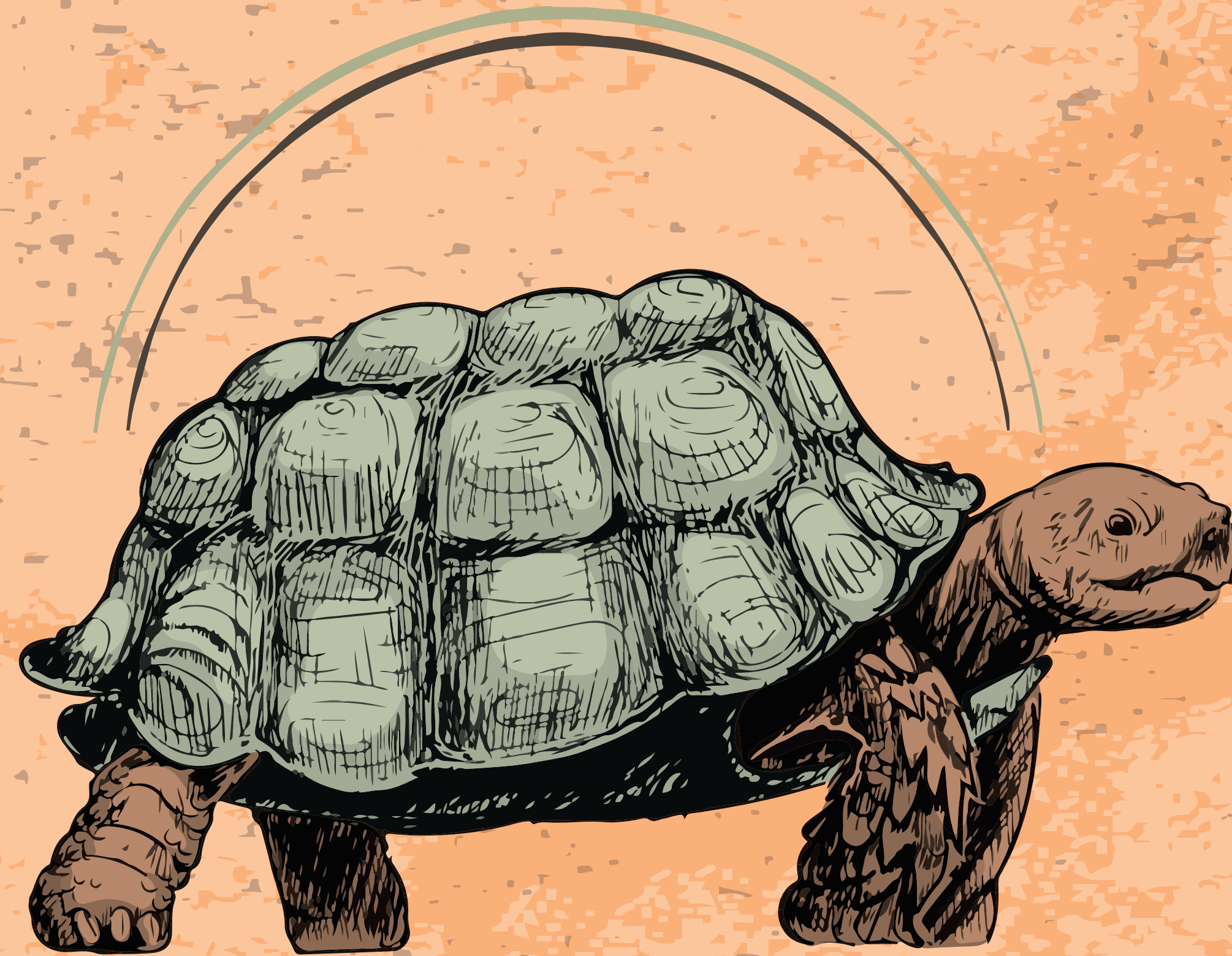


# **HTTP HEADERS THAT WILL MAKE YOUR WEBSITE GO FASTER**



**BY THIJS FERYN**

**SRE  
CON\_** EUROPE  
MIDDLE EAST  
AFRICA



*Slow websites*  
**SUCK**



# ***WEB PERFORMANCE IS AN ESSENTIAL PART OF THE USER EXPERIENCE***



***SLOW ~ DOWN***

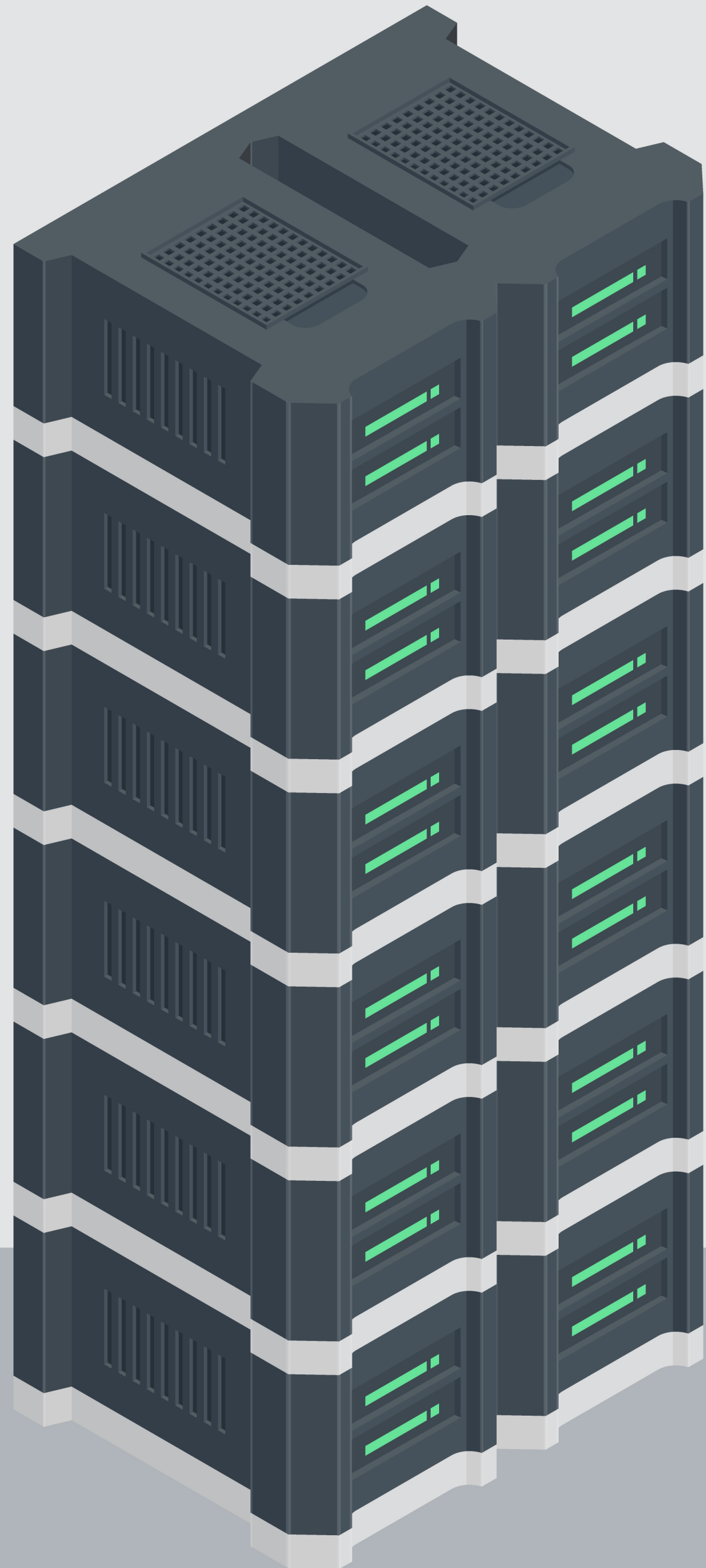






Google





***THROWING  
SERVERS  
AT THE PROBLEM***



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



# ***IDENTIFY SLOWEST PARTS***



**OPTIMIZE**

***AFTER A WHILE YOU HIT THE LIMITS***



# ***CACHE***



***HI,***



***I'M THIJS***



***I'M THE TECH  
EVANGELIST***

***AT***



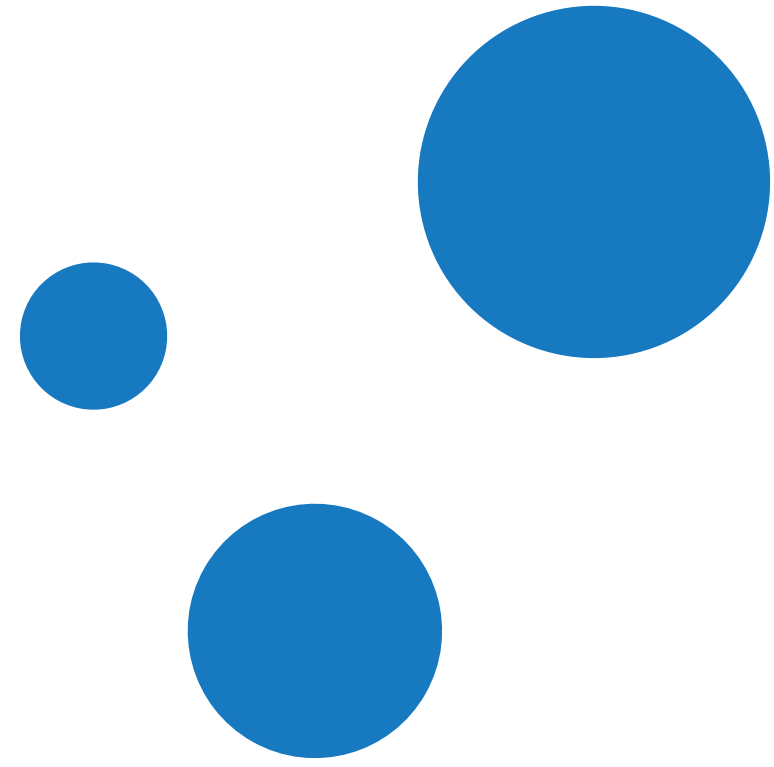
***VARNISH  
SOFTWARE***



***VARNISH***

**SOFTWARE**





***VARNISH CACHE***

# 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

**2005** - Open source Varnish Cache project begins

**2014** - *Varnish Enterprise* solution launched to support content-heavy, high-traffic businesses

**2020** - *Varnish Edge Cloud* launched to support 5G content delivery within telco networks

**2010** - Varnish Software spun out of Redpill Linpro

**2017** - Varnish start delivering custom, private CDNs to enterprises

***WE MAKE THE  
WEB FASTER***



# ***WE ENHANCE DIGITAL EXPERIENCES BY LOWERING NETWORK LATENCY***



***WE BUILD SOFTWARE-DEFINED  
WEB ACCELERATION & CONTENT  
DELIVERY SOLUTIONS***

# ACHIEVE GROWTH, PERFORMANCE & SUSTAINABILITY GOALS

**1.3 Tbps**  
per server

**1.17 Gbps**  
per watt

**WORLD'S FASTEST EDGE CONTENT DELIVERY SOFTWARE**

O'REILLY®

Compliments of  
**VARNISH**  
SOFTWARE



# 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





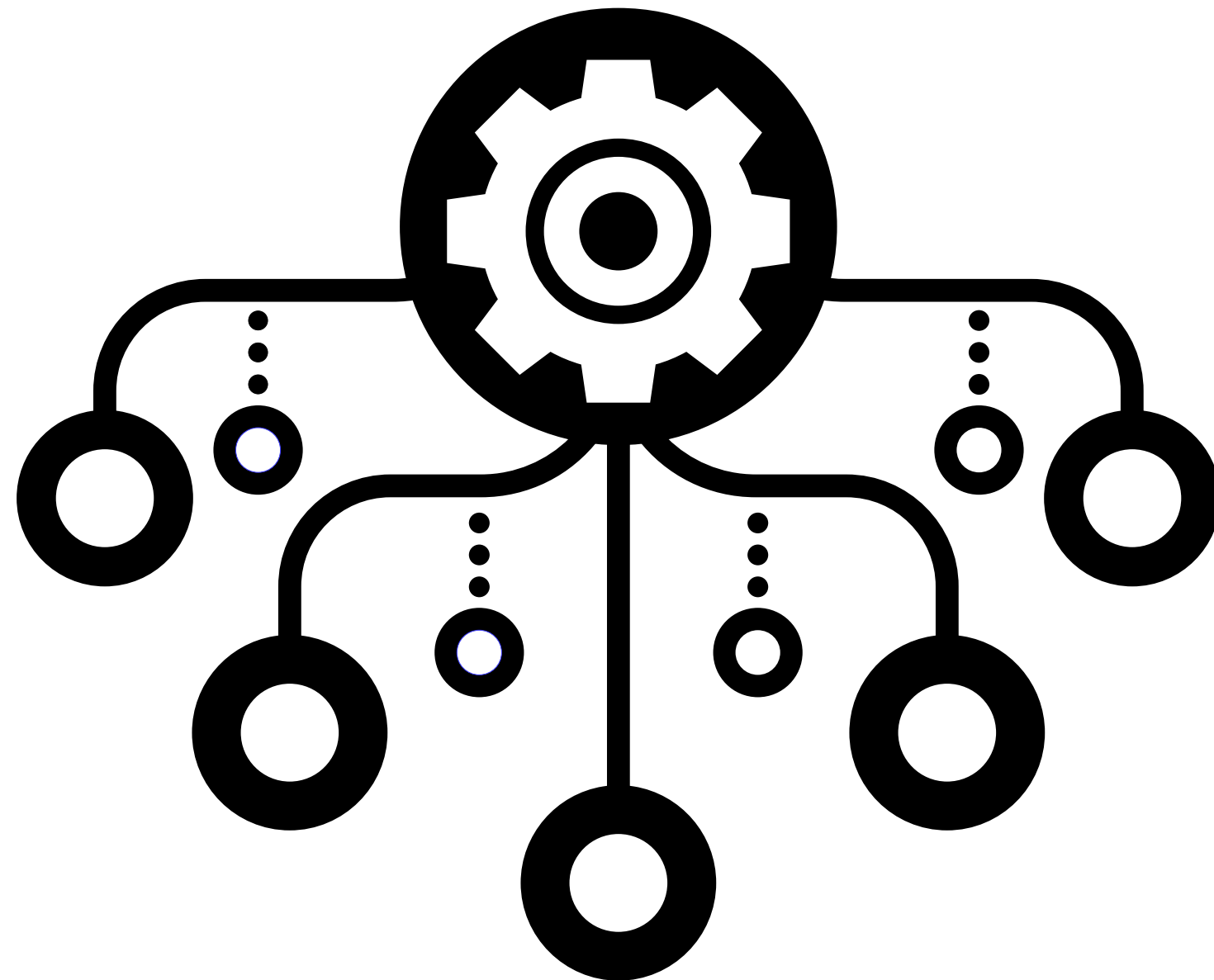
# ***CACHING***



# WHY CACHE



# ***HIGHER CONCURRENCY***



***HIGHER  
THROUGHPUT***



# *LOWER LATENCY*



# ***IMPROVE QUALITY OF EXPERIENCE***





**WHY  
RECOMPUTE  
IF THE DATA  
HASN'T  
CHANGED?**

# ***DIFFERENT KINDS OF CACHING***

- ✓ **LOCAL KEY-VALUE STORE**
- ✓ **FILE CACHE**
- ✓ **DISTRIBUTE CACHE**
- ✓ **BROWSER CACHE**
- ✓ **REVERSE CACHING PROXY**
- ✓ **CONTENT DELIVERY NETWORK**

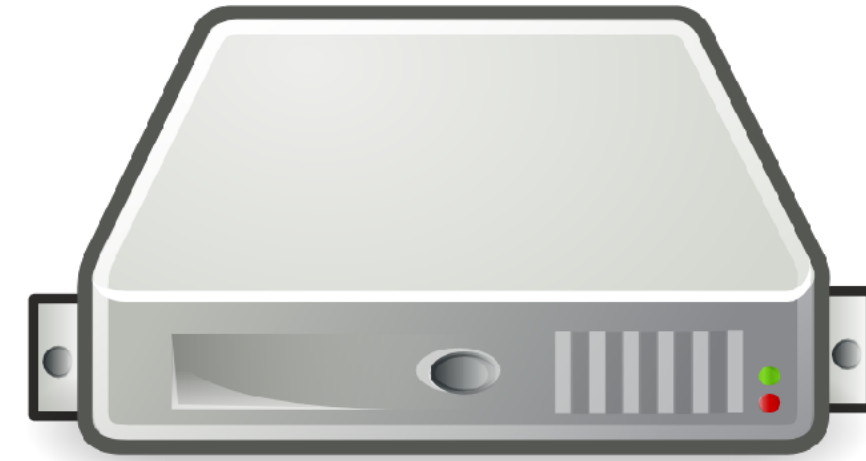




http://



***USER***

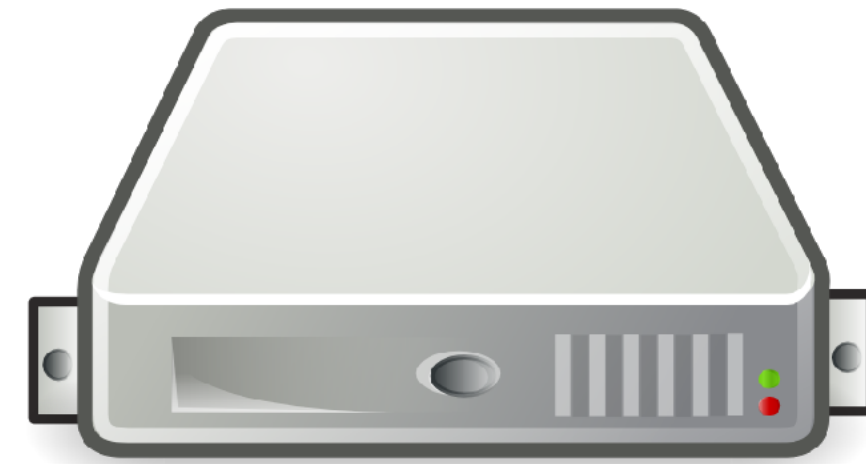


***SERVER***

# *BROWSER CACHE*



*USER*



*SERVER*

***SERVER CACHE***

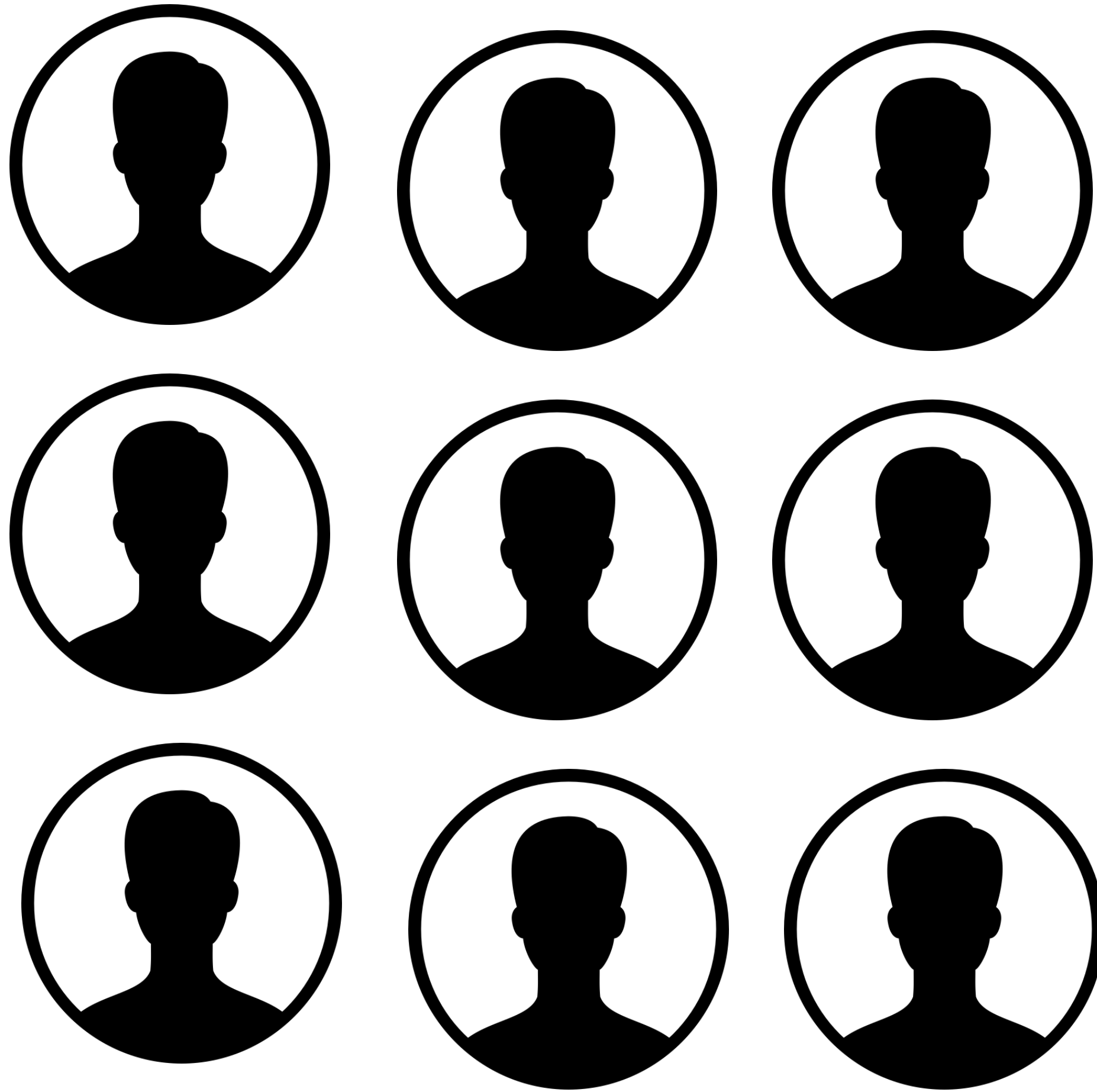


***USER***



***SERVER***

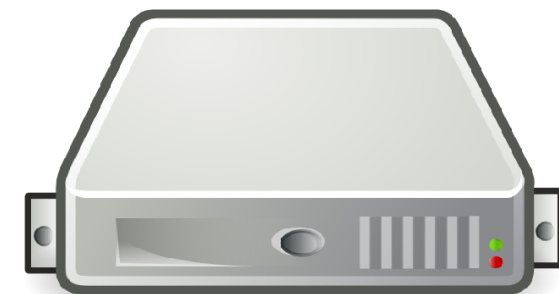
# ***UNDER PRESSURE***



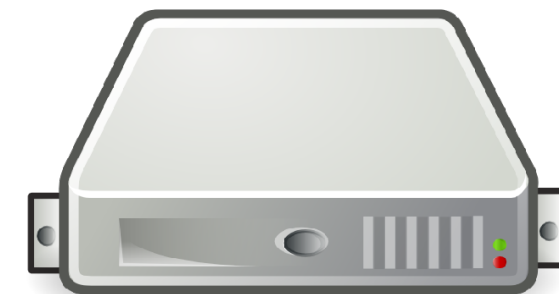
***SERVER***



***USER***



***PROXY***

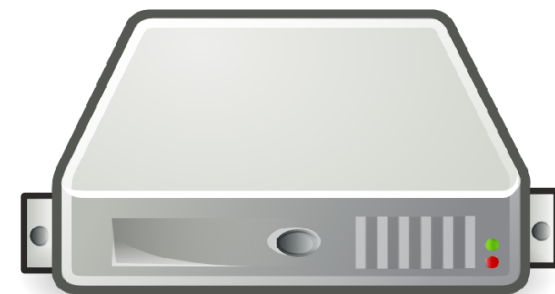


***SERVER***

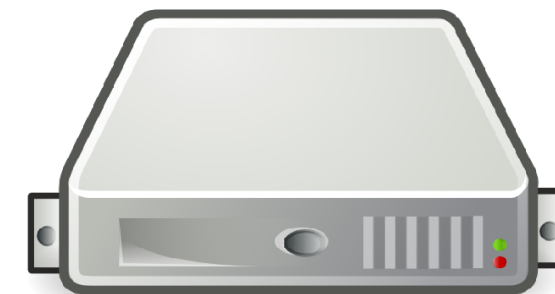
# REVERSE CACHING PROXY



**USER**

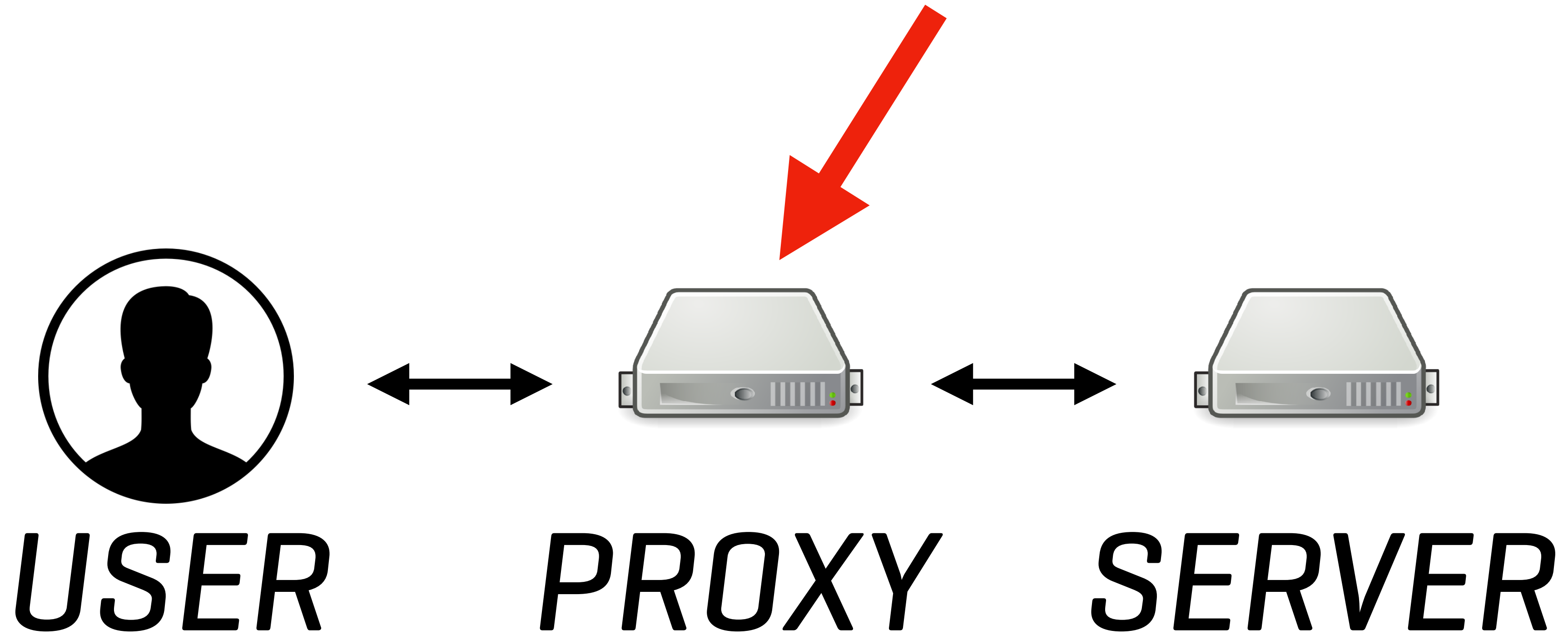


**PROXY**



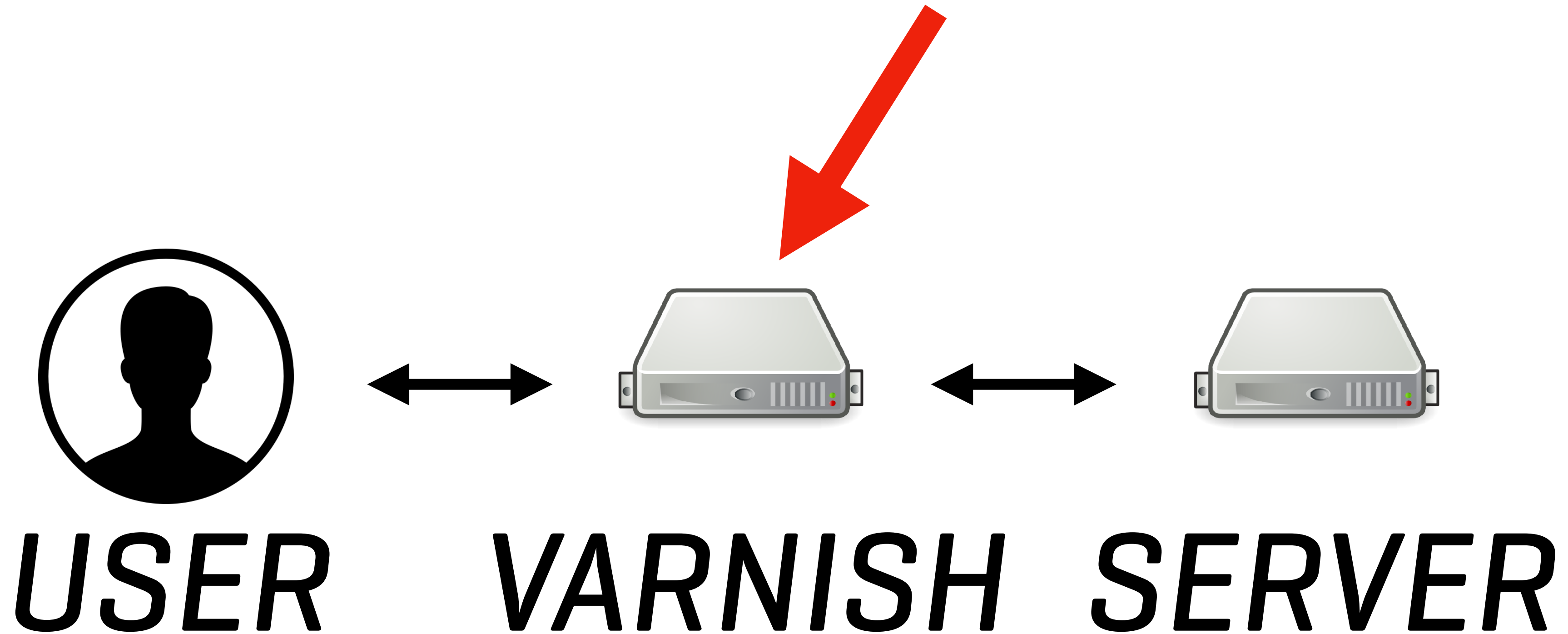
**SERVER**

# ***THE EDGE***

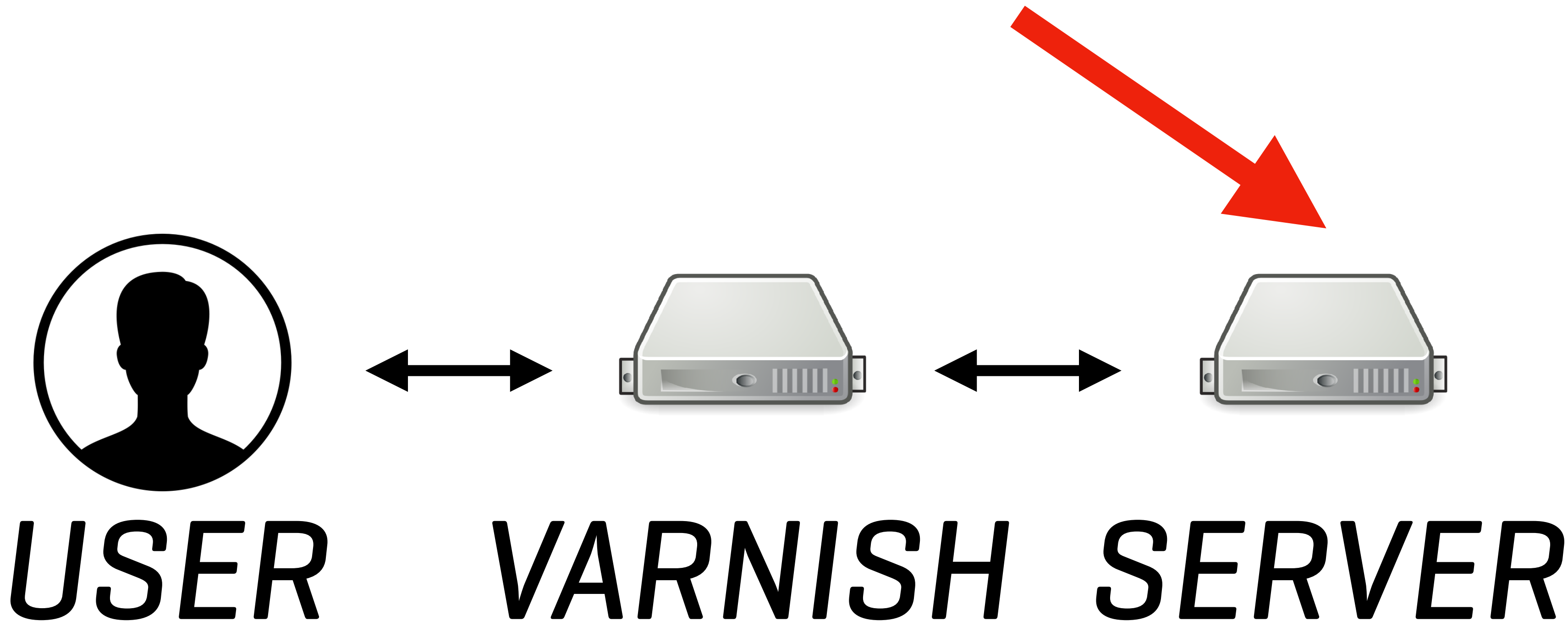




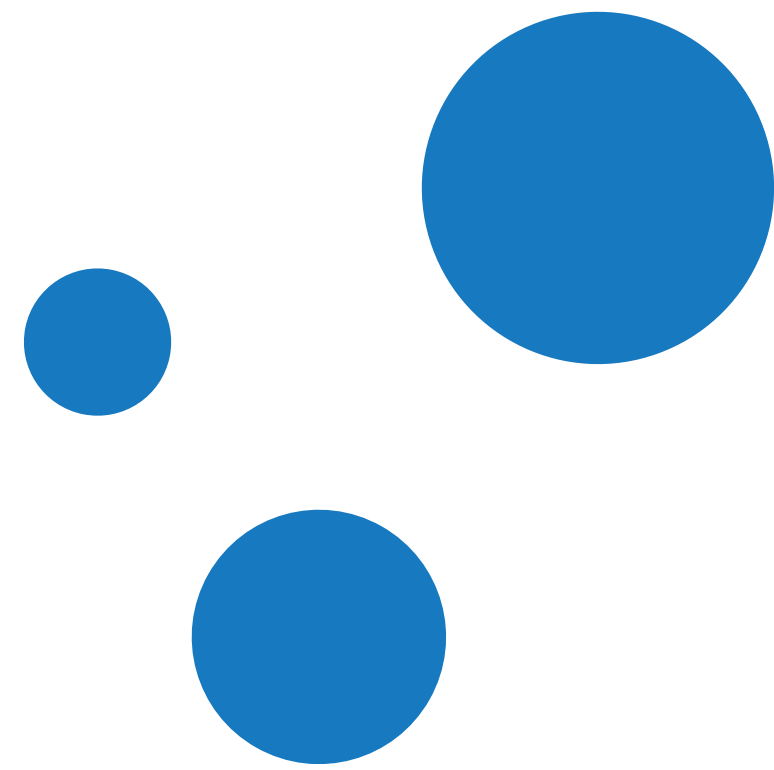
# ***THE EDGE***



# ***THE ORIGIN***







# *VARNISH CACHE*

***EVERY  
IMPLEMENTATION  
HAS ITS OWN  
CACHE POLICY  
CONFIGURATION***

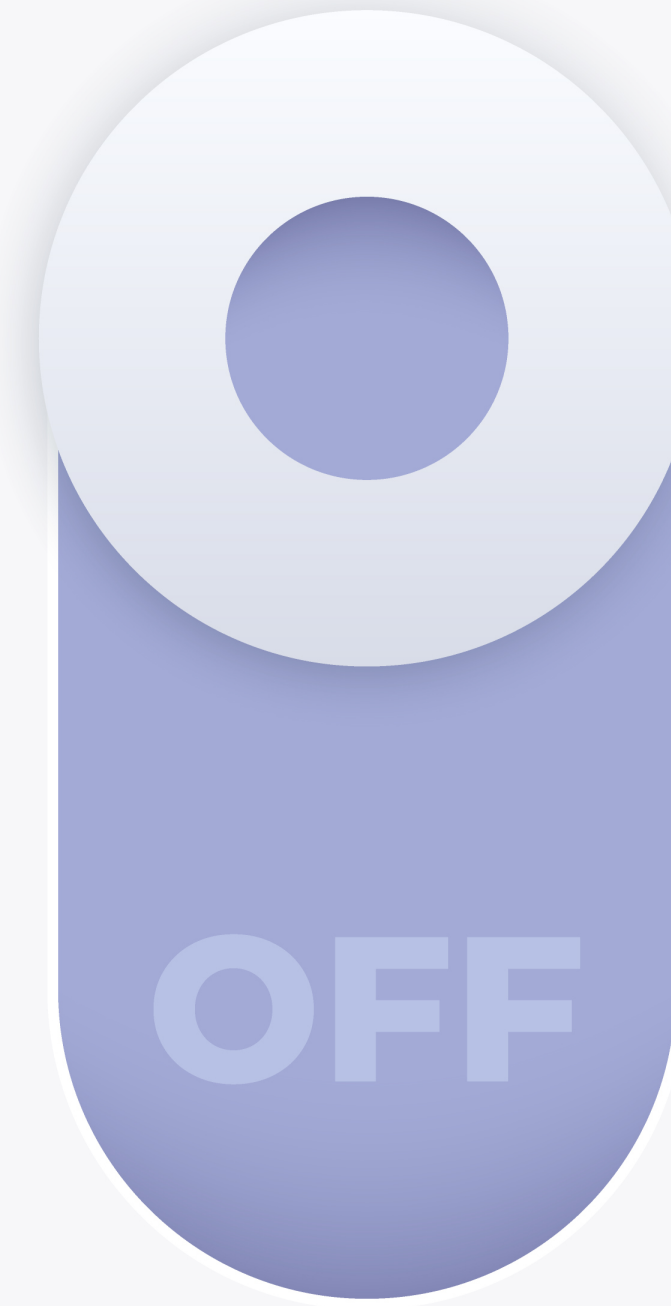




***HTTP HAS CONVENTIONAL  
BUILT-IN CACHING  
MECHANISMS***

Expires: Mon, 20 Feb 2023 21:31:06 GMT

# ***LIMITED OPTIONS***





Cache-Control: public, max-age=500

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



# ***KEY CACHING CONCEPTS***

# *CACHING*

**HOLD THE RESPONSE AND SERVE IT AGAIN UPON  
SUBSEQUENT REQUESTS**

# *PRIVATE CACHE*

**A CACHE THAT EXISTS IN THE CLIENT.  
E.G. A LOCAL DEVICE OR BROWSER CACHE.  
STORES DATA FOR A SINGLE USER.**

# *SHARED CACHE*

**A CACHE THAT SERVES MULTIPLE USERS.  
USUALLY A CACHING PROXY OR CDN.  
YOU SHOULD AVOID STORING PERSONALIZED DATA.**

# ***TIME TO LIVE***

**THE AMOUNT OF SECONDS AN OBJECT IS  
CONSIDERED FRESH.**

# ***FRESH CONTENT***

**CACHED OBJECT HASN'T EXPIRED YET.  
RESPONSE CAN BE REUSED FOR SUBSEQUENT  
REQUESTS.**



# ***STALE CONTENT***

**EXPIRED CONTENT THAT SHOULD BE REVALIDATED BEFORE SERVING. IS NOT DIRECTLY REMOVED FROM THE CACHE.**

# ***REVALIDATE CONTENT***

**ASK THE ORIGIN SERVER IF THE REQUESTED OBJECT  
IS STILL FRESH.**

# ***CACHE-CONTROL RESPONSE DIRECTIVES***

✓ PRIVATE

✓ PUBLIC

✓ IMMUTABLE

✓ MAX-AGE

✓ S-MAXAGE

✓ NO-CACHE

✓ NO-STORE

✓ NO-TRANSFORM

✓ MUST-REVALIDATE

✓ PROXY-REVALIDATE

✓ MUST-UNDERSTAND

✓ STALE-WHILE-REVALIDATE

✓ STALE-IF-ERROR

Cache-Control: public

Cache-Control: public

**CACHING ALLOWED, BOTH BY PRIVATE & SHARED CACHES**

Cache-Co

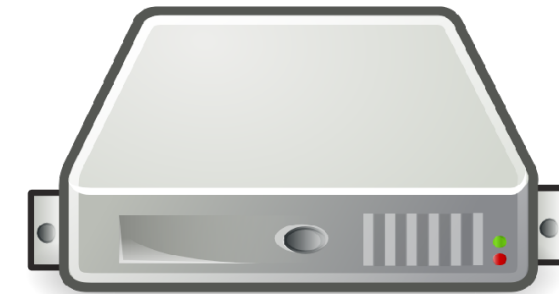
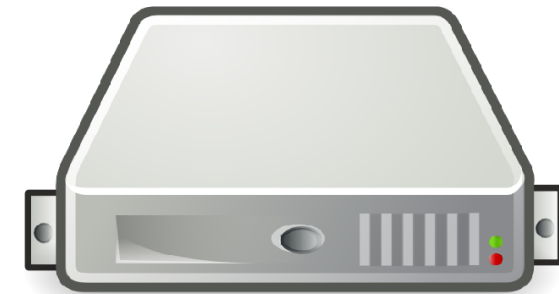
**BROWSER**

**PROXY  
SERVERS**

**CACHING ALLOWED, BOTH BY PRIVATE & SHARED CACHES**

**PRIVATE  
CACHE**

**PUBLIC  
CACHE**



**USER**

**PROXY**

**SERVER**

Cache-Control: private



Cache-Control: private

**CACHING ALLOWED, BUT ONLY BY PRIVATE CACHES**

Cache-Control: private=Set-Cookie

Cache-Control: private=Set-Cookie

**CACHING ALLOWED BY ALL CACHES, UNLESS A SET-COOKIE  
HEADER IS SET. THEN THE RESPONSE IS ONLY HANDLED BY  
PRIVATE CACHES**

Cache-Control: public, max-age=100

Cache-Control: public, max-age=100

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 100 SECONDS.**

Cache-Control: private, max-age=100

Cache-Control: private, max-age=100

**ONLY PRIVATE CACHES ALLOWED.  
CONTENT IS FRESH FOR 100 SECONDS.**

Cache-Control: public, s-maxage=100



Cache-Control: public, s-maxage=100

**ALL CACHES ALLOWED.  
CONTENT IN SHARED CACHES IS FRESH FOR 100 SECONDS.**

Cache-Control: public, max-age=60, s-maxage=100

Cache-Control: public, max-age=60, s-maxage=100

**ALL CACHES ALLOWED.**

**CONTENT IN PRIVATE CACHES IS FRESH FOR 60 SECONDS.**

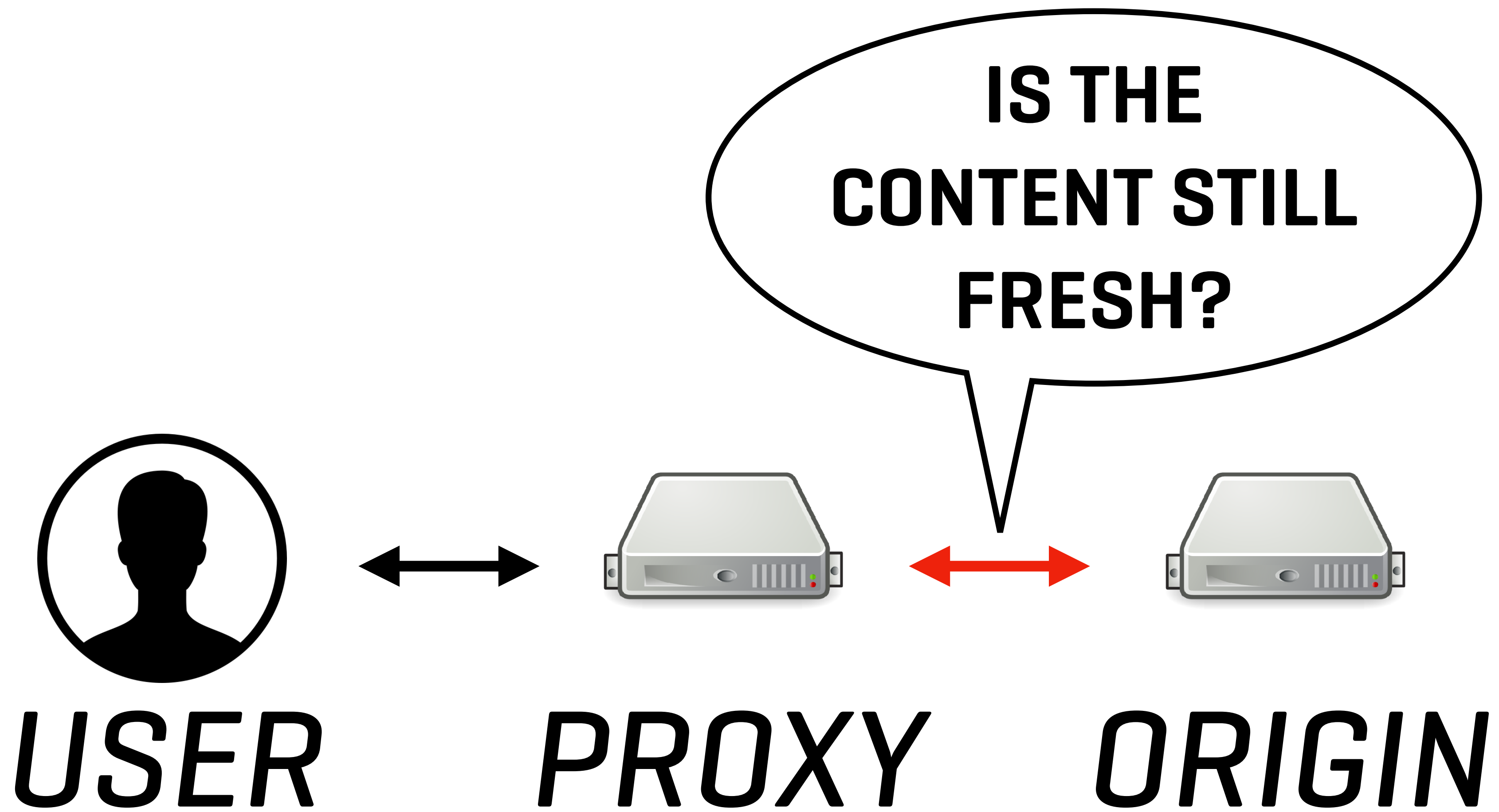
**CONTENT IN SHARED CACHES IS FRESH FOR 100 SECONDS.**

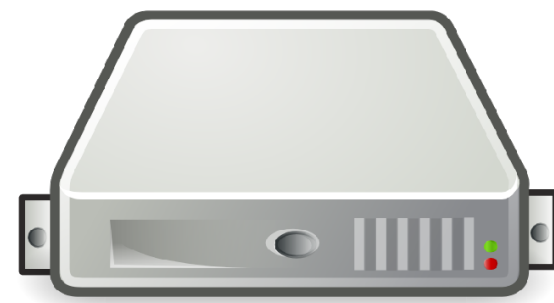
Cache-Control: public, max-age=60  
Age: 40

**AGE HEADER DESCRIBES THE TIME IN SECONDS  
THE OBJECT WAS IN A PROXY CACHE.**

Remaining TTL = TTL - Age

***REVALIDATION***



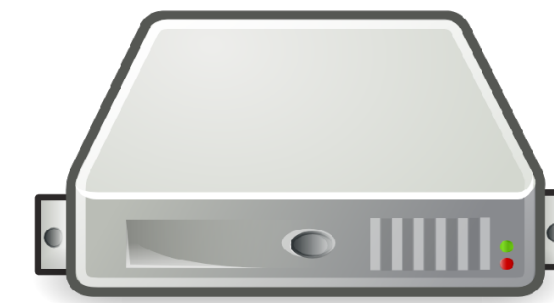


***PROXY***

GET / HTTP/1.1



HTTP/1.1 200 OK



***ORIGIN***



# ***CONDITIONAL REQUESTS***

HTTP/1.1 304 Not Modified

# *CONDITIONAL REQUESTS*

```
GET / HTTP/1.1  
Host: localhost
```

```
HTTP/1.1 200 OK  
Host: localhost  
Etag: 7c9d70604c6061da9bb9377d3f00eb27  
Content-type: text/html; charset=UTF-8
```

```
Hello world output
```

# *CONDITIONAL REQUESTS*

GET / HTTP/1.1

Host: localhost

If-None-Match: 7c9d70604c6061da9bb9377d3f00eb27

HTTP/1.1 304 Not Modified

Host: localhost

Etag: 7c9d70604c6061da9bb9377d3f00eb27

# *CONDITIONAL REQUESTS*

```
GET / HTTP/1.1  
Host: localhost
```

```
HTTP/1.1 200 OK  
Host: localhost  
Last-Modified: Fri, 22 Jul 2016 10:11:16 GMT  
Content-type: text/html; charset=UTF-8
```

```
Hello world output
```

# *CONDITIONAL REQUESTS*

GET / HTTP/1.1

Host: localhost

If-Last-Modified: Fri, 22 Jul 2016 10:11:16 GMT

HTTP/1.1 304 Not Modified

Host: localhost

Last-Modified: Fri, 22 Jul 2016 10:11:16 GMT



***VALIDATE  
QUICKLY***



***EARLY***

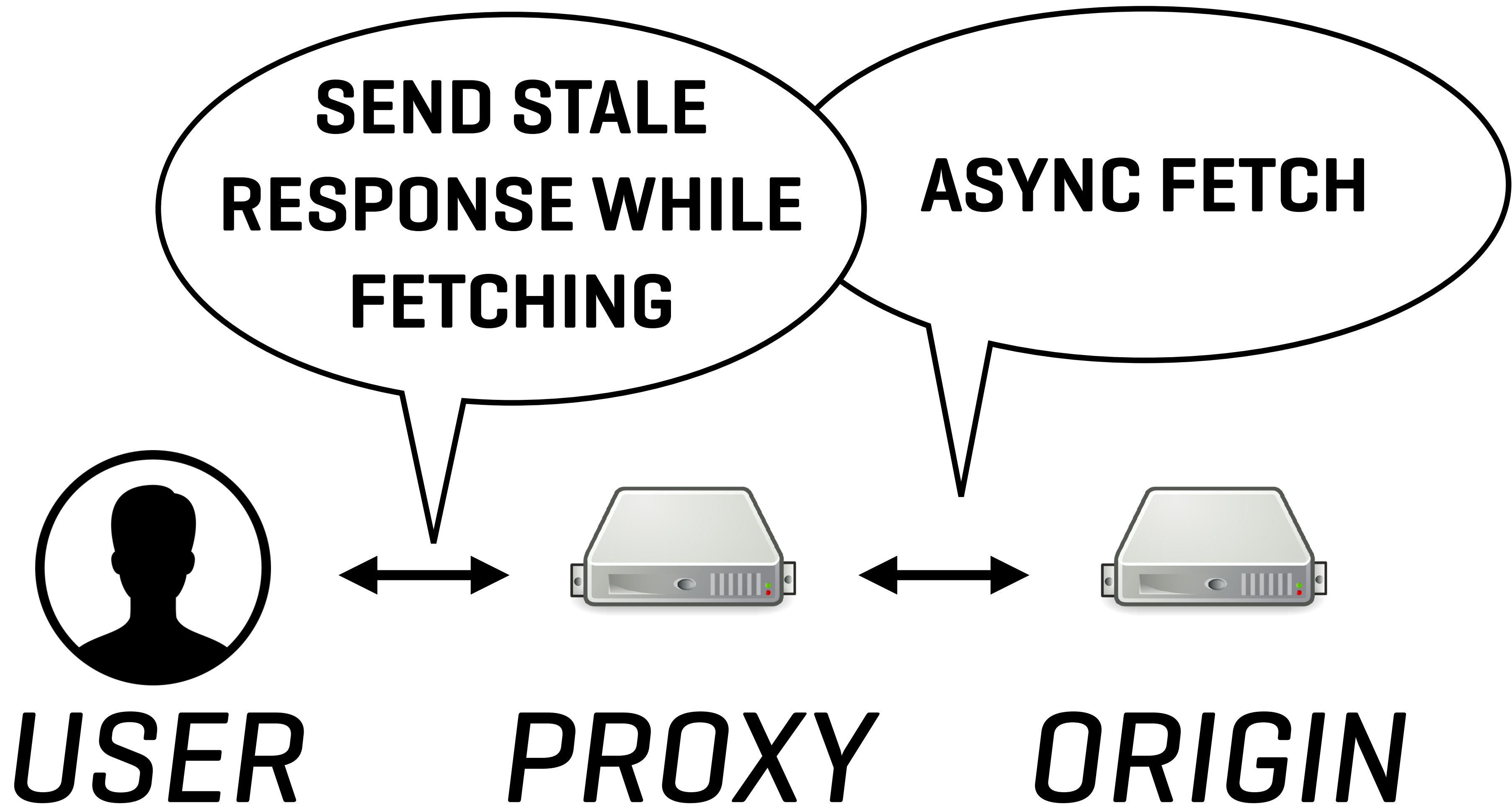




redis

***STORE &  
RETRIEVE  
ETAG***

***ASYNCHRONOUS  
REVALIDATION***



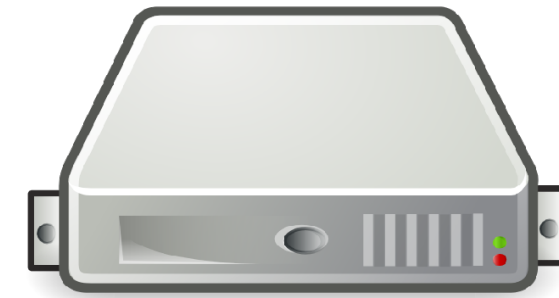
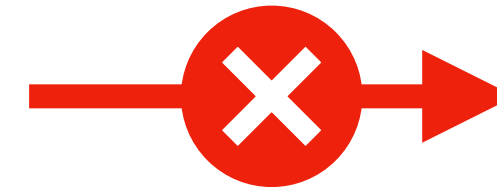
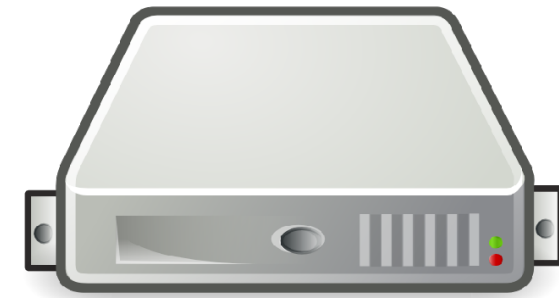
```
Cache-Control: public, max-age=3600, stale-  
while-revalidate=100
```

```
Cache-Control: public, max-age=3600, stale-while-revalidate=100
```

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 1 HOUR.  
STALE CONTENT CAN BE SERVED UP TO 100 SECONDS PAST  
THE TTL WHILE DOING AN ASYNCHRONOUS REVALIDATION.**

**SEND STALE  
RESPONSE WHILE  
FETCHING FAILS**

**FETCH FAILED**



***USER***

***PROXY***

***ORIGIN***

```
Cache-Control: public, max-age=3600, stale-  
if-error=86400
```

Cache-Control: public, max-age=3600, stale-  
if-error=86400

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 1 HOUR.  
STALE CONTENT CAN BE SERVED UP TO 1 DAY PAST THE TTL  
WHILE THE ORIGIN IS UNREACHABLE.**



Fresh =  $TTL > 0$

Async revalidation =  $TTL + stale > 0$

Synchronous revalidation =  $TTL + stale \leq 0$



**REVALIDATION  
CAN BE DONE  
CONDITIONALLY**

Fresh =  $TTL > 0$

Async revalidation =  $TTL + stale > 0$

Synchronous revalidation =  $TTL + stale \leq 0$

Cache-Control: public, max-age=3600, must-revalidate

Cache-Control: public, max-age=3600, must-revalidate

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 1 HOUR.  
SERVING STALE CONTENT NOT ALLOWED.**

Cache-Control: public, max-age=3600, proxy-revalidate

Cache-Control: public, max-age=3600, proxy-revalidate



**SAME AS MUST-  
REVALIDATE BUT FOR  
PROXY SERVERS**

Cache-Control: public, max-age=86400, immutable

Cache-Control: public, max-age=86400, immutable

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 1 DAY.  
CONTENT WILL NOT BE UPDATED WHILE FRESH**

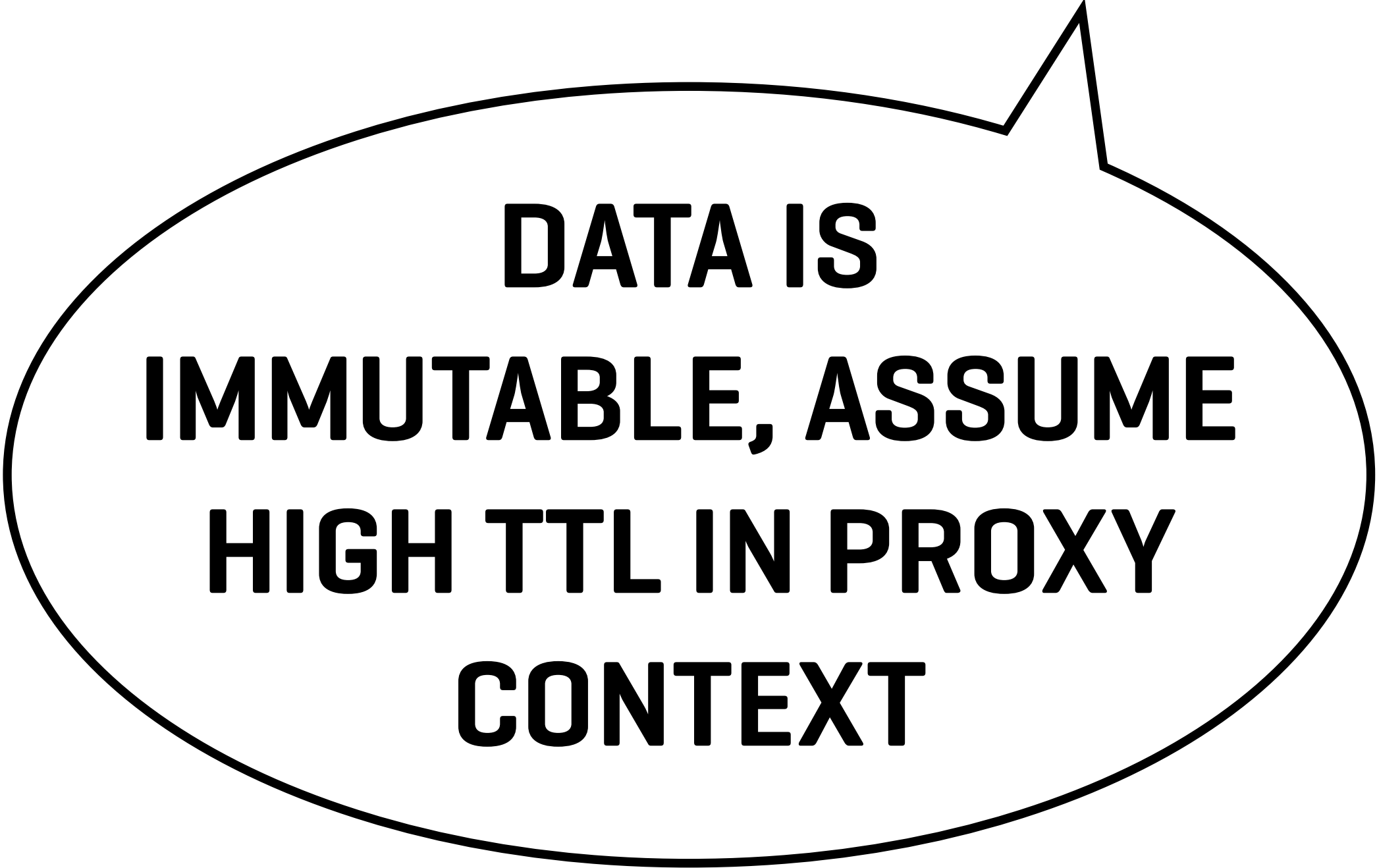


**USEFUL FOR  
BROWSER CACHING**

Cache-Control: public, max-age=86400, immutable

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 1 HOUR.  
CONTENT WILL NOT BE UPDATED WHILE FRESH**

Cache-Control: public, immutable



**DATA IS  
IMMUTABLE, ASSUME  
HIGH TTL IN PROXY  
CONTEXT**

Cache-Control: public, max-age=3600, no-transform

Cache-Control: public, max-age=3600, no-transform

**ALL CACHES ALLOWED.  
CONTENT IS FRESH FOR 1 HOUR.  
CONTENT CANNOT BE TRANSFORMED  
BY INTERMEDIARY CACHES**

Cache-Control: public, max-age=3600, no-transform



**EDGE COMPUTE**

Cache-Control: no-cache

Cache-Control: no-cache

**STORE OBJECT IN CACHE  
BUT REVALIDATE BEFORE EVERY REUSE**

Cache-Control: no-cache=Set-Cookie

**STORE OBJECT IN CACHE  
BUT REVALIDATE BEFORE EVERY REUSE  
IF THE SET-COOKIE HEADER IS SET**



Cache-Control: no-store

Cache-Control: no-store

**DON'T STORE OBJECT IN THE CACHE**

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

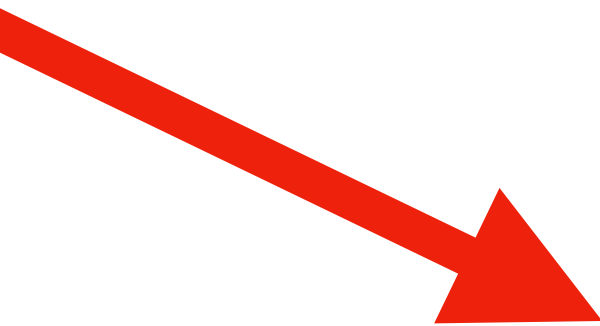


**TYPICAL ONE**

# ***CACHE VARIATIONS***

Vary: Accept-Language

```
GET / HTTP/1.1  
Host: test.com  
Accept-Language: fr
```



```
GET / HTTP/1.1  
Host: test.com  
Accept-Language: en
```



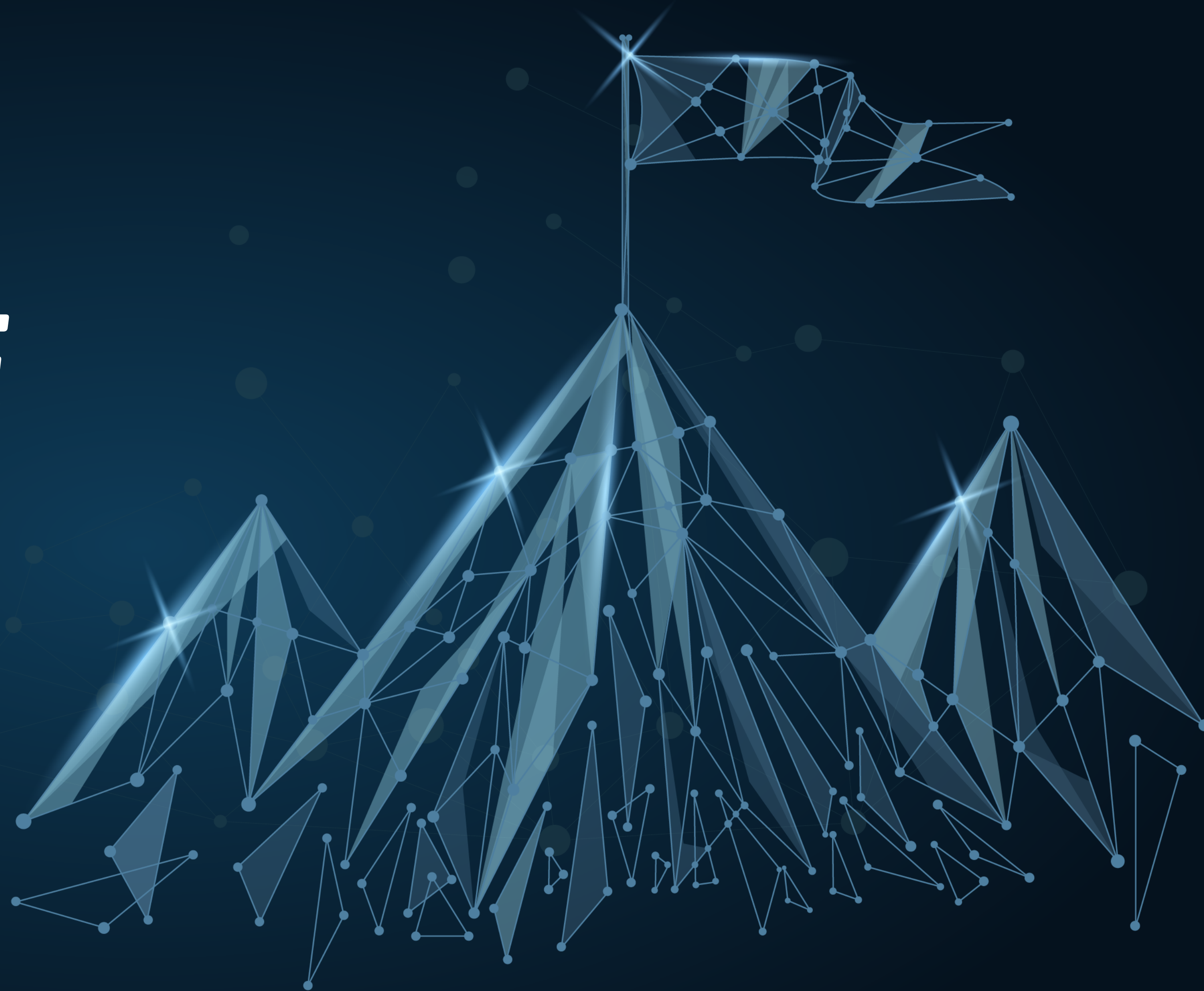
```
▶ http://test.com/  
-Accept-Language: fr  
-Accept-Language: nl  
-Accept-Language: en
```

Vary: Accept-Encoding, Accept-Language,  
X-Forwarded-Proto

***SURROGATES***



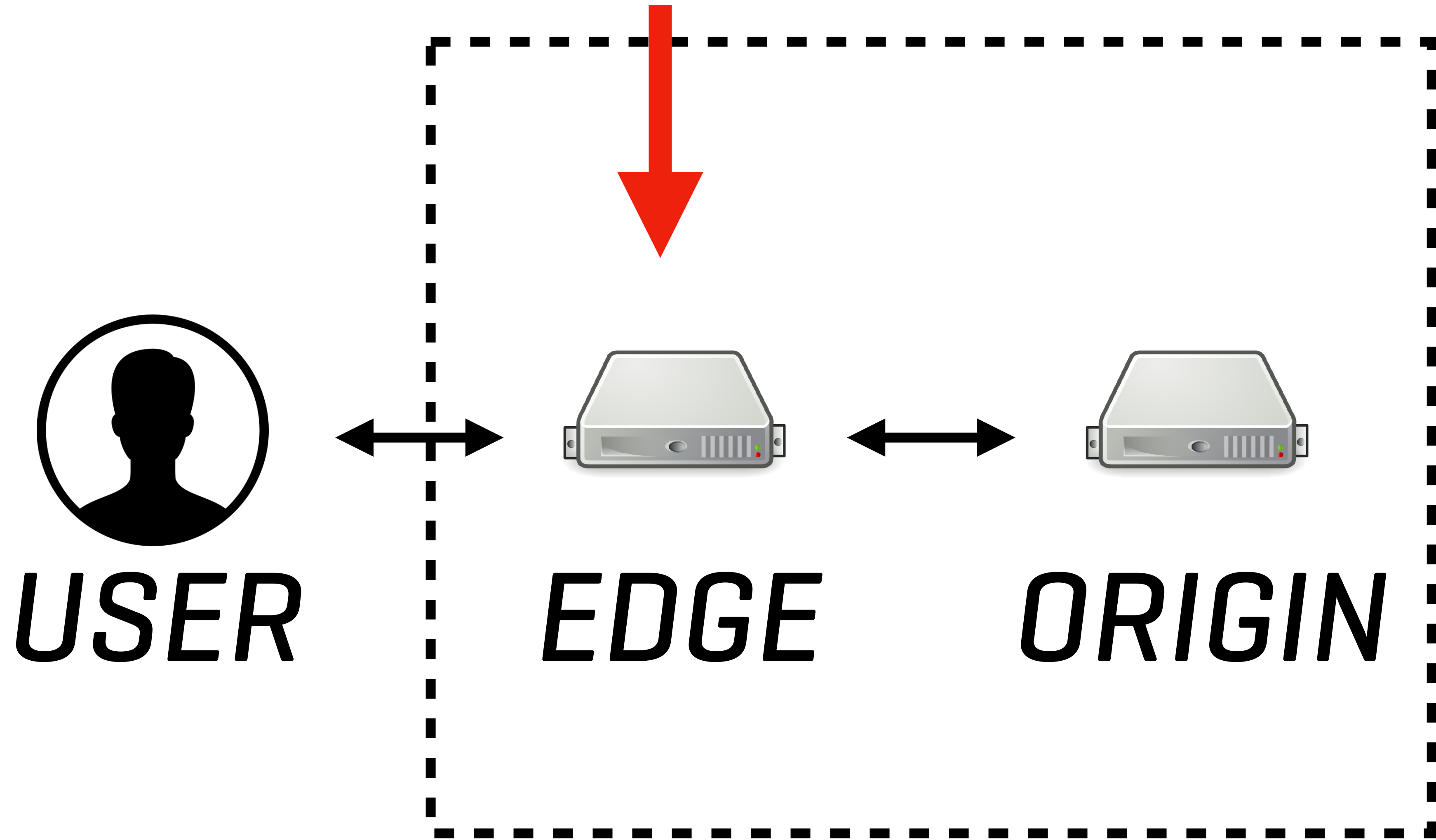
# ***THE EDGE***



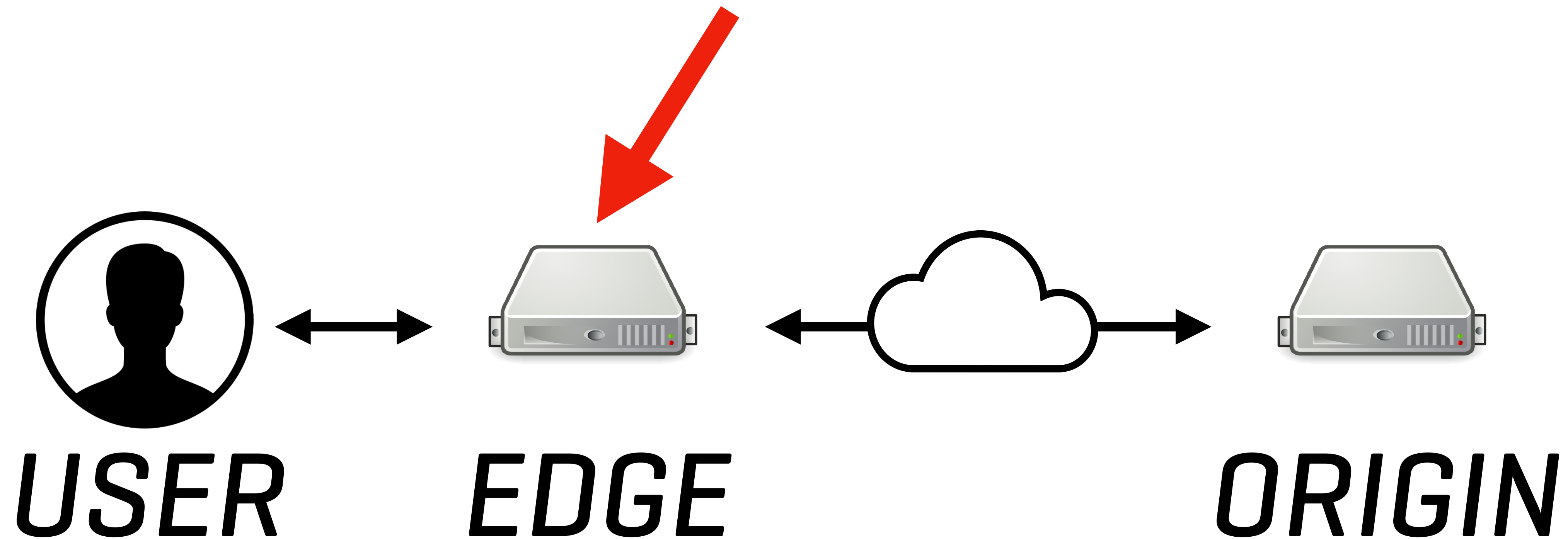
***CONTENT  
DELIVERY  
NETWORK***



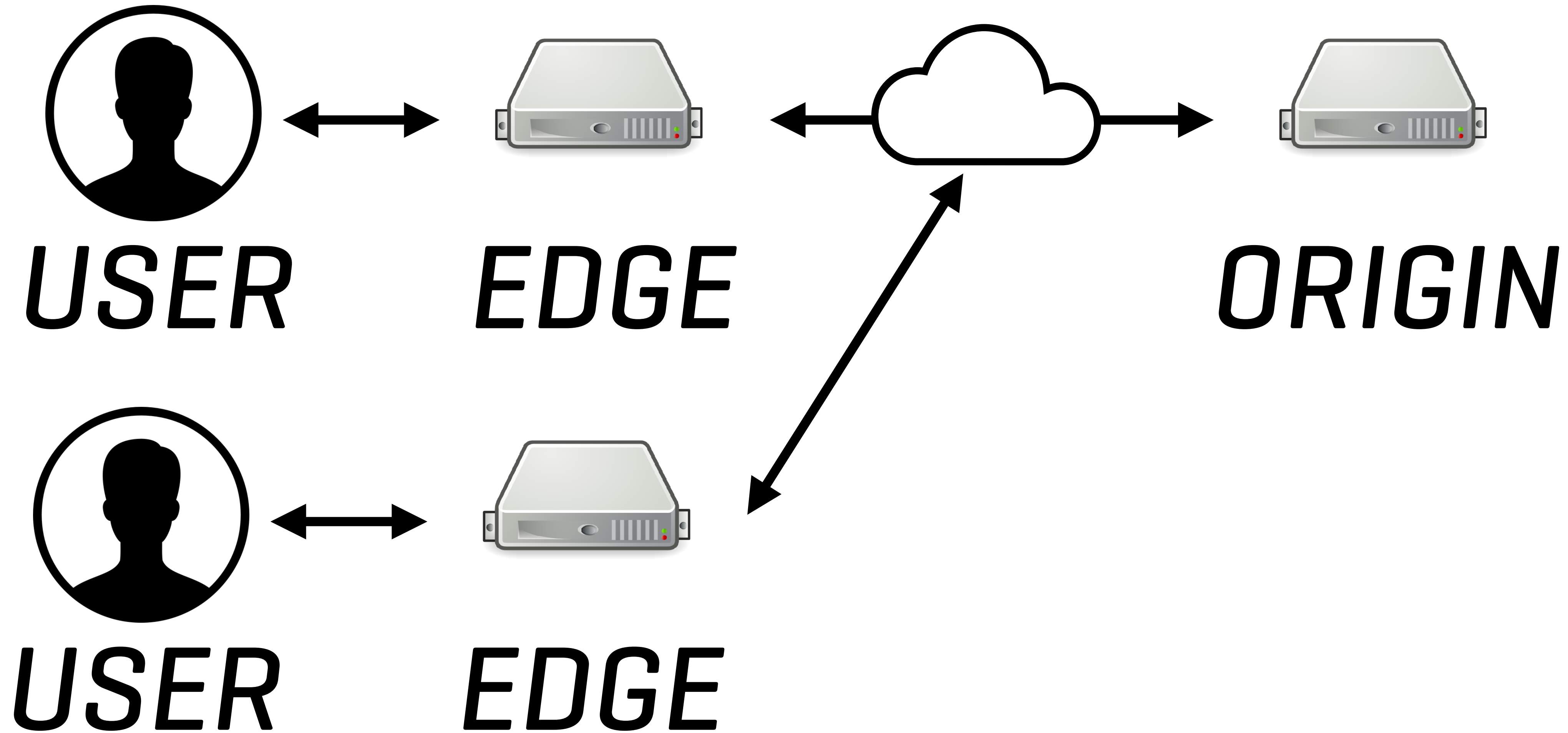
# ***THE EDGE IS NO LONGER IN THE ORIGIN DATA CENTER***



# ***THE EDGE MOVES CLOSER TO THE END USER***



# ***MULTIPLE EDGES***



Surrogate-Control: max-age=300

Surrogate-Control: max-age=300+100

Surrogate-Control: max-age=300+100



**STALENESS**



Surrogate-Control: no-store

```
Surrogate-Control: no-store-remote,  
max-age=3600
```

***SURROGATE CAPABILITY***

Surrogate-Capability: key="ESI/1.0"

Surrogate-Control: content="ESI/1.0"

Surrogate-Capability: `varnish="ESI/1.0"`

```
Surrogate-Control: max-age=60, max-age=86400;varnish, max-age=3600;cdn, content="ESI/1.0";varnish
```

Header

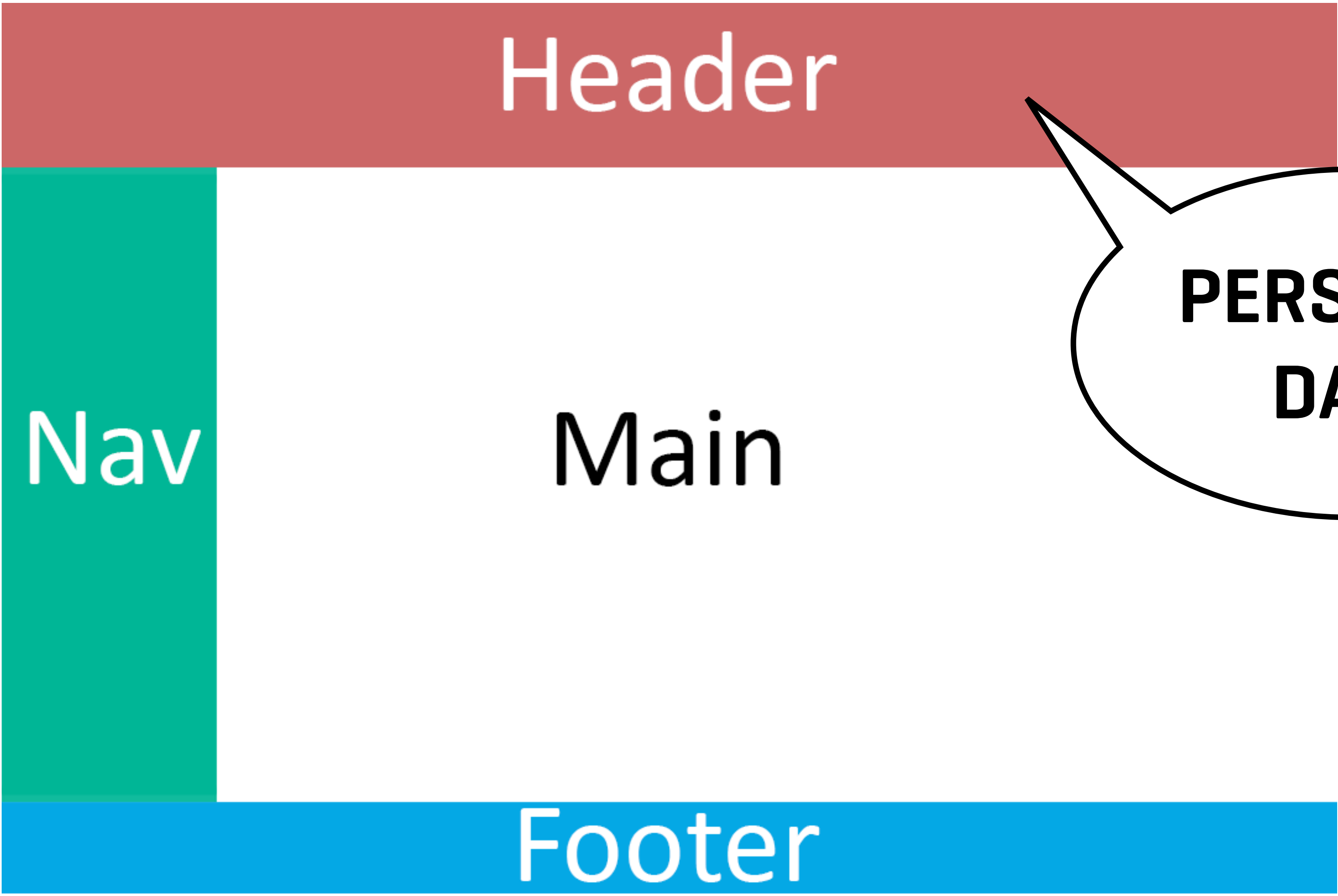
The diagram illustrates a standard web page layout. It consists of four main sections: a top header bar, a left navigation bar, a central main content area, and a bottom footer bar. The header bar is red, the navigation bar is green, the main content area is white, and the footer bar is blue. The text 'Header', 'Nav', 'Main', and 'Footer' is centered within their respective sections.

Nav

Main

Footer





Header

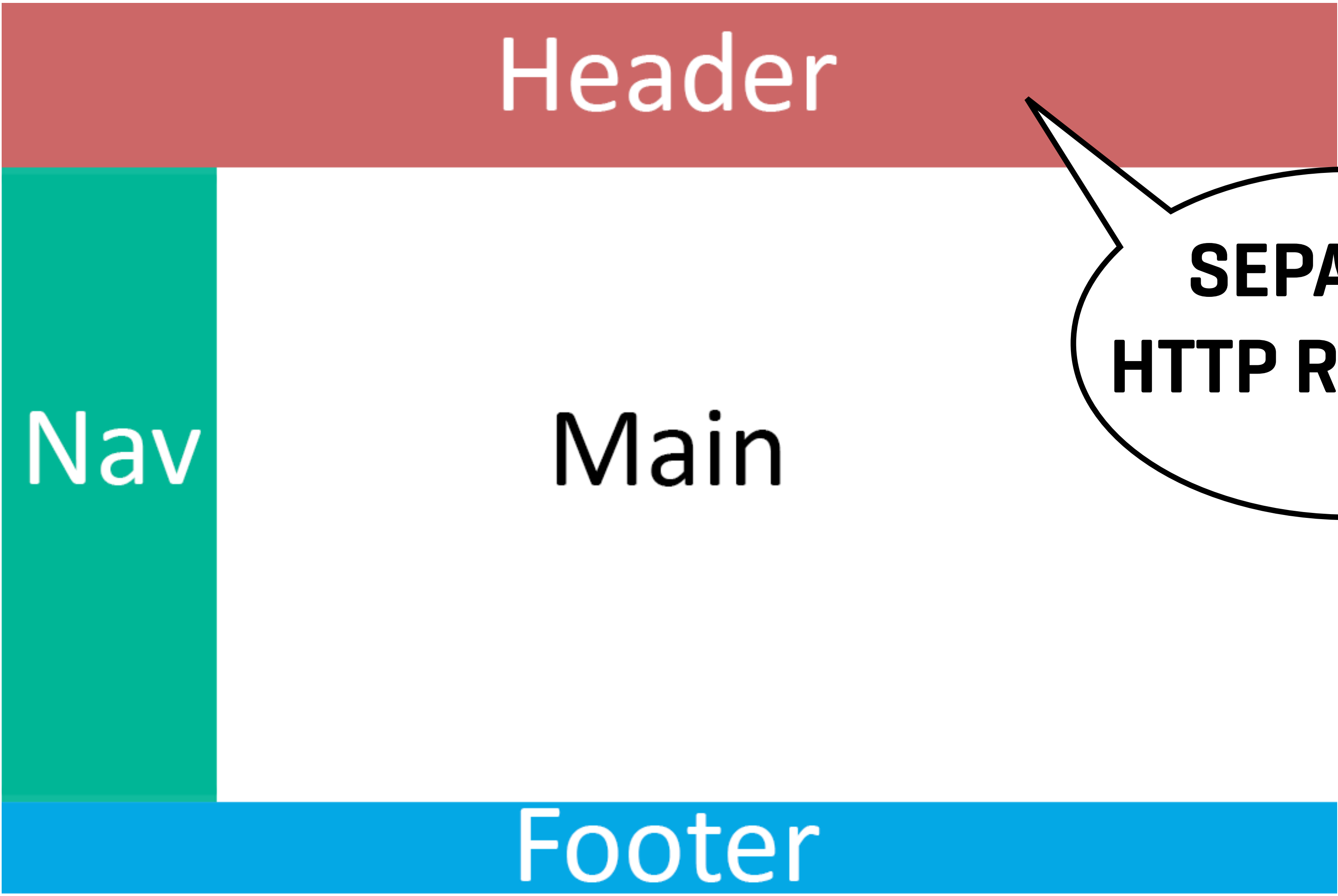
Nav

Main

Footer



**PERSONAL  
DATA**



Header

Nav

Main

Footer

**SEPARATE  
HTTP REQUEST**

**AJAX**

***EDGE-SIDE INCLUDES*** *ESI*

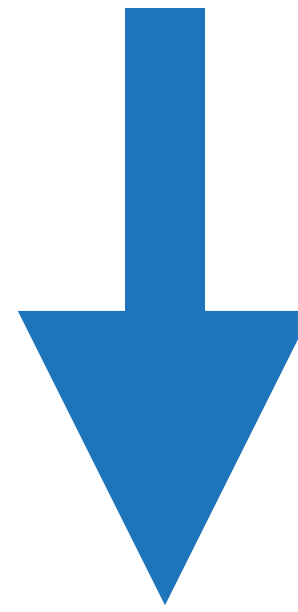
```
<esi:include src="/header" />
```

# ESI

- ✓ PLACEHOLDER
- ✓ PARSED BY EDGE CACHE *[VARNISH]*
- ✓ OUTPUT IS A COMPOSITION OF BLOCKS
- ✓ STATE PER BLOCK
- ✓ TTL PER BLOCK

***EDGE***

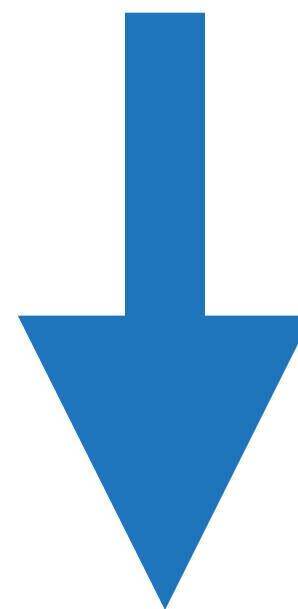
Surrogate-Capability: key="ESI/1.0"



***ORIGIN***

Surrogate-Control: content="ESI/1.0"

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



***EDGE***

Parse ESI placeholders

```
<!DOCTYPE html>
<html>
  <body>
    <esi:include src="/header" />
    <p>Welcome</p>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <body>
    <p>The current time is 21:07:53.</p>
    <p>Welcome</p>
  </body>
</html>
```

***THIS IS JUST THE TIP OF THE ICEBERG***

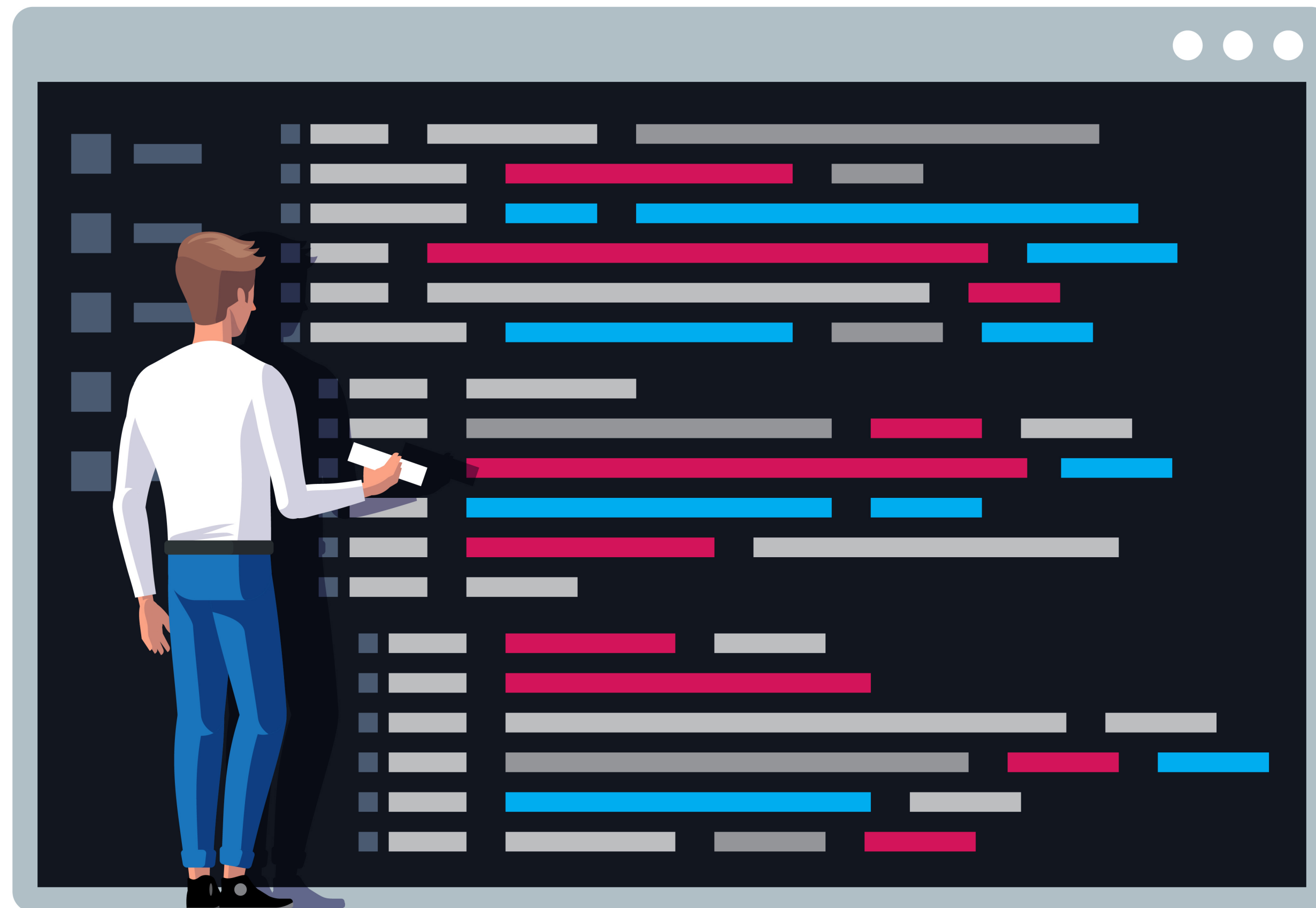


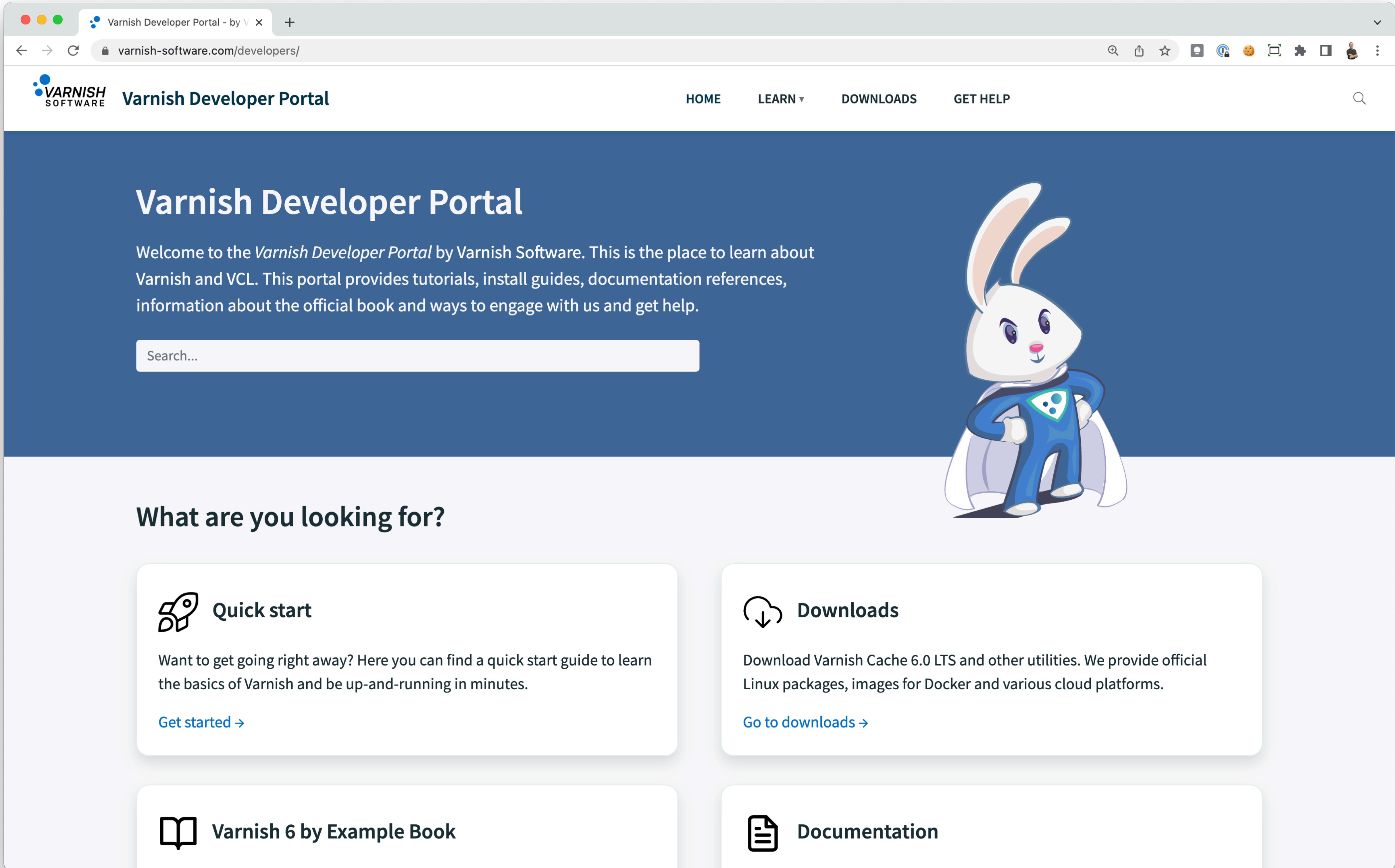
**REMEMBER  
THIS ONE?**

***EVERY  
IMPLEMENTATION  
HAS ITS OWN  
CACHE POLICY  
CONFIGURATION***



# ***VARNISH CONFIGURATION LANGUAGE***





# Varnish Developer Portal

Welcome to the *Varnish Developer Portal* by Varnish Software. This is the place to learn about Varnish and VCL. This portal provides tutorials, install guides, documentation references, information about the official book and ways to engage with us and get help.



## What are you looking for?

### Quick start

Want to get going right away? Here you can find a quick start guide to learn the basics of Varnish and be up-and-running in minutes.

[Get started →](#)

### Downloads

Download Varnish Cache 6.0 LTS and other utilities. We provide official Linux packages, images for Docker and various cloud platforms.

[Go to downloads →](#)

### Varnish 6 by Example Book

### Documentation

***[HTTPS://VARNISH-SOFTWARE.COM/DEVELOPERS](https://varnish-software.com/developers)***

Thijs Feryn

# VARNISH 6

---

BY EXAMPLE

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



THE END

