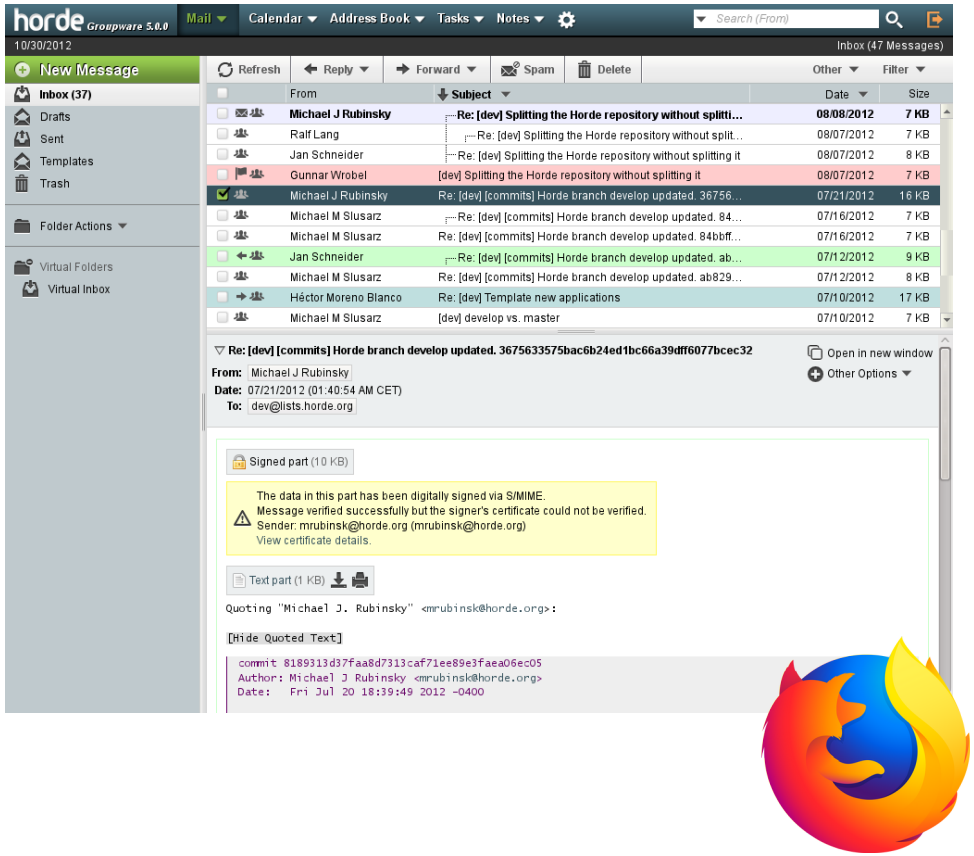
Ex. de dialogue avec un serveur SMTP

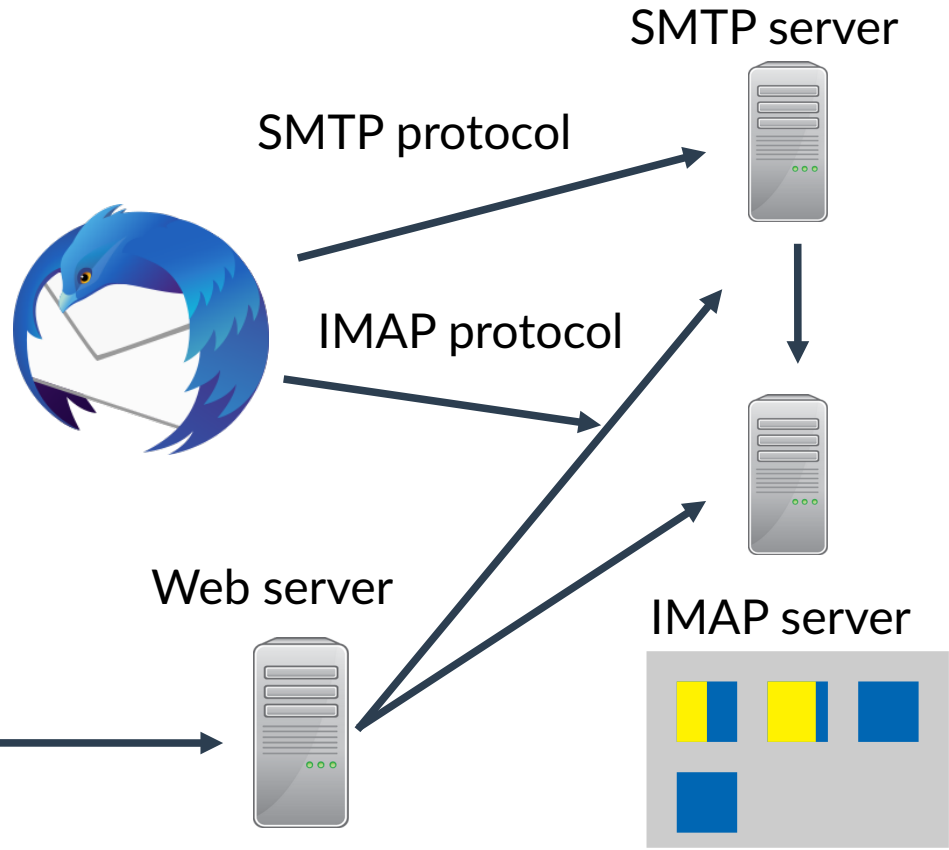
from *Computer Networking: A Top Down Approach, Jim Kurose, Keith Ross*

Simple Mail Transfer Protocol
Exemple:

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

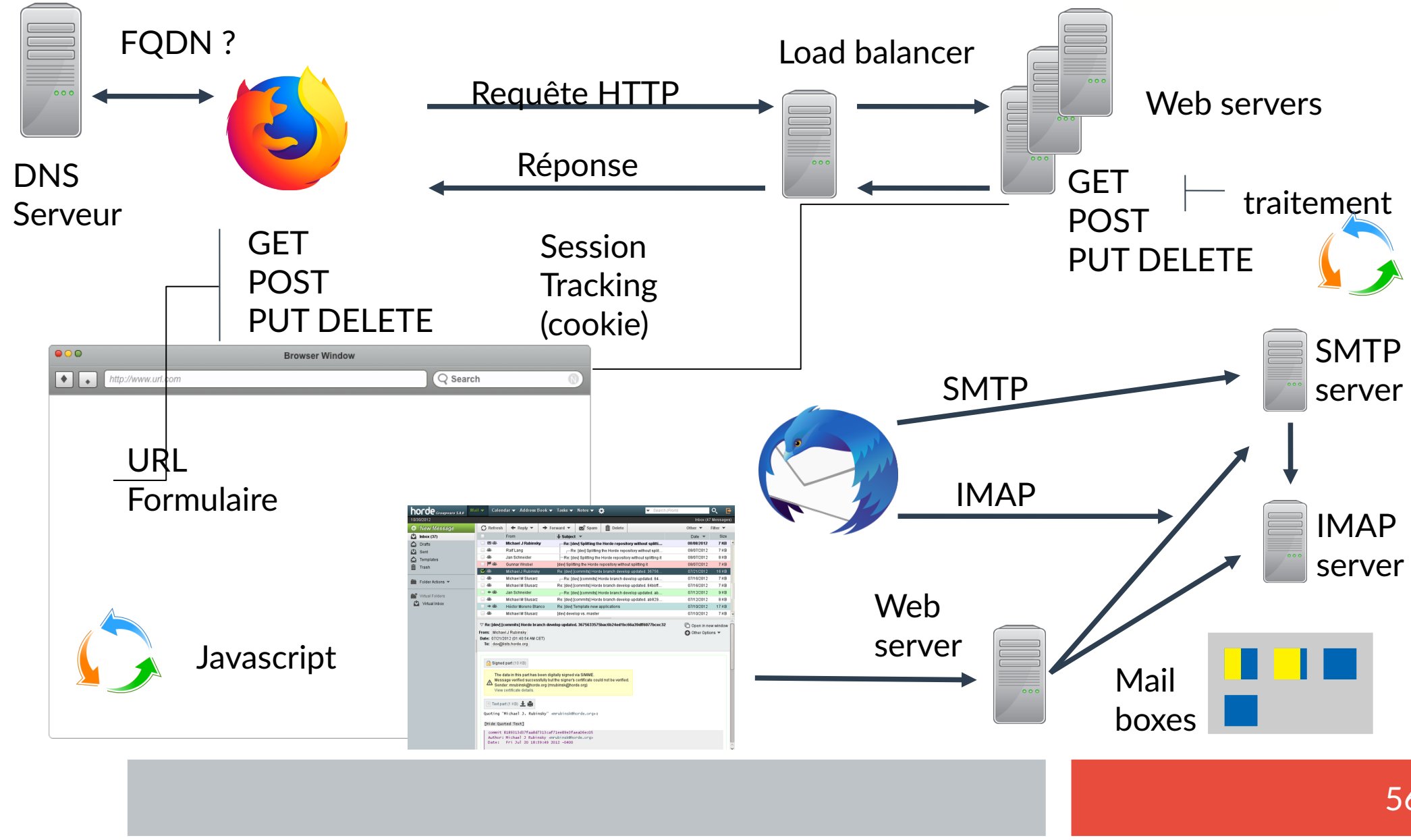
Webmail





Une application web qui accède aux serveurs IMAP ou SMTP. Certaines implémentations réimplémente la gestion des BAL afin d'accélérer le rendu de l'application web (e.g. Zimbra qui accède aux emails depuis une base Mysql).

Def incrémentale : c'est quoi le web ?



Références

Top level domains :

https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

Réseaux TCP/IP/Les serveurs DNS :

https://fr.wikibooks.org/wiki/R%C3%A9seaux_TCP/IP/Les_serveurs_DNS

Computer Networking: A Top-Down Approach 7th ed., J.F. Kurose and K.W. Ross

http://www-net.cs.umass.edu/kurose-ross-ppt-7e/

DNS classes :

https://miek.nl/2009/july/31/dns-classes/

POST vs. PUT : la confusion, Joachim Rousseau :

https://blog.xebia.fr/2014/03/17/post-vs-put-la-confusion/

Telnet IMAP Commands Note :

https://busylog.net/telnet-imap-commands-note/

Couches applicatives
14/10, suite