

Mastering Symfony2

SF2C2

Sensio**Labs**

Chapter 5

Caching

Introduction to HTTP Caching

HTTP Caching

The best way to improve the performance of an application is probably to cache its output and bypass it altogether.

HTTP Caching

Of course, this is very difficult
for highly dynamic websites!

HTTP Caching

Symfony2 cache system relies on the simplicity and power of the HTTP cache as defined in the HTTP specification (RFC2616).

HTTP Caching

Basically, if you already know HTTP **validation** and **expiration** caching models, you are ready to use most of the Symfony caching layer.

HTTP Caching – Kind of Caches

- Browser caches: Every browser comes with its own local cache that is mainly useful for when you hit “back” or when images are reused throughout a website;
- Proxy caches: A proxy is a shared cache as many people can be behind a single one. It’s usually installed by large corporations and ISPs to reduce latency and network traffic.
- Gateway caches: Like a proxy, it’s also a shared cache but on the server side. Installed by network administrators, it makes websites more scalable, reliable and performing better (CDNs like Akamai are gateway caches).

Browser caches

Client

Browser cache

Web Server

Client

Browser cache

Proxy caches

company1

Web Server

Client

Browser

Proxy cache

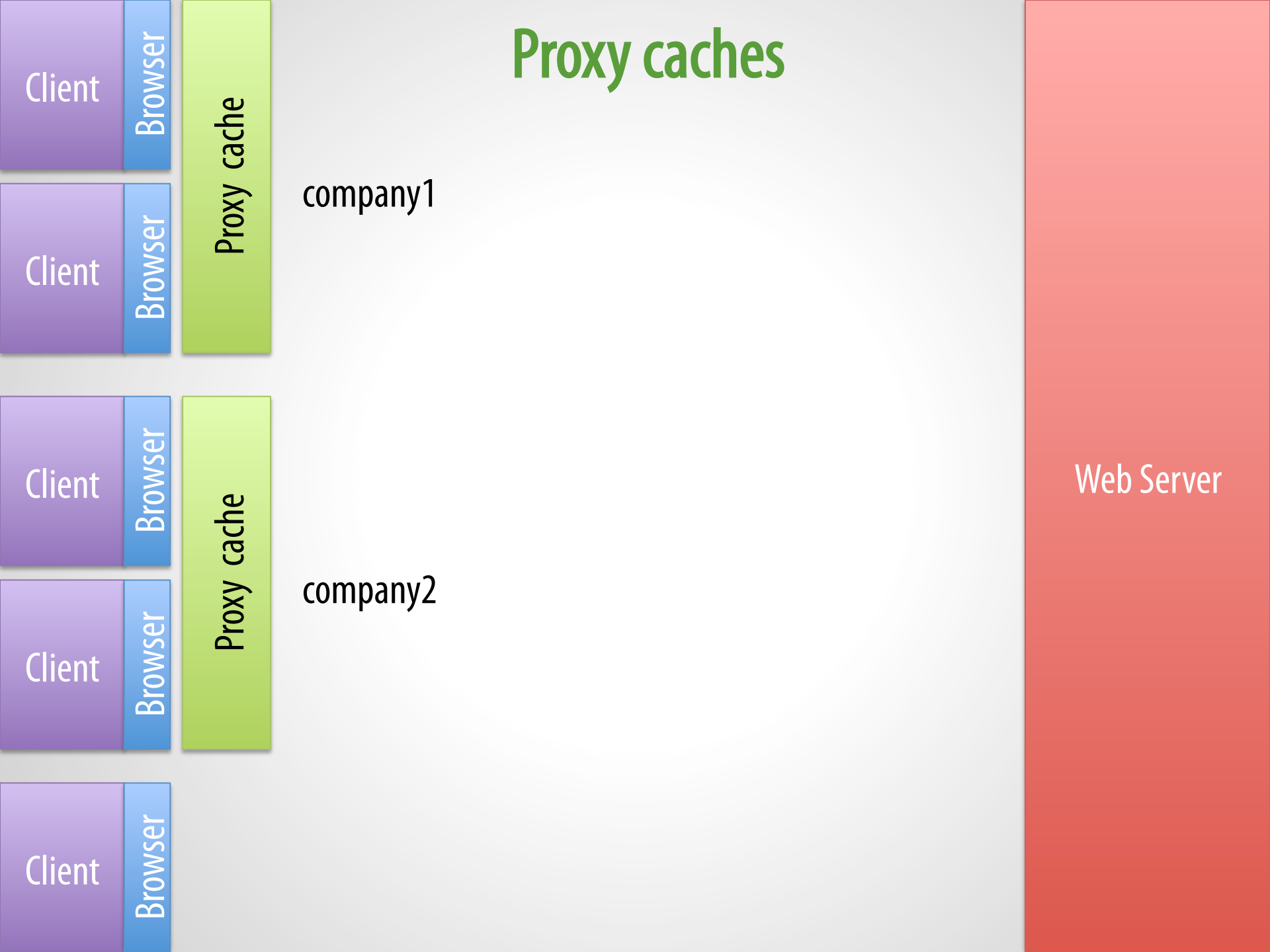
Client

Browser

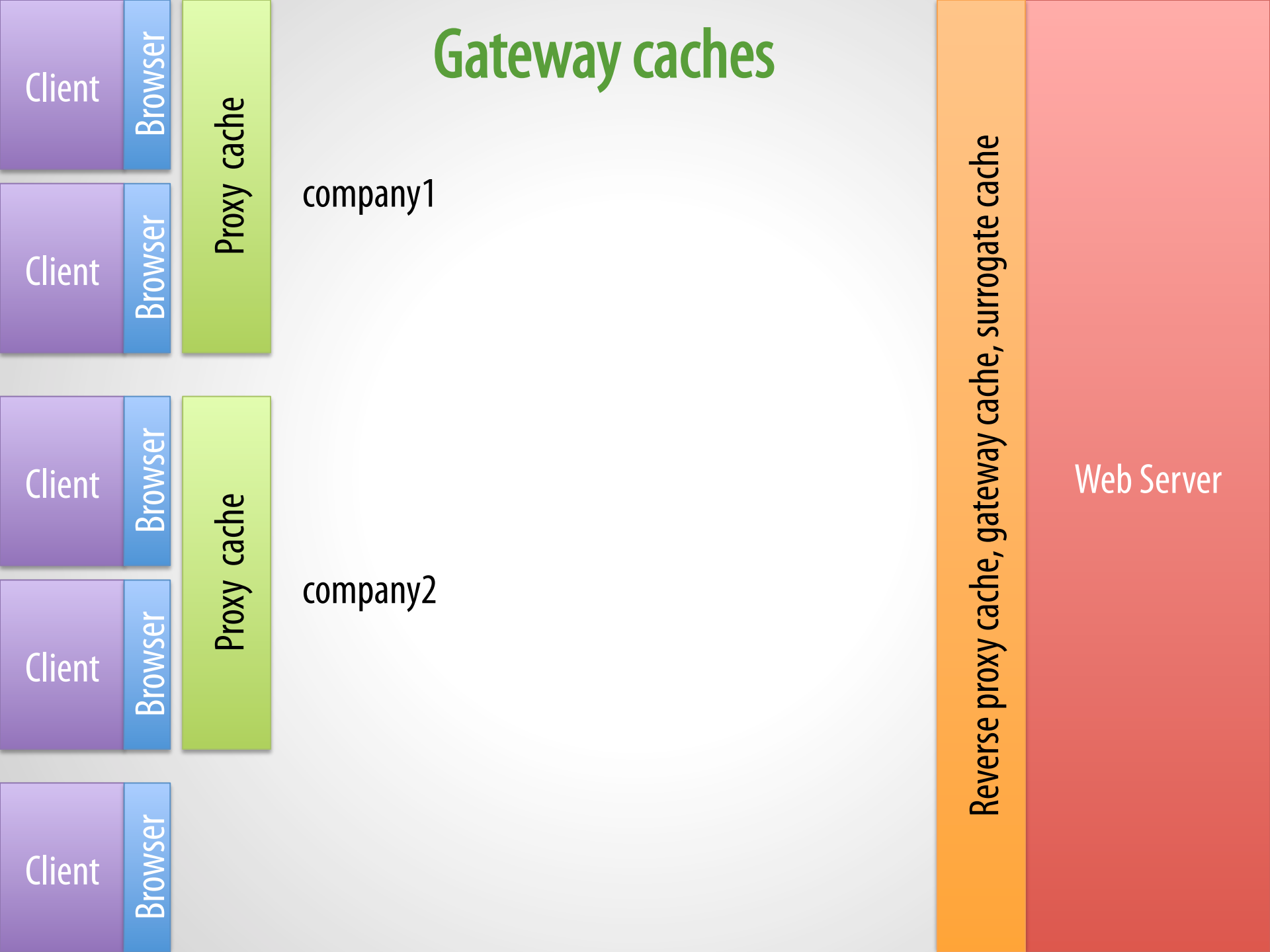
Client

Browser

Proxy caches



Gateway caches



What can be cached?

Only « safe » methods like GET and HEAD can be cached as they don't change the state of the resource.

Don't expect to cache a resource accessible from a PUT, DELETE or POST HTTP method.

HTTP Caching Models

HTTP Caching – Response API

Symfony2 provides a verbose and simple API for the **Response** object to manage response HTTP caching headers like **Expires**, **Etag**, **Last-Modified**, or **Cache-Control**.

```
# pass an array of headers as the third argument to the Response constructor
$response = new Response($content, $status, $headers);

# set a header value
$response->headers->set('Content-Type', 'text/plain');

# add a header value to the existing values
$response->headers->set('Vary', 'Accept', false);

# set a multi-valued header
$response->headers->set('Vary', array('Accept', 'Accept-Encoding'));

# delete a header
$response->headers->delete('Content-Type');
```

HTTP Caching – Expiration Model

The goal is to specify how long a **response** should be considered « **fresh** » by including a **Cache-Control** and/or an **Expires** header.

Caches that understand expiration will not make the same request until the cached version reaches its expiration time and becomes « **stale** ».

Advantage: save some CPU resources.

HTTP Caching – Validation Model

When some pages are really dynamic, the validation model uses a **unique identifier** and/or a **timestamp** to check if the page changed since the last request.

Identifiers are defined with the **Etag** header whereas timestamps with the **Last-Modified** response header field.

Advantage: reduce bandwidth usage.

HTTP Caching

The goal of both models is to never generate the same response twice.

HTTP Caching – Specification

An effort to rewrite the RFC 2616 is on the way.

- P4 – Conditional Requests
- P6 – Caching: browser and intermediary caches

<http://tools.ietf.org/wg/httpbis/>

Default Caching Strategy with Symfony2

By default, Symfony2 asks browsers (« **private** ») to not cache the page at all (« **no-cache** »).

En-têtes	Réponse	Cache
Réponse	voir le code source	
	Date Wed, 16 Feb 2011 16:14:51 GMT	
	Server Apache/2.2.17 (Unix) mod_ssl/2.2.17 OpenSSL/1.0.0c DAV/2 SVN/1.6.15 PHP/5.3.5	
	X-Powered-By PHP/5.3.5	
	Cache-Control private, must-revalidate	
	Expires Wed, 16 Feb 2011 16:15:01 GMT	
	x-debug-token 4d5bf7fbb10db	
	Connection close	
	Content-Type text/html; charset=UTF-8	
Requête	voir le code source	
	Host www.sf2-sflive.local	
	User-Agent Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10.6; fr; rv:1.9.2.13) Gecko/20101203 Firefox/3.6.13	
	Accept text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	
	Accept-Language fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3	
	Accept-Encoding gzip,deflate	
	Accept-Charset ISO-8859-1,utf-8;q=0.7,*;q=0.7	
	Keep-Alive 115	
	Connection keep-alive	
	Cookie PHPSESSID=rvnkkngi4t2nn3278o7rqlgdo7; _SESS=evdsgoub3rkgmtlajn4kc67417	
	Cache-Control max-age=0	

Expiration model

Expiration in Practice – Expires Header field

The « **Expires** header field gives the date/time after which the response is considered stale ».

```
$response->setExpires(new \DateTime('+600 seconds'));
```

The expiration date must be a valid GMT date format.

Expiration in Practice – Expires Header field

```
/**
 * @Route("/about")
 * @Template
 * @Cache(expires="+6 hours")
 */
public function aboutAction()
{
    return array();
}
```

Wed, March 2nd 16:00:00

GET /about HTTP/1.0
Host: http://domain.local

HTTP Request

GET /about HTTP/1.0
Host: http://domain.local

HTTP Request

?

Browser cache

Web Server

HTTP/1.1 200 OK
Expires: Thu, 03 Mar 2011 16:00:00 GMT

Some content

HTTP Response

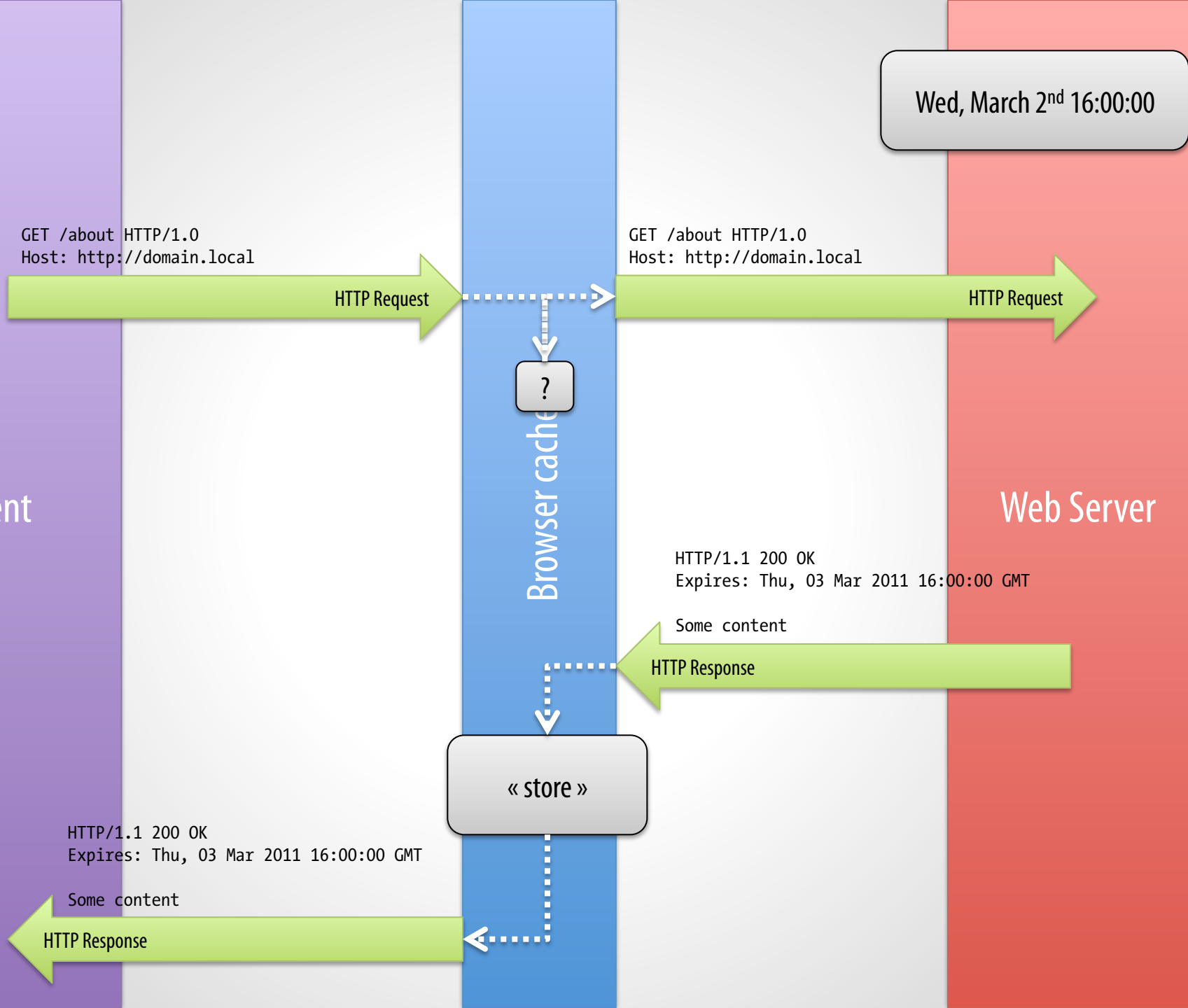
« store »

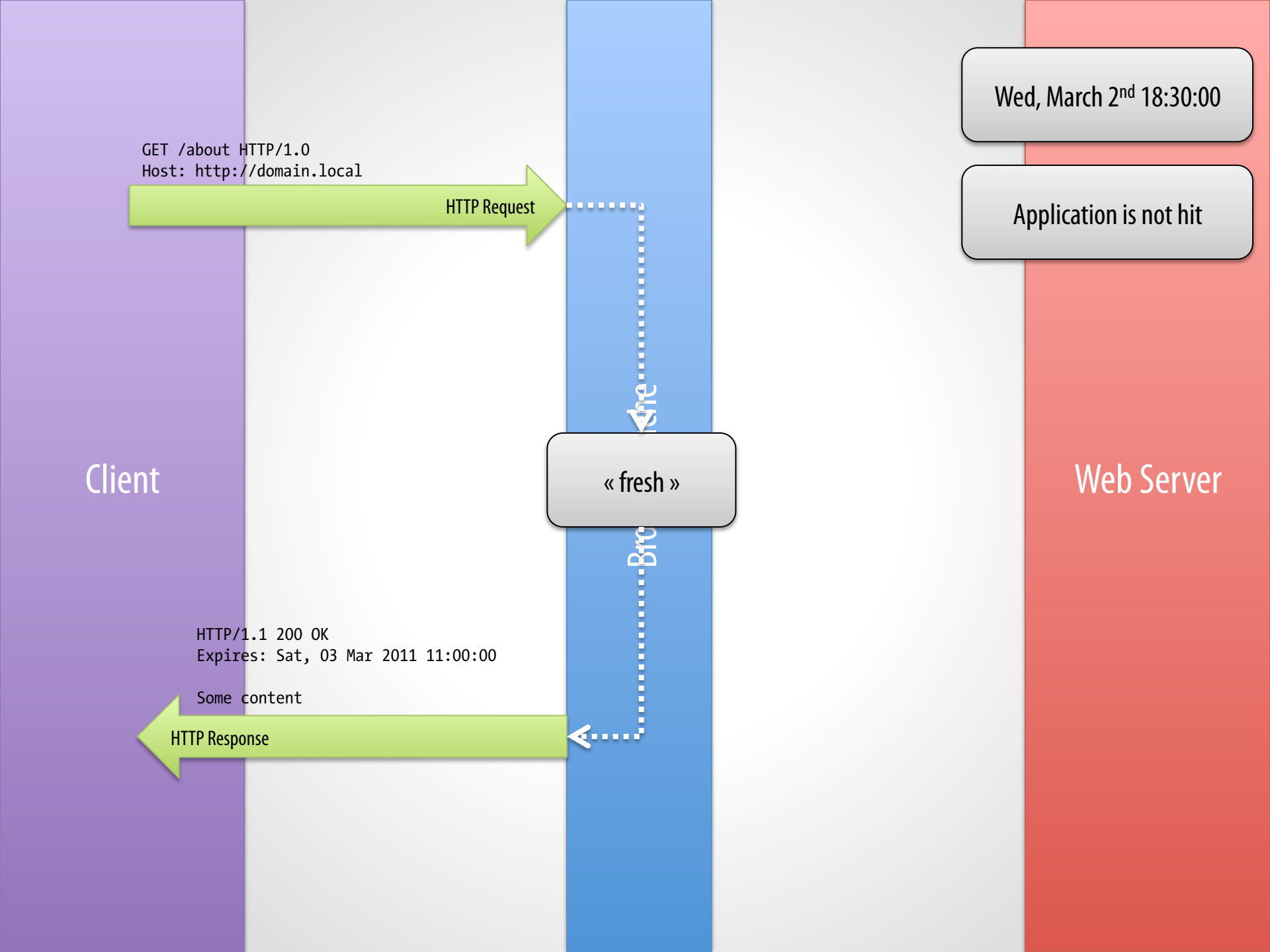
HTTP/1.1 200 OK
Expires: Thu, 03 Mar 2011 16:00:00 GMT

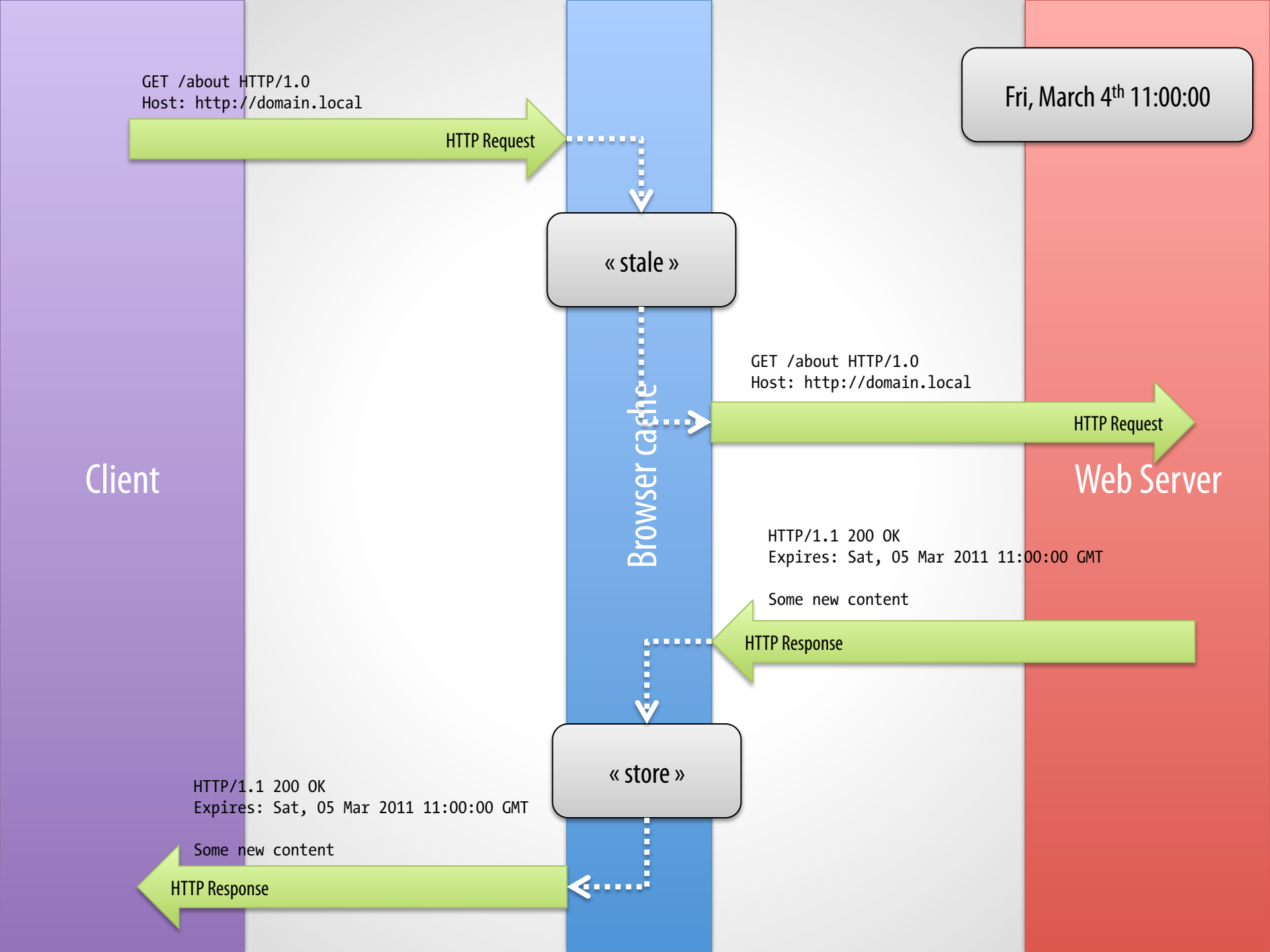
Some content

HTTP Response

Client







Expiration in Practice – Expires Header field

First limitation:

Both clocks of the browser and the server must be synchronized.

Second limitation:

The specification states that servers should not send « Expires » dates more than one year in the future.

Cache-Control Header field

The HTTP 1.1 protocol introduces the « Cache-Control » header field that is responsible to define caching strategy by specifying several directives.

For expiration, there are « max-age » and « s-maxage » directives that consider a resource « fresh » for a number of seconds since the date/time the response was generated.

```
Cache-Control: private, max-age=60
```

Browser side caching

```
$response->setMaxAge(600);
```

```
@Cache(maxage=600)
```

Server side caching

```
$response->setSharedMaxAge(600);
```

```
@Cache(smaxage=600)
```

Wed, March 2nd 16:00:00

Client

GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

?

Browser cache

Web Server

HTTP/1.1 200 OK
Cache-Control: private, max-age=3600

Some content

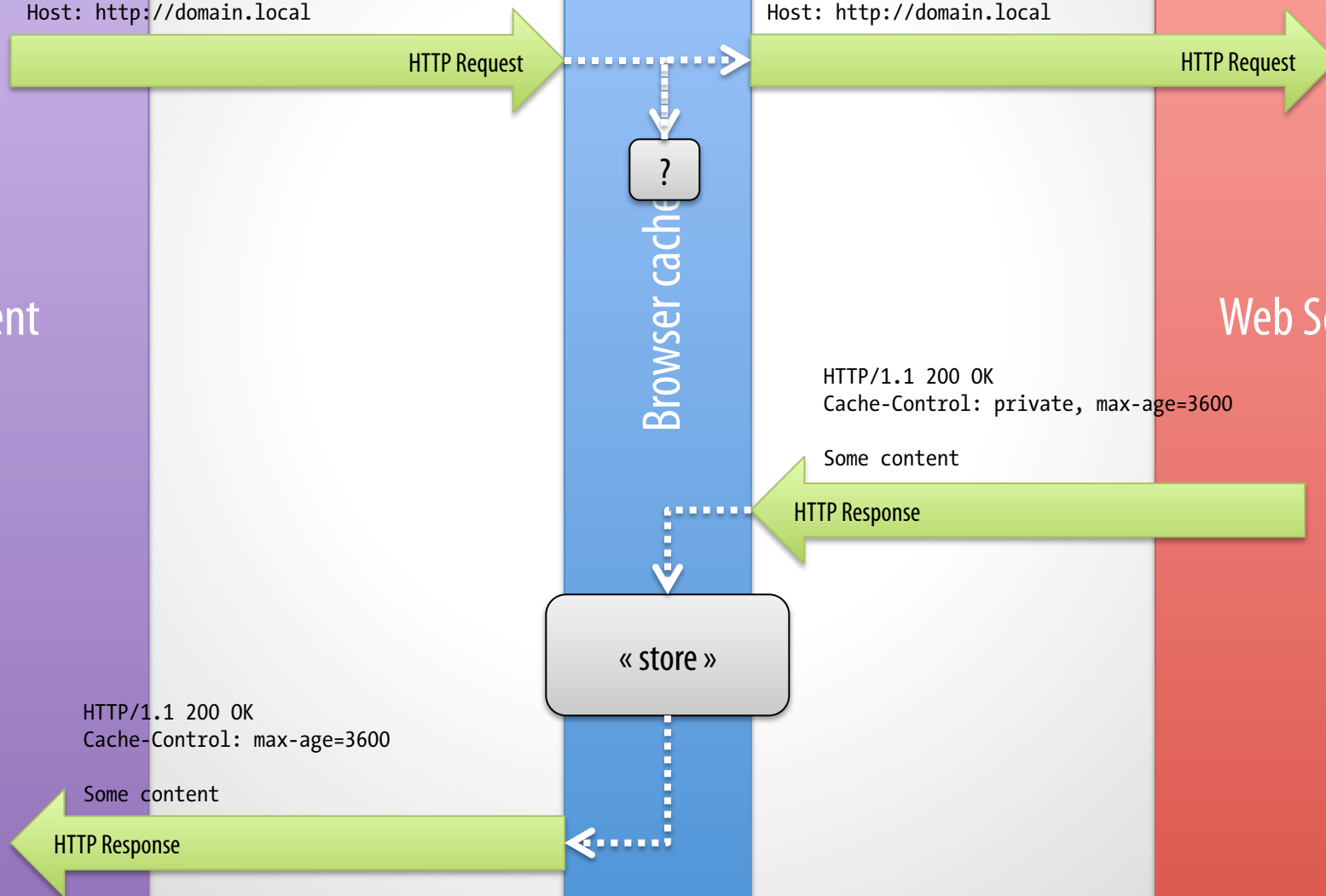
HTTP Response

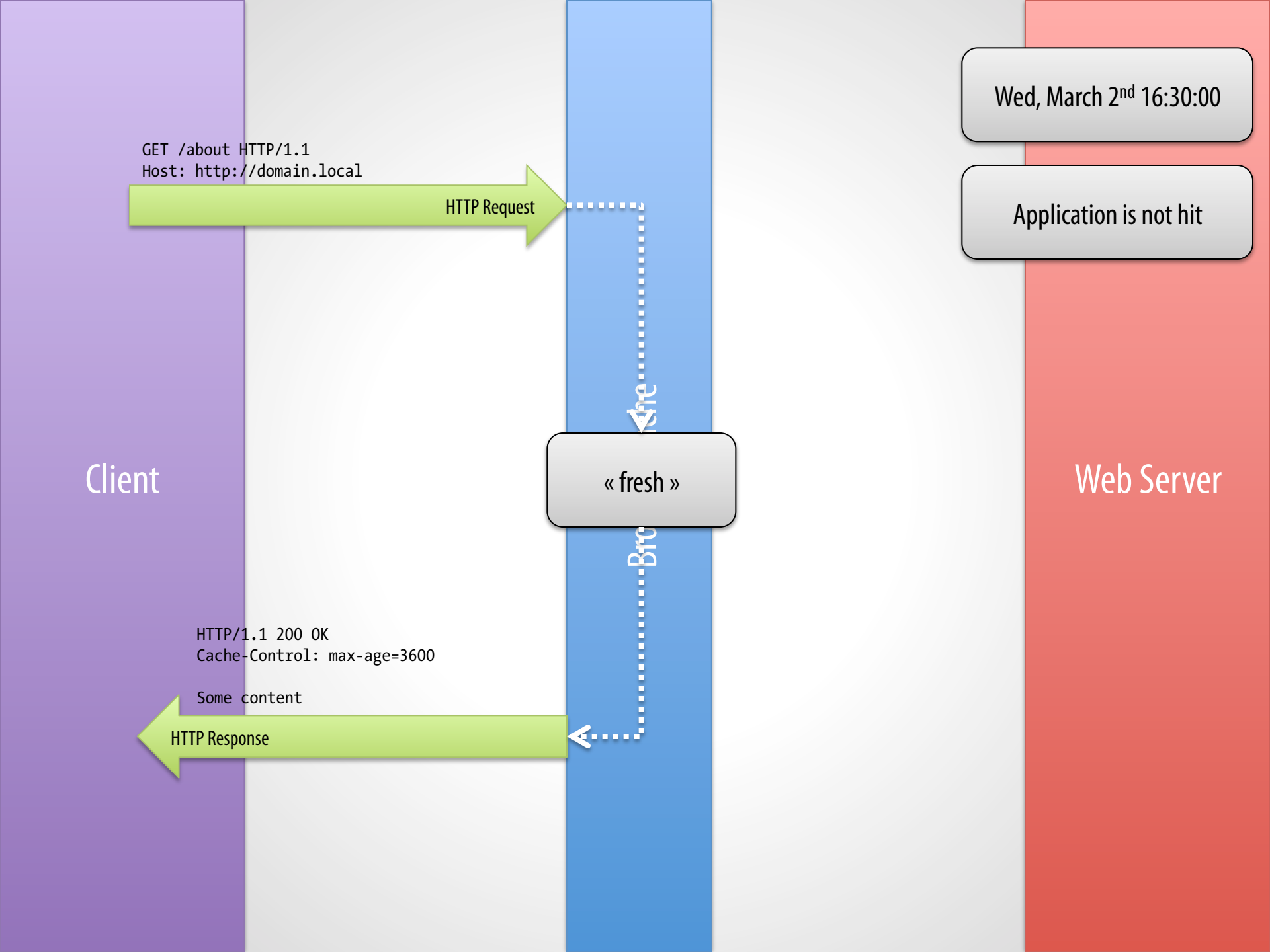
« store »

HTTP/1.1 200 OK
Cache-Control: max-age=3600

Some content

HTTP Response





GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

Wed, March 2nd 16:30:00

Application is not hit

Client

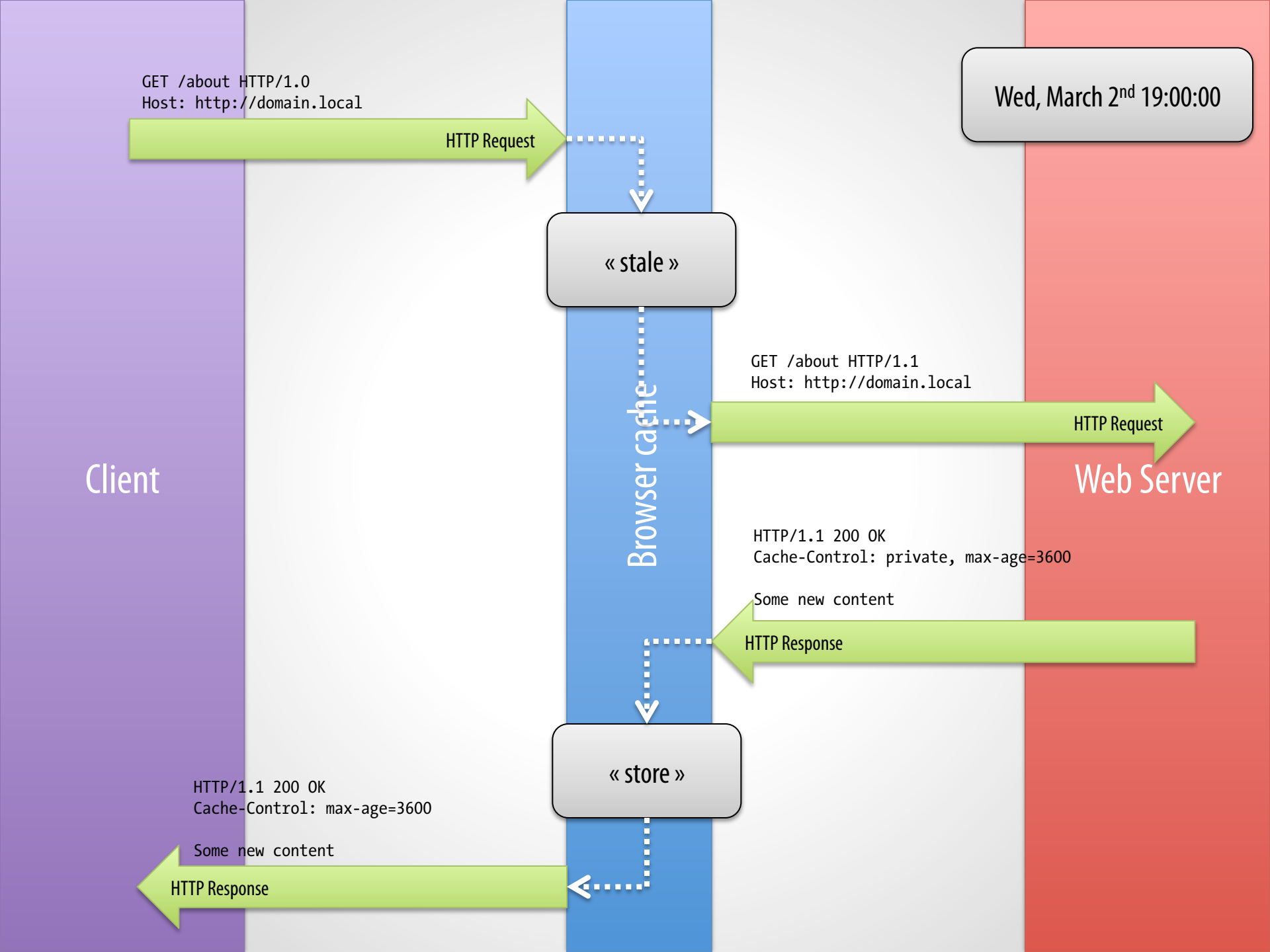
« fresh »

Web Server

HTTP/1.1 200 OK
Cache-Control: max-age=3600

Some content

HTTP Response



Validation model

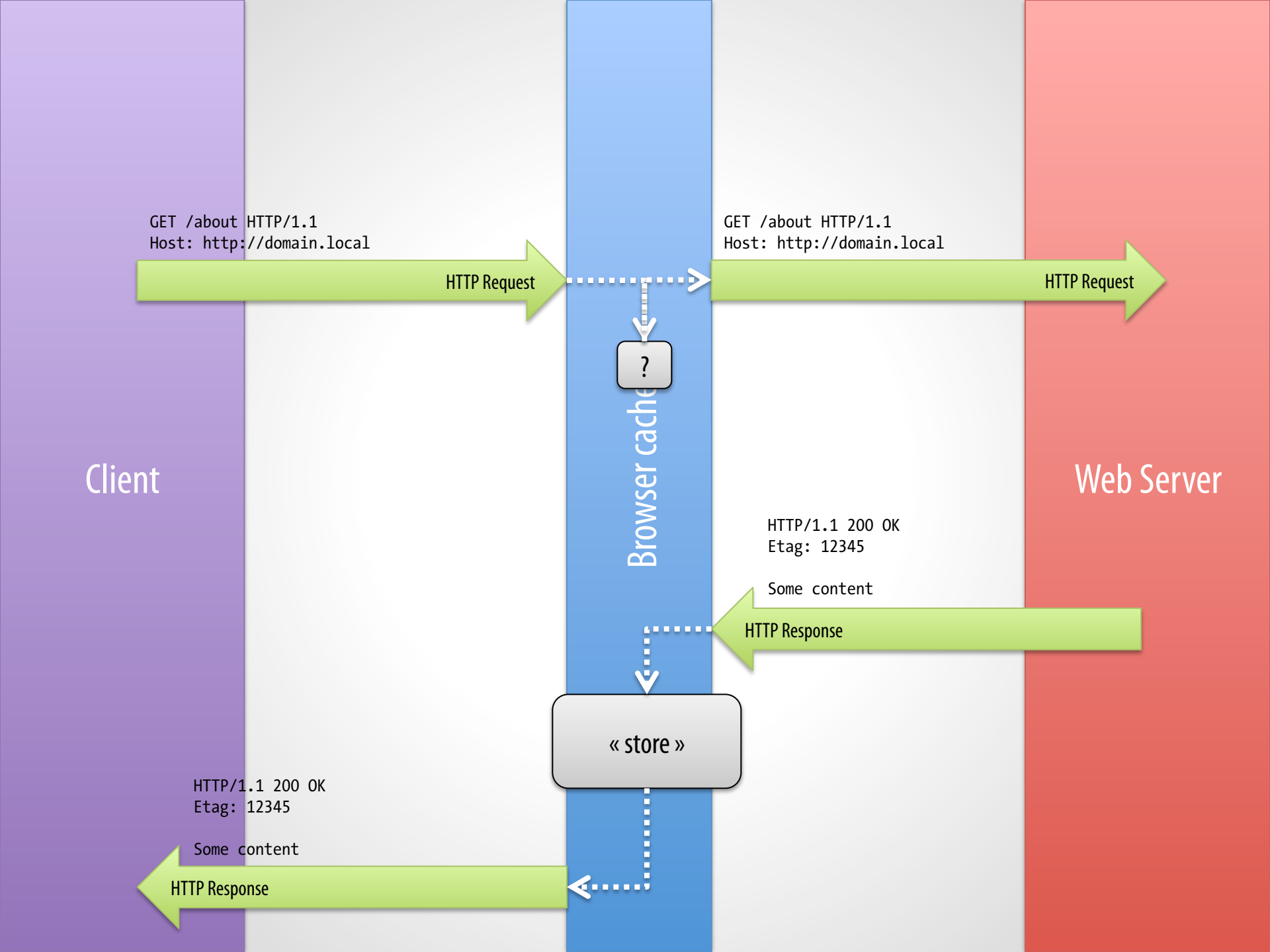
Validation in Practice

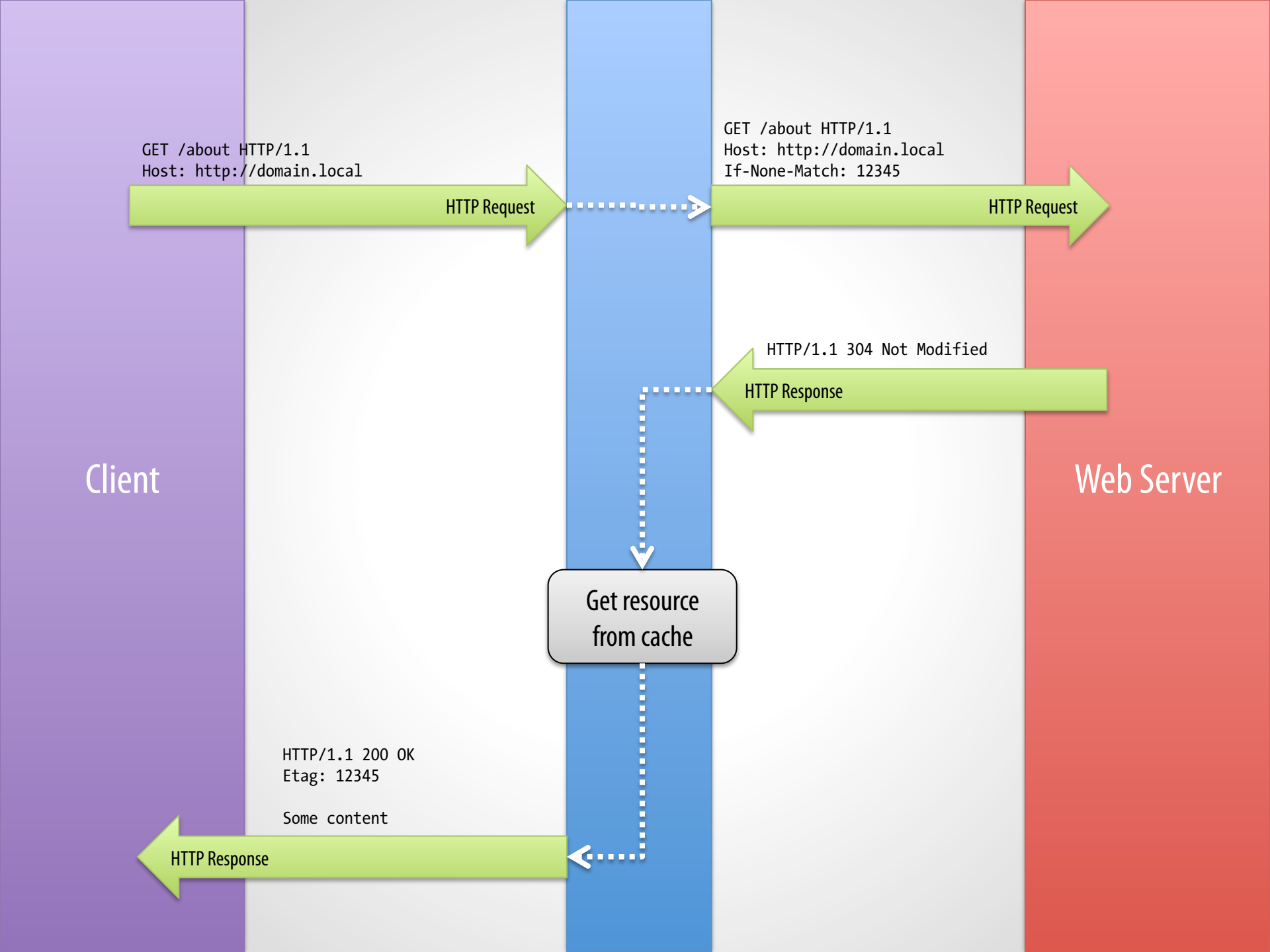
200 / 304

Validation in Practice – Etag Header Field

The « **Etag** » response-header field provides the current value of the entity-tag for one representation of the target resource.

```
$response->setETag( ' abc123456def' );
```





Client

GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

GET /about HTTP/1.1
Host: http://domain.local
If-None-Match: 12345

HTTP Request

HTTP/1.1 200 OK
Etag: abcdef

Some new content

HTTP Response

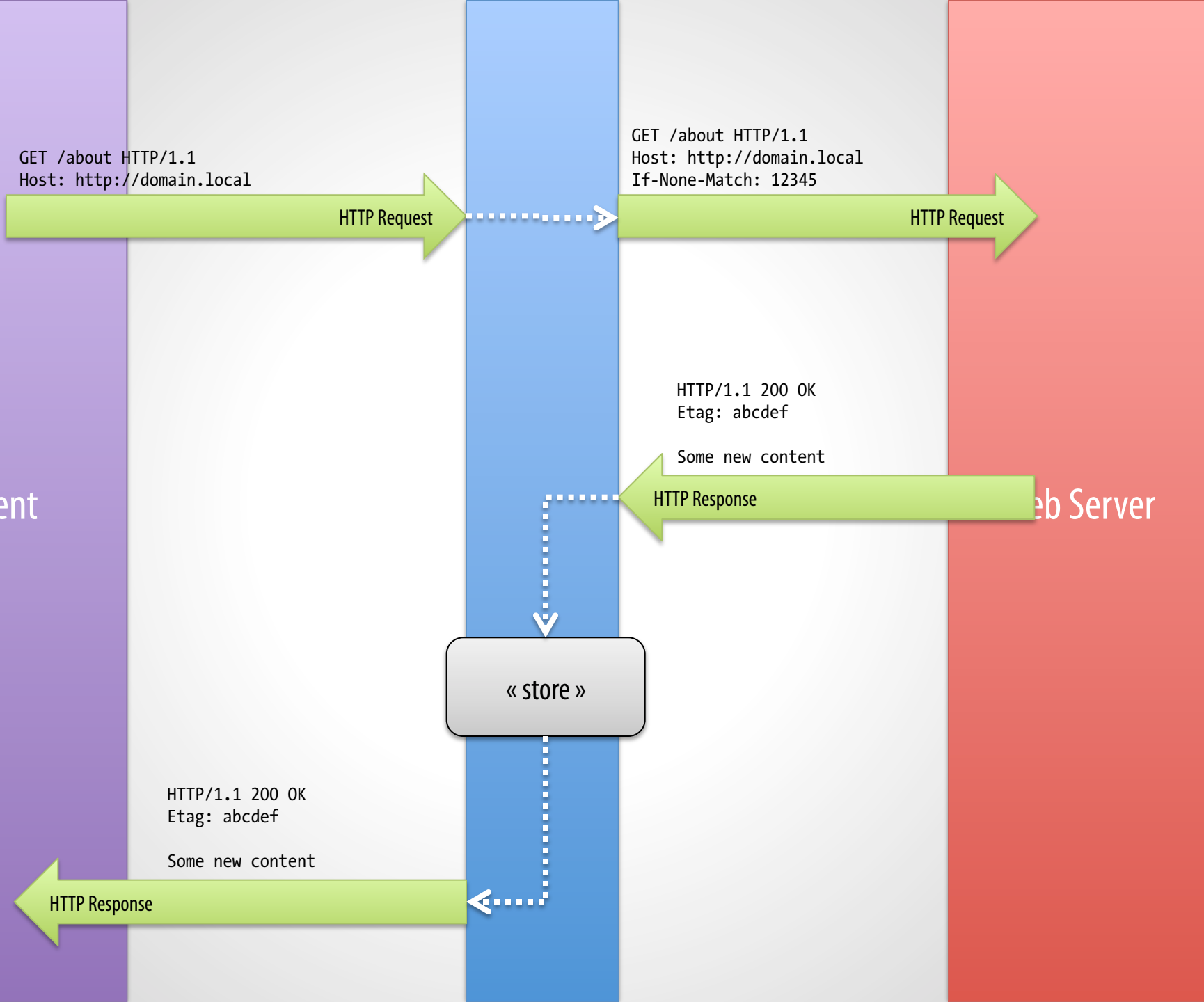
Web Server

« store »

HTTP/1.1 200 OK
Etag: abcdef

Some new content

HTTP Response



Validation in Practice – Last-Modified Header Field

The « **Last-Modified** response-header field indicates the date and time at which the origin server believes the representation was last modified ».

```
$article = ArticleDAO::findById($id);  
$response->setLastModified($article->getUpdatedAt());
```

Client

GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

?

Browser cache

GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

Web Server

HTTP/1.1 200 OK
Last-Modified: Wed, 02 Mar 2011 17:00:00 GMT

Some content

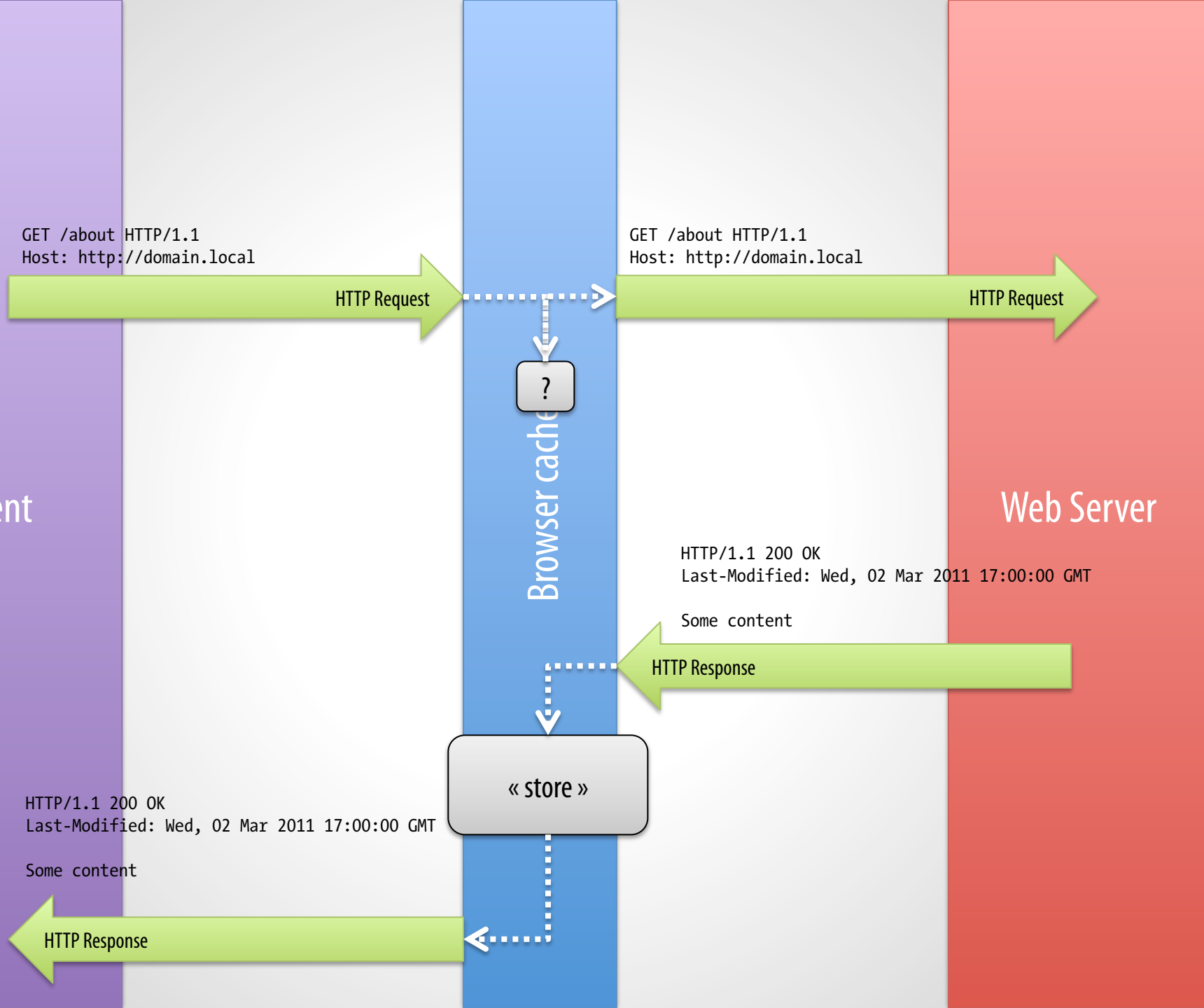
HTTP Response

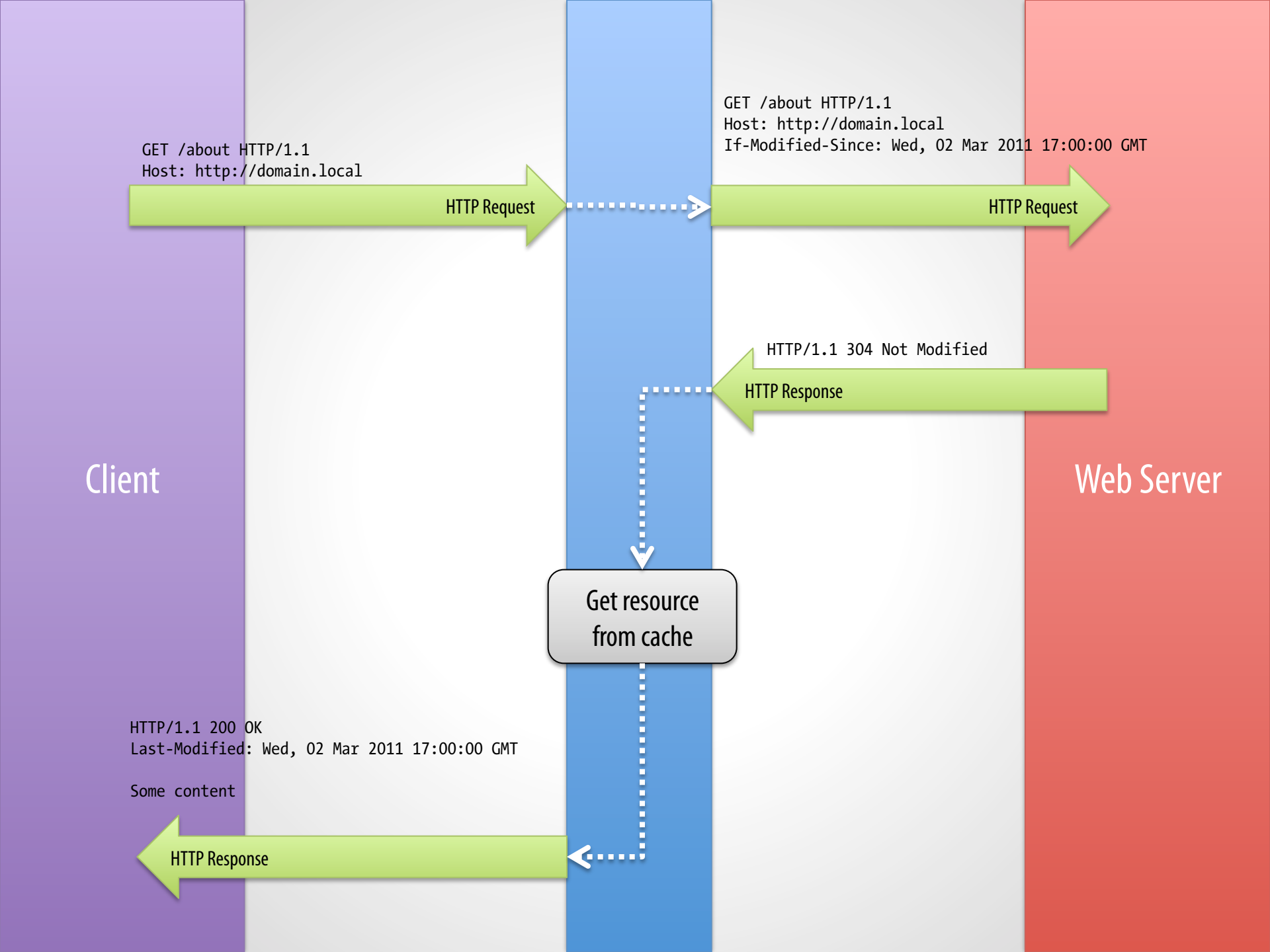
« store »

HTTP/1.1 200 OK
Last-Modified: Wed, 02 Mar 2011 17:00:00 GMT

Some content

HTTP Response





Client

GET /about HTTP/1.1
Host: http://domain.local

HTTP Request

GET /about HTTP/1.1
Host: http://domain.local
If-Modified-Since: Wed, 02 Mar 2011 17:00:00 GMT

HTTP Request

HTTP/1.1 200 OK
Last-Modified: Thu, 03 Mar 2011 10:30:00 GMT

Some new content

HTTP Response

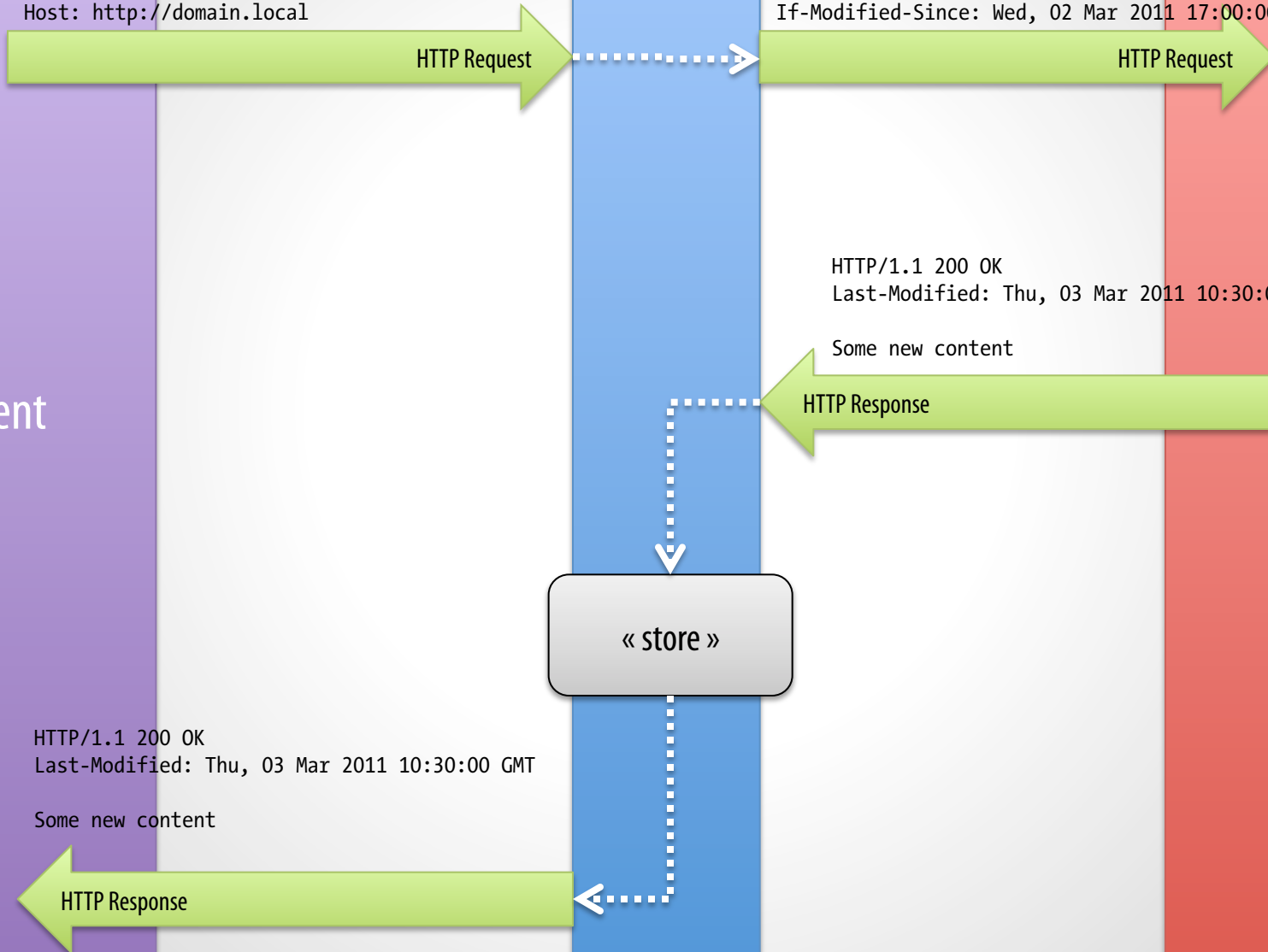
Web Server

« store »

HTTP/1.1 200 OK
Last-Modified: Thu, 03 Mar 2011 10:30:00 GMT

Some new content

HTTP Response



How to return 304 or 200?

```
$article = ArticleDAO::findOneById($id);

$response = new Response();
$response->setLastModified($article->getUpdatedAt());

if ($response->isNotModified($request)) {

    // send the 304 response immediately
    return $response;
}

return $this->render(
    'BlogBundle:Default:post.html.twig',
    array('article' => $article),
    $response
);
```

Expiration & Validation Together

Both « expiration and validation » models can live together but it's important to remember that **expiration wins over validation.**

```
// Set cache settings in one call
$response->setCache(array(
    'etag'           => $etag,
    'max_age'        => 10,
    'public'         => false,
));
```

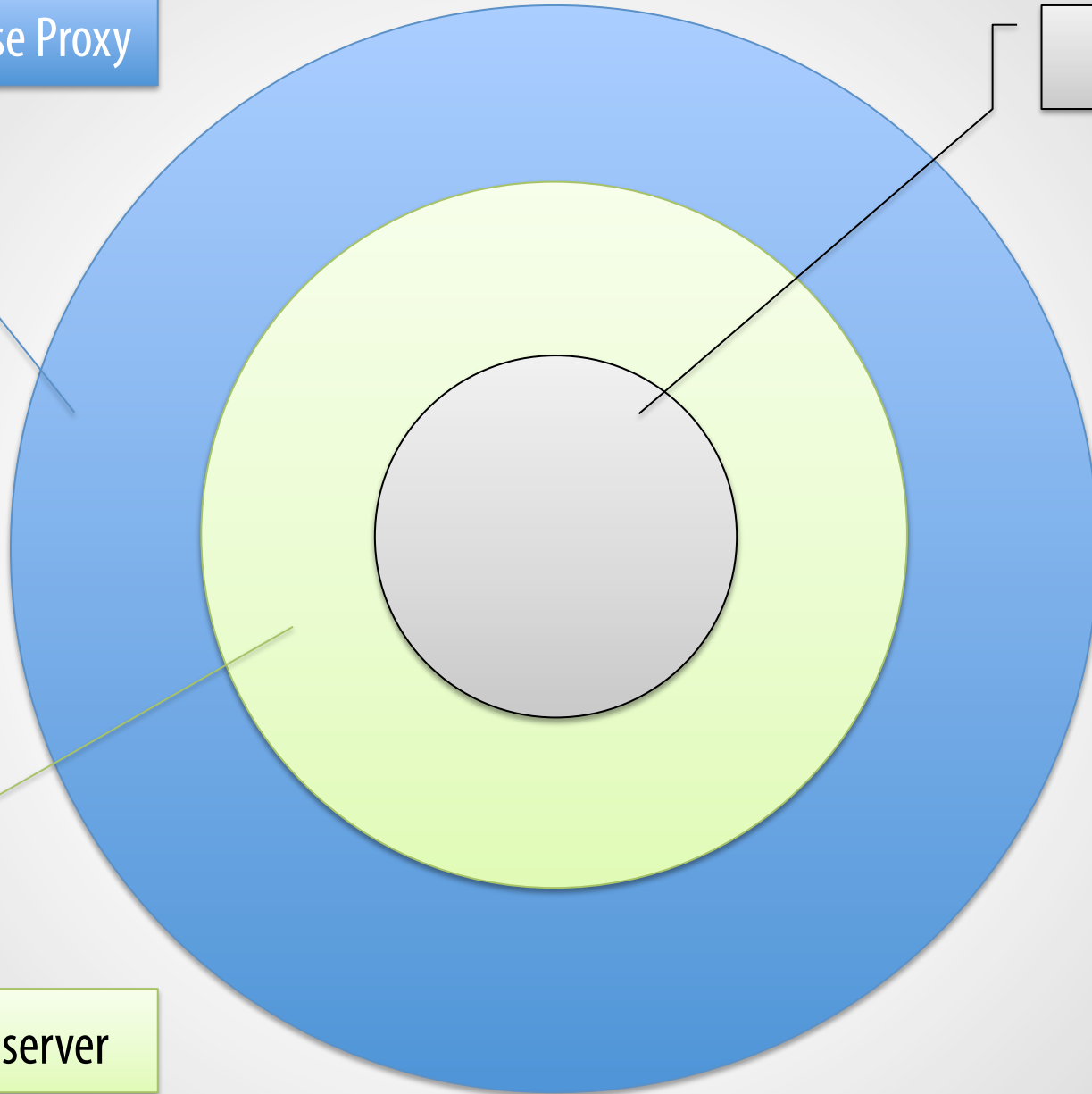
Reverse Proxy Caches

Reverse-Proxy cache sits
in front of the application
between the client and
the web server.

Reverse Proxy

AppKernel

Web server



Using a Reverse Proxy Cache



Documentation

Get up to speed

[See documentation](#)

Trac

Latest changes & updates

[Go to Trac](#)

Community support

Need help?

[Get support](#)

Varnish


Squid

PHP Reverse-Proxy

Enabling the Symfony2 Reverse-Proxy

```
# web/app.php
```

```
require_once __DIR__.'../../app/bootstrap.php.cache';  
require_once __DIR__.'../../app/AppKernel.php';  
require_once __DIR__.'../../app/AppCache.php';
```



```
use Symfony\Component\HttpFoundation\Request;
```

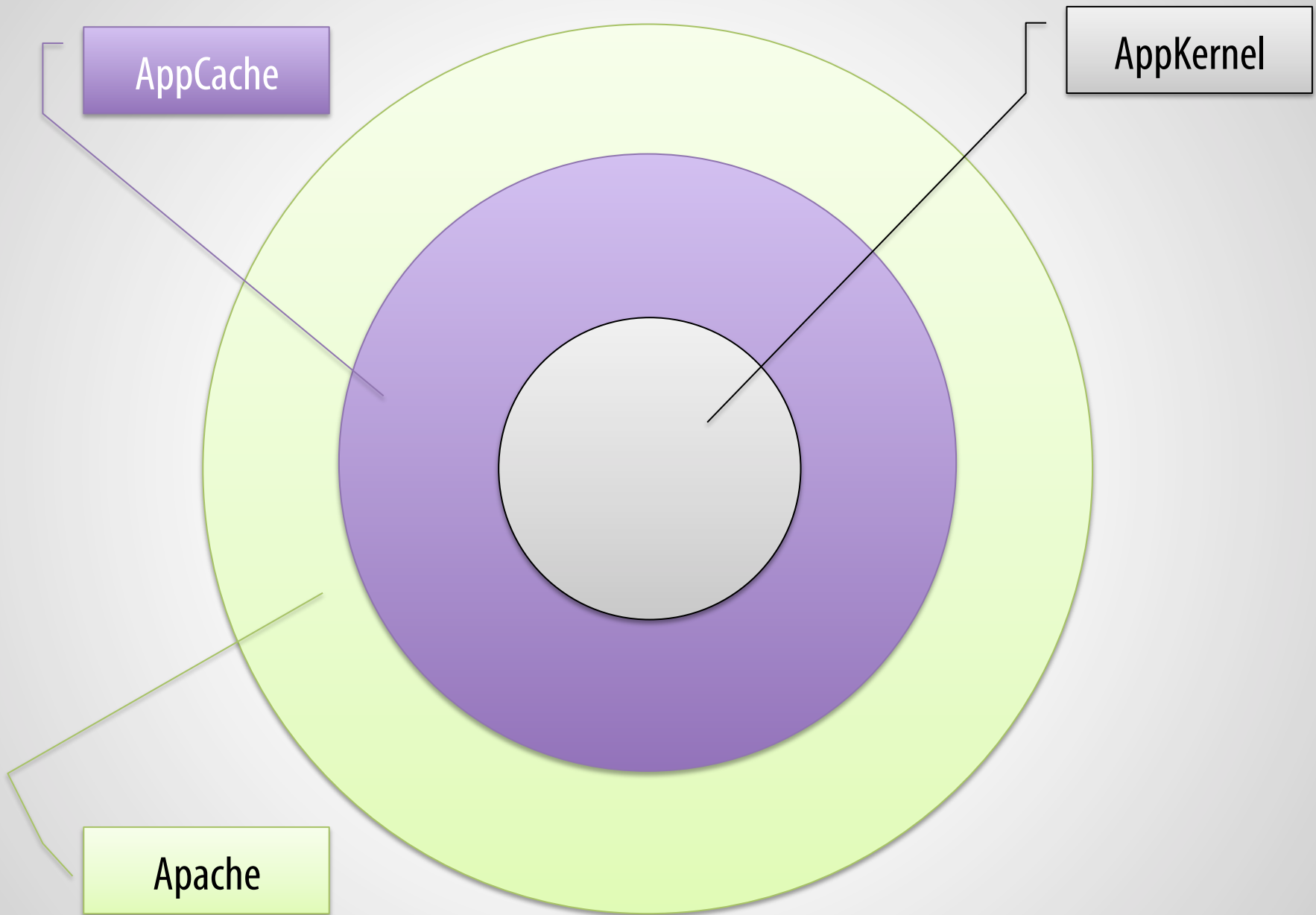
```
$request = Request::createFromGlobals();
```

```
$kernel = new AppKernel('prod', false);  
$kernel->loadClassCache();
```

```
$kernel = new AppCache($kernel);
```



```
$response = $kernel->handle($request);  
$response->send();
```



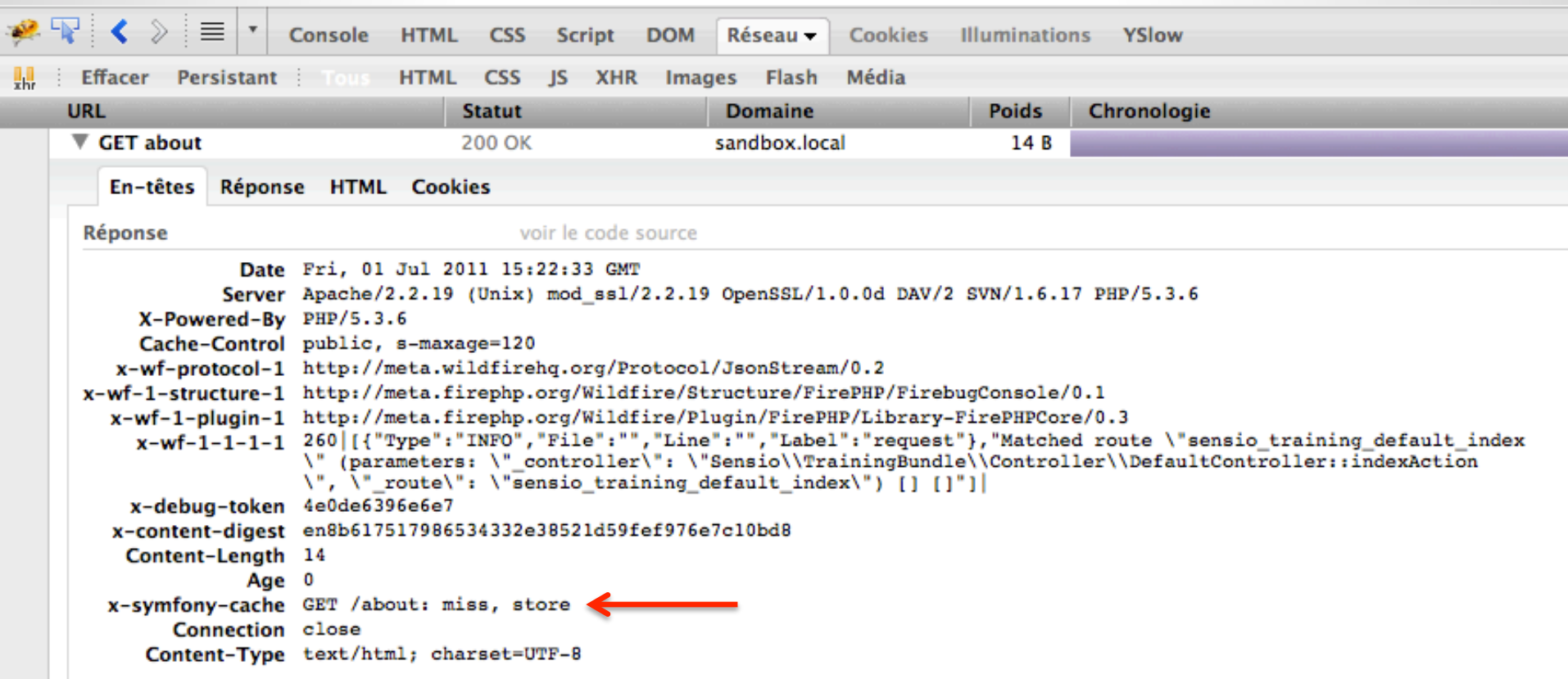
Caching a response on the server side

The generated **Response** must include a **s-maxage** **Cache-Control** directive and must be **public**.

```
/**
 * @Route("/about", name="about")
 * @Cache(smaxage=120)
 */
public function aboutAction()
{
    // ...
}
```

Cache Debugging : page is not cached yet

The `x-symfony-cache` debug header field indicates the page is not yet cached (`miss`), so the reverse proxy caches it (`store`).



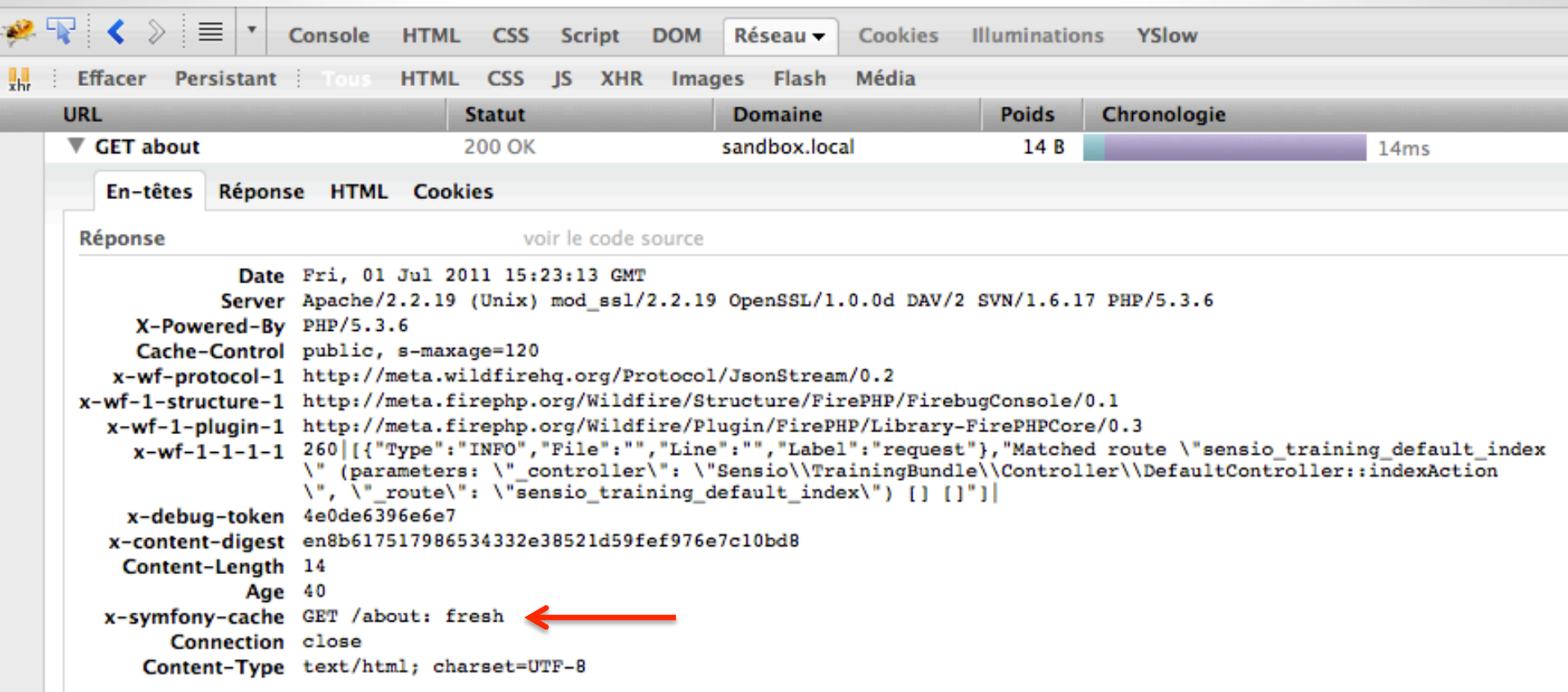
The screenshot shows a web browser's developer console with the 'Réseau' (Network) tab selected. A request to 'GET about' is shown with a status of '200 OK' and a size of '14 B'. The 'En-têtes' (Headers) sub-tab is active, displaying the response headers. A red arrow points to the 'x-symfony-cache' header, which has the value 'GET /about: miss, store'.

URL	Statut	Domaine	Poids	Chronologie
GET about	200 OK	sandbox.local	14 B	

En-têtes	Réponse	HTML	Cookies
Réponse voir le code source			
Date Fri, 01 Jul 2011 15:22:33 GMT			
Server Apache/2.2.19 (Unix) mod_ssl/2.2.19 OpenSSL/1.0.0d DAV/2 SVN/1.6.17 PHP/5.3.6			
X-Powered-By PHP/5.3.6			
Cache-Control public, s-maxage=120			
x-wf-protocol-1 http://meta.wildfirehq.org/Protocol/JsonStream/0.2			
x-wf-1-structure-1 http://meta.firephp.org/Wildfire/Structure/FirePHP/FirebugConsole/0.1			
x-wf-1-plugin-1 http://meta.firephp.org/Wildfire/Plugin/FirePHP/Library-FirePHPCore/0.3			
x-wf-1-1-1-1 260 [{"Type": "INFO", "File": "", "Line": "", "Label": "request"}, {"Matched route \"sensio_training_default_index\" (parameters: \"_controller\": \"Sensio\\TrainingBundle\\Controller\\DefaultController::indexAction\", \"_route\": \"sensio_training_default_index\") [] []}]			
x-debug-token 4e0de6396e6e7			
x-content-digest en8b617517986534332e38521d59fef976e7c10bd8			
Content-Length 14			
Age 0			
x-symfony-cache GET /about: miss, store ←			
Connection close			
Content-Type text/html; charset=UTF-8			

Cache Debugging : page is still valid

The **x-symfony-cache** debug header field indicates the page is still valid (**fresh**), so the reverse proxy gets it from its cache.



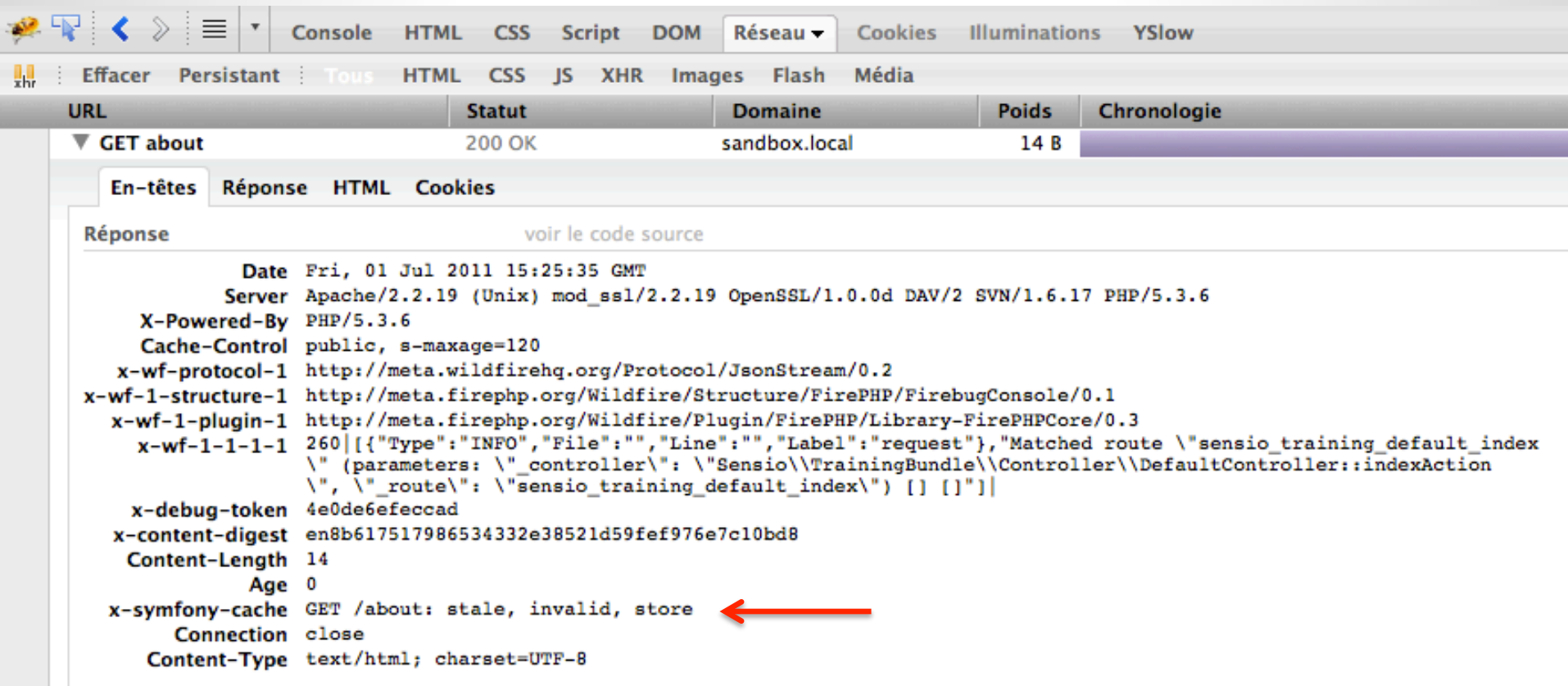
The screenshot shows the 'Réseau' (Network) tab in a web browser's developer tools. A request to 'GET about' is selected, showing a status of '200 OK' and a size of '14 B'. The 'En-têtes' (Headers) sub-tab is active, displaying the response headers. A red arrow points to the 'x-symfony-cache' header, which has the value 'GET /about: fresh'.

URL	Statut	Domaine	Poids	Chronologie
GET about	200 OK	sandbox.local	14 B	14ms

En-têtes	Réponse	HTML	Cookies
Réponse			
Date	Fri, 01 Jul 2011 15:23:13 GMT		
Server	Apache/2.2.19 (Unix) mod_ssl/2.2.19 OpenSSL/1.0.0d DAV/2 SVN/1.6.17 PHP/5.3.6		
X-Powered-By	PHP/5.3.6		
Cache-Control	public, s-maxage=120		
x-wf-protocol-1	http://meta.wildfirehq.org/Protocol/JsonStream/0.2		
x-wf-1-structure-1	http://meta.firephp.org/Wildfire/Structure/FirePHP/FirebugConsole/0.1		
x-wf-1-plugin-1	http://meta.firephp.org/Wildfire/Plugin/FirePHP/Library-FirePHPCore/0.3		
x-wf-1-1-1-1	260 {"Type":"INFO","File":"","Line":"","Label":"request"},"Matched route \"sensio_training_default_index\" (parameters: \"_controller\": \"Sensio\\TrainingBundle\\Controller\\DefaultController::indexAction\", \"_route\": \"sensio_training_default_index\") [] []"]		
x-debug-token	4e0de6396e6e7		
x-content-digest	en8b617517986534332e38521d59fef976e7c10bd8		
Content-Length	14		
Age	40		
x-symfony-cache	GET /about: fresh ←		
Connection	close		
Content-Type	text/html; charset=UTF-8		

Cache Debugging : page is stale

The `x-symfony-cache` debug header field indicates the page is no more fresh (`stale, invalid`), so the reverse proxy stores the new version in its cache (`store`).



The screenshot shows the 'Réseau' (Network) tab in a web browser's developer tools. The selected request is 'GET about' with a status of '200 OK' and a size of '14 B'. The 'En-têtes' (Headers) sub-tab is active, displaying the 'Réponse' (Response) headers. A red arrow points to the 'x-symfony-cache' header, which has the value 'GET /about: stale, invalid, store'.

URL	Statut	Domaine	Poids	Chronologie
GET about	200 OK	sandbox.local	14 B	

En-têtes	Réponse	HTML	Cookies
Réponse voir le code source			
Date	Fri, 01 Jul 2011 15:25:35 GMT		
Server	Apache/2.2.19 (Unix) mod_ssl/2.2.19 OpenSSL/1.0.0d DAV/2 SVN/1.6.17 PHP/5.3.6		
X-Powered-By	PHP/5.3.6		
Cache-Control	public, s-maxage=120		
x-wf-protocol-1	http://meta.wildfirehq.org/Protocol/JsonStream/0.2		
x-wf-1-structure-1	http://meta.firephp.org/Wildfire/Structure/FirePHP/FirebugConsole/0.1		
x-wf-1-plugin-1	http://meta.firephp.org/Wildfire/Plugin/FirePHP/Library-FirePHPCore/0.3		
x-wf-1-1-1-1	260 [{ "Type": "INFO", "File": "", "Line": "", "Label": "request", "Matched route \"sensio_training_default_index\" (parameters: \"_controller\": \"Sensio\\TrainingBundle\\Controller\\DefaultController::indexAction\", \"_route\": \"sensio_training_default_index\") [] [] }]		
x-debug-token	4e0de6efeccad		
x-content-digest	en8b617517986534332e38521d59fef976e7c10bd8		
Content-Length	14		
Age	0		
x-symfony-cache	GET /about: stale, invalid, store ←		
Connection	close		
Content-Type	text/html; charset=UTF-8		

Edge Side
Includes

<esi:include src="http://..." />

[What is Symfony?](#)[Get started](#)[Documentation](#)[Marketplace](#)[Community](#)[Services](#)[Blog](#)[About](#)[Home](#) » [Marketplace](#)

Marketplace

Under construction...

No ESI

NEWS FROM THE BLOG

A week of symfony #241 (8->14 August 2011)

August 15, 2011

A week of symfony #240 (1->7 August 2011)

August 09, 2011

symfony 1.4.13 released

August 05, 2011

[Visit Symfony's blog >](#)

IN THE NEWS



Symfony 2 trainings

Be the first
to be trained

[REGISTER TODAY](#)

UPCOMING TRAINING SESSIONS

Getting Started with Symfony2
Köln - 2011-08-25

Getting Started with Symfony2
New York City - 2011-09-12

Mastering Symfony2
New York City - 2011-09-14

[View all sessions >](#)

What is Symfony?

Get started

Documentation

Marketplace

Community

Services

Blog

About



[Home](#) » [Marketplace](#)

Marketplace

Under construction...

With ESI

NEWS FROM THE BLOG

A week of symfony #241 (8->14 August 2011)

August 15, 2011

A week of symfony #240 (1->7 August 2011)

August 09, 2011

symfony 1.4.13 released

August 05, 2011

[Visit Symfony's blog](#) >

IN THE NEWS



Symfony 2
trainings

Be the first
to be trained

REGISTER TODAY

```
<esi:include ... />
```

Browser

GET /foo HTTP/1.1
Host: foo.org

HTTP/1.1 200 OK

Lorem ipsum dolor sit amet,
Lorem ipsum dolor

Browser Cache

GET /foo HTTP/1.1
Host: foo.org

HTTP/1.1 200 OK

Lorem ipsum dolor sit amet,
Lorem ipsum dolor

Gateway Cache

GET /foo HTTP/1.1
Host: foo.org

HTTP/1.1 200 OK
C-C: s-maxage=10

Lorem ipsum dolor

<esi:include src="http.." />

GET /bar HTTP/1.1
Host: foo.org

HTTP/1.1 200 OK
C-C: s-maxage=5

Lorem ipsum dolor

GET /bar
C-C: s-maxage=5
Lorem

GET /foo
C-C: s-maxage=10
Lor <esi:include />

Your PHP application

Enabling Edge Side Includes

Edge Side Includes can be enabled from the main `config.yml` configuration file.

```
# app/config/config.yml
framework:
  # ...
  esi: ~
```

Rendering an ESI tag

Symfony provides a `render_esi()` Twig function that generates an Edge Side Include tag. It takes a controller reference and an array of variables as its arguments.

```
{{ render_esi(controller('AcmeBlogBundle:Blog:comments', { 'max': 10 } )) }}
```

You can also pass an absolute URI instead of a controller reference.

```
{{ render_esi(url('last_comments', { 'max': 10 } )) }}
```

Rendering an ESI Response

```
public function commentsAction($max)
{
    $comments = CommentDAO::getMostRecent($max);

    $response = $this->render(
        'AcmeBlogBundle:Blog:comments',
        array('comments' => $comments)
    );
    $response->setSharedMaxAge(600);

    return $response;
}
```

SensioLabs Training Department

Address

92-98 Boulevard Victor Hugo
92 115 Clichy Cedex
France

Phone

+33 140 998 205

Email

training@sensiolabs.com

training.sensiolabs.com

SensioLabs