Sys 1 - Amphi 8

Couche applicative
Exemple de protocoles en TCP (HTTP) et UDP (DNS)

Guillaume DIDIER - 07/11/2023

Bonnes vacances ?

Where are we ?

Planning du cours

Date	Amphi	Pratique (TD/TP/Projet)	Références
17/10/2023	Introduction, Socket TCP	Projet (binôme): Crawler HTTP & Dice Roll	Kurose 1.1, 1.5, 1.7, 2.7.2 CS:APP 11.1, 11.2, 11.4
07/11/2023	Pas de CM (Parrainage)	Projet (binôme): Crawler HTTP & Dice Roll	
14/11/2023	**Protocoles Applicatifs over TCP, UDP, e.g HTTP, DNS**	**Projet (binôme): Crawler HTTP & Dice Roll**	**Kurose 2.1, 2.2, 2.4, 2.7 CS:APP 11.3, 11.5, 11.6,**
17/11/2023	Protocoles de Transport	Pas de TP (rattrapage CM)	Kurose Ch 3, CS:APP 11.3
21/11/2023	TCP suite et fin, protocole IP, notion de Lien	TP (seul): Reliable Data Transport	Kurose Ch 3, 4.1, 4.3, 6.1
28/11/2023	Forwarding + Routage	TP (seul): Reliable Data Transport	Kurose 4.1, 4.3, 5.1, 5.2
05/12/2023	Routage (un peu plus de lien)	TD : IP + Routage	Kurose 5.1, 5.2, 5.4 (6)
12/12/2023	Conclusion : Les défis d'internet	TD : Révisions	Divers sections du Kurose

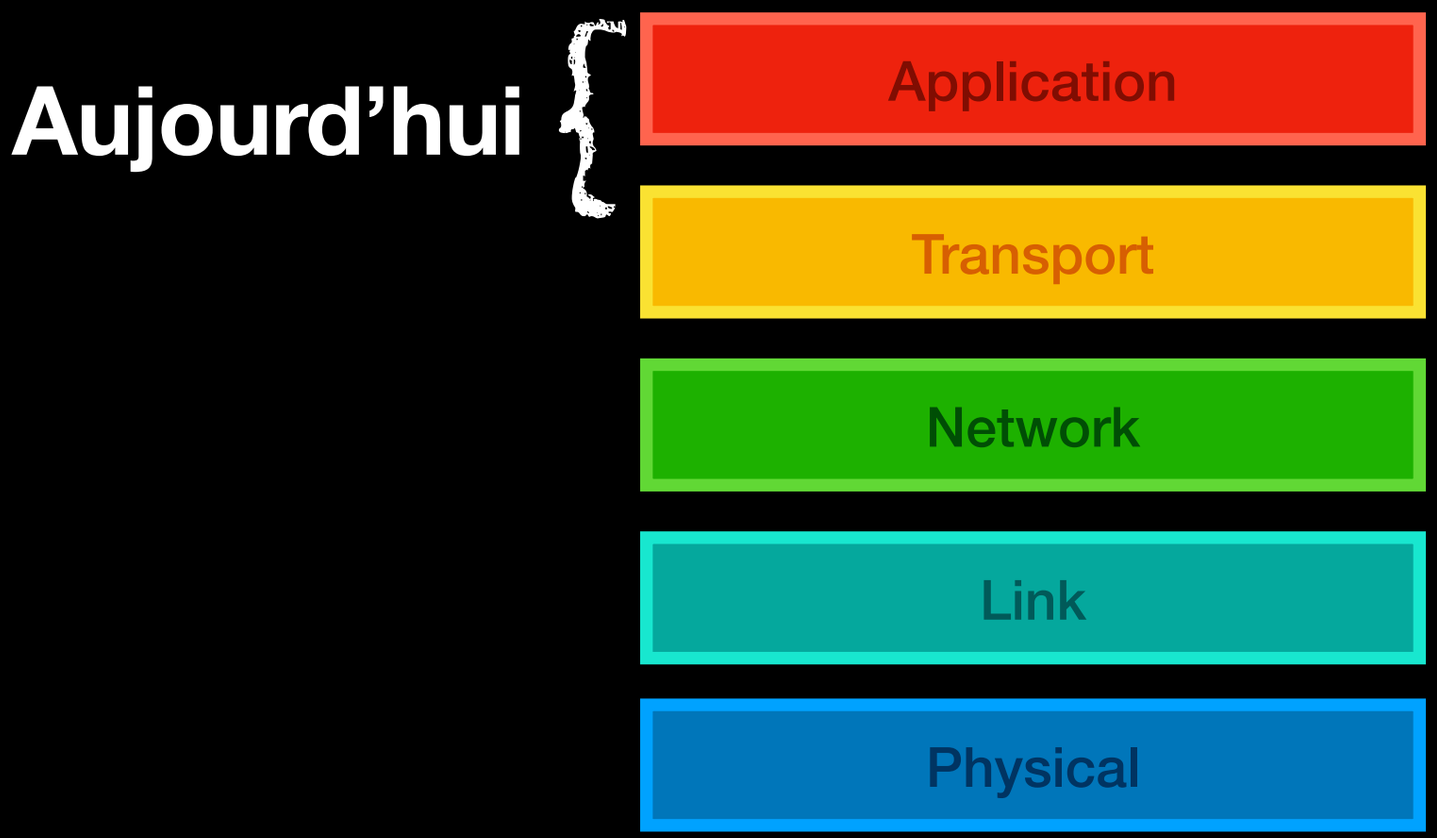
Programme du jour

Protocoles applicatifs

TCP vs UDP, les 2 principaux transports sous-jacents
HTTP (over TCP)
DNS (over UDP)

Rappel: Les couches réseau

Ou sommes nous aujourd’hui ?



Rappel: Sémantique TCP

Reliable transport - No latency / throughput guaranty

Garanties :

- Pas de perte (retransmission)
- Arrivée ordonnée
- Détection d’erreur (somme de contrôle)

Coûts associés

Best effort :

- Latence
- Bande-passante

One level down

Que fourni la couche IP ?

Transmission d'un paquet à travers le réseau, en best effort.

Taille maximale (End-to-end) Perte OK, sans garantie de latence ou bande-passante

Ouverture connection TCP :
Échange de 3 paquets

Why UDP ?

Pourquoi TCP n’est pas toujours adapté

Vidéo conférence / audio / streaming

Préférer image en faible résolution, que flux haché

Échange type Question Réponse courte

Coût d’ouverture (3 paquets) trop lourd

Temps réel, gestion spécifique de la BP - latence etc

Laisser l’application gérer

Flux multiples - TCP head of line blocking

Ne pas bloquer tous les flux pendant la retransmission

Le transport UDP

Le strict minimum

UDP fournit :

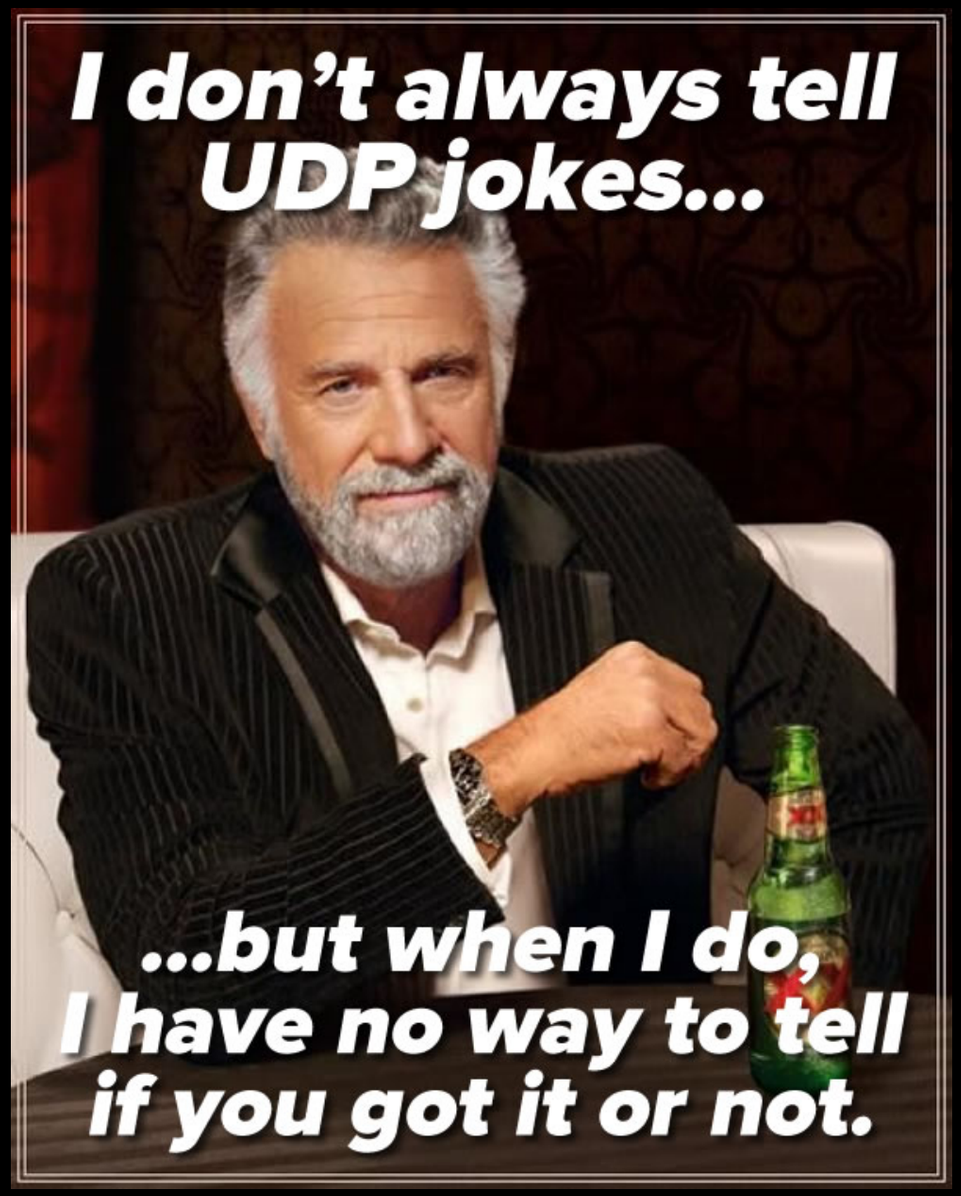
Somme de contrôle

Démultiplexage par port.

That's all !

UDP vs TCP

Un peu d’humour



"Hi, I'd like to hear a TCP joke."
"Hello, would you like to hear a TCP joke?"
"Yes, I'd like to hear a TCP joke."
"OK, I'll tell you a TCP joke."
"Ok, I will hear a TCP joke."
"Are you ready to hear a TCP joke?"
"Yes, I am ready to hear a TCP joke."
"Ok, I am about to send the TCP joke. It will last 10 seconds, it has two characters, it does not have a setting, it ends with a punchline."
"Ok, I am ready to get your TCP joke that will last 10 seconds, has two characters, does not have an explicit setting, and ends with a punchline."
"I'm sorry, your connection has timed out. ... Hello, would you like to hear a TCP joke?"

UDP vs TCP en pratique

Comment utiliser UDP avec des sockets

Pas de mode facile

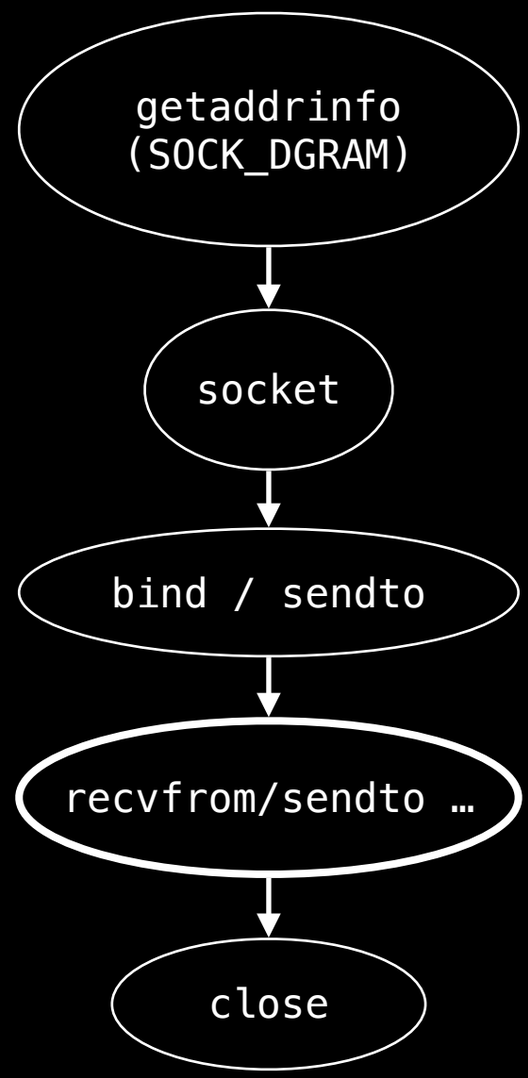
Principales modifications :

SOCK_STREAM -> SOCK_DGRAM dans getaddrinfo

```
read -> ssize_t recvfrom(int socket, void *buffer,
                         size_t length, int flags,
                         struct sockaddr *address,
                         socklen_t *address_len);

write -> ssize_t sendto(int socket, const void *buffer,
                        size_t length, int flags,
                        const struct sockaddr *dest_addr,
                        socklen_t dest_len;
```

Peu de distinction client / server,
à la place de listen/accept ou connect:
Utiliser bind (ou coté client sendto)



man …

HTTP

Le protocol derrière le (World Wide) Web

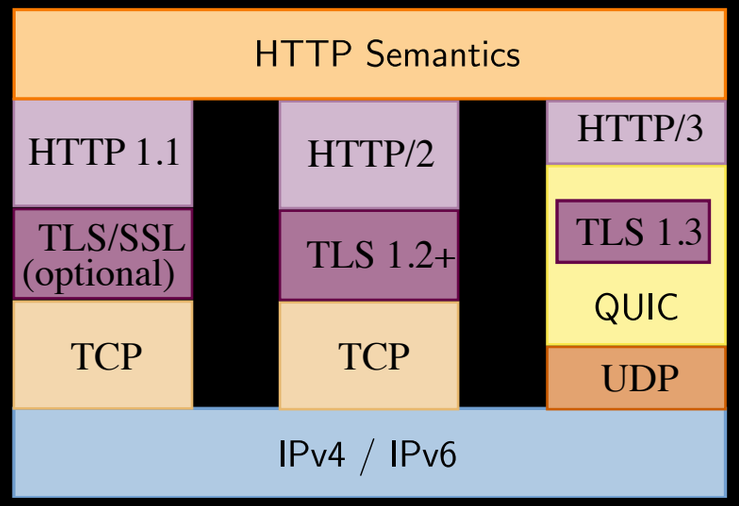
```
GET / HTTP/1.0\r\n
Host: ens-rennes.fr\r\n
\r\n
```


```
HTTP/1.1 200 OK\r\n
…\r\n
\r\n
```

```
<!DOCTYPE html>
<!--[if lte IE 7]> <html class="ie7 oldie no-js" xmlns="http://www.w3.org/
1999/xhtml" lang="fr" xml:lang="fr"> <![endif]-->
<!--[if IE 8]> <html class="ie8 oldie no-js" xmlns="http://www.w3.org/1999/
xhtml" lang="fr" xml:lang="fr"> <![endif]-->
<!--[if gt IE 8]><!-->
<html class="no-js" xmlns="http://www.w3.org/1999/xhtml" lang="fr"
      xml:lang="fr"> <!--<![endif]-->
<head>
…
```

Repère historiques

Ça date de quand tout ça



HTTP/0.9		1990	Première version de Tim Berner-Lee
HTTP/1.0	RFC 1945	Mai 1996	Premier standard
HTTP/1.1	RFC 2068 + RFC 2616	Janvier 1997 + Juin 1999	Keep-alive connection, Host header
HTTP/1.1 (clarifié)	RFC 7230 à RFC 7237	Février 2014	Résoud des imprécisions et ambiguïtés
HTTP/2.0	RFC 7540	Mai 2015	(Evolved from Google's SPDY) Multiplex request on 1 TCP connection + header compression
HTTP/3.0	RFC 9114	Juin 2022	HTTP over QUIC (over UDP), plutôt que TCP. Quic embarque la cryptographie TLS. (3x faster than HTTP/1.1)

Les RFCs de HTTP/1.1 et 2.0 ont été révisée en 2022 (RFC 9110-9113)

Anatomie des URL / URI (RFC 3986)

Comment localiser une ressource sur internet

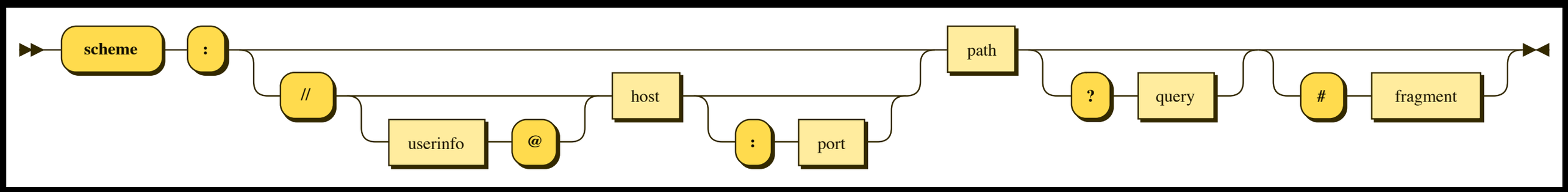
`https://mquinson.frama.io/ensr-arcsys1/#anciens-examens`

`ssh://git@github.com:22/llvm/llvm-project.git`

`https://www.ens-rennes.fr/search?beanKey=&l=0&RH=KERLANN_FR&q=L3+SIF`

```
URI = scheme ":" ["//" authority] path ["?" query] ["#" fragment]

authority = [userinfo "@"] host [":" port]
```



Anatomie d'une requête simple

HTTP / 1.1

```
GET /media/SYS1/R1-1080p.m4v HTTP/1.1\r\n
Host: media.guillaumedidier.fr\r\n
Connection: Close\r\n
\r\n
```

CRLF line endings

METHOD: GET, POST, ... (see RFC)

```
METHOD /path HTTP/1.1\r\n
```

Path must be at least /

```
Header: Value\r\n
...
Header: Value\r\n
```

Host header mandatory,
"Connection: Close" may be useful

```
\r\n
```

Empty line to signal end of request

```
<Data if applicable>
```

e.g. for PUT / POST

Anatomie d’une réponse

HTTP / 1.1

```
HTTP/1.1 301 Moved Permanently\r\n
Content-Type: text/html\r\n
Content-Length: 169\r\n
Connection: close\r\n
Location: https://media.guillaumedidier.fr/SYS1/R1-1080p.m4v\r\n
\r\n
<html data …>
```

```
HTTP/1.1 200 OK\r\n
Server: nginx/1.24.0\r\n
Date: Tue, 31 Oct 2023 12:47:37 GMT\r\n
Content-Type: video/mp4\r\n
Content-Length: 87550031\r\n
Last-Modified: Sun, 15 Oct 2023 15:35:48 GMT\r\n
Connection: keep-alive\r\n
Keep-Alive: timeout=20\r\n
ETag: "652c06d4-537e84f"\r\n
Accept-Ranges: bytes\r\n
\r\n
<mp4 data …>
```

```
HTTP/1.1 STATUS_CODE Message\r\n
Header: Value\r\n
...
Header: Value\r\n
\r\n
<Data if applicable>
```

Des requêtes plus complexes

HTTP / 1.1

- Méthodes GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE (RFC 9110)
  Utilisé pour les API web

- Garder la connection ouverte pour enchainer plusieurs requêtes.
  (HTTPS : Réutilisation de la session TLS et connection TCP)

- Beaucoup d’en-têtes possibles.

Pourquoi HTTP 2 et 3 ?

Le web s’est alourdi

- Head of Line Blocking
- Paralleliser les requêtes, mais utiliser une seule connection TCP (HTTP/2)
- Mieux gérer la retransmission: Utiliser QUIC/UDP (HTTP/3)
- Réduire le coup d’ouvrir / ré-ouvrir une connection
- Nettoyer les standards

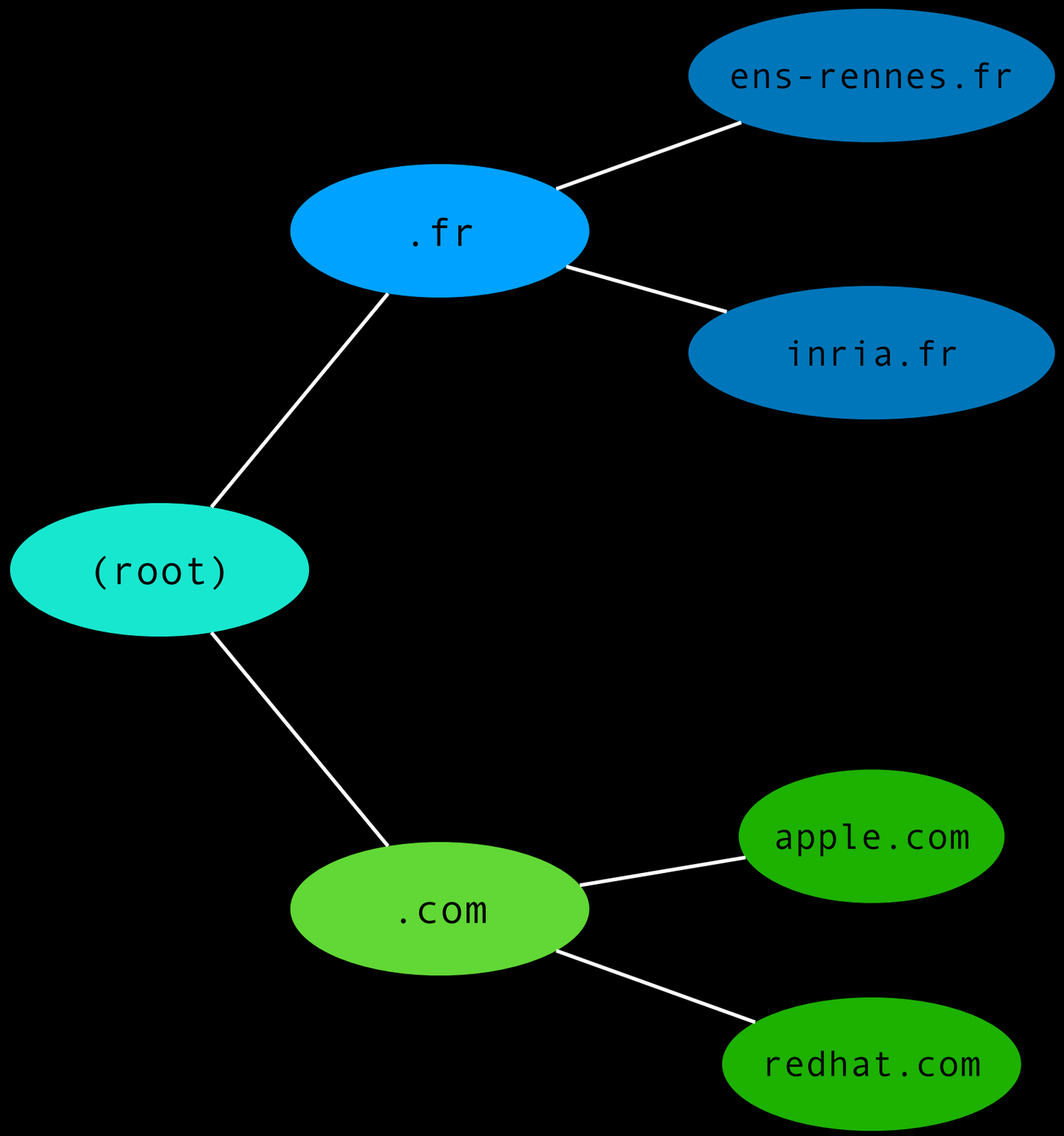
Quid de la sécurité

HTTPS: HTTP over TLS (Transport Layer Security)

- Hiérarchie de certificats (Crypto Asymétrique)
- Le serveur envoie son certificat public
- Le client vérifie la chaîne de signatures, jusqu’à une racine de confiance
- Utilisation de crypto asymétrique pour obtenir un secret partagé
- Tunnel comme TCP mais chiffré
- On cause HTTP dans le tunnel

Le protocole DNS: Domain Name System

Comment nommer les choses sur internet ?



The Problem

Humans want readable names - not numbers

```
# Host file
…
www.ens-rennes.fr 80.247.238.89
...
inria.fr          128.93.162.83
www.inria.fr      128.93.162.83
...
irisa.fr          131.254.254.107
www.irisa.fr      131.254.254.107

...
```

Does not scale !

> 10^8 names,

Host file is several GB

Updates? at least once a day.

> 10^9 clients,

Even incremental update is hard on the network. (If even possible)

Comment résoudre ce problème

Les idées clefs derrière le DNS - Be Lazy

1. Demander ce dont on a besoin

2. Déléguer

3. Garder en cache

BONUS : Différents types d'entrées, extensible

Anatomie d'une entrée DNS

Où se trouve mquinson.frama.io

NAME	TTL	CLASS	TYPE	VALUE
Question:	Time To Live : Durée de validité			
mquison.frama.io.		IN	AAAA IPv6	
Réponse:				
mquison.frama.io.	2581	IN	CNAME	frama.io.
frama.io.	2581	IN	AAAA	2a01:4f8:231:4c99::42

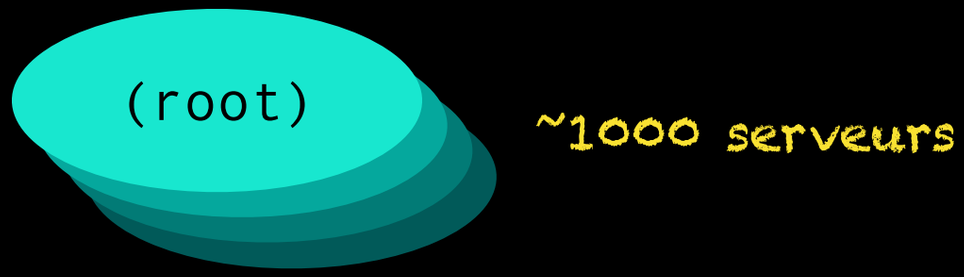
Question:				
mquison.frama.io.		IN	A IPv4	
Réponse:				
mquison.frama.io.	2780	IN	CNAME	frama.io.
frama.io.	2780	IN	A	176.9.183.74

IN pour Internet

https://en.wikipedia.org/wiki/List_of_DNS_record_types

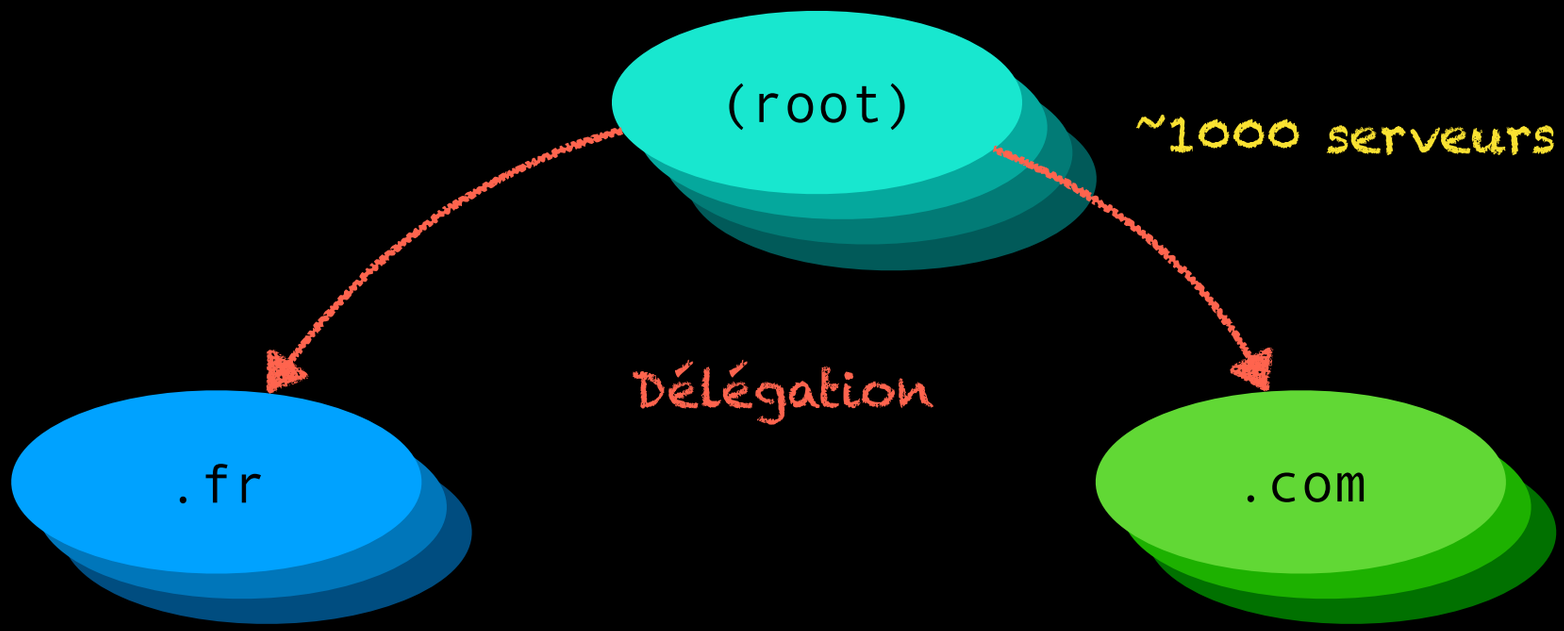
Déléguer ?

Une hiérarchie d’autorité



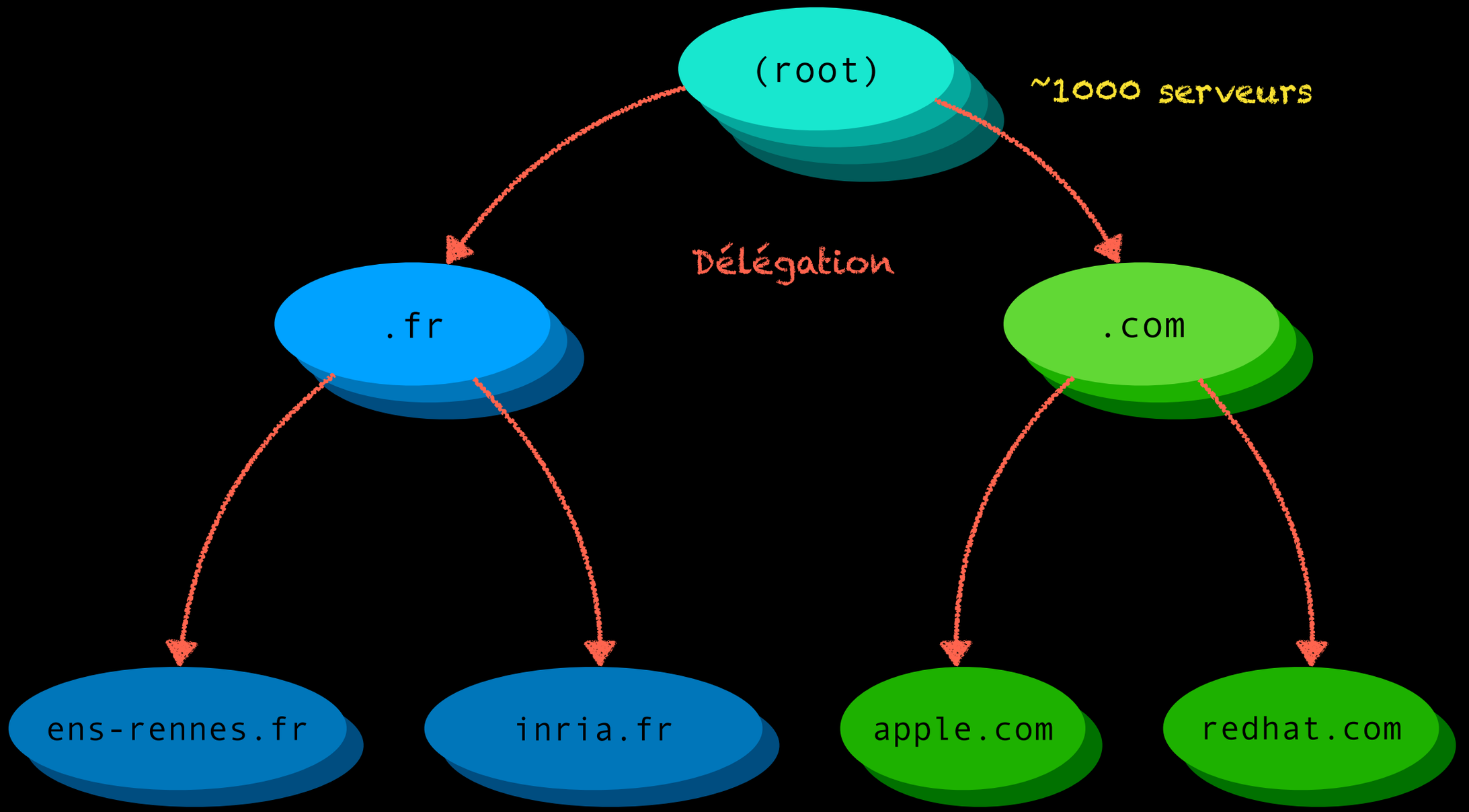
Déléguer ?

Une hiérarchie d'autorité



Déléguer ?

Une hiérarchie d'autorité



Au moins 2 serveurs

Utilisation de DNS

Requête itérative (sans mise en cache)

(root)

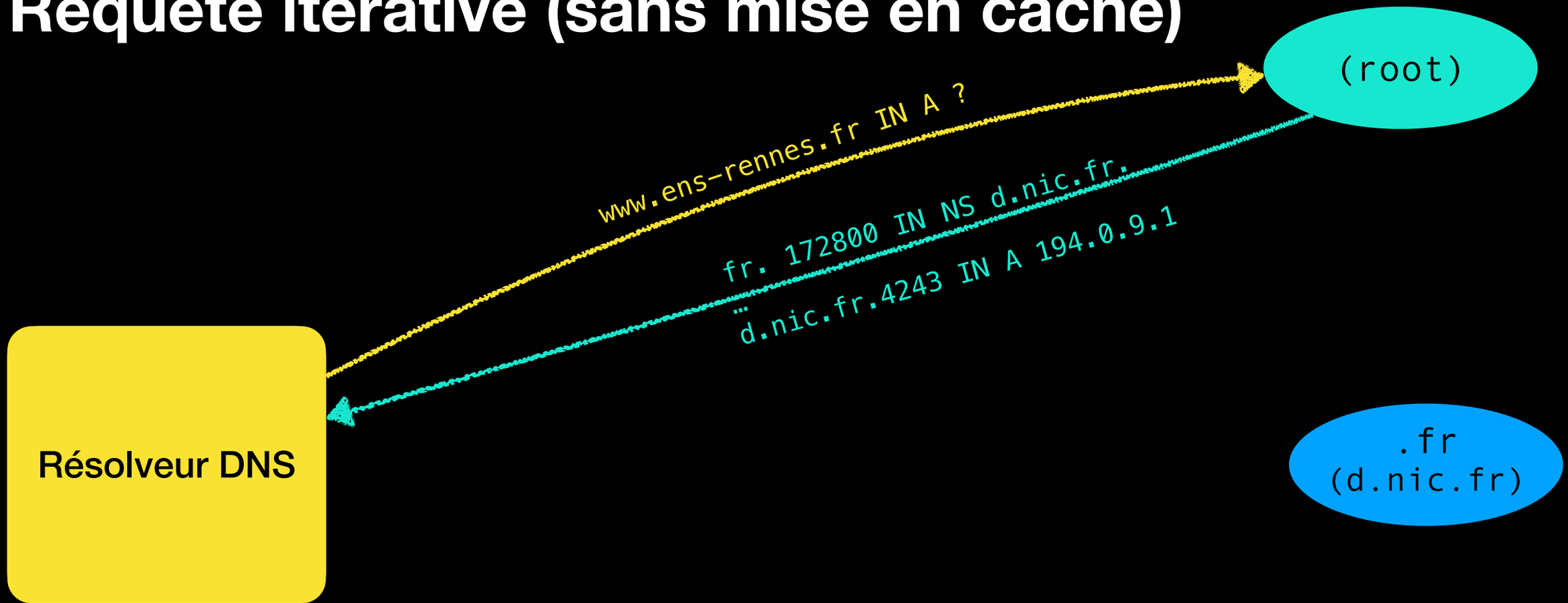
Résolveur DNS

Utilisation de DNS

Requête itérative (sans mise en cache)

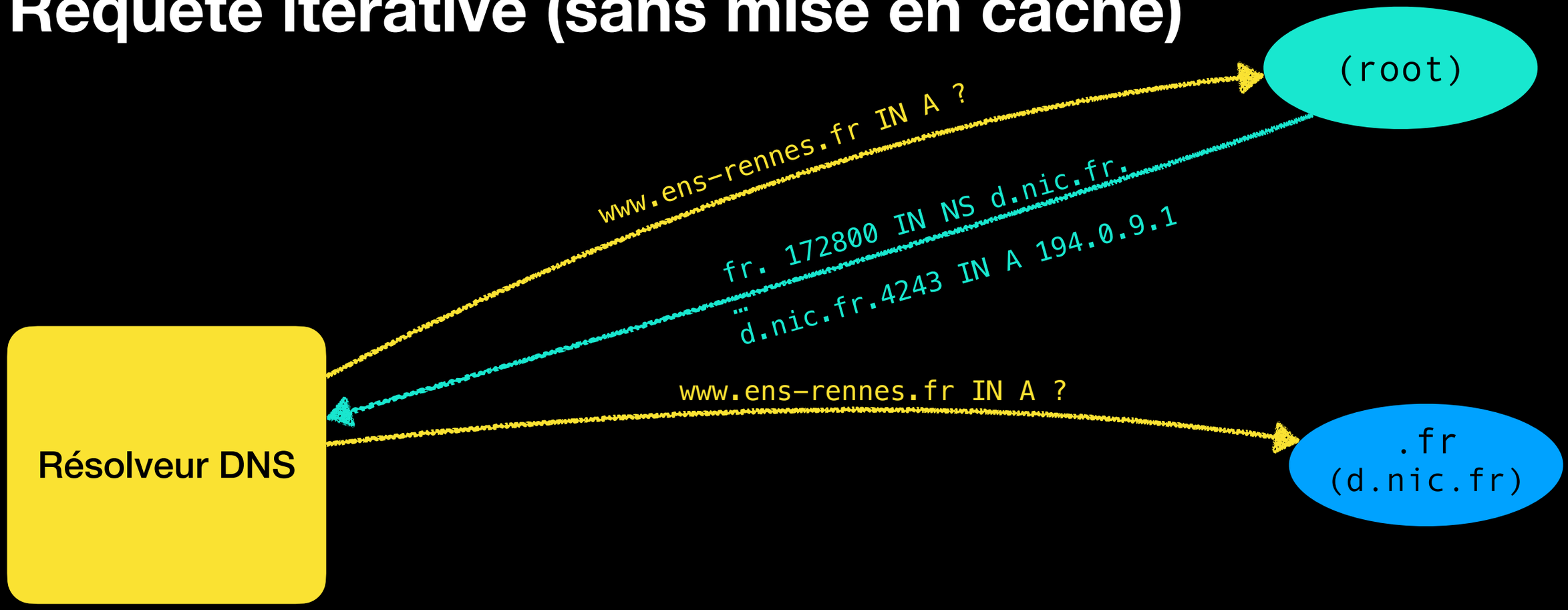
Utilisation de DNS

Requête itérative (sans mise en cache)



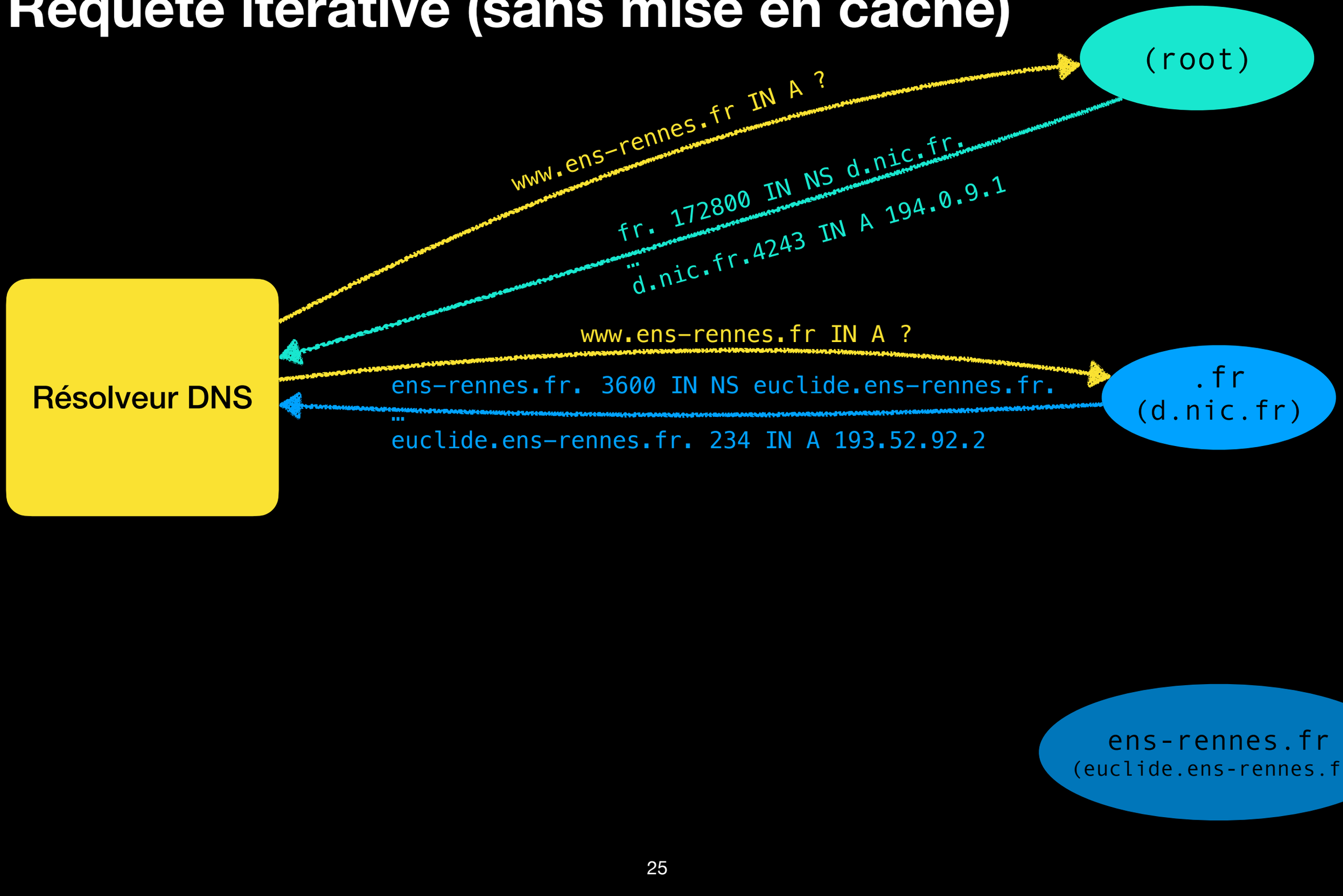
Utilisation de DNS

Requête itérative (sans mise en cache)



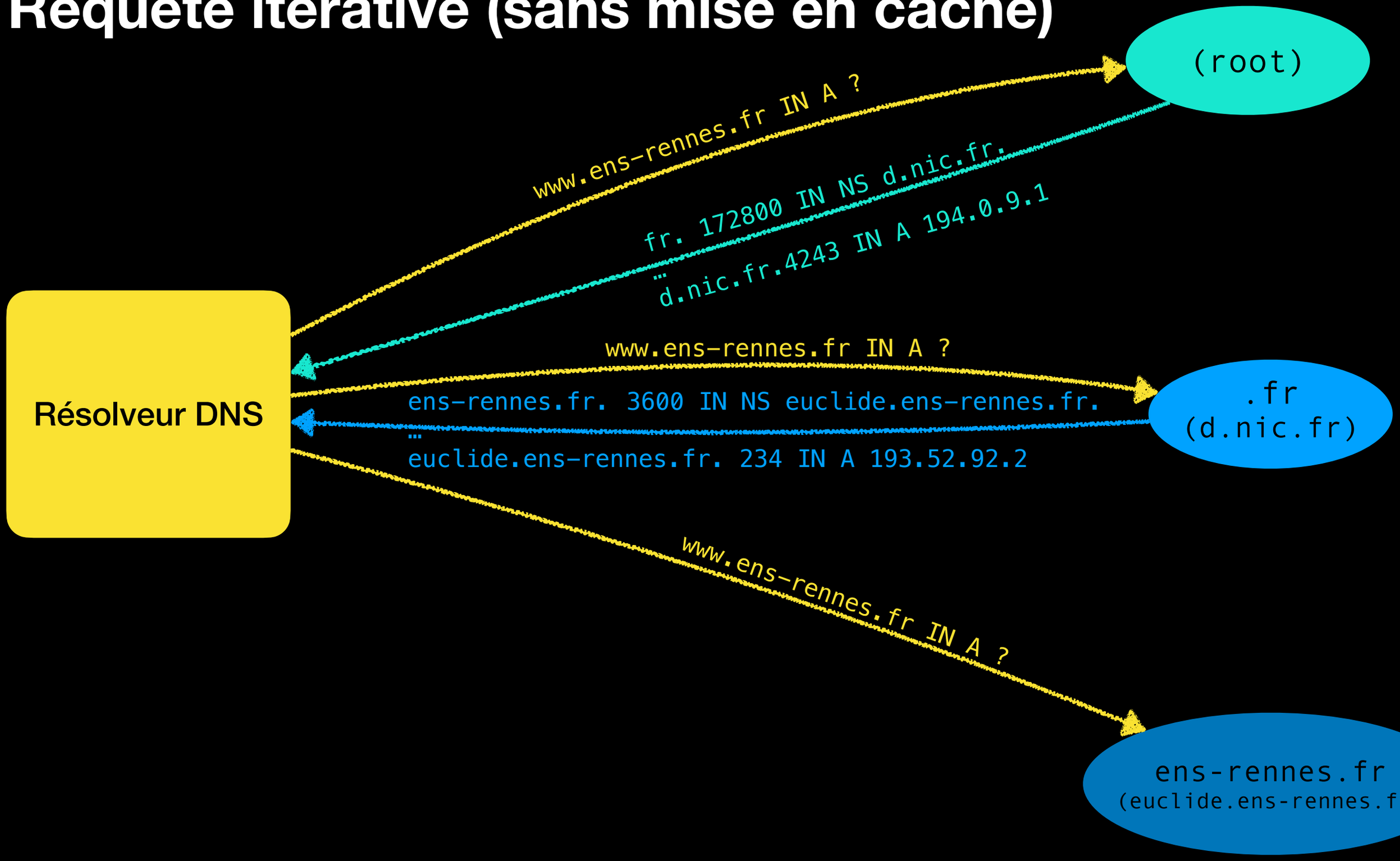
Utilisation de DNS

Requête itérative (sans mise en cache)



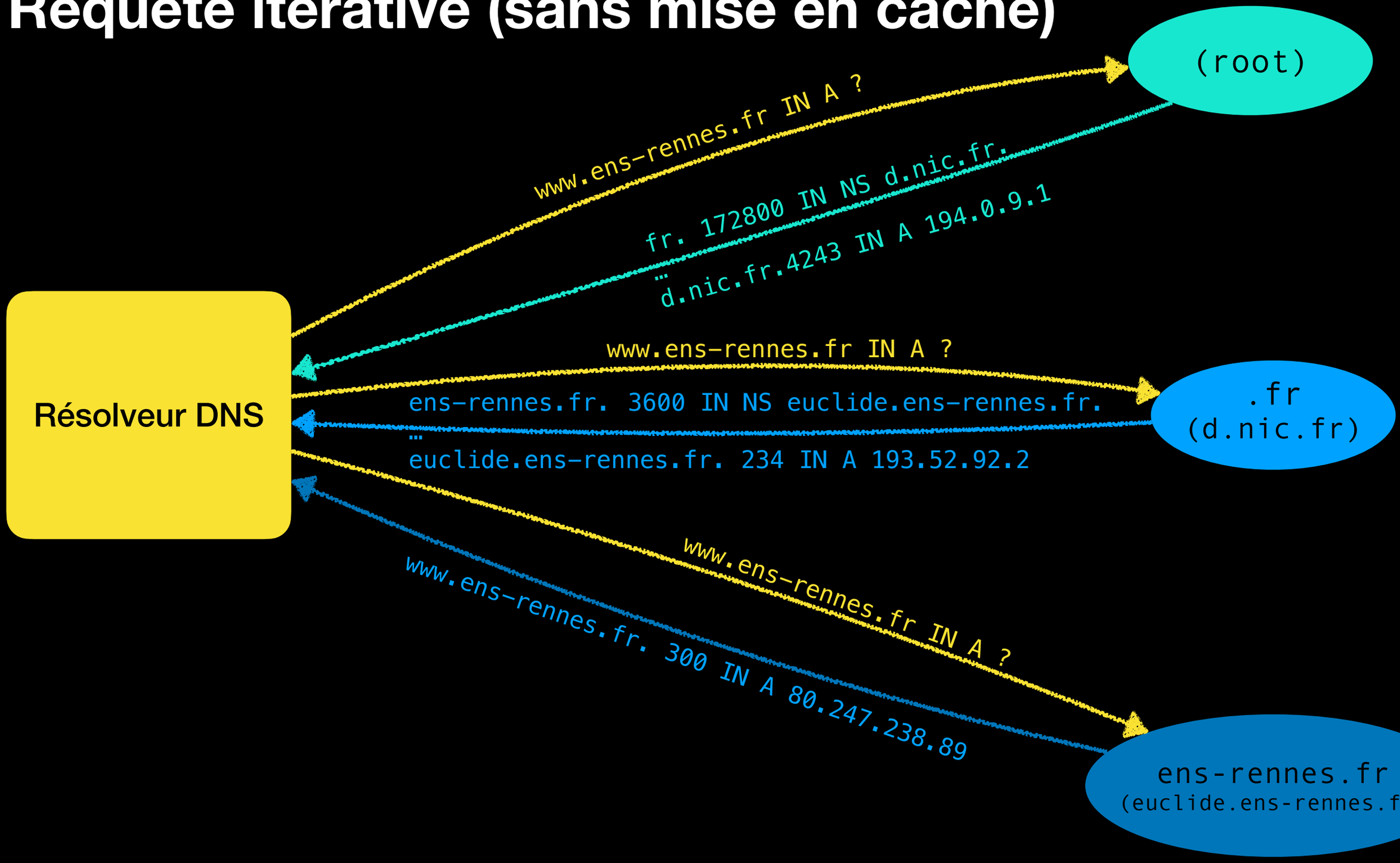
Utilisation de DNS

Requête itérative (sans mise en cache)



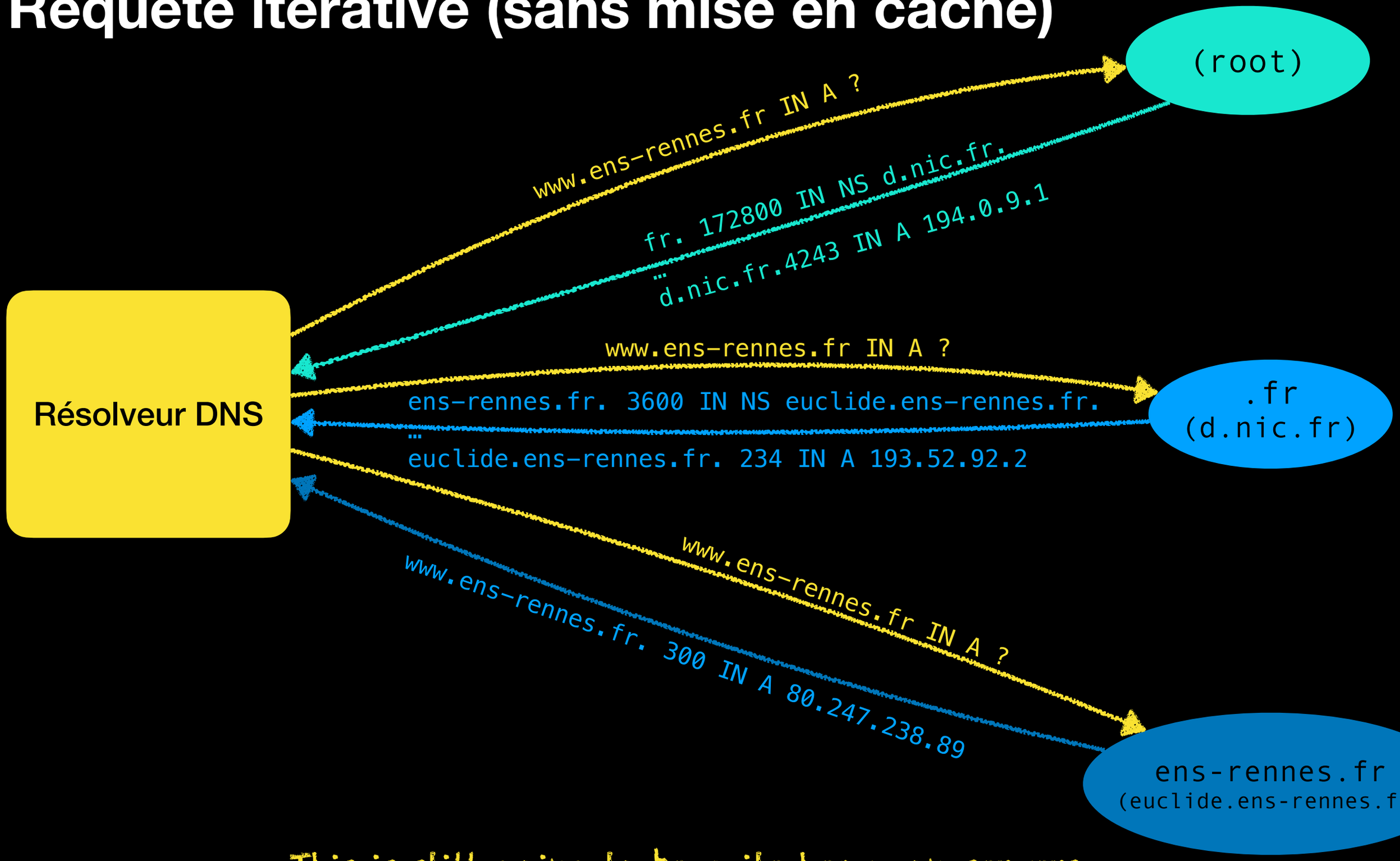
Utilisation de DNS

Requête itérative (sans mise en cache)



Utilisation de DNS

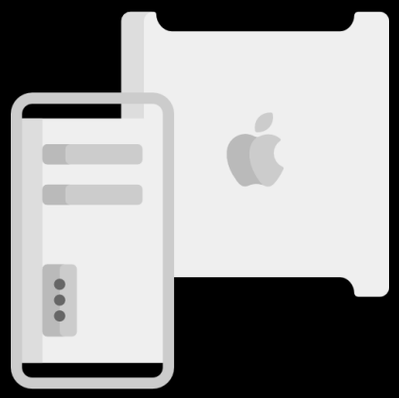
Requête itérative (sans mise en cache)



This is still going to be quite heavy on servers

Résolveur récursif et caching

Scaling to the internet



Résolveur DNS
Local

Cache DNS

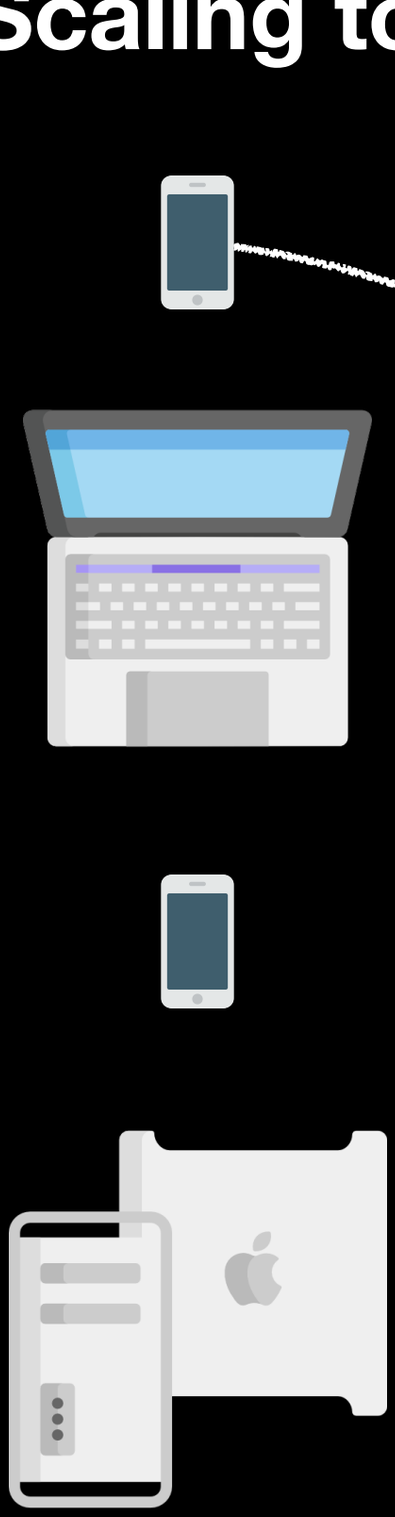
(root)

.fr
(d.nic.fr)

ens-rennes.fr
(euclide.ens-rennes.fr)

Résolveur récursif et caching

Scaling to the internet



Résolveur DNS
Local

Cache DNS

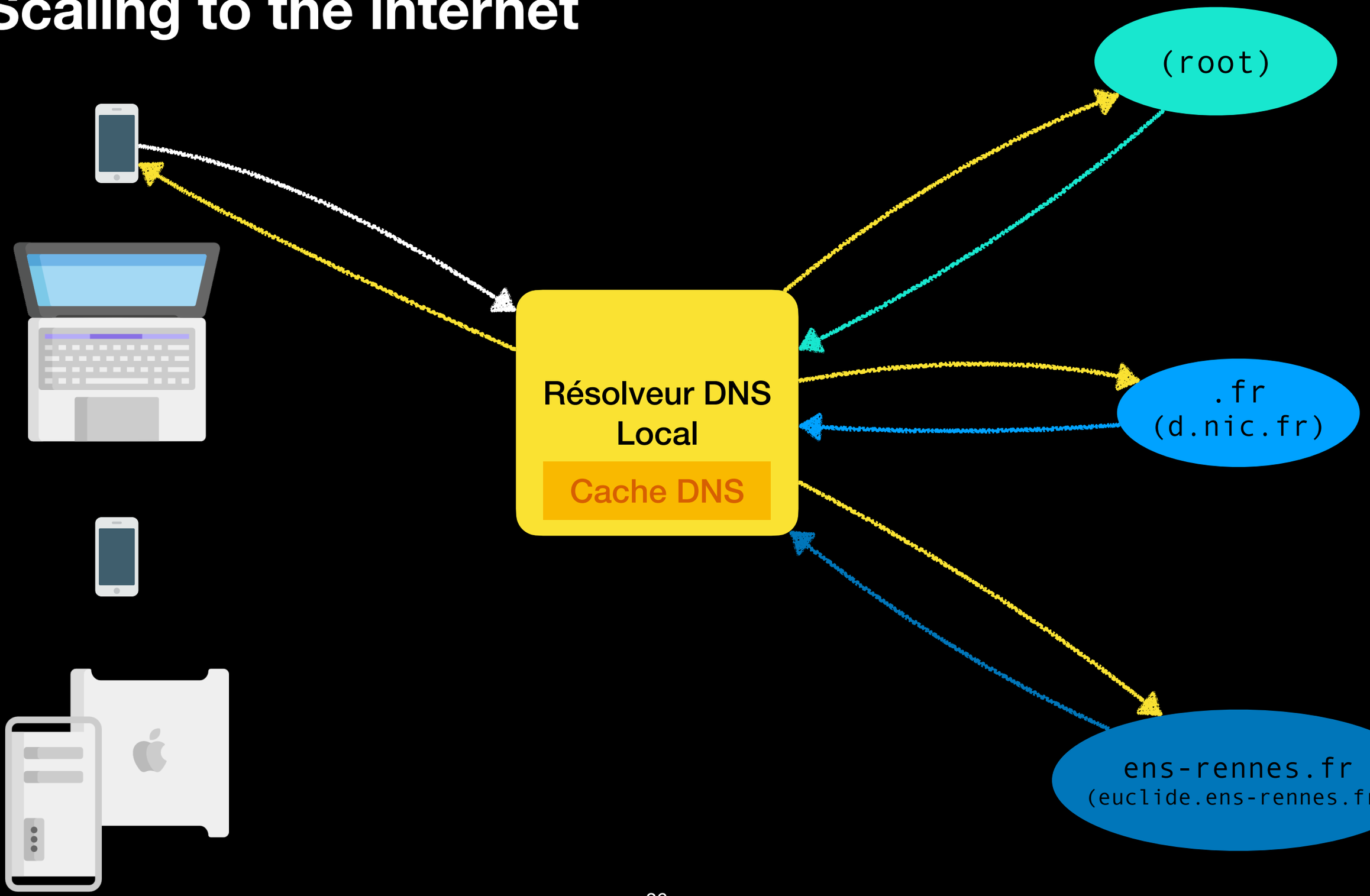
(root)

.fr
(d.nic.fr)

ens-rennes.fr
(euclide.ens-rennes.fr)

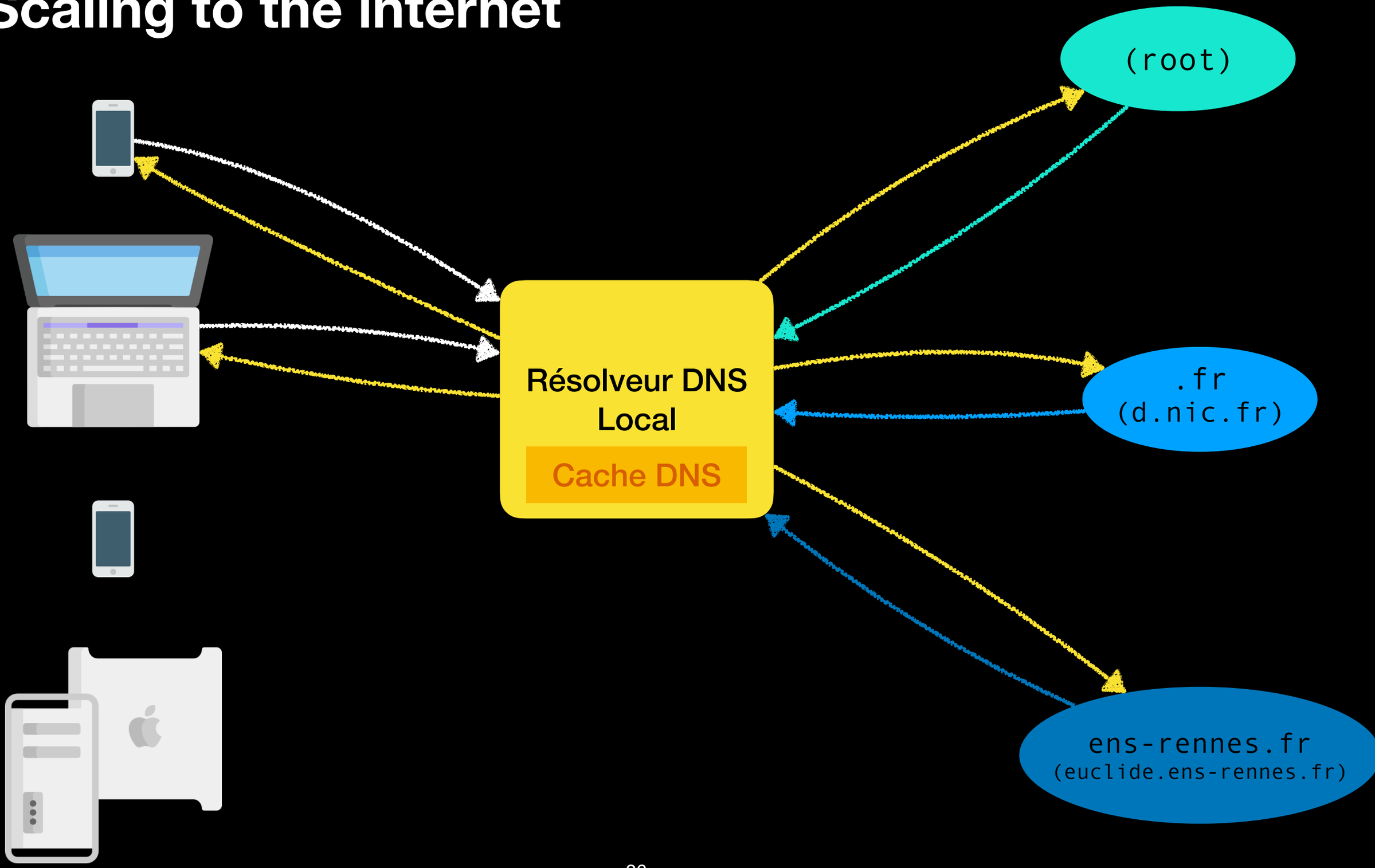
Résolveur récursif et caching

Scaling to the internet



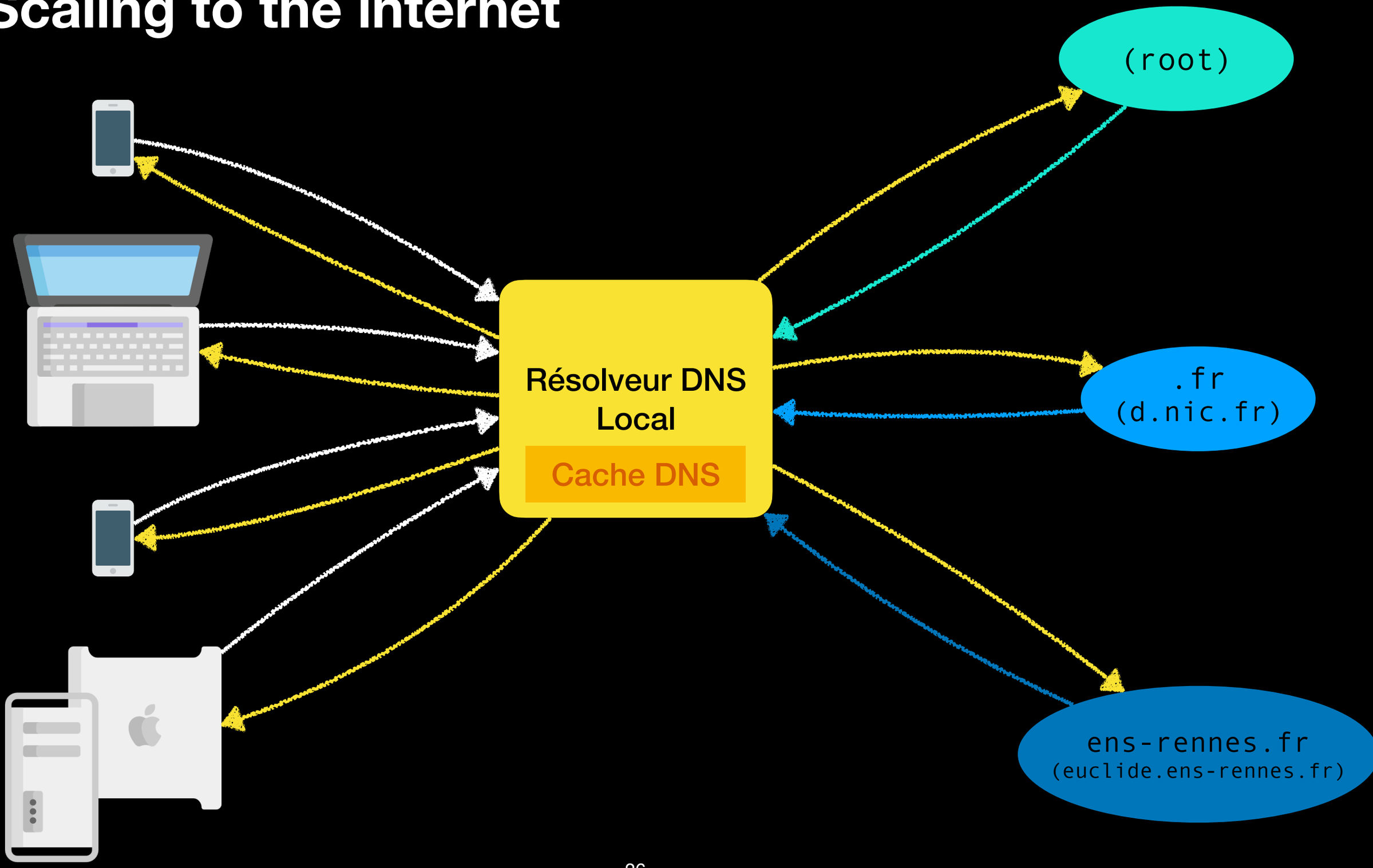
Résolveur récursif et caching

Scaling to the internet



Résolveur récursif et caching

Scaling to the internet



Invalidation des caches

L’autre problème difficile

TTL : Time to Live

Compromis entre
fréquence de MàJ
Efficacité du Cache

DNS over UDP ou TCP

Quel protocole de transport

Requête courte ? UDP/53 !

Requête trop longue ? TCP/53, du coup

Requête confidentielle ? DNS over TLS (over TCP)
DNS over QUIC un jour ?

DNS has issues

Et la sécurité dans tout ça

- Votre FAI est-il digne de confiance ?

Pas vraiment

- Diffuser des fausses réponses sur un Wifi public ?

En UDP, ça se fait bien >o)

- Comment vérifier qu'une réponse est authentique ?

DNSSEC existe, est-il utilisé ?

Résumé

DNS in a nutshell

- Hiérarchie de serveurs autoritaires
- Résolveurs locaux qui effectuent l’itération et mettent en cache
- TTL configurable (détermine le temps de propagation)
- Base de donnée distribuée.
- Pilier fondamental d’internet

Questions ?

N’oubliez pas, e-mail à l’équipe (3 personnes),
Office Hours jeudi 18h-19h sur discord

Résumé

What did we see today?

- Différence de service entre TCP et UDP
- Comment utiliser UDP avec des sockets
- Un exemple de protocol application (TCP) : HTTP
- Le système DNS

Est-ce que vos notes ont tout ça ?

La prochaine fois

Teaser

- Discussion rapide de ce que fourni la couche IP
- Comment garantir un transport fiable au dessus d'un réseau non-fiable (perte de Paquet)
- Les problèmes de débit et latence associés