

# **BUILD YOUR OWN CDN WITH VARNISH**

**BY THIJS FERYN**



**ConFoo.ca**  
WEB TECHNO CONFERENCE



*Slow websites*  
**SUCK**

# WEB PERFORMANCE IS AN ESSENTIAL PART OF THE USER EXPERIENCE



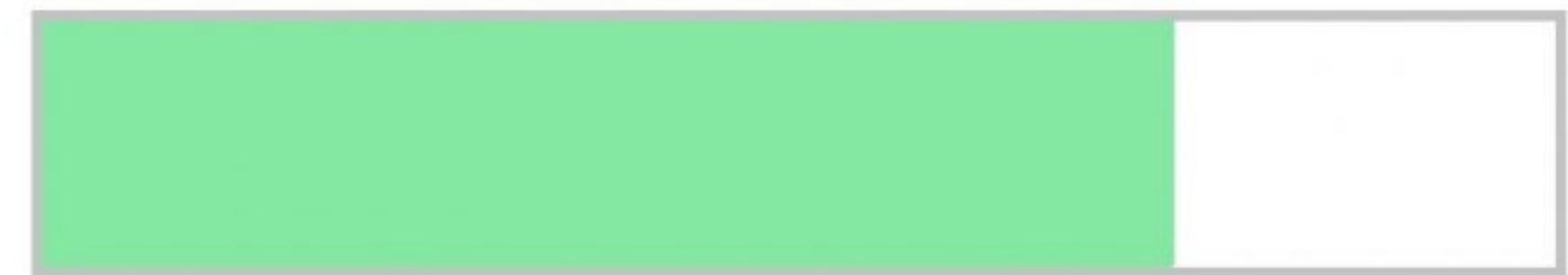
# SLOW~DOWN







Google





**THROWING  
SERVERS  
AT THE PROBLEM**

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



# **IDENTIFY SLOWEST PARTS**



# OPTIMIZE

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



# CACHE



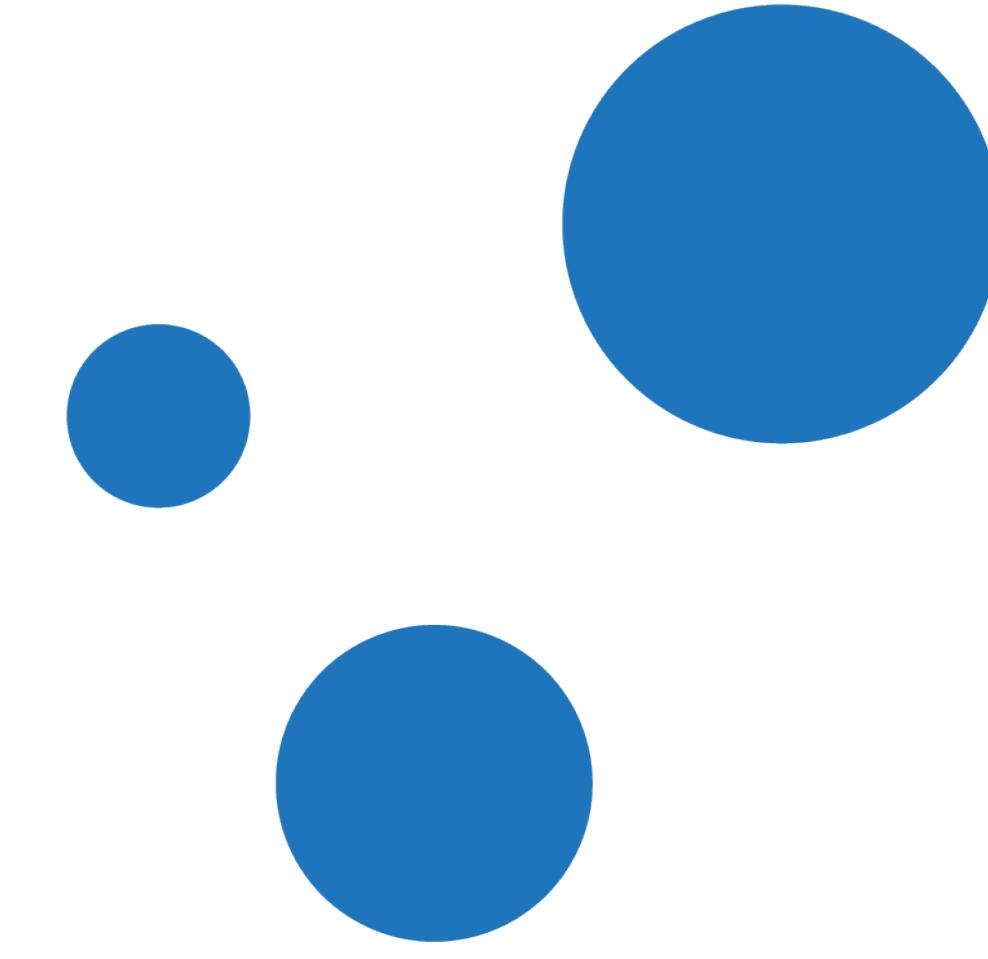
**HI,**



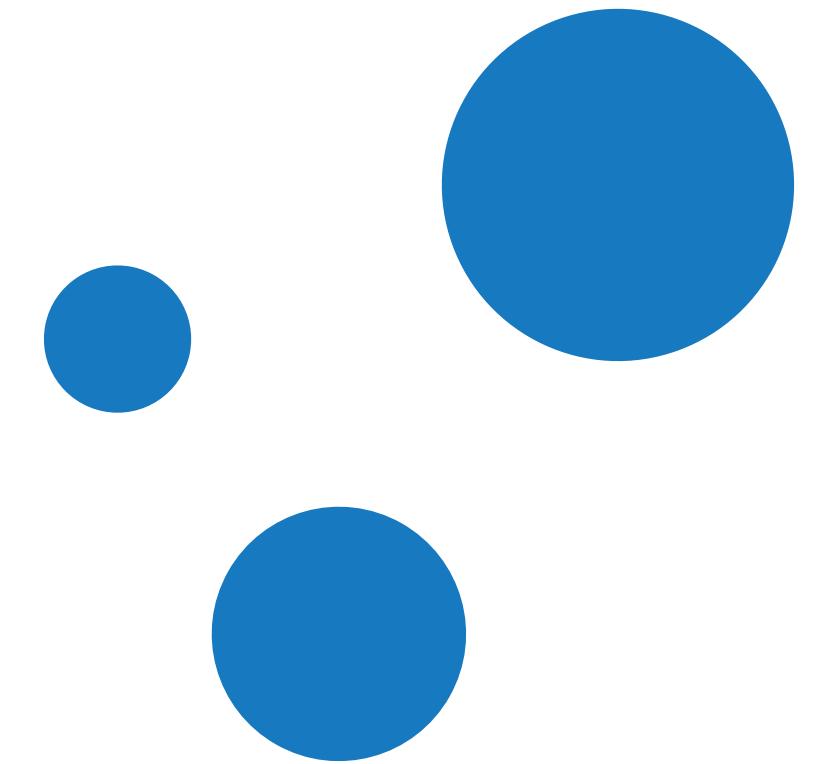
**I'M THIJS**



**I'M THE TECH  
EVANGELIST  
AT • VARNISH  
SOFTWARE**



**VARNISH  
SOFTWARE**



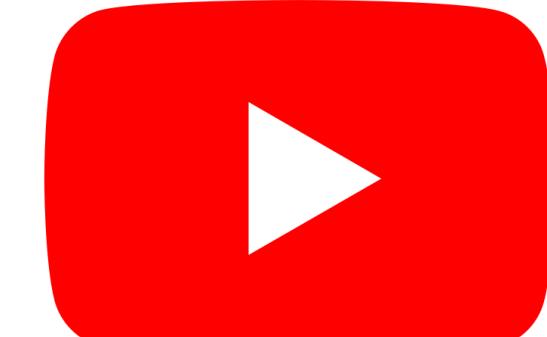
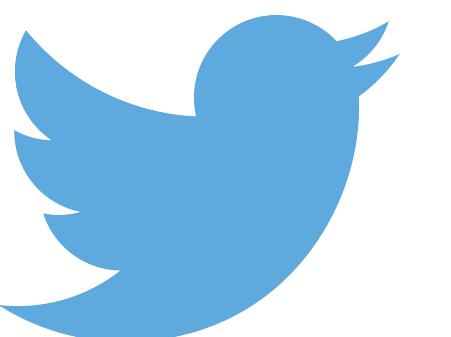
# **VARNISH CACHE**

**9,000,000 WEBSITES  
21% OF THE TOP 10K WEBSITES**

I'M



@THIJSFERYN



O'REILLY®



# Getting Started with Varnish Cache

ACCELERATE YOUR WEB APPLICATIONS

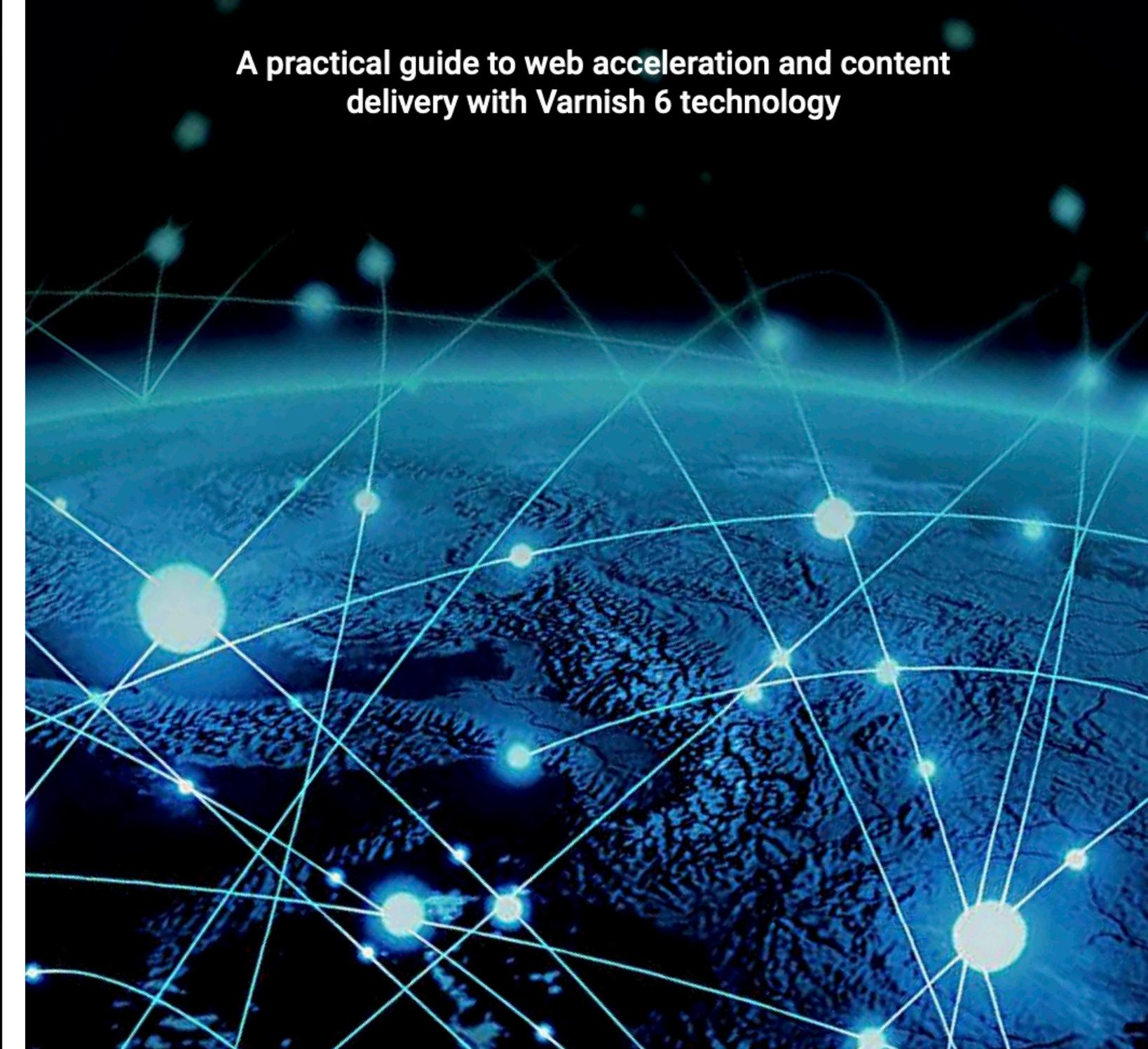
Thijs Feryn

Thijs Feryn

# VARNISH 6

BY EXAMPLE

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



# CONTENT DELIVERY CHALLENGES



# FAST



# **GLOBAL AVAILABILITY**

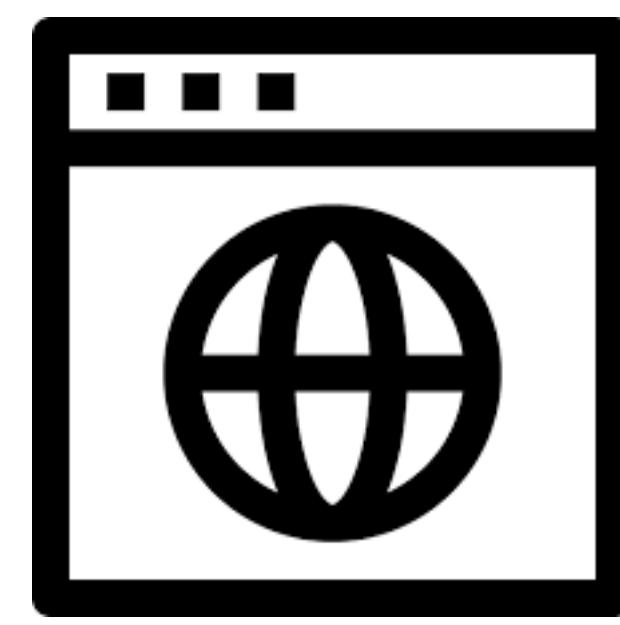
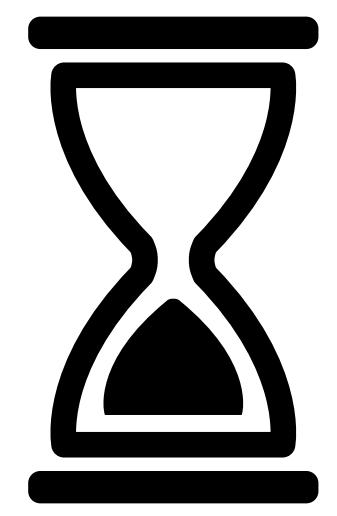


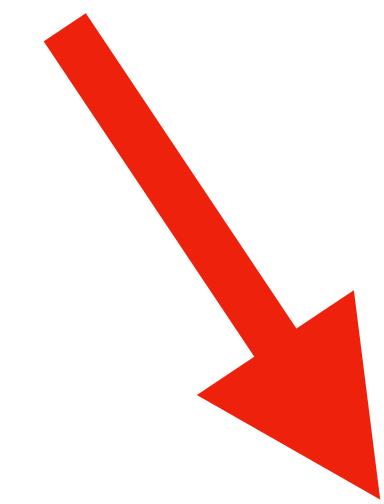
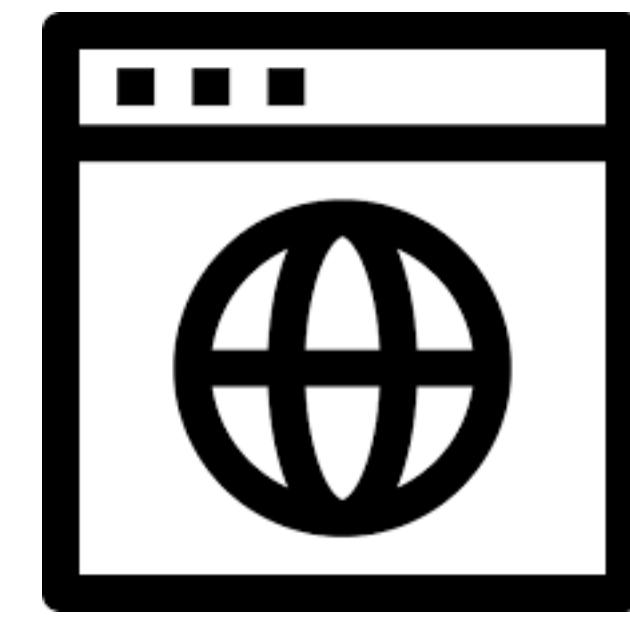
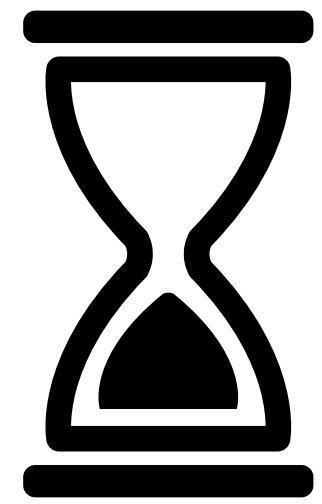
# STABLE



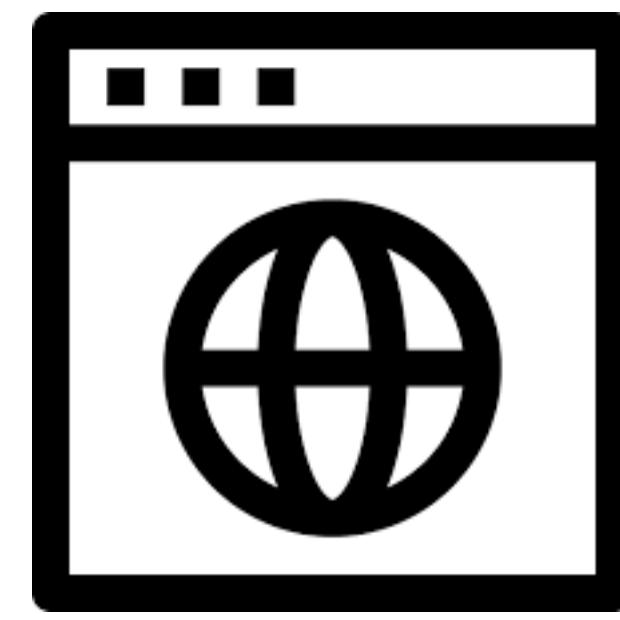
# MORE THAN JUST WEB PAGES



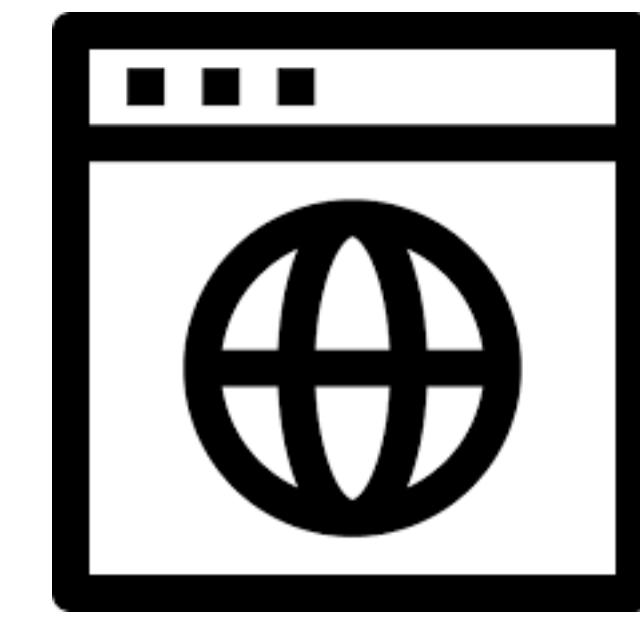
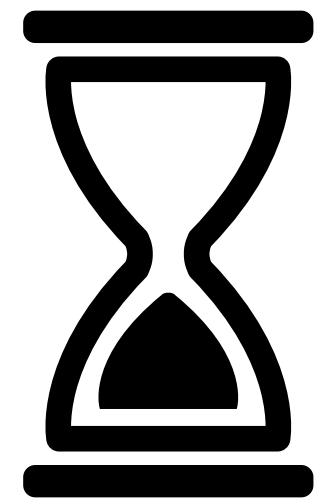




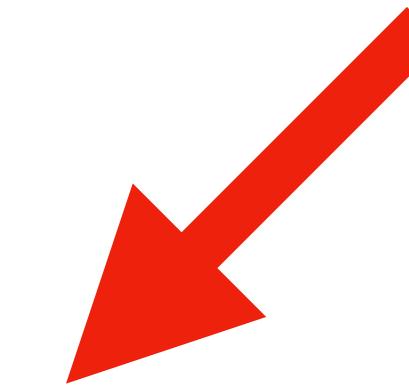
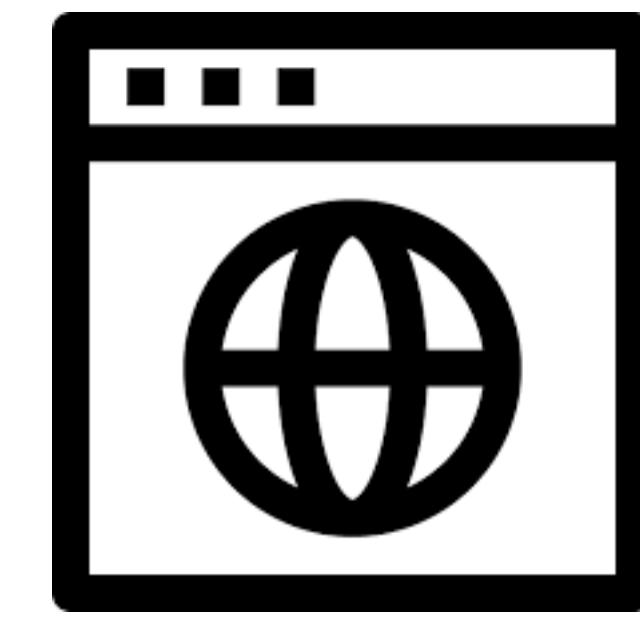
**REDUCE *TIME* TO GENERATE  
THE RESPONSE**



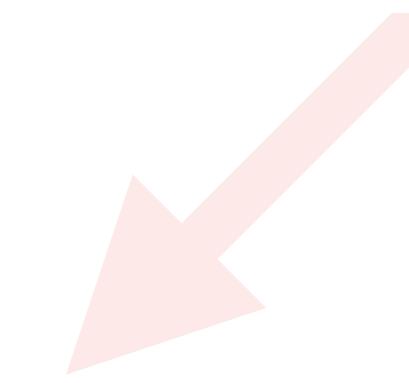
***REDUCE THE **SIZE** OF THE RESPONSE***



**REDUCE THE *DISTANCE* BETWEEN  
CLIENT & SERVER**



**OPTIMIZE *BROWSER* RENDERING**



*OPTIMIZE **BROWSER** RENDERING*

**HIGHER  
THROUGHPUT**

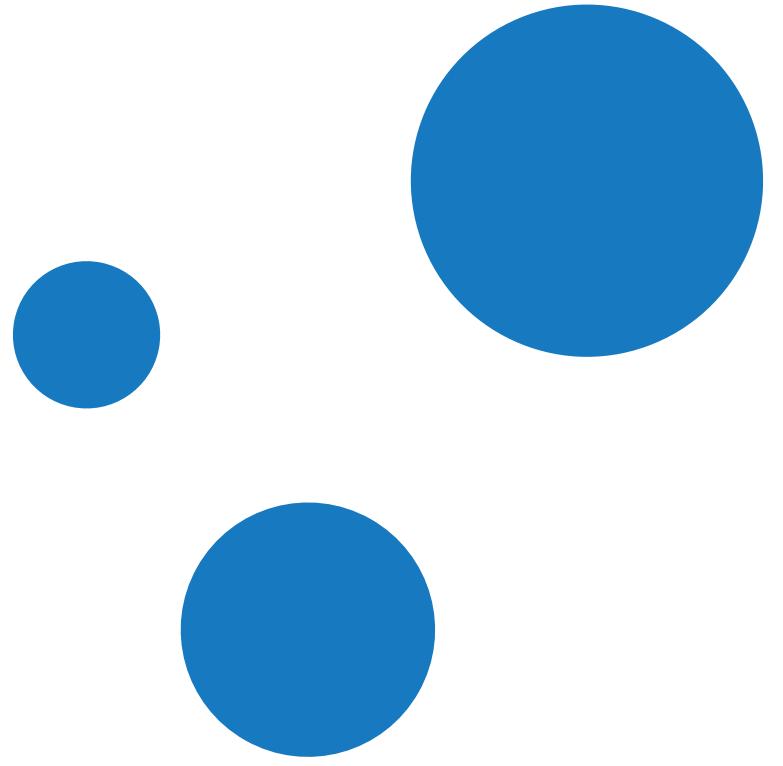


# LOWER LATENCY







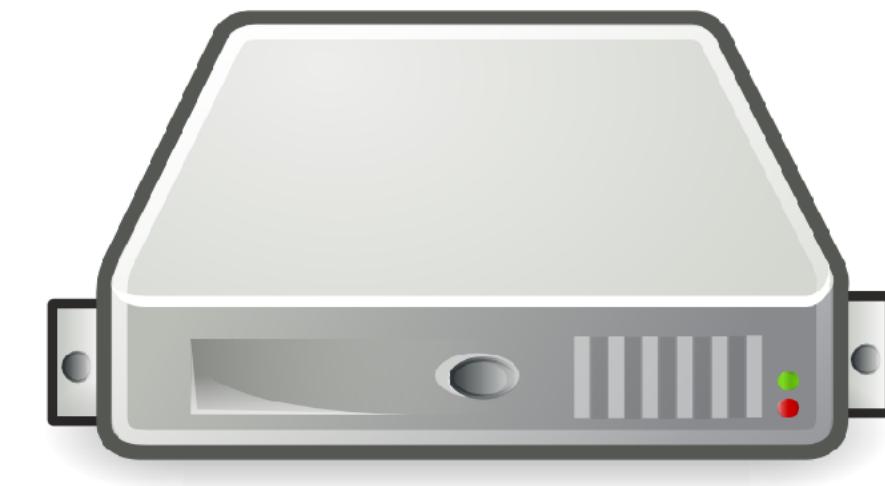


**VARNISH CACHE**

# **NORMALLY**

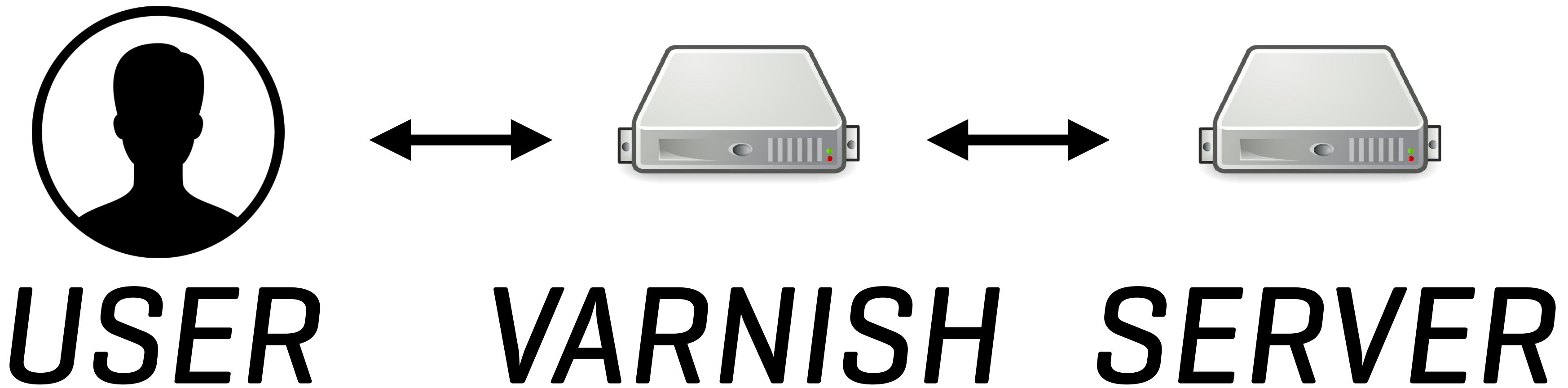


**USER**



**SERVER**

# **WITH VARNISH**



# WHY IS VARNISH SO POWERFUL?

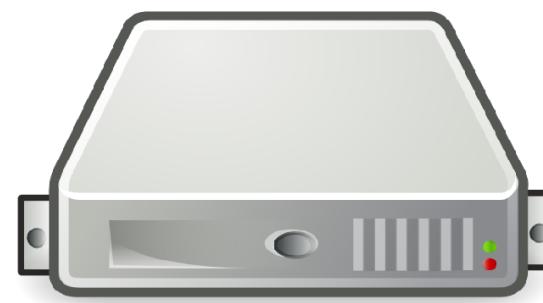


# WHY IS VARNISH SO POWERFUL?

- ✓ EXTREMELY LOW RESOURCE CONSUMPTION
- ✓ EXTREMELY STABLE
- ✓ NO DISK ACCESS AT RUNTIME
- ✓ 500 GBPS PER SERVER\*
- ✓ CONTENT STREAMING
- ✓ REAL-TIME CONTENT INVALIDATION
- ✓ REQUEST COALESCING
- ✓ VARNISH CONFIGURATION LANGUAGE
- ✓ VARNISH MODULES (VMODS)
- ✓ COMPLIES TO HTTP BEST PRACTICES

SINGLE BACKEND  
FETCH

QUEUED  
CONNECTIONS

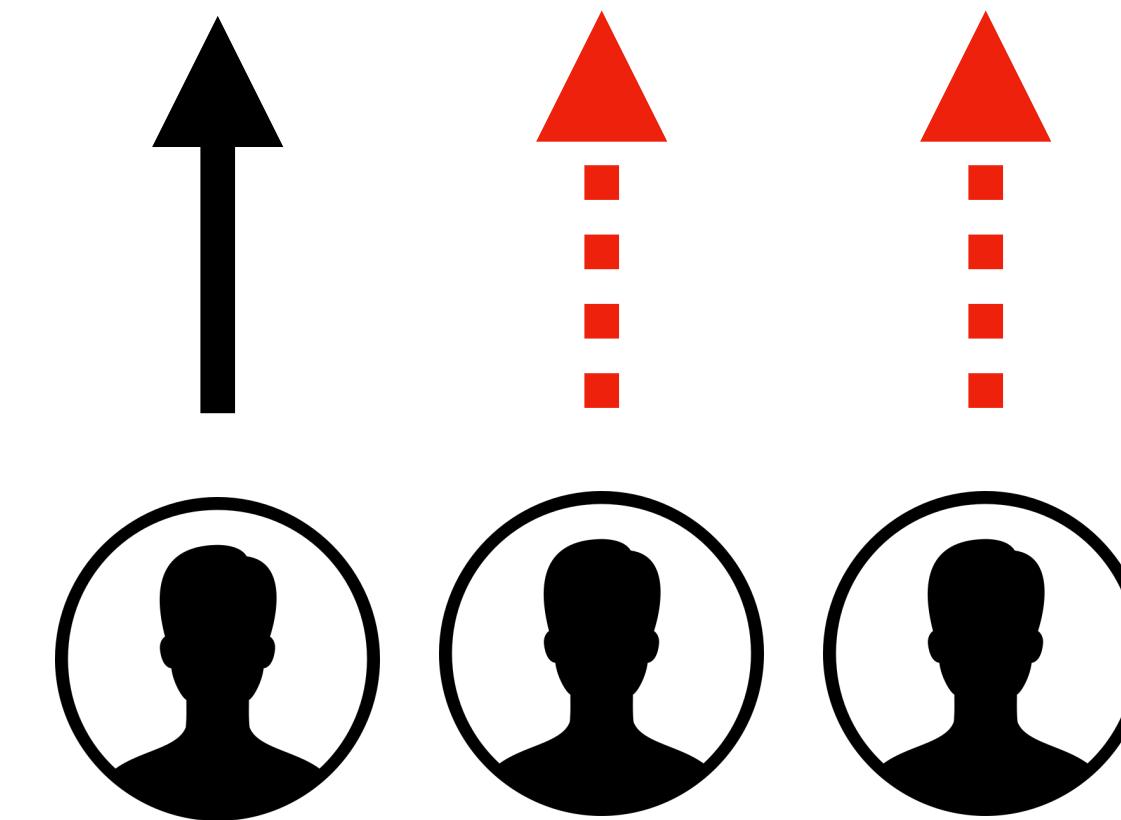


**ORIGIN**



**REQUEST  
COALESCING**

VARNISH



# ***CONTROLLING THE CACHE WITH HTTP***

Expires: Wed, 26 Feb 2020 23:30:00 GMT

Cache-control: public, max-age=3600, s-maxage=86400

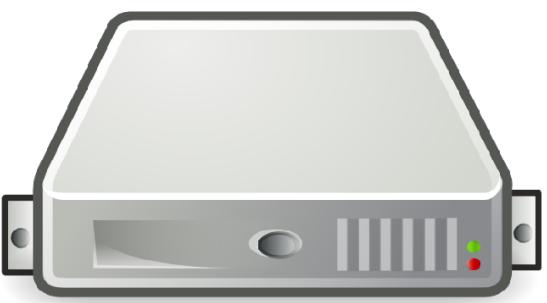
Cache-control: private, no-cache, no-store

Vary: Accept-Language

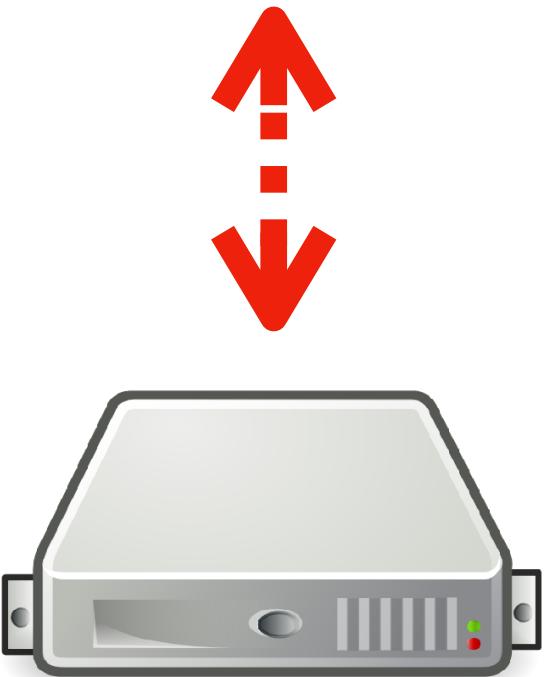
# **GRACE MODE**

Cache-control:public, max-age=3600,  
**stale-while-revalidate=200**

BACKGROUND  
FETCH

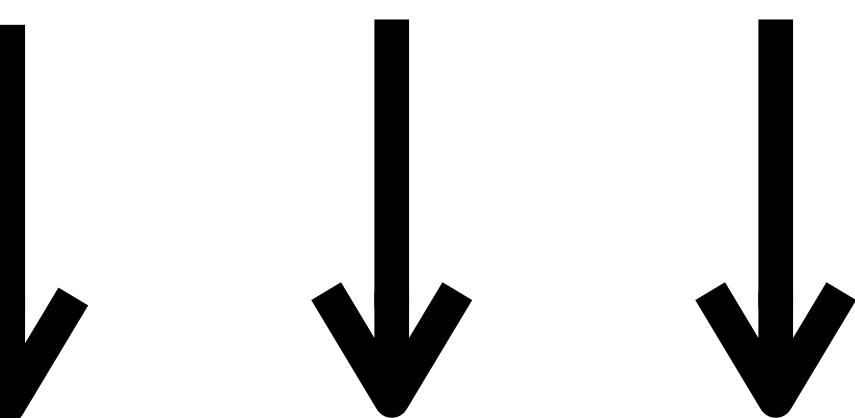


**ORIGIN**



**GRACE**

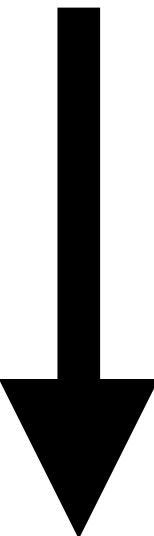
SERVE STALE  
OBJECT



# **CONDITIONAL REVALIDATION**

GET / HTTP/1.1

Host: example.com



HTTP/1.1 200 OK

Cache-control: max-age=3600

Etag: 87719aa72fb32c718001e747c75a93fd

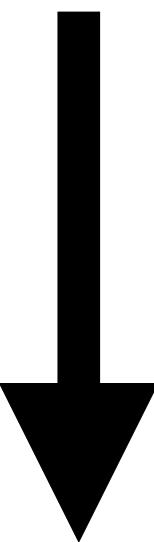
Hello world

# **CONDITIONAL REVALIDATION**

GET / HTTP/1.1

Host: example.com

If-None-Match: 87719aa72fb32c718001e747c75a93fd



HTTP/1.1 304 Not Modified

Cache-control: max-age=3600

Etag: 87719aa72fb32c718001e747c75a93fd

# **VARNISH CONFIGURATION LANGUAGE**

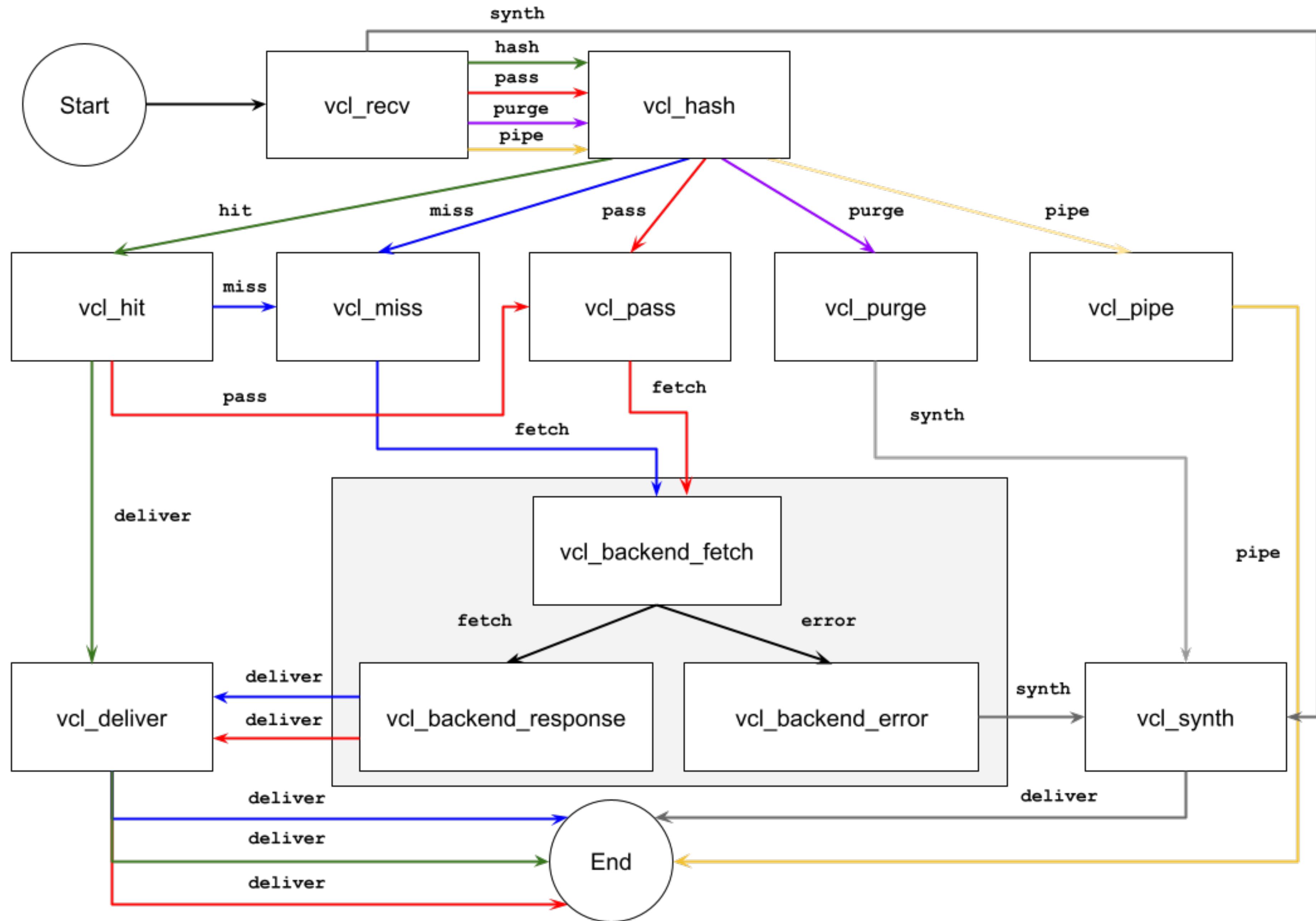


# VARNISH CONFIGURATION LANGUAGE

- ✓ DOMAIN-SPECIFIC LANGUAGE *(DSL)*
- ✓ CURLY BRACES *(LIKE C, JAVA, ...)*
- ✓ EXTENDS STANDARD BEHAVIOR
- ✓ PRE-DEFINED SUBROUTINES
- ✓ REFLECTS FINITE STATE MACHINE
- ✓ TRANSLATED INTO C-CODE WHEN VARNISH STARTS

# VCL CAPABILITIES

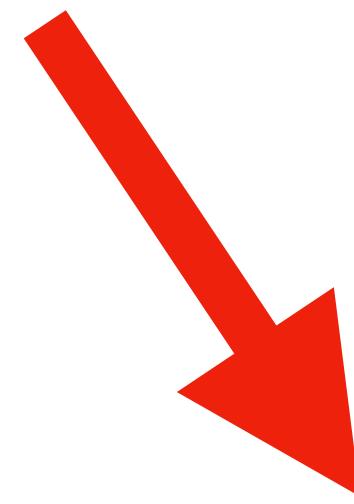
- ✓ REQUEST HANDLING
- ✓ REQUEST ROUTING
- ✓ RESPONSE MANIPULATION
- ✓ BACKEND SELECTION
- ✓ CONTROLLING THE CACHE
- ✓ DECISION-MAKING "ON THE EDGE"



```
vcl 4.1;

backend default {
    .host = "backend.example.com";
    .port = "80";
}

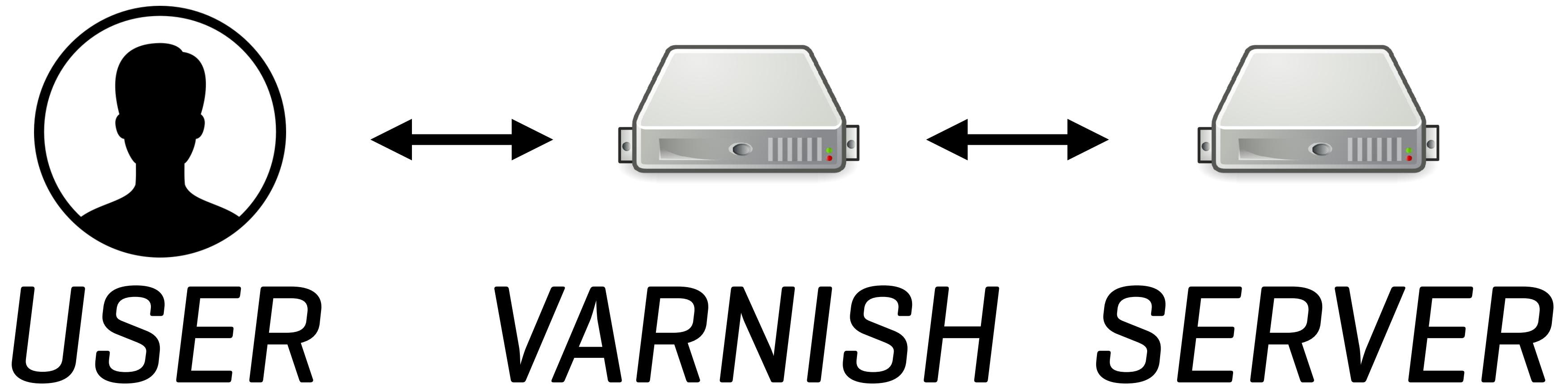
sub vcl_recv {
    if(req.url ~ "^/admin(/.*)?") {
        return(pass);
    }
}
```



**WE'VE JUST LOWERED THE  
TIME TO FIRST BYTE**

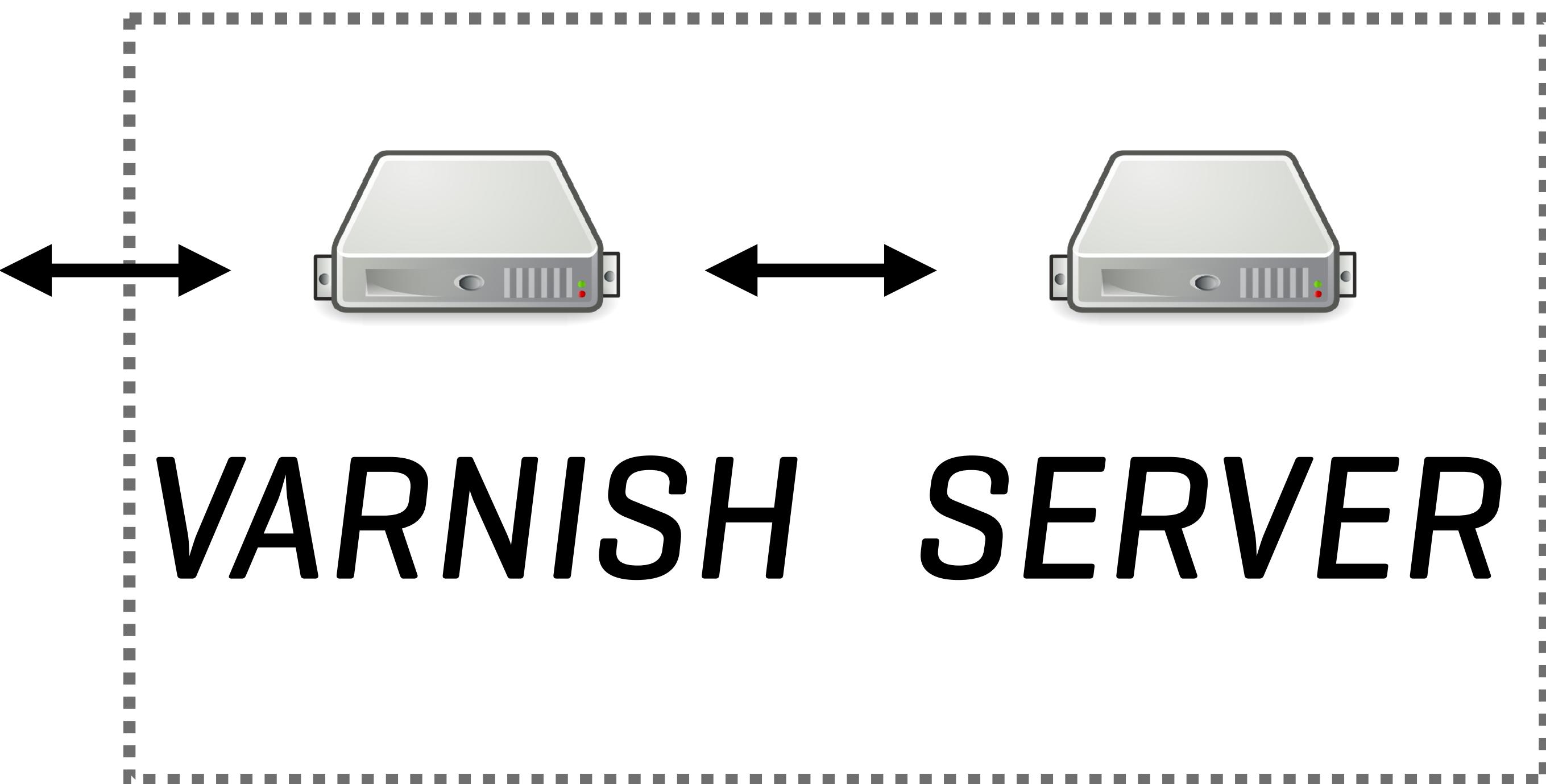
# ORIGIN SHIELD



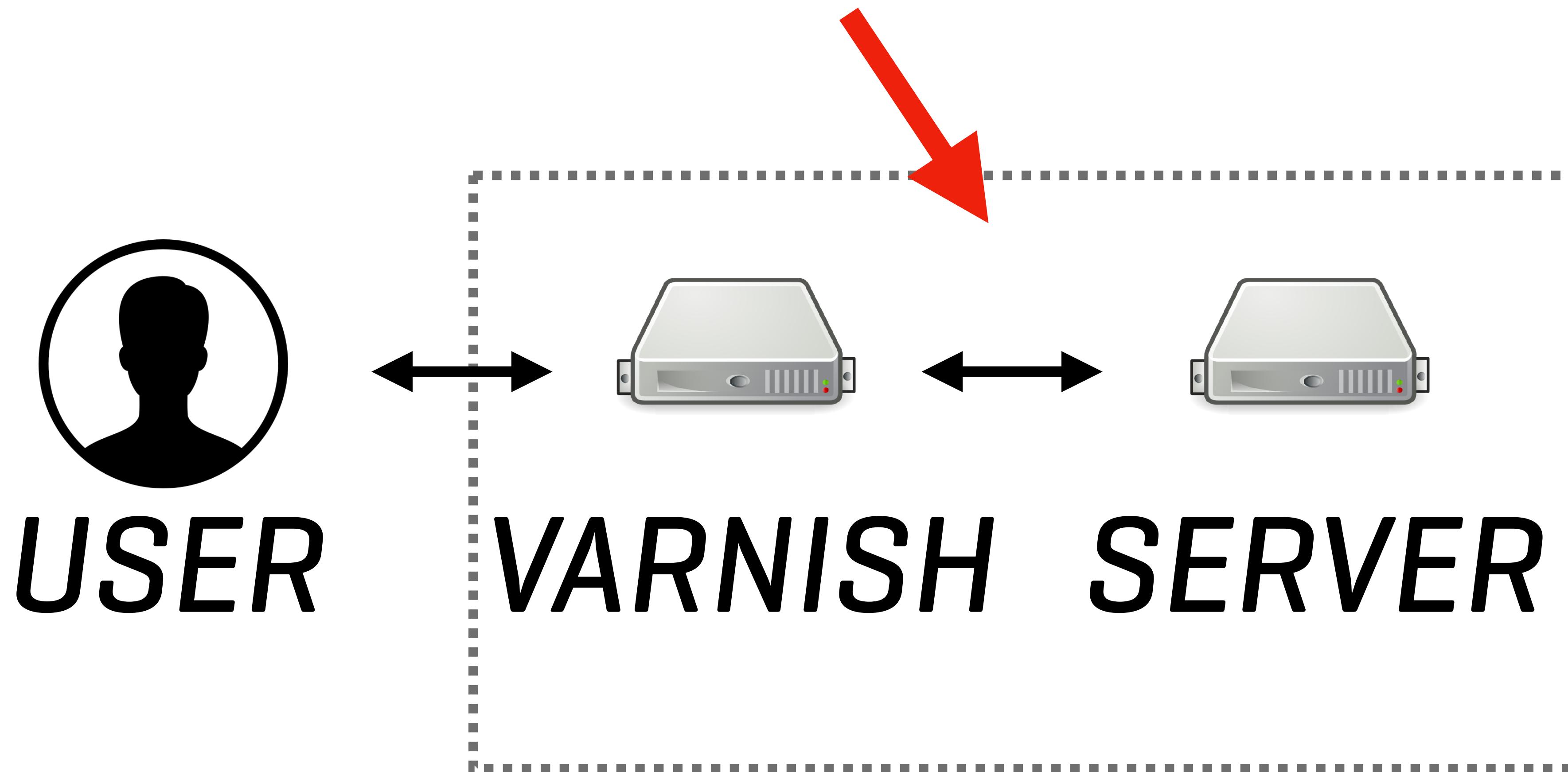




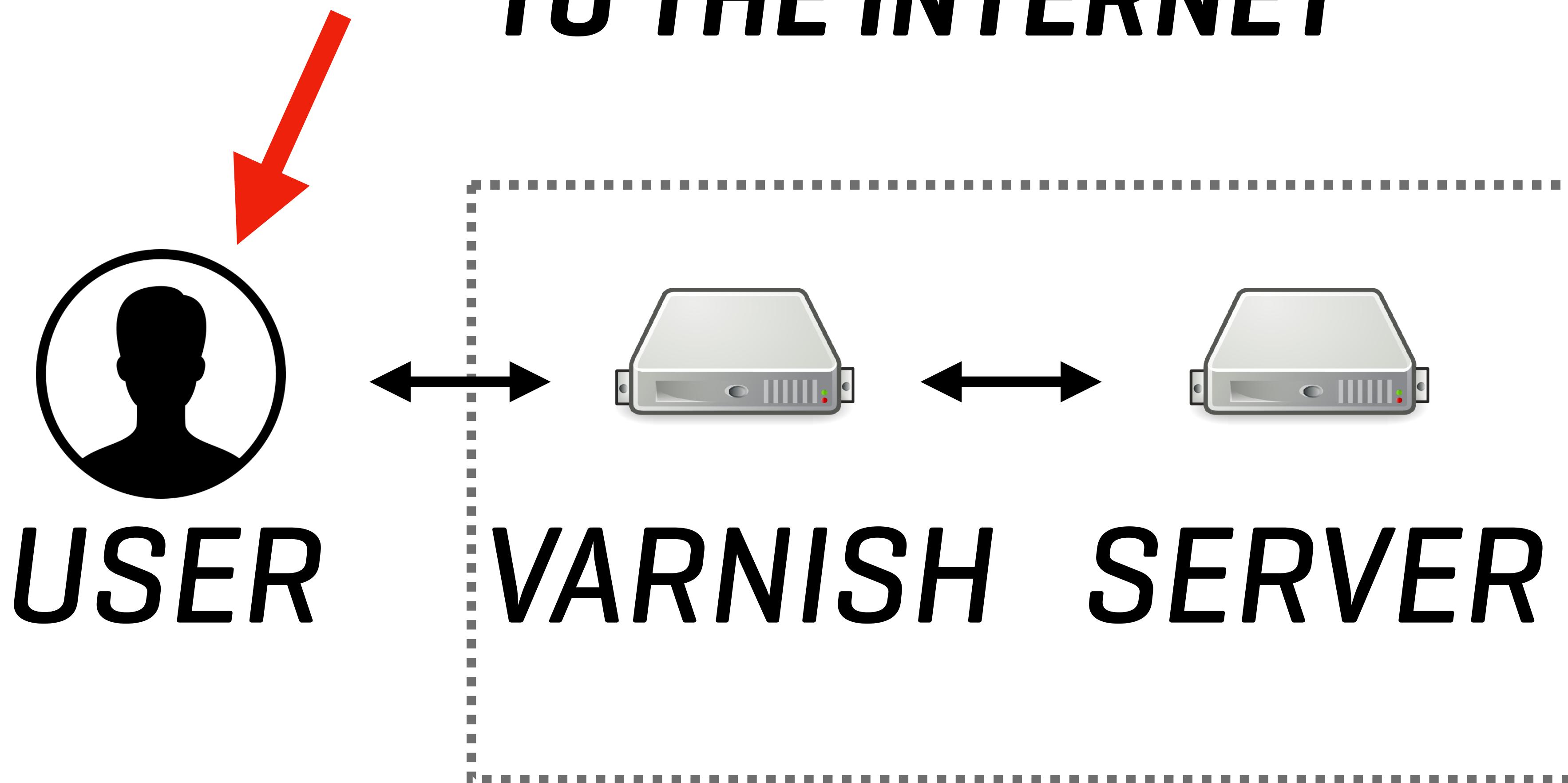
**USER**

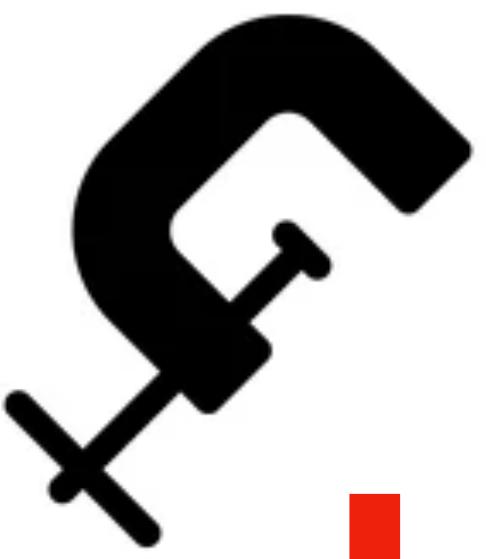


# **ORIGIN DATACENTER**



**SOMEWHERE CONNECTED  
TO THE INTERNET**





***REDUCE THE SIZE OVER THE WIRE***



**COMPRESSION**

# **COMPRESSION**

- ✓ GZIP COMPRESSION FOR TEXT-BASED CONTENT
- ✓ OPTIONAL BROTLI COMPRESSION *(VARNISH ENTERPRISE)*
- ✓ REDUCE THE SIZE OF THE IMAGES *(VARNISH ENTERPRISE)*

```
vcl 4.1;

backend default {
    .host = "backend.example.com";
    .port = "80";
}

sub vcl_backend_response {
    if(beresp.http.Content-Type ~ "^text|application/" || beresp.http.Content-Type == "image/svg+xml") {
        set beresp.do_gzip = true;
    }
}
```

```
vcl 4.1;

import brotli;

backend default {
    .host = "backend.example.com";
    .port = "80";
}

sub vcl_init {
    brotli.init(BOTH, transcode = true);
}

sub vcl_backend_response {
    if(beresp.http.Content-Type ~ "^text|application/" ||  

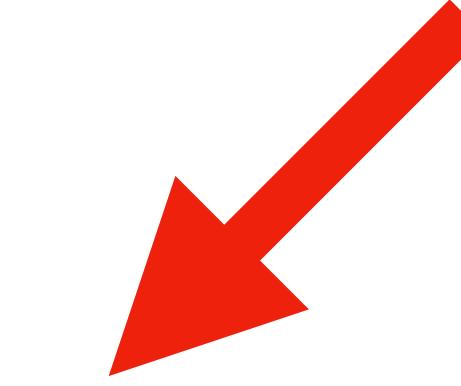
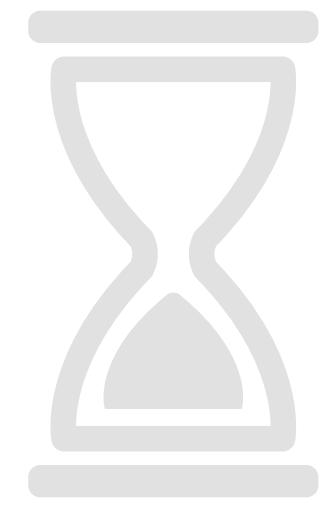
        beresp.http.Content-Type == "image/svg+xml") {
        brotli.compress();
    }
}
```

```
vcl 4.1;

import image;
import headerplus;

backend default {
    .host = "backend.example.com";
    .port = "80";
}

sub vcl_backend_response {
    headerplus.init(beresp);
    if (beresp.http.content-type ~ "^image/(jpeg|png)$") {
        if (bereq.http.Accept ~ "image/webp") {
            image.webp();
        }
        headerplus.attr_set("Vary", "Accept");
    }
    headerplus.write();
}
```



**REDUCE THE *DISTANCE* BETWEEN  
CLIENT & SERVER**

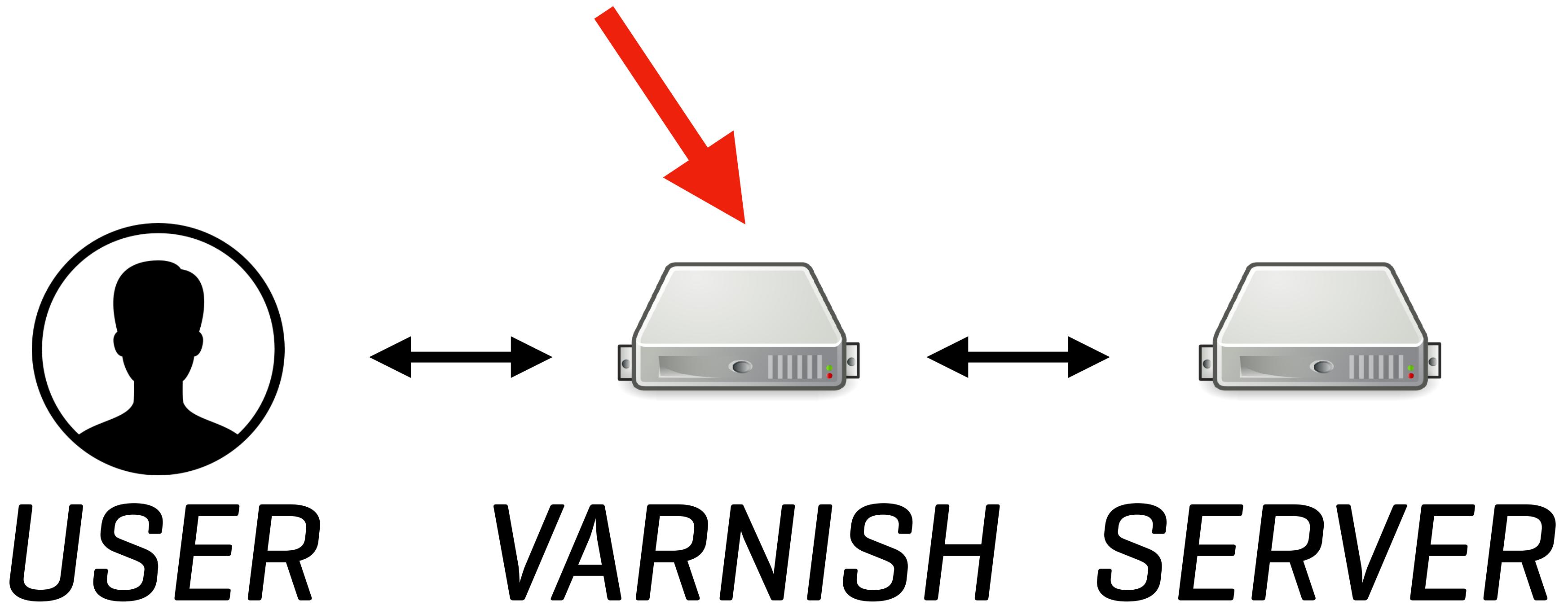
# **NETWORK LIMITATIONS**



# **NETWORK LIMITATIONS**

- ✓ NETWORK LATENCY BECAUSE OF DISTANCE
- ✓ REDUCED THROUGHPUT BECAUSE OF NETWORK SATURATION

# **ONE VARNISH SERVER IS NO LONGER ENOUGH**





# **WHY NOT USE A CDN PRODUCT?**



**VARNISH IS CDN SOFTWARE!**

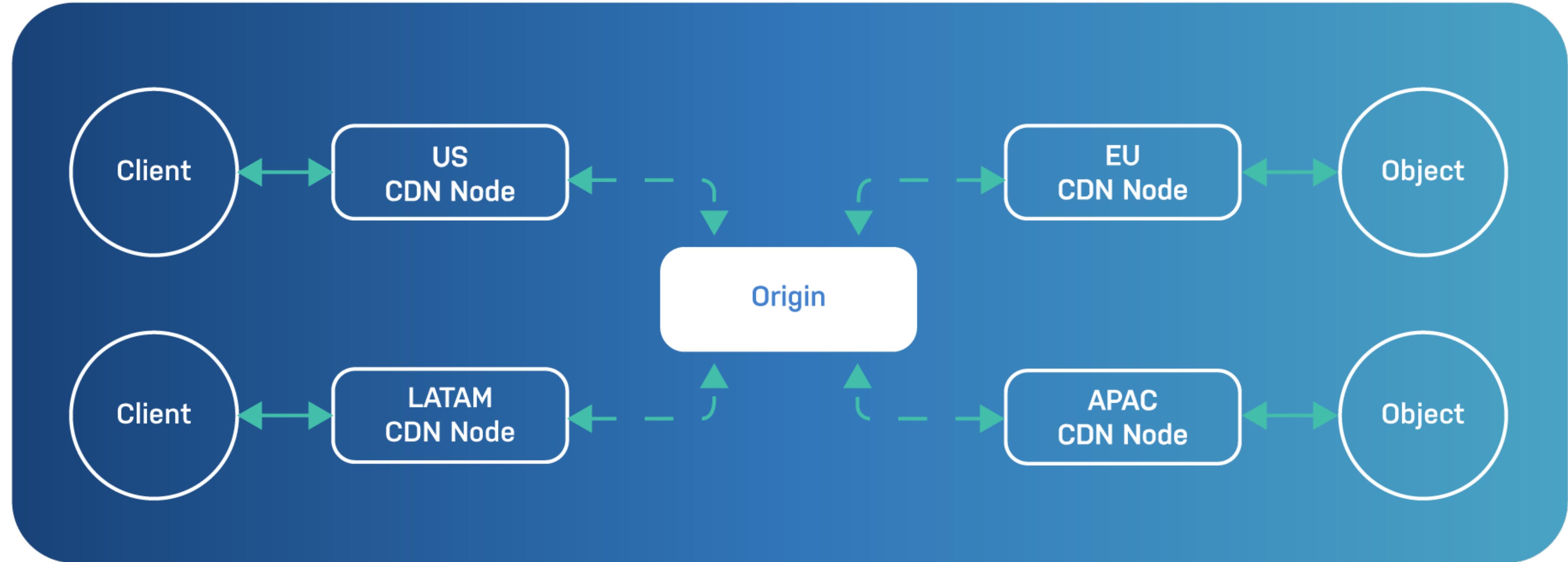
# **VARNISH AS A PRIVATE CDN**

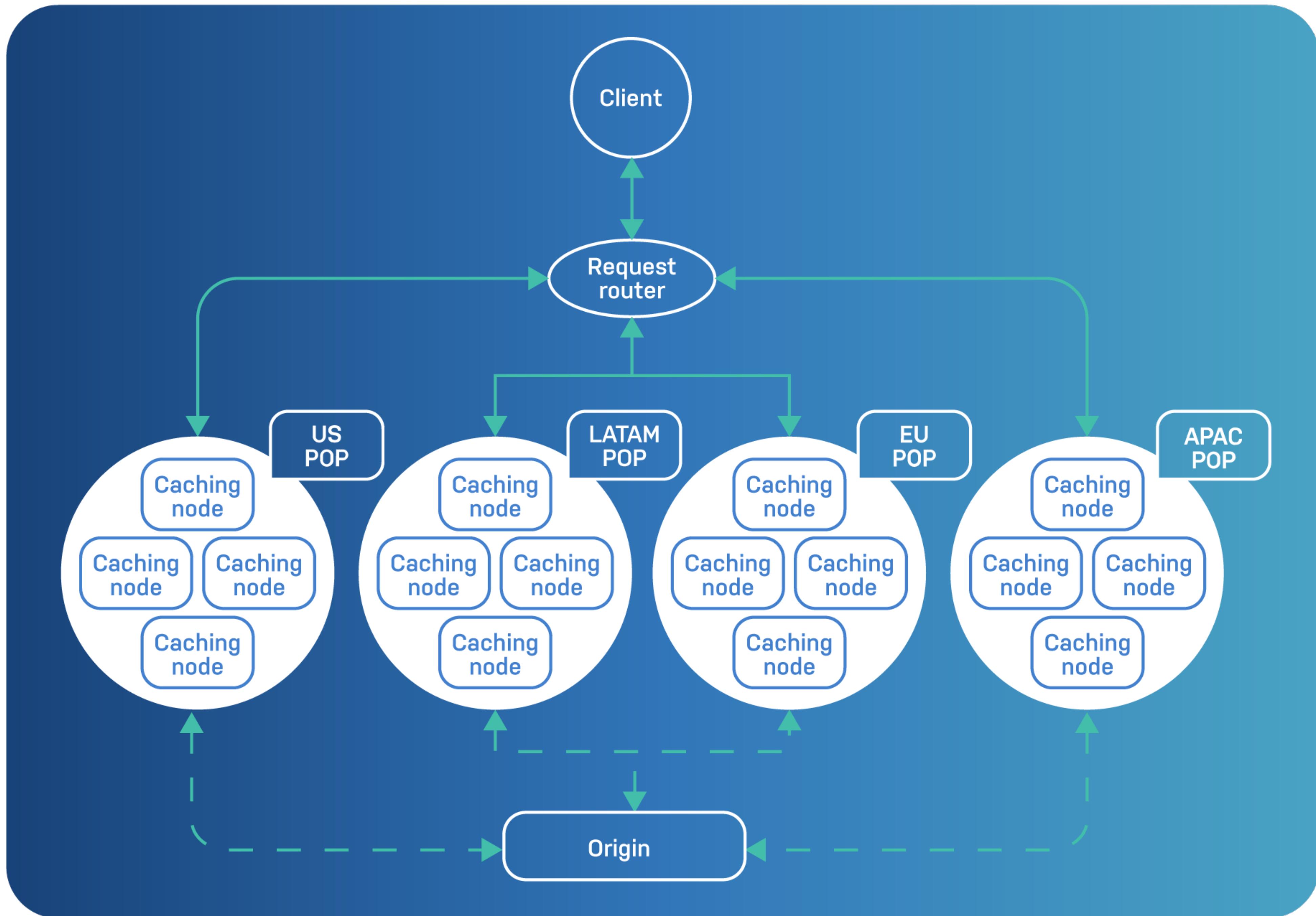
- ✓ SERVERS & RESOURCES AREN'T SHARED
- ✓ MORE CONTROL OVER CONTENT DELIVERY FLOW
- ✓ FOR SELECTED LOCALIZED POPS
- ✓ IF YOU ALREADY HAVE THE INFRASTRUCTURE
- ✓ REDUCE COST
- ✓ HYBRID: INCREASE COVERAGE LOCATIONS OF YOUR CDN
- ✓ HYBRID: PROTECT YOUR ORIGIN FROM CDN REVALIDATION

# **WHY VARNISH?**

# WHY VARNISH?

- ✓ PERFORMANCE & THROUGHPUT
- ✓ REQUEST COALESCING
- ✓ BACKEND REQUEST ROUTING
- ✓ HORIZONTAL SCALABILITY
- ✓ CONTROL
- ✓ TRANSPARENCY





# **REQUEST ROUTER**

- ✓ ENTRYPPOINT
- ✓ LOCATES "BEST" POINT OF PRESENCE
- ✓ NO SINGLE POINT OF FAILURE
- ✓ DISTRIBUTED
- ✓ OFTEN DNS-BASED WITH GEOLOCATION CAPABILITIES



Route53

**POWERDNS** ::::

## Quick create record [Info](#)

[Switch to wizard](#)

### ▼ Record 1

**Record name** [Info](#)

www

.diy-cdn.varnish-software.com

Valid characters: a-z, 0-9, ! " # \$ % & ' ( ) \* + , - / ; < = > ? @ [ \ ] ^ \_ ` { | } . ~

**Record type** [Info](#)

A – Routes traffic to an IPv4 address and so...

**Value** [Info](#)

192.0.2.235

[Delete](#) Alias

Enter multiple values on separate lines.

**TTL (seconds)** [Info](#)

300

[1m](#) [1h](#) [1d](#)

Recommended values: 60 to 172800 (two days)

**Routing policy** [Info](#)

Geolocation

**Location**[Choose location](#)

Default

Continents

Africa

Antarctica

Asia

Europe

North America

Oceania

South America

Countries

Andorra

United Arab Emirates

Afghanistan

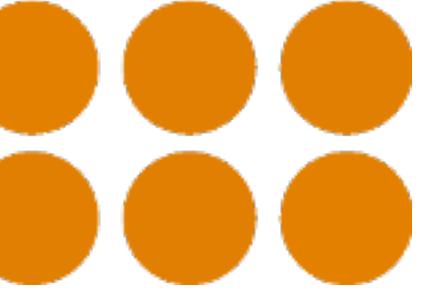
Antigua and Barbuda

Anguilla

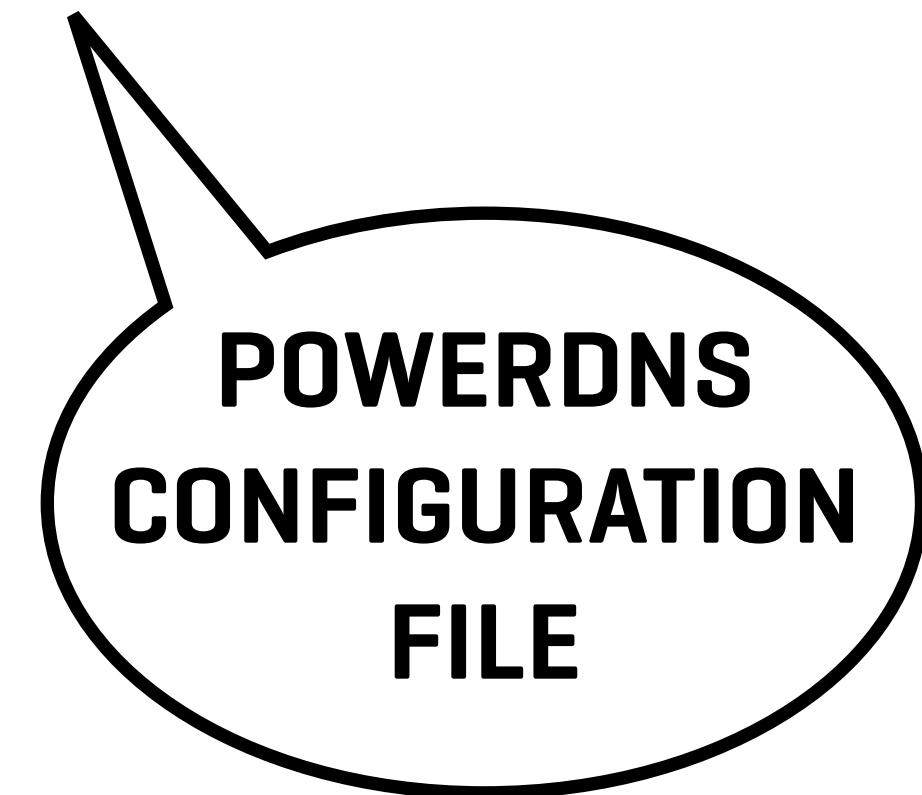
Albania

### ► View existing records

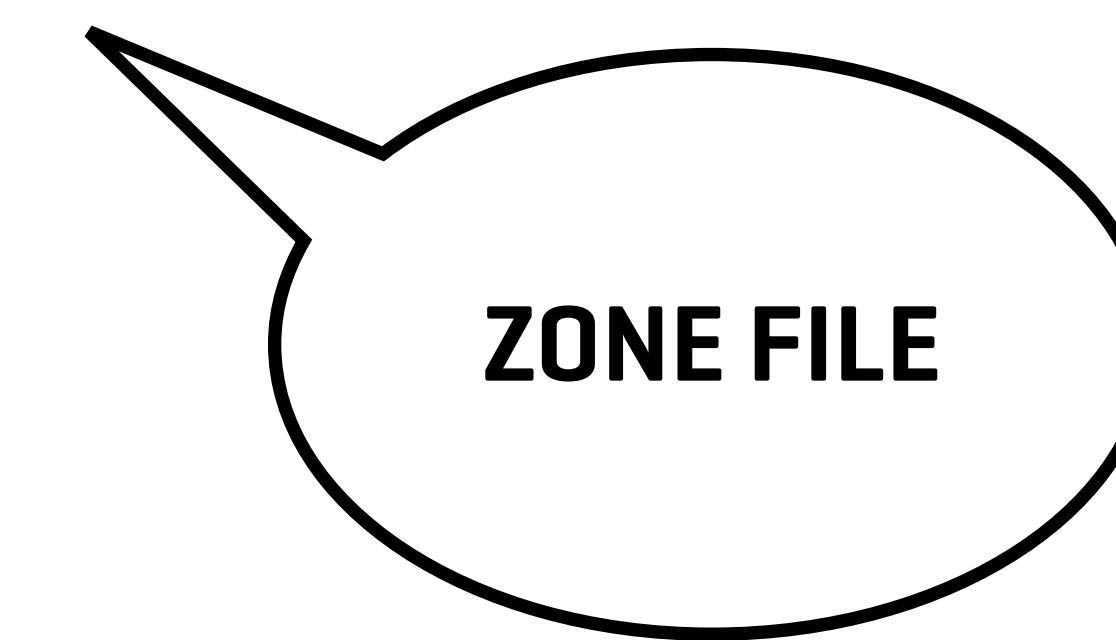
The following table lists the existing records in diy-cdn.varnish-software.com.

**POWERDNS** 

```
launch=geoip
geoip-database-files=/usr/share/GeoIP/GeoIP.dat,/usr/share/GeoIP/
GeoIPv6.dat
geoip-zones-file=/etc/powerdns/zone
```

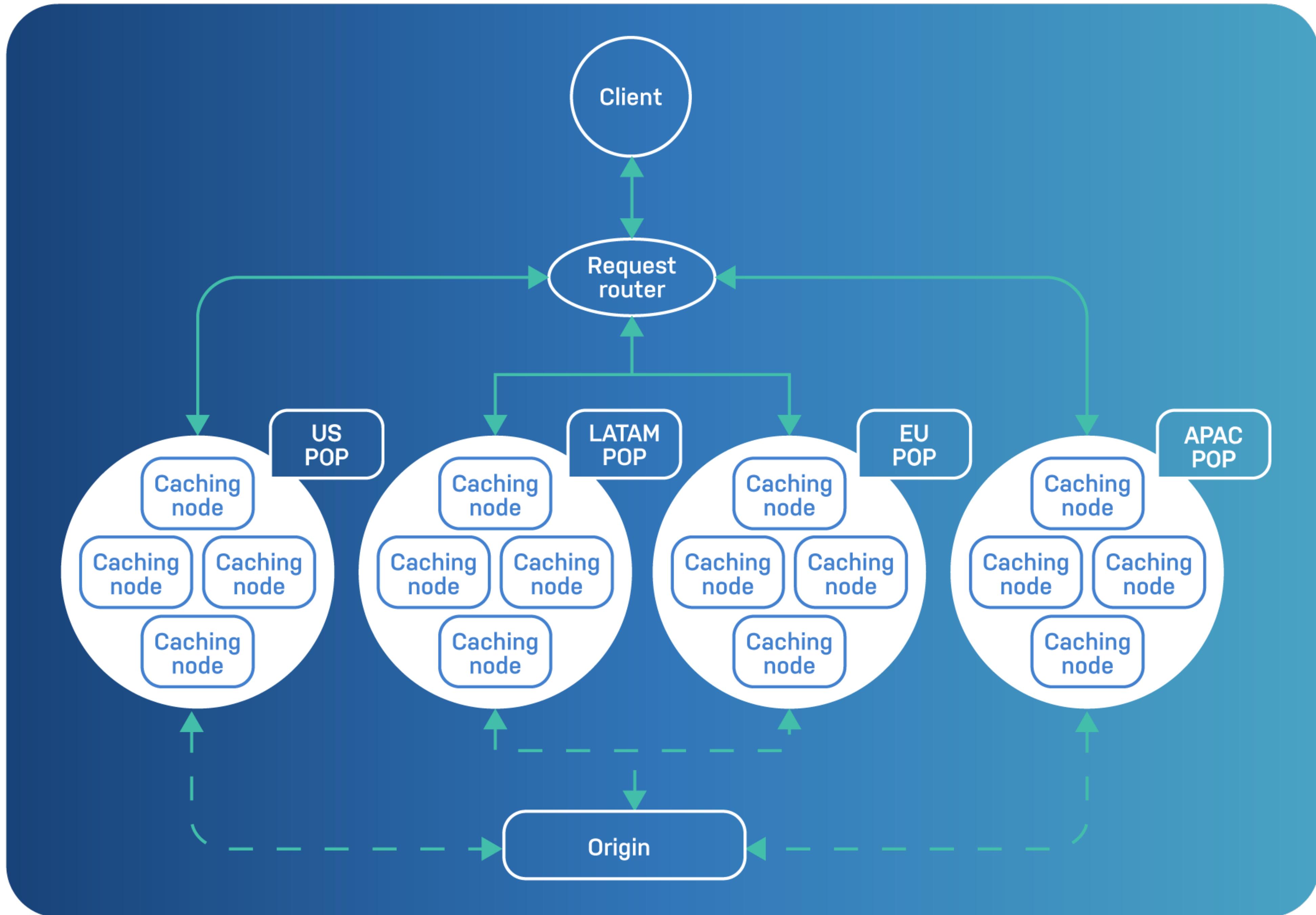


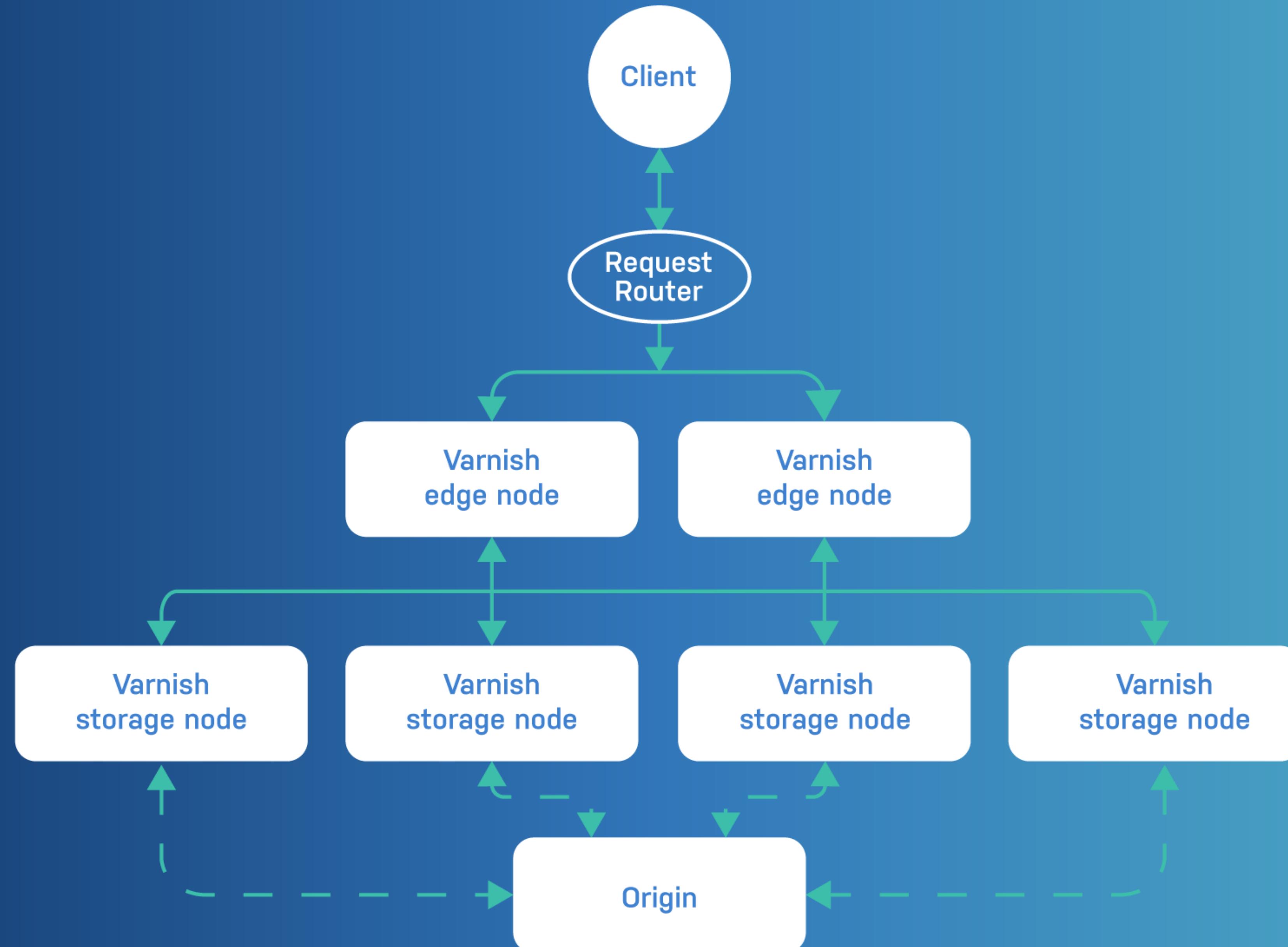
```
domains:
- domain: example.com
- ttl: 300
- records:
  example.com:
    - soa: ns1.example.com hostmaster.example.com 2014090125 7200 3600
  1209600 3600
    - ns: ns1.example.com
    - a: 192.168.0.1
  eu.www.example.com:
    - a: 192.0.0.2
  na.www.example.com:
    - a: 192.0.0.3
  as.www.example.com:
    - a: 192.0.0.4
  af.www.example.com:
    - a: 192.0.0.5
  sa.www.example.com:
    - a: 192.0.0.6
- services:
  www.example.com: [%cn.www.example.com', 'eu.www.example.com'%]
```



```
domains:
- domain: example.com
- ttl: 300
- records:
  example.com:
    - soa: ns1.example.com hostmaster.example.com 2014090125 7200 3600
  1209600 3600
    - ns: ns1.example.com
    - a: 192.168.0.1
  eu.www.example.com:
    - a: 192.0.0.2
  na.www.example.com:
    - a: 192.0.0.3
  as.www.example.com:
    - a: 192.0.0.4
  af.www.example.com:
    - a: 192.0.0.5
  sa.www.example.com:
    - a: 192.0.0.6
- services:
  www.example.com: [%cn.www.example.com', 'eu.www.example.com'%]
```

<b>%%</b> :	literal %
<b>%co:</b>	<b>With legacy GeolP database: three letter country name, MMDB &amp; others: ISO3166 country code</b>
<b>%cc:</b>	<b>ISO3166 country code.</b>
<b>%cn:</b>	<b>ISO3166 continent code.</b>
<b>%af:</b>	<b>v4 or v6.</b>
<b>%re:</b>	<b>Region code</b>
<b>%na:</b>	<b>AS organization name (spaces are converted to _)</b>
<b>%as:</b>	<b>AS number</b>
<b>%ci:</b>	<b>City name</b>
<b>%loc:</b>	<b>LOC record style expansion of location</b>
<b>%lat:</b>	<b>Decimal degree latitude</b>
<b>%lon:</b>	<b>Decimal degree longitude</b>





# **EDGE TIER**

- ✓ DIRECT INTERFACE WITH CLIENTS
- ✓ HANDLE TLS
- ✓ AUTHENTICATION, RATE LIMITING, THROTTLING
- ✓ STORES "HOT" DATA IN MEMORY
- ✓ CONNECTS TO THE STORAGE TIER VIA SHARDING

# **SHARDING**

- ✓ MAKE REQUESTS GO TO THE SAME BACKEND NODE
- ✓ USES CONSISTENT HASHING
- ✓ URL AS THE HASHING KEY
- ✓ SCALE STORAGE HORIZONTALLY

```
vcl 4.1;

import directors;

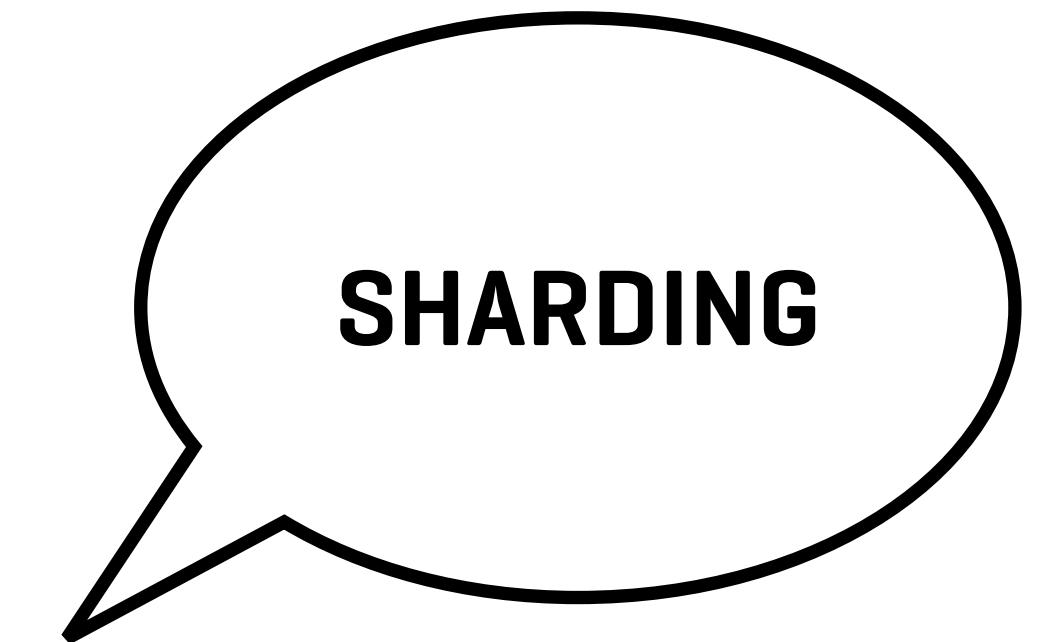
backend storage1 {
    .host = "storage1.example.com";
    .port = "80";
}

backend storage2 {
    .host = "storage2.example.com";
    .port = "80";
}

backend storage3 {
    .host = "storage3.example.com";
    .port = "80";
}

sub vcl_init {
    new storage_tier = directors.shard();
    storage_tier.add_backend(storage1, rampup=5m);
    storage_tier.add_backend(storage2, rampup=5m);
    storage_tier.add_backend(storage3, rampup=5m);
    storage_tier.reconfigure();
}

sub vcl_recv {
    set req.backend_hint = storage_tier.backend(URL);
}
```



```
vcl 4.1;

import directors;

backend storage1 {
    .host = "storage1.example.com";
    .port = "80";
}

backend storage2 {
    .host = "storage2.example.com";
    .port = "80";
}

backend storage3 {
    .host = "storage3.example.com";
    .port = "80";
}

sub vcl_init {
    new storage_tier = directors.shard();
    storage_tier.add_backend(storage1, rampup=5m);
    storage_tier.add_backend(storage2, rampup=5m);
    storage_tier.add_backend(storage3, rampup=5m);
    storage_tier.reconfigure();
}

sub vcl_recv {
    set req.backend_hint = storage_tier.backend(URL);
}
```

# **STORAGE TIER**

- ✓ STORE AS MUCH CONTENT AS POSSIBLE
- ✓ SERVE AS SECONDARY ORIGIN
- ✓ LONGER TTLS
- ✓ STORE OBJECT IN MEMORY WITH VARNISH CACHE
- ✓ DON'T USE THE "FILE" STORAGE ENGINE
- ✓ MASSIVE STORAGE ENGINE WITH VARNISH ENTERPRISE

# **MASSIVE STORAGE ENGINE**

- ✓ VARNISH ENTERPRISE STORAGE ENGINE
- ✓ MIX BETWEEN MEMORY & DISK STORAGE
- ✓ CACHE MASSIVE AMOUNTS OF DATA ON DISK
- ✓ CUSTOM PAGE CACHE MECHANISM
- ✓ DISK PERSISTENCE
- ✓ EXTREMELY FAST & EFFICIENT

# **MASSIVE STORAGE ENGINE**

- ✓ AUTOMATIC MEMORY MANAGEMENT
- ✓ MULTIPLE STORAGE LOCATIONS (BOOKS & STORES)
- ✓ VCL INTERFACE THROUGH "VMOD\_MSE"
- ✓ HIGHLY CONFIGURABLE

```
env: {
    id = "mse";
    memcache_size = "auto";

    books = ( {
        id = "book";
        directory = "/var/lib/mse/book";

        stores = ( {
            id = "store1";
            filename = "/var/lib/mse/store1.dat";
            size = "500G";
            tags = ( "small", "ssd" );
        }, {
            id = "store2";
            filename = "/var/lib/mse/store2.dat";
            size = "10T";
            tags = ( "big", "sata" );
        } );
    } );
};

};
```

```
env: {
    id = "mse";
    memcache_size = "auto";

    books = ( {
        id = "book";
        directory = "/var/lib/mse/book";

        stores = ( {
            id = "store1";
            filename = "/var/lib/mse/store1.dat";
            size = "500G";
            tags = ( "small", "ssd" );
        }, {
            id = "store2";
            filename = "/var/lib/mse/store2.dat";
            size = "10T";
            tags = ( "big", "sata" );
        } );
    } );
};

};
```

```
env: {
    id = "mse";
    memcache_size = "auto";

    books = ( {
        id = "book";
        directory = "/var/lib/mse/book";

        stores = ( {
            id = "store1";
            filename = "/var/lib/mse/store1.dat";
            size = "500G";
            tags = ( "small", "ssd" );
        }, {
            id = "store2";
            filename = "/var/lib/mse/store2.dat";
            size = "10T";
            tags = ( "big", "sata" );
        } );
    } );
};

};
```

```
env: {
    id = "mse";
    memcache_size = "auto";

    books = ( {
        id = "book";
        directory = "/var/lib/mse/book";

        stores = ( {
            id = "store1";
            filename = "/var/lib/mse/store1.dat";
            size = "500G";
            tags = ( "small", "ssd" );
        }, {
            id = "store2";
            filename = "/var/lib/mse/store2.dat";
            size = "10T";
            tags = ( "big", "sata" );
        } );
    } );
};

};
```

```
env: {
    id = "mse";
    memcache_size = "auto";

    books = ( {
        id = "book";
        directory = "/var/lib/mse/book";

        stores = ( {
            id = "store1";
            filename = "/var/lib/mse/store1.dat";
            size = "500G";
            tags = ( "small", "ssd" );
        }, {
            id = "store2";
            filename = "/var/lib/mse/store2.dat";
            size = "10T";
            tags = ( "big", "sata" );
        } );
    } );
};

};
```

```
vcl 4.1;

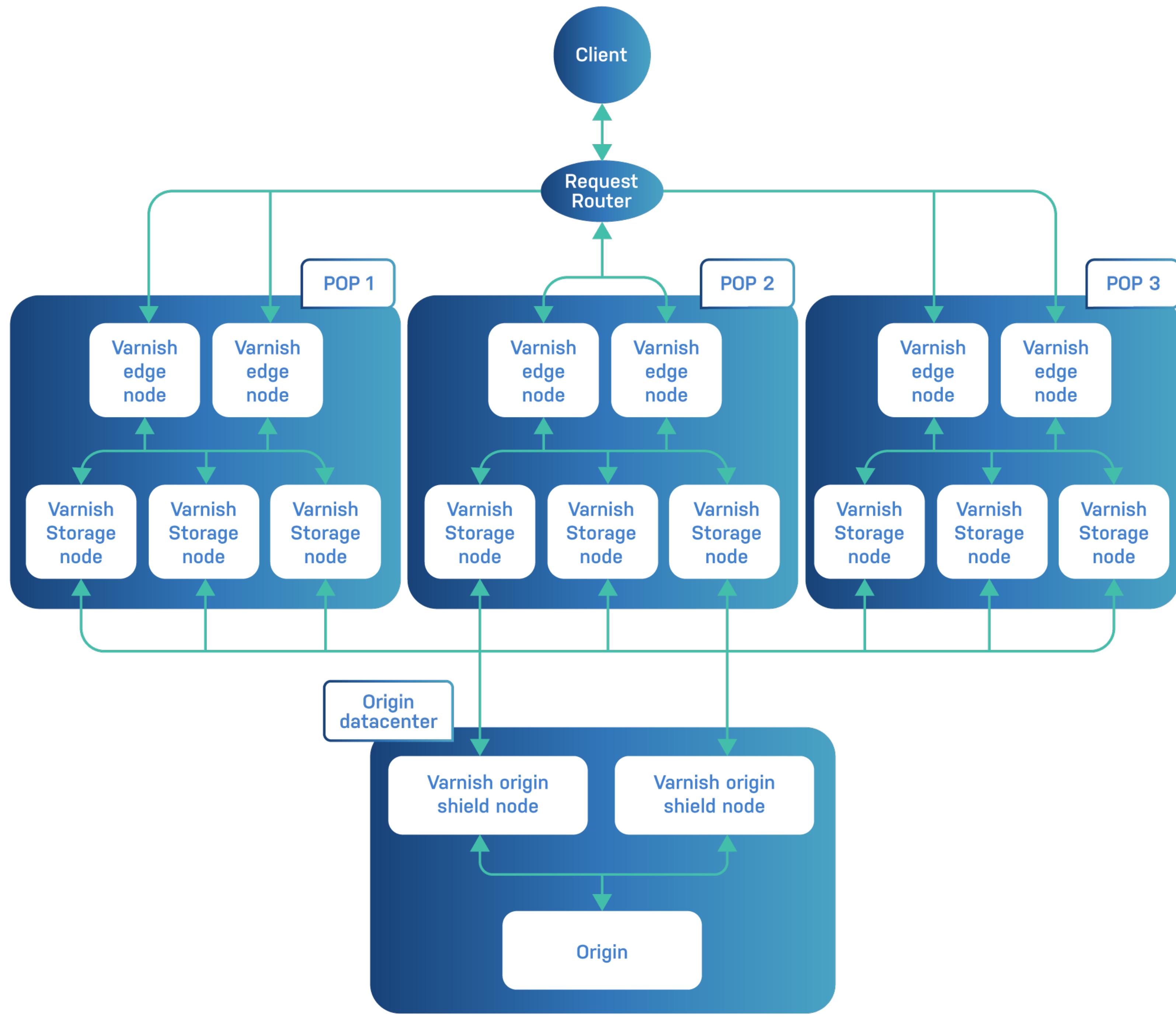
import mse;
import std;

sub vcl_backend_response {
    if (beresp.ttl < 120s) {
        mse.set_stores("none");
    } else {
        if (beresp.http.Transfer-Encoding ~ "chunked" ||
            std.integer(beresp.http.Content-Length,0) > std.bytes("100M")) {
            mse.set_stores("sata");
        } else {
            mse.set_stores("ssd");
        }
    }
}
```

```
vcl 4.1;

import mse;
import std;

sub vcl_backend_response {
    if (beresp.ttl < 120s) {
        mse.set_stores("none");
    } else {
        if (beresp.http.Transfer-Encoding ~ "chunked" ||
            std.integer(beresp.http.Content-Length,0) > std.bytes("100M")) {
            mse.set_stores("sata");
        } else {
            mse.set_stores("ssd");
        }
    }
}
```



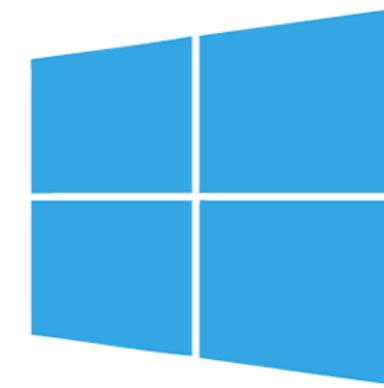
# **ORIGIN SHIELD TIER**

- ✓ TRADITIONAL VARNISH USE CASE
- ✓ PROTECT THE ORIGIN
- ✓ COULD BE INTEGRATED IN THE STORAGE TIER
- ✓ CONTAINS THE CACHING RULES
- ✓ HAS MOST OF THE SECURITY-RELATED FEATURES
- ✓ STALE-IF-ERROR LOGIC

# ORIGIN SHIELD IN A HYBRID CDN SETUP



**"IT'S NOT ABOUT THE CACHE  
HITS, IT'S ABOUT HOW GOOD  
YOUR MISSES ARE"**



Microsoft  
Azure



Google Cloud

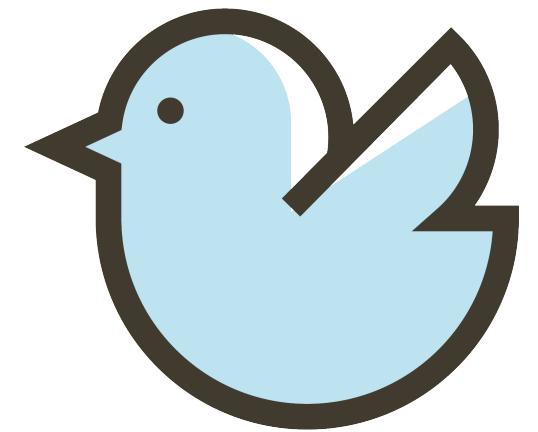
OFFICIAL MACHINE IMAGES ON CLOUD MARKETPLACES

THE END





***HTTPS://FERYN.EU***



***HTTPS://TWITTER.COM/THIJSFERYN***



***HTTPS://INSTAGRAM.COM/THIJSFERYN***