



Caching mit Contao 4.8

für AnwenderInnen

Warum ich?

- Der erste Pull Request im Bezug auf Caching auf den Monat genau vor 3 Jahren. Damals Contao 4.3.

contao/core-bundle (pre Monorepo)

- 576
- 577
- 742
- 743
- 1263
- 1388 (leofeyer)
- 1478
- 1611
- 1612
- 1630 (aschempp)

friendsofsymfony/http-cache

- 373
- 419
- 422
- 423
- 437

friendsofsymfony/http-cache-bundle

- 363
- 435

contao/contao

- 117
- 137
- 147
- 148
- 173
- 351
- 365
- 389
- 469 (ausi)
- 482
- 483 (aschempp)
- 592 (aschempp)
- 611

toflar/psr6-symfony-http-cache-store

- 45 Commits seit 27.09.2017

symfony/symfony

- 22079 (ausi)
- 24075
- 25583
- 26532 (aschempp)
- 27309
- 32455

Für AnwenderInnen?

- **Nicht** für EndanwenderInnen, die den Inhalt bearbeiten.
- Sondern für **fortgeschrittene** Contao-Administratoren, die die Webseite betreuen.
- = Es wird technisch, aber ihr wollt ja auch etwas mehr wissen, als die EndanwenderInnen! :-)

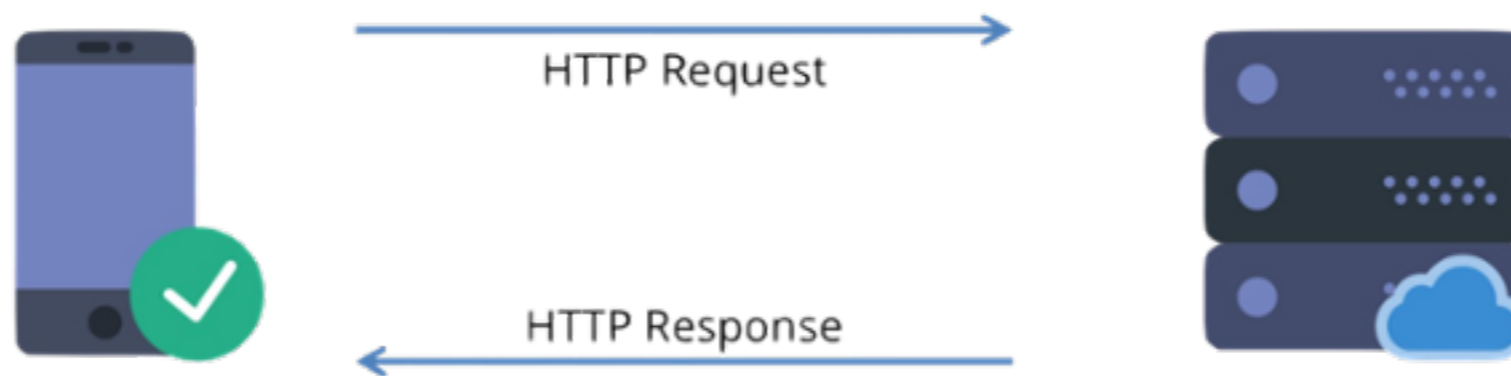
Es ändert sich doch eh wieder!

- **NEIN!**
- Okay, die einzige Konstante ist der Wandel aber...
 - ...zum ersten Mal **so richtig** happy mit dem was wir haben!
 - ...**keine** Änderungen für die 4.9 in Aussicht was das Handling nach aussen hin angeht.
 - ...**bestes** Caching-Framework aller CMSe dieser Welt! 😊
 - 🖋️ 📜



Back to the basics

Back to the basics



Request

```
GET /de/ HTTP/1.1
```

```
Cookie: PHPSESSID=.....
```

```
Host: contao.org
```

```
Connection: close
```

```
User-Agent: Mozilla/5.0 (...) Gecko/20100101 Firefox/69.0
```

```
Accept: text/html
```

Response

```
HTTP/2.0 200 OK
```

```
Date: Thu, 09 Oct 2019 12:42:00 GMT
```

```
Content-Length: 88
```

```
Content-Type: text/html
```

```
Cache-Control: no-cache
```

```
<html>
```

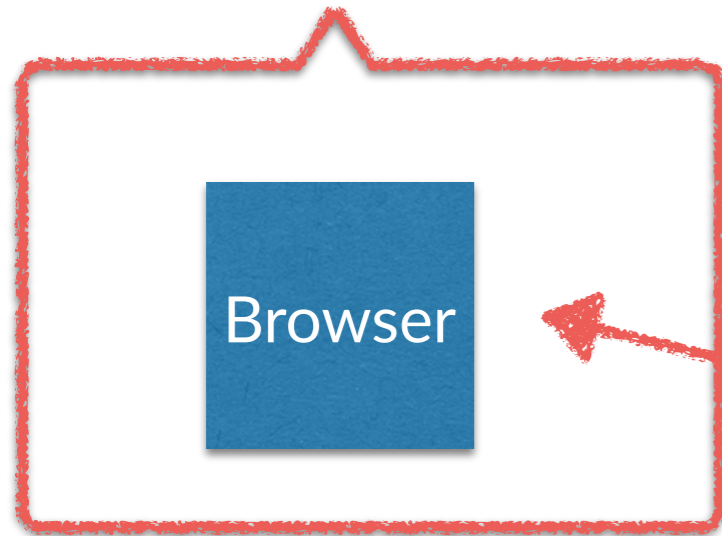
```
<body>
```

```
<h1>Hello, World!</h1>
```

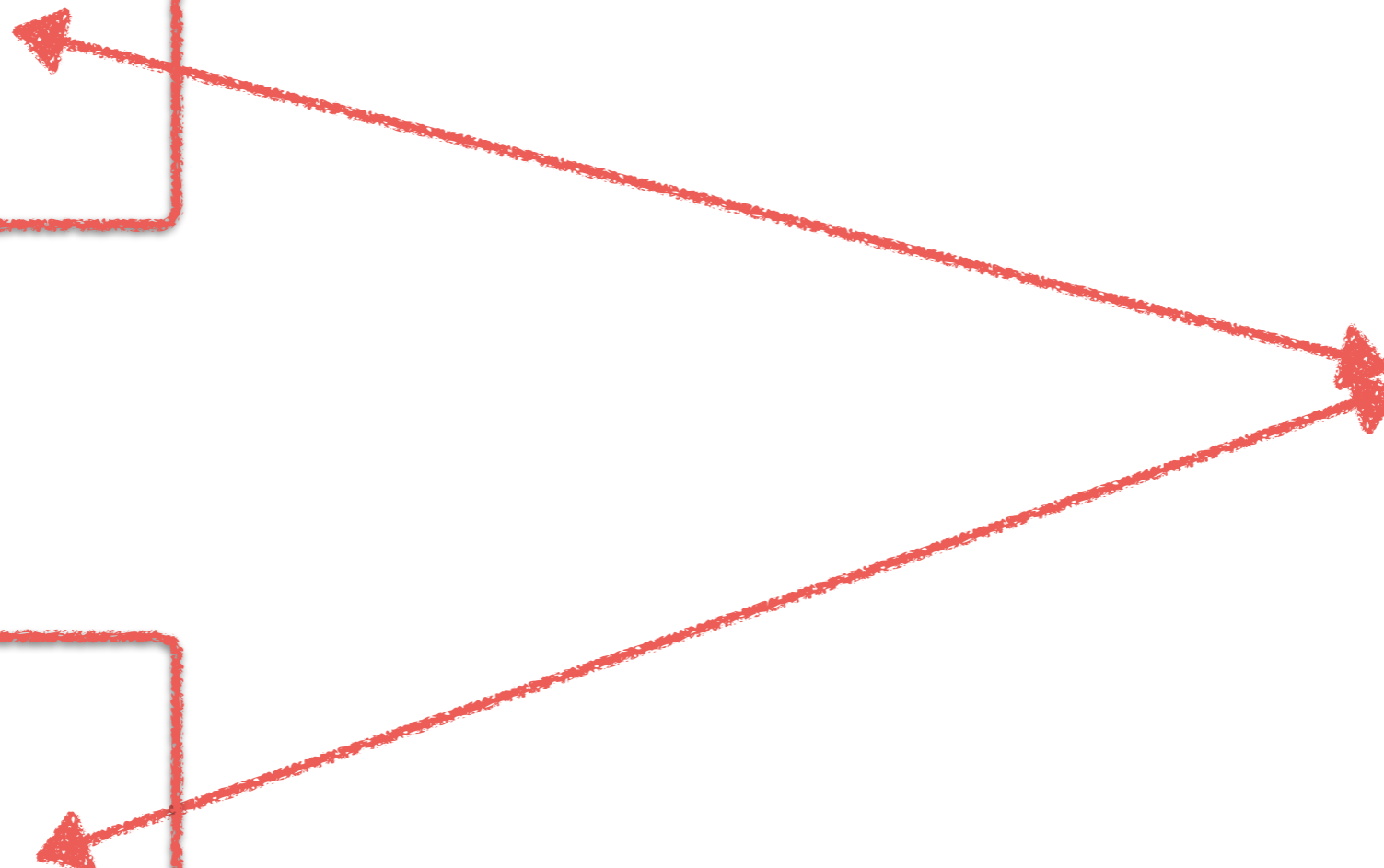
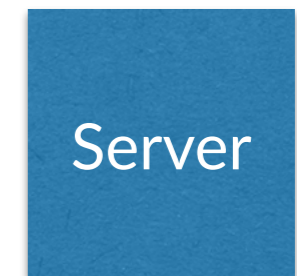
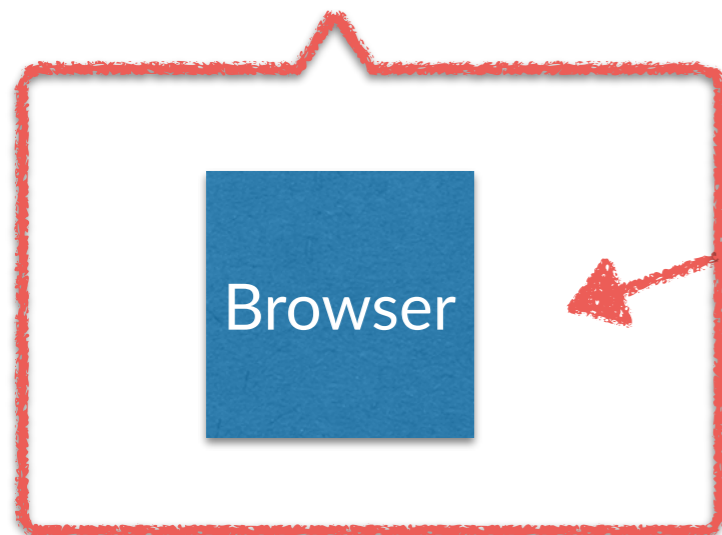
```
</body>
```

```
</html>
```

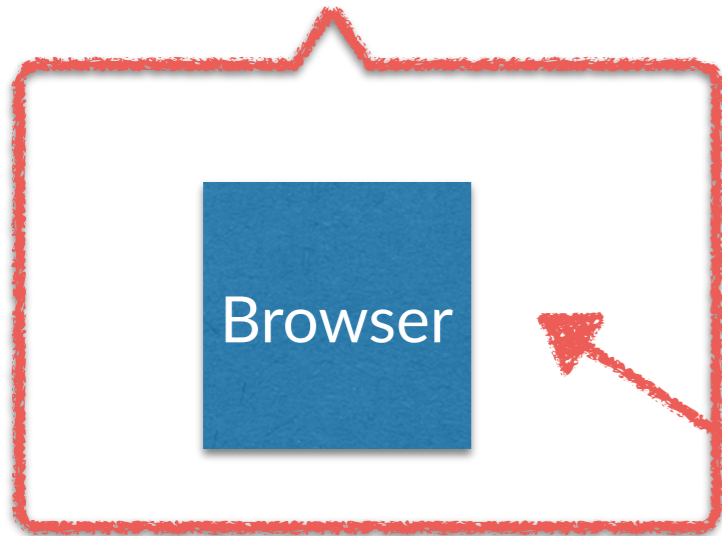

Private
Cache



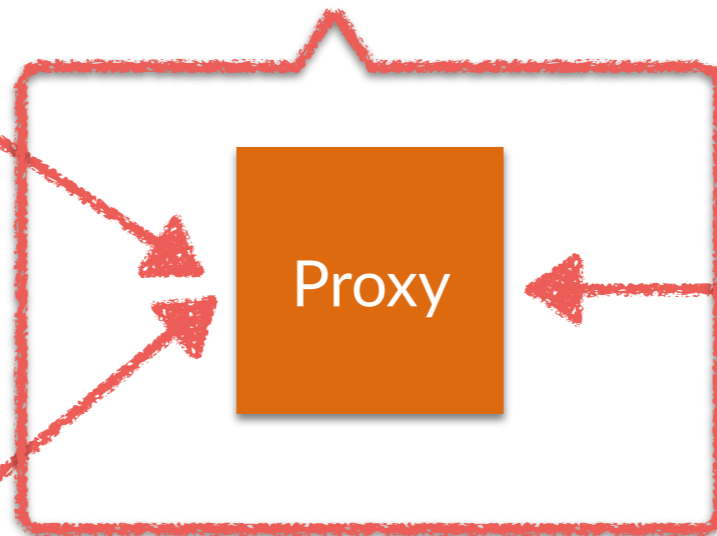
Private
Cache



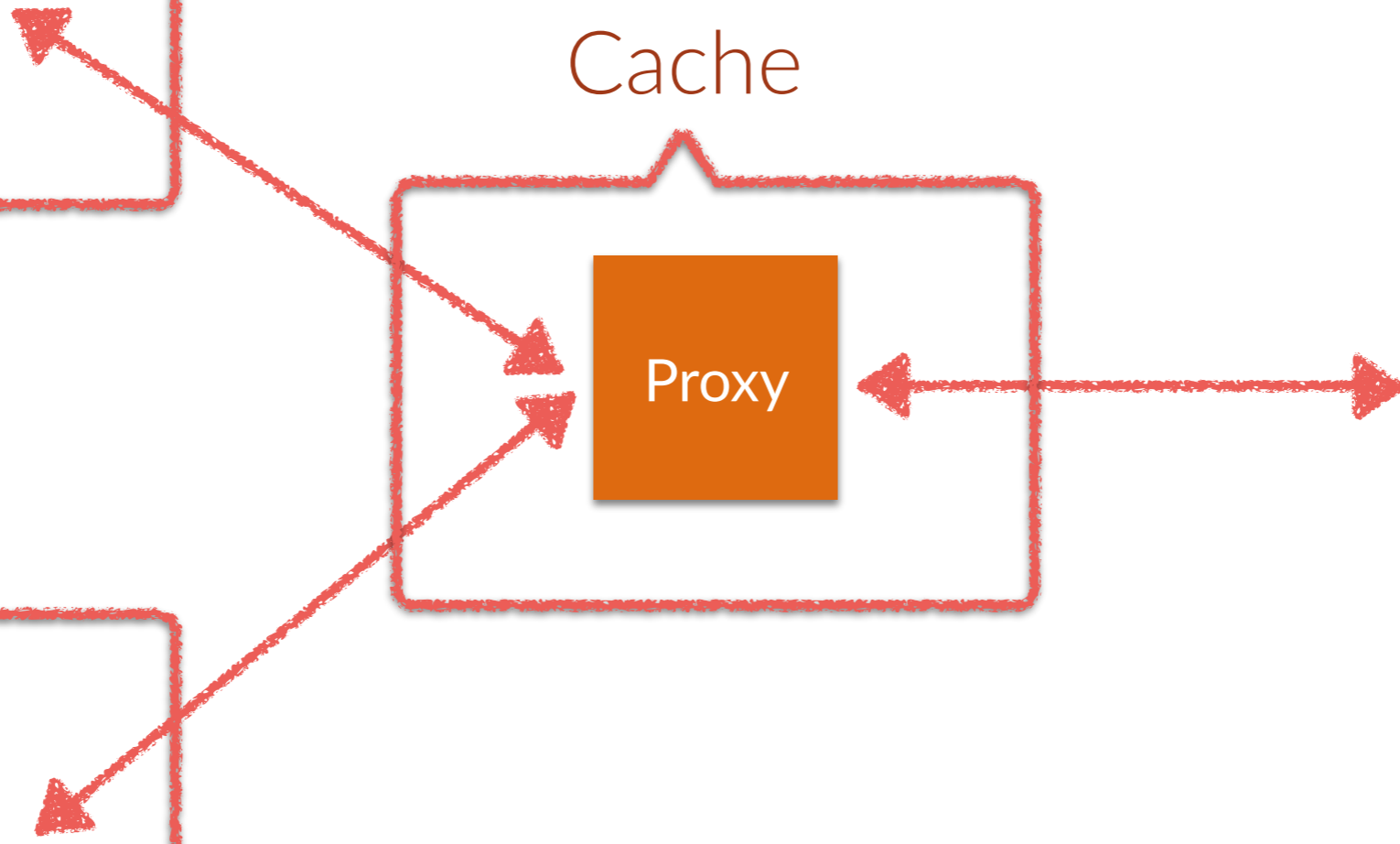
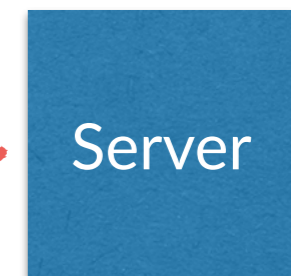
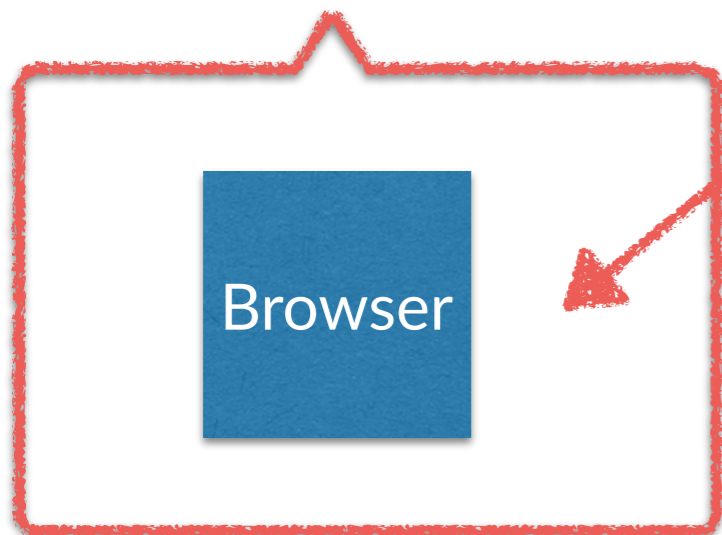
Private
Cache



Shared/Public
Cache



Private
Cache





Konfiguration

in Contao

▼ Cache settings

Set cache timeouts

Set cache timeout values for the page and its subpages.

Private cache timeout

30 minutes

The number of seconds after which the page should no longer be considered fresh by the browser.

Shared cache timeout

0 (do not cache)

The number of seconds after which the page should no longer be considered fresh by shared

Always load from shared cache

Always load this page from the shared cache, even if a member is logged in. Note that you can no

```
HTTP/2.0 200 OK
```

```
Date: Thu, 09 Oct 2019 12:42:00 GMT
```

```
Content-Length: 88
```

```
Content-Type: text/html
```

```
Cache-Control: private, max-age=1800
```

```
Vary: Cookie
```

```
<html>
```

```
<body>
```

```
<h1>Hello, World!</h1>
```

```
</body>
```

```
</html>
```

▼ Cache settings

Set cache timeouts

Set cache timeout values for the page and its subpages.

Private cache timeout

30 minutes

The number of seconds after which the page should no longer be considered fresh by the browser.

Shared cache timeout

60 minutes

The number of seconds after which the page should no longer be considered fresh by shared caches.

Always load from shared cache

Always load this page from the shared cache, even if a member is logged in. Note that you can no

HTTP/2.0 200 OK

Date: Thu, 09 Oct 2019 12:42:00 GMT

Content-Length: 88

Content-type: text/html

Cache-Control: public, s-maxage=3600, max-age=1800

Vary: Cookie

<html>

<body>

<h1>Hello, World!</h1>

</body>

</html>

Set cache timeouts

Set cache timeout values for the page and its subpages.

Private cache timeout

30 minutes ▼

The number of seconds after which the page should no longer be considered fresh by a private

Shared cache timeout

60 minutes ▼

The number of seconds after which the page should no longer be considered fresh by a shared

Always load from shared cache

Always load this page from the shared cache, even if a member is logged in. Note that you can no

```
HTTP/2.0 200 OK
```

```
Date: Thu, 09 Oct 2019 12:42:00 GMT
```

```
Content-Length: 88
```

```
Content-type: text/html
```

```
Cache-Control: public, s-maxage=3600, max-age=1800
```

```
Vary: Cookie
```

```
<html>
```

```
<body>
```

```
<h1>Hello, World!</h1>
```

```
</body>
```

```
</html>
```



That's it!

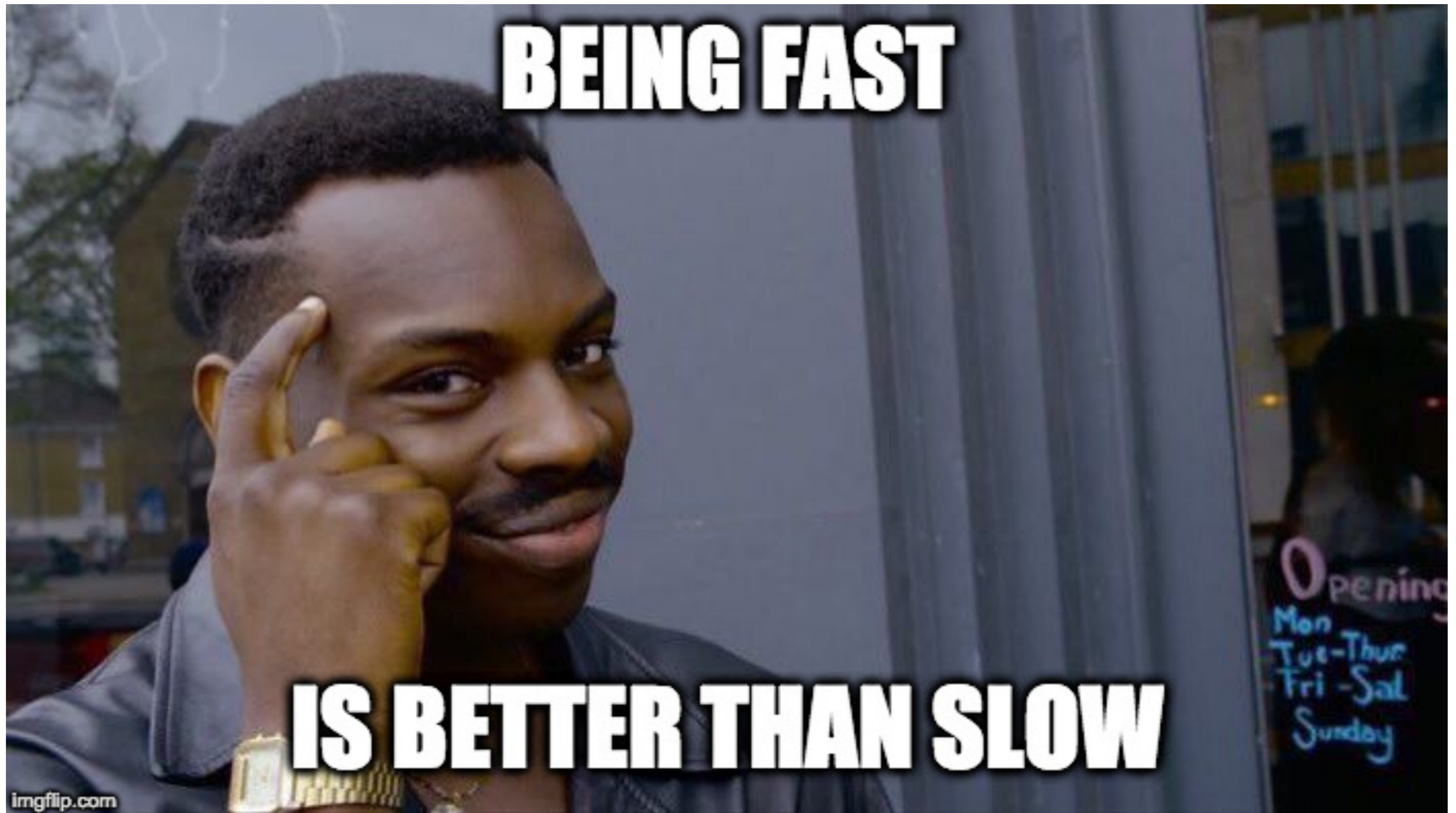
für die EndanwenderInnen

Warum überhaupt ein Cache?

- Leistung (CPU, RAM) kostet immer weniger.
- PHP OPCCache wird immer besser.
- «Mein Contao ist doch auch ohne Cache schnell!»

Performance!

- Auch wenn der Server schnell ist:
 - Contao Boot-Prozess
 - Routing (zu welcher Seite gehört die URL)
 - Datenbankabfragen für Seiten, Artikel, Module, Inhaltselemente
 - Template Rendering
- Performance is about conversion rates and SEO (PageSpeed, Lighthouse, ...)
- Performance is about UX! **Performance is about the users!**



imgflip.com



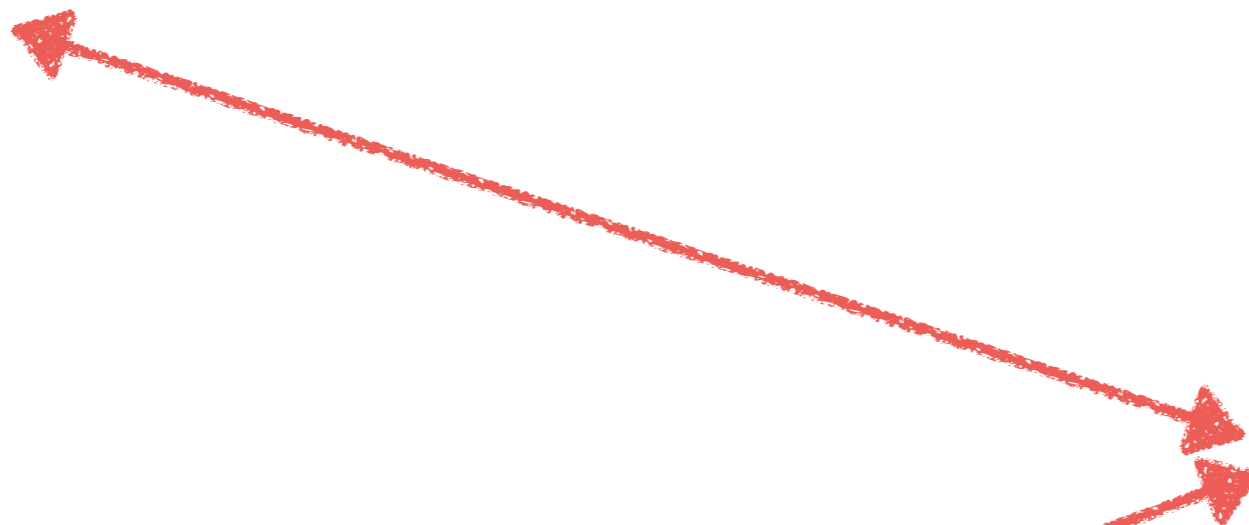
Wozu der Umbau?

Der eingebaute HTML-Cache in der 3.5 war doch okay?

Standards sind toll!

- Masseinheiten
- DIN
- ISO
- RFC

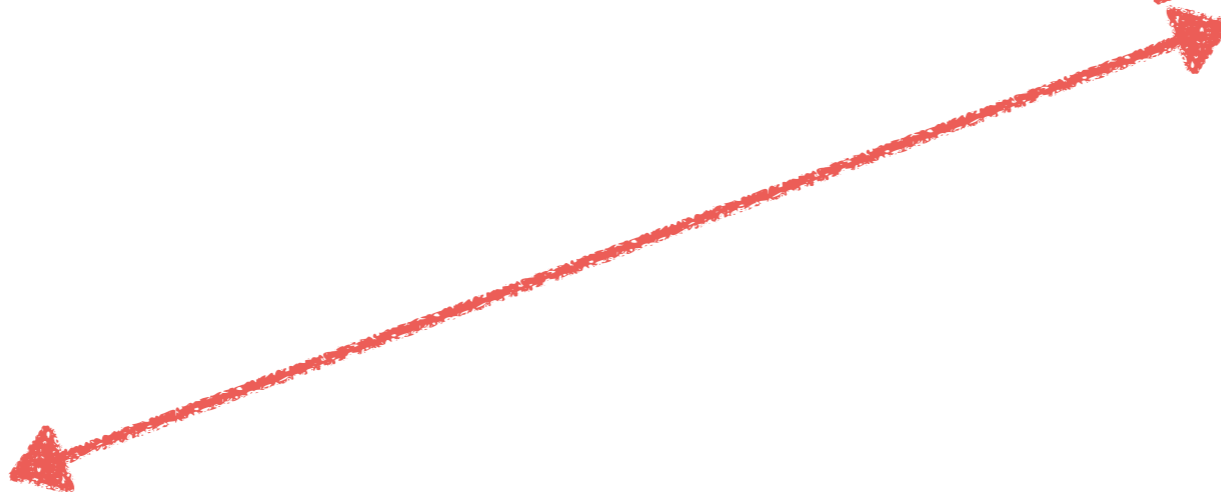
Browser



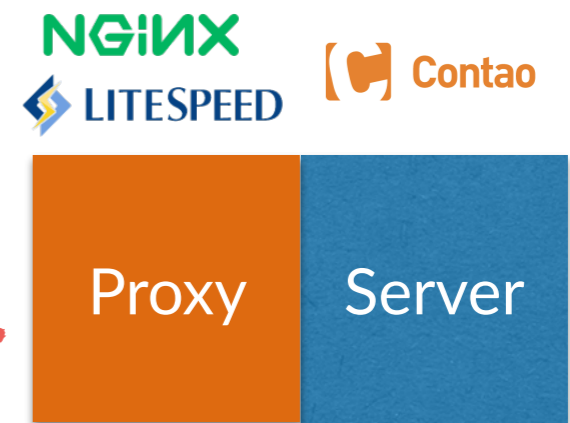
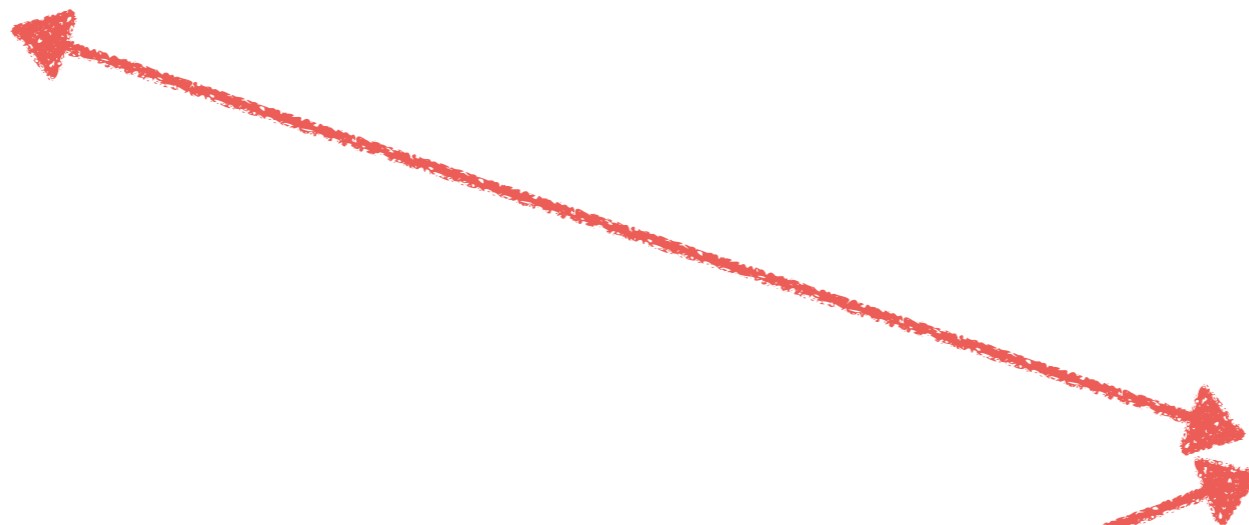
 Symfony  Contao



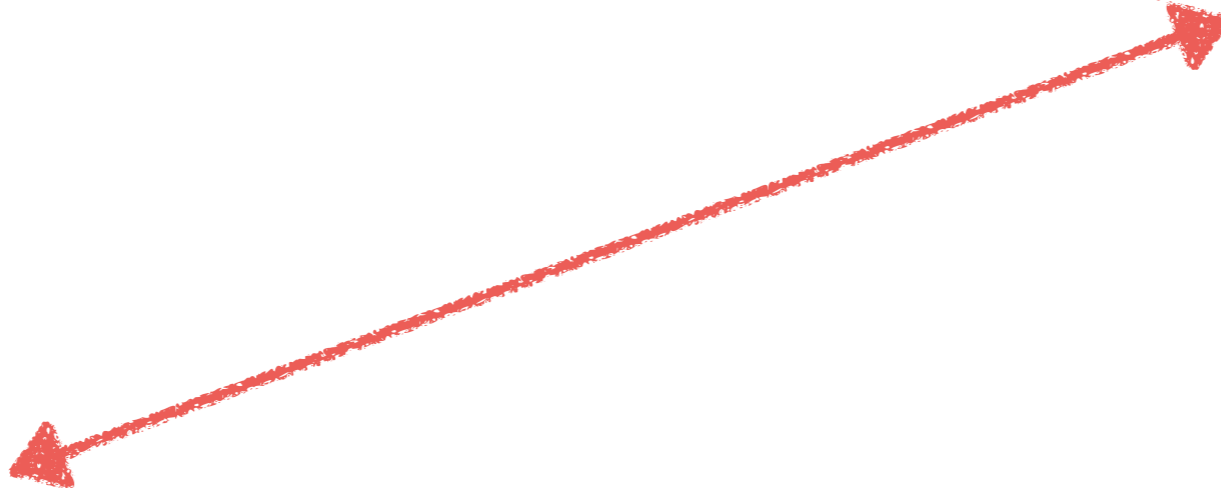
Browser



Browser



Browser



Browser

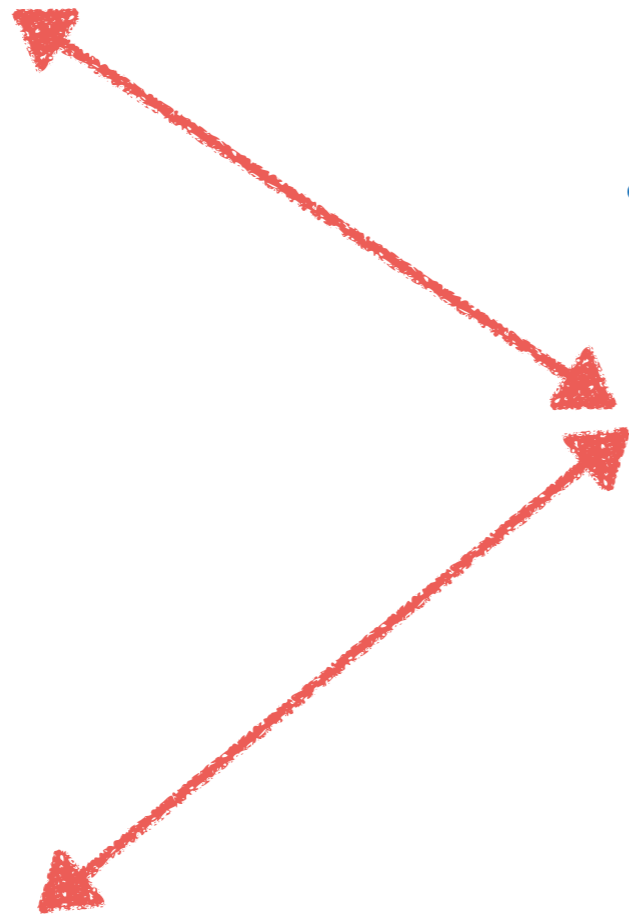
Browser

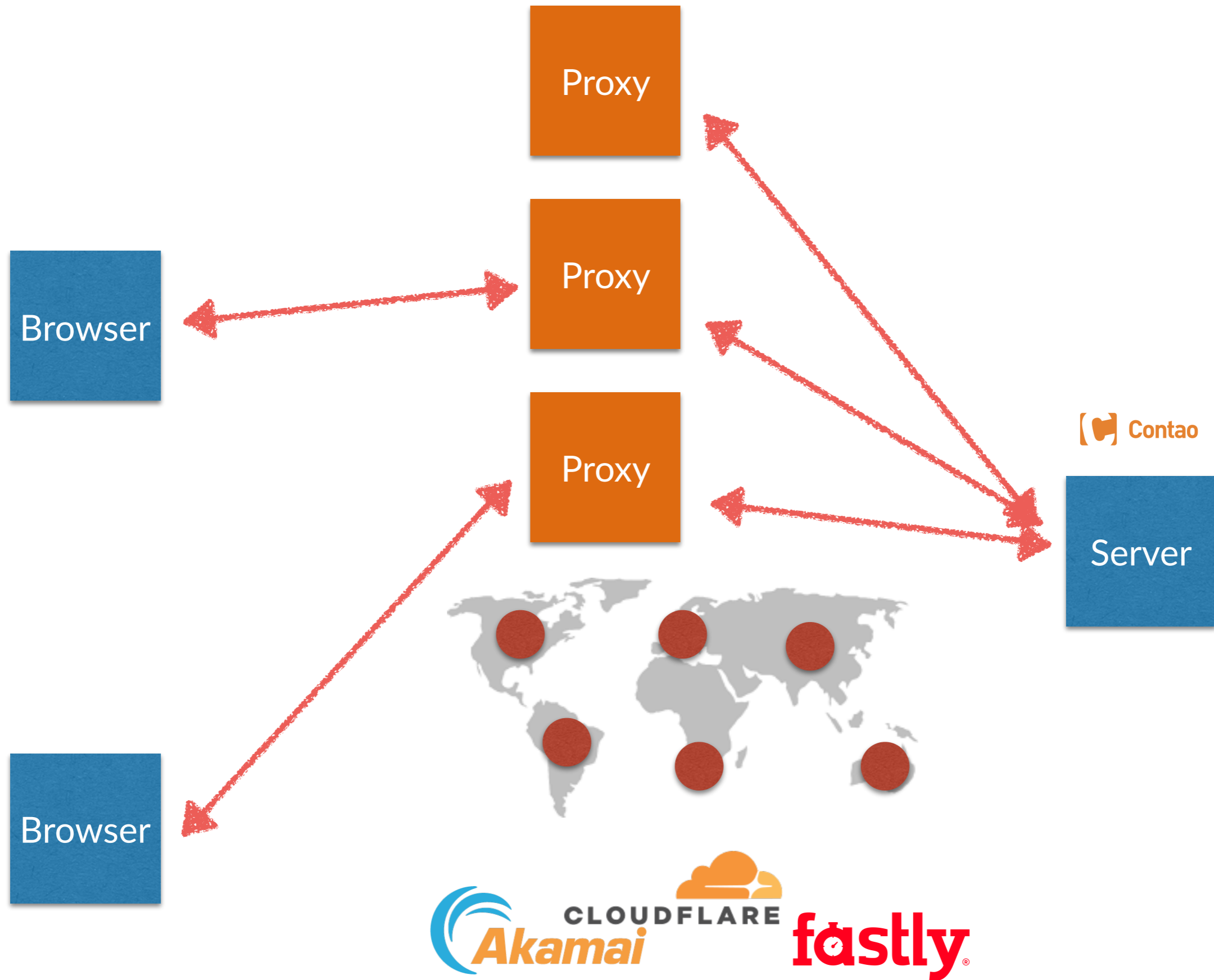


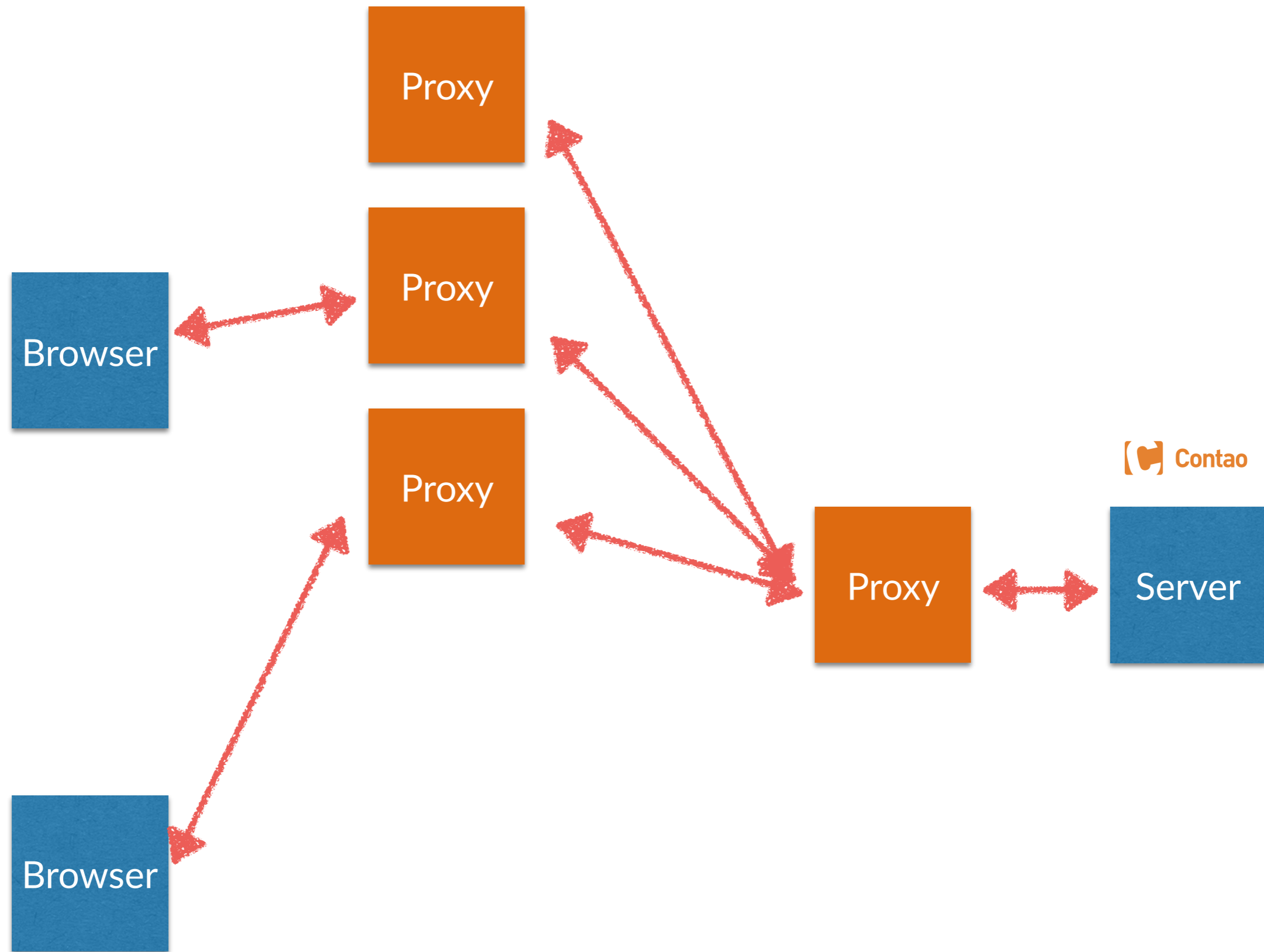
Proxy

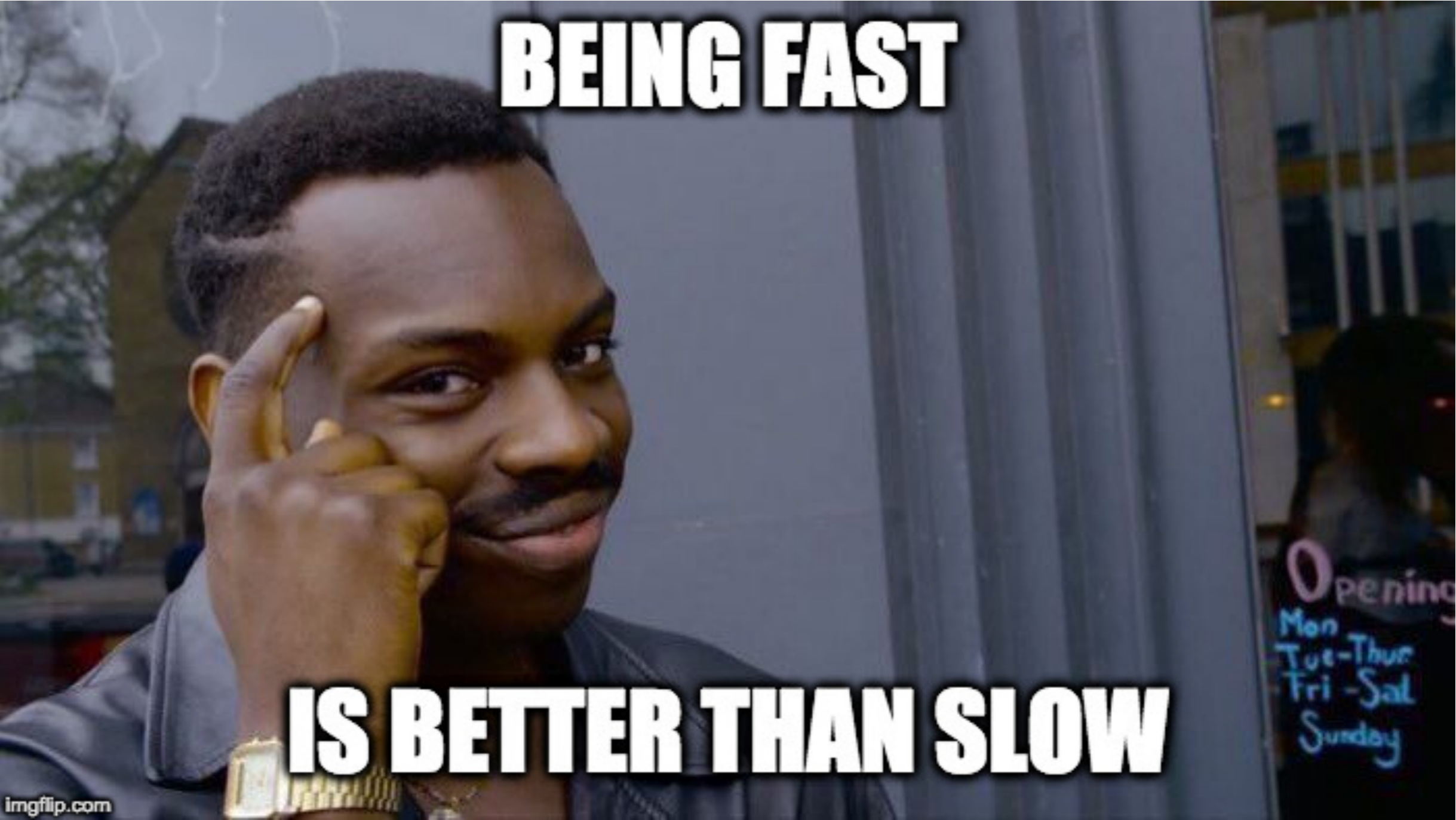


Server









imgflip.com



Der Shared Cache

ist worüber wir heute sprechen!

Inspektor Konsole Debugger Stilbearbeitung Laufzeitanalyse Speicher Netzwerkanalyse Web-Speicher Barrierefreiheit

Adressen durchsuchen

| Sta... | Me... | Host | Datei | Ursprung | Typ | Übertragen | Grö... | 0 ms | 10,24 s | 20,48 s | Kopfzeilen | Cookies | Parameter | Antwort | Zeit | Aufrufliste | Sicherheit | |
|--------|-------|------|-------|----------|------|------------|---------|--------|---------|---------|-------------------------------------|---------|-----------|---------|------|-------------|------------|--|
| 200 | GET | | /de/ | document | html | 11,68 KB | 63,8... | 256 ms | | | Angefragte Adresse: https://.../de/ | | | | | | | |

Anfragemethode: GET
Externe Adresse: 46...:443
Status-Code: 200 OK
Version: HTTP/2.0

Bearbeiten und erneut senden

Kopfzeilen durchsuchen

Antwortkopfzeilen (61/ B) Kopfzeilen (unformati

- age: 32
- cache-control: max-age=1800, private
- contao-cache: fresh
- content-length: 11347
- content-type: text/html; charset=utf-8
- date: Wed, 31 Jul 2019 22:50:16 GMT

Nicht cachebar

```
HTTP/2.0 200 OK
Date: Thu, 09 Oct 2019 12:42:00 GMT
Content-Length: 88
Content-type: text/html
Contao-Cache: miss
Age: 0
```

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```

Wird neu gecached

```
HTTP/2.0 200 OK  
Date: Thu, 09 Oct 2019 12:42:00 GMT  
Content-Length: 88  
Content-type: text/html  
Content-Cache: miss/store  
Age: 0
```

```
<html>  
<body>  
<h1>Hello, World!</h1>  
</body>  
</html>
```

Aus dem Cache

```
HTTP/2.0 200 OK  
Date: Thu, 09 Oct 2019 12:42:00 GMT  
Content-Length: 88  
Content-Type: text/html  
Contao-Cache: fresh  
Age: 42
```

```
<html>  
<body>  
<h1>Hello, World!</h1>  
</body>  
</html>
```

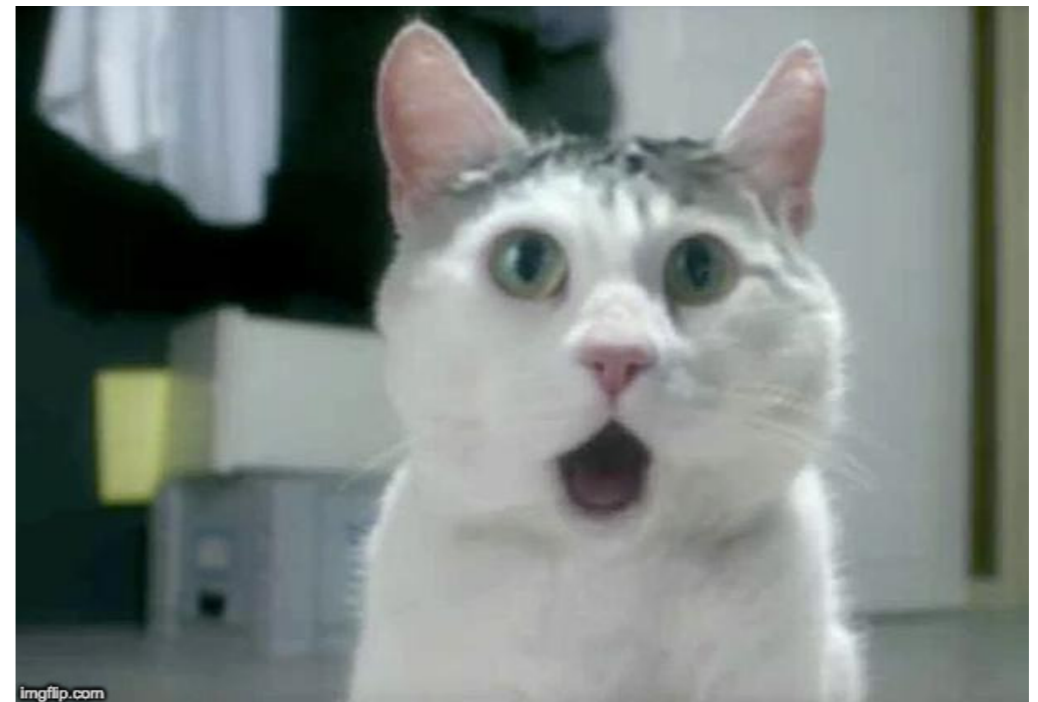


Problematik #1

Wie erkennen wir ob ein Benutzer eingeloggt ist?

Problemstellung

- Contao muss dem Cache mitteilen können, dass er <https://www.domain.de/login.html> anders cachen soll, je nach dem ob ich eingeloggt bin oder nicht. Ansonsten würde immer das Login-Formular angezeigt, statt «eingeloggt als k.jones».



Der Vary-Header

- Teilt dem Cache mit, unterschiedliche Varianten pro URI abzulegen.
- Man merke sich: «Hallo Cache, bitte vary-iere» 🙋🏻 😊

Der Vary-Header

```
GET /hello-world  
Accept: text/html
```

```
HTTP/2.0 200 OK  
Cache-Control: public, s-maxage=300  
Vary: Accept
```

```
<html>  
<body>  
<h1>Hello, World!</h1>  
</body>  
</html>
```

Ich darf diesen Eintrag für 300 Sekunden cachen, aber nur dann ausliefern, wenn der Request den Header «Accept» mit dem exakten (!) Inhalt «text/html» enthält!

Der Vary-Header

```
GET /hello-world  
Accept: application/json
```

```
HTTP/2.0 200 OK  
Cache-Control: public, s-maxage=300  
Vary: Accept
```

```
{  
  "body": "Hello World!"  
}
```

Ich darf diesen Eintrag für 300 Sekunden cachen, aber nur dann ausliefern, wenn der Request den Header «Accept» mit dem exakten (!) Inhalt «application/json» enthält!

Nicht eingeloggt

```
GET /hello-world  
Accept: text/html
```

```
HTTP/2.0 200 OK  
Cache-Control: public, s-maxage=300  
Vary: Cookie
```

Eingeloggt

```
GET /hello-world  
Accept: text/html  
Cookie: PHPSESSID=b7a861e4ae0664b3a82ec2fbf239e5ff
```

```
HTTP/2.0 200 OK  
Cache-Control: private, no-store  
Vary: Cookie
```

Nicht eingeloggt

Set cache timeouts

Set cache timeout values for the page and its subpages.

Private cache timeout

30 minutes ▼

The number of seconds after which the page should no longer be considered fresh by a private

Shared cache timeout

60 minutes ▼

The number of seconds after which the page should no longer be considered fresh by a shared

Always load from shared cache

Always load this page from the shared cache, even if a member is logged in. Note that you can no

HTTP/2.0 200 OK

Date: Thu, 09 Oct 2019 12:42:00 GMT

Content-Length: 88

Content-Type: text/html

Cache-Control: public, s-maxage=3600, max-age=1800

~~Vary: Cookie~~

<html>

<body>

<h1>Hello, World!</h1>

</body>

</html>





Problematik #2

Frontend-Vorschau

Die preview.php

- web/index.php
- web/preview.php
- Frontend-Vorschau:
<https://www.domain.de/preview.php/profil.html>



Problematik #3

Cookies

Problematik

- Vary bezieht sich immer auf den Header und dessen gesamten Inhalt.
- Sowas wie «Vary: Cookie=PHPSESSID» existiert nicht.
 - (es gab mal einen Key-Header Draft der das ändern sollte aber dieser hat es leider nie über den Draft-Status hinaus geschafft...)
- Ergo: Wir können nicht auf bestimmte Cookies vary-en. Der Cache ist somit nicht in der Lage **irrelevante** von **relevanten** Cookies zu unterscheiden.
Caching wird somit deaktiviert, sobald mindestens ein Cookie vorhanden ist!
 - irrelevant = hat keinen Einfluss auf den Cache und soll somit eigentlich nicht berücksichtigt werden.
 - relevant = hat Einfluss auf den Cache und soll berücksichtigt werden.

Irrelevante Cookies?

- GDPR-Cookiebar, die am HTML nichts ändert
- Google Analytics
- Matomo (Piwik)
- Facebook Pixel
- Cloudflare
- uvm.

Lösung

- Blacklist, Contao kennt eine interne Liste an irrelevanten Cookies und löscht diese bevor der Cache gefragt wird. (siehe «StripCookiesSubscriber»)
 - '(.*)?modal(.*)?'
 - '_ga'
 - '_pk_id'
 - '_fbp'
- Whitelist, kann per Umgebungsvariable konfiguriert werden und übersteuert die mitgelieferte Blacklist.

- Cache-Speicher
- ▼ Cookies
- Indexed DB
- Local Storage
- Session Storage

Einträge durchsuchen

| Name | Domain | Pfad | Läuft ab am | Zuletzt zugegriffen | Wert | table.headers.cookies.isHttpOnly | sameSite |
|----------|--------|------|--------------|-------------------------------|------|----------------------------------|----------|
| myModal1 | | /de/ | Sitzungsende | Wed, 31 Jul 2019 23:47:11 GMT | 0 | false | Unset |

Informiert die Entwickler

- Cookies werden zu 90%* falsch eingesetzt!
- Wenn der Server die Information nicht verwertet, braucht es kein Cookie! Teilt ihm keine Informationen mit, die er nicht gebrauchen kann!
- JavaScript kann sessionStorage, localStorage (Vorsicht: blocking API, also synchron) oder IndexedDB nutzen!
- Meinen Blogpost lesen: <https://contao.org/de/news/die-jagd-auf-ueberfluessige-cookies.html>

* Yanick'sche Statistik

Relevante Cookies?

- PHPSESSID (PHP Session)
 - Potenziell persönliche Daten (Session !== eingeloggt als User)
- CSRF-Schutz
- A/B-Testing
- Google Analytics Daten werden vom Server ausgewertet

Whitelist

- Entweder als echte Umgebungsvariable
- Oder .env-Datei im Wurzelverzeichnis

```
COOKIE_WHITELIST=PHPSESSID,csrf_https-contao_csrf_token
```

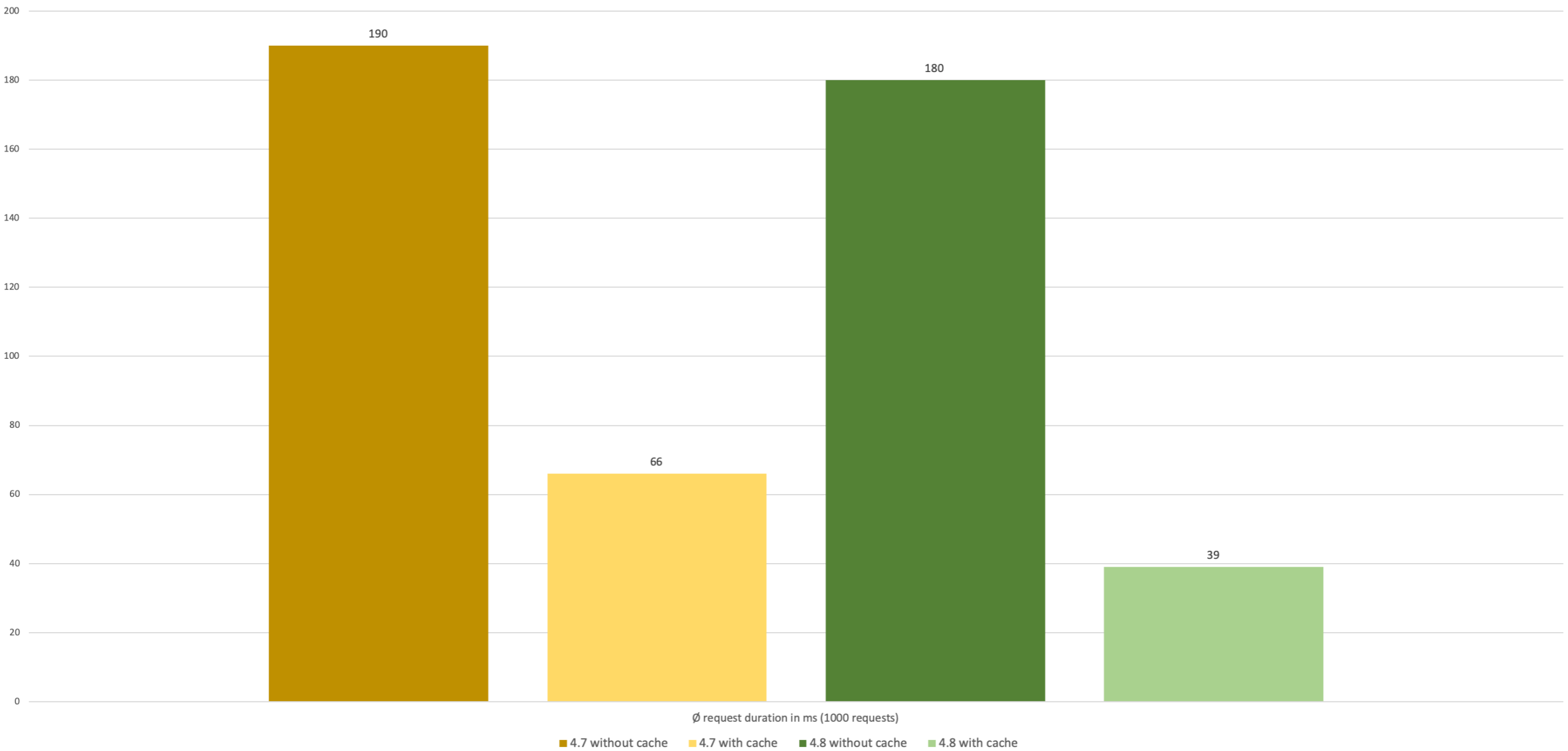


Bonus 1

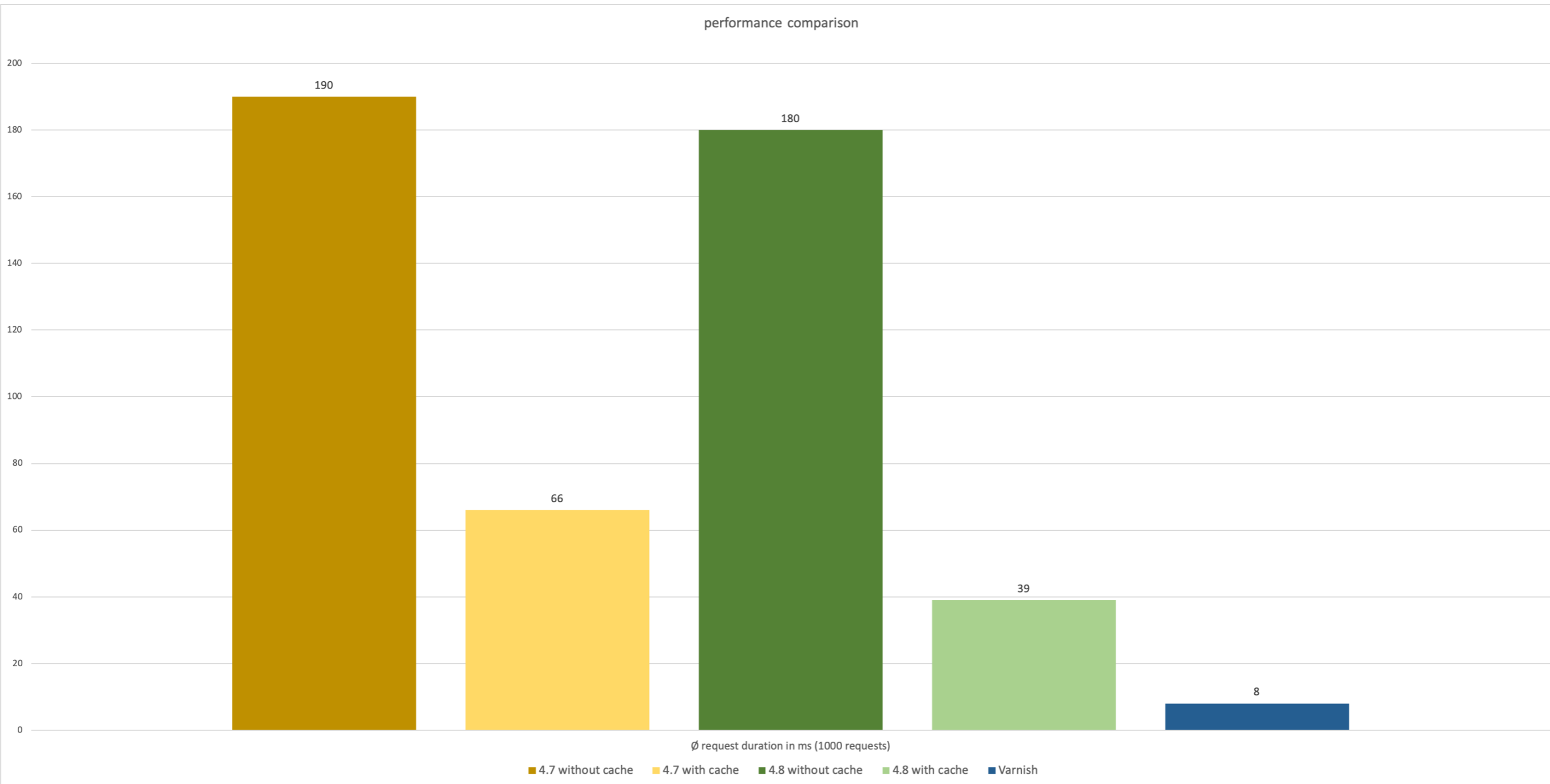
Performance-Vergleiche

Eingebauter Cache

performance comparison



Varnish





Bonus 2

Das beste Caching-Framework!

Unbekannte Clients /
Zugriff nicht möglich

Browser

Browser

Bekannte Server /
Zugriff möglich

 Symfony  Contao

Proxy

Server

Unbekannte Clients /
Zugriff nicht möglich

Browser

Browser

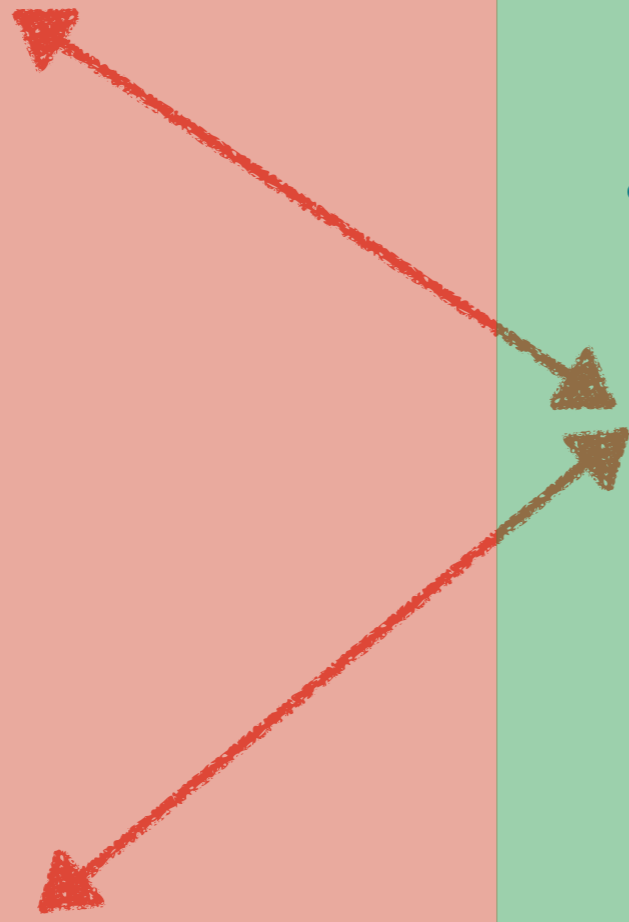
Bekannte Server /
Zugriff möglich



Proxy



Server



Cache Tagging

```
HTTP/2.0 200 OK
Date: Thu, 09 Oct 2019 12:42:00 GMT
Content-Length: 88
Content-Type: text/html
Cache-Control: public, s-maxage=1800
X-Cache-Tags: contao.db.tl_article.489,contao.db.tl_content.2180,contao.db.tl_content.
2182,contao.db.tl_content.2184,contao.db.tl_content.2186,contao.db.tl_content.
2185,contao.db.tl_content.2183,contao.db.tl_content.2181,contao.db.tl_module.
489,contao.db.tl_module.,contao.db.tl_article.386,contao.db.tl_content.
793,contao.db.tl_content.779,contao.db.tl_content.781,contao.db.tl_content.
783,contao.db.tl_content.782,contao.db.tl_content.784,contao.db.tl_content.
780,contao.db.tl_content.785,contao.db.tl_content.786,contao.db.tl_content.
794,contao.db.tl_content.791,contao.db.tl_content.787,contao.db.tl_content.
792,contao.db.tl_module.386,contao.db.tl_article.387,contao.db.tl_content.
795,contao.db.tl_content.797,contao.db.tl_content.801,contao.db.tl_content.
802,contao.db.tl_module.2,contao.db.tl_content.798,contao.db.tl_content.
799,contao.db.tl_content.803,contao.db.tl_content.800,contao.db.tl_content.
796,contao.db.tl_module.387,contao.db.tl_page.22
```

```
<html>
<body>
<h1>Hello, World!</h1>
</body>
</html>
```


Cache Tag Purging



```
PURGE / HTTP/1.1  
Host: localhost  
X-Cache-Tags: contao.db.tl_content.42
```

Contao Integration

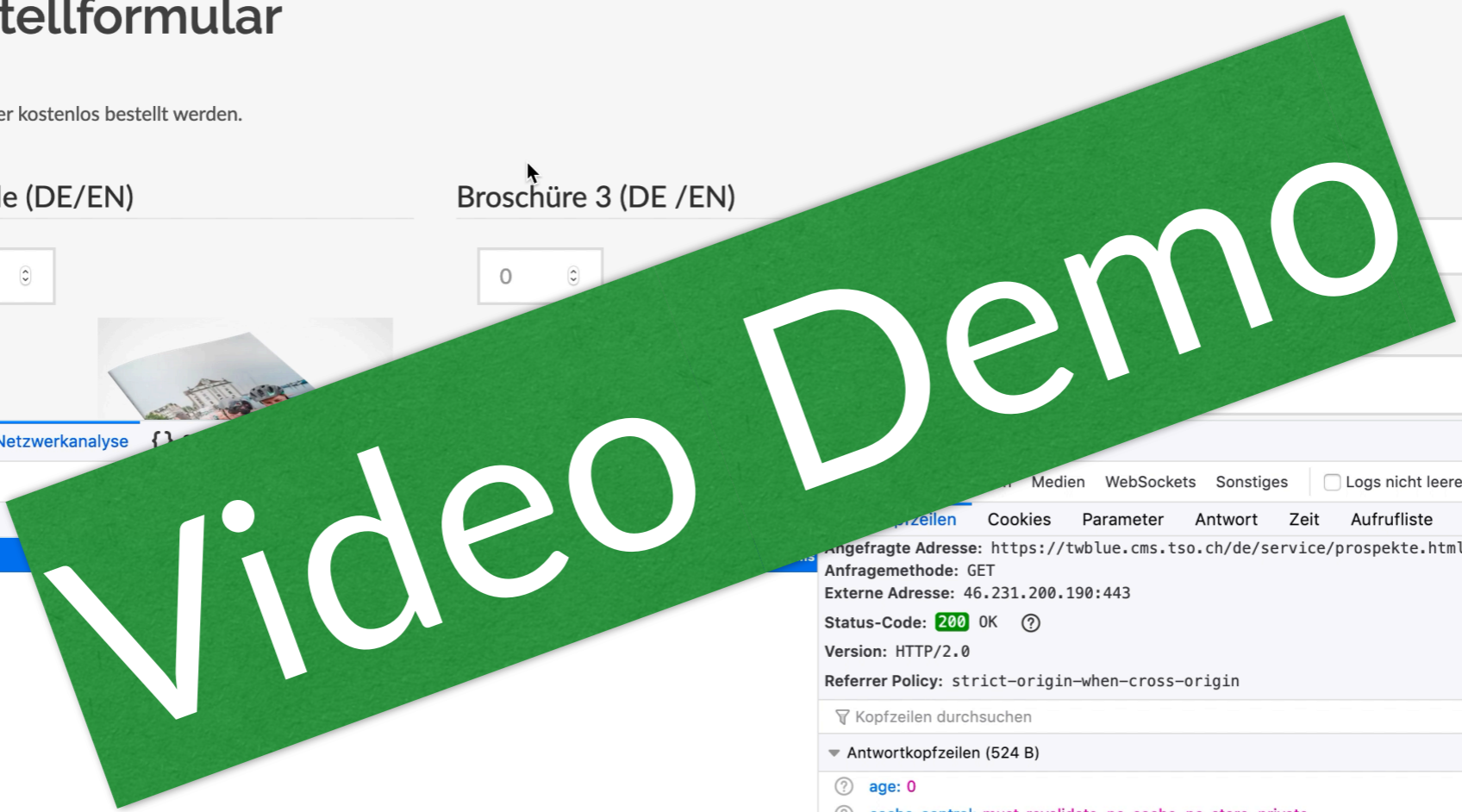
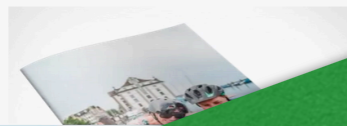
- Sendet beim Editieren automatisch Cache-Tag-Invalidation-Requests nach dem Schema
 - contao.db.<Tabelle>
 - contao.db.<Tabelle>.<ID>
- Erweiterbar durch den «oninvalidate_cache_tags_callback»
 - Beispielsweise invalidiert das «comments-bundle» beim Editieren eines Kommentars die Tags der Quelle.
- Tagging im Frontend ist Aufgabe des Entwicklers

Bestellformular

Können hier kostenlos bestellt werden.

Example (DE/EN)

Broschüre 3 (DE /EN)



Inspector Konsole Debugger Netzwerkanalyse

Adressen durchsuchen

| Status | Metho... | Host | Datei |
|--------|----------|-------------------|----------------|
| 200 | GET | twblue.cms.tso.ch | prospekte.html |

Medien WebSockets Sonstiges Logs nicht leeren Cache deaktivieren Keine Drosselung HAR

Anfragen Cookies Parameter Antwort Zeit Aufrufliste Sicherheit

Angefragte Adresse: https://twblue.cms.tso.ch/de/service/prospekte.html
Anfragemethode: GET
Externe Adresse: 46.231.200.190:443
Status-Code: 200 OK
Version: HTTP/2.0
Referrer Policy: strict-origin-when-cross-origin

Kopfzeilen durchsuchen

Antwortkopfzeilen (524 B) Kopfzeilen (unformatiert)

- age: 0
- cache-control: must-revalidate, no-cache, no-store, private
- contao-cache: miss
- content-encoding: gzip
- content-length: 7324
- content-type: text/html; charset=utf-8
- date: Mon, 07 Oct 2019 18:33:16 GMT
- expires: -1

Eine Anfrage 43,62 KB / 7,66 KB übertragen Beendet: 2,80 s DOMContentLoaded: 2,20 s load: 2,57 s

FOSHttpCache(Bundle)

- Symfony HttpCache mit toflar/psr6-symfony-http-cache-store
- Varnish
- erweiterbar, PR's offen für
 - OpenLiteSpeed
 - Fastly



Zusammenfassung

- Contao 4.8 enthält - abgesehen von «der Checkbox»™ - die gleichen Cache-Einstellungsmöglichkeiten wie Contao 4.4, aber...
 - ...wir sind 100% standardkonform!
 - ...wir sind schneller out-of-the-box!
 - ...wir bieten ein Meer an Möglichkeiten das Setup noch schneller zu machen!



?

Kontakt

- Yanick Witschi / terminal42 gmbh
- Contao Core Entwickler
- API's, Performance, Caching
- Tennis, Basketball, Faustball
- Scotch



@toflar

