

Globetrotter.de
Ausrüstung



High Performance E-Commerce

ElitePartner: Mobile mit AngularJS und REST

XING AG [DE] <https://www.xing.com/events/elitepartner-mobile-mit-angularjs-und-rest-1335333>

XING Startseite Jobs Projekte Gruppen Events Unternehmen

Erweiterte Suche

Andere zu XING einladen Hilfe & Kontakt Ausloggen

Überblick Gästeliste (38)

ElitePartner: Mobile mit AngularJS und REST

10.12.2013 Hamburg, DE, EliteMedianet GmbH

Kartenansicht

Überblick: Wir würden Euch gerne in gemütlicher vorweihnachtlicher Runde unsere neue mobile Webseite unter dem Motto "Tech Stack - pants down" vorstellen.

Für meinen Kalender exportieren

Gästeliste (38)

Marcel Sauer
EliteMedianet GmbH (TOMORROW FOCUS AG)

Oliver Ochs
IT-Beratung

Christian Hartmann

Oliver Zeigermann
<http://zeigermann.eu>

Mehr

Das Event hat schon stattgefunden.

Die Teilnahme an diesem Event ist begrenzt. Daher werden nur Personen, die mit "Ja" geantwortet haben, als verbindliche Zusage auf der Gästeliste berücksichtigt.

ElitePartner: Mobile mit AngularJS und REST

Beginn: Di, 10.12.2013, 16:30 (CET)

Moin Moin,

wir würden Euch gerne in gemütlicher vorweihnachtlicher Runde unsere neue mobile Webseite unter dem Motto "Tech Stack - pants down" vorstellen.

ElitePartner hat im vergangenen Dezember den ersten Schritt gemacht und sich der Hamburger Entwickler-Community gestellt. Wir finden die Idee großartig und ziehen nach!



Leider konnte mein Frontend-Kollege Christian Haller bei unserem Event nicht dabei sein, da er „verhindert“ war :)

AGENDA

- Softwarearchitektur
- Tech Stack
- Testing
- Deployment
- Testimonials
- Q&A

The image shows the exterior of a Globetrotter store building. The building has a modern, red-colored facade made of large rectangular panels. There are several large, rectangular windows arranged in a grid-like pattern. The windows show the interior of the store, which is brightly lit and filled with various items, including clothing and equipment. The sky above the building is a pale blue with some light clouds. The overall appearance is clean and professional.

Globetrotter
Ausrüstung

Globetrotter
Ausrüstung

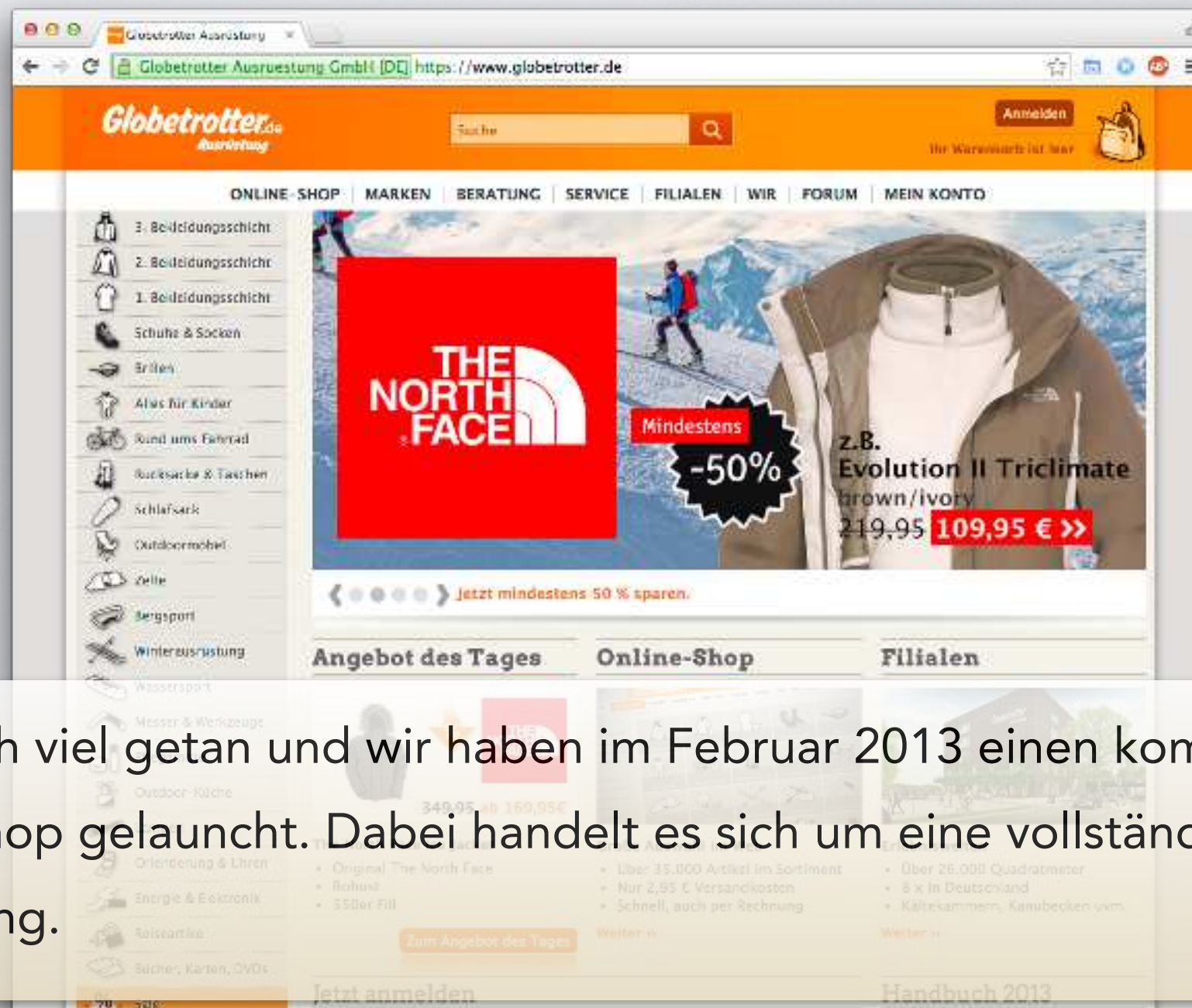
Globetrotter ist für seine großartigen Filialen bekannt, wie z.B. den hier abgebildeten Hamburger Store.



Aber auch online mischen wir schon seit den Anfängen des Internets mit - so hatten wir 1993 bereits eine BTX-Seite. Die Domain [globetrotter.de](http://www.globetrotter.de) ist älter als [google.de](http://www.google.de)!

STATUS QUO

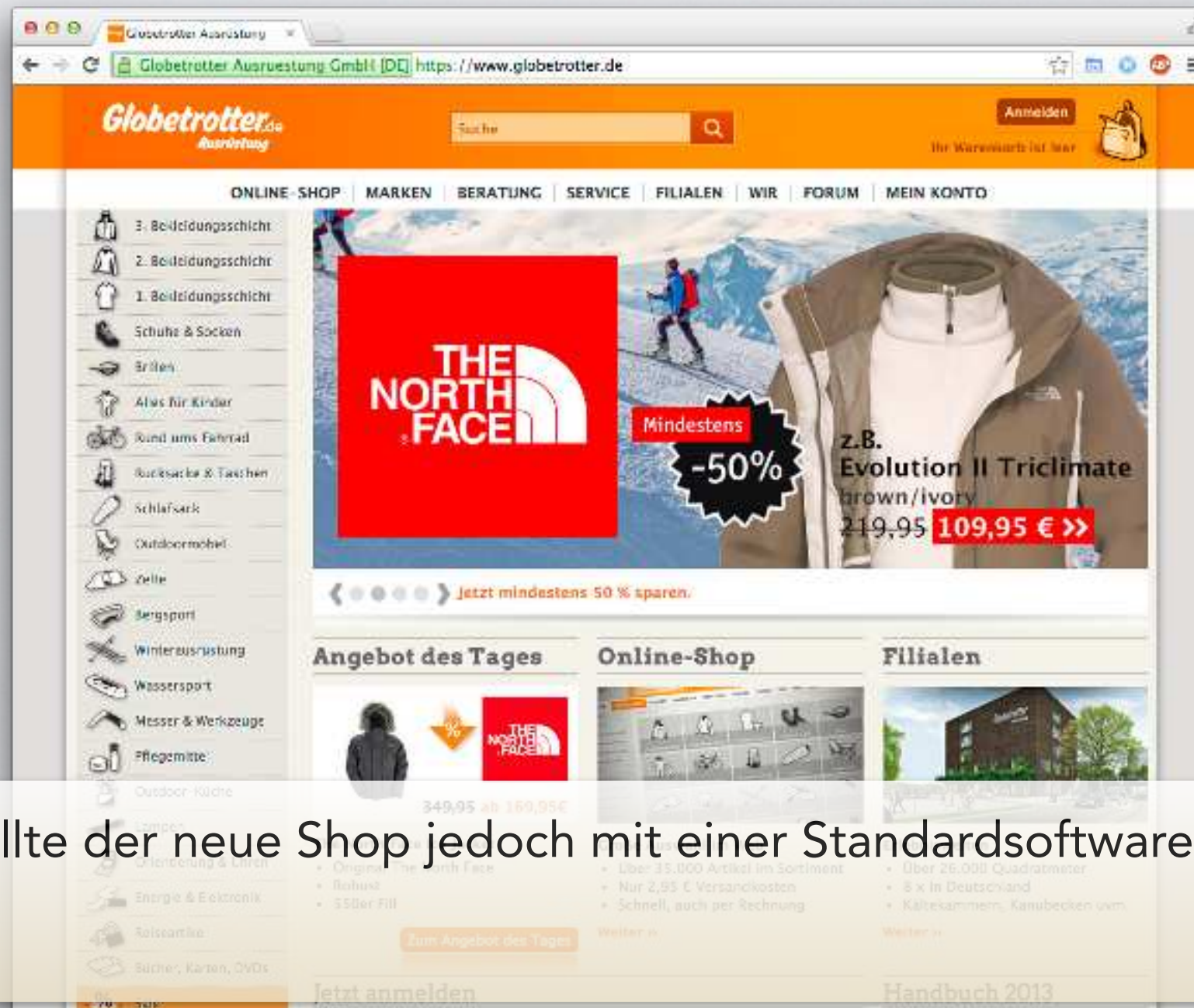
Eigenentwicklung



Seitdem hat sich viel getan und wir haben im Februar 2013 einen komplett neu entwickelten Shop gelauncht. Dabei handelt es sich um eine vollständige Eigenentwicklung.

URSPRÜNGLICHER PLAN

Standardsoftware



Ursprünglich sollte der neue Shop jedoch mit einer Standardsoftware umgesetzt werden.

WHAT HAPPENED?

FEATURES, FEATURES, FEATURES

- Produktvergleich
- Merklisten
- Gutscheine
- Bestellhistorie
- Backoffice (Artikel- und Bestellverwaltung)

FEATURES, FEATURES, FEATURES

- Produktvergleich
- Merklisten
- Gutscheine
- Bestellhistorie

Standardlösungen locken mit umfangreichen Featurelisten. Leider gibt es mit

- Backoffice (Artikel- und Bestellverwaltung)
Features, die für eine breite Masse von Installationen und Unternehmen geschrieben wurden, ein Problem...

„Die Features passen fast zu unseren Anforderungen.“

–Random Globetrotter Employee

CUSTOMIZING

Bestehende Software wird an individuelle Anforderungen angepasst



CUSTOMIZING

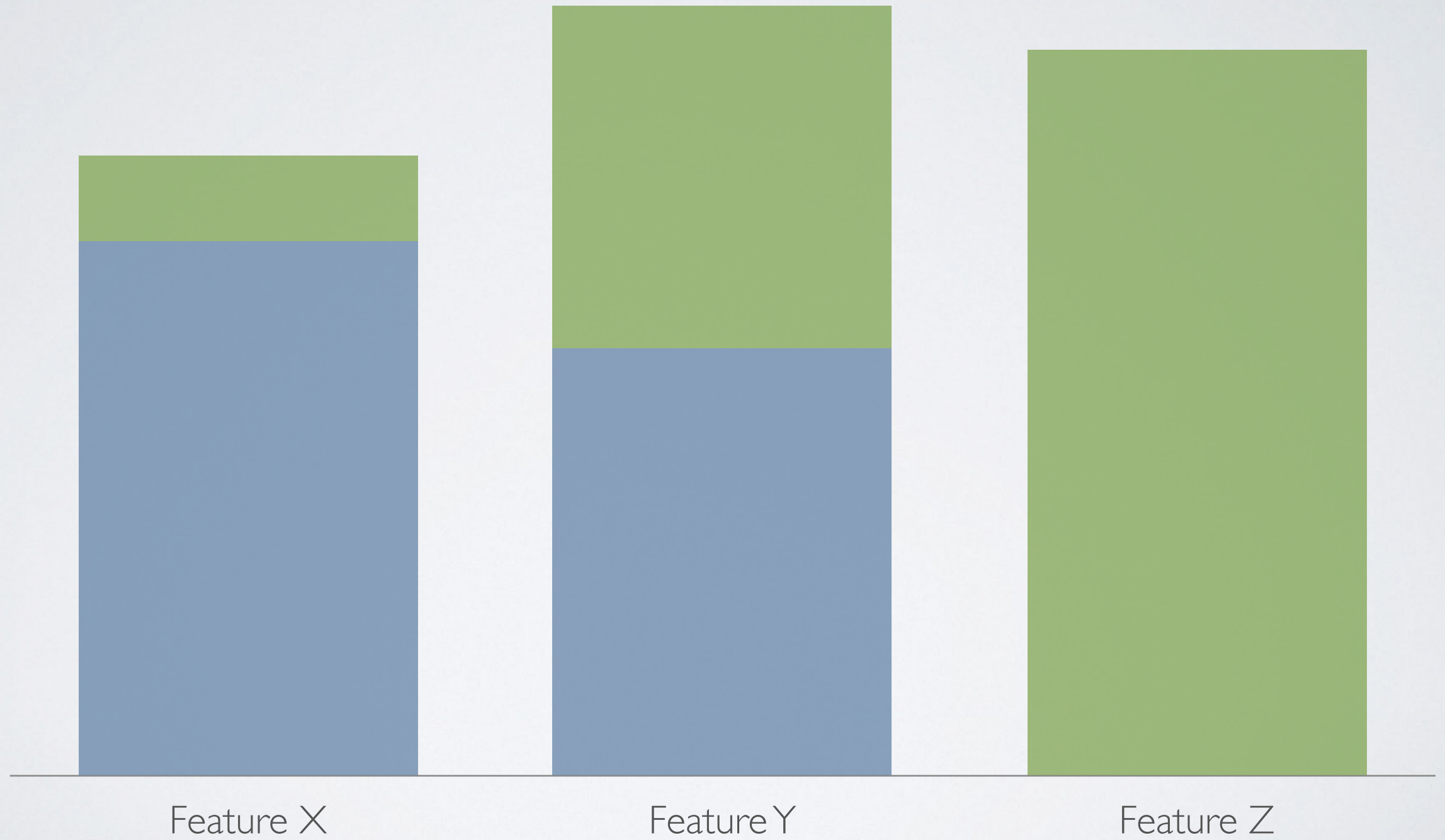
Bestehende Software wird an individuelle Anforderungen angepasst



Die vermeintliche Lösung: Customizing! Jede Standardsoftware bietet Möglichkeiten, Features an individuelle Anforderungen anzupassen.

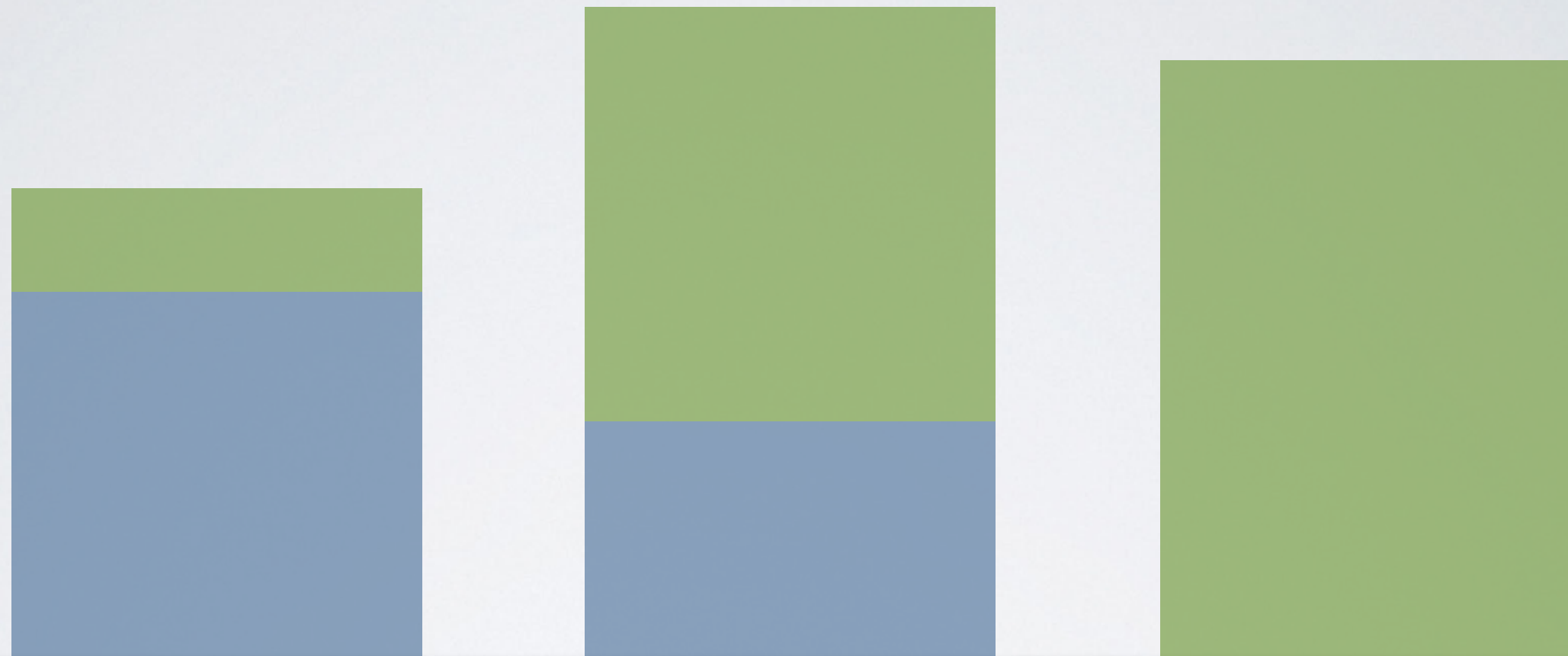
■ Core Code

■ Customization



■ Core Code

■ Customization



Das führt mitunter aber dazu, dass eine nicht unerhebliche Menge Code geschrieben werden muss, damit das Feature den Anforderungen entspricht. So wird eine i.d.R. ohnehin schon große Codebasis weiter aufgebläht, was Wartung und Bewahrung der Updatefähigkeit stark erschwert.

Feature X

Feature Y

Feature Z

HOUSTON, ...

- Artikelimport
 - Varianten- und Merkmalstrukturen passen nicht zusammen
- Performance
 - aufwändiges Caching, Probleme bei Invalidierung
 - ohne Cache ein CPU Core je Request

HOUSTON, ...

- Artikelimport
 - Varianten- und Merkmalstrukturen passen nicht zusammen

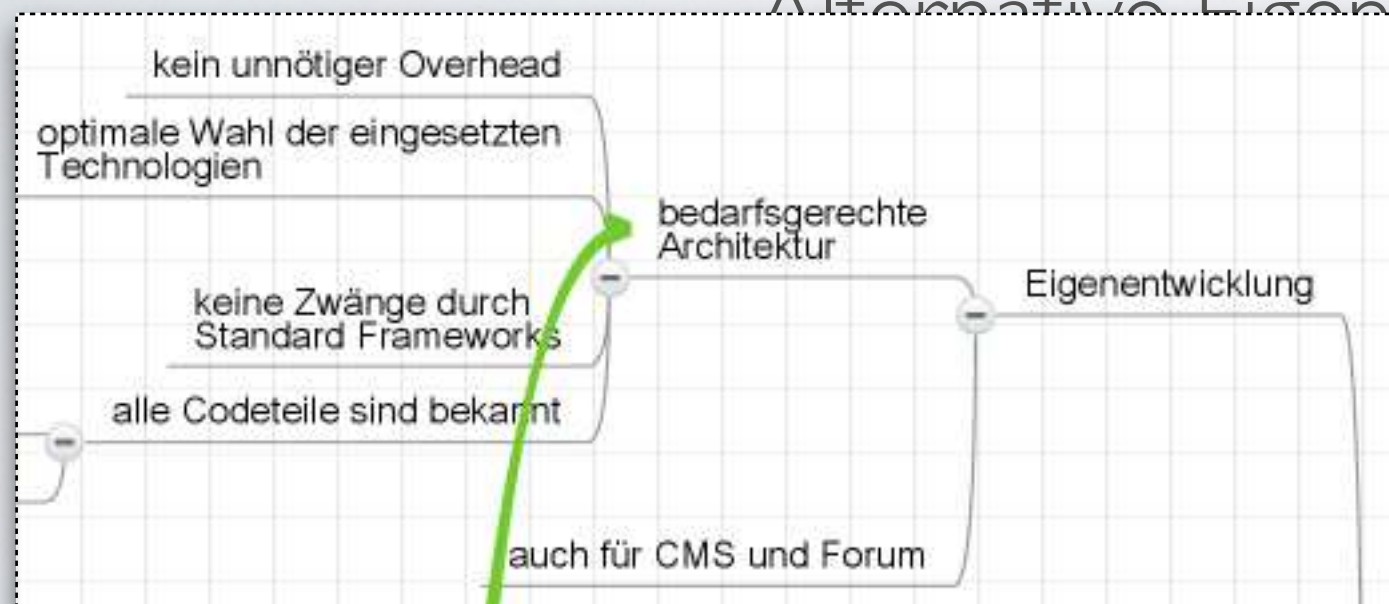
In unserem Fall hatten wir weitere Schwierigkeiten: Der Import unserer Artikeldaten erwies sich als kompliziert, da die Datenstrukturen sehr unterschiedlich waren. Das weitaus größere Problem stellte allerdings die Performance des Shops dar. Ohne aufwändiges Caching war unser Prototyp quasi nicht benutzbar, was auch den Entwickleralltag nicht angenehmer machte. Ausgiebige Lasttests führten zu der bitteren Erkenntnis, dass wir in einer Produktivumgebung pro Request einen CPU-Core hätten vorsehen müssen. Das war nicht akzeptabel.



Die Summe der aufgeführten Probleme führte zu der Erkenntnis: Wir steuern auf eine Sackgasse zu!

THE WAY OUT

Alternative Eigenentwicklung



Der Traum

GT21 Reboot



Der für uns einzige sinnvolle Ausweg: Wir entwickeln eine auf uns zugeschnittene, eigene E-Commerce Plattform. Das aufkeimen dieser Idee führte zu der hier gezeigten Mindmap, in der wir versuchten, Chancen und Risiken zu ermitteln.

SOFTWAREARCHITEKTUR

Dreizehntes Hamburg Web Performance Meetup / Treffen der PHP- Usergroup Hamburg

10. September 2013 · 19:00

OTTO



High-performance Websites,
Arne Blankerts und Stefan Pribsch von thePHP.cc

Kurze Antwortzeiten sind heute elementar, nicht nur für Online-Shops. Was tun, wenn die Performance nicht ausreicht? Stärkere Server kaufen?

In die Cloud migrieren? Noch eine Caching-Schicht einführen, oder die Schuld auf die Datenbank schieben? Das kann man alles tun, aber bevor man Hardware-Hersteller glücklich macht, sollte man

Arne Blankerts und Stefan Pribsch von thephp.cc haben am 10.09.13 bereits einen Vortrag über moderne Architekturen von Websites gesprochen und dabei auch unsere Plattform vorgestellt. Es gibt einen Videomitschnitt von Rainer Schleevoigt:

<http://lecture2go.uni-hamburg.de/veranstaltungen/-/v/15297>

ARCHITECTURE IS THE KEY

- Wichtig: nicht sofort in Technologien denken
(„Webprojekte macht man jetzt mit Rails“)
- Technologieunabhängige Architektur steht am Anfang

Webserver

StoreFront

Search

DataPool

Middleware

StoreBack

ERP & PIM

Webserver

StoreFront

Search

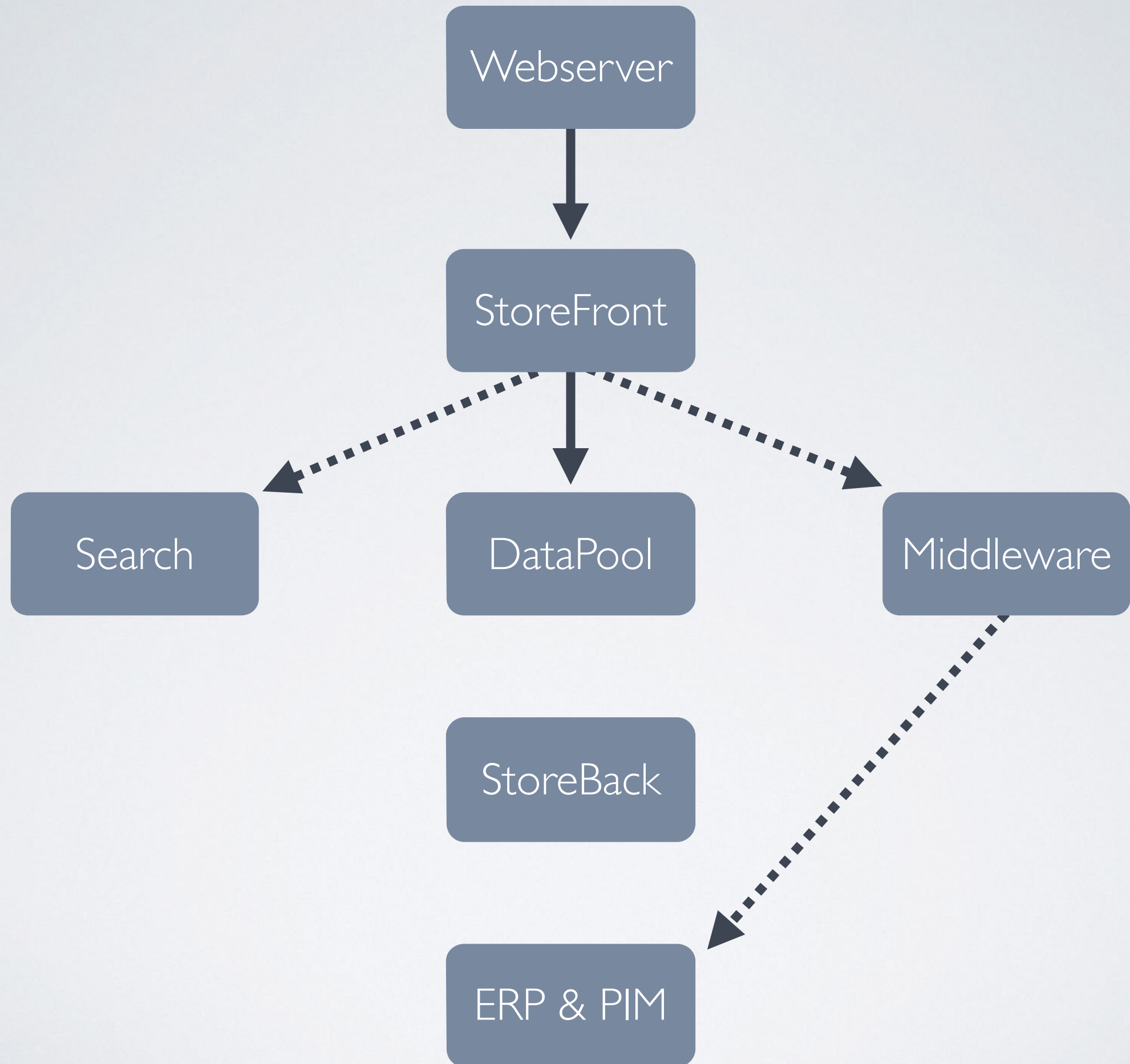
DataPool

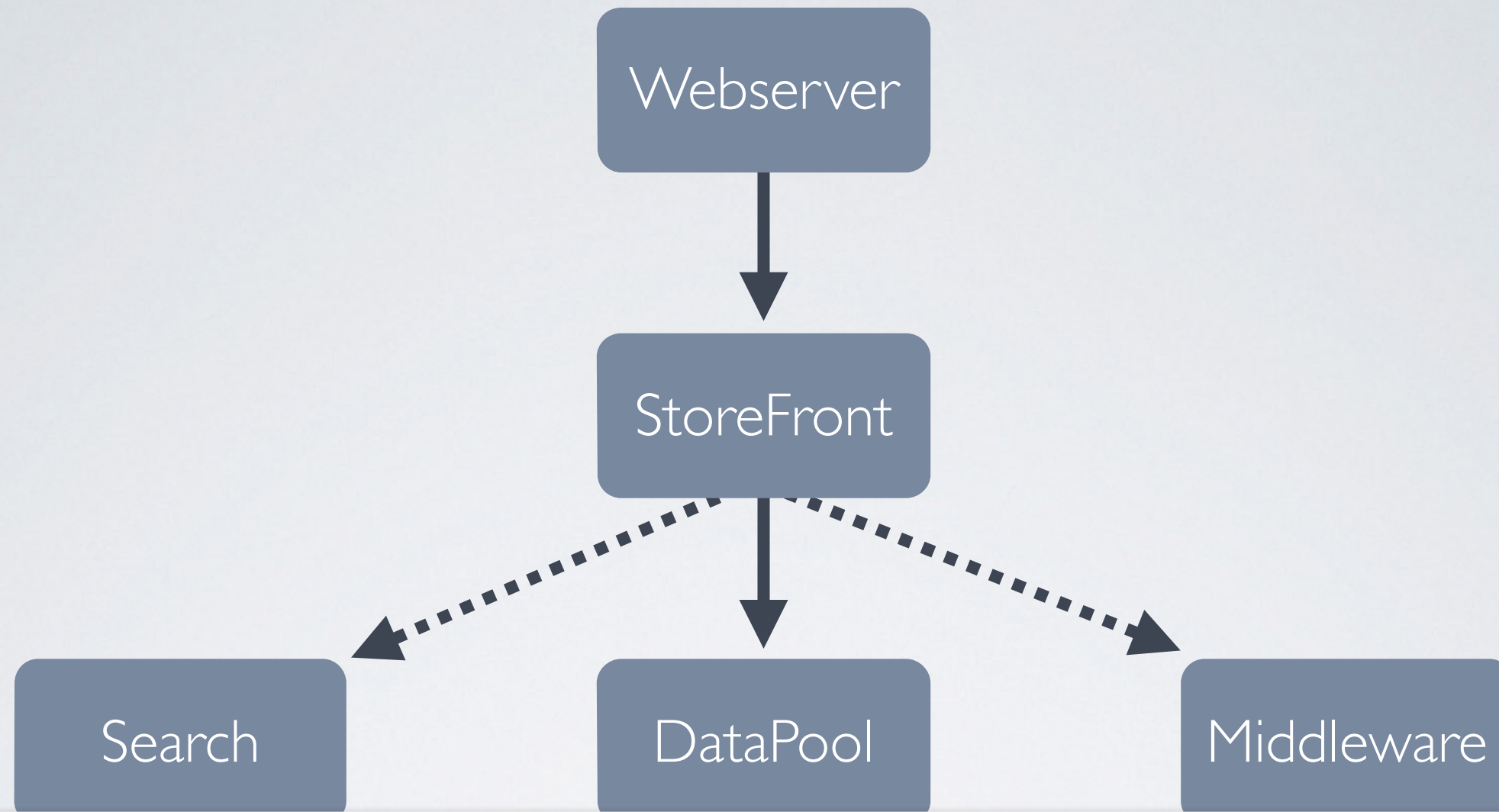
Middleware

StoreBack

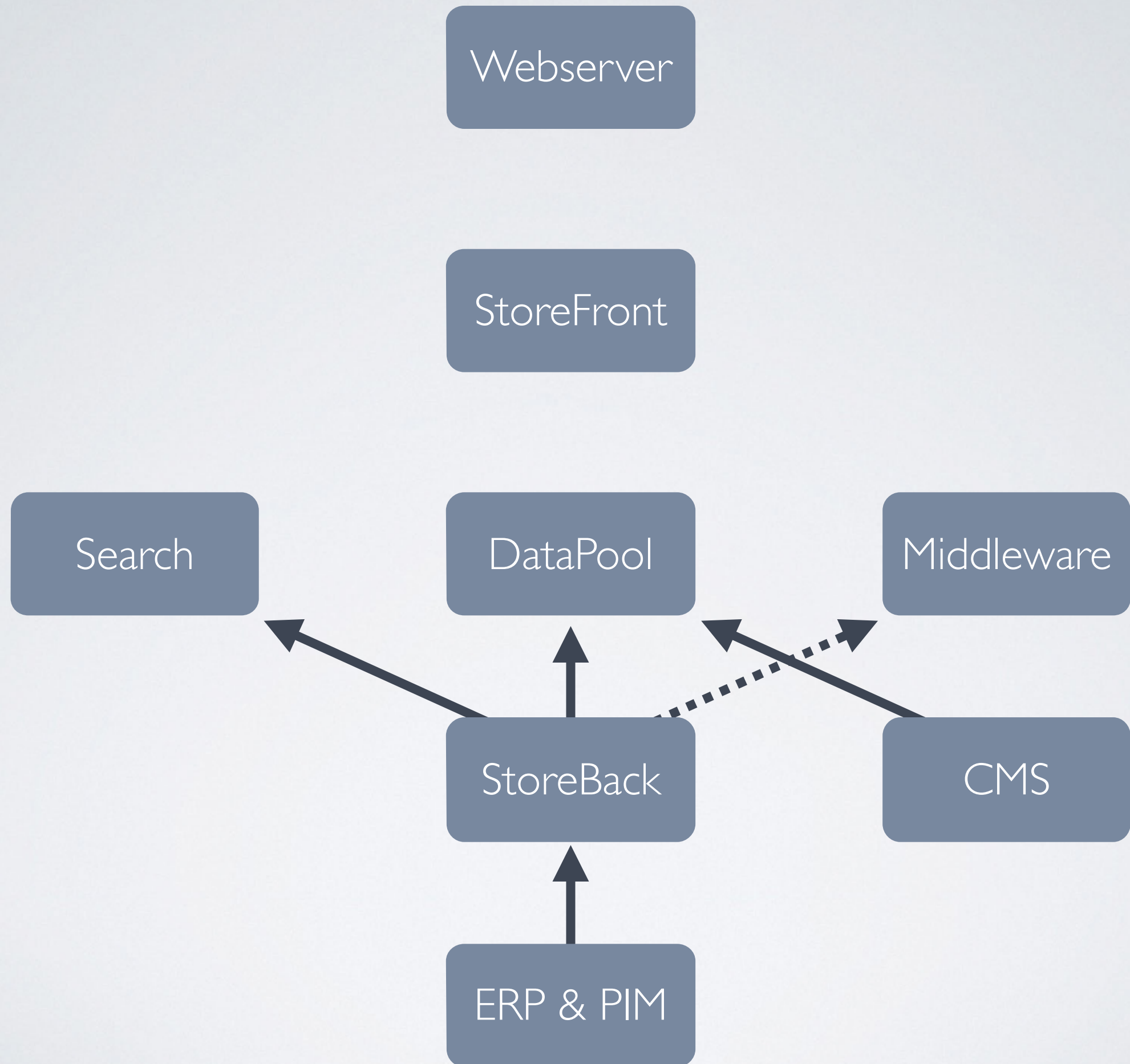
ERP & PIM

Die wichtigsten Komponenten unserer Architektur. Die Grundidee: Das Store Frontend sollte so dumm wie möglich sein - das führt zu hoher Verarbeitungsgeschwindigkeit. Aufwändige Prozesse sollten unabhängig von eingehenden Requests in einem Backend laufen.





Beispiel: Ein eingehender Request (z.B. für eine Artikeldetailseite) geht ein. Unser StoreFront soll möglichst wenig Arbeit haben, damit der Request schnell beantwortet werden kann. Dazu legen wir so viele Seitenbestandteile wie möglich als vorgerenderte HTML-„Snippets“ in einem DataPool ab. Das Frontend soll sich diese Snippets dann holen und in eine Seite einkleben. Je nach Request müssen ggf. auch noch eine Suchmaschine und eine Middleware befragt werden.



Webserver

StoreFront

Search

DataPool

Middleware

Datenversorgung: Das ERP System schickt uns neue Artikeldaten. Im StoreBack werden diese verarbeitet, es entstehen die genannten Snippets, die in den DataPool gelegt werden. Zusätzlich wird die Suchmaschine mit den neuen Daten versorgt. Ggf. werden auch Daten an die Middleware gesendet.

Auch das CMS arbeitet als „Snippet-Lieferant“ und exportiert Seiten als HTML in den DataPool. Somit ist das CMS an der Verarbeitung von eingehenden Requests gar nicht beteiligt und ist leicht austauschbar.

ERP & PIM

CMS

1-Personen-Zelte

151Bewertungen ★★★★★



Extrem niedriges Gewicht und kleines Packmaß machen das Einmannzelt zum idealen Begleiter auf Touren, bei denen es auf jedes Gramm Gewicht ankommt. Bei uns finden Sie sowohl die minimalistische Not-Unterkunft als auch das vollwertige, komfortable 1-Personen-Zelt mit Platz für Ihr Gepäck. » **Mehr**

» **4-Seasons Video**

1-Personen-Zelte durchsuchen



Alle Filter zurücksetzen ✕

Marken durchsuchen

- ☐ Helsport (1)
- ☐ Hilleberg (3)
- ☐ Nordisk (2)
- ☐ Vaude (1)
- ☐ Wechsel (4)

Stichwörter durchsuchen

- ☐ 1 Personen Zelt (11)
- ☐ Geodät (2)
- ☐ Kuppelzelt (4)
- ☐ Neu 2013 (5)
- ☐ Tunnelzelt (6)

☐ Nur **sofort lieferbare** Artikel

Preis / €

0

665

... mehr Filter anzeigen ▼

Sortieren:

Name

Preis

Bewertung



★★★★★



★★★★★



★★★★★



★★★★★



Beispiel für die Verwendung von Snippets: eine Kategorieseite

399,95 €

495,00 €

665,00 €

595,00 €

Hilleberg Unna

changes when the
backend receives
new data



changes when a user
submits a new rating

depends on the request

changes when the
backend receives
new data



changes when a user
submits a new rating

Ein Artikel in einem Kategorielisting besteht aus zwei Snippets, die fertig
vorgerendert im DataPool liegen. Nur der Preis muss während des Requests ermittelt
werden, da er je nach User variieren kann (z.B. durch individuelle Kundenrabatte oder
abweichende MwSt.-Sätze).

◀ Zurück



[mehr zu diesem Artikel](#) ▼

HILLEBERG
THE TENTMAKER

Hilleberg Akto grün

495,00 €

Liefertermin: 21.11.13



Bestellnummer: 10.05.61-002

Farbe



Größe

☒ 1

Liefertermin:
21.11.13

495,00 €

Menge

In den Warenkorb legen ▶



Siehe auch: Handbuch, 2013 Seite 516

Details

Rewertungen

Material

Zubehör

Verfügbarkeit

◀ Zurück

[mehr zu diesem Artikel](#) ▼



HILLEBERG
THE TENTMAKER

Hilleberg Akto grün

495,00 €

Liefertermin: 21.11.13



Bestellnummer: 10.05.61-002

Farbe



Größe



Liefertermin:
21.11.13

495,00 €

Ein weiteres Beispiel: Die Artikeldetailseite wird fast vollständig vorgerendert. Nur die markierten Bereiche werden nachgeladen und ggf. während des Requests berechnet.



Unser StoreFront weiß kaum etwas über die Artikel, da nur eine Artikelnummer für die Identifizierung der benötigten Snippets gebraucht wird. Das führt zu extrem schnellen Antwortzeiten (siehe auch „Testimonials“).

Details

Rewertungen

Material

Zubehör

Verfügbarkeit

TECH STACK

TECH STACK

```
1 <<<<<< HEAD
2 Developer A thinks we should use Redis
3 =====
4 Developer B thinks we should use PostgreSQL
5 >>>>>> 926fb8478ff513110488d198107cbb3700cf80fe
```

```
6
7 >>>>>> 0504p8418442J3JJ0488qJ08J0\CPP3\00C48046
8 0040b0c4p 0UTUX2 46 24000 0 026 4024046205
```


TECH STACK

Webserver

NGINX

StoreFront



DataPool



Search



Middleware



StoreBack





redis

REDIS

- In ANSI C geschriebener Key-Value Storage
- Unterstützt verschiedene Datentypen:
 - Strings, Hashes, Lists, Sets und Sorted Sets
- Viele Operationen sind atomar

REDIS

- Daten werden vollständig im RAM gehalten
- zwei optionale Wege zur Persistierung
- Master/Slave Replication
- Support von Transaktionen

REDIS

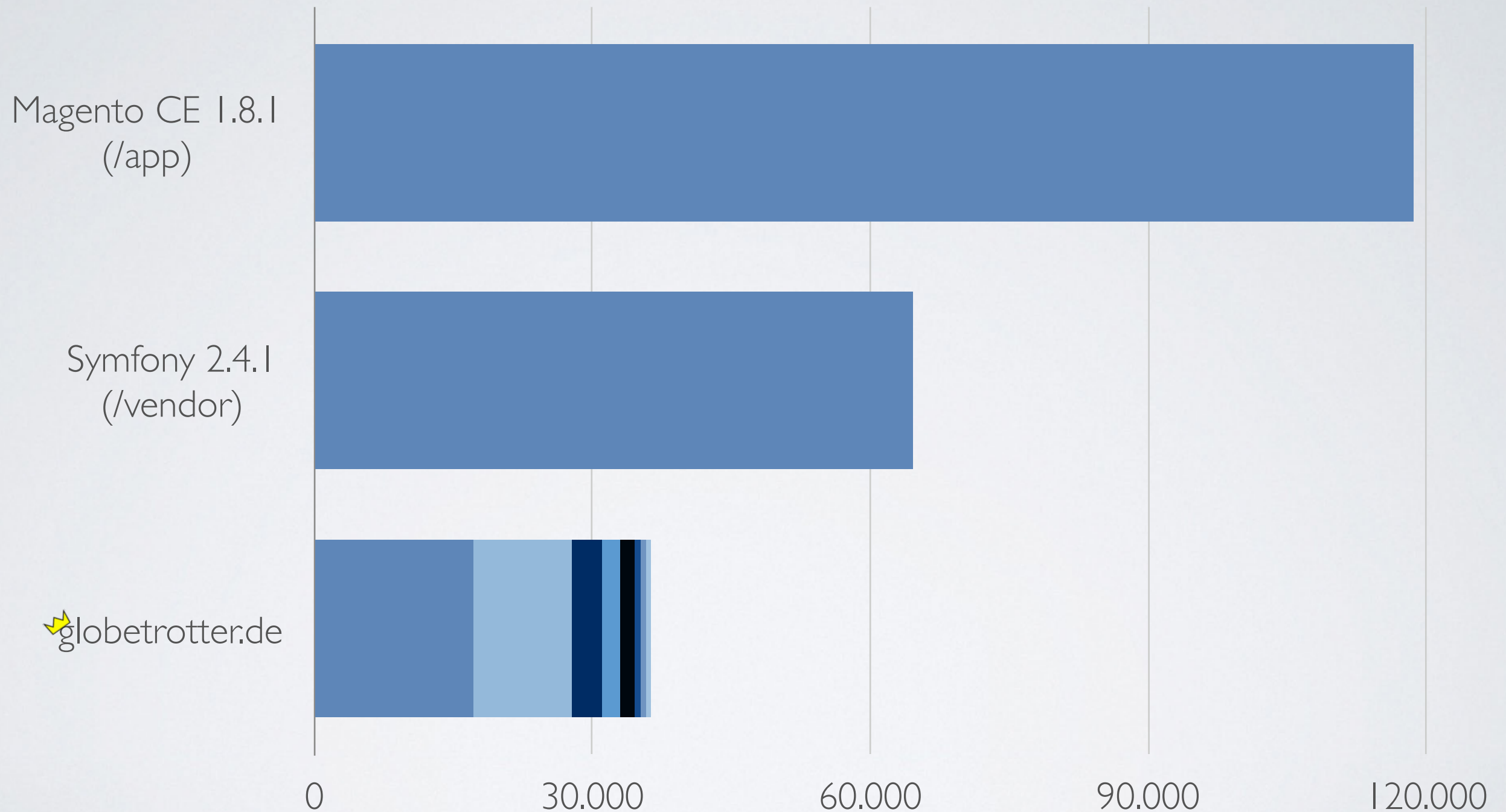
- Key TTL
- Publish / Subscribe
- Lua Scripting
- (Clustering)

```
2. sheuer@vgta01-test: ~ (ssh)
sheuer@vgta01-test:~$ redis-benchmark -q -s /srv/redis/tmp/redis.sock
PING_INLINE: 138888.89 requests per second
PING_BULK: 147058.81 requests per second
SET: 93457.95 requests per second
GET: 192307.69 requests per second
INCR: 112359.55 requests per second
LPUSH: 119047.62 requests per second
LPOP: 131578.95 requests per second
SADD: 204081.64 requests per second
SPOP: 200000.00 requests per second
LPUSH (needed to benchmark LRANGE): 129870.13 requests per second
LRANGE_100 (first 100 elements): 67114.09 requests per second
LRANGE_300 (first 300 elements): 22075.05 requests per second
LRANGE_500 (first 450 elements): 13262.60 requests per second
LRANGE_600 (first 600 elements): 10362.69 requests per second
MSET (10 keys): 46948.36 requests per second

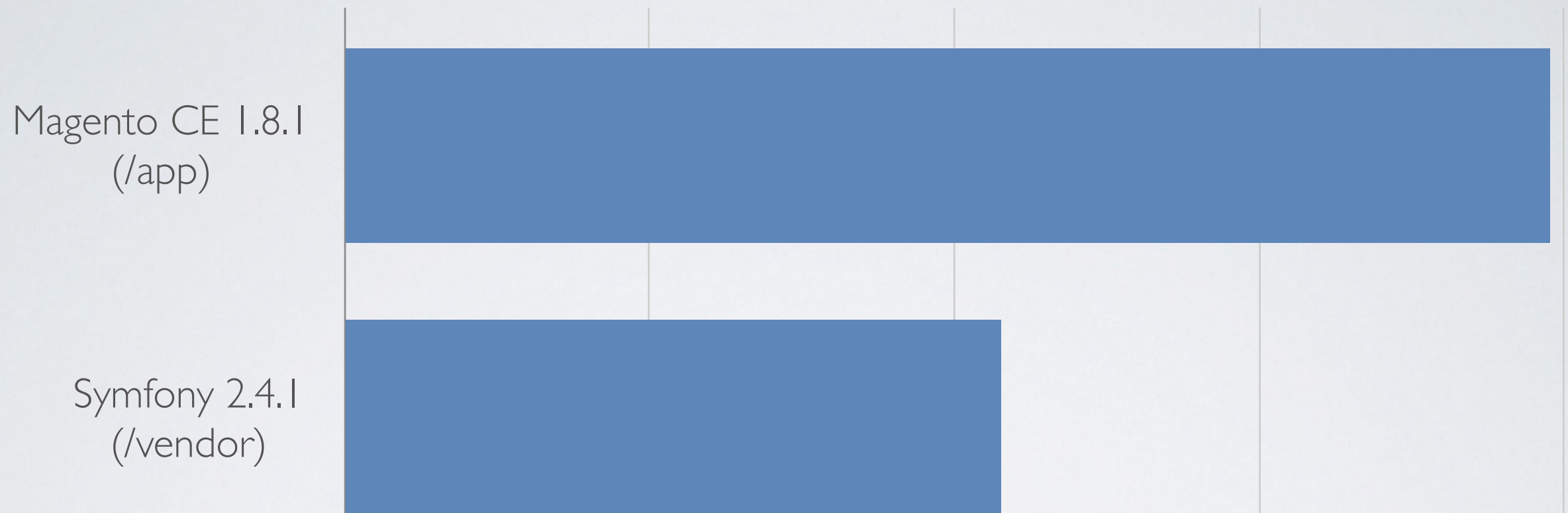
sheuer@vgta01-test:~$
```

Screenshot von redis-benchmark auf einer unserer Testmaschinen. ca. 93.500 SET-Anweisungen und ca. 192.300 GET-Anweisungen pro Sekunde sprechen eine deutliche Sprache!

logical lines of code (LLOC), tests excluded



logical lines of code (LLOC), tests excluded



Vergleich unserer Codebasis (Stand 21.01.14) mit Magento und Symfony. Es wird deutlich, dass wir mit verhältnismäßig wenig Code eine vollständige Applikation gebaut haben, während mit den Vergleichsprodukten noch keine einsatzfähige Lösung vorliegt. Hier würde jeweils ein nicht unerheblicher Teil zusätzlichen Codes entstehen, um einen mit unserem Shop vergleichbaren Stand zu erhalten.

TOOLCHAIN

GIT + GITORIOUS

The screenshot displays the Gitorious web interface. At the top, there is a navigation bar with links for 'Dashboard', 'Administration', a user profile for '~sebastian', a notification icon, and a 'Logout' button. Below this, the Gitorious logo is on the left, and 'Activities', 'Projects', and 'Teams' are on the right. A search bar is located in the top right corner. The main content area shows the 'GT21' project, with a sub-section for the 'design' repository. The repository description is 'Repository für die HTML-Umsetzung des neuen Designs'. Below this, there are buttons for 'Clone & push urls', 'GIT', and 'SSH'. The SSH URL is 'git@git.globetrotter.de:gt21/design.git'. At the bottom of the repository section, there are buttons for 'Commit log', 'Source tree', 'Merge requests (0)', 'Clone repository', and 'Unwatch'. On the right side, there is a sidebar with a list of actions: 'Open wiki', 'View project members', 'Add repository', and 'Edit project'. Below this, there is a 'Project information' section showing 'License: None' and 'Owner: Entwickler'. At the bottom of the sidebar, there is a 'Repository clones' section.

GT21

globetrotter.de Relaunch

design

Repository für die HTML-Umsetzung des neuen Designs

Clone & push urls ☐ GIT ☒ SSH git@git.globetrotter.de:gt21/design.git

Commit log Source tree Merge requests (0) Clone repository Unwatch

Open wiki

View project members

Add repository

Edit project

Project information

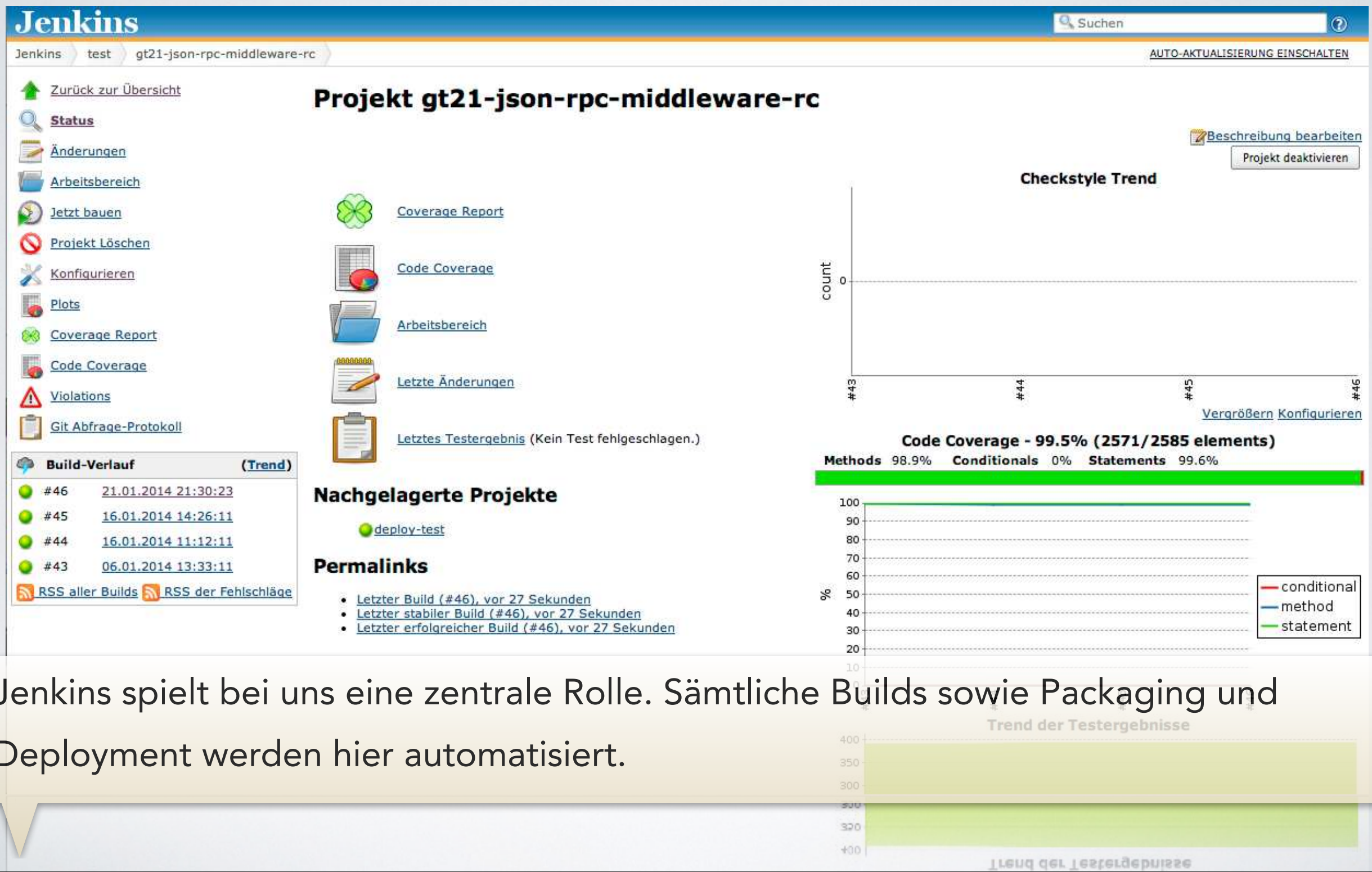
License: None

Owner: Entwickler

Repository clones

Gitorious wird bei uns zur Verwaltung der Git-Repositories eingesetzt. Ein großer Teil der Features, wie Forking und Pull Requests, werden nicht genutzt.

JENKINS CI



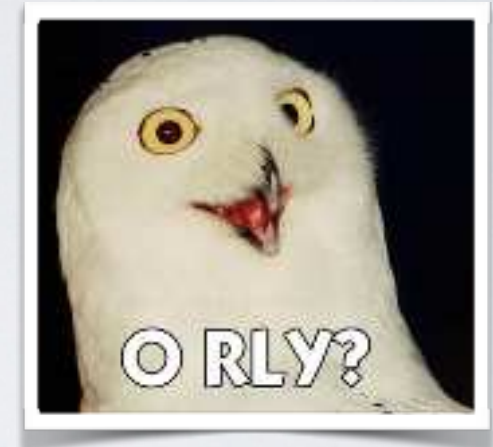
Jenkins spielt bei uns eine zentrale Rolle. Sämtliche Builds sowie Packaging und Deployment werden hier automatisiert.

TOOLCHAIN

- Jenkins Jobs werden bei jedem Git Push getriggert
 - Build-Config mit ANT
 - Führt alle Tests aus
 - Triggert Folgejobs, z.B. Deployment

TESTING

- Unit Tests PHPUnit



- Integrationstests PHPUnit



Auch unsere Integrationstests schreiben wir in PHPUnit!

TESTING

```
1  <?php
2  namespace Globetrotter\StoreFront\Tests\Integration\Cart;
3  use Globetrotter\StoreFront\Cart\Cart;
4  use Globetrotter\StoreFront\SessionManagerInterface;
5  use Globetrotter\StoreFront\Tests\Integration\IntegrationTestCase;
6
7  class AddToCartTest extends IntegrationTestCase
8  {
9      public function testAddProductToCart()
10     {
11         $builder = $this->_getApplicationBuilder();
12         $builder->setUri('/cart/add');
13         $builder->addPost('sku', '100561001');
14         $builder->addPost('qty', '1');
15
16         $app = $builder->build();
17         $app->run();
18
19         $factory = $app->getFactory();
20
21         /** @var Cart $cart */
22         $cart = $factory->getCart();
23         $this->assertSame(1, $cart->getProductCount());
24         $this->assertCartContainsProduct('100561001', $cart);
25     }
```


TESTING

```
1  <?php
2  namespace Globetrotter\StoreFront;
3
4  require __DIR__ . '/../src/autoload.php';
5
6  $application = new Application(
7      new HttpRequest($_GET, $_POST, $_SERVER, $_COOKIE), new NativeSessionManager()
8  );
9  $application->run()->flush();
10
```

Unsere index.php. Die Application Klasse enthält das gesamte Bootstrapping. Request und Session werden bei der Objekterzeugung übergeben, so dass sie in Tests leicht ausgetauscht werden können.

DEPLOYMENT

- Wir könnten Continuous Delivery...
 - ...aber wir trauen uns (noch) nicht :)
- Grundidee: mehrere Versionen parallel betreiben
- Nutzung vom Paketsystem des Betriebssystems

DEPLOYMENT

Alle	configs	deployment	dev	monitoring	prod	stage	test	+
S	W	Name ↓	Letzter Erfolg	Letzter Fehlschlag	Letzte Dauer			
		deploy-develop	23 Stunden (#215)	Unbekannt	25 Sekunden			
		deploy-service	23 Minuten (#10)	1 Monat 12 Tage (#1)	10 Sekunden			
		deploy-stage	4 Stunden 56 Minuten (#125)	Unbekannt	15 Sekunden			
		deploy-test	36 Minuten (#123)	Unbekannt	25 Sekunden			

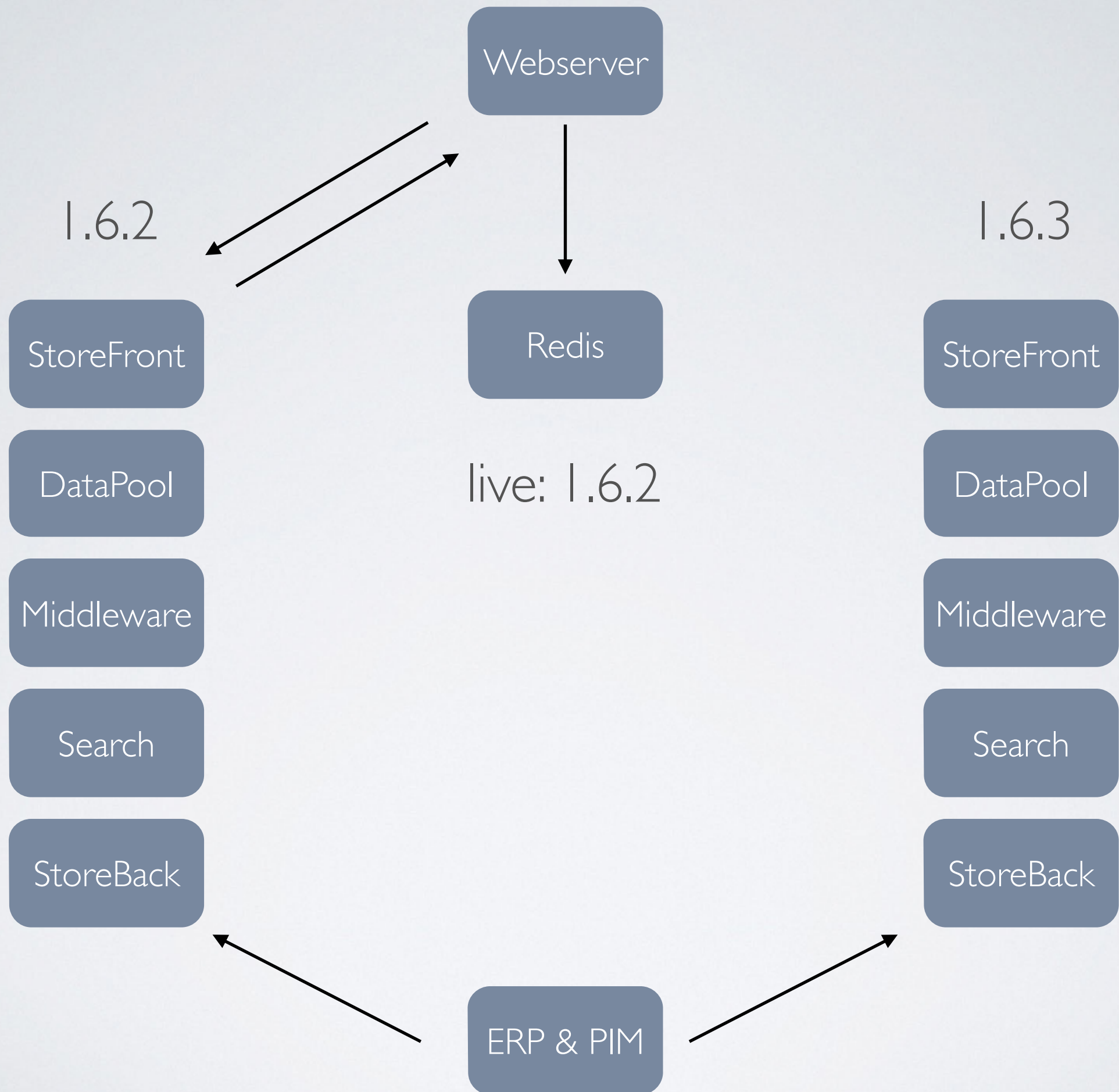


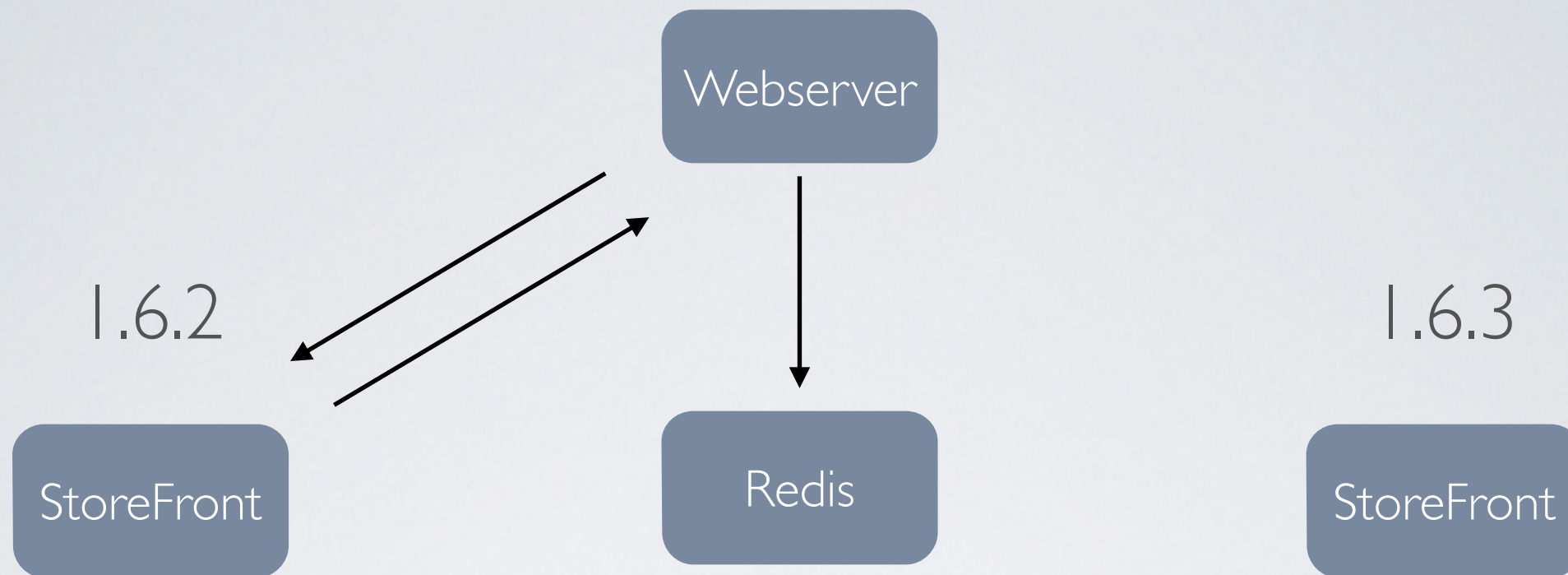
Jenkins

DEPLOYMENT

```
3. sheuer@vstorefronta01-stage: ~ (ssh)
sheuer@vstorefronta01-stage:~$ dpkg --get-selections | grep storefront
gt21-storefront-1.6.2          install
gt21-storefront-1.6.3          install
gt21-storefront-config-1.6.2   install
gt21-storefront-config-1.6.3   install
sheuer@vstorefronta01-stage:~$ dpkg -s gt21-storefront-1.6.3
Package: gt21-storefront-1.6.3
Status: install ok installed
Priority: optional
Section: all
Maintainer: Sebastian Heuer <sebastian.heuer@globetrotter.de>
Architecture: all
Version: 736-stage
Depends: php5-fpm (>= 5.3.6), gt21-datapool-1.6.3, gt21-middleware-api-1.6.3
Description: GT21 StoreFront
sheuer@vstorefronta01-stage:~$
```

Alle Komponenten werden während des Build-Prozesses in Debian Packages verpackt. Damit erfolgt das Deployment über die Paketverwaltung des Betriebssystems. Das ermöglicht auch eine einfache Abbildung und Auflösung von Paketabhängigkeiten.

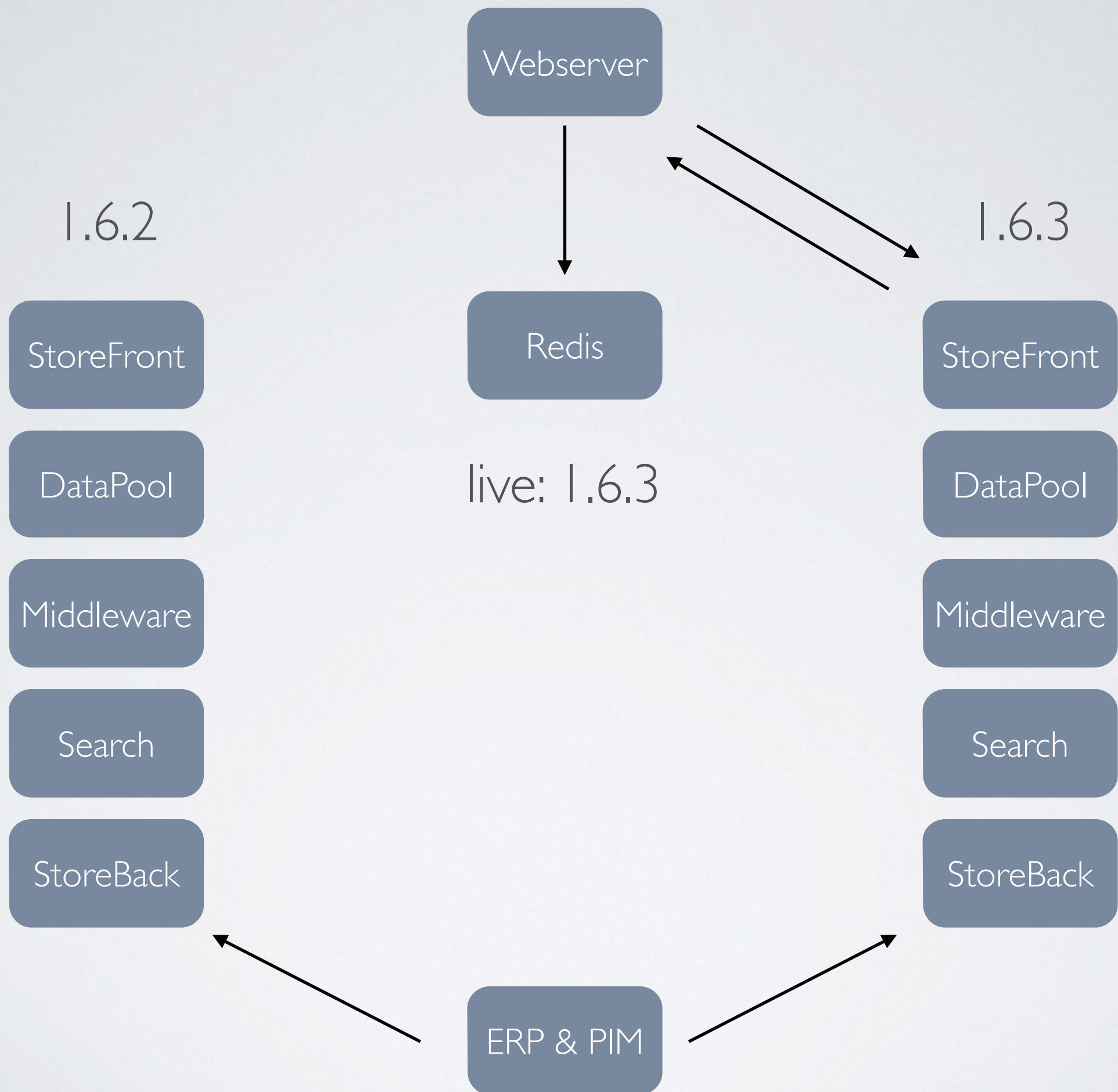


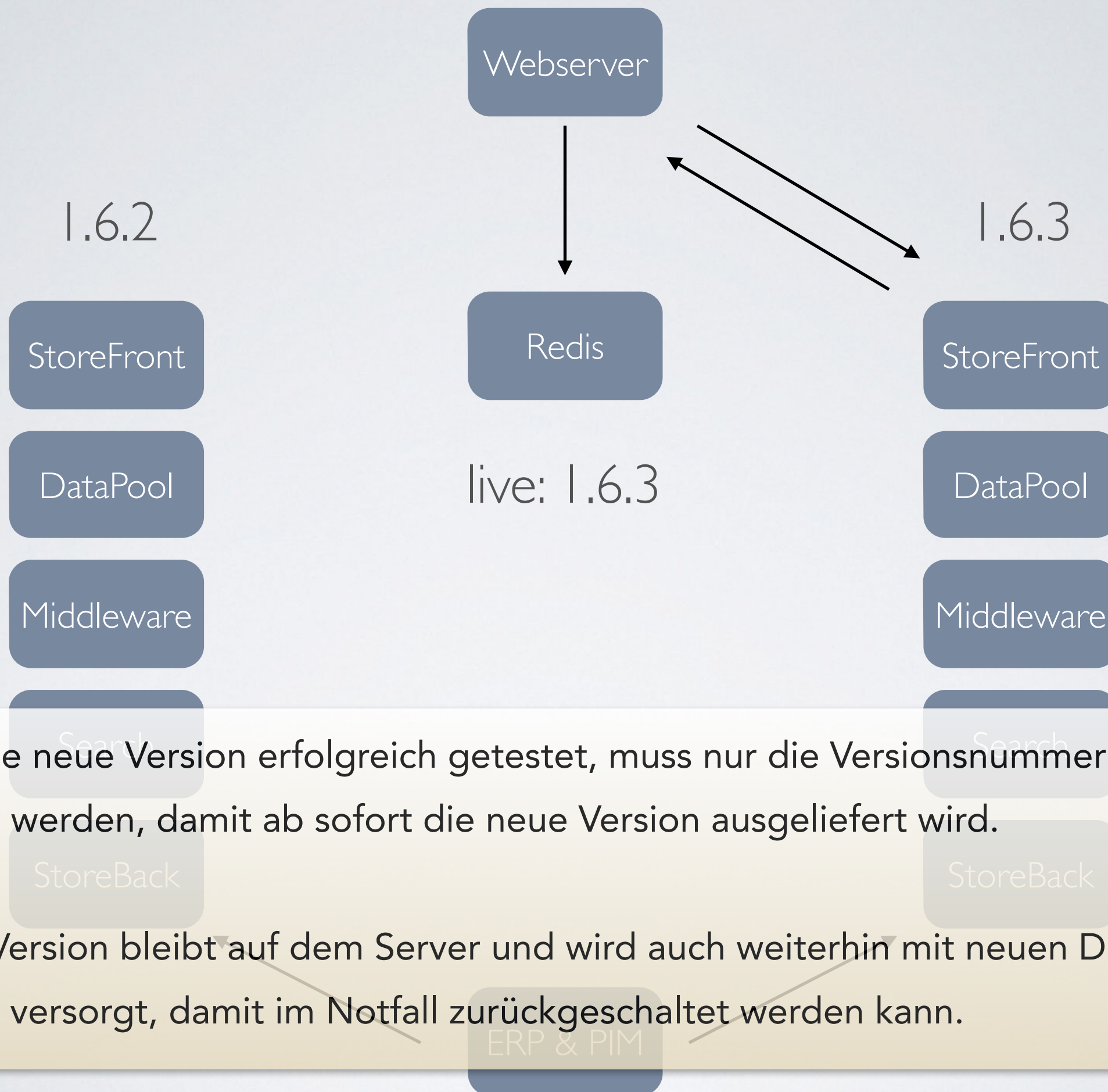


Beispiel der Versionierung: Auf den Produktivmaschinen sind die Packages der Shopversion 1.6.2 installiert. In einer zentralen, versionsunabhängigen Redis-Instanz wird in einem Key die aktuelle Live-Version gespeichert.

Der Webserver fragt Redis bei einem eingehenden Request nach dieser Version und gibt diesen dann an den passenden StoreFront weiter.

Parallel kann auf dem Server bereits Version 1.6.3 installiert und vorbereitet werden. Ebenfalls wichtig: Alle Pakete tragen eine gemeinsame Versionsnummer im Namen. Somit können auch mehrere Versionen der Applikation parallel installiert werden.





Wurde die neue Version erfolgreich getestet, muss nur die Versionsnummer in Redis geändert werden, damit ab sofort die neue Version ausgeliefert wird.

Die alte Version bleibt auf dem Server und wird auch weiterhin mit neuen Daten aus dem ERP versorgt, damit im Notfall zurückgeschaltet werden kann.

TESTIMONIALS

Es folgen Performance-Auswertungen von unserem Hoster Metaways zu einem Tag mit der bisher höchsten Last seit dem Launch.

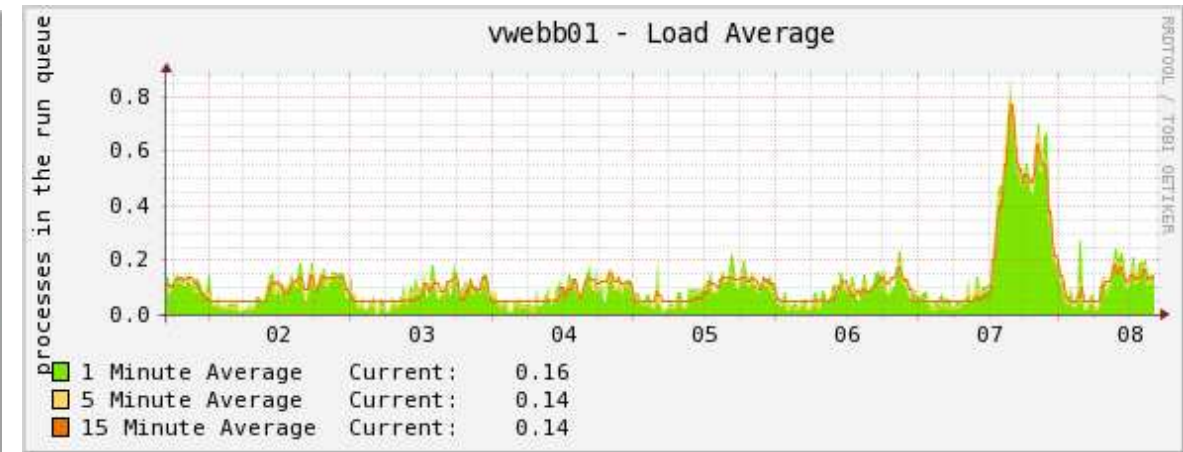
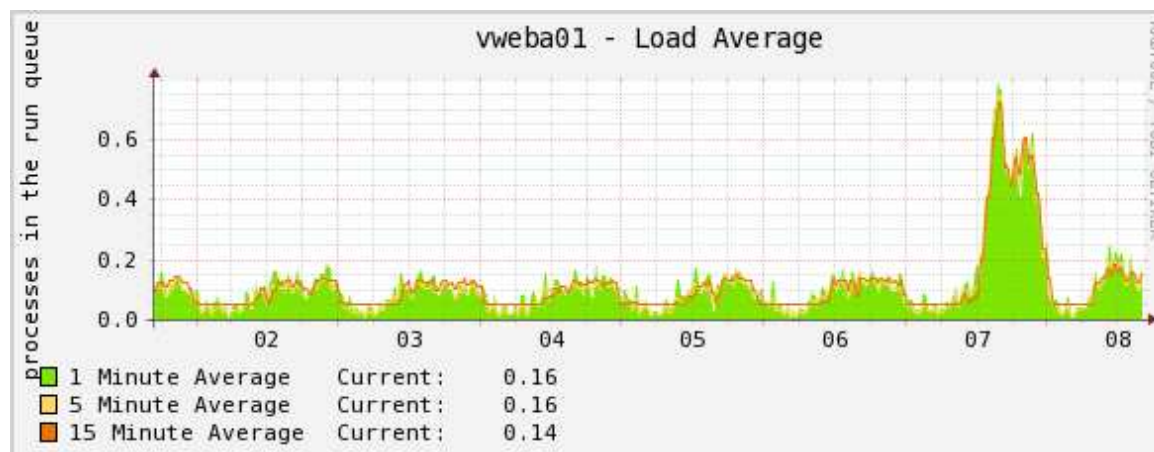
Metaways betreibt sämtliche Umgebungen (dev, test, stage und live) sowie die Entwicklungswerkzeuge.

- Trotz dieses hohen Durchsatzes sind keine Auswirkungen auf die Response Zeiten zu erkennen. Die Seite hat weiterhin performant ausgeliefert (~170ms für die Startseite, ~270ms für Seite 'Geschenkkard 50')

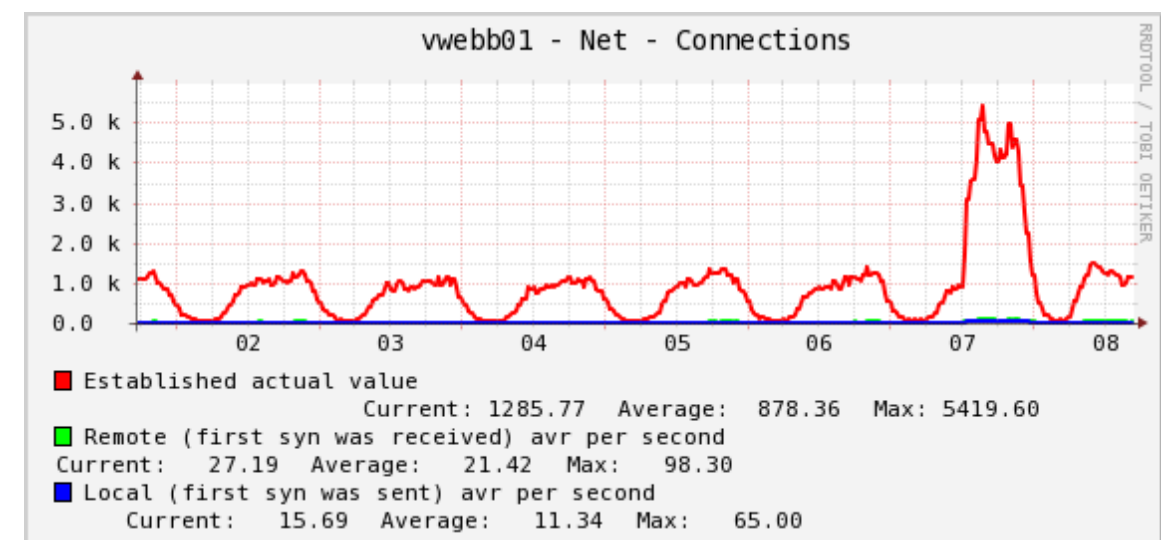
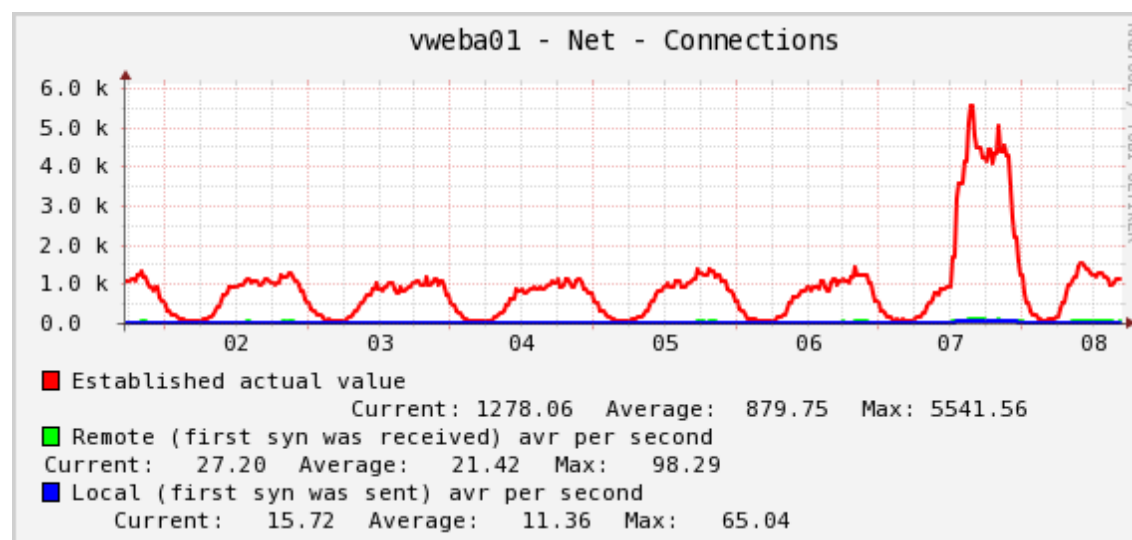
Bei den genannten Antwortzeiten handelt es sich um die Angabe „first byte to client“, d.h. ein Client erhält auf unserer Startseite nach durchschnittlich 170ms eine Antwort! Die Zeiten für das serverseitige PHP-processing liegen bei ca. 35ms.

Webserver Grafiken

- Load



- Net Connections



```

CPU: 12.15 VML906: 13.38 WAX: 82.06
[CPU] (13.12 2.0 1.0 2.0) 0.0 0.0 0.0
CPU: 13.50 VML906: 13.45 WAX: 82.38
[CPU] (13.12 2.0 1.0 2.0) 0.0 0.0 0.0
CPU: 13.10 VML906: 13.12 WAX: 82.10
[CPU] (13.12 2.0 1.0 2.0) 0.0 0.0 0.0

```

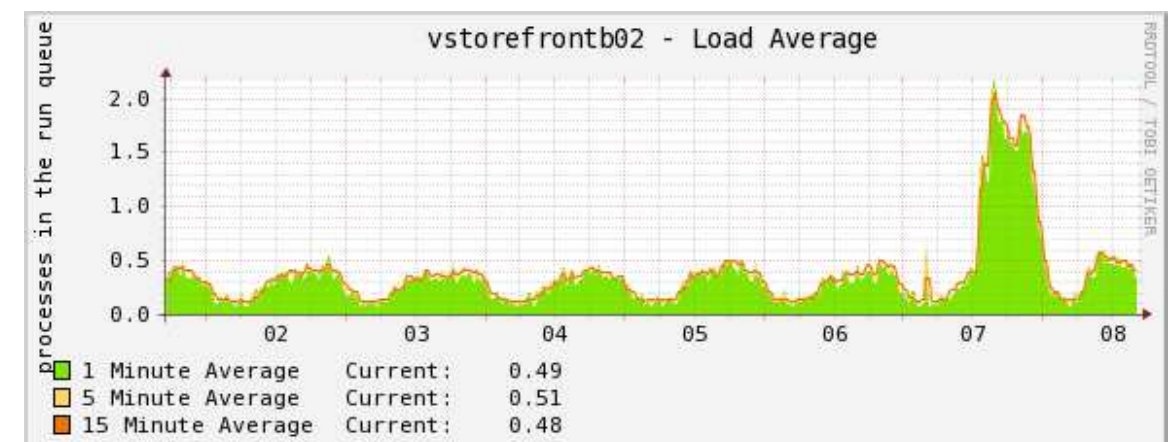
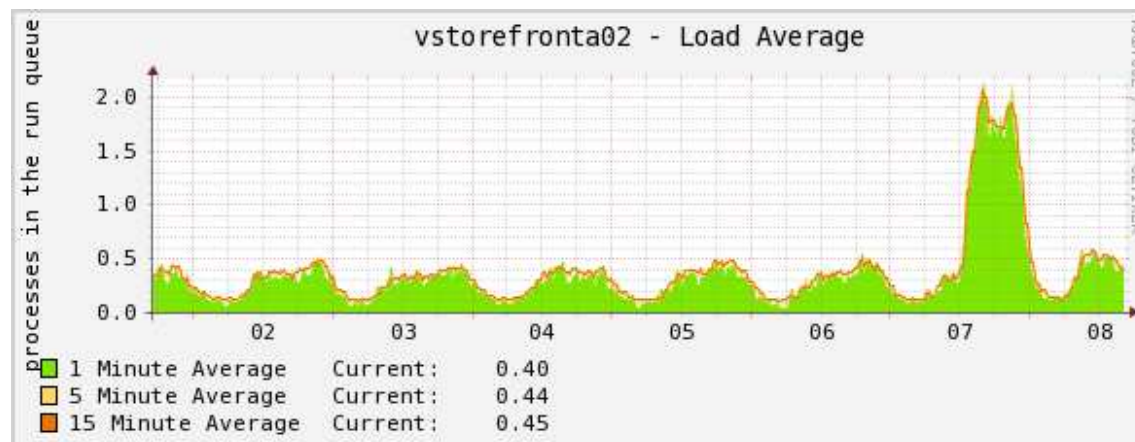
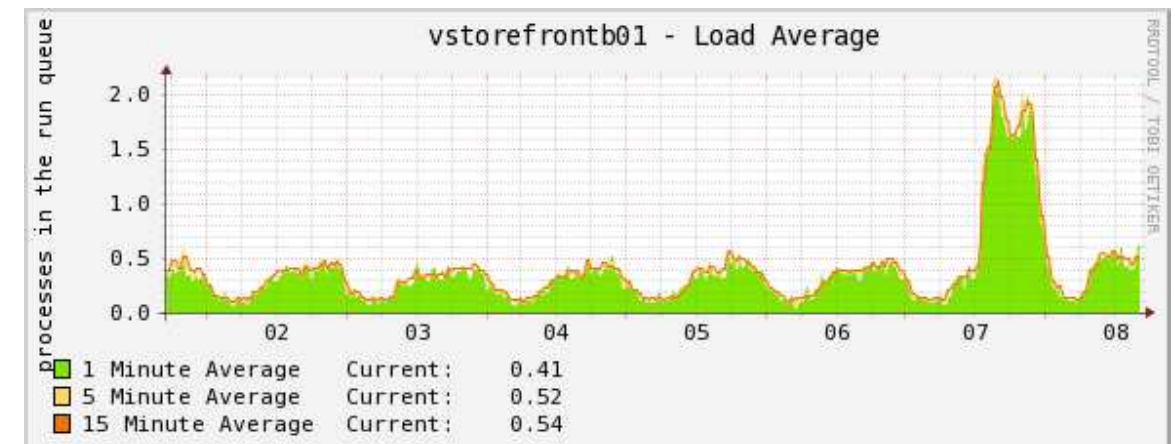
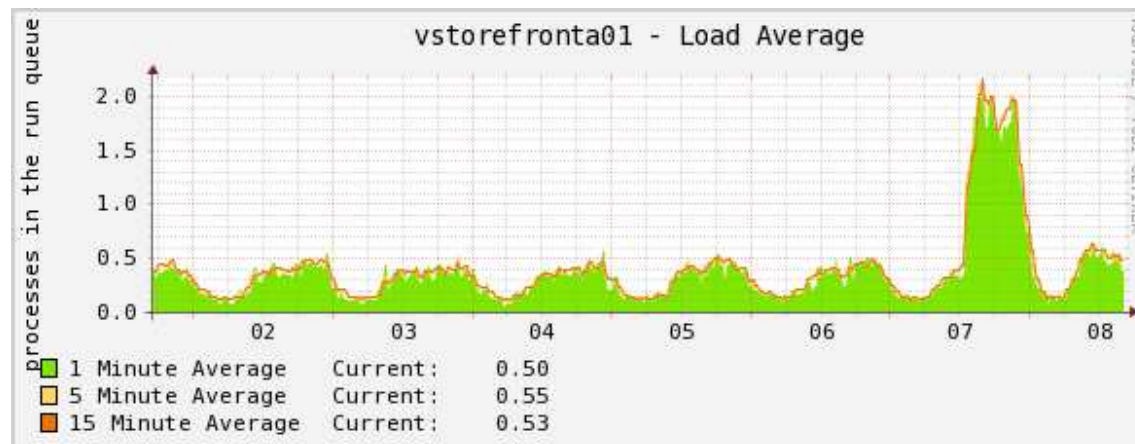
```

CPU: 12.08 VML906: 13.34 WAX: 82.08
[CPU] (13.12 2.0 1.0 2.0) 0.0 0.0 0.0
CPU: 13.78 VML906: 13.45 WAX: 82.38
[CPU] (13.12 2.0 1.0 2.0) 0.0 0.0 0.0
CPU: 13.02 VML906: 13.12 WAX: 82.10
[CPU] (13.12 2.0 1.0 2.0) 0.0 0.0 0.0

```

Storefront Grafiken

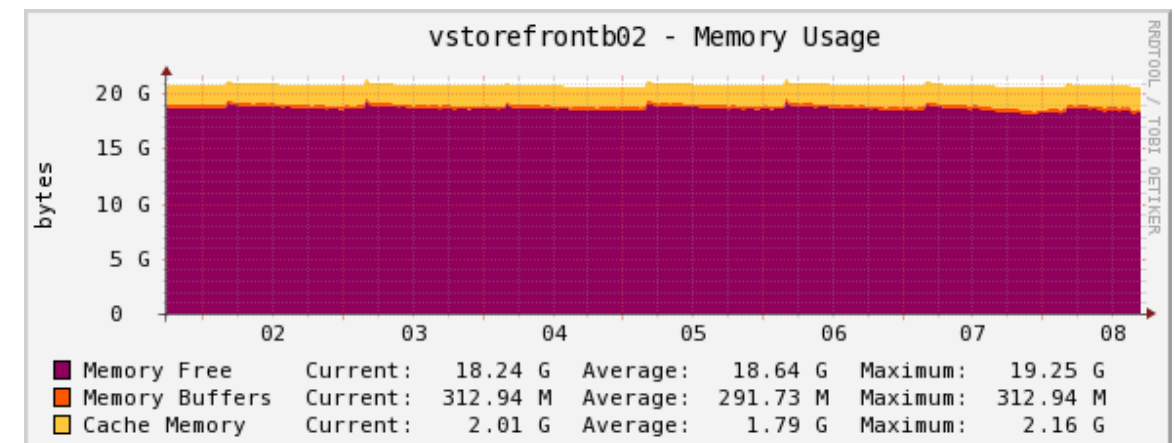
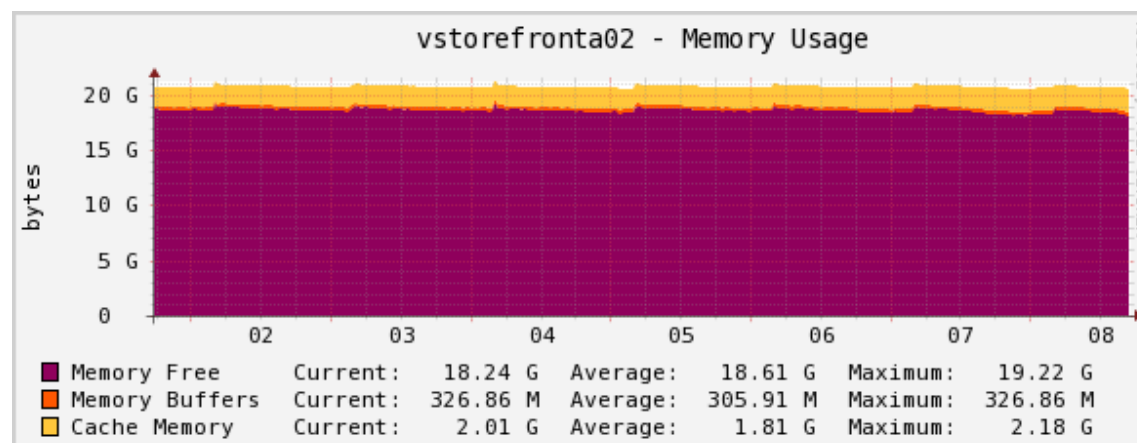
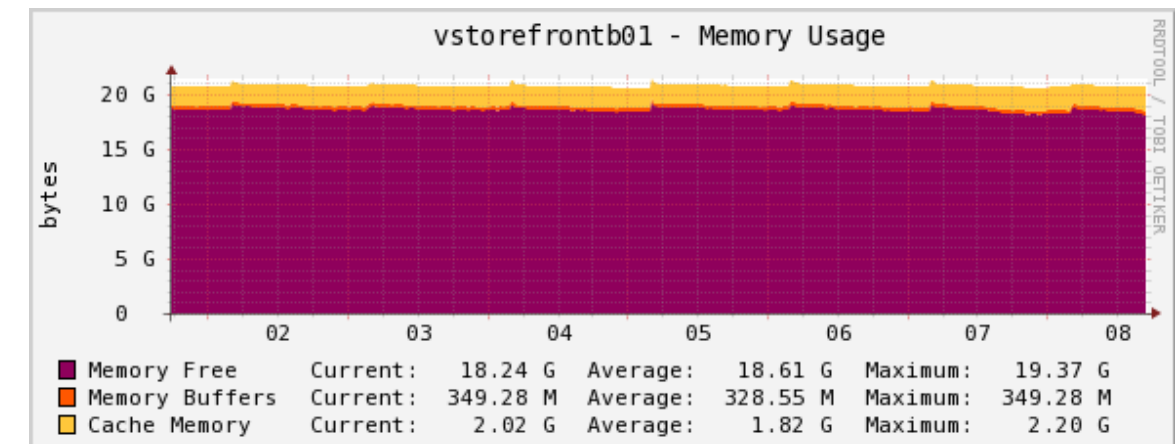
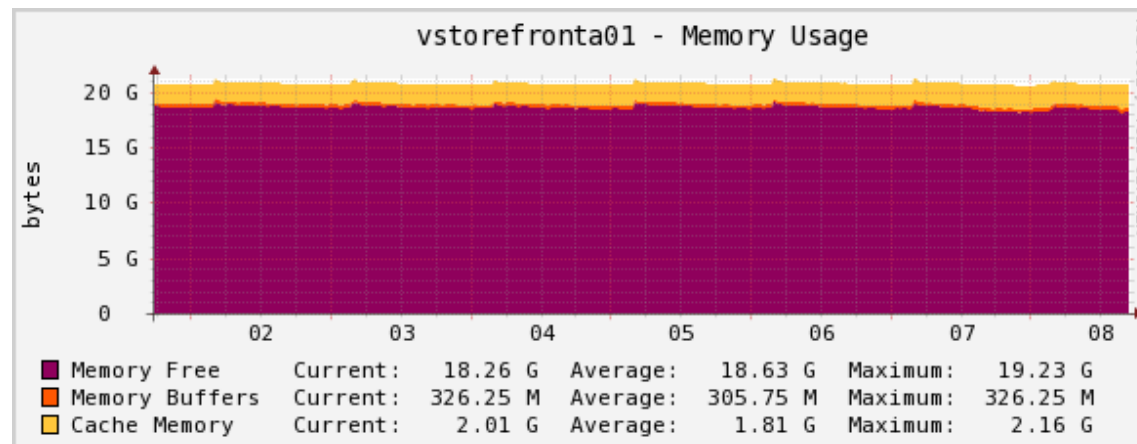
- Load



Der maximale Load unserer vier StoreFronts lag bei 2. Jede Maschine hat 10 Kerne!

Storefront Grafiken

- Memory



Der Speicherverbrauch hat sich trotz der höheren Last nicht merklich verändert
(Hinweis: Die Grafiken zeigen den freien Speicher).

Storefront Grafiken

- PHP FPM Statistiken (hier vstorefronta01)



Erfreulich: Trotz des starken Anstiegs der Requests/s (grün) hat sich die Anzahl der aktiven PHP-FPM-Prozesse kaum erhöht. Das zeigt sehr deutlich, wie schnell das PHP-Processing ist!

FAZIT

Warum war die Eigenentwicklung die richtige Entscheidung für uns?

- Wir wollen kein „me too commerce“
- Wir haben die Chance, mit Innovationen den Markt mitzugestalten
- Wir wollen die volle Kontrolle über unsere Applikation
- Wir haben die richtigen Leute! (nur nicht genug)

FAZIT

Warum war die Eigenentwicklung die richtige Entscheidung für uns?

- Wir wollen kein „me too commerce“
- Wir haben die Chance, mit Innovationen den Markt mitzugestalten
- Wir wollen die volle Kontrolle über unsere Applikation

Dezenter Hinweis: We are hiring!

<http://t3n.de/jobs/job/senior-backend-entwicklerin-fur-php/>

THE ~~SKY~~ TEAM IS THE LIMIT

- Bei der Umsetzung von Features gibt es quasiTM keine Grenzen mehr
- Das Team kennt jede einzelne Codezeile
- Schnelles Team = schnelle Weiterentwicklung
- Langsames Team = langsame Weiterentwicklung



Q&A



FOLLOW US!

@GlobetrotterDEV



Globetrotter.de
Ausrüstung



High Performance E-Commerce