

Webinar zum Thema: Caching Infrastrukturen

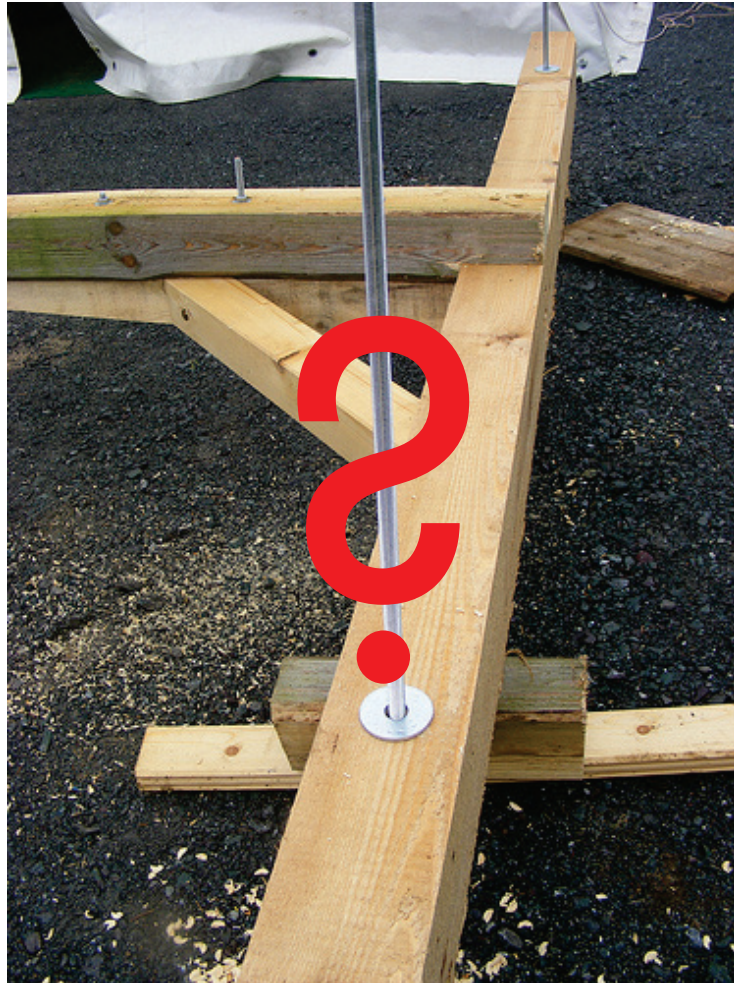
Stefan Staudenmeyer | 20.09.2011



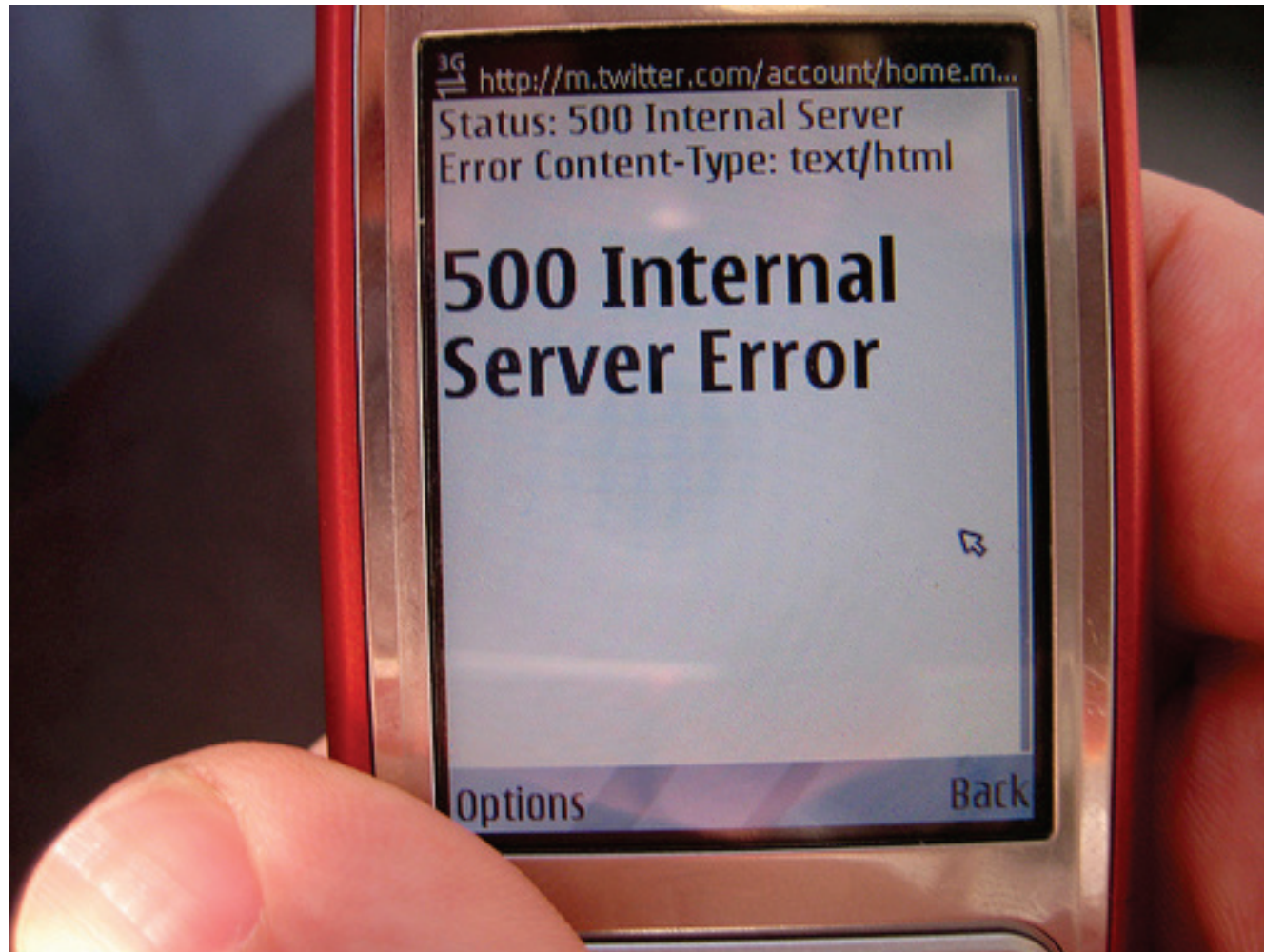
Über Infrastruktur...



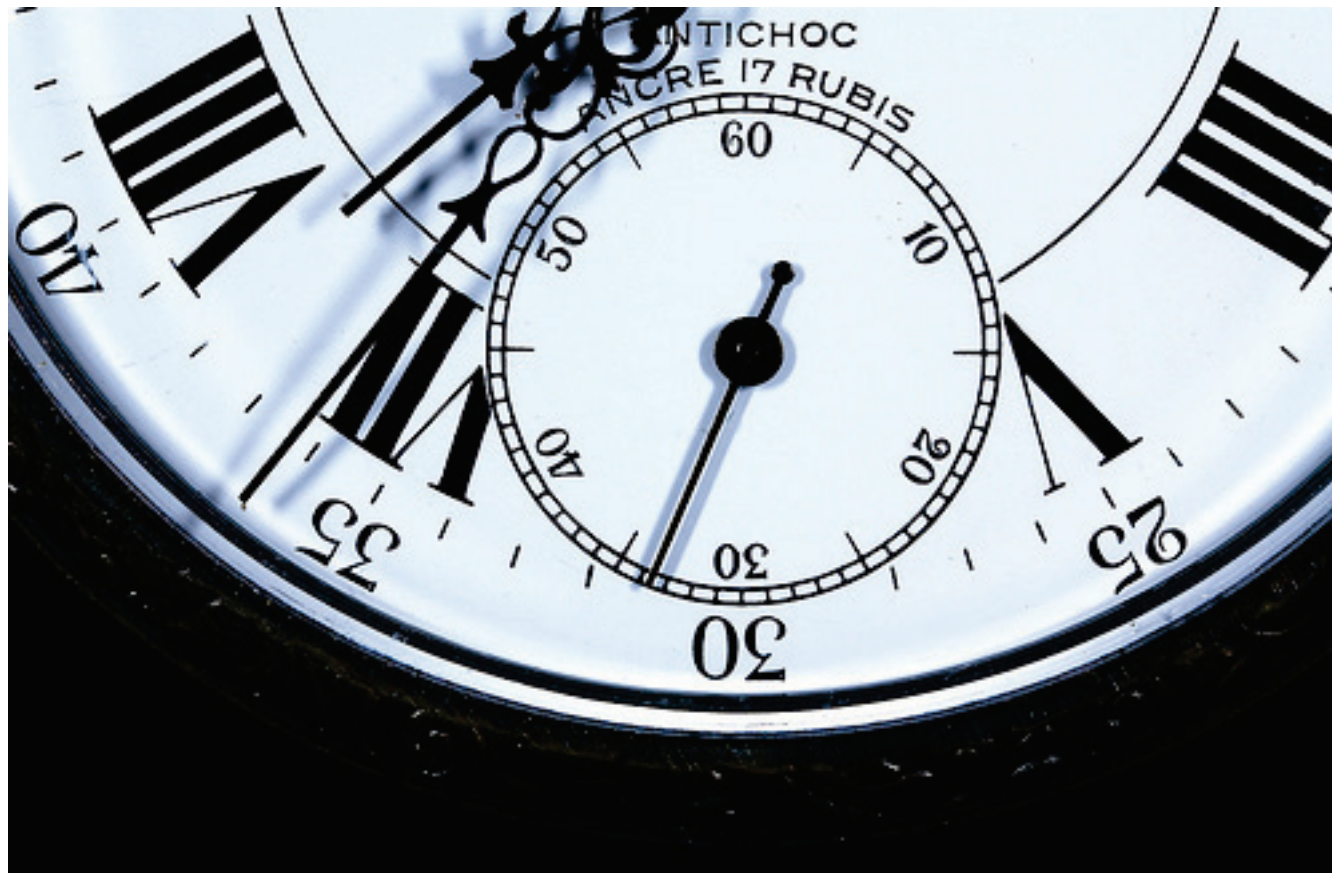
Über Infrastruktur...



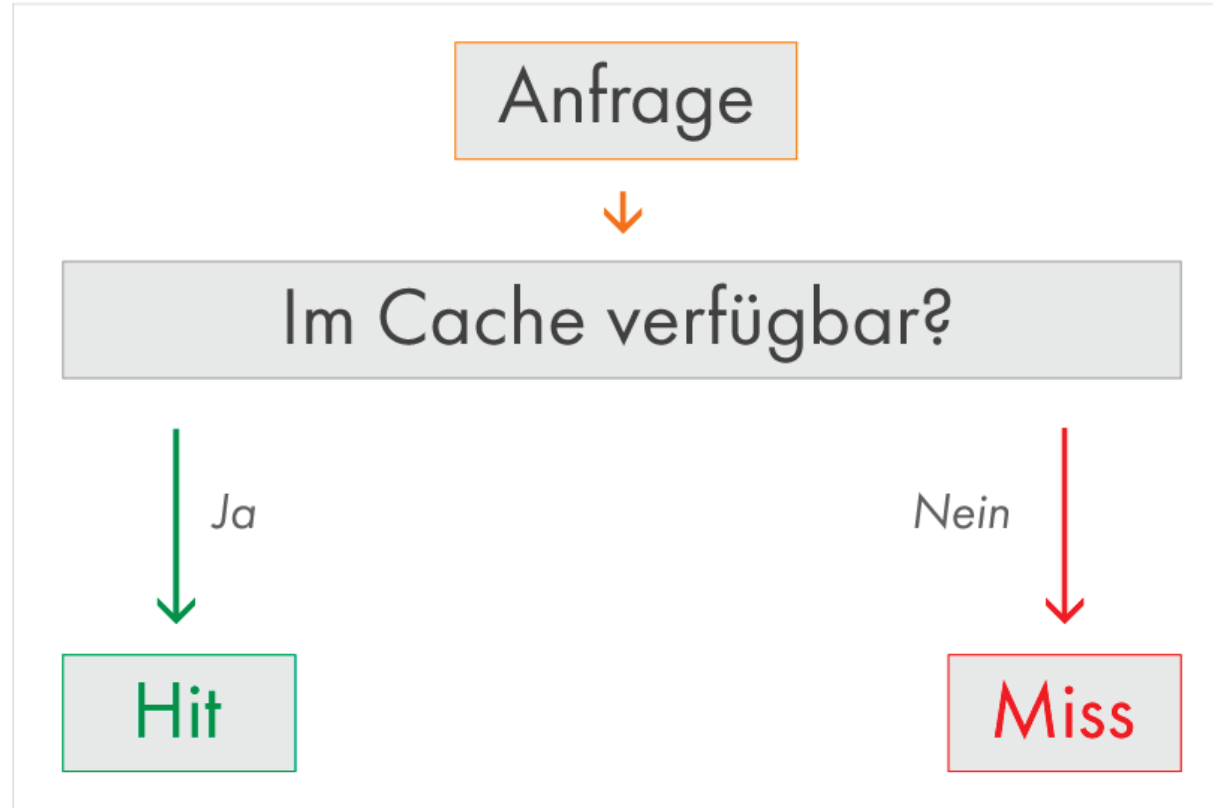
Arbeit zur Wiedervorlage



Don't repeat yourself!



Was geschieht?

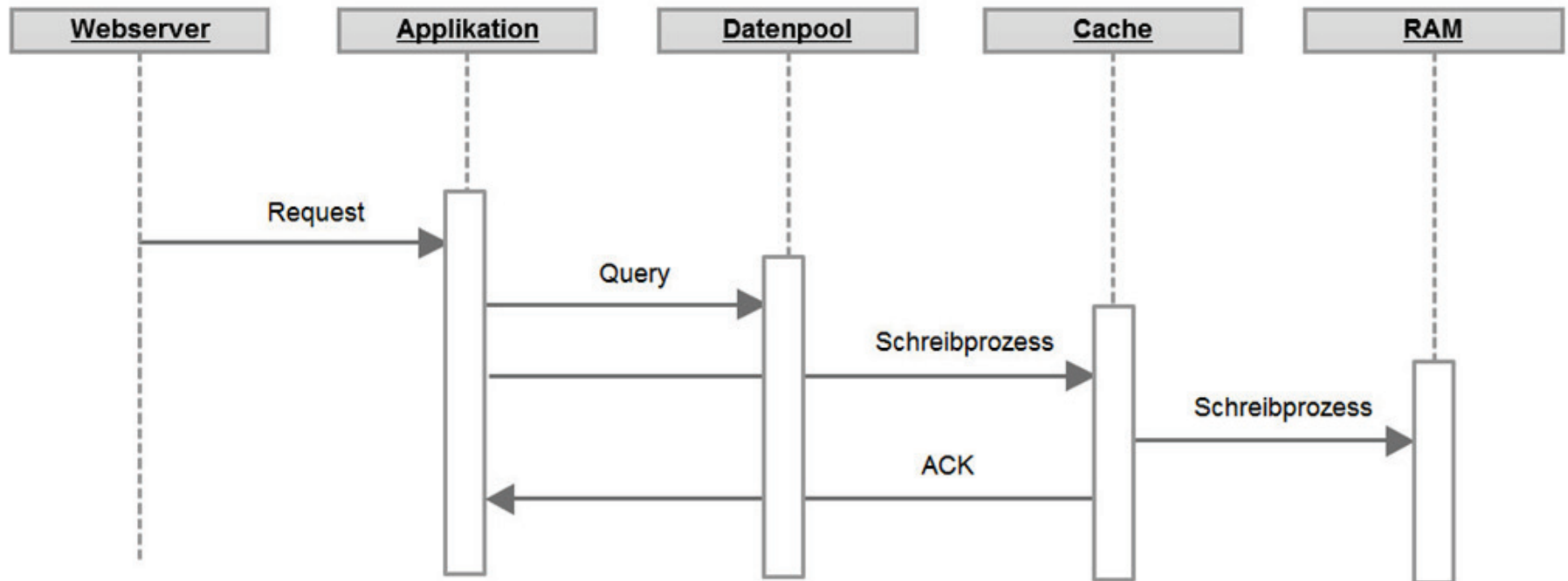


Die Schichten einer Software-Infrastruktur...



Wann werden Daten vom Cache in den Hauptspeicher „geACKt“?

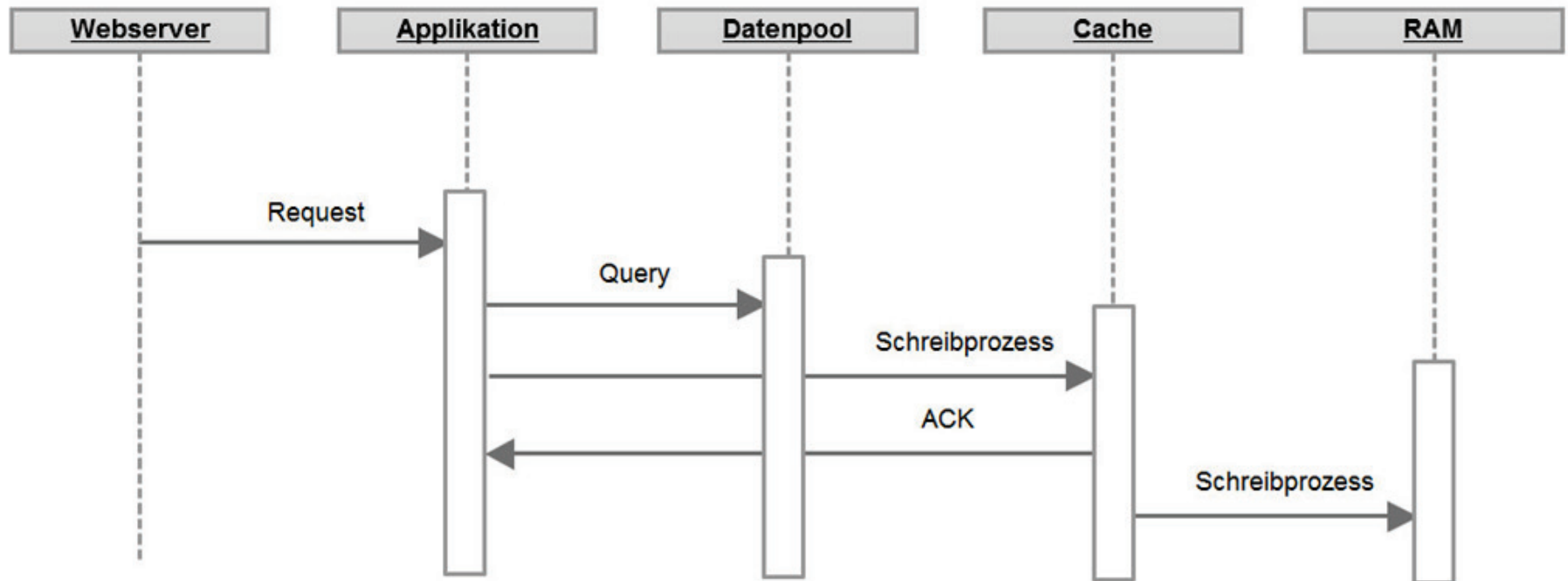
Write through



Ganz hinten ? Write through !

Wann werden Daten vom Cache in den Hauptspeicher „geACKt“?

Write back



Verzögert? Write back!



Welche Daten „invalidiere“ ich
wenn der Cache vollläuft?

Den am wenigsten Frequentierten? Least frequently used (LFU)!

```
mysql> select * from temporarily;
+-----+-----+-----+-----+-----+
| name   | creation          | last_access          | access_count | will_be_deleted |
+-----+-----+-----+-----+-----+
| doerte | 2010-05-12 18:00:15 | 2011-09-05 21:34:00 | 15           | <<              |
| rene   | 2010-10-22 14:00:02 | 2010-01-01 00:00:00 | 37           |                  |
| patrick | 1998-02-08 05:00:35 | 2010-12-24 21:14:00 | 25           |                  |
+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```



Welche Daten „invalidiere“ ich
wenn der Cache vollläuft?

Den als letztes verwendeten? Least recently used (LRU)!

```
mysql> select * from temporarily;
```

name	creation	last_access	access_count	will_be_deleted
doerte	2010-05-12 18:00:15	2011-09-05 21:34:00	15	
rene	2010-10-22 14:00:02	2010-01-01 00:00:00	37	<<
patrick	1998-02-08 05:00:35	2010-12-24 21:14:00	25	

```
3 rows in set (0.00 sec)
```



Welche Daten „invalidiere“ ich
wenn der Cache vollläuft?

Den ältesten Eintrag? First in First out (FIFO)!

```
mysql> select * from temporarily;
```

name	creation	last_access	access_count	will_be_deleted
doerte	2010-05-12 18:00:15	2011-09-05 21:34:00	15	
rene	2010-10-22 14:00:02	2010-01-01 00:00:00	37	
patrick	1998-02-08 05:00:35	2010-12-24 21:14:00	25	<<

```
3 rows in set (0.00 sec)
```



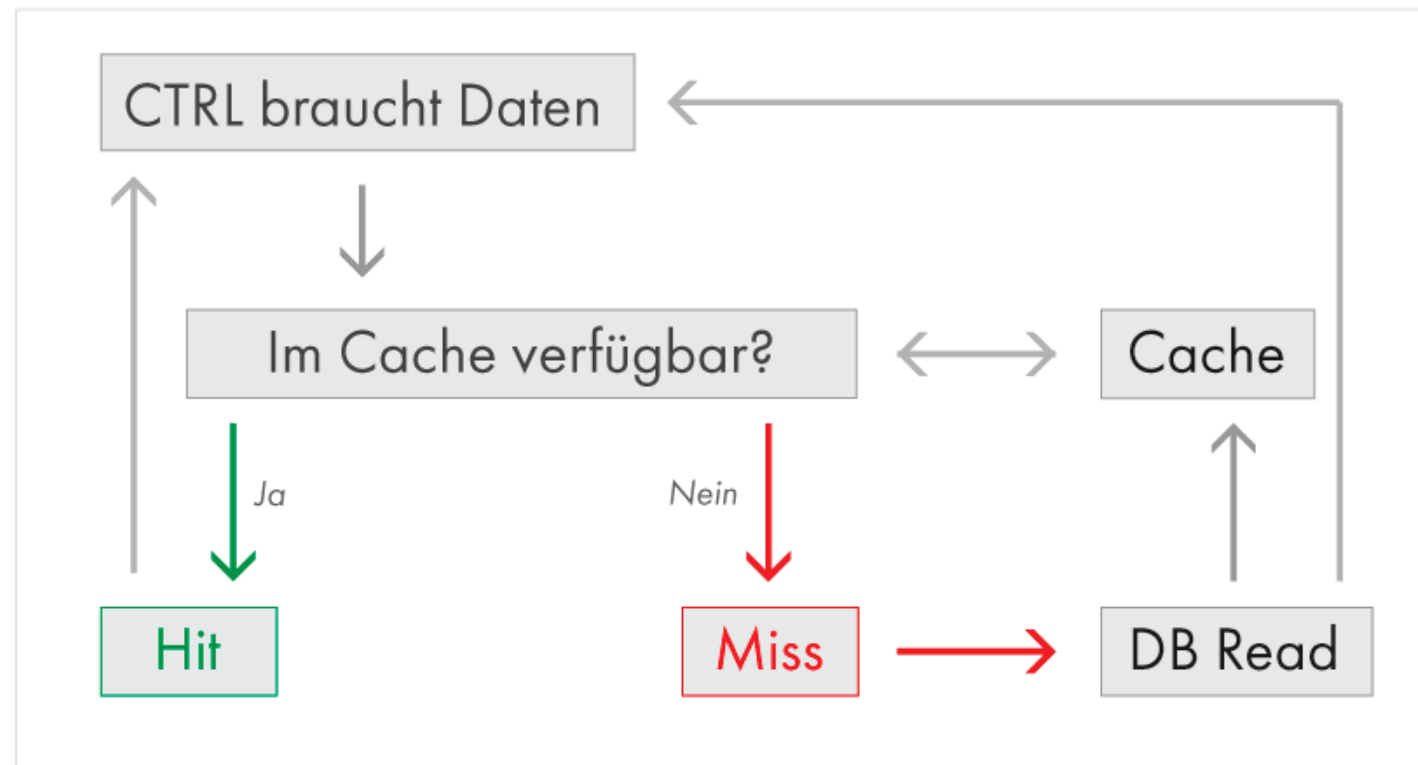

Applikationsgesteuertes Caching?

Wenn es um Models / Datenstrukturen geht?



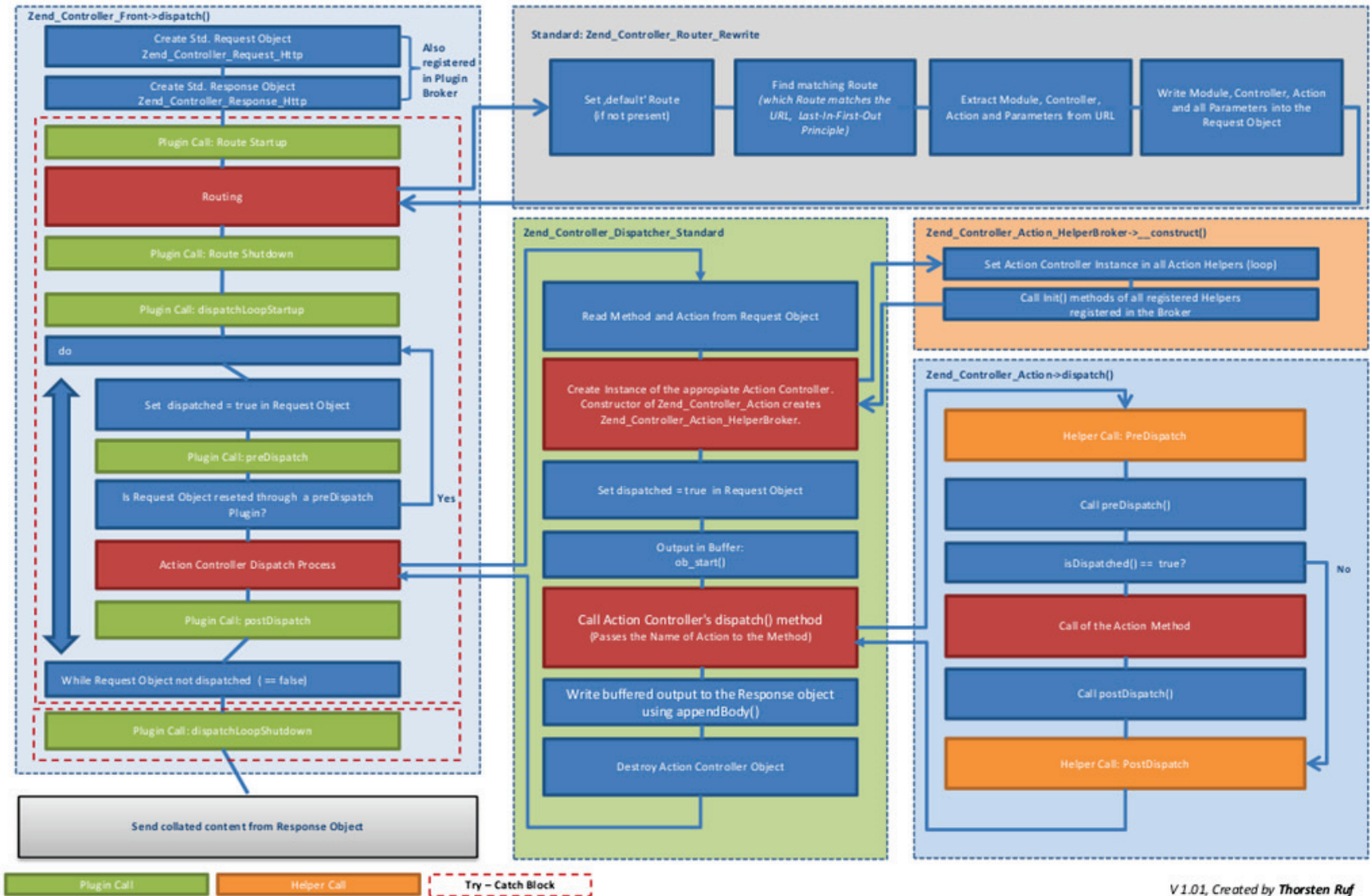
Das Zend Framework lässt uns die Wahl!

Applikationsgesteuertes Caching?



ZF + Caching?

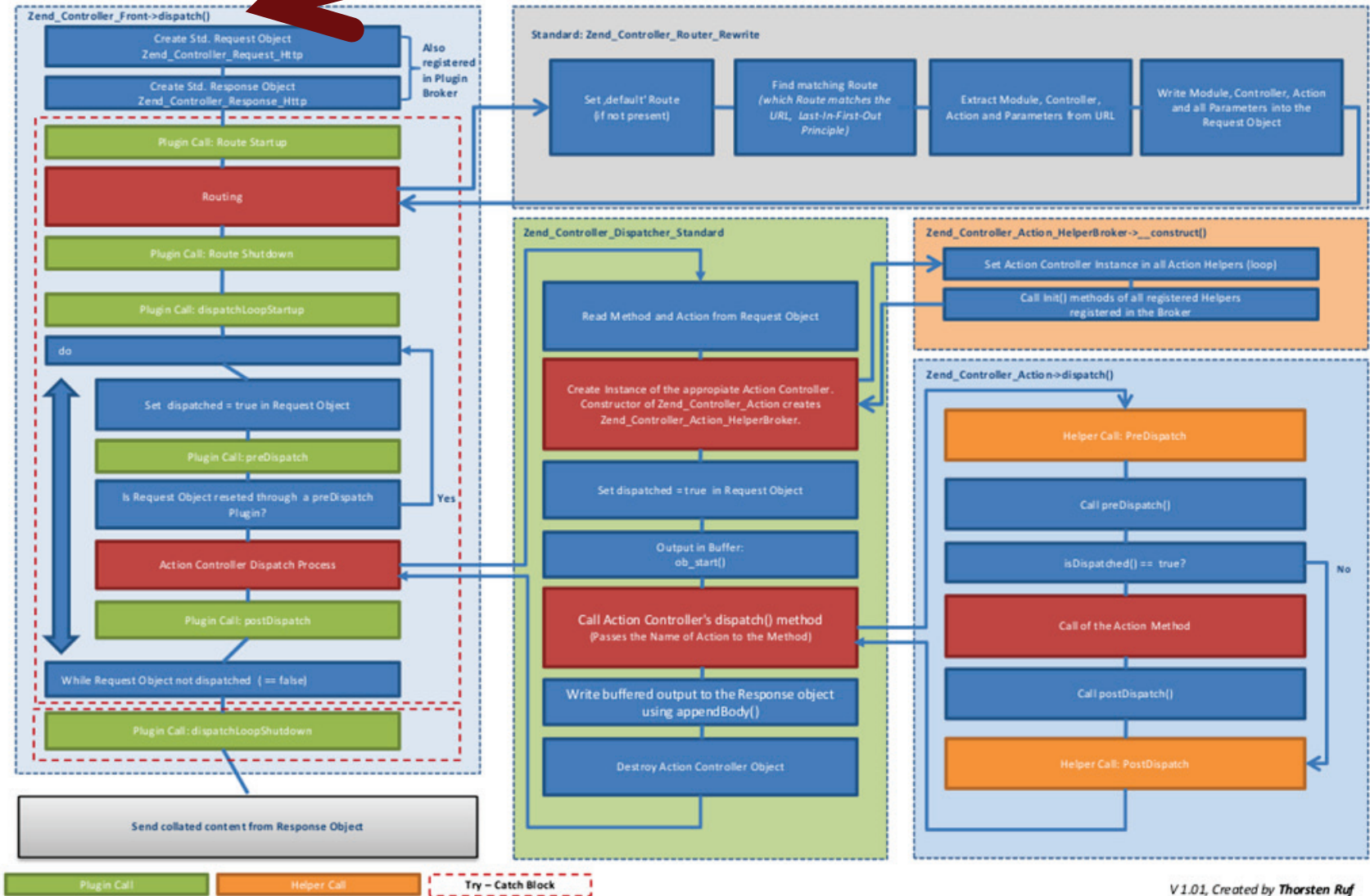
ZEND FRAMEWORK Dispatch Process Overview



ZF + Caching?

Zuvor: Reverse-Proxy & APC

ZEND FRAMEWORK Dispatch Process Overview

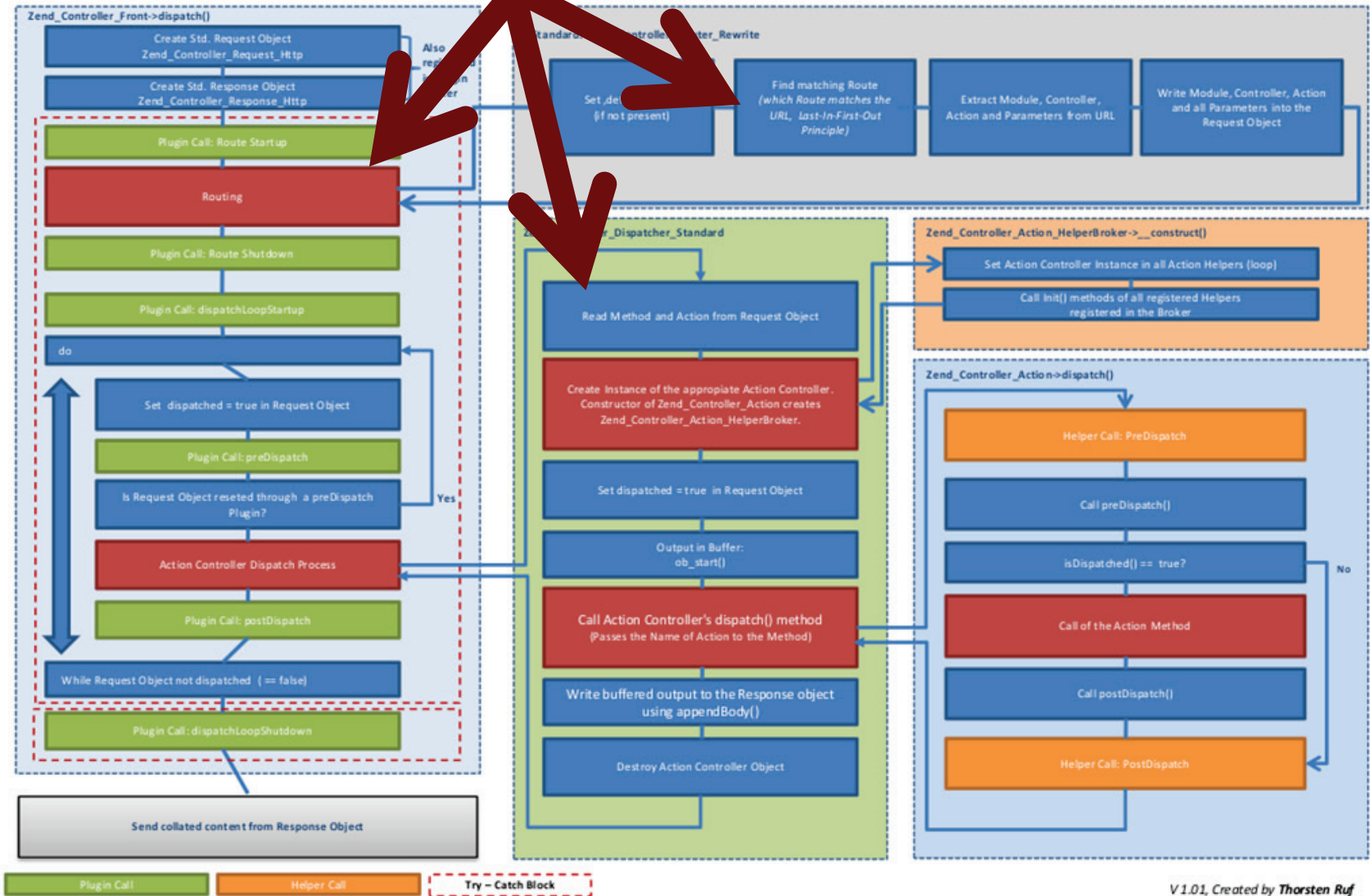


V 1.01, Created by Thorsten Ruf

ZF + Caching?

Routing: Mapping durch gecachte Files

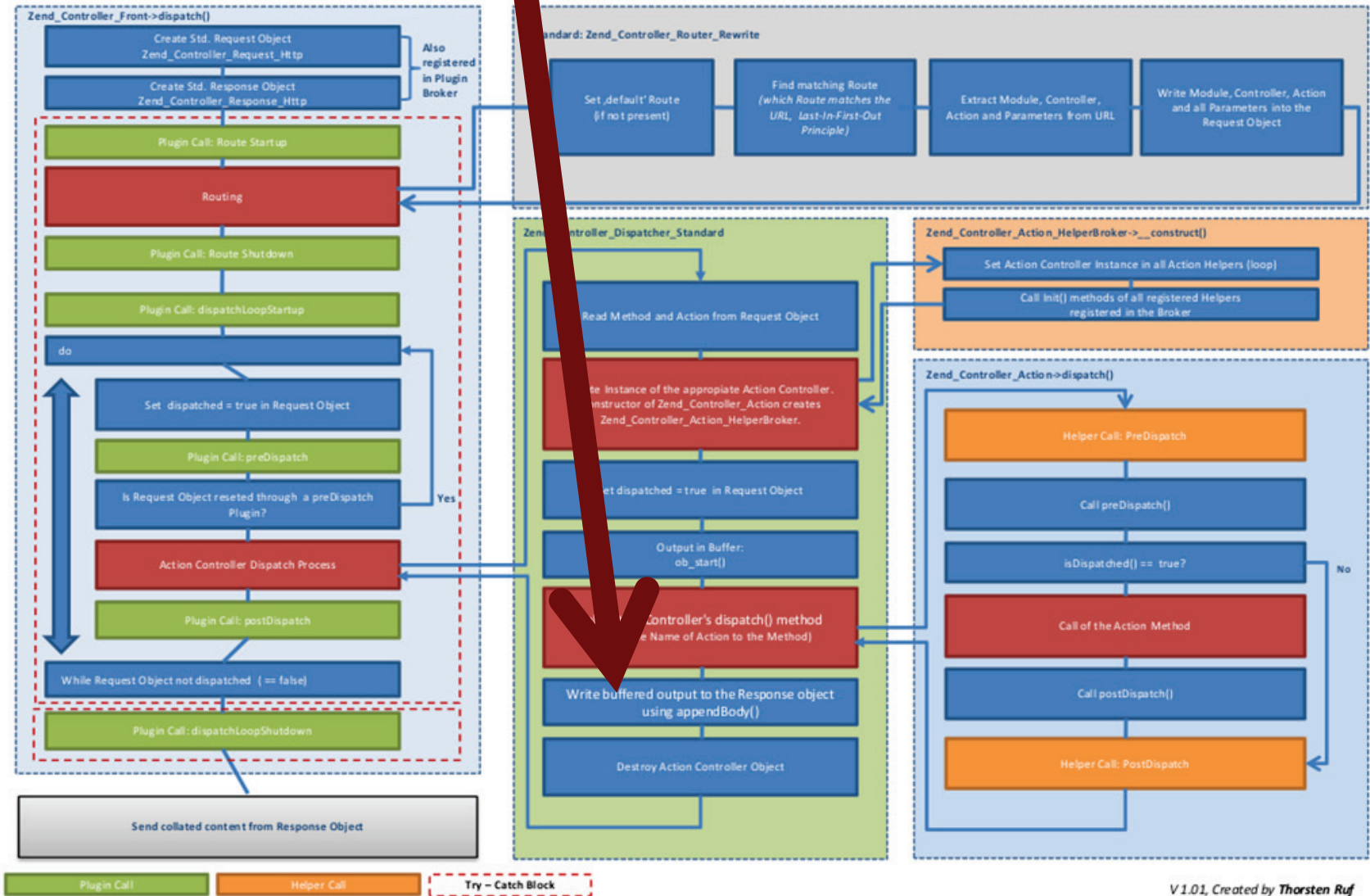
ZEND FRAMEWORK Dispatch Process Overview



ZF + Caching?

Templating: Twig?

ZEND FRAMEWORK Dispatch Process Overview



V 1.01, Created by Thorsten Ruf



ZF + Caching?

- | Models:
 - ORM Cache?



ZF + Caching?

| Models:

- ORM Cache?
- Resultset – Cache per Key/Value Store?



ZF + Caching?

| Models:

- ORM Cache?
- Resultset – Cache per Key/Value Store?
- DBeigenes Cachesystem?



ZF + Caching?

| Models:

- ORM Cache?
- Resultset – Cache per Key/Value Store?
- DBeigenes Cachesystem?
- APC (Opcode – Cache)?



ZF + Caching?

| Models:

- ORM Cache?
- Resultset – Cache per Key/Value Store?
- DBeigenes Cachesystem?
- APC (Opcode – Cache)?

| Reflection Party?

- Parameter Cache
- Objekt Cache
- ...



Netzwerk-Dateisysteme?

| Verteilter Filecache?



Netzwerk-Dateisysteme?

| Verteilter Filecache?

- Reads/Writes?



Netzwerk-Dateisysteme?

| Verteilter Filecache?

- Reads/Writes?
- Locks?



Applikationsgesteuertes Caching?

PAC ?

HMVC?



Applikationsgesteuertes Caching? SOA!

- | Frontcontroller hat gewisse Module zu rendern



Applikationsgesteuertes Caching? SOA!

- | Frontcontroller hat gewisse Module zu rendern
- | Modul-Models fragen Cachepool nach Daten
 - Wenn vorhanden:
 - View rendern



Applikationsgesteuertes Caching? SOA!

- | Frontcontroller hat gewisse Module zu rendern
- | Modul-Models fragen Cachepool nach Daten
 - Wenn vorhanden:
 - View rendern
 - Wenn nicht vorhanden:
 - DB abfragen
 - Das Ergebnis in Memcache setzen
 - View rendern



Doch auch SOA bietet bottlenecks bzgl. Caching!

- | Physikalischer Abstand zwischen voneinander abhängigen Nodes



Doch auch SOA bietet bottlenecks bzgl. Caching!

- | Physikalischer Abstand zwischen voneinander abhängigen Nodes
 - Zeit der Requests untereinander



Doch auch SOA bietet bottlenecks bzgl. Caching!

- | Physikalischer Abstand zwischen voneinander abhängigen Nodes
 - Zeit der Requests untereinander
 - Failover bei verteilten Caches



Doch auch SOA bietet bottlenecks bzgl. Caching!

- | Physikalischer Abstand zwischen voneinander abhängigen Nodes
 - Zeit der Requests untereinander
 - Failover bei verteilten Caches
- | Verteilte Front- / Backend- / Servicenodes



Doch auch SOA bietet bottlenecks bzgl. Caching!

- | Physikalischer Abstand zwischen voneinander abhängigen Nodes
 - Zeit der Requests untereinander
 - Failover bei verteilten Caches
- | Verteilte Front- / Backend- / Servicenodes
 - Session – Transport (ReST-Services)



Doch auch SOA bietet bottlenecks bzgl. Caching!

- | Physikalischer Abstand zwischen voneinander abhängigen Nodes
 - Zeit der Requests untereinander
 - Failover bei verteilten Caches
- | Verteilte Front- / Backend- / Servicenodes
 - Session – Transport (ReST-Services)
 - Zyklische Abhängigkeit von Daten



Memcached – Ein Key/Value-Store

Man hinterlege Daten (values) hinter einer eindeutigen Bezeichnung (key) im Zwischenspeicher...



Varnish – Ein Reverse-Proxy

Man hinterlege Inhalte vor dem Webserver,
um ihn für statische Inhalte umgehen zu können.



HTTP – Caching?

- | Idempotente Requests können z.B. vom Browser gecached werden!



HTTP – Caching?

- | Idempotente Requests können z.B. vom Browser gecached werden!
- | Browser machen ihr eigenes Ding! Beispiel:
Die Voransicht in der Blank-Seite von Google Chrome...



„Performance“ >> Responsiveness

Der User loggt sich ein ...

„Performance“ >> Responsiveness

Der User loggt sich ein ...

... hat er eine neue Nachricht

...

„Performance“ >> Responsiveness

Der User loggt sich ein ...

... hat er eine neue Nachricht

...

... will er sie höchstwahrscheinlich
lesen!

Responsiveness: You're doing it right!



Responsiveness: You're doing it right!

**SAT1SPIELE**

SAT1.DE

SPIELE ▶ MEIN SAT1 SPIELE ▶

COLOUR YOUR SITE ●●●●●●●●

Connect

LOGIN

ALLE SPIELE

☐ DIE BELIEBTESTEN

☐ VON A-Z

SPIELE NACH GENRE

☐ ABENTEUER (8)

☐ ACTION (6)

☐ ARCADE (6)

☐ GESCHICKLICHKEIT (45)

☐ KARTEN (5)

☒ LOGIK (86)

☐ MANAGEMENT (11)

☐ QUIZ (6)

☐ ROLLENSPIEL (0)

☐ SIMULATION (1)

☐ SPORT (0)

☐ STRATEGIE (17)

☐ TV-SERIEN (2)

☐ YETISPORTS (0)

SPIELE NACH ART

☐ GELD-TURNIERE (41)

☐ BROWSER GAMES (1)

☐ DOWNLOAD GAMES (10)

☐ FLASH GAMES (11)

☐ IM HANDEL (0)

☐ WEBPUZZLES (23)



SAT.1 KLASSIKER
online sehen



HOME ▶ AUSWAHL

LOGIK (86)

ALPHABETISCH ● NEU ●

**MINER SPEED**
LOGIK
• GESCHICKLICHKEIT
ONLINE SPIELEN

**STARSWEeper**
LOGIK • STRATEGIE
ONLINE SPIELEN

**CIRCUS SURPRISE**
LOGIK ...
ONLINE SPIELEN

**CANDY CRUSH**
LOGIK
ONLINE SPIELEN

**ALAMANDI**
LOGIK
• GESCHICKLICHKEIT ...
GRATIS DOWNLOAD

**AIRPORT MANIA**
LOGIK • MANAGEMENT
• ARCADE
GRATIS DOWNLOAD

**MAHJONGG DIMENSIONS DELUXE**
GRATIS DOWNLOAD

**AGATHA CHRISTIE BUNDLE 3-1**
GRATIS DOWNLOAD

**EIN TRAUM IN GELB**
LOGIK ...
ONLINE SPIELEN

**DAS ORIGINAL**
GESCHICKLICHKEIT
• LOGIK
ONLINE SPIELEN

**BLÜTENPRACHT**
GESCHICKLICHKEIT
• LOGIK
ONLINE SPIELEN

**BASEBALL**
LOGIK
• GESCHICKLICHKEIT
ONLINE SPIELEN

MEIN SAT1 SPIELE

DEINE VORTEILE

- + Spiele gegen echte Gegner
- + Gewinne tolle Preise
- + Platziere Dich in den Highscores
- + Gestalte Deinen eigenen Avatar
- + Sammle begehrte Awards

JETZT REGISTRIEREN



Responsiveness: You're doing it right!



super complications during typing



Suche

Ungefähr 2.520.000.000 Ergebnisse (0,11 Sekunden)

Google.com in English [Erweiterte Suche](#)



Alles



Bilder



Videos



News



Shopping



Mehr

Fürth

[Standort ändern](#)

Das Web

[Seiten auf Deutsch](#)

[Seiten aus Deutschland](#)

[Übersetzte Seiten](#)

Alle

[Neueste](#)

[Letzte 24 Std.](#)

[Letzte 2 Tage](#)

[Letzte Woche](#)

[Letzter Monat](#)

[Letztes Jahr](#)

[Zeitraum festlegen...](#)

[Mehr Optionen](#)

[SUPER 2011 - Download - CHIP Online](#)

★★★★★ 24450 Erfahrungsberichte

SUPER 2011 Build 48 Englisch Free-Download kostenlos. SUPER ist ein kostenloses Konvertierungs-Programm für nahezu alle Audio- und Videoformate.

[www.chip.de](#) › [Downloads](#) › [Audio & Video](#) › [Converter](#) - [Im Cache](#)

[SUPER - Download](#)

SUPER, Download virengeprüft und kostenlos. SUPER 2011.47: Vielseitiger Video- und Audio-Konverter. SUPER bringt eine Ein-Klick-Lösung rund um alle gängigen ...

[super.softonic.de/](#) - [Im Cache](#) - [Ähnliche Seiten](#)

[SUPER ©](#) - [\[Diese Seite übersetzen \]](#)

SUPER © "The Player" surpasses any known player by supporting just ... SUPER © "The Encoder" is the fastest and simplest tool to encode full ...

[www.erightsoft.com/SUPER.html](#) - [Im Cache](#) - [Ähnliche Seiten](#)

[SUPER zum Download auf Freeware.de](#)

26. Apr. 2011 ... SUPER kostenlos zum Download: Freeware.de bietet die Software SUPER vom Hersteller Erightsoft in Version 2011 build 48.

[www.freeware.de](#) › [...](#) › [CD Ripper](#) - [Im Cache](#) - [Ähnliche Seiten](#)

[SUPER 2011 build 48 Download - Video-/Audio-Tool - PC-WELT](#)

27. Apr. 2011 ... Die Freeware SUPER 2011 konvertiert Ihnen Ihre Video- und Audiodateien in alle gängigen Formate. Sie können Ihre Videodateien für Pocket-PCs ...

[www.pcwelt.de](#) › [...](#) › [Grafik & Video](#) - [Im Cache](#) - [Ähnliche Seiten](#)

[SUPER 2011 Build 48 in Ripper & Encoder - Audio & Video - Windows ...](#)

26. Apr. 2011 ... Konvertiert nahezu alle Video- und Audioformate SUPER aus der Kategorie Ripper & Encoder von eRightSoft ist ein Freeware-Programm in deutsch ...

[www.zdnet.de](#) › [...](#) › [Ripper & Encoder](#) - [Im Cache](#) - [Ähnliche Seiten](#)



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Domain.tld/populaere_anlaufstelle fragt nach einem Deploy den Cache nach Daten



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Domain.tld/populaere_anlaufstelle fragt nach einem Deploy den Cache nach Daten
- | Cache hat keine Daten – Fragt die DB



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Domain.tld/populaere_anlaufstelle fragt nach einem Deploy den Cache nach Daten
- | Cache hat keine Daten – fragt die DB
- | DB hat die Daten, gibt sie an den Cache und die Applikation weiter



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Domain.tld/populaere_anlaufstelle fragt nach einem Deploy den Cache nach Daten
- | Cache hat keine Daten – fragt die DB
- | DB hat die Daten, gibt sie an den Cache und die Applikation weiter
- | Hunderte Requests/Sek. auf verschiedene Bereiche



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Domain.tld/populaere_anlaufstelle fragt nach einem Deploy den Cache nach Daten
- | Cache hat keine Daten – fragt die DB
- | DB hat die Daten, gibt sie an den Cache und die Applikation weiter
- | Hunderte Requests/Sek. auf verschiedene Bereiche
- | Applikation hängt



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Nach einem Deploy wärmt ein Script den Cache auf und lädt Daten stark frequentierter Seiten vor



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Nach einem Deploy wärmt ein Script den Cache auf und lädt Daten stark frequentierter Seiten vor
- | Die ersten User kommen auf `Domain.tld/populaere_anlaufstelle(N! Plural!)`



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Nach einem Deploy wärmt ein Script den Cache auf und lädt Daten stark frequentierter Seiten vor
- | Die ersten User kommen auf `Domain.tld/populaere_anlaufstelle(N! Plural!)`
- | Die Caches werden nach Daten befragt und die ersten Schreiboperationen auf den DBs finden statt



Verlass dich nicht drauf... Oder: Warm ´em up...

- | Nach einem Deploy wärmt ein Script den Cache auf und lädt Daten stark frequentierter Seiten vor
- | Die ersten User kommen auf `Domain.tld/populaere_anlaufstelle(N! Plural!)`
- | Die Caches werden nach Daten befragt und die ersten Schreiboperationen auf den DBs finden statt
- | Die Applikation hängt nicht



Caching und Hardware sind keine Golden Hammer...

| ... und erschlagen keine Performanceprobleme bei:



Caching und Hardware sind keine Golden Hammer...

- | ... und erschlagen keine Performanceprobleme bei:
 - Kaputten Datenbankstrukturen



Caching und Hardware sind keine Golden Hammer...

- | ... und erschlagen keine Performanceprobleme bei:
 - Kaputten Datenbankstrukturen
 - Kaputter Software / fehlenden Features



Caching und Hardware sind keine Golden Hammer...

- | ... und erschlagen keine Performanceprobleme bei:
 - Kaputten Datenbankstrukturen
 - Kaputter Software / fehlenden Features
 - Skalierungsproblemen generell



Caching und Hardware sind keine Golden Hammer...

- | ... und erschlagen keine Performanceprobleme bei:
 - Kaputten Datenbankstrukturen
 - Kaputter Software / fehlenden Features
 - Skalierungsproblemen generell
 - Fehlender Architekturplanung



Schlussendlich zum Thema Architektur / Infrastruktur...

| Es gibt keine Generallösung für Ihre Applikation!



Schlussendlich zum Thema Architektur / Infrastruktur...

- | Es gibt keine Generallösung für Ihre Applikation!
- | Es gibt nicht nur einen Idealzustand!



Schlussendlich zum Thema Architektur / Infrastruktur...

- | Es gibt keine Generallösung für Ihre Applikation!
- | Es gibt nicht nur einen Idealzustand!
- | Sie kennen Ihre Applikation am besten!



Schlussendlich zum Thema Architektur / Infrastruktur...

- | Es gibt keine Generallösung für Ihre Applikation!
- | Es gibt nicht nur einen Idealzustand!
- | Sie kennen Ihre Applikation am besten!
- | Seien Sie agil!



Schlussendlich zum Thema Architektur / Infrastruktur...

- | Es gibt keine Generallösung für Ihre Applikation!
- | Es gibt nicht nur einen Idealzustand!
- | Sie kennen Ihre Applikation am besten!
- | Seien Sie agil!
- | Kommunikation ist essentiell, jedoch zu viele Köche verderben den Brei!



Schlussendlich zum Thema Architektur / Infrastruktur...

- | Es gibt keine Generallösung für Ihre Applikation!
- | Es gibt nicht nur einen Idealzustand!
- | Sie kennen Ihre Applikation am besten!
- | Seien Sie agil!
- | Kommunikation ist essentiell, jedoch zu viele Köche verderben den Brei!
- | Analysieren Sie den Ist-Zustand bevor Sie Caching einsetzen /
umtauschen / entfernen / ... !!!!!!!!!!!

Einen schönen Tag wünscht:



Referent

Stefan Staudenmeyer

Stefan.Staudenmeyer@mayflower.de

@doerteDev

Mayflower GmbH

Pleichertorstr. 2

97070 Würzburg

Quellen:

| Lektüre:

- „Don't repeat yourself“
Andrew Hunt and David Thomas, 1999, Addison Wesley, ISBN 0-201-61622-X.

| FlickrR

- http://www.flickr.com/photos/uwe_schubert/3661784433
- <http://www.flickr.com/photos/rvoegtli/5688343678>
- <http://www.flickr.com/photos/angusf/2464076104>
- <http://www.flickr.com/photos/7960798@N07/2379621715>

| PHP-Professionals

- <http://www.php-professionals.com/wp-content/uploads/2009/01/zf-flow-diagram1.png>