# DHCP - DNS
# REVPROXY

# DHCP

Dynamic Host Configuration Protocol

# COMMUNIQUER SUR UN RÉSEAU

| | | |
|---|---|---|
| **Adresse MAC** | identifiant de la carte réseau | `e8:f1:40:03:58:85` |
| | | |
| | | |

# COMMUNIQUER SUR UN RÉSEAU

| | | |
|---|---|---|
| **Adresse MAC** | identifiant de la carte réseau | `e8:f1:40:03:58:85` |
| **Adresse IP** | adresse de la machine sur le réseau | `192.168.1.69/24` |

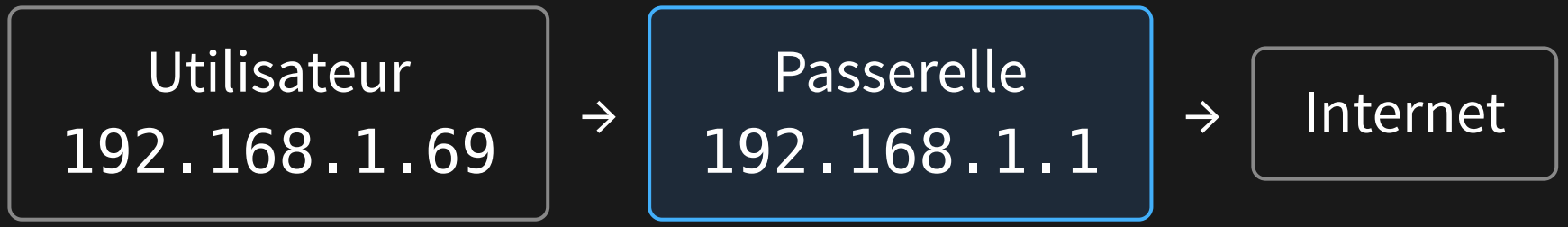
# COMMUNIQUER SUR UN RÉSEAU

| | | |
|---|---|---|
| **Adresse MAC** | identifiant de la carte réseau | `e8:f1:40:03:58:85` |
| **Adresse IP** | adresse de la machine sur le réseau | `192.168.1.69/24` |
| **Table ARP** | correspondance IP → MAC sur le réseau local | `ip neigh` |

# COMMUNIQUER SUR UN RÉSEAU

| | | |
|---|---|---|
| **Adresse MAC** | identifiant de la carte réseau | `e8:f1:40:03:58:85` |
| **Adresse IP** | adresse de la machine sur le réseau | `192.168.1.69/24` |
| **Table ARP** | correspondance IP → MAC sur le réseau local | `ip neigh` |
| **Passerelle par défaut** | routeur à qui envoyer tout ce qui sort du réseau local | `192.168.1.1` |

# COMMUNIQUER SUR UN RÉSEAU




Pour sortir du réseau local, on passe par la passerelle par défaut

# JE PEUX ME CO ?

Nouvelle machine
e8:f1:40:03:58:85
IP : ???

?

Routeur
192.168.1.1

# JE PEUX ME CO ?

Nouvelle machine
`e8:f1:40:03:58:85`
`IP : ???`

?

Routeur
`192.168.1.1`

Une machine qui arrive sur le réseau n'a qu'une adresse MAC

# JE PEUX ME CO ?

Nouvelle machine
`e8:f1:40:03:58:85`
`IP : ???`

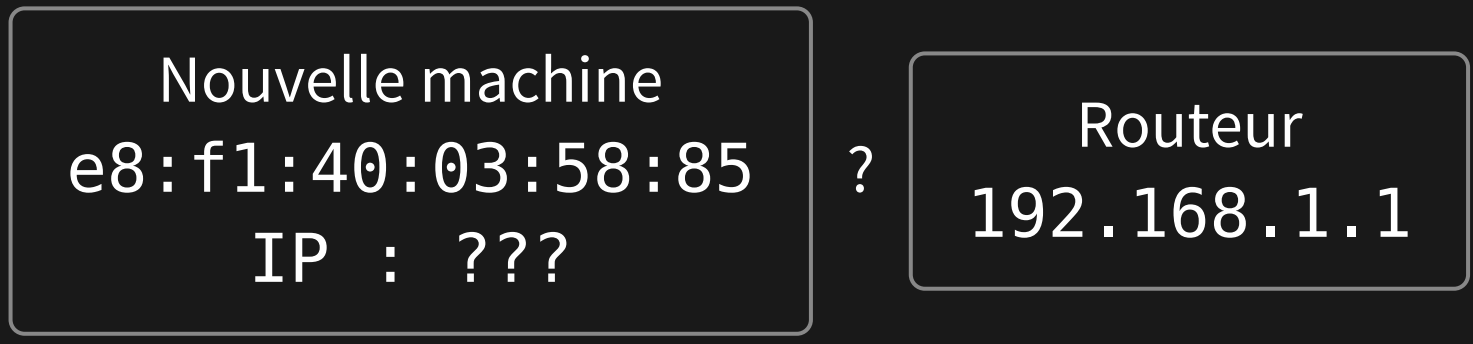
?

Routeur
`192.168.1.1`

Une machine qui arrive sur le réseau n'a qu'une adresse MAC

Configurer l'IP à la main sur chaque machine ? Non merci.

# JE PEUX ME CO ?




Une machine qui arrive sur le réseau n'a qu'une adresse MAC

Configurer l'IP à la main sur chaque machine ? Non merci.

**DHCP** : un serveur distribue automatiquement la configuration réseau

# DORA

Client

Serveur DHCP

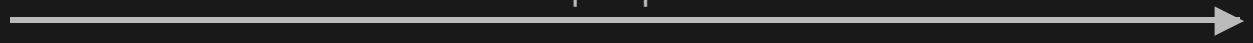
# DORA

Client

Serveur DHCP

DHCP **DI**SCOVER

Y'a quelqu'un ?



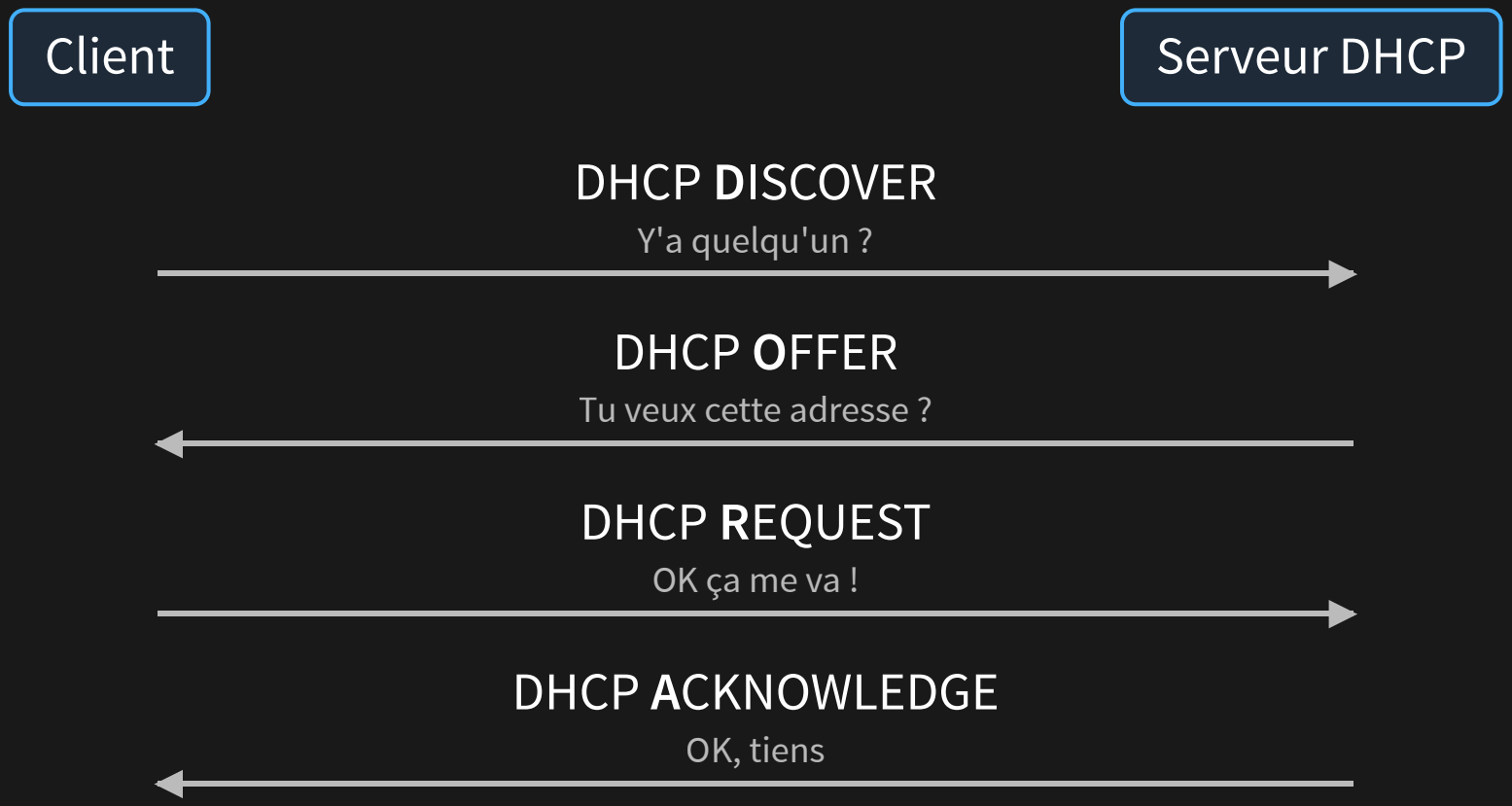
# DORA


Client
Serveur DHCP
DHCP DISCOVER
Y'a quelqu'un ?
DHCP OFFER
Tu veux cette adresse ?

# DORA


Client
Serveur DHCP
DHCP DISCOVER
Y'a quelqu'un ?
DHCP OFFER
Tu veux cette adresse ?
DHCP REQUEST
OK ça me va !

# DORA


Client
Serveur DHCP
DHCP DISCOVER
Y'a quelqu'un ?
DHCP OFFER
Tu veux cette adresse ?
DHCP REQUEST
OK ça me va !
DHCP ACKNOWLEDGE
OK, tiens

# DISCOVER

Le client envoie une requête en **broadcast** pour savoir s'il existe un serveur DHCP sur le réseau

| | | |
|---|---|---|
| Source | `0.0.0.0:68` | le client n'a pas encore d'IP |
| Destination | `255.255.255.255:67` | tout le réseau local |
| Contenu | l'adresse MAC du client | |

# DISCOVER

Le client envoie une requête en **broadcast** pour savoir s'il existe un serveur DHCP sur le réseau

| | | |
|---|---|---|
| Source | `0.0.0.0:68` | le client n'a pas encore d'IP |
| Destination | `255.255.255.255:67` | tout le réseau local |
| Contenu | l'adresse MAC du client | |

DHCP utilise UDP : port 67 pour le serveur, 68 pour le client

# OFFER

Le serveur propose au client une adresse choisie dans sa **pool** d'adresses IP non utilisées

Pool
`192.168.1.10` → `192.168.1.200`

→

Proposition
`192.168.1.69`

# OFFER

Le serveur propose au client une adresse choisie dans sa **pool** d'adresses IP non utilisées

Pool
`192.168.1.10` → `192.168.1.200`

→

Proposition
`192.168.1.69`

Plusieurs serveurs DHCP peuvent répondre donc le client choisit une seule offre

# REQUEST

Le client indique qu'il accepte la proposition d'adresse IP

# REQUEST

Le client indique qu'il accepte la proposition d'adresse IP

En **broadcast** pour que les autres serveurs voient que leur offre est refusée

# ACK

Le serveur confirme et envoie toutes les informations nécessaires pour se connecter au réseau

# ACK

Le serveur confirme et envoie toutes les informations nécessaires pour se connecter au réseau

| | |
|---|---|
| **Adresse IP** | `192.168.1.69` |
| | |
| | |
| | |

# ACK

Le serveur confirme et envoie toutes les informations nécessaires pour se connecter au réseau

| | |
|---|---|
| **Adresse IP** | `192.168.1.69` |
| **Masque** | `255.255.255.0` (`/24`) |
| | |
| | |

# ACK

Le serveur confirme et envoie toutes les informations nécessaires pour se connecter au réseau

| | |
|---|---|
| **Adresse IP** | `192.168.1.69` |
| **Masque** | `255.255.255.0` (`/24`) |
| **Passerelle** | `192.168.1.1` |
| | |

# ACK

Le serveur confirme et envoie toutes les informations nécessaires pour se connecter au réseau

| | |
|---|---|
| **Adresse IP** | `192.168.1.69` |
| **Masque** | `255.255.255.0` (`/24`) |
| **Passerelle** | `192.168.1.1` |
| **Serveurs DNS** | `1.1.1.1` |

# ACK

Le serveur confirme et envoie toutes les informations nécessaires pour se connecter au réseau

| | |
|---|---|
| **Adresse IP** | `192.168.1.69` |
| **Masque** | `255.255.255.0` (`/24`) |
| **Passerelle** | `192.168.1.1` |
| **Serveurs DNS** | `1.1.1.1` |
| **Durée du bail** | `86400` s |

# LE BAIL

# LE BAIL

- À 50 % du bail : le client redemande au même serveur (REQUEST en unicast)

# LE BAIL

- À 50 % du bail : le client redemande au même serveur (REQUEST en unicast)
- À 87,5 % : il redemande à n'importe quel serveur (broadcast)

# LE BAIL

- À 50 % du bail : le client redemande au même serveur (`REQUEST` en unicast)
- À 87,5 % : il redemande à n'importe quel serveur (broadcast)
- Si le bail expire, il perd son adresse et recommence un `DISCOVER`

# ET MAINTENANT ?



# ET MAINTENANT ?




« Je suis 192.168.1.69 et je veux aller sur internet »

# DNS

Domain Name System

# POURQUOI LE DNS ?

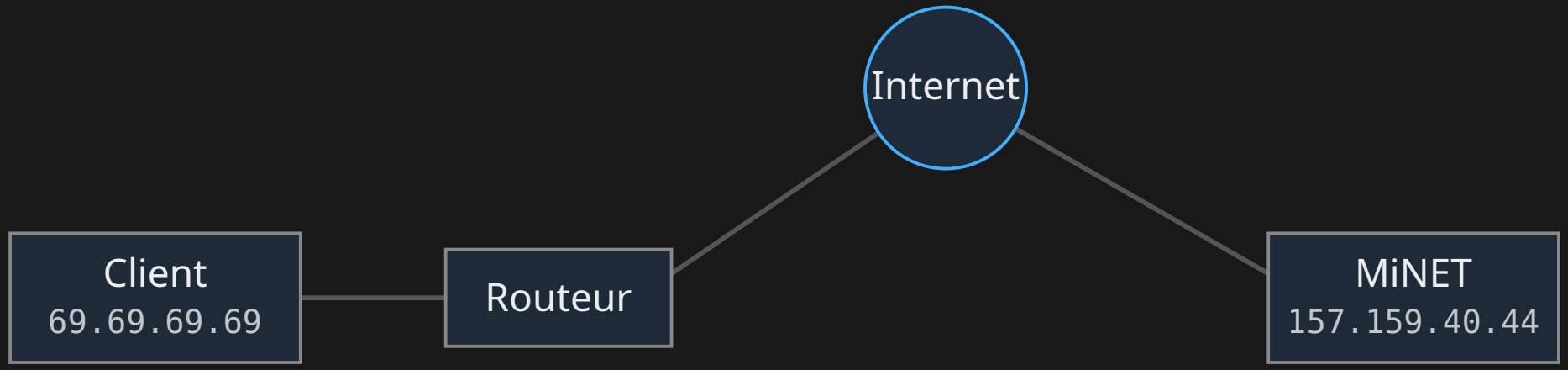
« Je veux voir le site de MiNET, c'est quoi l'IP déjà ? »

# POURQUOI LE DNS ?

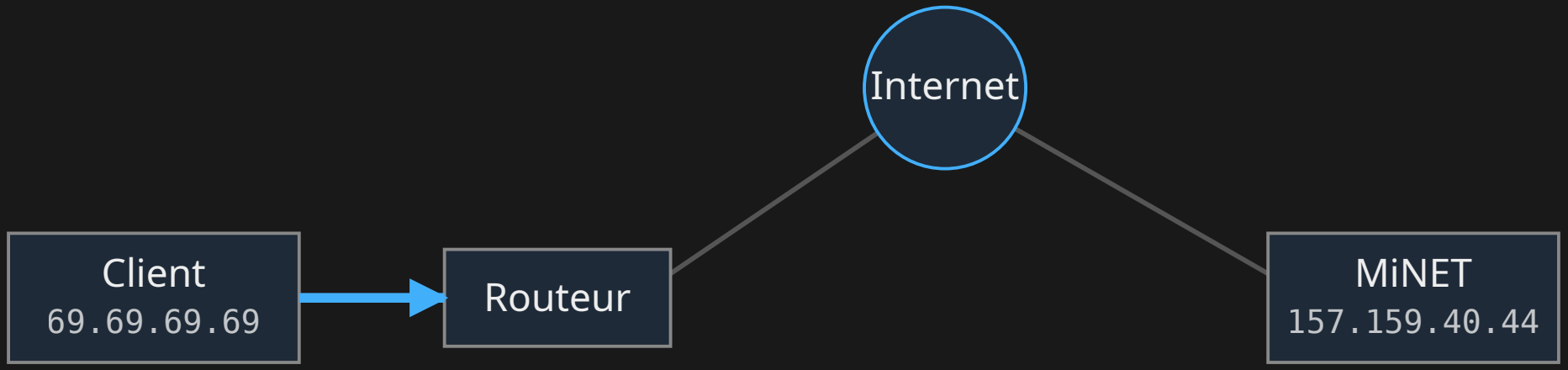
« Je veux voir le site de MiNET, c'est quoi l'IP déjà ? »

`minet.net` → `157.159.40.44`

# SANS DNS

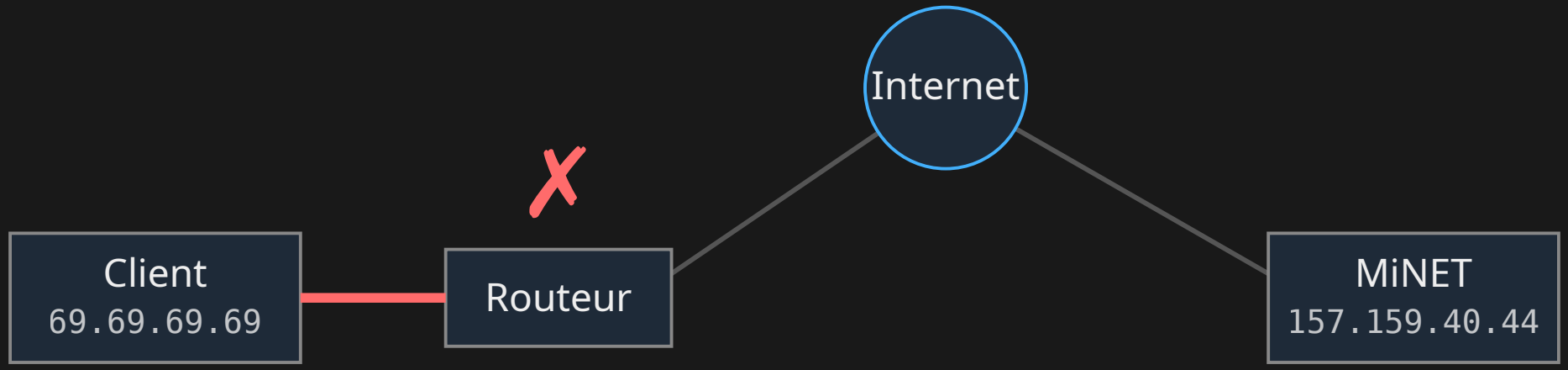


# SANS DNS



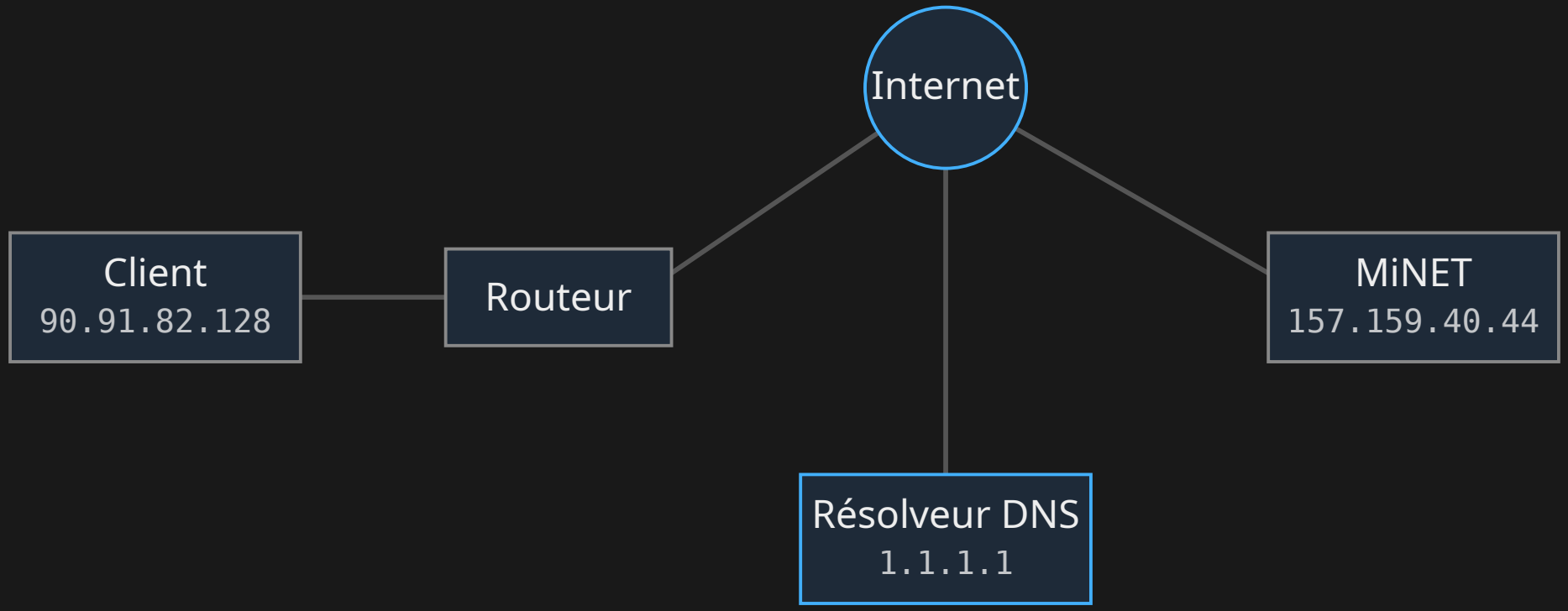

Client : « Montre-moi minet.net »

# SANS DNS

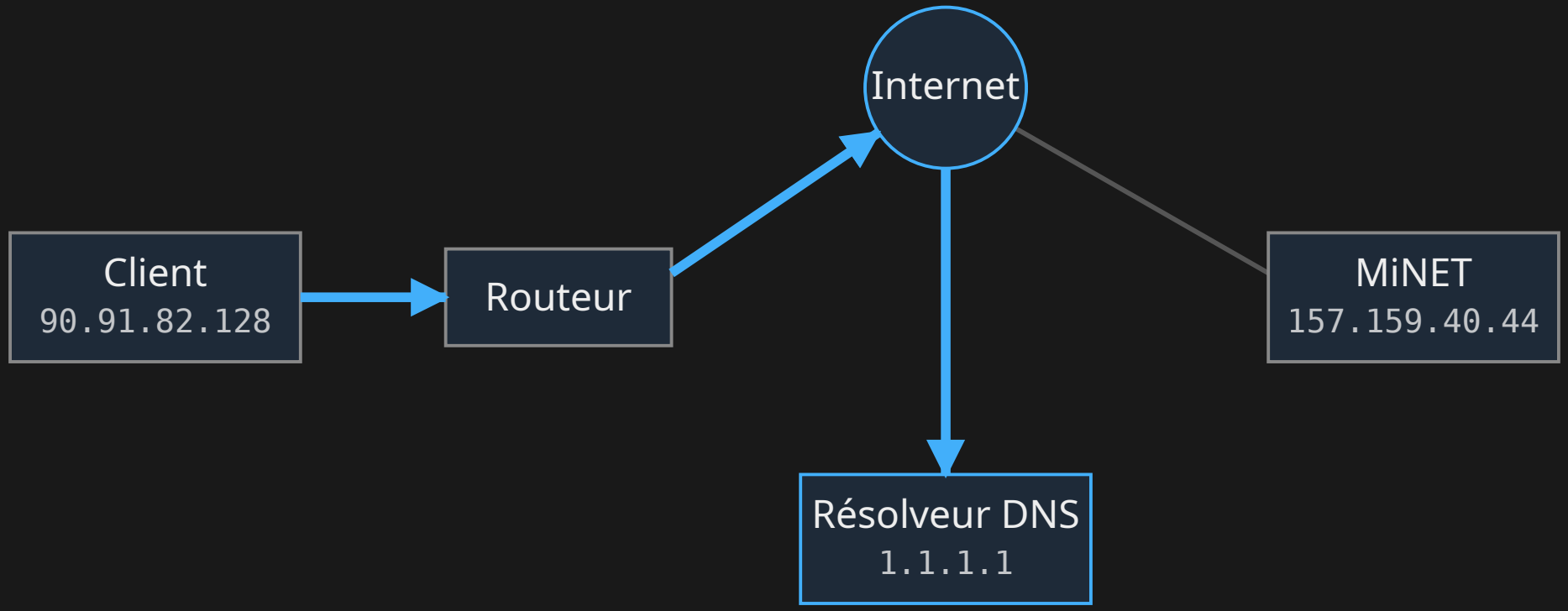



Routeur : « Je ne comprends que les IP, c'est quoi ça ? »

# AVEC DNS

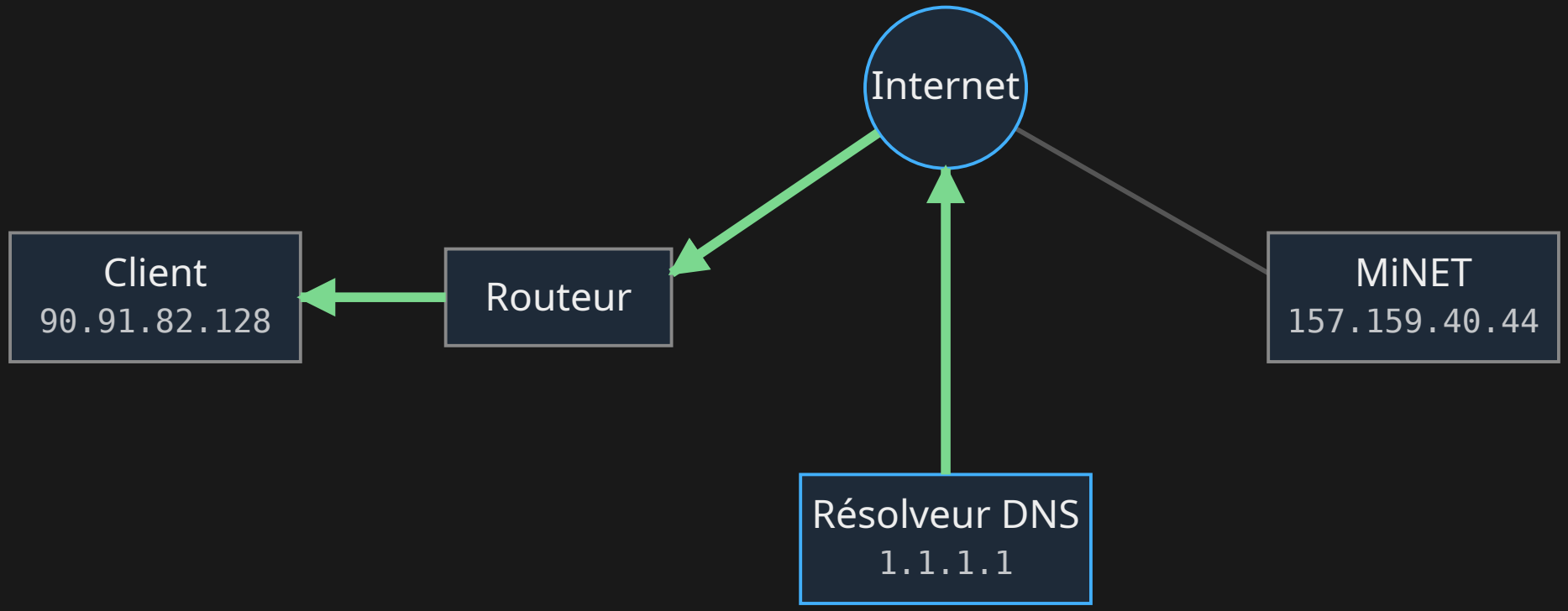

Internet
Client
90.91.82.128
Routeur
MiNET
157.159.40.44
Résolveur DNS
1.1.1.1

# AVEC DNS



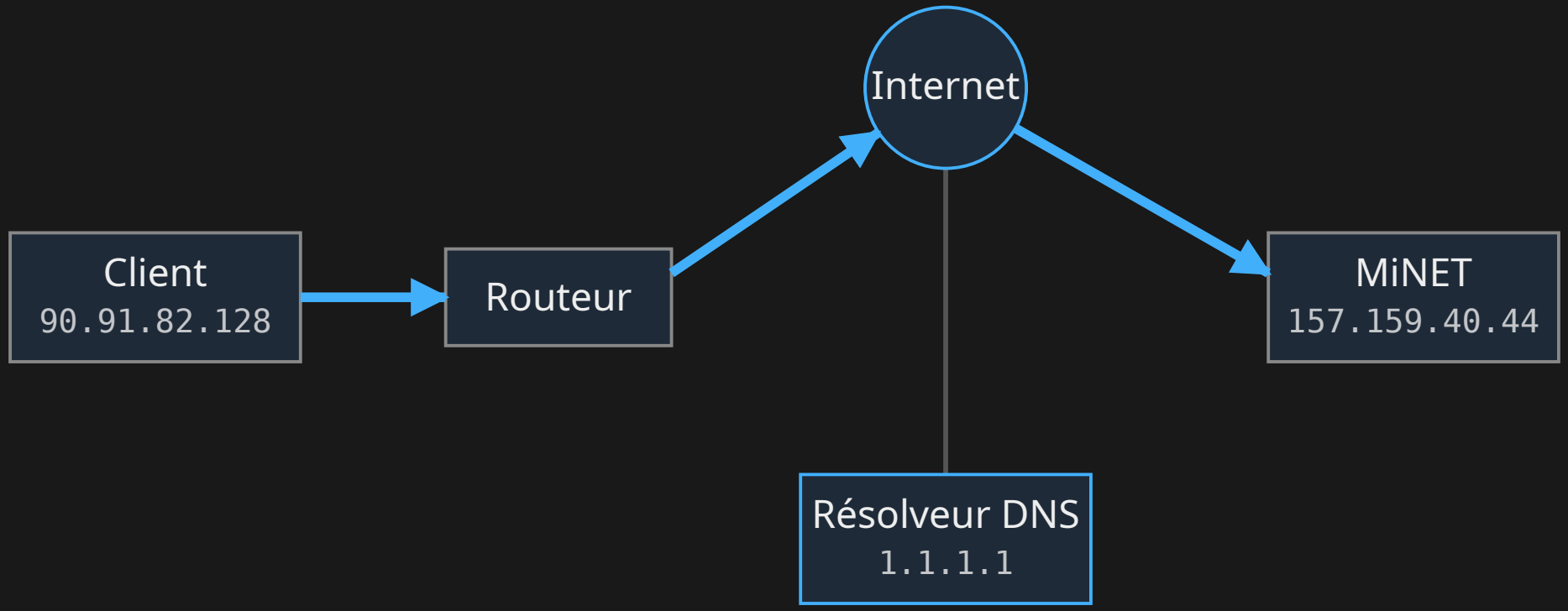

Client → 1.1.1.1 : « Quelle est l'adresse de minet.net ? »

# AVEC DNS



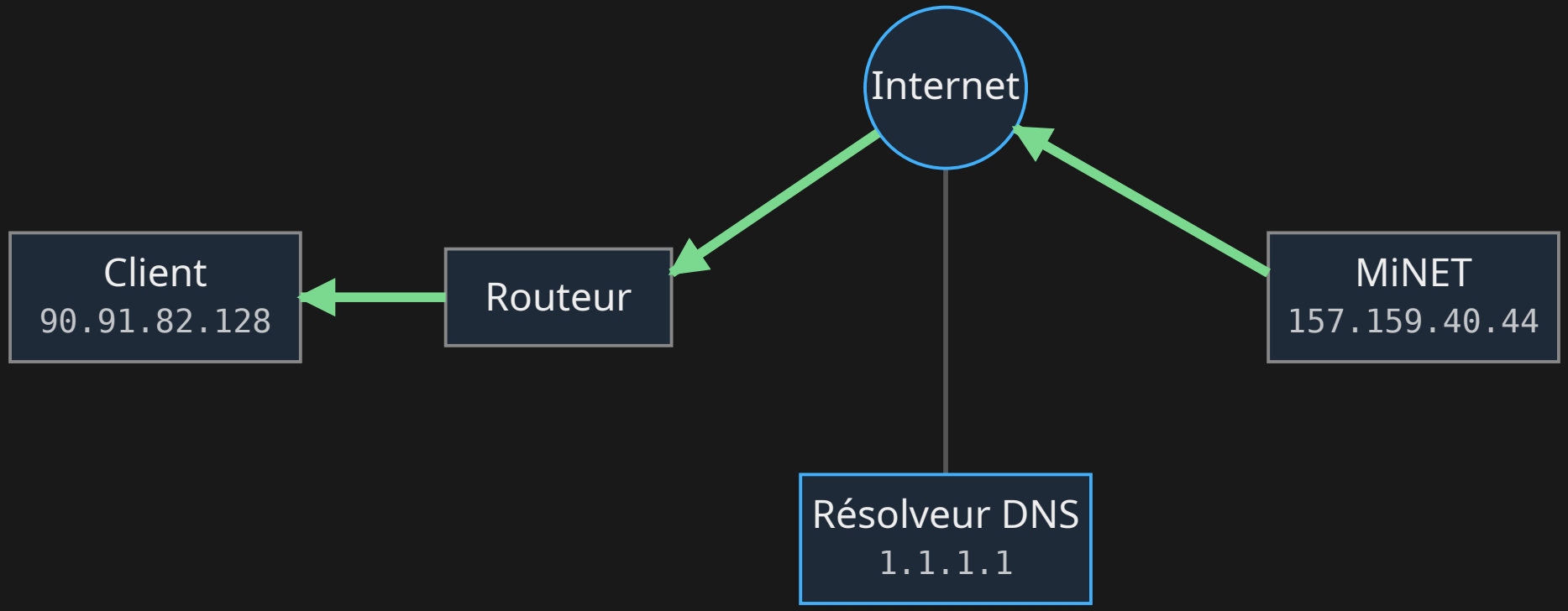

1.1.1.1 → Client : « minet.net est à 157.159.40.44 »

# AVEC DNS




Client → MiNET : « Montre-moi 157.159.40.44 »

# AVEC DNS




MiNET → Client : « Voilà la page » ✓

# UN NOM DE DOMAINE

formations.minet.net._

# UN NOM DE DOMAINE

`formations.minet.net.`

| | | |
|---|---|---|
| `.` | racine (root) | 13 serveurs racine, `a` à `m.root-servers.net` |

# UN NOM DE DOMAINE

`formations.minet.net.`

| | | |
|---|---|---|
| `.` | racine (root) | 13 serveurs racine, `a` à `m.root-servers.net` |
| `net` | TLD (Top Level Domain) | géré par Verisign |
| | | |

# UN NOM DE DOMAINE

`formations.minet.net.`

| | | |
|---|---|---|
| `.` | racine (root) | 13 serveurs racine, `a` à `m.root-servers.net` |
| `net` | TLD (Top Level Domain) | géré par Verisign |
| `minet` | domaine | acheté chez un registrar |

# UN NOM DE DOMAINE

`formations.minet.net.`

| | | |
|---|---|---|
| `.` | racine (root) | 13 serveurs racine, `a` à `m.root-servers.net` |
| `net` | TLD (Top Level Domain) | géré par Verisign |
| `minet` | domaine | acheté chez un registrar |
| `formations` | sous-domaine | créé par minet |

# UN NOM DE DOMAINE

`formations.minet.net.`

| | | |
|---|---|---|
| `.` | racine (root) | 13 serveurs racine, `a` à `m.root-servers.net` |
| `net` | TLD (Top Level Domain) | géré par Verisign |
| `minet` | domaine | acheté chez un registrar |
| `formations` | sous-domaine | créé par minet |

On lit de droite à gauche : chaque niveau délègue au suivant

# RÉSOLUTION RÉCURSIVE

Le client ne demande qu'au résolveur

Client

Résolveur `1.1.1.1`

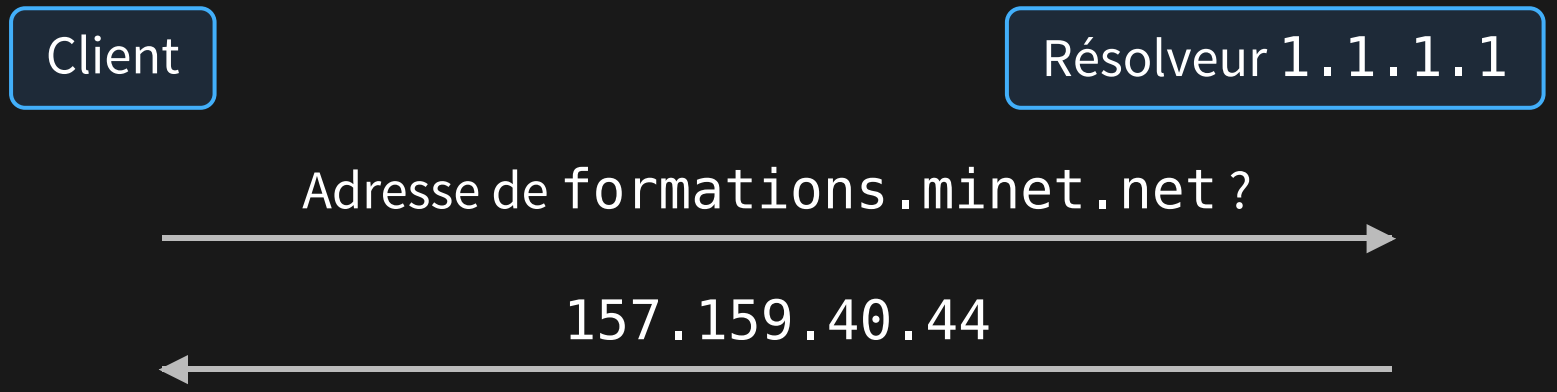
# RÉSOLUTION RÉCURSIVE

Le client ne demande qu'au résolveur



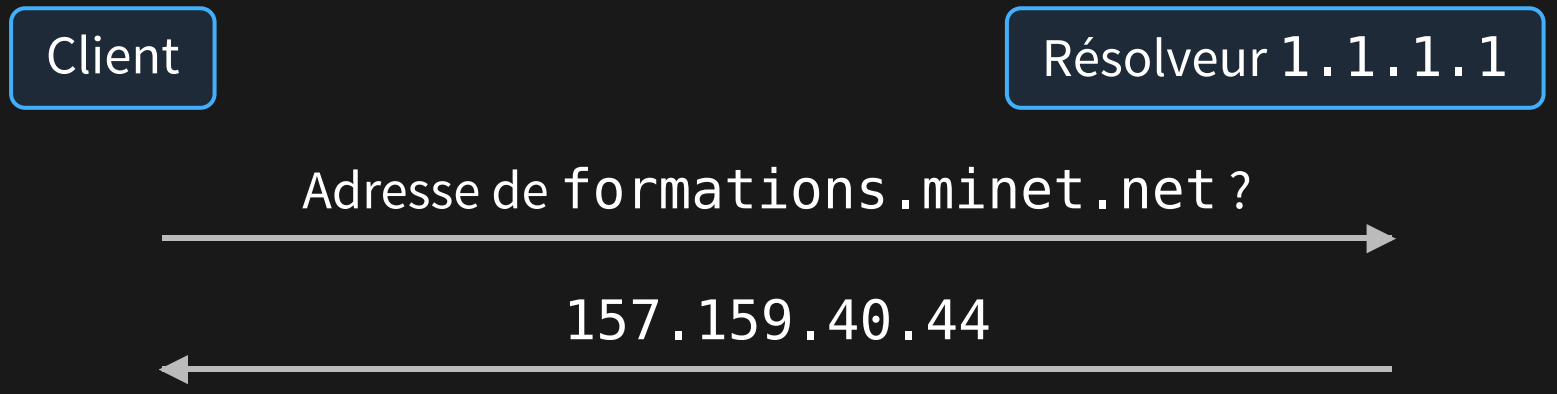
# RÉSOLUTION RÉCURSIVE

Le client ne demande qu'au résolveur



# RÉSOLUTION RÉCURSIVE

Le client ne demande qu'au résolveur




Mais comment 1.1.1.1 trouve-t-il la réponse ?

# SERVEUR RACINE

Résolveur `1.1.1.1`

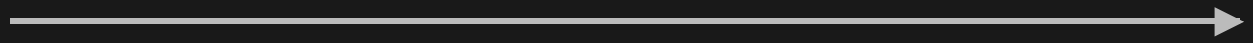
Racine `e.root-servers.net`

# SERVEUR RACINE

Résolveur `1.1.1.1`

Racine `e.root-servers.net`

`.net` ?

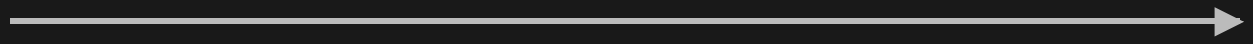


# SERVEUR RACINE

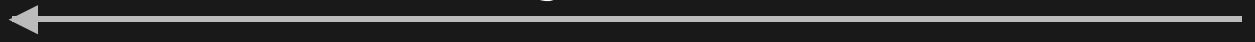
Résolveur `1.1.1.1`

Racine `e.root-servers.net`

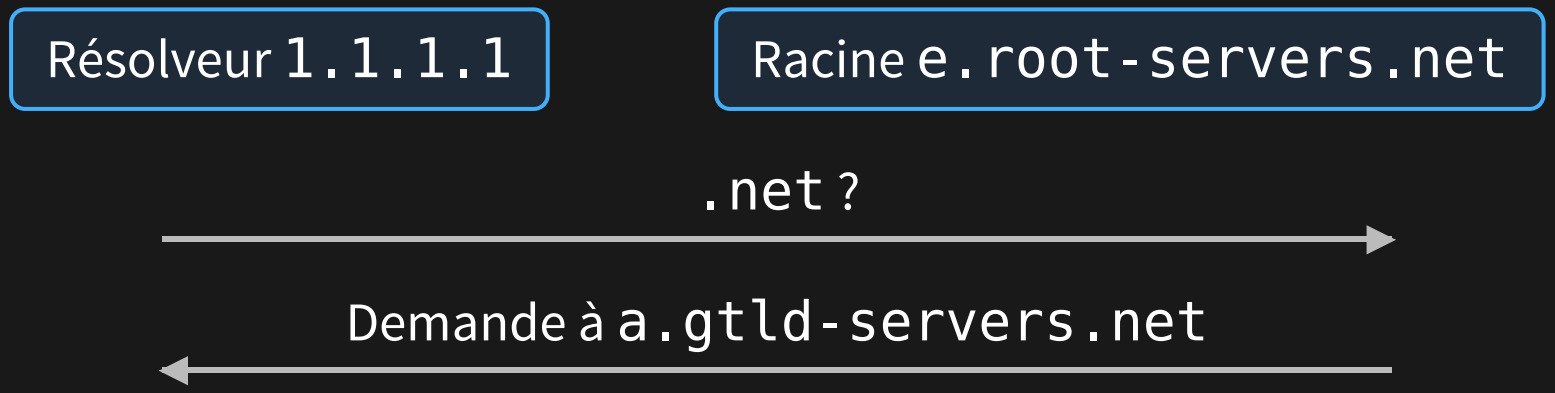
`.net` ?



Demande à `a.gtld-servers.net`

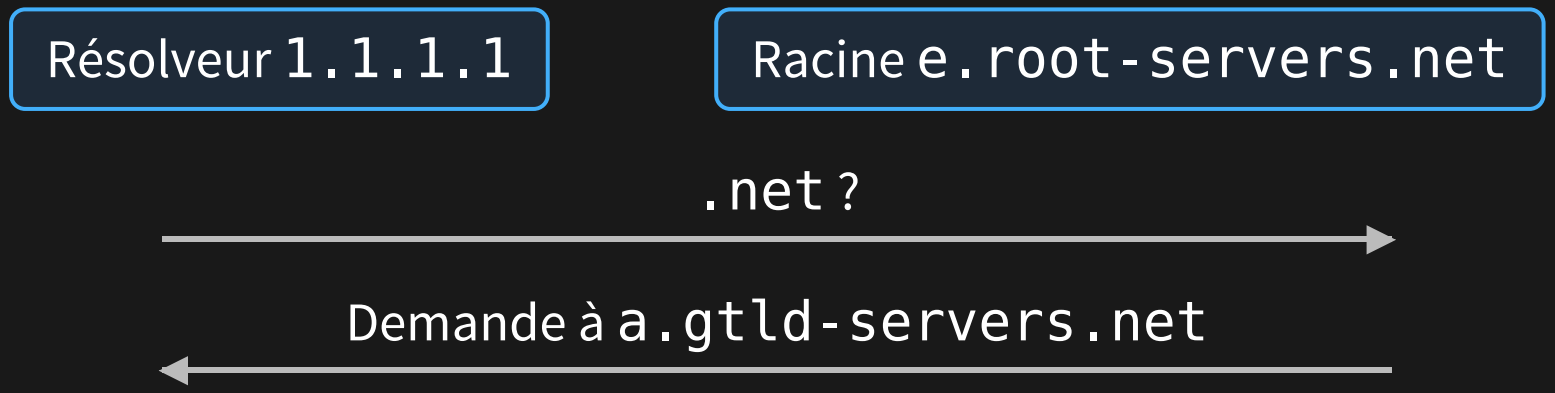


# SERVEUR RACINE




La racine ne connaît pas la réponse, mais sait qui gère `.net`

# SERVEUR RACINE




La racine ne connaît pas la réponse, mais sait qui gère `.net`

`a.gtld-servers.net` est lui-même en `.net`...

**Boucle ?**

# GLUE RECORD

Le serveur transmet un **nom** de serveur et non une IP

# GLUE RECORD

Le serveur transmet un **nom** de serveur et non une IP

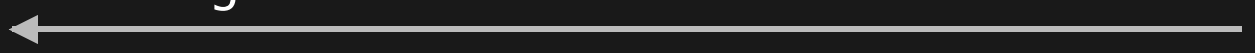
Pour éviter la boucle, il « colle » l'adresse IP au nom

# GLUE RECORD

Le serveur transmet un **nom** de serveur et non une IP

Pour éviter la boucle, il « colle » l'adresse IP au nom

Résolveur `1.1.1.1` Racine `e.root-servers.net`



# SERVEUR TLD

Résolveur `1.1.1.1`

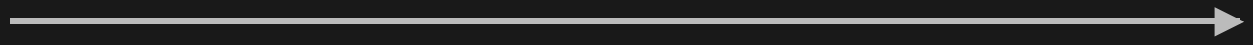
TLD `a.gtld-servers.net`

# SERVEUR TLD

Résolveur `1.1.1.1`

TLD `a.gtld-servers.net`

`minet.net ?`

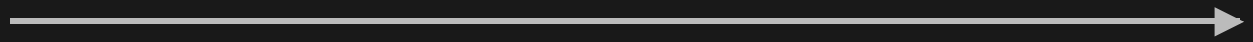


# SERVEUR TLD

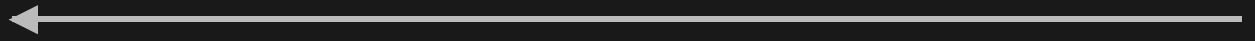
Résolveur `1.1.1.1`

TLD `a.gtld-servers.net`

`minet.net` ?



Demande à `ns1.minet.net`



# SERVEUR TLD

Résolveur `1.1.1.1`

TLD `a.gtld-servers.net`

`minet.net` ?

Demande à `ns1.minet.net`

Le TLD sait quel serveur fait **autorité** sur `minet.net`

# SERVEUR AUTORITAIRE

Résolveur `1.1.1.1`

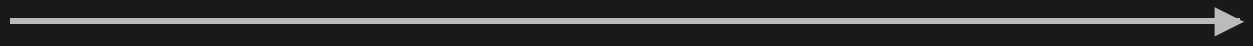
Autoritaire `ns1.minet.net`

# SERVEUR AUTORITAIRE

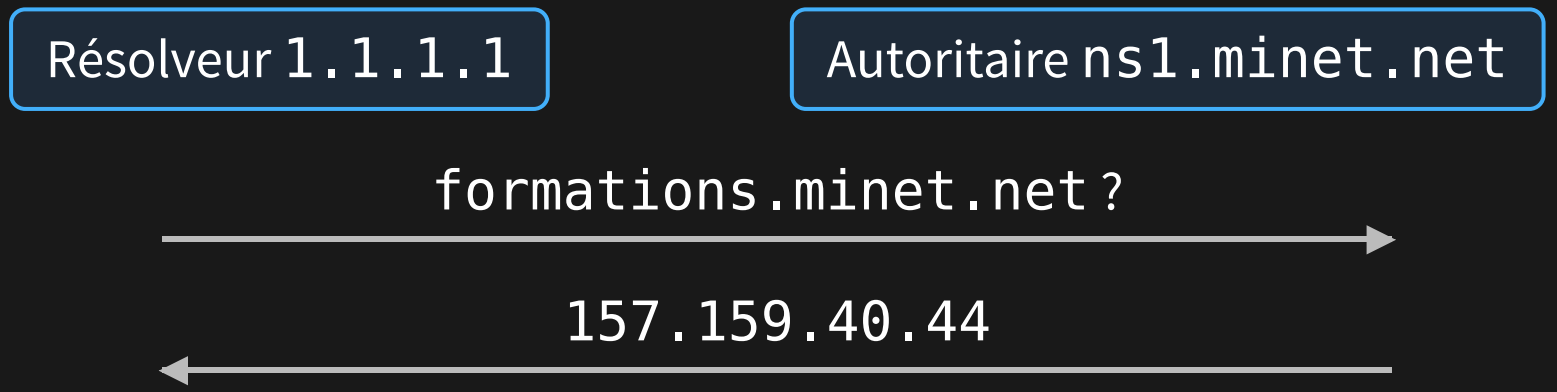
Résolveur `1.1.1.1`

Autoritaire `ns1.minet.net`

`formations.minet.net` ?



# SERVEUR AUTORITAIRE



# RÉSOLUTION RÉCURSIVE

| | Question | Serveur | Réponse |
|---|---|---|---|
| 1 | `formations.minet.net` | Résolveur<br>`1.1.1.1` | … |
| | | | |
| | | | |
| | | | |
| | | | |

# RÉSOLUTION RÉCURSIVE

| | Question | Serveur | Réponse |
|---|---|---|---|
| 1 | `formations.minet.net` | Résolveur `1.1.1.1` | … |
| 2 | `.net ?` | Racine | `a.gtld-servers.net` |
| | | | |
| | | | |
| | | | |

# RÉSOLUTION RÉCURSIVE

| | Question | Serveur | Réponse |
|---|---|---|---|
| 1 | `formations.minet.net` | Résolveur `1.1.1.1` | … |
| 2 | `.net ?` | Racine | `a.gtld-servers.net` |
| 3 | `minet.net ?` | TLD | `ns1.minet.net` |

# RÉSOLUTION RÉCURSIVE

| | Question | Serveur | Réponse |
|---|---|---|---|
| 1 | `formations.minet.net` | Résolveur `1.1.1.1` | … |
| 2 | `.net ?` | Racine | `a.gtld-servers.net` |
| 3 | `minet.net ?` | TLD | `ns1.minet.net` |
| 4 | `formations.minet.net` | Autoritaire | `157.159.40.44` |

# RÉSOLUTION RÉCURSIVE

| | Question | Serveur | Réponse |
|---|---|---|---|
| 1 | `formations.minet.net` | Résolveur `1.1.1.1` | … |
| 2 | `.net ?` | Racine | `a.gtld-servers.net` |
| 3 | `minet.net ?` | TLD | `ns1.minet.net` |
| 4 | `formations.minet.net` | Autoritaire | `157.159.40.44` |
| 5 | Le résolveur répond au client | | `157.159.40.44` |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net` → `157.159.40.44` |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net → 157.159.40.44` |
| AAAA | nom → IPv6 | |
| | | |
| | | |
| | | |
| | | |
| | | |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net → 157.159.40.44` |
| AAAA | nom → IPv6 | |
| CNAME | nom → autre nom (alias) | |
| | | |
| | | |
| | | |
| | | |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net → 157.159.40.44` |
| AAAA | nom → IPv6 | |
| CNAME | nom → autre nom (alias) | |
| NS | serveur qui fait autorité sur la zone | `minet.net → ns1.minet.net` |
| | | |
| | | |
| | | |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net → 157.159.40.44` |
| AAAA | nom → IPv6 | |
| CNAME | nom → autre nom (alias) | |
| NS | serveur qui fait autorité sur la zone | `minet.net → ns1.minet.net` |
| MX | serveur mail du domaine | |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net` → `157.159.40.44` |
| AAAA | nom → IPv6 | |
| CNAME | nom → autre nom (alias) | |
| NS | serveur qui fait autorité sur la zone | `minet.net` → `ns1.minet.net` |
| MX | serveur mail du domaine | |
| TXT | texte libre (SPF, vérifications…) | |

# LES ENREGISTREMENTS

| | | |
|---|---|---|
| A | nom → IPv4 | `minet.net → 157.159.40.44` |
| AAAA | nom → IPv6 | |
| CNAME | nom → autre nom (alias) | |
| NS | serveur qui fait autorité sur la zone | `minet.net → ns1.minet.net` |
| MX | serveur mail du domaine | |
| TXT | texte libre (SPF, vérifications…) | |
| PTR | IP → nom (DNS inverse) | |

# EN PRATIQUE

```
dig minet.net
dig minet.net AAAA
dig @1.1.1.1 minet.net
dig +trace formations.minet.net
dig -x 157.159.40.44
```

# EN PRATIQUE

```
dig minet.net
dig minet.net AAAA
dig @1.1.1.1 minet.net
dig +trace formations.minet.net
dig -x 157.159.40.44
```

`+trace` : refait la résolution étape par étape, depuis la racine

# REVERSE PROXY

Exemple : nginx

# LE PROBLÈME

MiNET a **une seule** IP publique : `157.159.40.44`

# LE PROBLÈME

MiNET a **une seule** IP publique : `157.159.40.44`

`minet.net` `wiki.minet.net` `grafana.minet.net`

# LE PROBLÈME

MiNET a **une seule** IP publique : `157.159.40.44`

`minet.net` `wiki.minet.net` `grafana.minet.net`

…et plein de services, chacun sur sa machine

# LE PROBLÈME

MiNET a **une seule** IP publique : `157.159.40.44`

`minet.net` `wiki.minet.net` `grafana.minet.net`

…et plein de services, chacun sur sa machine

Mais un seul port 443 (HTTPS) sur cette IP.

**Qui répond ?**

# PROXY OU REVERSE PROXY ?

**Proxy** (forward) : parle au nom des **clients**



# PROXY OU REVERSE PROXY ?

**Proxy** (forward) : parle au nom des **clients**




**Reverse proxy** : parle au nom des **serveurs**



# PROXY OU REVERSE PROXY ?

**Proxy** (forward) : parle au nom des **clients**

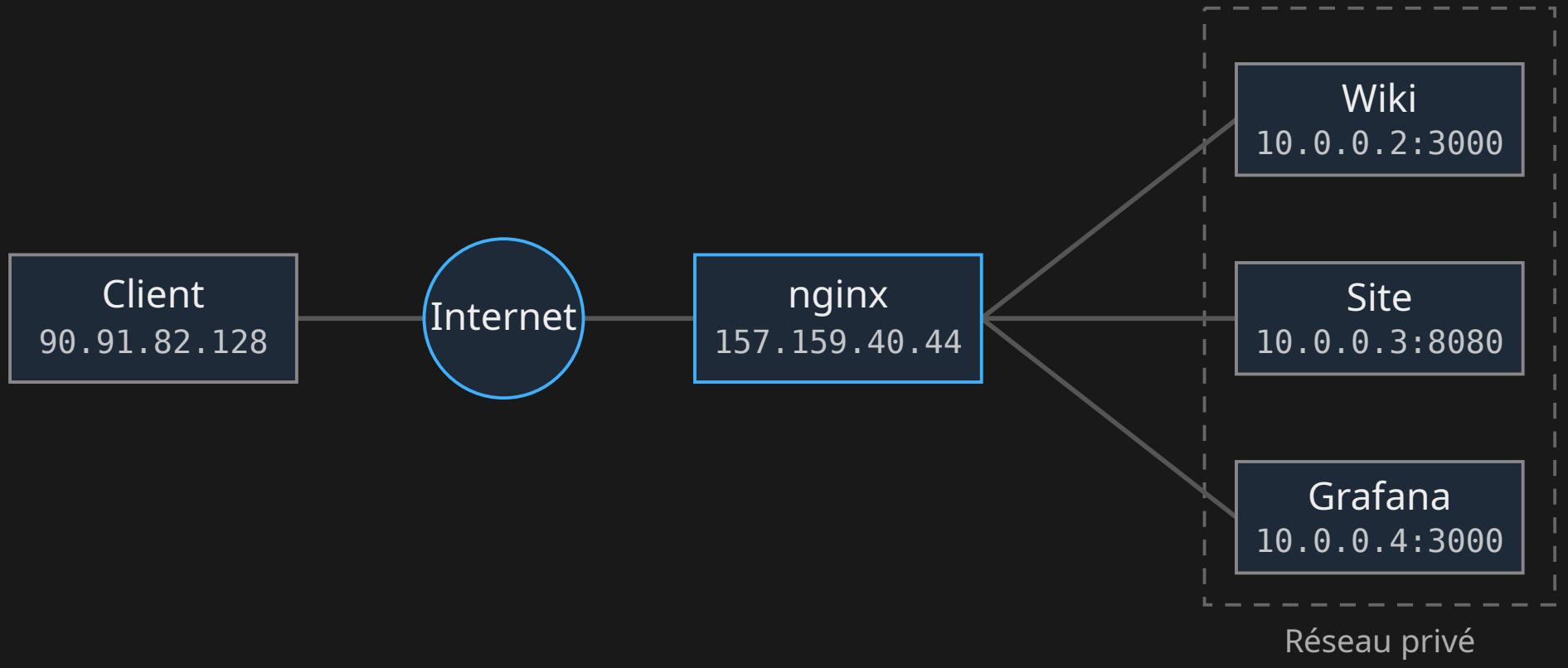



**Reverse proxy** : parle au nom des **serveurs**

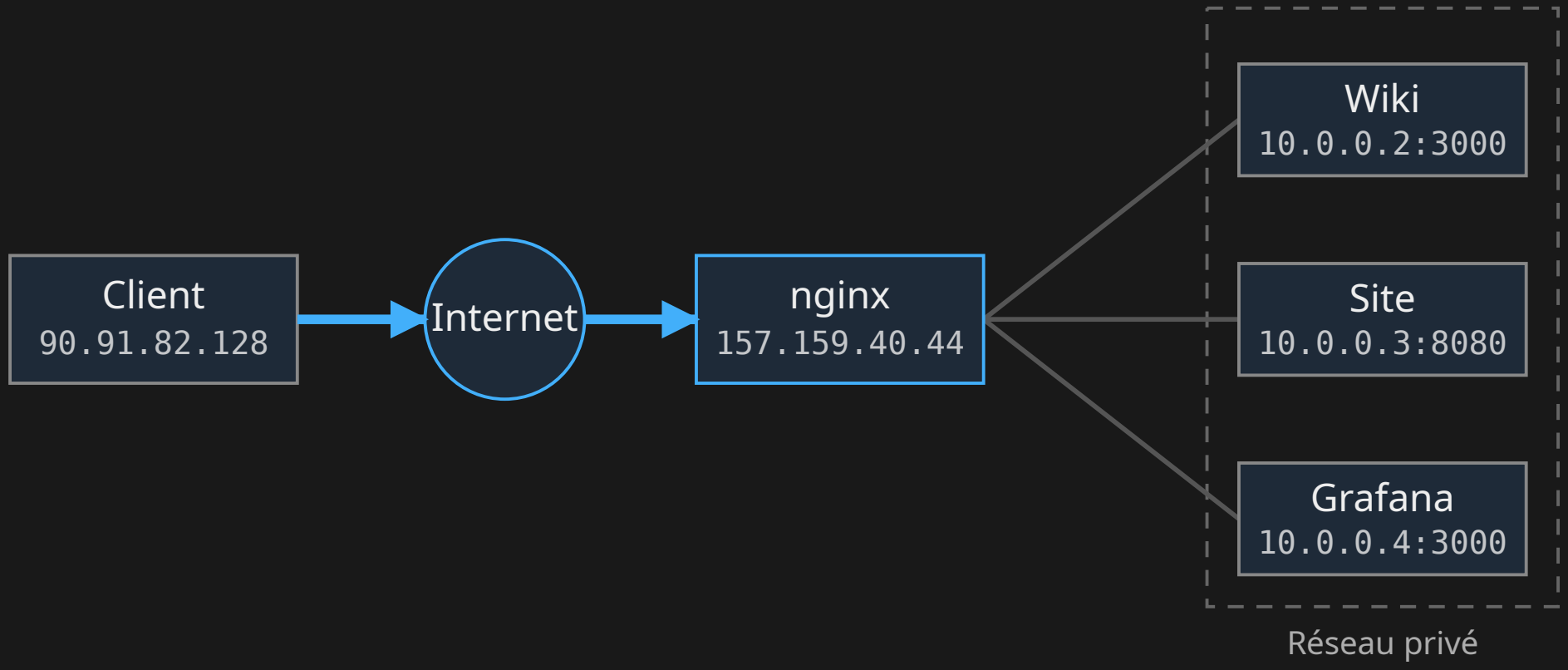



Le client croit parler directement au site, il ne voit que le reverse proxy

# AVEC UN REVERSE PROXY

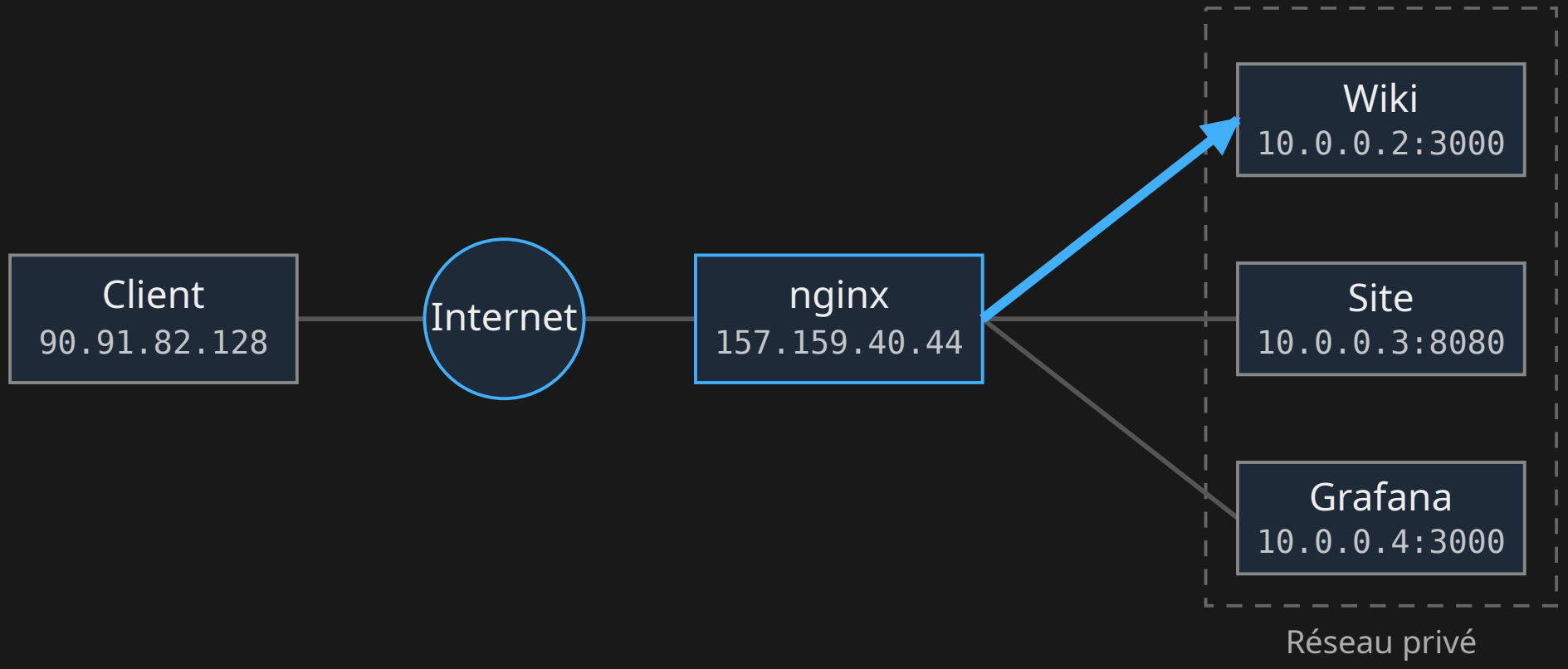


# AVEC UN REVERSE PROXY



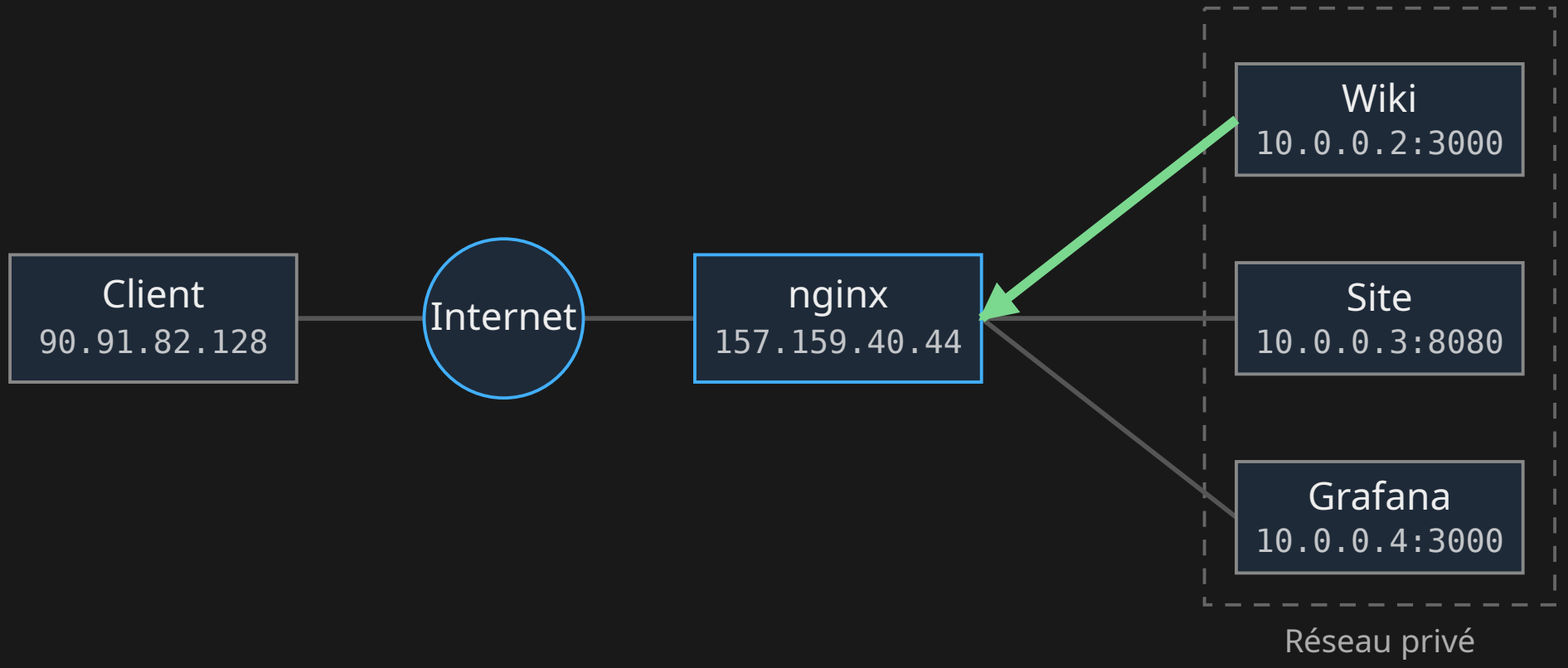

Client → nginx : « Montre-moi wiki.minet.net »

# AVEC UN REVERSE PROXY



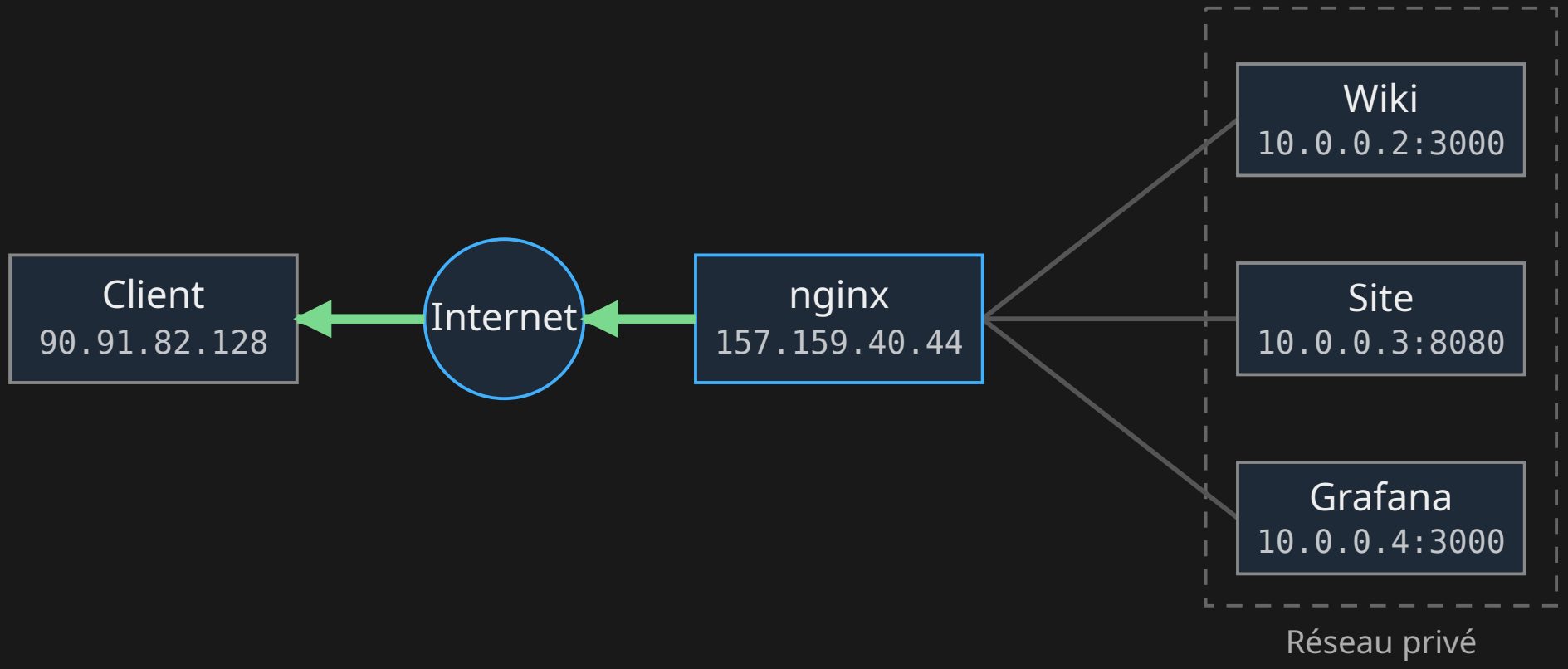

nginx : « wiki ? C'est 10.0.0.2:3000 » et transmet

# AVEC UN REVERSE PROXY




Wiki → nginx : « Voilà la page »

# AVEC UN REVERSE PROXY




nginx → Client : « Voilà la page » ✓

# COMMENT CHOISIR LE BON SERVEUR ?

Toutes les requêtes arrivent sur la même IP… mais le client dit **quel site** il veut

```
1 GET / HTTP/1.1
2 Host: wiki.minet.net
3 User-Agent: curl/8.11.0
```

# COMMENT CHOISIR LE BON SERVEUR ?

Toutes les requêtes arrivent sur la même IP… mais le client dit **quel site** il veut

```
1 GET / HTTP/1.1
2 Host: wiki.minet.net
3 User-Agent: curl/8.11.0
```

nginx lit l'en-tête `Host` et choisit le serveur correspondant

# COMMENT CHOISIR LE BON SERVEUR ?

Toutes les requêtes arrivent sur la même IP… mais le client dit **quel site** il veut

```
GET / HTTP/1.1
Host: wiki.minet.net
User-Agent: curl/8.11.0
```

nginx lit l'en-tête `Host` et choisit le serveur correspondant

En HTTPS, le nom est aussi envoyé dès la connexion TLS (**SNI**) pour choisir le bon certificat

# DEUX CONNEXIONS

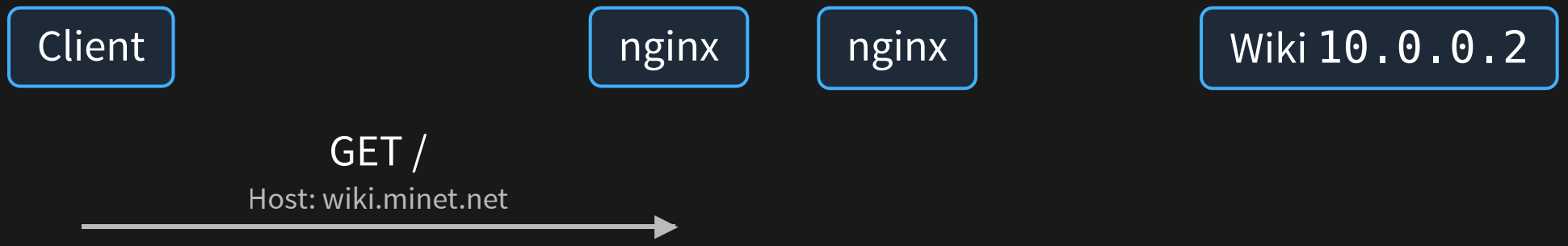
Client

nginx

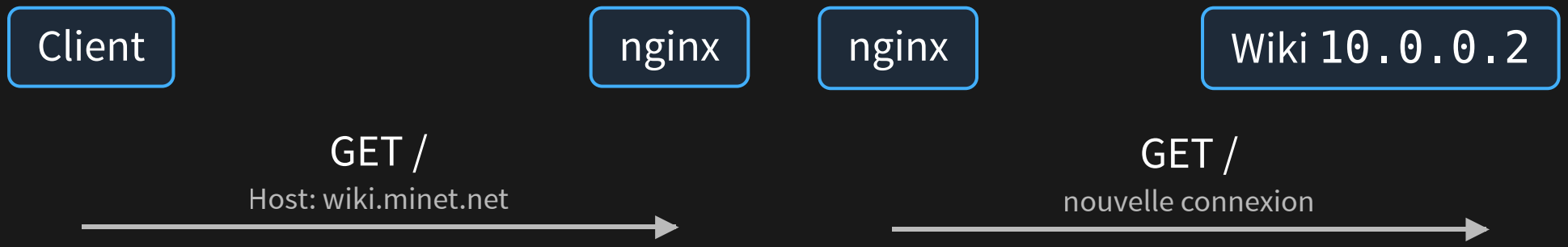
nginx

Wiki `10.0.0.2`

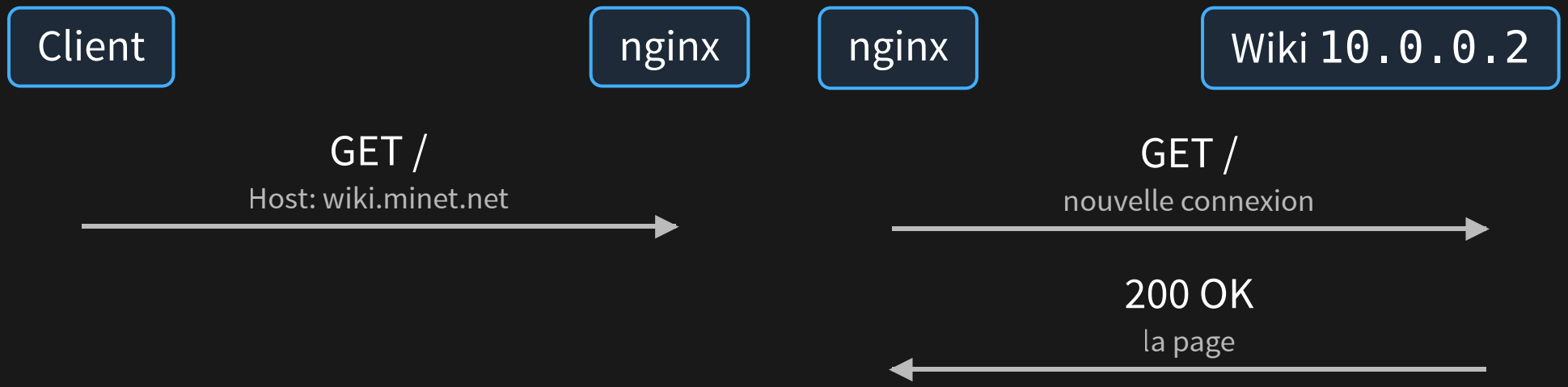
# DEUX CONNEXIONS



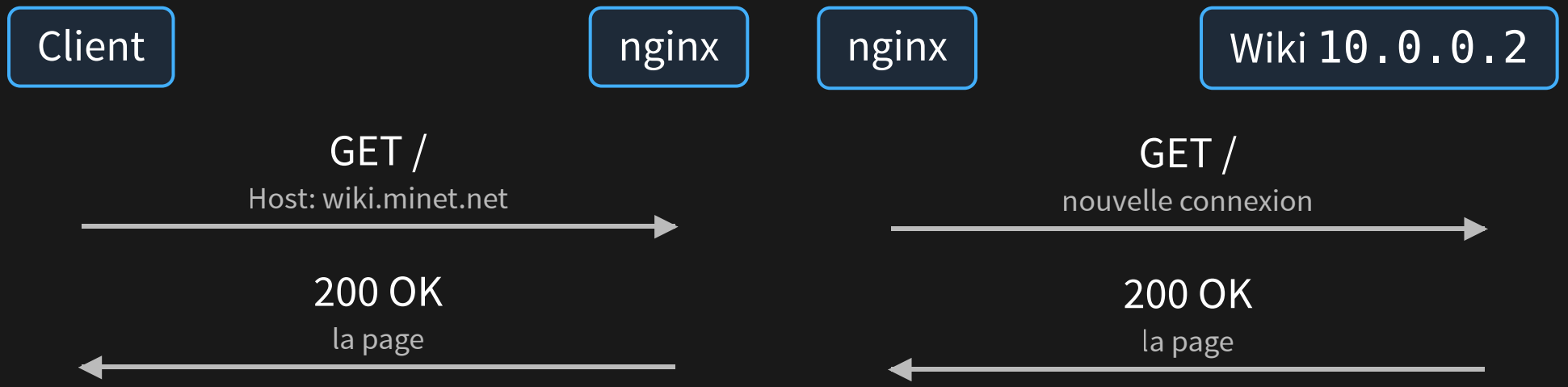
# DEUX CONNEXIONS



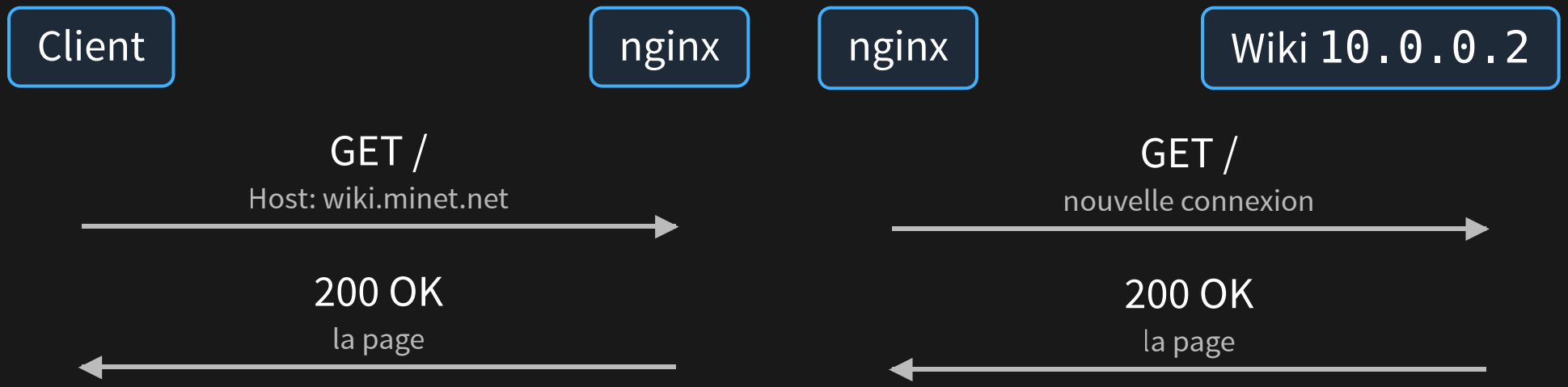
# DEUX CONNEXIONS



# DEUX CONNEXIONS



# DEUX CONNEXIONS




Le wiki ne voit jamais le client : il ne parle qu'à **nginx**

# NGINX

Serveur web **et** reverse proxy, très utilisé (dont chez MiNET)

```
server {
    listen 80;
    server_name wiki.minet.net;

    location / {
        proxy_pass http://10.0.0.2:3000;
    }
}
```

# NGINX

Serveur web **et** reverse proxy, très utilisé (dont chez MiNET)

```
server {
    listen 80;
    server_name wiki.minet.net;

    location / {
        proxy_pass http://10.0.0.2:3000;
    }
}
```

| | |
|---|---|
| `listen` | port sur lequel nginx écoute |
| | |

# NGINX

Serveur web **et** reverse proxy, très utilisé (dont chez MiNET)

```
server {
    listen 80;
    server_name wiki.minet.net;

    location / {
        proxy_pass http://10.0.0.2:3000;
    }
}
```

| | |
|---|---|
| `listen` | port sur lequel nginx écoute |
| `server_name` | comparé à l'en-tête `Host` |

# NGINX

Serveur web **et** reverse proxy, très utilisé (dont chez MiNET)

```
server {
    listen 80;
    server_name wiki.minet.net;

    location / {
        proxy_pass http://10.0.0.2:3000;
    }
}
```

| | |
|---|---|
| `listen` | port sur lequel nginx écoute |
| `server_name` | comparé à l'en-tête `Host` |
| `proxy_pass` | où transmettre la requête |

# PLUSIEURS SITES

Un bloc `server` par site, nginx choisit grâce à `server_name`

| `server_name` | `proxy_pass` |
| --- | --- |
| | |
| | |

# PLUSIEURS SITES

Un bloc `server` par site, nginx choisit grâce à `server_name`

| server_name | | proxy_pass |
|---|---|---|
| minet.net | → | http://10.0.0.3:8080 |
| | | |

# PLUSIEURS SITES

Un bloc `server` par site, nginx choisit grâce à `server_name`

| server_name | | proxy_pass |
|---|---|---|
| minet.net | → | http://10.0.0.3:8080 |
| wiki.minet.net | → | http://10.0.0.2:3000 |

# PLUSIEURS SITES

Un bloc `server` par site, nginx choisit grâce à `server_name`

| server_name | | proxy_pass |
|---|---|---|
| minet.net | → | http://10.0.0.3:8080 |
| wiki.minet.net | → | http://10.0.0.2:3000 |
| grafana.minet.net | → | http://10.0.0.4:3000 |

# PLUSIEURS SITES

Un bloc `server` par site, nginx choisit grâce à `server_name`

| **server_name** | | **proxy_pass** |
|---|---|---|
| `minet.net` | → | `http://10.0.0.3:8080` |
| `wiki.minet.net` | → | `http://10.0.0.2:3000` |
| `grafana.minet.net` | → | `http://10.0.0.4:3000` |

Côté DNS : les trois noms pointent vers la **même IP**, celle de nginx

# PLUSIEURS SITES

Un bloc `server` par site, nginx choisit grâce à `server_name`

| server_name | | proxy_pass |
|---|---|---|
| minet.net | → | http://10.0.0.3:8080 |
| wiki.minet.net | → | http://10.0.0.2:3000 |
| grafana.minet.net | → | http://10.0.0.4:3000 |

Côté DNS : les trois noms pointent vers la **même IP**, celle de nginx

Un fichier par site dans `sites-available/`, activé par un lien dans `sites-enabled/`

# QUI EST LE CLIENT ?



# QUI EST LE CLIENT ?




Le wiki pense que toutes les requêtes viennent de nginx. **Logs inutiles, bans impossibles**

# QUI EST LE CLIENT ?




Le wiki pense que toutes les requêtes viennent de nginx. **Logs inutiles, bans impossibles**

```
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
```

# QUI EST LE CLIENT ?




Le wiki pense que toutes les requêtes viennent de nginx. **Logs inutiles, bans impossibles**

```
proxy_set_header Host $host;
proxy_set_header X-Real-IP $remote_addr;
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header X-Forwarded-Proto $scheme;
```

nginx ajoute des en-têtes pour transmettre la vraie IP et le protocole d'origine

# HTTPS : TERMINAISON TLS



# HTTPS : TERMINAISON TLS




nginx déchiffre, le serveur derrière reçoit du HTTP simple

# HTTPS : TERMINAISON TLS




nginx déchiffre, le serveur derrière reçoit du HTTP simple

```
listen 443 ssl;
ssl_certificate     /etc/ssl/wiki/fullchain.pem;
ssl_certificate_key /etc/ssl/wiki/privkey.pem;
```

# HTTPS : TERMINAISON TLS




nginx déchiffre, le serveur derrière reçoit du HTTP simple

```
listen 443 ssl;
ssl_certificate     /etc/ssl/wiki/fullchain.pem;
ssl_certificate_key /etc/ssl/wiki/privkey.pem;
```

Les certificats sont gérés à **un seul endroit** (Let's Encrypt + `certbot`)

# RÉPARTITION DE CHARGE

Un même service sur plusieurs machines

```
upstream wiki {
    server 10.0.0.2:3000;
    server 10.0.0.5:3000;
}
# ...
proxy_pass http://wiki;
```

# RÉPARTITION DE CHARGE

Un même service sur plusieurs machines

```
upstream wiki {
    server 10.0.0.2:3000;
    server 10.0.0.5:3000;
}
# ...
proxy_pass http://wiki;
```

Par défaut, chacun son tour (*round-robin*)

# RÉPARTITION DE CHARGE

Un même service sur plusieurs machines

```
upstream wiki {
    server 10.0.0.2:3000;
    server 10.0.0.5:3000;
}
# ...
proxy_pass http://wiki;
```

Par défaut, chacun son tour (*round-robin*)

Si un serveur ne répond plus, nginx l'écarte temporairement

# CE QUE ÇA APPORTE

# CE QUE ÇA APPORTE

| | |
|---|---|
| **Un seul point d'entrée** | plein de services derrière une IP et un port |
| | |
| | |
| | |
| | |
| | |

# CE QUE ÇA APPORTE

| | |
|---|---|
| **Un seul point d'entrée** | plein de services derrière une IP et un port |
| **HTTPS centralisé** | les certificats au même endroit |
| | |
| | |
| | |
| | |

# CE QUE ÇA APPORTE

| | |
|---|---|
| **Un seul point d'entrée** | plein de services derrière une IP et un port |
| **HTTPS centralisé** | les certificats au même endroit |
| **Sécurité** | les serveurs restent dans le réseau privé |
| | |
| | |

# CE QUE ÇA APPORTE

| | |
|---|---|
| **Un seul point d'entrée** | plein de services derrière une IP et un port |
| **HTTPS centralisé** | les certificats au même endroit |
| **Sécurité** | les serveurs restent dans le réseau privé |
| **Répartition de charge** | plusieurs serveurs pour un même site |

# CE QUE ÇA APPORTE

| | |
|---|---|
| **Un seul point d'entrée** | plein de services derrière une IP et un port |
| **HTTPS centralisé** | les certificats au même endroit |
| **Sécurité** | les serveurs restent dans le réseau privé |
| **Répartition de charge** | plusieurs serveurs pour un même site |
| **Cache, compression** | soulage les serveurs derrière |

# CE QUE ÇA APPORTE

| | |
|---|---|
| **Un seul point d'entrée** | plein de services derrière une IP et un port |
| **HTTPS centralisé** | les certificats au même endroit |
| **Sécurité** | les serveurs restent dans le réseau privé |
| **Répartition de charge** | plusieurs serveurs pour un même site |
| **Cache, compression** | soulage les serveurs derrière |
| **Logs** | tout le trafic web passe au même endroit |

BLOOKET

# TP

## DHCP



https://wiki.minet.net/fr/tps/dev/dhcp

## DNS



https://wiki.minet.net/fr/tps/dev/dns

## REVPROXY



https://wiki.minet.net/fr/tps/tp-formations/TP-RevProxy