

# PHP Usergroup Berlin



6. April 2010

Ein Leben mit und ohne Magento

# Wer wir sind ...



**Volker Pilz (Zalando)**

[xing.com/profile/Volker\\_Pilz](http://xing.com/profile/Volker_Pilz)

**Senior Software Developer**

**Davor: VZ Netzwerke Ltd.**



**Daniel Nowak (Rocket Internet)**

[xing.com/profile/Daniel\\_Nowak](http://xing.com/profile/Daniel_Nowak)

**Senior Software Developer**

**Davor: Captain Ad, Infopark**

# Zalando

- Gegründet im Oktober 2008
- Sitz in Berlin, Zinnowitzer Straße 1
- ca. 10.000 verschiedene Artikel
- ca. 160 Mitarbeiter
- Davon ca. 23 in der Software-Entwicklung
- Zalando Shop basierend auf Magento Community Edition, Zalando Lounge basierend auf Magento Enterprise Edition

# Grundproblem



# Typical Magento Scaling

- Zu Beginn meist 1 Server: WS+DBS auf einer Maschine
- Dann 1 WS + 1 DBS
- WS++
- DBS: Master-Slave Replikation
- Event. Server für: Memcache, Static, Squid, Mail, Administration

# Aktuelles Zalando Setup

- 14 Webserver
- 1 DB-Master + 7 DB-Slaves
- 4 Memcache Server
- 2 Service Server
- 3 Squid Server
- 2 Solr/Lucene Server
- 2 Admin Server

-----  
 $\sum$  35 Server

# Herausforderung

Scaling from this ...



... to this



# Grundlegendes über Magento

- Entwickelt von Varien Inc. (L.A.)
- Version 1.0 vom 31.3.2008
- Aktuell: Version 1.4.x (seit Februar 2010)
- Best New Open Source Project 2008
- [magentocommerce.com](http://magentocommerce.com)
- Ca. 1.500 Extensions auf Magento Connect



# Vorteile von Magento

- Aktuelle State-of-the-art Features
- Open Source Software
- Starke Community - CAB (Community Advisory Board), Meet Magento, Foren usw.
- Mächtiges Backend
- Flexibilität & Erweiterbarkeit
- Extensions mit Magento Connect
- Zend Framework

# Nachteile von Magento

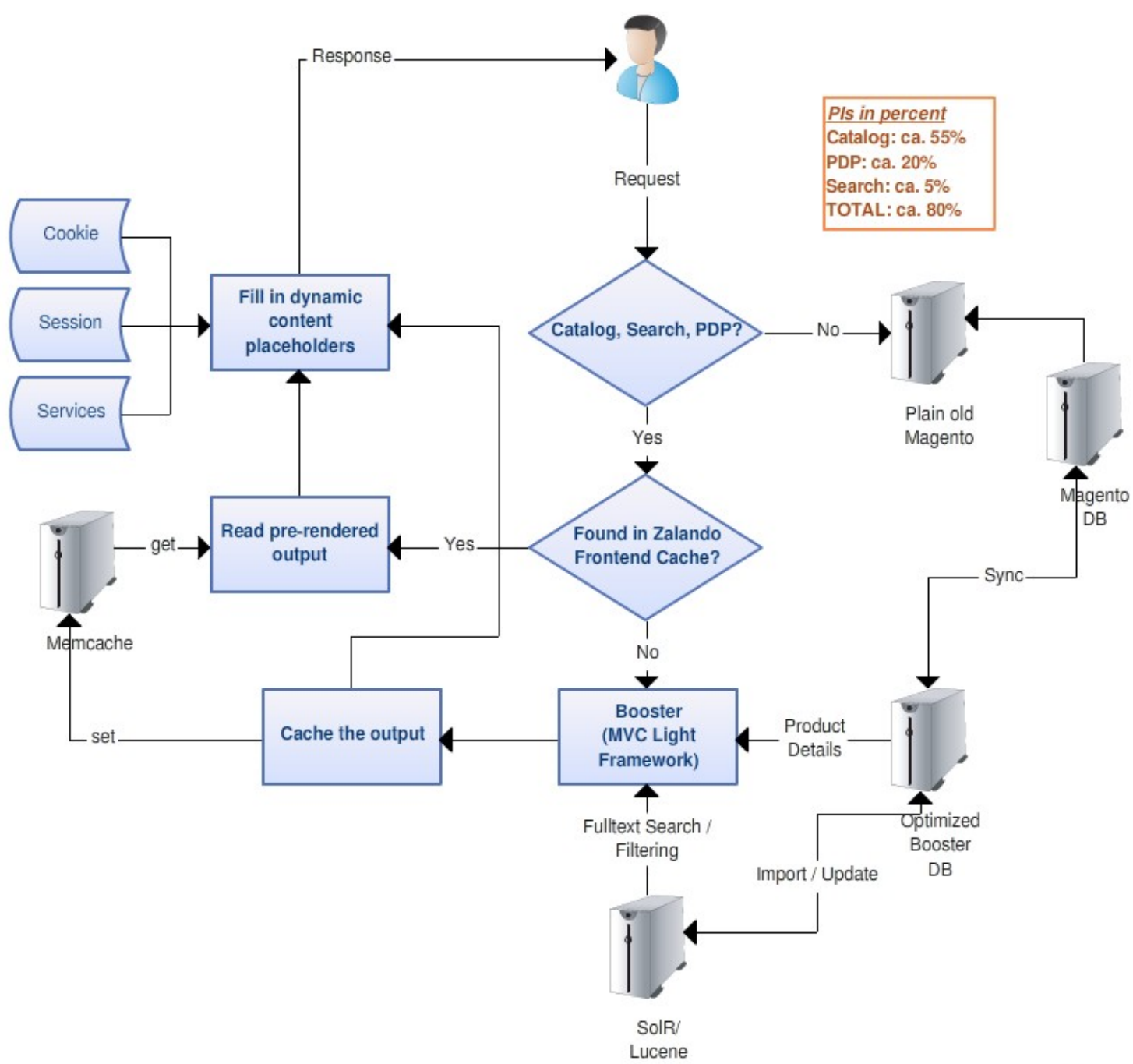
- Keine UnitTests (auch nicht bei wichtigen Komponenten)
- Schlechte Dokumentation (dementsprechend flache Lernkurve für Neueinsteiger)
- Hardwarehungrig und teilweise schlechte Performance
- Teure Enterprise-Version mit nützlichen Features

# Magento Performance

- EAV (Entity-Attribute-Value) Prinzip ist langsam  
=> Gegenmaßnahme: Flat Tables
- Exzessive Instanziierung von Objekten
- Teilweise verworrene Programmabläufe und viel „Magic“
- Vor allem bei Admin-Operationen wird lesend auf den Master zugegriffen
- Filterung/Suche generiert komplexe Queries

# Projekt: Booster

- Ziel: Schnelle Antwortzeiten bei gleichzeitiger Reduzierung der Last auf den Web- und Datenbank-Servern
- Umsetzungszeit Phase 1 (ohne Volltext-Suche): 2 Wochen
- Umsetzungszeit Phase 2 (mit Volltext-Suche): 1 Woche
- 4 Software-Entwickler



[online diagramming & design] [creately.com](http://creately.com)

# Cache Marker

- Post-Processing mit Markern
- preg\_replace\_callback
- Datenbasis: Session, Cookies usw.

The screenshot shows the Zalando website interface for a Converse CHUCKS sneaker. At the top, there are navigation links and a search bar. The breadcrumb trail is highlighted with a red box: "Startseite > Katalog > CHUCKS - Sneaker - vivid blue". Navigation links for "vorheriger Artikel" and "nächster Artikel" are also highlighted with a red box. The main product image shows a blue sneaker. To the right, the price is 64,95 € and there are buttons for "In den Warenkorb" and "Weiterempfehlen". A sidebar on the right titled "Das könnte Ihnen auch gefallen" shows recommended products, highlighted with a blue box. Below the main product, there is a "Zuletzt angeschaut" section with four items, highlighted with a red box. The footer contains a list of brands and a newsletter sign-up form.



# Warum Solr/Lucene?

- Performanz
- Facetten-Suche
- Sprachunabhängigkeit
- Business-erprobt in großen Szenarien
- Fähigkeit zur Replikation / Skalierbarkeit
- OpenSource

# Herausforderungen

1. Import der Daten
2. Anbindung an PHP
3. Loadbalancing / Failover
4. Zeitnahe Updates
5. Facetten
6. Volltext-Suche



# 1. Import der Daten

- Daten-Chaos in Magento
  - Sortierung
  - Produktanzeige
  - Redundante und fehlerhafte Daten
- Lösung: Verwendung des magento-eigenen Flat-Table-Cache

## 2. Anbindung an PHP

- solr-php-client  
<http://code.google.com/p/solr-php-client/>
- Eigener Service-Wrapper
- Alternative:  
PECL-Extension  
<http://pecl.php.net/package/solr>
- JSON als Austauschformat (nicht XML oder serialisierte PHP-Objekte)

# 3. Loadbalancing / Failover

- 2 Solr-Server (1x Master, 1x Slave)
- Replikation (20 sec)
- Failover und Loadbalancing übernimmt PHP-Client
- Tipp 1: Schema sollte von Anfang an mit repliziert werden
- Tipp 2: rand-Funktion durch mt\_rand ersetzen im Balancer

# 4. Zeitnahe Updates

- Minütliche inkrementelle Updates
- Relevant für Warenbestand
- Events von Magento lösen Eintrag in Update-Queue aus
- Stündlicher Import der Gesamtdaten
- Solr schreibt beim Update den kompletten Eintrag für ein Produkt neu

# 5. Facetten

- Solr-Standard-Feature
- Super-performant
- ... `&facet=true&facet.field=color`

# 6. Volltext-Suche

- Suggestions
- Gewichtung der einzelnen Felder zueinander (Farbe, Saison, Name, Beschreibung, ...)
- Vertipper ( „Marc o'Polo“)
- Sortierung der Suchergebnisse

# Solr bei Zalando

- ca. 16 MB Indexgröße zu 15-16 GB Magento-Datenbank
- ca. 10.000 Produkte
- 17 Felder
- 8 Facetten
- 8 Felder für Freitext-Suche
- Index-Aufbau: < 4 sec
- Durchschnittliche Antwortzeit: 20 ms

# Performance: Grundlagen

- Durchführung verschiedenster Last-Tests
  - Volltext-Suche
  - Feld-Suche
  - Mit / ohne Filter, Sortierung usw.
- Test-Tool: JMeter  
<http://jakarta.apache.org/jmeter>
- Setup: 1 DB-Server, 1 Web-Server, 1 Solr-Server, 1 Last-Test-Server



# Performance: Ergebnis

- Limitierender Faktor: Web-Server
- ca. 120 req/s
- Antwortzeiten unter 150 ms bei Feld-Suchen
- Unter 300 ms bei Volltext-Suchen
- Last auf dem Solr-Server: 1 CPU-Kern ausgelastet (von 16 Kernen)
- Ergebnis: Verhältnis Web-Server : Solr-Server in etwa 16:1

# Performance Optimierungen

- Magento interner Cache (Konfiguration, Layout, CMS-Blöcke usw.)
- MySQL Query Cache



- loadCached-Methode bei Produkten (relevant z.B. für Warenkorb, Checkout)
- Zalando Frontend Cache
- Zalando Booster

# Ergebnisse

- Reduzierung der Db-Server-Last auf ca. 5%
- Reduzierung der Webserver-Last um ca. 50%
- Lastspitzen können besser abgefangen werden
- Flexiblere Möglichkeiten bei Suche/Filterung
- Aber: höher Aufwand bei Layout-Änderungen, neuen Produkt-Attributen, Such-Filtern, neuen oder veränderten Kategorien usw.

# Danke

## Q & A

Bei weiteren Fragen gerne per Mail an:

[volker.pilz@zalando.de](mailto:volker.pilz@zalando.de) oder

[daniel.nowak@rocket-internet.de](mailto:daniel.nowak@rocket-internet.de)