

Leverage HTTP to deliver cacheable Drupal websites



DRUPALCAMP GHENT

23 & 24 NOVEMBER 2018

By Thijs Feryn

Slow websites suck



**Web performance is
an essential part of
the user experience**



Slow ~ Down







Google



Mo'

Mo'

Mo'

money

servers

problems



Identify slowest parts



The background features a complex, abstract pattern of wavy, overlapping lines in shades of blue and green. A prominent dark blue horizontal band runs across the middle of the image, serving as a backdrop for the text. The overall effect is a sense of depth and movement, reminiscent of a digital or data visualization.

Optimize

After a while you
hit the limits



Caché



Hi,



I'm Thijs

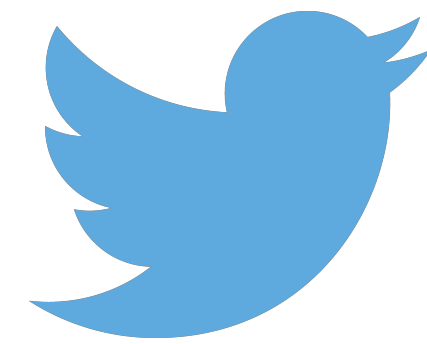


I'm an
Evangelist

at  **Combell**
Your host on the internet

I'm

@thijstferyn



O'REILLY®

Compliments of
VARNISH
SOFTWARE



Getting Started with Varnish Cache

ACCELERATE YOUR WEB APPLICATIONS

Thijs Feryn



Don't
recompute
if the data
hasn't
changed

Handwritten character: 丸 (Maru)

Handwritten text: 丸印 (Maru In)

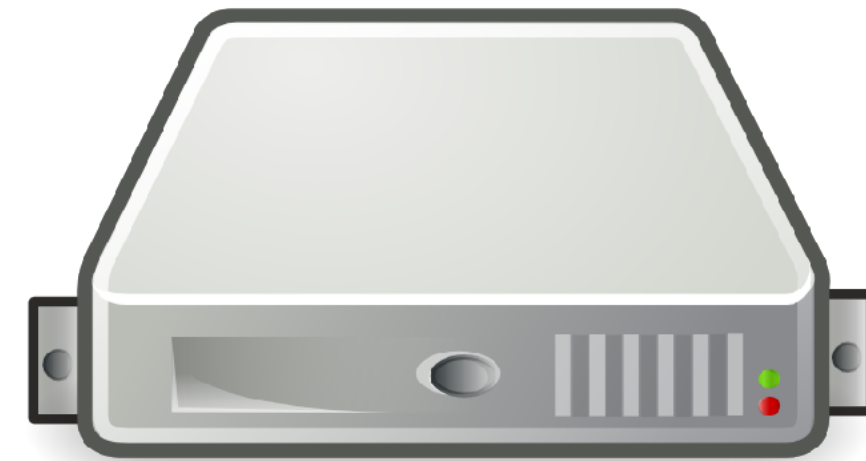
Handwritten symbols: A colon followed by two vertical lines and a series of vertical strokes.

Cache-control: public, max-age=300



Reverse
coaching
proxy

Normally



User

Server

With ReCaPro *

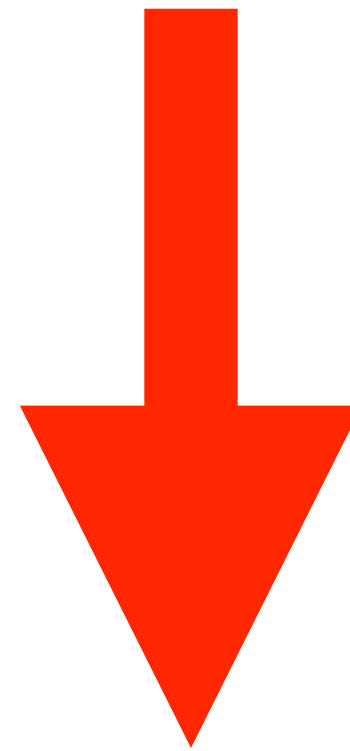


User ReCaPro Server

* Reverse Caching Proxy




```
<?php
$settings[ 'reverse_proxy' ] = TRUE;
$settings[ 'reverse_proxy_addresses' ] = array( '127.0.0.1' );
```



Drupal\Core\StackMiddleware\ReverseProxyMiddleware

Drupal\Core\StackMiddleware\ReverseProxyMiddleware

```
// Initialize proxy settings.
if ($settings->get('reverse_proxy', FALSE)) {
    $ip_header = $settings->get('reverse_proxy_header', 'X_FORWARDED_FOR');
    $request::setTrustedHeaderName($request::HEADER_X_FORWARDED_FOR, $ip_header);

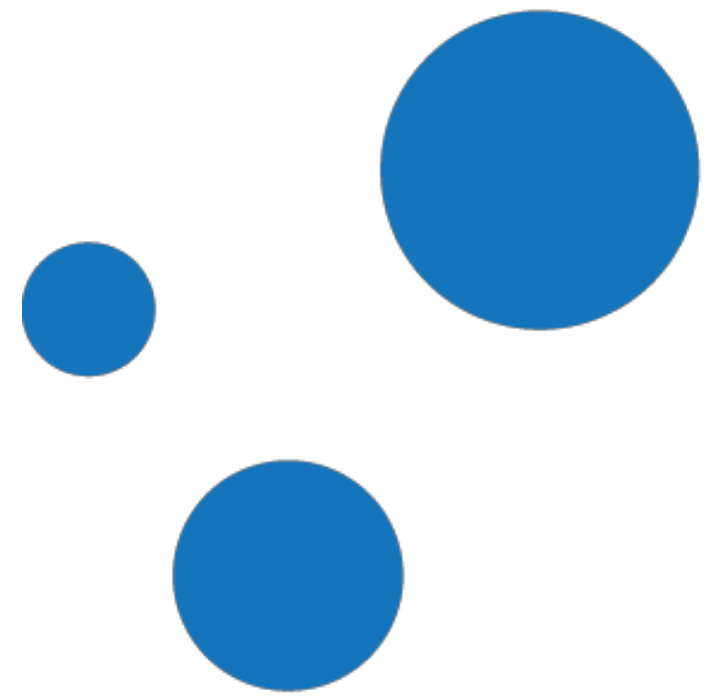
    $proto_header = $settings->get('reverse_proxy_proto_header', 'X_FORWARDED_PROTO');
    $request::setTrustedHeaderName($request::HEADER_X_FORWARDED_PROTO, $proto_header);

    $host_header = $settings->get('reverse_proxy_host_header', 'X_FORWARDED_HOST');
    $request::setTrustedHeaderName($request::HEADER_X_FORWARDED_HOST, $host_header);

    $port_header = $settings->get('reverse_proxy_port_header', 'X_FORWARDED_PORT');
    $request::setTrustedHeaderName($request::HEADER_X_FORWARDED_PORT, $port_header);

    $forwarded_header = $settings->get('reverse_proxy_forwarded_header', 'FORWARDED');
    $request::setTrustedHeaderName($request::HEADER_FORWARDED, $forwarded_header);

    $proxies = $settings->get('reverse_proxy_addresses', []);
    if (count($proxies) > 0) {
        $request::setTrustedProxies($proxies, Request::HEADER_X_FORWARDED_ALL |
Request::HEADER_FORWARDED);
    }
}
```



VARNISH CACHE

traffic  server™

NGINX

Content Delivery Network





Page Cache

services:

http_middleware.page_cache:

class: Drupal\page_cache\StackMiddleware\PageCache

arguments: ['@cache.page', '@page_cache_request_policy', '@page_cache_response_policy']

tags:

- { **name:** http_middleware, **priority:** 200, **responder:** true }

cache.page:

class: Drupal\Core\Cache\CacheBackendInterface

tags:

- { **name:** cache.bin }

factory: cache_factory:get

arguments: [page]



```
protected function lookup(Request $request, $type = self::MASTER_REQUEST, $catch = TRUE)
{
    if ($response = $this->get($request)) {
        $response->headers->set('X-Drupal-Cache', 'HIT');
    }
    else {
        $response = $this->fetch($request, $type, $catch);
    }

    if ($request->cookies->has(session_name()) &&
        in_array('Cookie', $response->getVary()) &&
        !$response->headers->hasCacheControlDirective('no-cache')) {
        $response->setPrivate();
    }
}

...
```




In an ideal world

In an ideal world

- ✓ **Stateless**
- ✓ **Well-defined TTL**
- ✓ **Cache / no-cache per resource**
- ✓ **Cache variations**
- ✓ **Conditional requests**
- ✓ **Placeholders for non-cacheable content**



Reality
sucks

**This website wants
to use cookies.**

OK

No way



Google Analytics



Time To Live

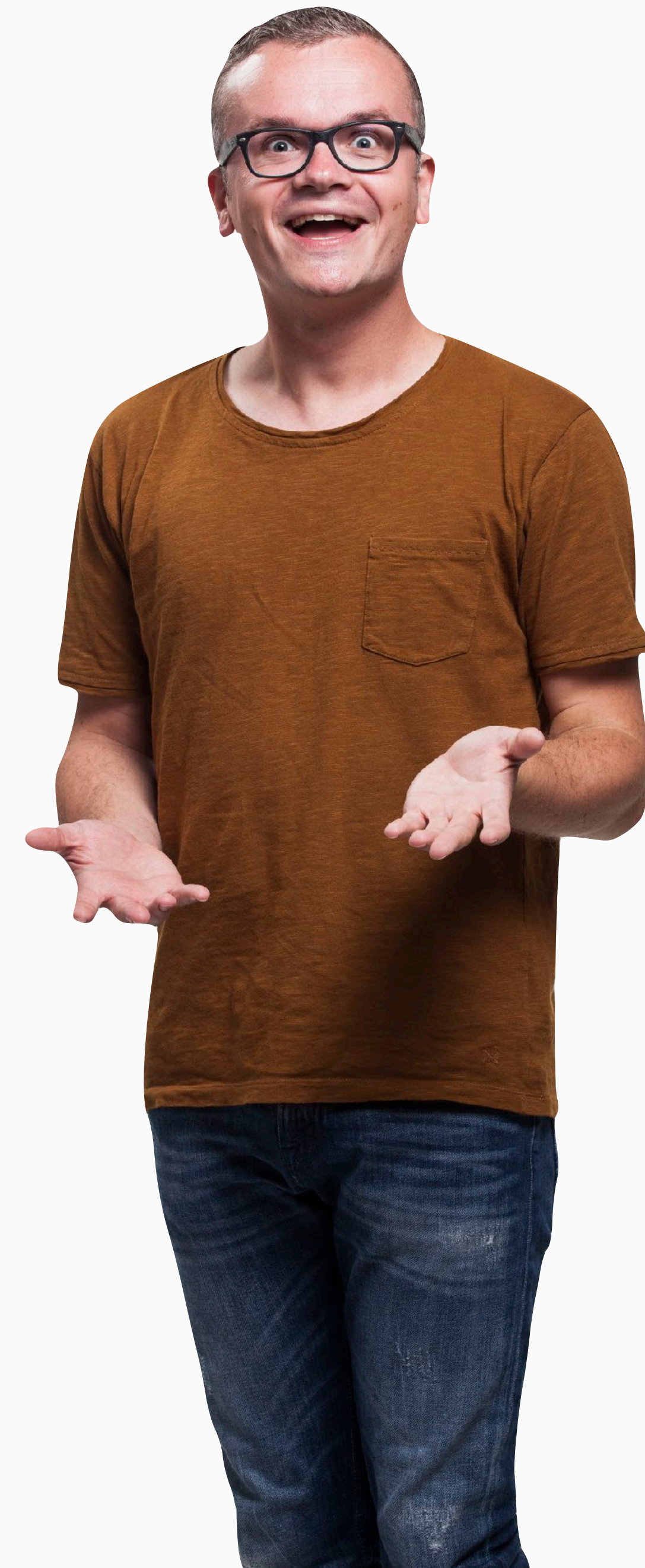


Cache variations



Legacy

What if we
could design
our software
with HTTP
caching in
mind?



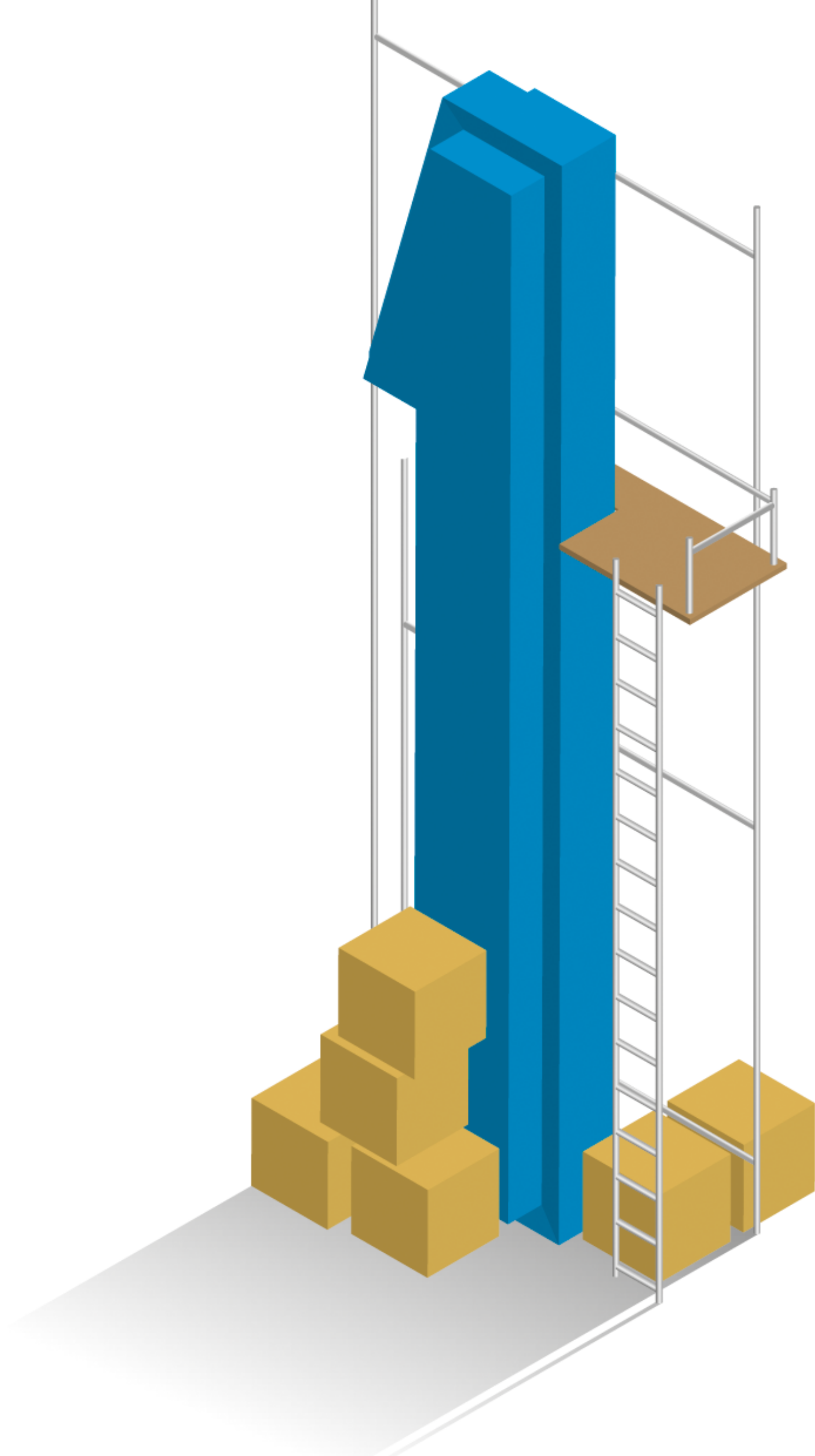
Caching state of mind

✓ **Portability**

→ **Consistent caching behavior**

✓ **Developer empowerment**

→ **Control**



Cache-Control

HTTP caching mechanisms

Expires: Sat, 09 Sep 2017 14:30:00 GMT

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

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

Performance

Purge

[Home](#) » [Administration](#) » [Configuration](#) » [Development](#)

▼ CLEAR CACHE

Clear all caches

▼ CACHING

Browser and proxy cache maximum age

5 min

This is used as the value for max-age in Cache-Control headers.

▼ BANDWIDTH OPTIMIZATION

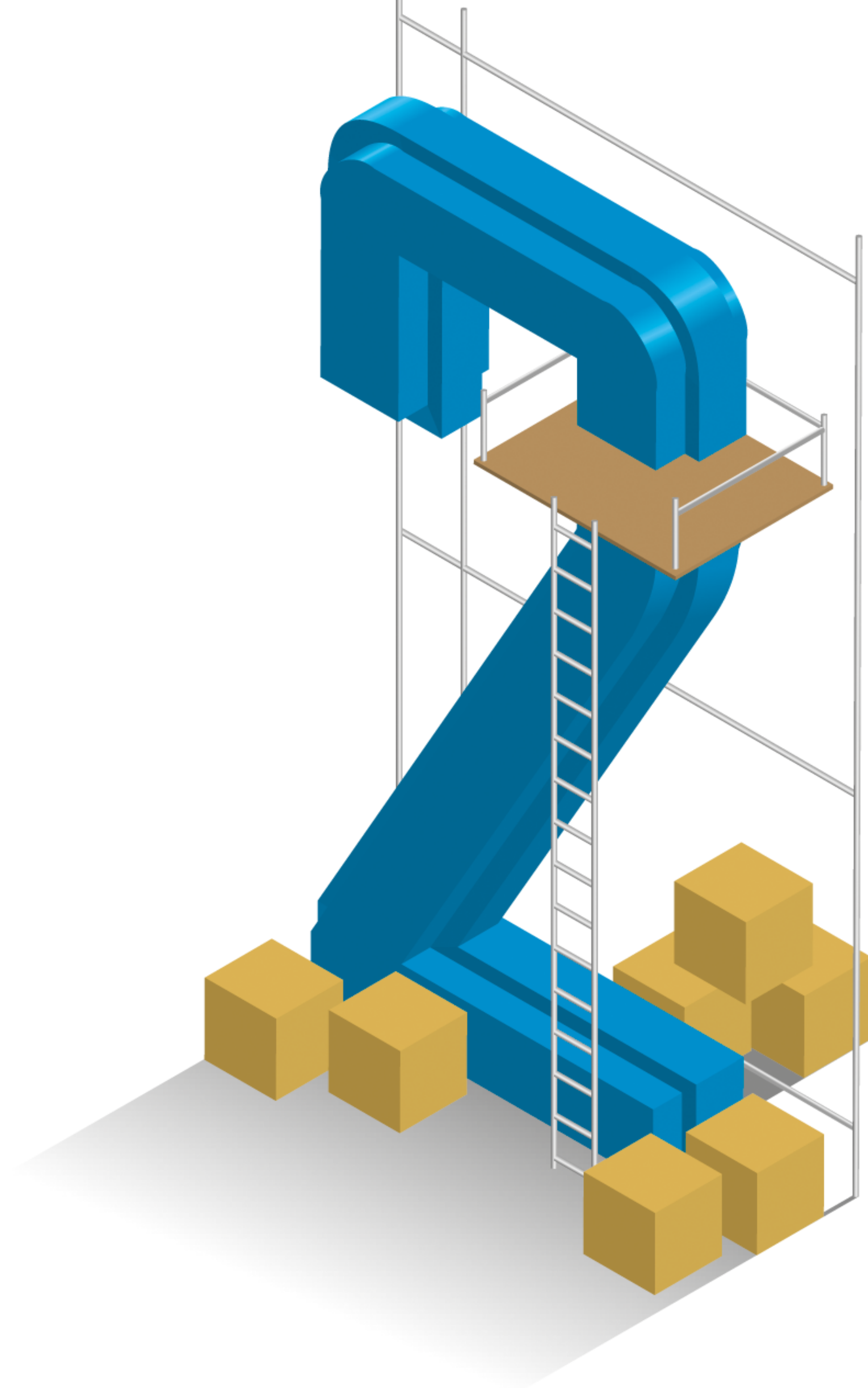
External resources can be optimized automatically, which can reduce both the size and number of requests made to your website.

- Aggregate CSS files
- Aggregate JavaScript files

Save configuration

HTTP/1.1 200 OK
Server: nginx/1.10.3
Content-Type: text/html; charset=UTF-8
Cache-Control: max-age=300, public
Date: Thu, 22 Nov 2018 13:08:39 GMT
X-Drupal-Dynamic-Cache: HIT
X-UA-Compatible: IE=edge
X-UA-Compatible: IE=edge
Content-language: en
X-Content-Type-Options: nosniff
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Frame-Options: SAMEORIGIN
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Last-Modified: Thu, 22 Nov 2018 13:08:57 GMT
ETag: W/"1542892137"
X-Generator: Drupal 8 (<https://www.drupal.org>)
X-Drupal-Cache: HIT
Vary: Accept-Encoding
X-Varnish: 196651 276984
Age: 149
Via: 1.1 varnish (Varnish/6.0)
X-Varnish-Cache: HIT
Accept-Ranges: bytes

HTTP/1.1 200 OK
Server: nginx/1.10.3
Content-Type: text/html; charset=UTF-8
Cache-Control: must-revalidate, no-cache, private
Date: Thu, 22 Nov 2018 13:13:50 GMT
X-Drupal-Dynamic-Cache: UNCACHEABLE
X-UA-Compatible: IE=edge
Content-language: en
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Vary:
X-Generator: Drupal 8 (<https://www.drupal.org>)
Surrogate-Control: no-store, content="BigPipe/1.0"
Set-Cookie: SESS2921d032e0cb254cf4f507503f1f7066=Jr6KuMl-n2aGySwbXUK-
DCV7IGZi34JL_nNDP-MLQYY; expires=Sat, 15-Dec-2018 16:47:10 GMT; Max-
Age=2000000; path=/; domain=.185.115.217.244.xip.io; HttpOnly
X-Varnish: 277020
Age: 0
Via: 1.1 varnish (Varnish/6.0)
X-Varnish-Cache: MISS



Conditional
requests

Only fetch
payload that has
changed

HTTP/1.1 200 OK

Otherwise:

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/1.0 304 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/1.0 304 Not Modified

Host: localhost

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

Cache-Control: public, max-age=100,
s-maxage=500, stale-while-revalidate=20

Validate



quickly

Exit



early



redis

**Store
& retrieve
Etag**

src/EventListener/ConditionalRequestListener.php

```
<?php
namespace App\EventListener;

use Symfony\Bridge\Monolog\Logger;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpKernel\Event\GetResponseEvent;
use Symfony\Component\HttpKernel\Event\FilterResponseEvent;
use SymfonyBundles\RedisBundle\Redis\Client as RedisClient;

class ConditionalRequestListener
{
    protected $redis;
    protected $logger;
    public function __construct(RedisClient $redis)
    {
        $this->redis = $redis;
    }
    protected function isModified(Request $request, $etag)
    {
        if ($etags = $request->getETags()) {
            return in_array($etag, $etags) || in_array('*', $etags);
        }
        return true;
    }
}
...

```

src/EventListener/ConditionalRequestListener.php

```
{
    $this->redis = $redis;
    $this->logger = $logger;
}
protected function isModified(Request $request, $etag)
{
    if ($etags = $request->getETags()) {
        return in_array($etag, $etags) || in_array('*', $etags);
    }
    return true;
}
public function onKernelRequest(GetResponseEvent $event)
{
    $request = $event->getRequest();
    $etag = $this->redis->get('etag:'.md5($request->getUri()));
    if (!$this->isModified($request, $etag)) {
        $event->setResponse(Response::create('Not Modified', Response::HTTP_NOT_MODIFIED));
    }
}
public function onKernelResponse(FilterResponseEvent $event)
{
    $response = $event->getResponse();
    $request = $event->getRequest();

    $etag = md5($response->getContent());
    $response->setEtag($etag);
    if ($this->isModified($request, $etag)) {
        $this->redis->set('etag:'.md5($request->getUri()), $etag);
    }
}
}
```



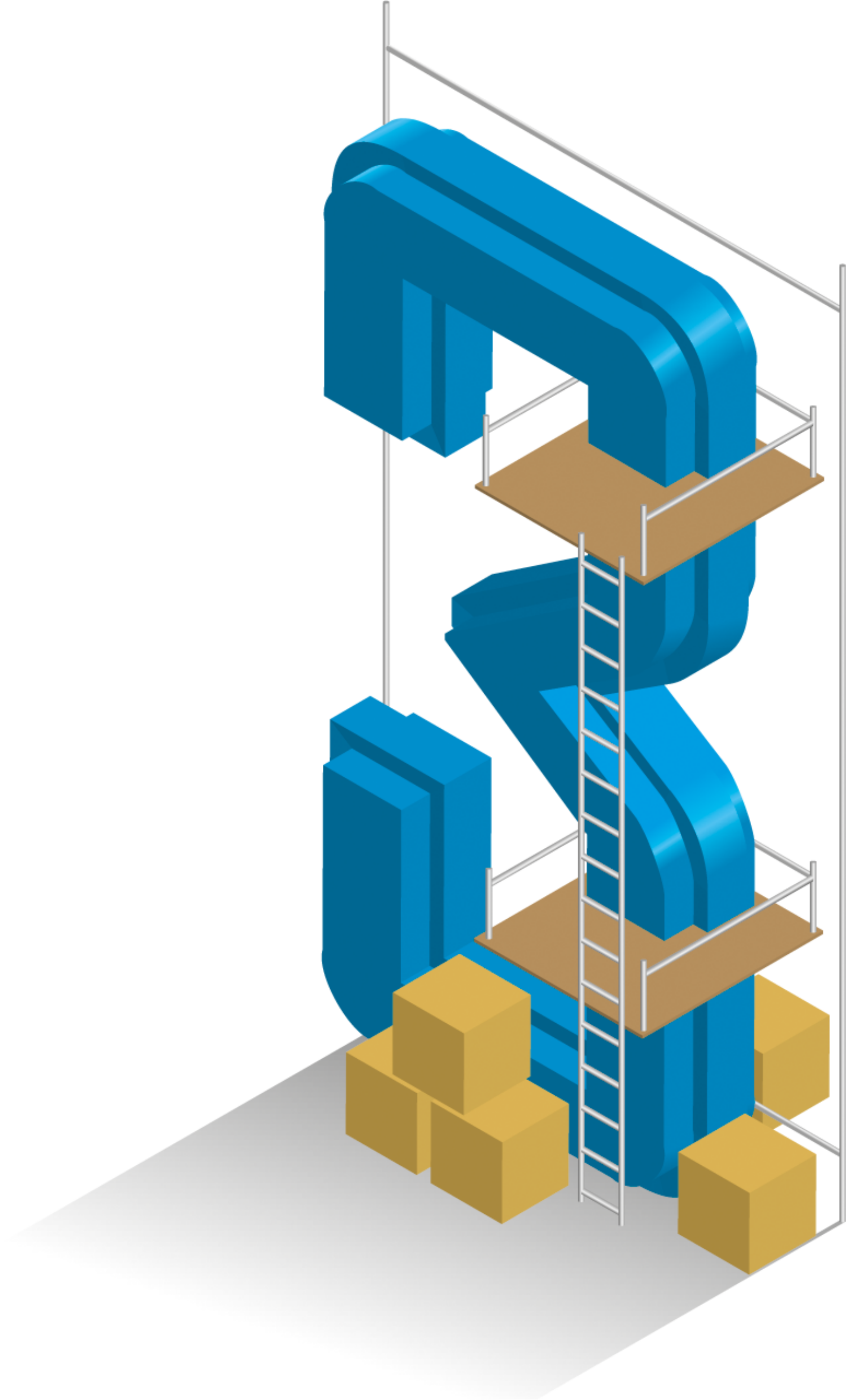
Drupal\page_cache\StackMiddleware\PageCache

```
$last_modified = $response->getLastModified();
if ($last_modified) {
    // See if the client has provided the required HTTP headers.
    $if_modified_since = $request->server->has('HTTP_IF_MODIFIED_SINCE') ? strtotime($request->server->get('HTTP_IF_MODIFIED_SINCE')) : FALSE;
    $if_none_match = $request->server->has('HTTP_IF_NONE_MATCH') ? stripslashes($request->server->get('HTTP_IF_NONE_MATCH')) : FALSE;

    if ($if_modified_since && $if_none_match
        // etag must match.
        && $if_none_match == $response->getEtag()
        // if-modified-since must match.
        && $if_modified_since == $last_modified->getTimestamp()) {
        $response->setStatusCode(304);
        $response->setContent(NULL);

        // In the case of a 304 response, certain headers must be sent, and the
        // remaining may not (see RFC 2616, section 10.3.5).
        foreach (array_keys($response->headers->all()) as $name) {
            if (!in_array($name, ['content-location', 'expires', 'cache-control', 'vary'])) {
                $response->headers->remove($name);
            }
        }
    }
}
```

Content composition & placeholders



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

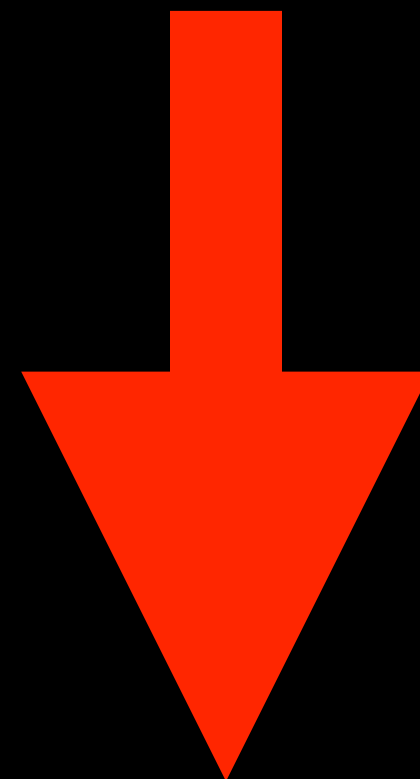
Main

Footer

**Shopping
cart or account
information**

session

cookie



No cache

Code

renders

single HTTP

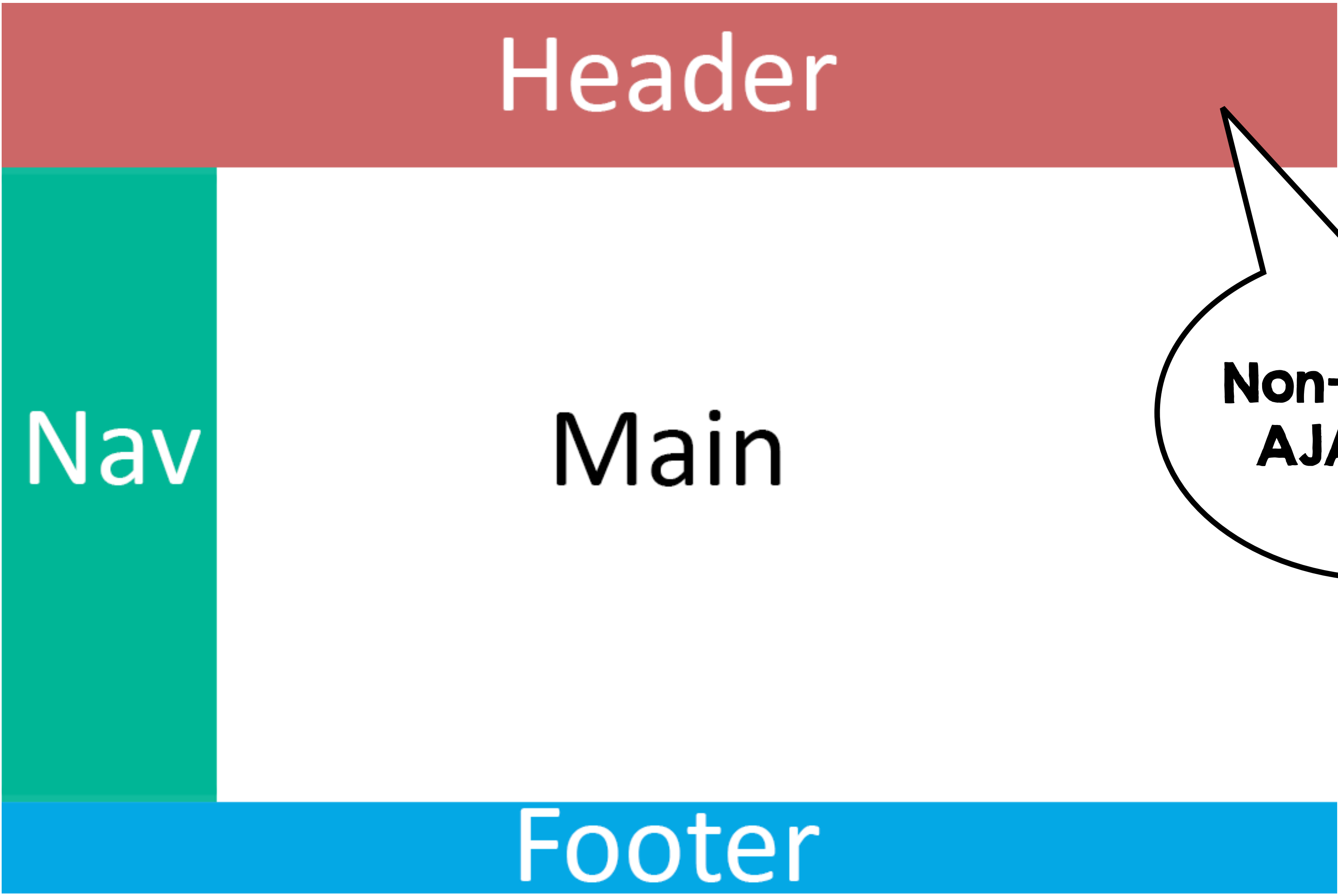
response

Lowest
common
denominator:
no cache

Placeholders



AJAX



Header

Nav

Main

Footer



**Non-cached
AJAX call**

Edge Side
includes

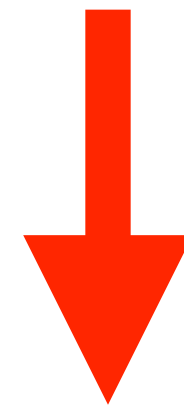
Edge Side Includes

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

- ✓ Placeholder
- ✓ W3C standard
- ✓ Parsed by Varnish
- ✓ Output is a composition of blocks
- ✓ State per block
- ✓ TTL per block

Varnish

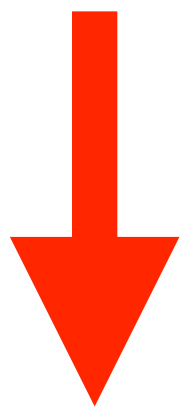
Surrogate-Capability: key="ESI/1.0"



Backend

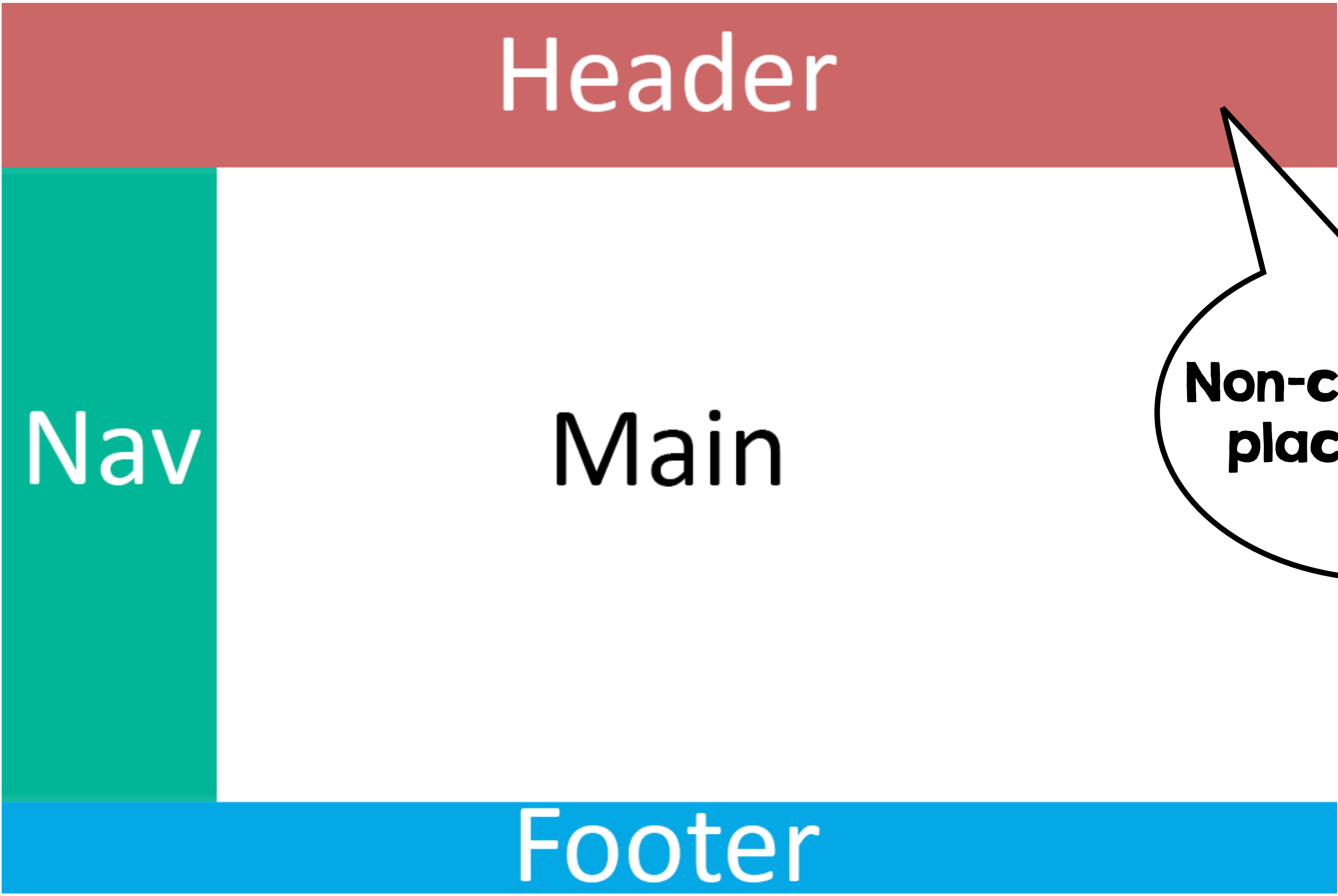
Surrogate-Control: content="ESI/1.0"

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



Varnish

Parse ESI placeholders



Non-cached ESI placeholder

ES

vs

AJAX

Edge-Side Includes

- ✓ **Server-side**
- ✓ **Standardized**
- ✓ **Processed on the “edge”, no in the browser**
- ✓ **Generally faster**

- **Sequential**
- **One fails, all fail**
- **Limited implementation in Varnish**

AJAX

- ✓ **Client-side**
- ✓ **Common knowledge**
- ✓ **Parallel processing**
- ✓ **Graceful degradation**
- **Processed by the browser**
- **Extra roundtrips**
- **Somewhat slower**

Composition
at the view
layer



Dynamic Page Cache

Drupal\dynamic_page_cache\EventSubscriber\DynamicPageCacheSubscriber

```
public function onRouteMatch(GetResponseEvent $event) {  
    // Don't cache the response if the Dynamic Page Cache request policies are  
    // not met. Store the result in a static keyed by current request, so that  
    // onResponse() does not have to redo the request policy check.  
    $request = $event->getRequest();  
    $request_policy_result = $this->requestPolicy->check($request);  
    $this->requestPolicyResults[$request] = $request_policy_result;  
    if ($request_policy_result === RequestPolicyInterface::DENY) {  
        return;  
    }  
  
    // Sets the response for the current route, if cached.  
    $cached = $this->renderCache->get($this->dynamicPageCacheRedirectRenderArray);  
    if ($cached) {  
        $response = $this->renderArrayToResponse($cached);  
        $response->headers->set(self::HEADER, 'HIT');  
        $event->setResponse($response);  
    }  
}
```

Auto-placeholding

```
<?php
```

```
use Drupal\Core\Block\BlockBase;
```

```
class TheDateTimeBlock extends BlockBase  
{
```

```
    /**
```

```
     * {@inheritdoc}
```

```
    */
```

```
    public function build()  
{
```

```
        return [  
            '#markup' => $this->t('The current date and time is @time', [  
                '@time' => date('Y-m-d H:i:s')  
            ]),  
            '#cache' => [  
                'max-age' => 5,  
                'tags' => ['block:the_date_time']  
            ],  
            '#create_placeholder' => false  
        ];
```

```
    }
```

```
}
```

BigPipe

Accept-Ranges: bytes
Age: 0
Cache-Control: must-revalidate, no-cache, private
Content-Encoding: gzip
Content-language: en
Content-Type: text/html; charset=UTF-8
Date: Thu, 22 Nov 2018 16:30:40 GMT
Expires: Sun, 19 Nov 1978 05:00:00 GMT
Server: nginx/1.10.3
Surrogate-Control: no-store, content="BigPipe/1.0"
Transfer-Encoding: chunked
Vary: Accept-Encoding
Via: 1.1 varnish (Varnish/6.0)
X-Content-Type-Options: nosniff
X-Drupal-Dynamic-Cache: UNCACHEABLE
X-Frame-Options: SAMEORIGIN
X-Generator: Drupal 8 (<https://www.drupal.org>)
X-UA-Compatible: IE=edge
X-Varnish: 196708
X-Varnish-Cache: MISS

```
<span data-big-pipe-placeholder-  
id="callback=Drupal%5CCore%5CRender%5CElem  
ent%5CStatusMessages%3A%3ArenderMessages&a  
mp;args%5B0%5D&amp;token=_HAdUpwWmet0T0Te2  
PSiJuMntExoshbm1kh2wQzzzAA"></span>
```



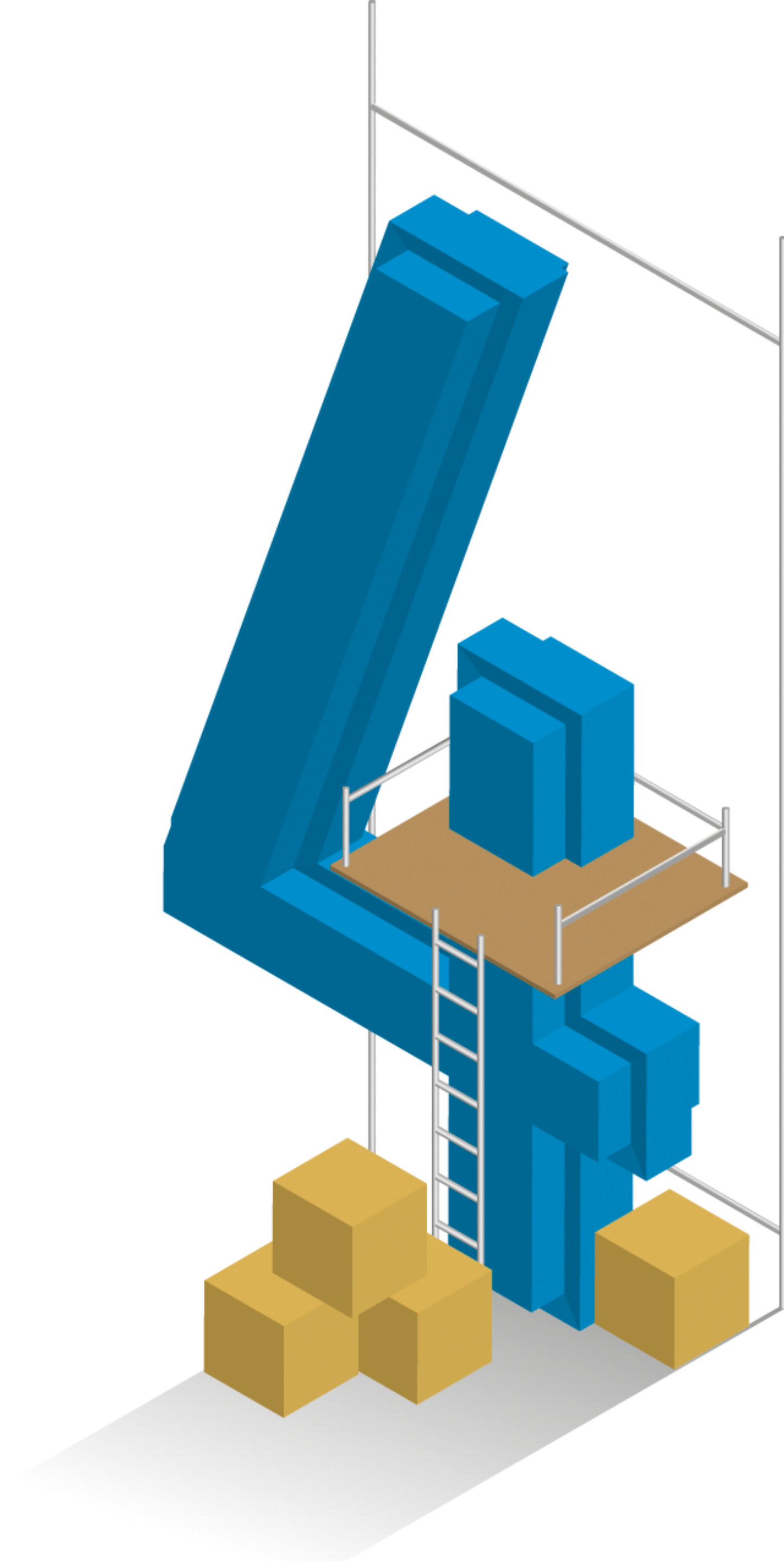
```
<script type="application/vnd.drupal-ajax" data-big-pipe-  
replacement-for-placeholder-with-  
id="callback=Drupal%5CCore%5CRender%5CElement%5CStatusMessages%3A%3A  
renderMessages&args%5B0%5D&token=_HAdUpwWmet0T0Te2PSiJuMntEx  
oshbm1kh2wQzzzAA">
```

```
[{"command":"insert","method":"replaceWith","selector":"[data-big-  
pipe-placeholder-  
id=\"callback=Drupal%5CCore%5CRender%5CElement%5CStatusMessages%  
3A%3ArenderMessages%5B0%5D%5Ctoken=_HAdUpwWmet0T0Te2PSi  
JuMntExoshbm1kh2wQzzzAA\"]", "data":"","settings":null}]
```

```
</script>
```

Don't cache BigPipe responses in Varnish

```
if (beresp.http.Surrogate-Control ~ "BigPipe/1.0") {  
    set beresp.do_stream = true;  
    set beresp.ttl = 0s;  
}
```



Cache
variations



How do you identify
an object in cache?

The URL identifies
objects in cache



What if the content
of a URL varies
based on the value
of a request
header?

Cache variations

GET / HTTP/1.1

Host: localhost

Accept-Language: en, nl, de

HTTP/1.1 200 OK

Host: localhost

Content-Language: en

Content-type: text/html; charset=UTF-8

Hello world output

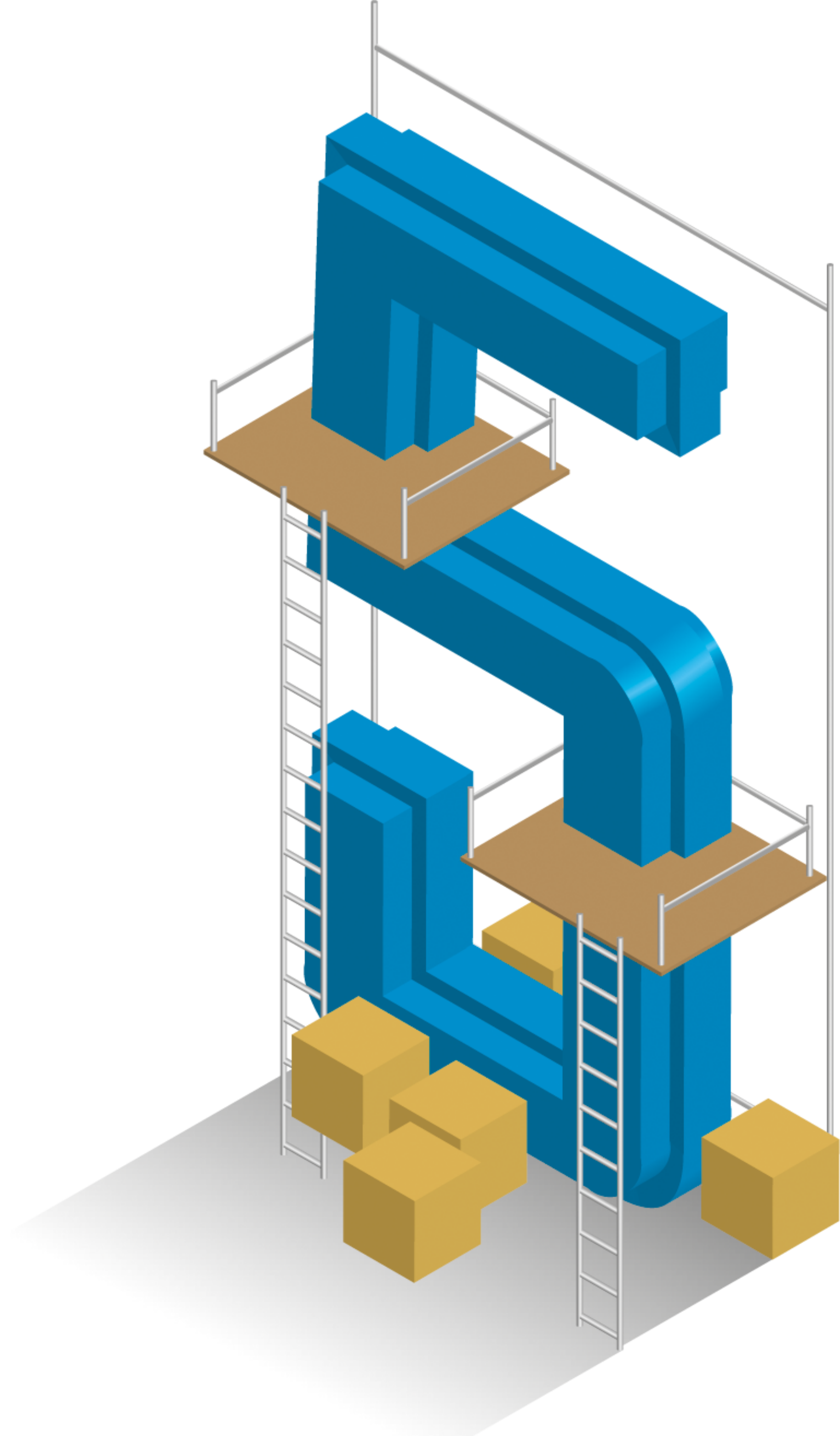


**Response
header**



**Request
header
value**

Vary: Accept-Language



Content
invalidation

There's only one
thing worse than
not caching
enough

It's caching too
much or too long

Purgging





Required modules

- ✓ **purge**
- ✓ **purge_purger_http**

▼ PURGE

- Purge** ▶ Provides a generic external cache invalidation API and queue service.
- Purge Drush** ▶ Administrative Drush commands for Purge.
- Purge Tokens** ▶ Adds tokens support for external cache invalidation.
- Purge UI** ▶ Administrative interface for Purge.

▼ PURGE – PROCESSORS

- Cron processor** ▶ Processes the queue every time cron runs, recommended for most configurations.
- Late runtime processor** ▶ Process the queue on every request, this is only recommended on high latency configurations.

▼ PURGE – QUEUERS

- Core tags queuer** ▶ Queues every tag that Drupal invalidates internally.

▼ PURGE – REVERSE PROXIES & CDNS

- Generic HTTP Purger** ▶ Generic and highly configurable purger that makes HTTP requests, best suits custom configurations.
- Generic HTTP Tags Header** ▶ Exports the Purge-Cache-Tags response header to support tag-based invalidation. You only need this for

▼ CACHE INVALIDATION

Each layer of caching on top of your site is cleared by a purger. Purgers are provided by third-party modules and support one or more types of cache invalidation.

CACHING LAYER	URL	P	TAG	E	R	U	PATH	D
Drupal Origin			✓					
Varnish purger			✓					
Add purger								
Public Endpoint								

▼ QUEUE

The queue holds items that need refreshing, hold your mouse over the column titles for more details.

QUEUERS	QUEUE	PROCESSORS
Drush p-queue-add	Database	Drush p-queue-work
Core tags queuer		Drush p-invalidate
Purge block(s)		Cron processor
		Late runtime processor
		Purge block(s)

Configure Varnish purger



Name *

A label that describes this purger.

Type

What sort of item will this purger clear?

Request

Headers

Body

Performance

Tokens

Success resolution

In this section you configure how a single HTTP request looks like.

Hostname

Port

Path

Request Method

Scheme

Cancel

Save configuration

Configure Varnish purger



Name *

A label that describes this purger.

Type

What sort of item will this purger clear?

Request

Headers

Body

Performance

Tokens

Success resolution

Configure the outbound HTTP headers, leave empty to delete.

HEADER

VALUE

Add header

Cancel

Save configuration

```
# Only allow BAN requests from IP addresses in the 'purge' ACL.
if (req.method == "BAN") {
    # Same ACL check as above:
    if (!client.ip ~ purge) {
        return (synth(403, "Not allowed.));
    }

    if (req.http.Purge-Cache-Tags) {
        ban("obj.http.Purge-Cache-Tags ~ " + req.http.Purge-Cache-Tags);
    }
    else {
        return (synth(403, "Purge-Cache-Tags header missing.));
    }

    # Throw a synthetic page so the request won't go to the backend.
    return (synth(200, "Ban added.));
}
```

```
acl purge {  
    "127.0.0.1";  
}
```

Purge-Cache-Tags: block:the_date_time block_view
config:block.block.bartik_account_menu config:block.block.bartik_branding
config:block.block.bartik_breadcrumbs config:block.block.bartik_content
config:block.block.bartik_footer config:block.block.bartik_help
config:block.block.bartik_local_actions
config:block.block.bartik_local_tasks config:block.block.bartik_main_menu
config:block.block.bartik_messages config:block.block.bartik_page_title
config:block.block.bartik_powered config:block.block.bartik_search
config:block.block.bartik_tools config:block.block.thedateandtimeblock
config:block_list config:color.theme.bartik config:filter.format.basic_html
config:search.settings config:system.menu.account config:system.menu.footer
config:system.menu.main config:system.menu.tools config:system.site
config:user.role.anonymous config:views.view.frontpage http_response node:1
node:2 node_list node_view rendered user:1

High TTL +
punging

LOW TTL +
conditional
requests



You're not done yet

Only keep Drupal cookies in Varnish

```
vcl 4.0;
```

```
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 == "") {
            unset req.http.Cookie;
        }
        else {
            return (pass);
        }
    }
}
```

THE END



O'REILLY®

Compliments of
VARNISH
SOFTWARE



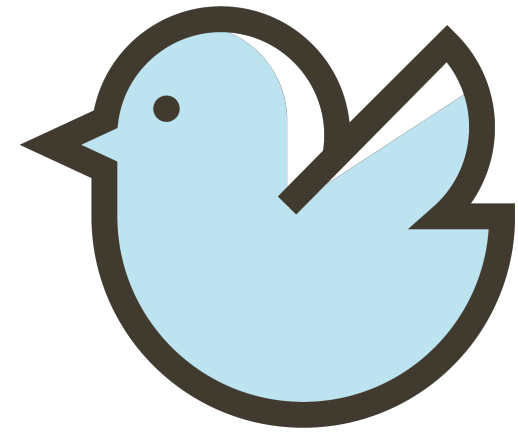
Getting Started with Varnish Cache

ACCELERATE YOUR WEB APPLICATIONS

Thijs Feryn



<https://feryn.eu>



<https://twitter.com/ThijsFeryn>



<https://instagram.com/ThijsFeryn>