

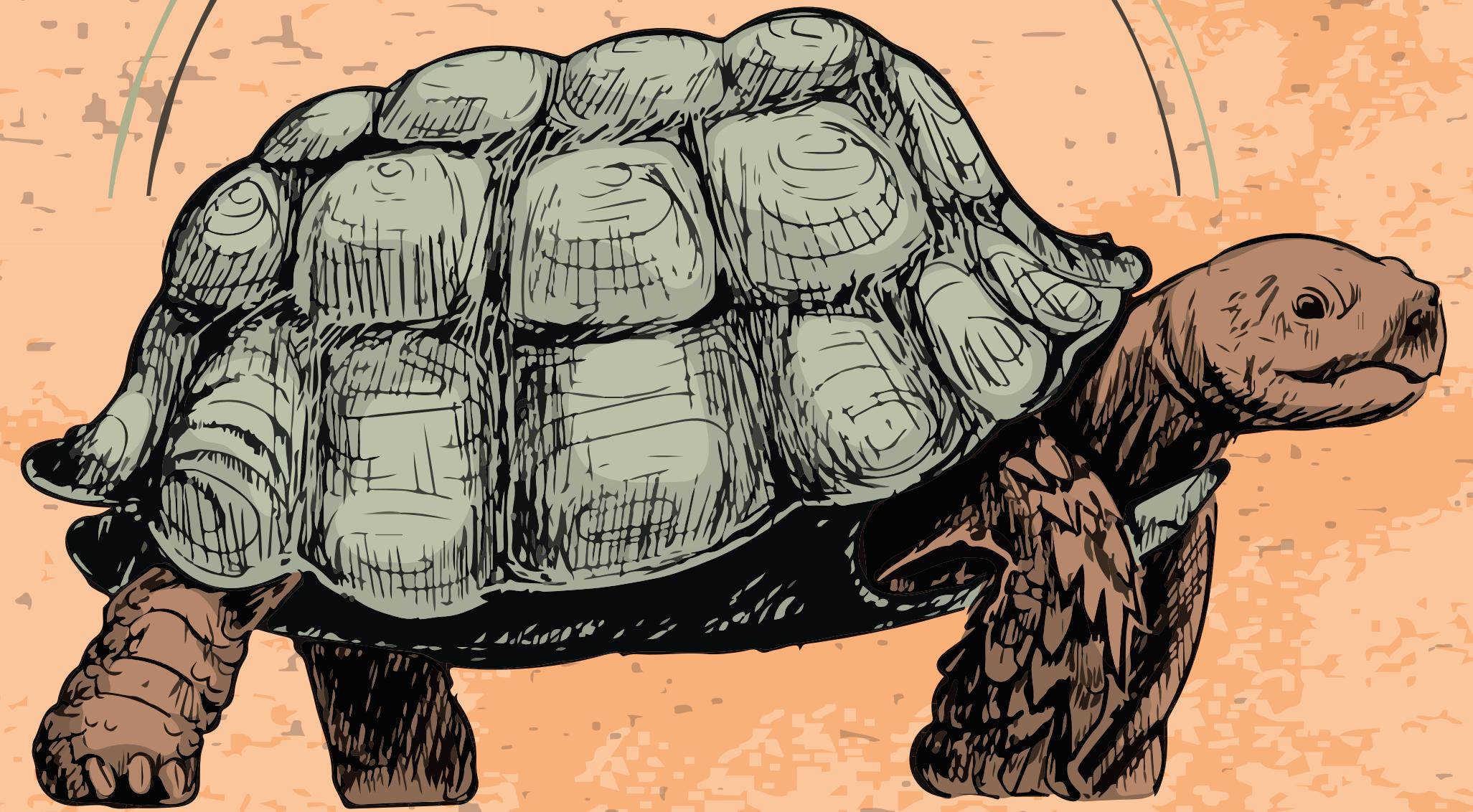
# ACCÉLÉREZ VOS API AVEC VARNISH CACHE



PAR THIJS FERYN



ConFoo.ca  
WEB TECHNO CONFERENCE



*Les sites lents sont*  
**CHIANTS**

**LA VITESSE DE VOTRE SITE FAIT  
PARTI DE L'EXPERIENCE  
D'UTILISATEUR**



**LENT~  
INDISPONIBLE**







Google





**AJOUTER DES  
SERVEURS**

**MO' MONEY  
MO' SERVERS  
MO' PROBLEMS**



# ***IDENTIFIER LES DÉLAIS***



# OPTIMISER

# **ATTEINDRE LES LIMITES**



# CACHING



**JE M'APPELLE THIJS**





**JE SUIS  
L'ÉVANGELISTE  
TECHNIQUE  
CHEZ**



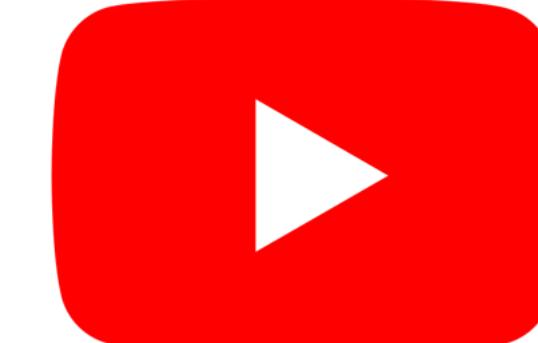
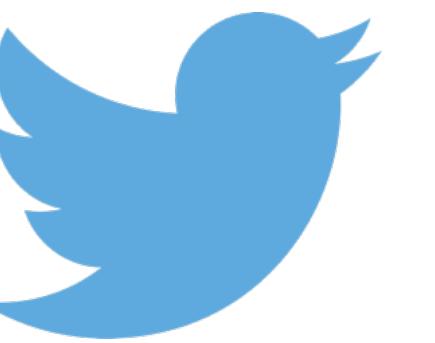
**VARNISH  
SOFTWARE**

The text is arranged in several lines. 'JE SUIS' is at the top in large bold black letters. Below it is 'L'ÉVANGELISTE' in a slightly smaller bold black font. Underneath that is 'TECHNIQUE' in a bold black font. To the left of 'TECHNIQUE' is the word 'CHEZ' in a bold black font. To the right of 'CHEZ' is the Varnish Software logo, which consists of three blue circles of increasing size from bottom-left to top-right, followed by the company name 'VARNISH SOFTWARE' in a bold black sans-serif font.

**JE SUIS**



**@THIJSFERYN**



O'REILLY®



# Getting Started with Varnish Cache

ACCELERATE YOUR WEB APPLICATIONS

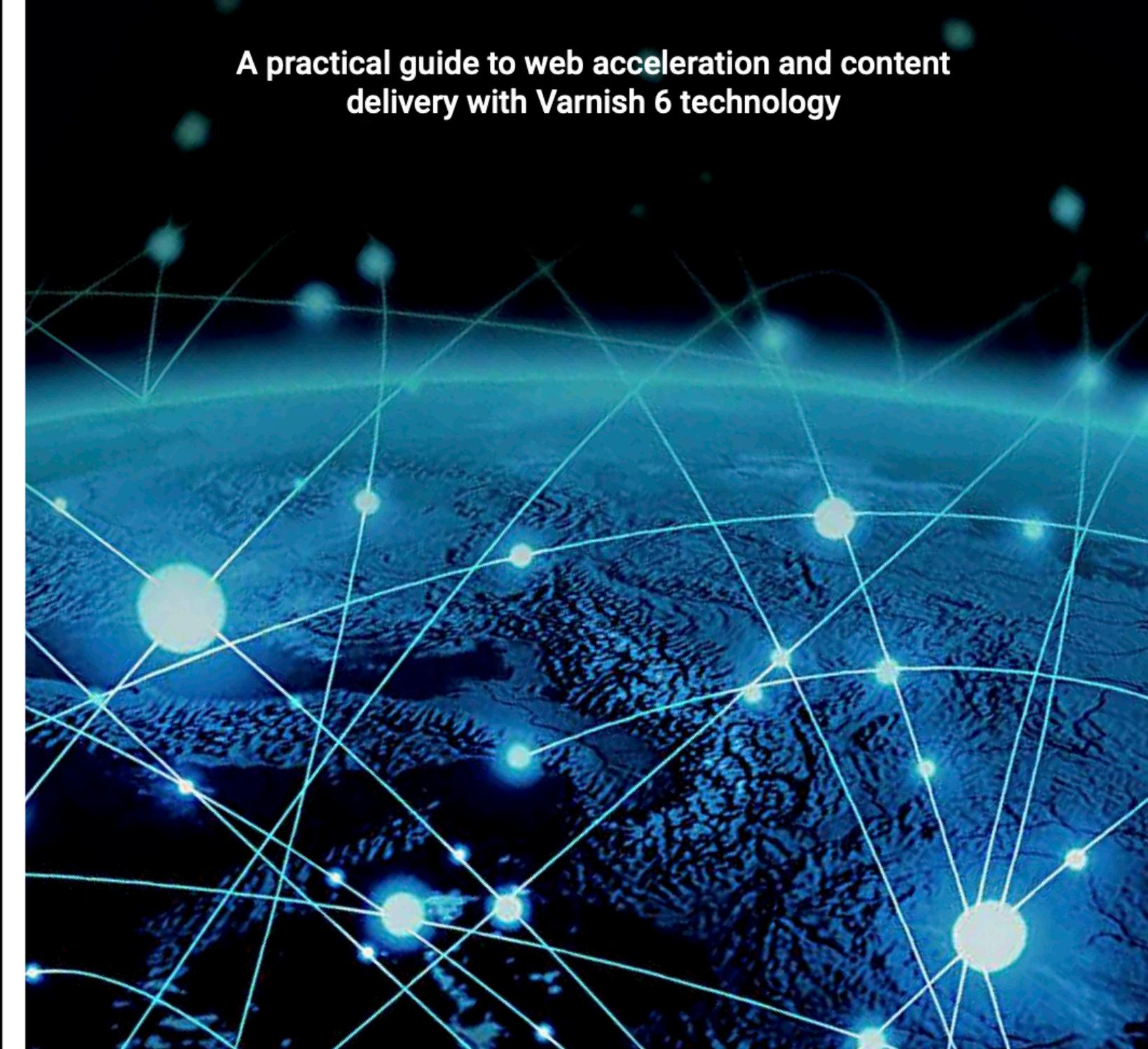
Thijs Feryn

Thijs Feryn

# VARNISH 6

BY EXAMPLE

A practical guide to web acceleration and content delivery with Varnish 6 technology





**NOUS CRÉONS DES SOLUTIONS  
D'ACCÉLÉRATION WEB & DE  
DIFFUSION DE CONTENU  
DÉFINIÉS PAR LOGICIEL**

# About Varnish Software

We deliver enterprise solutions based on open source **Varnish Cache**



**10M**



**22%**



**10M**

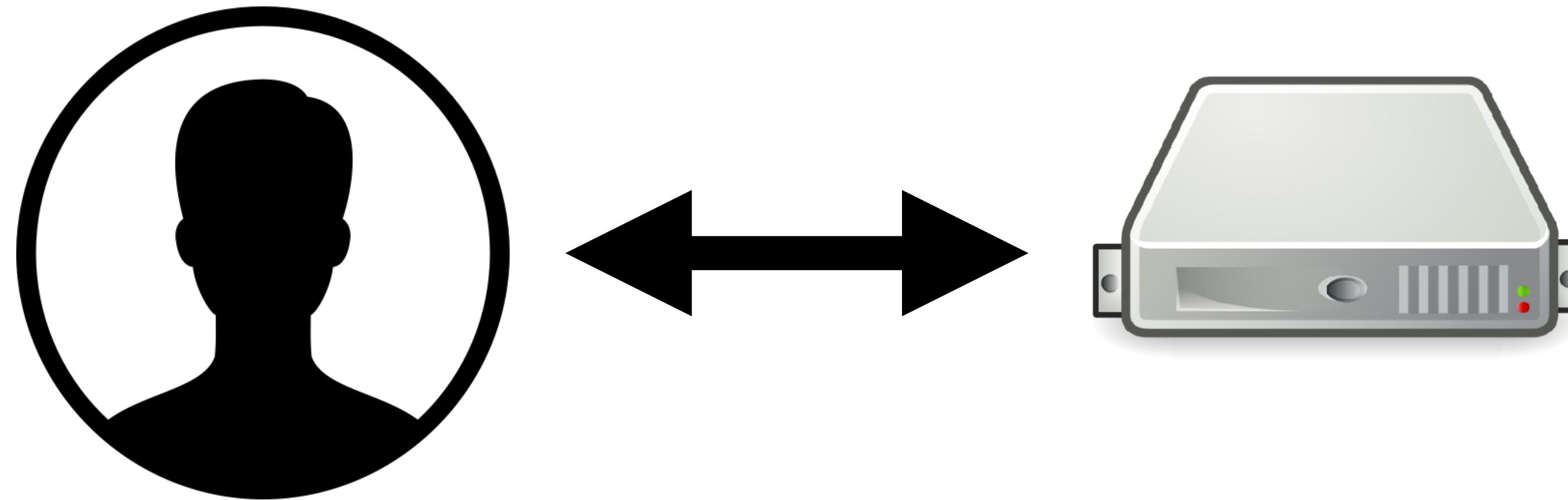
**10M** active websites worldwide  
powered by **Varnish**

**22%** of the world's top 10,000  
sites use **Varnish**

**10M** pulls on Docker hub of the  
**Varnish** image

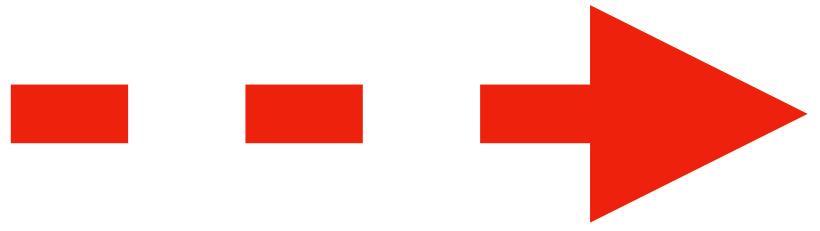
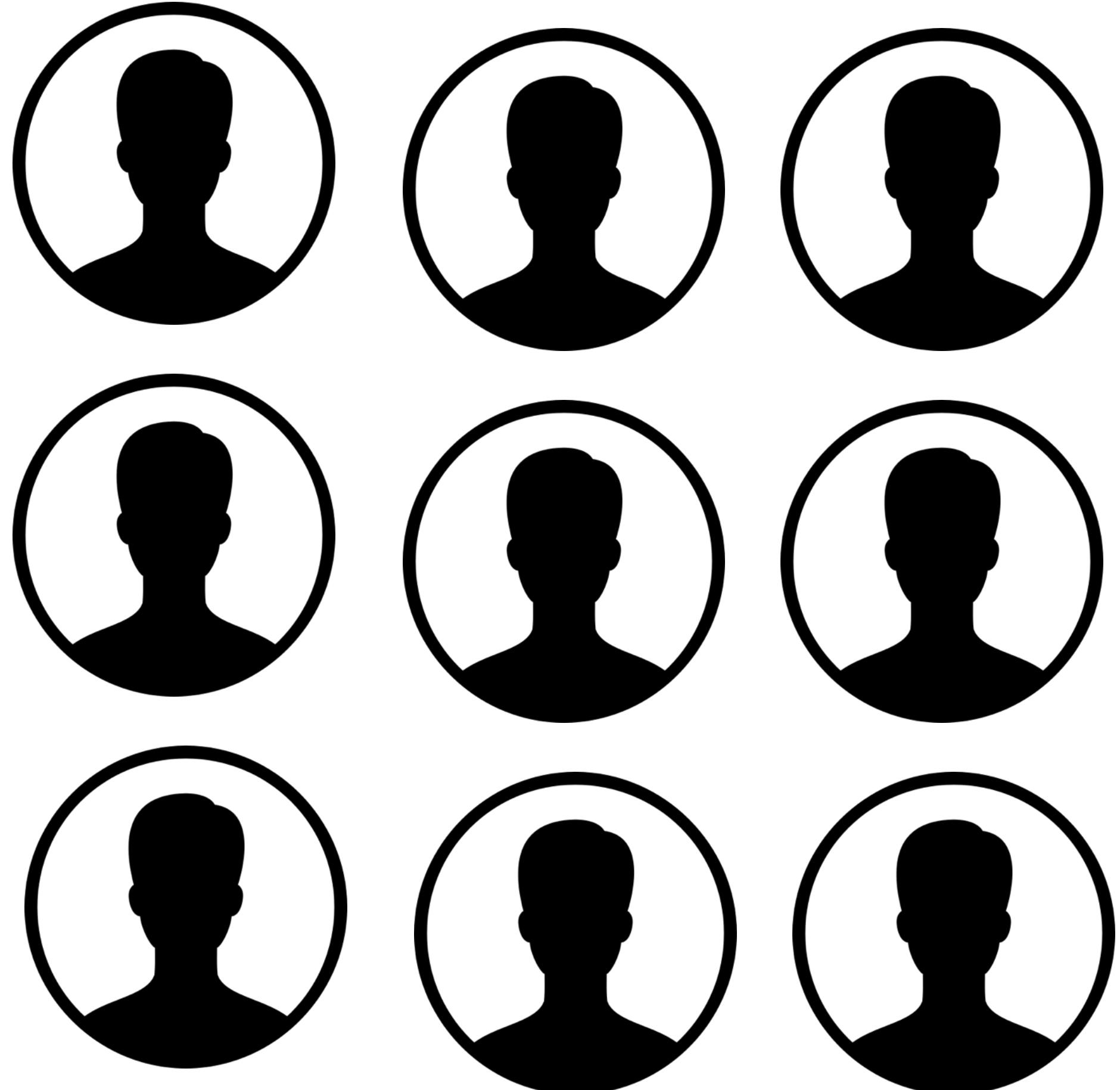


# **NORMALEMENT**



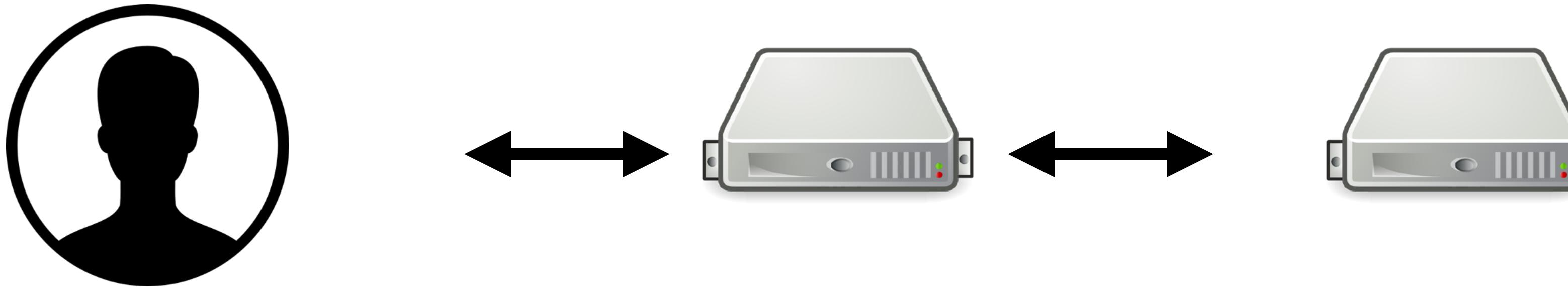
**UTILISATEUR      SERVEUR**

# **SOUS PRESSION**



**SERVEUR**

# **REVERSE CACHING PROXY**



**UTILISATEUR    VARNISH    SERVEUR**



**http://**



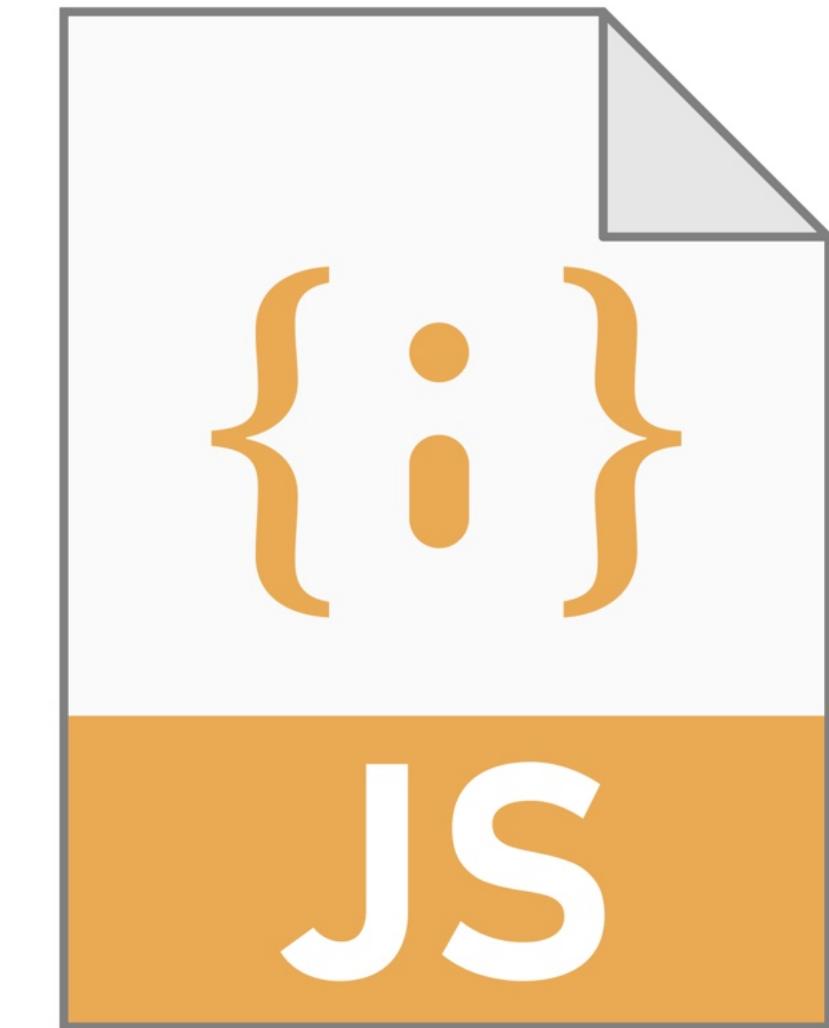
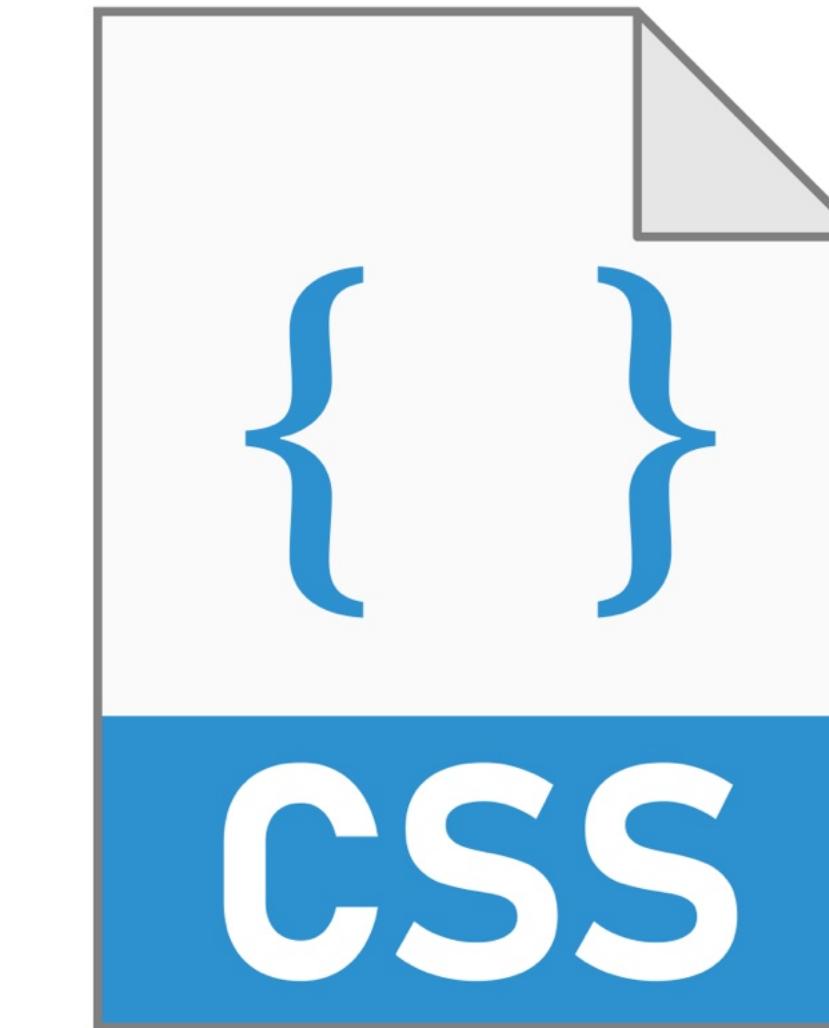
***HTTP À DES MÉCANISMES DE  
MISE EN CACHE INTÉGRÉS***

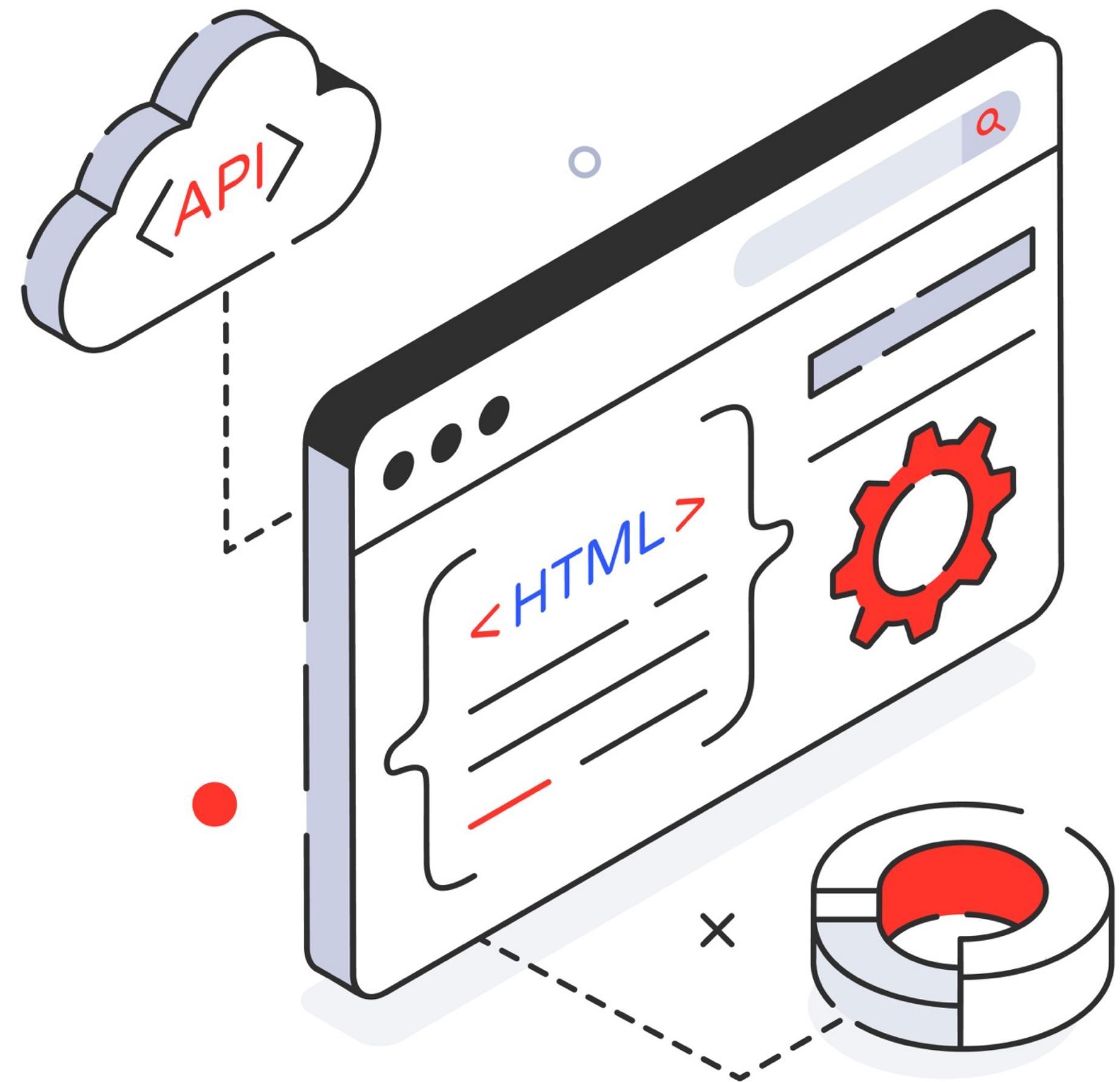
Cache-Control: public, max-age=500

**Cache-Control: private, no-cache, no-store**

***INVENTÉ POUR  
ACCÉLÉRER LES  
SITES***

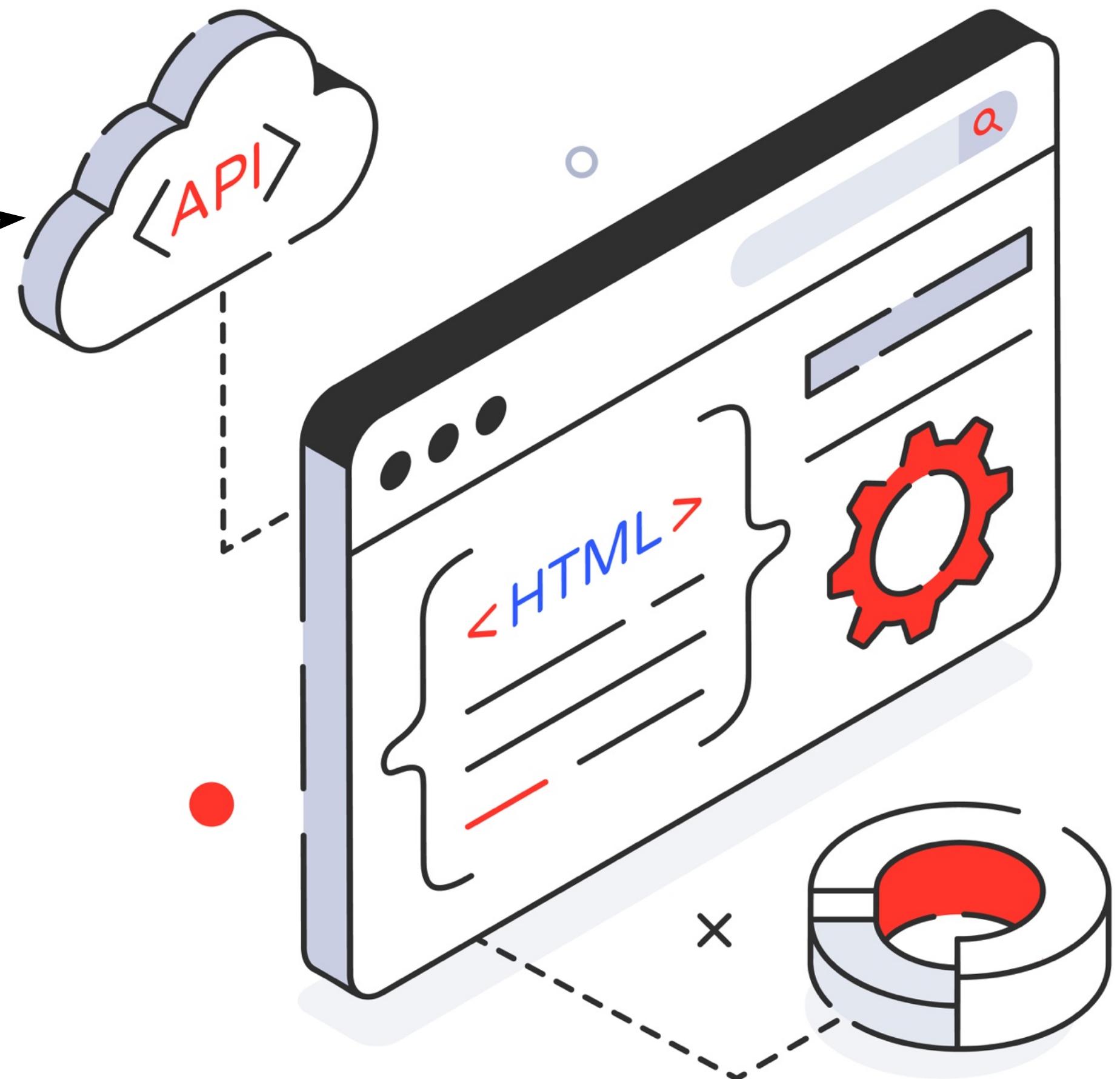
C'EST  
FACILE POUR LES  
METTRE IN CACHE





**LES ARCHITECTURES WEB SONT DEVENUES PLUS COMPLEXES**

**LES  
APIS ONT  
AUSSI BESOIN  
DE CACHING**



# **USAGE API**

- ✓ REQUÊTES AJAX EN HTML
- ✓ TRANSMETTRE DES DONNÉES AUX APPAREILS MOBILES
- ✓ ENVELOPPÉ PAR DES OUTILS CLI
- ✓ MICROSERVICES ET COMMUNICATION MACHINE A MACHINE



**VARNISH S'EN FOUT, CAR POUR  
VARNISH TOUT EST HTTP**

**LES API REST SONT PLUS FACILES À  
METTRE EN CACHE CAR ELLES SONT  
PLUS EXPRESSIVES EN TERMES DE HTTP**

# **APIS REST & MISE EN CACHE HTTP**

- ✓ **LES MÉTHODES DE REQUÊTE DÉCRIVENT L'ACTION.**  
(SEULEMENT GET & HEAD SONT MIS EN CACHE)
- ✓ **RESOURCES IDENTIFIÉES PAR L'URL.**  
(URL & ENTÊTE HOST)
- ✓ **LA NÉGOCIATION DE CONTENU PEUT EXPLOITER L'ENTÊTE "VARY" POUR CRÉER DES VARIATIONS DE CACHE**  
(ACCEPT, ACCEPT-LANGUAGE, ACCEPT-ENCODING)
- ✓ **LES ENTÊTES "CACHE-CONTROL" SONT RESPECTÉS**
- ✓ **LES DEMANDES CONDITIONNELLES SONT RESPECTÉES**  
(ETAG/LAST-MODIFIED)

VARNISH A UN  
STANDARD



COMPORTEMENT  
TRÈS PRUDENT

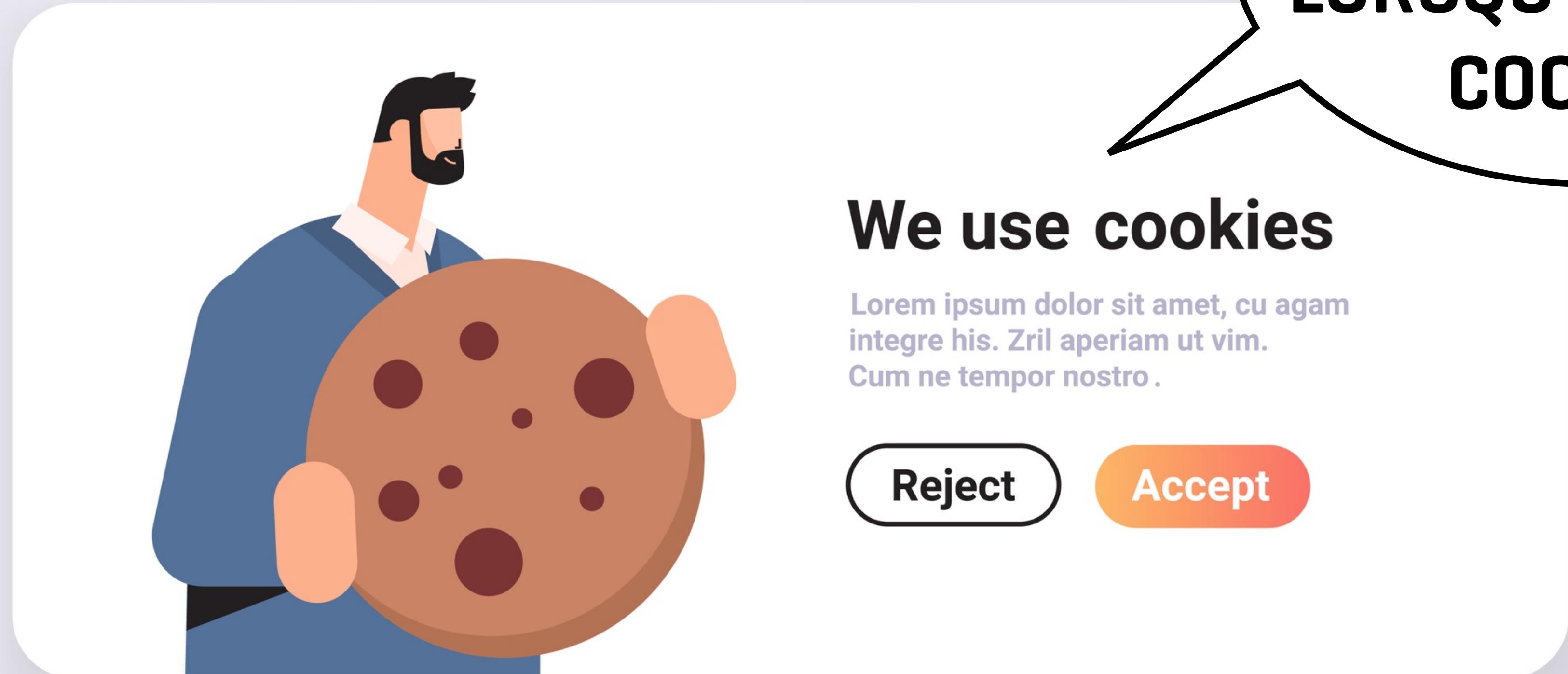
**VARNISH NE  
CACHE PAS  
LORSQU'IL Y A DES  
COOKIES**

## We use cookies

*Lorem ipsum dolor sit amet, cu agam  
integre his. Zril aperiam ut vim.  
Cum ne tempor nostro .*

Reject

Accept



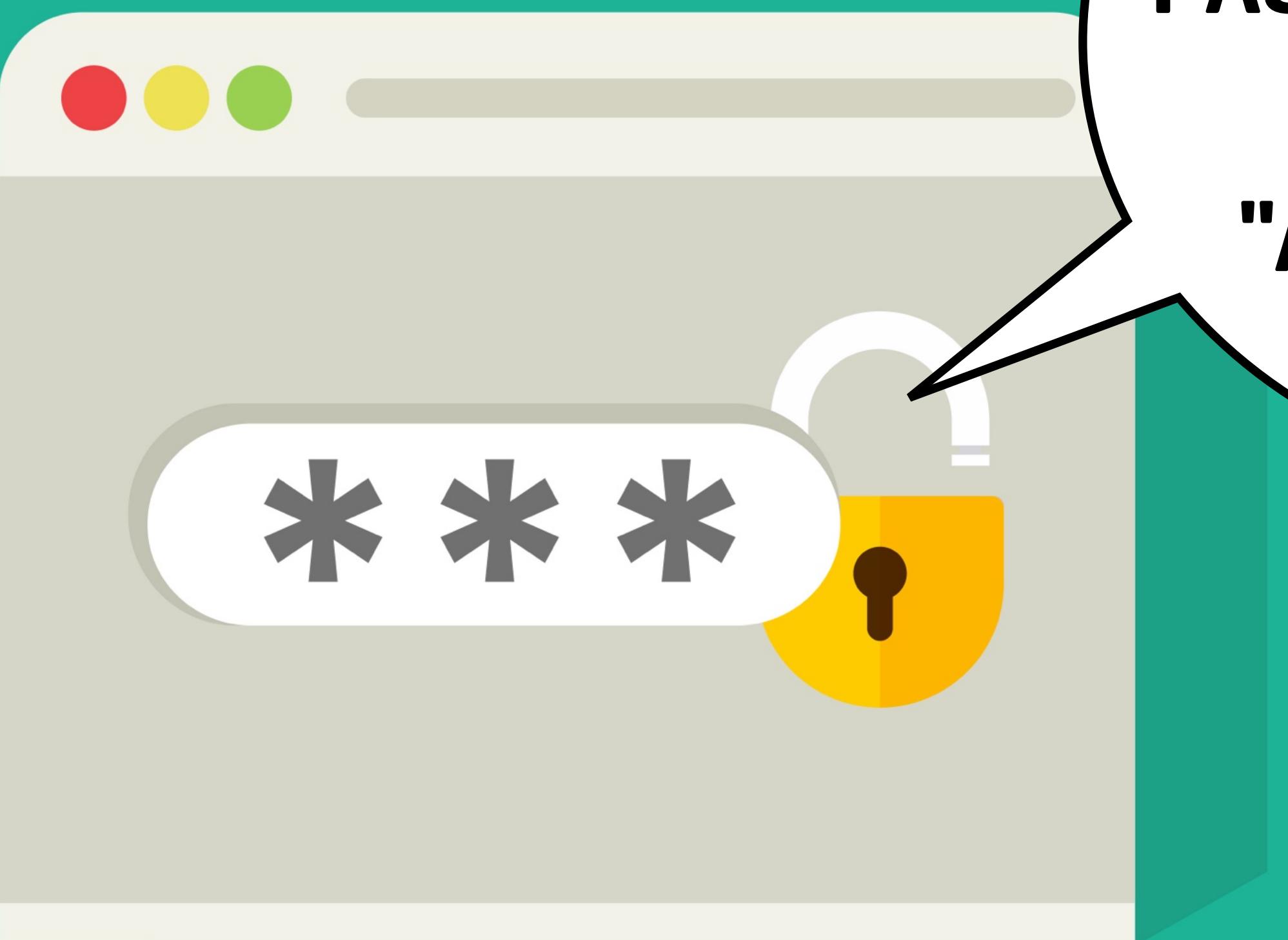


PAS EN CACHE

**CONTENU PRIVÉ**

**LES COOKIES NE SONT PAS  
NÉCESSAIRES AUX API REST  
CAR LE CLIENT DÉTIENT LE "STATE"**

**VARNISH NE CACHE  
PAS LORSQU'IL Y A UN  
ENTÊTE  
"AUTHORIZATION"**



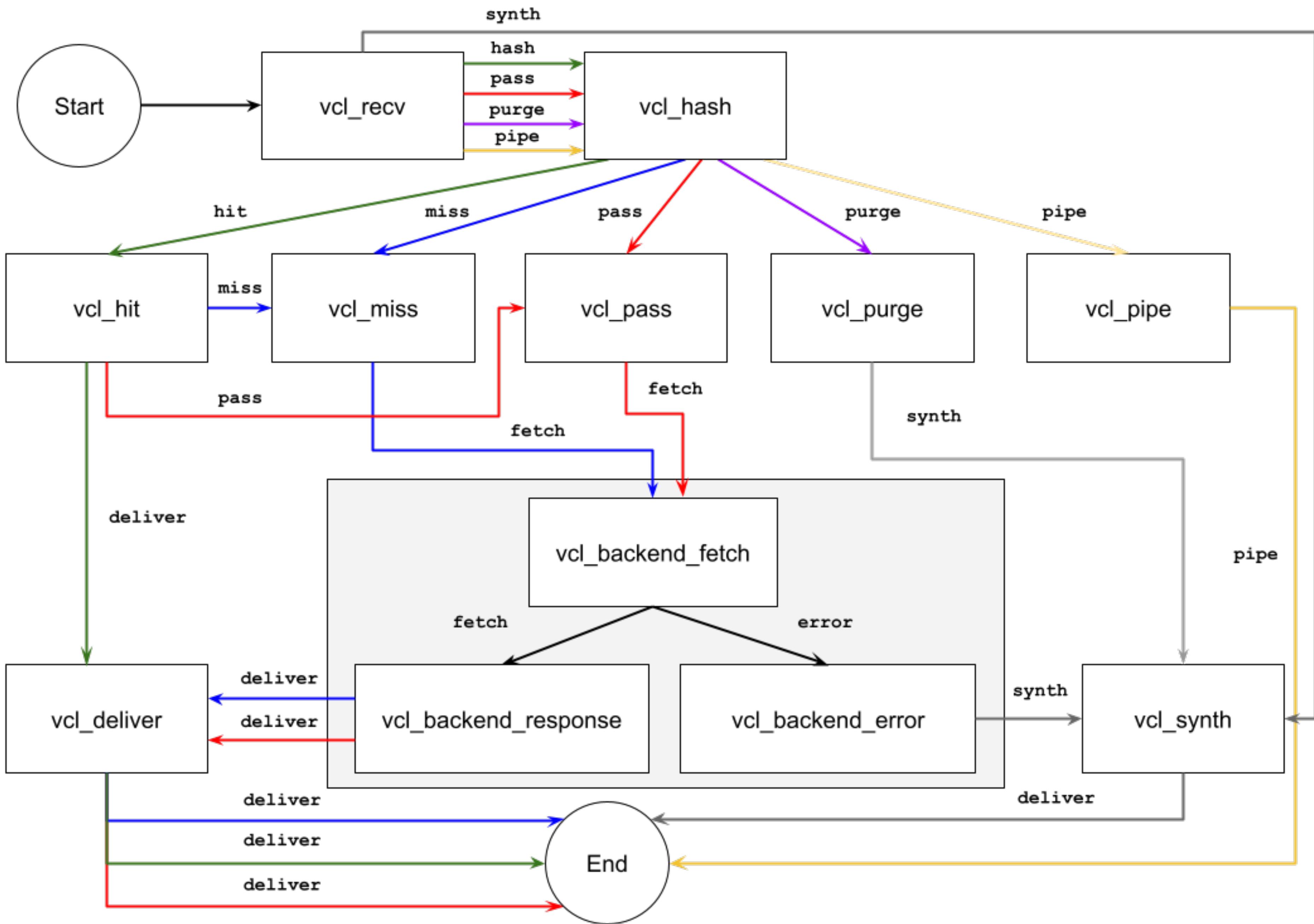
***LE COMPORTEMENT INTÉGRÉ PEUT  
ÊTRE ÉTENDU À TRAVER LE  
"VARNISH CONFIGURATION LANGUAGE"***

# **VARNISH CONFIGURATION LANGUAGE**



# CAPACITÉS VCL

- ✓ TRAITEMENT DES REQUÊTES
- ✓ ROUTAGE DES REQUÊTES
- ✓ MANIPULATION DES RÉPONSES
- ✓ SÉLECTION DU BACK-END
- ✓ CONTRÔLER LE CACHE
- ✓ PRISE DE DÉCISION SUR LE "EDGE"



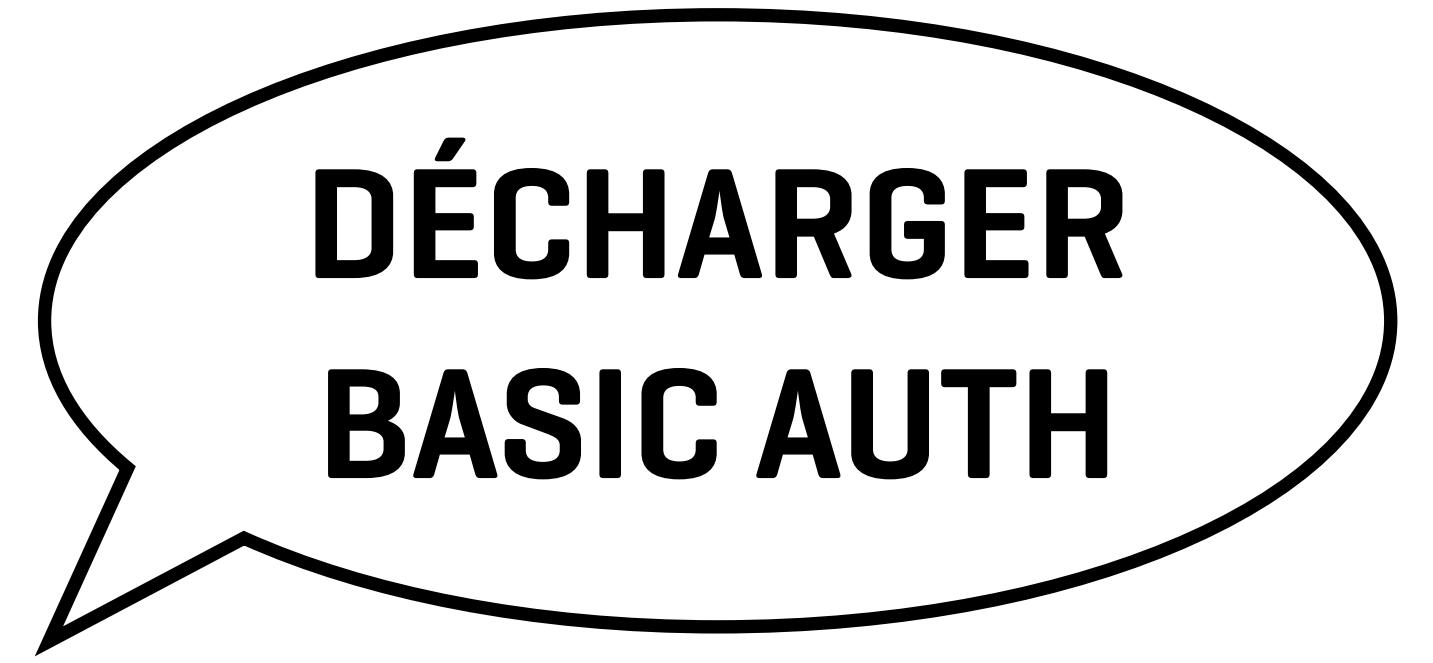
```
vcl 4.1;

import basicauth;

sub vcl_recv {
    if (!basicauth.match("/var/www/.htpasswd", req.http.Authorization)) {
        return (synth(401, "Restricted"));
    }

    if (req.method == "GET" || req.method == "HEAD") {
        return (hash);
    }
}

sub vcl_synth {
    if (resp.status == 401) {
        set resp.http.WWW-Authenticate = {"Basic realm=\"Restricted area\""};
    }
}
```



## DÉCHARGER BASIC AUTH

```
vcl 4.1;

import basicauth;

sub vcl_recv {
    if (!basicauth.match("/var/www/.htpasswd", req.http.Authorization)) {
        return (synth(401, "Restricted"));
    }

    if (req.method == "GET" || req.method == "HEAD") {
        return (hash);
    }
}

sub vcl_synth {
    if (resp.status == 401) {
        set resp.http.WWW-Authenticate = {"Basic realm=\"Restricted area\""};
    }
}
```

*ET JWT?*

```
vcl 4.1;

sub vcl_recv {
    if (req.http.Cookie) {
        set req.http.Cookie = ";" + req.http.Cookie;
        set req.http.Cookie = regsuball(req.http.Cookie, ";+", ";");
        set req.http.Cookie = regsuball(req.http.Cookie,
            ";(SESS[a-zA-Z0-9]+|SSESS[a-zA-Z0-9]+|NO_CACHE)=",
            "; \1=");
        set req.http.Cookie = regsuball(req.http.Cookie, ":[^:][^:]*", "");
        set req.http.Cookie = regsuball(req.http.Cookie, "[; ]+|[; ]+$", "");

        if (req.http.cookie ~ "^\\s*$") {
            unset req.http.Cookie;
        }
    }
}
```



**Only  
keep certain  
cookies**

**eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.** eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6Ikpvag4gRG91IiwiaWF0IjoxNTg1MjMwODIyLCJleHAiOjE1ODUyMzA5NzJ9. **3GFtCMMmyXsJjSd308FJfpWUrhLeT\_wlQUj9\_Nfnk0w**

# JSON WEB TOKENS

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiawF0IjoxNTg1MjMwODIyLCJleHAiOjE1ODUyMzA5NzJ9.3GFtCMMmyXsJjSd308FJfpWUrhLeT\_wlQUj9\_Nfnk0w

- ✓ 3 PARTIES
- ✓ POINT SÉPARÉ
- ✓ JSON CODÉ EN BASE64
- ✓ ENTÊTE
- ✓ CHARGE UTILE
- ✓ SIGNATURE [HMAC AVEC SECRET]

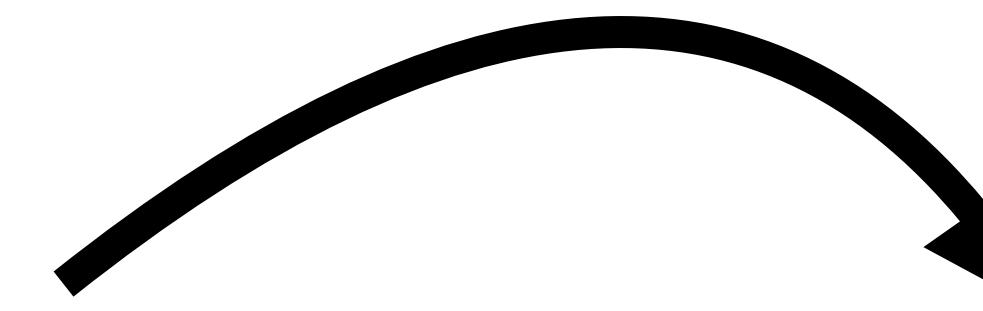
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTg1MjMwODIyLCJleHAiOjE1ODUyMzA5NzJ9

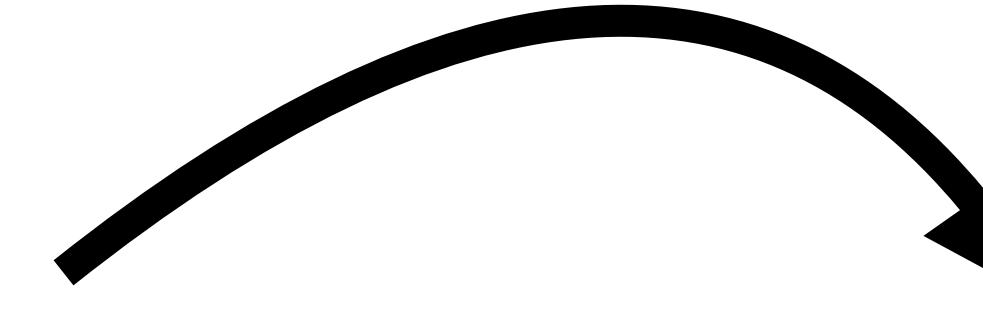
3GFtCMMmyXsJjSd3O8FJfpWUrhLeT\_wlQUj9\_Nfnk0w



```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```



```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1585230822,  
  "exp": 1585230972  
}
```



```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  secret  
)
```

Authorization: Bearer <token>

```
vcl 4.1;

import digest;
import std;

sub vcl_recv {
    if(req.http.Authorization !~ "^\n      Bearer [^\\.] + \\. [^\\.] + \\. [^\\.] + ([ ]*;[^;]+)*$") {
        return(synth(403, "JWT token required"));
    }

    set req.http.jwtHdr = regsub(req.http.Authorization,"Bearer ([^\\.]+)\\.[^\\.]+\\.[^\\.]+" ,"\1");
    set req.http.jwtType = regsub(digest.base64url_decode(req.http.jwtHdr), {"^.*?"typ"\s*:\n\s*(\w+)" .*?$$"}, "\1");
    set req.http.jwtAlg = regsub(digest.base64url_decode(req.http.jwtHdr), {"^.*?"alg"\s*:\n\s*(\w+)" .*?$$"}, "\1");

    if(req.http.jwtType != "JWT" || req.http.jwtAlg != "HS256") {
        return(synth(403, "Invalid token header"));
    }

    set req.http.jwtRawPayload = regsub(req.http.Authorization,"Bearer [^\\.] + \\. ([^\\.] + )\\.\n[^\\.] +","\1");
```

```
set req.http.jwtSig = regsub(req.http.Authorization,"Bearer [^\\.]+\\.[^\\.]+\\.( [^\\.]+)","\\1");
set req.http.jwtCurSig =
digest.base64url_nopad_hex(digest.hmac_sha256("secret",req.http.jwtHdr + "."
+req.http.jwtRawPayload));

if(req.http.jwtSig != req.http.jwtCurSig) {
    return(synth(403, "Invalid token signature"));
}

set req.http.jwtPayload = digest.base64url_decode(req.http.jwtRawPayload);
set req.http.jwtExp = regsub(req.http.jwtPayload,{"^.*?"exp"\s*:\s*(\w+).*?\$"}, "\\1");
set req.http.username = regsub(req.http.jwtPayload,{"^.*?"username"\s*:
\s*"( \w+)" .*?\$"}, "\\1");

if(std.time(req.http.jwtExp,now) < now) {
    return(synth(403, "Token expired"));
}
```

# **VMOD\_JWT**

## **EN VARNISH ENTERPRISE**

```
vcl 4.1;

sub vcl_recv {
    if (req.http.Cookie) {
        set req.http.Cookie = ";" + req.http.Cookie;
        set req.http.Cookie = regsuball(req.http.Cookie, ";+", ";");
        set req.http.Cookie = regsuball(req.http.Cookie,
            ";(SESS[a-z0-9]+|SSESS[a-z0-9]+|NO_CACHE)=", "; \1=");
        set req.http.Cookie = regsuball(req.http.Cookie, ":[^:][^:]*", "");
        set req.http.Cookie = regsuball(req.http.Cookie, "[; ]+|[; ]+$", "");

        if (req.http.cookie ~ "^\\s*$") {
            unset req.http.Cookie;
        }
    }
}
```



**Only  
keep certain  
cookies**

```
vcl 4.1;

import jwt;

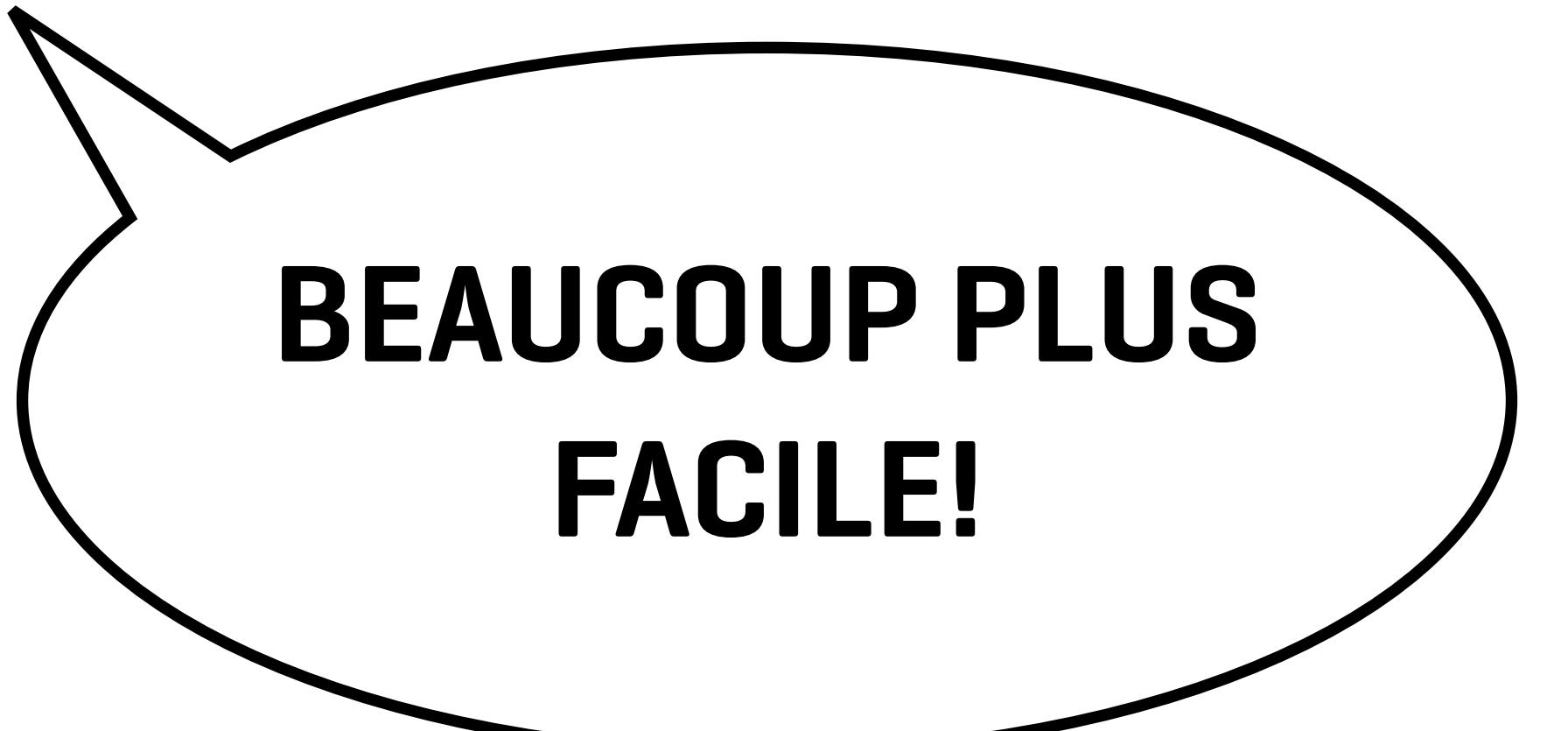
sub vcl_init {
    new jwt_reader = jwt.reader();
}
sub vcl_recv {
    if (!jwt_reader.parse(regsub(req.http.Authorization,"^Bearer (.+)$","\1")) ||
        !jwt_reader.set_key("secret") ||
        !jwt_reader.verify("HS256")) {
        return (synth(401, "Invalid token"));
}
}
```

```
vcl 4.1;

import jwt;

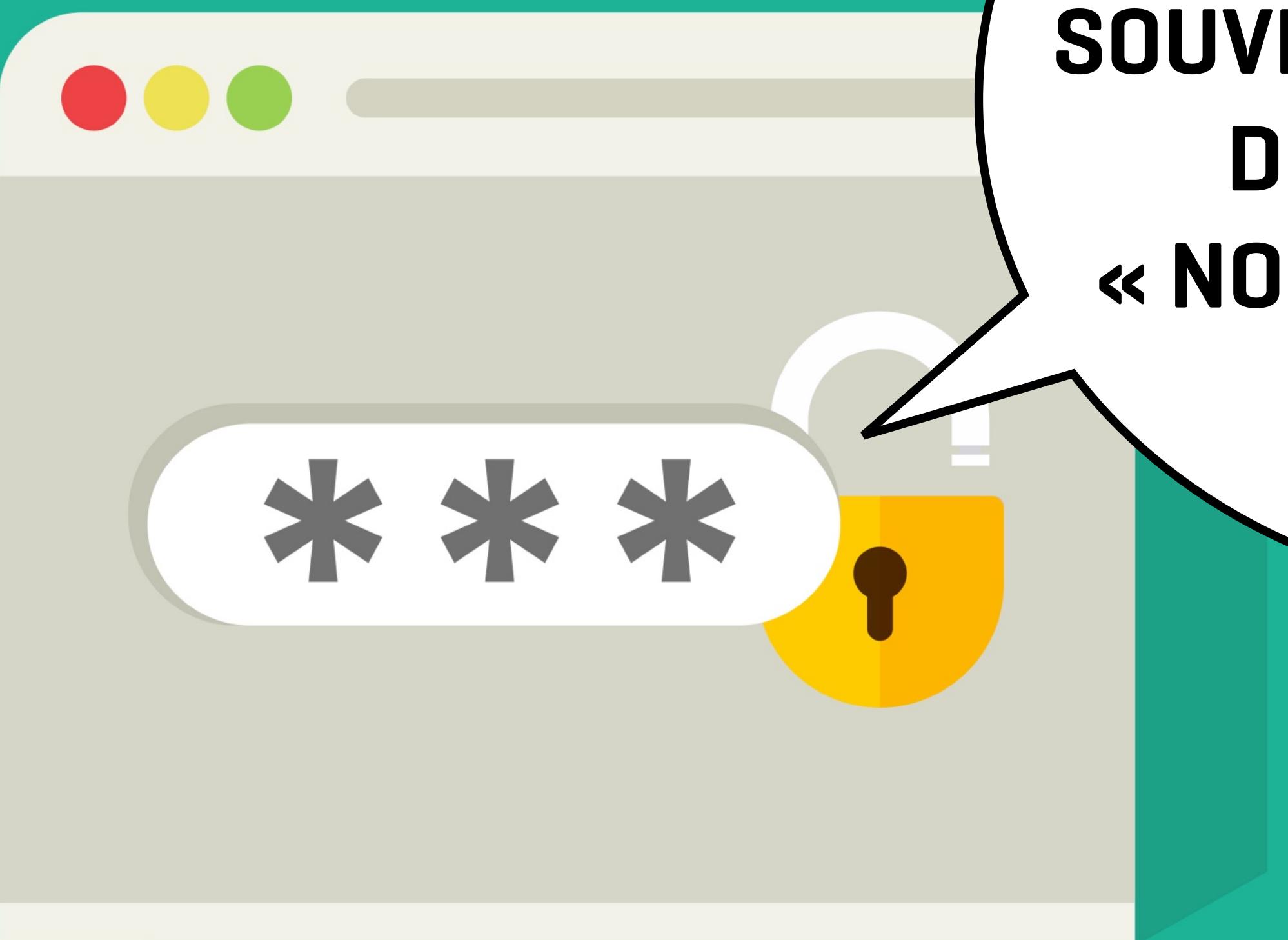
sub vcl_init {
    new jwt_reader = jwt.reader();
}

sub vcl_recv {
    if (!jwt_reader.parse(regsub(req.http.Authorization,"^Bearer (.+)$","\1")) ||
        !jwt_reader.set_key("secret") ||
        !jwt_reader.verify("HS256")) {
        return (synth(401, "Invalid token"));
}
}
```



**BEAUCOUP PLUS  
FACILE!**

**L'AUTORISATION EST  
SOUVENT UTILISÉE DANS  
DES SCÉNARIOS  
« NON MIS EN CACHE »**



**UTILISER DES ESPACES  
RÉSERVÉS POUR SÉPARER  
LE CONTENU "CACHEABLE"  
ET LE CONTENU  
"NON-CACHEABLE"**



# AJAX

POUR LES SITES

# ***EDGE SIDE INCLUDES***

ESI

```
<esi:include src="/header" />
```

**SOUS-  
REQUÊTE ESI**

```
<esi:include src="/header" />
```

- ✓ **BALISE D'ESPACE RÉSERVÉ**
- ✓ **ANALYSÉ PAR VARNISH**
- ✓ **UNE COMPOSITION DE BLOCS**
- ✓ **"STATE" PAR BLOC**
- ✓ **"TTL" PAR BLOC**

```
sub vcl_backend_fetch {
    set bereq.http.Surrogate-Capability = "key=ESI/1.0";
}

sub vcl_backend_response {
    if (beresp.http.Surrogate-Control ~ "ESI/1.0") {
        unset beresp.http.Surrogate-Control;
        set beresp.do_esi = true;
    }
}
```

**ANNONCER  
SUPPORT ESI**

```
sub vcl_backend_fetch {  
    set bereq.http.Surrogate-Capability = "key=ESI/1.0";  
}  
  
sub vcl_backend_response {  
    if (beresp.http.Surrogate-Control ~ "ESI/1.0") {  
        unset beresp.http.Surrogate-Control;  
        set beresp.do_esi = true;  
    }  
}
```

**NÉGOCIER  
AVEC LE  
BACKEND**

GET /api/user/1234

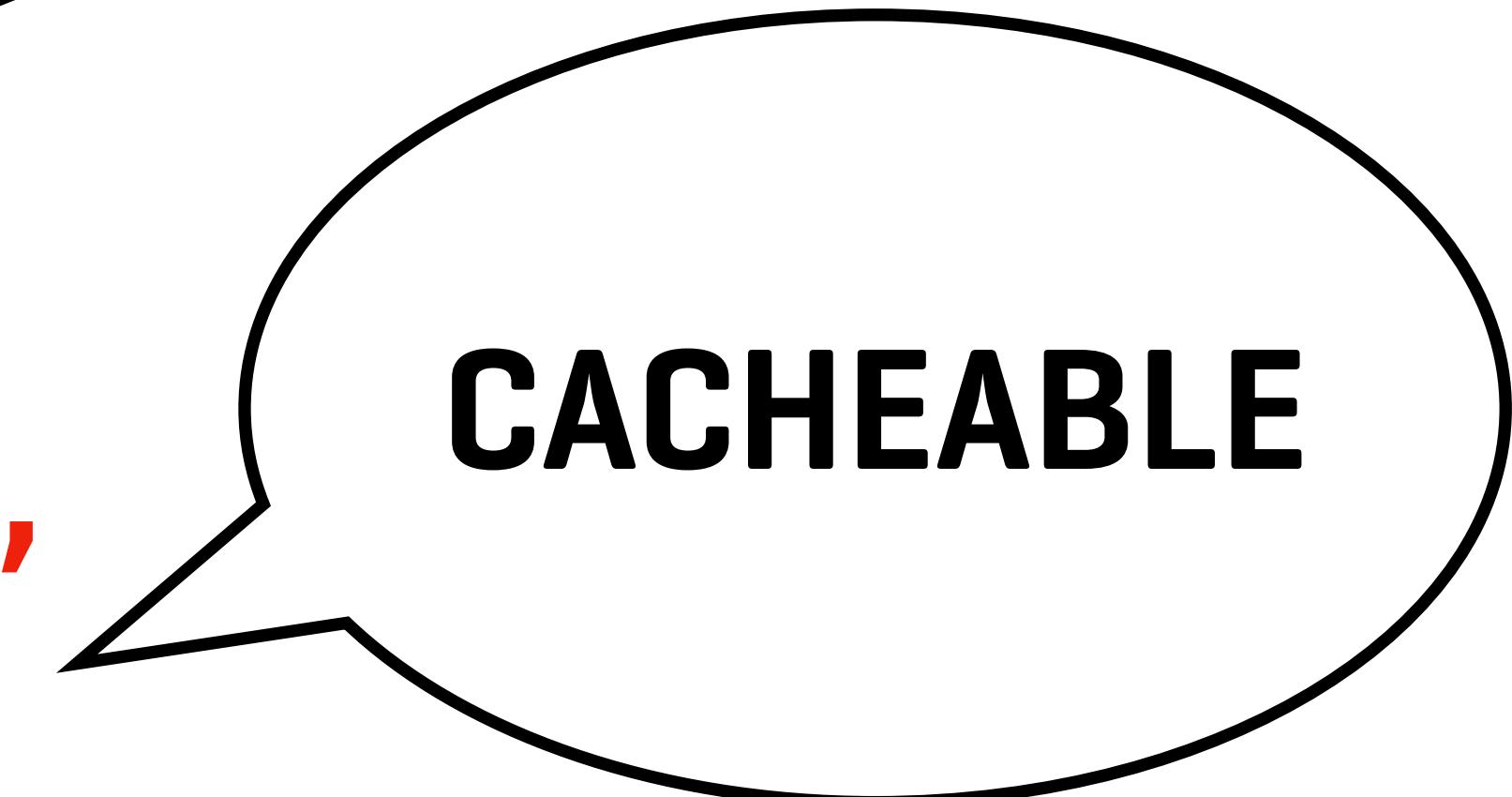
Authorization: Bear <token>

```
{  
    "username" : "johndoe",  
    "firstname" : "John",  
    "lastname" : "Doe",  
    "session": {  
        "loggedIn": true,  
        "itemsInCart": 5,  
        "sessionId" : "fqt89o4p023n1xie75f6gftc8h"  
    }  
}
```

GET /api/user/1234

Authorization: Bear <token>

```
{  
    "username" : "johndoe",  
    "firstname" : "John",  
    "lastname" : "Doe",  
    "session": {  
        "loggedIn": true,  
        "itemsInCart": 5,  
        "sessionId" : "fqt89o4p023n1xie75f6gftc8h"  
    }  
}
```



GET /api/user/1234

Authorization: Bear <token>

{

  "username" : "johndoe",

  "firstname" : "John",

  "lastname" : "Doe",

  "session": <esi:include src="/esi/session" />

}



**UNCACHEABLE**

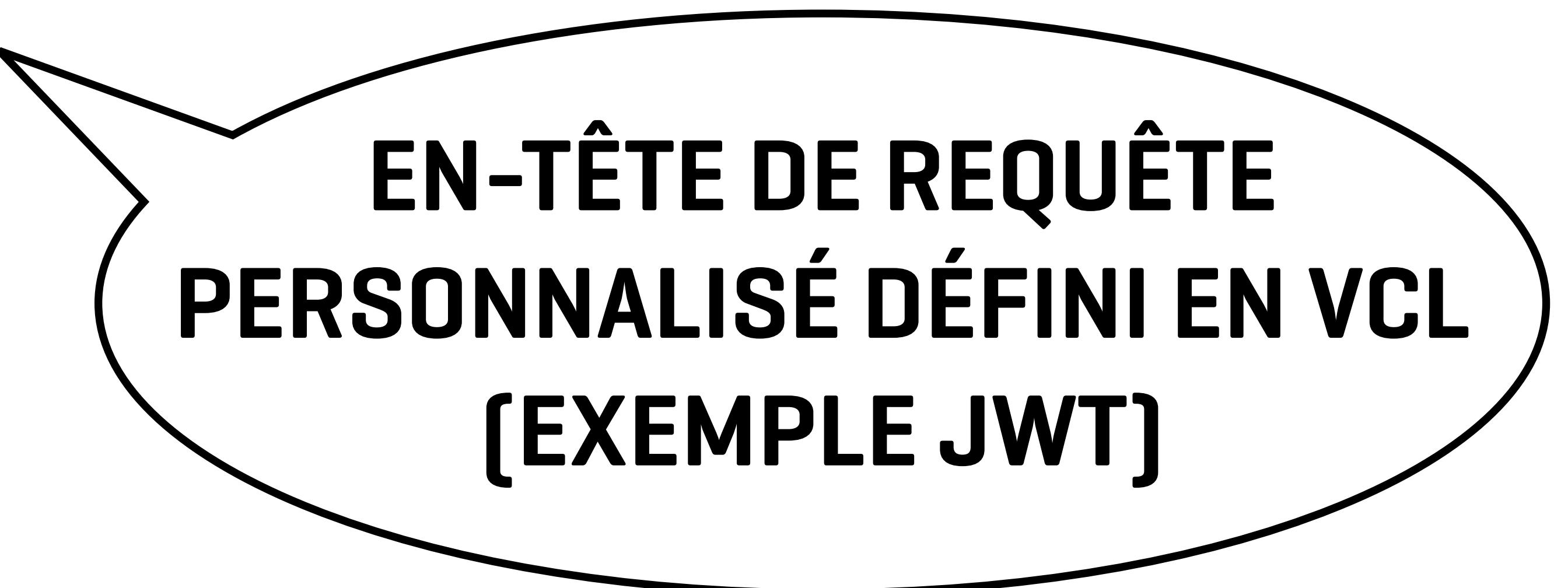
**NE METTEZ PAS DE  
L'ESI EN CACHE**

**Cache-Control: private, no-cache, no-store**

**CRÉER UNE  
VARIATION PAR  
SESSION**

Cache-Control: public, max-age=3600  
Vary: Authorization

**Cache-Control: public, max-age=3600  
Vary: jwtUsername**



**EN-TÊTE DE REQUÊTE  
PERSONNALISÉ DÉFINI EN VCL  
(EXEMPLE JWT)**

# **CALCULER LES INFORMATIONS DE SESSION SUR LE "EDGE"**



# *HTTP SYNTHÉTIQUE*

```
sub vcl_recv {  
    return(synth(200));  
}
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>200 0K</title>
  </head>
  <body>
    <h1>Error 200 0K</h1>
    <p>Synthetic output</p>
    <h3>Guru Meditation:</h3>
    <p>XID: 32770</p>
    <hr>
    <p>Varnish cache server</p>
  </body>
</html>
```



USER



VARNISH

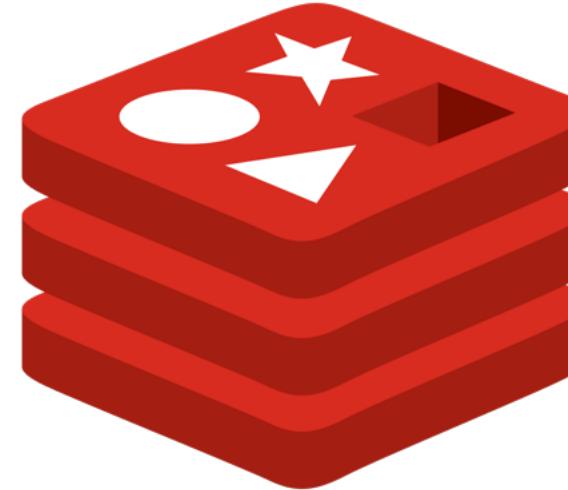
GET /esi/session  
Authorization: Bearer <token>

HTTP/1.1 200 OK  
Content-Type: application/json  
X-Varnish: 163854 196622  
Age: 27

```
{  
    "loggedIn": true,  
    "itemsInCart": 5,  
    "sessionId" : "fqt89o4p023n1xe75f6gftc8h"  
}
```

**JSON GÉNÉRÉ  
PAR VARNISH**

# **RÉCUPÉRER LES DONNÉES DE SESSION DEPUIS**



**redis**

<https://github.com/carlosabalde/libvmod-redis>

```
> EXISTS fqt89o4p023n1xie75f6gftc8h
(integer) 1
>
> HMGET fqt89o4p023n1xie75f6gftc8h loggedIn itemsInCart
1) "true"
2) "5"
>
```

```
vcl 4.1;

import redis;

sub vcl_init {
    new redis_client = redis.db(
        location="172.24.2.12:6379",
        shared_connections=false,
        max_connections=1);
}

sub vcl_recv {
    if(req.url == "/esi/session") {
        return(synth(200));
    }
}
```

```
sub vcl_synth {
    if(req.url == "/esi/session") {
        redis_client.command("EXISTS");
        redis_client.push(req.http.sessionId);
        redis_client.execute();

        set resp.http.Content-Type = "application/json";
        if(redis_client.get_reply() == "0") {
            set resp.body = "{}";
            set resp.status = 404;
            return(deliver);
        }

        redis_client.command("HMGET");
        redis_client.push(req.http.sessionId);
        redis_client.push("loggedIn");
        redis_client.push("itemsInCart");
        redis_client.execute();

        set resp.body = "{\"
            \"loggedIn\": \"\"} + redis_client.get_array_reply_value(0) + \"\",
            \"ItemsInCart\": \"\"} + redis_client.get_array_reply_value(1) + \"\",
            \"sessionId\": \"\"} + req.http.sessionId + \"\",
        }\"};
        return(deliver);
    }
}
```

```
sub vcl_synth {
    if(req.url == "/esi/session") {
        redis_client.command("EXISTS");
        redis_client.push(req.http.sessionId);
        redis_client.execute();

        set resp.http.Content-Type = "application/json";
        if(redis_client.get_reply() == "0") {
            set resp.body = "{}";
            set resp.status = 404;
            return(deliver);
        }

        redis_client.command("HMGET");
        redis_client.push(req.http.sessionId);
        redis_client.push("loggedIn");
        redis_client.push("itemsInCart");
        redis_client.execute();

        set resp.body = "{\"loggedIn\": \"\"} + redis_client.get_array_reply_value(0) + \"\",
                      \"ItemsInCart\": \"\"} + redis_client.get_array_reply_value(1) + \"\",
                      \"sessionId\": \"\"} + req.http.sessionId + \"\",
                      \"\"}";
        return(deliver);
    }
}
```

**EXTRAITER  
L'ID DE SESSION  
DE JWT DANS  
VCL\_RECV**

# **AUTRES TRUCS API UTILISANT LA COLLECTION VARNISH-MODULES**

<https://github.com/varnish/varnish-modules>

```
vcl 4.1;

import std;
import bodyaccess;

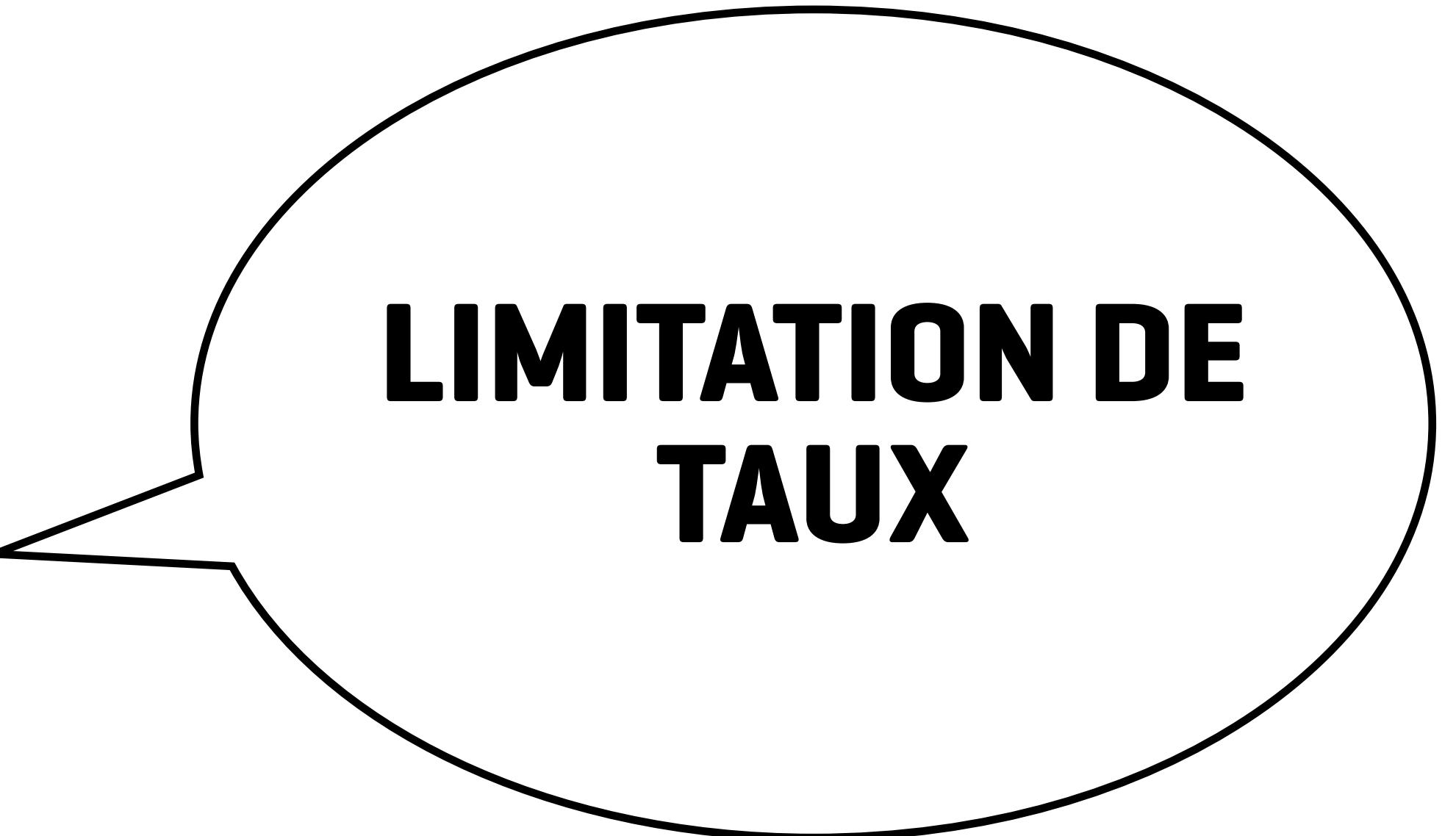
sub vcl_recv {
    std.cache_req_body(10KB);
    set req.http.x-method = req.method;
    return(hash);
}

sub vcl_hash {
    bodyaccess.hash_req_body();
}

sub vcl_backend_fetch {
    set bereq.method = bereq.http.x-method;
}
```

**CACHE  
CORPS DE LA  
REQUÊTE POUR  
POST, PUT,  
PATCH, ...**

```
vcl 4.1;  
  
import tcp;  
  
sub vcl_recv {  
    tcp.set_socket_pace(1024);  
}
```

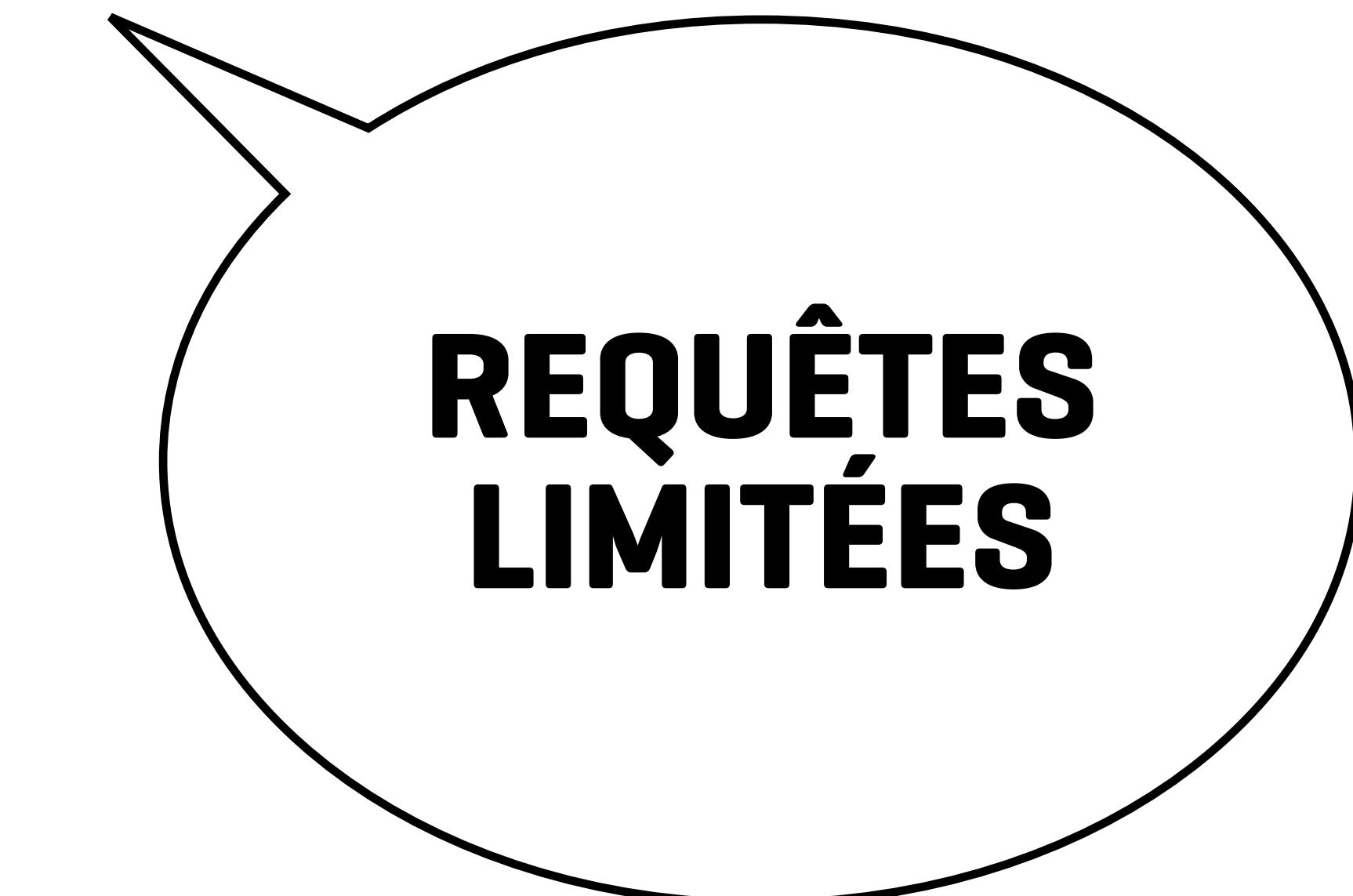


**LIMITATION DE  
TAUX**

```
vcl 4.1;

import vsthrottle;

sub vcl_recv {
    if (vsthrottle.is_denied(client.identity, 15, 10s, 30s)) {
        return (synth(429, "Too Many Requests. You can retry in "
        + vsthrottle.blocked(client.identity, 15, 10s, 30s)
        + " seconds."));
    }
}
```



```
vcl 4.1;

import accept;

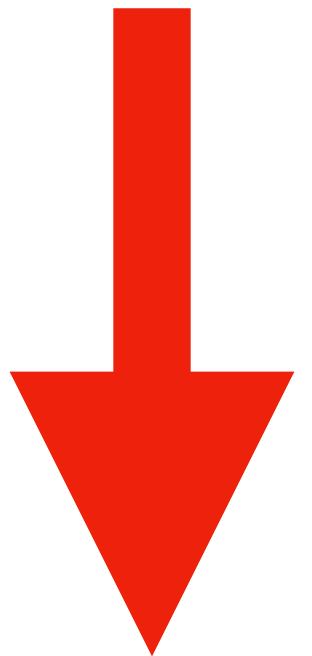
sub vcl_init {
    new lang = accept.rule("en");
    lang.add("en");
    lang.add("fr");
    lang.add("nl");
}

sub vcl_recv {
    set req.http.Accept-Language = lang.filter(req.http.Accept-
Language);
}

sub vcl_hash {
    hash_data(req.http.Accept-Language);
}
```

**AMÉLIORER LE  
TAUX DE RÉUSSITE  
SUR LES VARIATIONS DE  
CACHE DE "ACCEPT-  
LANGUAGE"**

Accept-language: fr-FR,fr;q=0.9,en-US;q=0.8,en;q=0.7



Accept-language: fr

**ET IL YA BEAUCOUP PLUS EN  
VARNISH ENTERPRISE**

# **CARACTÉRISTIQUES DE VARNISH ENTERPRISE**

- ✓ MODIFIER LA RÉPONSE AVANT DE LE METTRE EN CACHE
- ✓ PERMETTRE/BLOQUER/SEGMENTER TRAFFIC EN FONCTION DE GÉOLOCATION
- ✓ FAIRE DES REQUÊTES HTTP EN VCL
- ✓ MODULE JSON
- ✓ MODULE JWT
- ✓ BACKENDS DYNAMIQUES & SYNTHÉTIQUES

THE END

