

# Enterprise Webapplikationen mit PHP



# Agenda

- 🔊 **Die Geschichte und Beschreibung von PHP**
- 🔊 **Entwicklungsumgebung und Livesystem**
- 🔊 **Enterprise Tools**
- 🔊 **Frameworks**
- 🔊 **PHP und Java - eine kurze Gegenüberstellung**

# Geschichte

von PHP



## Geschichte

- PHP 1 und 2
- PHP 3
- PHP 4
- PHP 5
- Ausblick auf PHP 6

# PHP 1 und PHP 2

Die Geschichte von PHP



## PHP 1

- Personal Homepage Tools
- Juni 1995
- Entwickelt von Rasmus Lerdorf
- Ziel: Tracking und Realtime Logging von Benutzerzugriffen



## PHP 2 (eigentlich PHP/FI)

- Personal Homepage Tools/Form Interpreter
- April 1996
- Conditionals und While Loops
- Dynamische GIF Erzeugung
- Datenbankunterstützung mSQL und Postges95
- Variablen und Arrays
- File Uploads

# PHP 3

Die Geschichte von PHP



## PHP Hypertext Preprocessor

- Rewrite des Parsers durch Zeev Suraski und Andi Gutmans
- Unterstützung für 6 Datenbank + ODBC
- Installation auf ca. 50.000 Domänen

# PHP 4

Die Geschichte von PHP



## PHP 4.0

- Basis: Zend Engine 1
- einfache Objektorientierung
  - Klassen
  - Klassenvariablen
  - aber nur „public“ Modifier



## PHP 4.1

- Einführung der Superglobals (`$_POST`, `$_GET`, `$_SESSION` ...)



## PHP 4.2

- `register_globals` ist nun standardmäßig ausgeschaltet



## PHP 4.3

- Einführung CLI (Command Line Interface)
- PEAR (PHP Extension and Application Library) wird mitgeliefert

# PHP 5

Die Geschichte von PHP



## PHP 5.0

- Zend Engine 2
- „private“ Klassenattribute und Methoden
- Überladung, Exceptions und Reflection API
- MySQLi - Objektorientierter Zugang zu MySQL
- SimpleXML
- „Magische Methoden“ (\_\_\_set, \_\_\_get, \_\_\_call ...)



## PHP 5.1

- PDO wird mitgeliefert



## PHP 5.2

- JSON, Zip, Objektorientierte Datumserweiterung



## PHP 5.3

- Namensräume, PHAR (PHP Archive), verbesserter Garbage Collector

# PHP 6

Ein Ausblick



## PHP 6.0

- noch in der Entwicklung
- Umstellung auf Unicode
- `register_globals`, `magic_quotes` und `safe_mode` werden abgeschafft

# Grundlagen

zu PHP

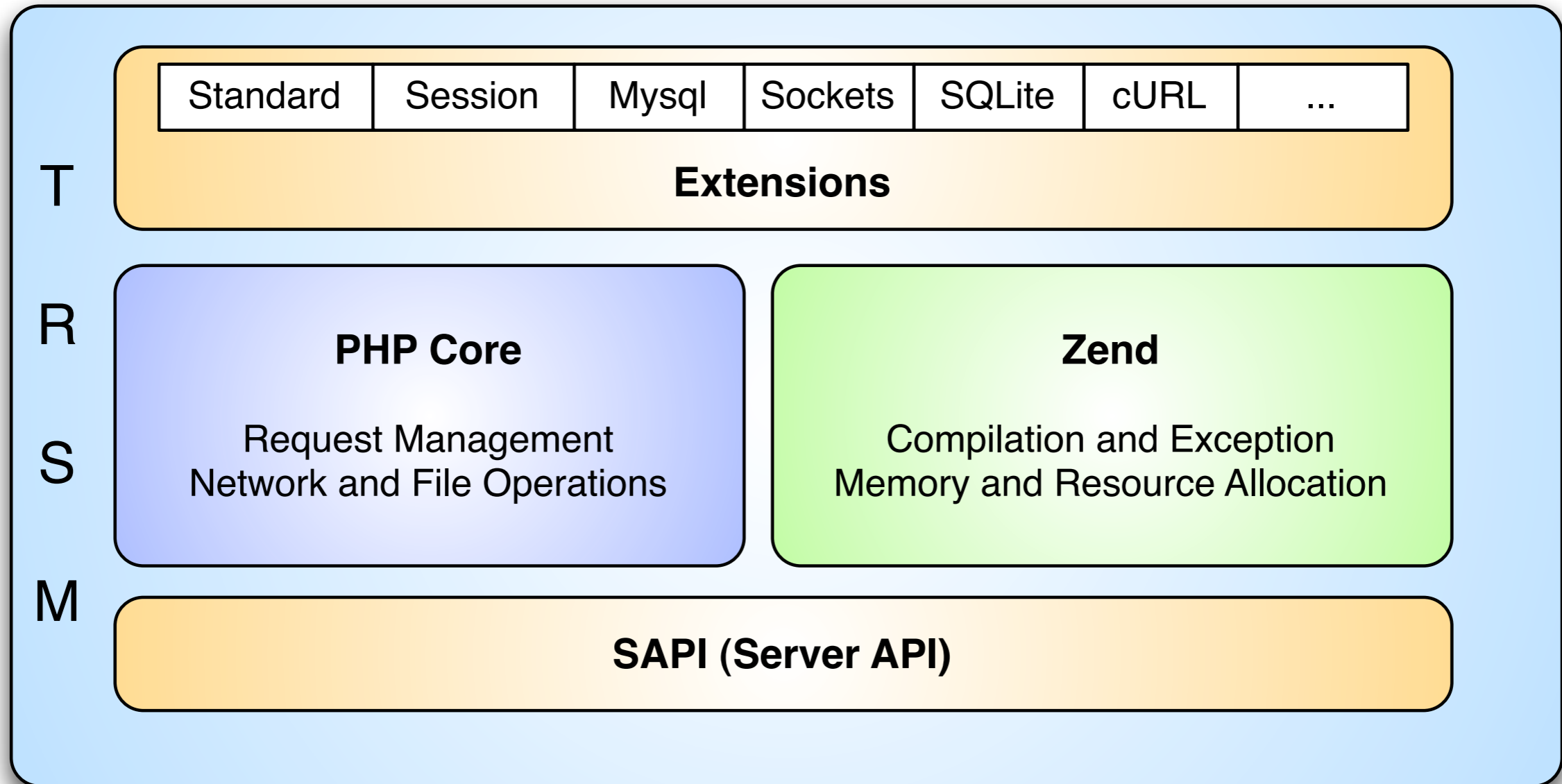


## Grundlagen

- Funktionsweise
- PEAR
- PECL
- „Hello World“

# Funktionsweise von PHP

Beschreibung von PHP



# PEAR und PECL

Beschreibung von PHP



## PEAR - PHP Extension and Application Repository

- PHP Pakete
- pear installer oder Pakete per tgz-Download
- Verzeichnisorientierte Klassenstruktur
  - Contact\_Vcard\_Parse => Contact/Vcard/Parse im lokalen Pear-Repository
- Status des Pakets
  - devel, alpha, beta, stable

## Vorteile

- einfache Nutzung
- umfangreiches Repository

## Nachteile

- Code gelegentlich veraltet bzw. nicht mehr gewartet
- Inter-PEAR Abhängigkeiten

# PEAR und PECL

Beschreibung von PHP



## **PECL - PHP Extension Community Library**

- Schwesterprojekt zu PEAR
- Repository für C-Erweiterungen

## **Vorteile**

- schnelle Einbindung
- umfangreiche Bibliothek

## **Nachteile**

- Entwicklungsumgebung muss vorhanden sein
- teilweise zweifelhafte Güte der Erweiterung
- Möglicherweise veraltet

# Hello World

in PHP

```
1 <?php
2
3 echo "Hello World";
4
5 ?>
```

```
1 <?php
2
3 $exampleHello = "Hello ";
4 $exampleInt   = 2;
5 define(WORLD, "World ");
6
7
8 echo $exampleHello . WORLD . $exampleInt;
9
10 ?>
```

# Entwicklung

Überblick



## Entwicklungsumgebung

- Eclipse
- Netbeans
- Zend Studio
- Alternative Umgebungen

# Eclipse als IDE



## Bekannte JAVA IDE



## PDT Erweiterung notwendig



## Vorteile:

- Zugang zu allen Eclipse Plugins (Subclipse bspw)
- gutes Highlighting und Vervollständigung
- meist gute Navigation durch die Klassen



## Nachteile:

- keine „Enterprise Tools“ Unterstützung (bspw Unit Tests)
- keine Refactoring Unterstützung
- keine „Code Generation“ Unterstützung

# Netbeans als IDE



**Ebenfalls JAVA IDE**



**PHP Edition verfügbar**



**Vorteile:**

- Versionsverwaltung ohne Plugin vorhanden (CVS, SVN, Mercurial)
- Unterstützung von Unit Tests
- Unterstützung für das Symfony Framework
- ansonsten wie Eclipse



**Nachteile:**

- wie Eclipse, besonders Refactoring

# Zend Studio als IDE

## kommerzielle IDE auf Basis von Eclipse

### Vorteile:

- Refactoring, Interface Support
- Unit Tests, Profiler und Code Coverage Support
- Integration des Zend Servers
- „Quick fix“
- Integration des Zend Framework

### Nachteile:

- Kosten
- sehr Zend-lastig (Zend Server und Framework)

# Alternativen



## Vim

- ohne grafische Oberfläche nutzbar
- PHP extension jedoch schlecht gewartet (Stand PHP 5.1.4)
- keine Navigation durch die Klassen



## Textmate

- Code Templates sind gut
- Vervollständigung nicht besonders toll
- keine Navigation durch die Klassen



## Fazit:

- durchaus nutzbar in besonderen Fällen
- Produktivität ist geringer als bei den vorgestellten IDEs

# Entwicklung

Überblick



## Livesystem

- Standard Linux
- Zend Application Server
- Windows

# Linux als Livesystem



## **LAMP ist das typische PHP-System**

- Linux
- Apache (als Webserver)
- MySQL (als Datenbank)
- PHP (mod\_php)

## **Vorteile:**

- Installation relativ einfach
- Anleitungen für Problemfälle und Sonderkonfigurationen im Netz
- Hohe Verbreitung bei den Hostern

## **Nachteil:**

- nicht genormt und nicht immer aktuell

# Zend Server



## **kommerzieller Applicationserver von Zend**

- PHP Stack mit Support
- integriertes Caching (für Seiten)
- Caching API (für Datensets)
- Zend Optimizer
- Java-Connector
- Debugger-Schnittstelle
- Überwachungs- und Diagnosefunktionen
- Apache oder IIS als Basis
- Community Edition mit Einschränkungen verfügbar



## **Nachteil:**

- entfaltet komplette Funktionen nur mit Zend Studio
- Kosten von ca 1200 Eur p.a.

# Sonstige Systeme



## **Windows/Windows Server mit IIS**

- Windows Cache Extension
- Microsoft unterstützt PHP explizit und fördert die Entwicklung

## **Vorteile:**

- Tolle Integration, wenn Windows Infrastruktur vorhanden
- Anbindung an COM
- Skript als Service
- relativ genormt








## **Nachteile:**


- Einige Module (insb. aus PECL) funktionieren nicht
- kleinere Community

# Enterprise Tools

Überblick



-  **Qualitätssicherung**
-  **Dokumentation**
-  **Deployment**
-  **Continuous Integration**
-  **Debugging**
-  **Caching**
-  **Closed Source**

 **„You cannot control, what you cannot measure“  
(Tom DeMarco)**

 **Einhaltung von Coding Guidelines**

 **Beurteilung der Code-Qualität**

# PHPUnit

Unit Testing unter PHP



- **DAS Unit Testing Framework für PHP**
- **xUnit kompatibel**
- **Mocking Framework integriert**
- **Code Coverage Report**
- **Ausgabeformate (json, junit xml, tap xml)**
- **Generierung von Dokumentation**
- **TestCase Generierung durch Annotations**
- **Durch Extensions erweiterbar**

# PHP CodeSniffer (phpcs)



- **native PHP Anwendung**
- **erkennt Verletzungen eines Coding Standards**
- **ca. 130 Regeln werden mitgeliefert**
- **einige Regelsets (Standards) bereits definiert**
- **verschiedene Ausgabeformate (bspw checkstyle)**
  
- **Implementierung eigener Regeln möglich**
- **Anpassung vorhandener Regeln durch Ableitung**
- **Eigene Standards sind möglich**

# PHP depend

- **In PHP entwickelt**
- **Installation per PEAR**
- **Überprüft Abhängigkeiten zwischen Klassen**
- **Errechnet Komplexität per Softwaremetrik**
  - zyklomatische Komplexität (McCabe)
  - NPath Komplexität (Nejmeh)
- **bestimmt verschiedene LOC Werte**
- **Ausgabe eine XML-Datei - kompatibel zu jdepend**

# PHP Mess Detector (phpmd)

- 🔊 **in PHP entwickelt**
- 🔊 **bringt ein Basis-Regelset mit**
  - Software Metrik (NPath und zyklomatische Komplexität)
  - Regeln die sich auf Klassen beziehen
  - Regeln die sich auf Methoden an sich beziehen
- 🔊 **Ausgabe ist eine xml Datei im pmd-xml Format**

# PHP Copy&Paste Detector (phpcpd)



- in PHP geschrieben**
- findet „Copy&Paste“-Code**
- 5 gleiche Zeilen werden als Kopie angesehen**
- Ausgabeformat pmd-cpd xml**

```
sb@ubuntu ~ % phpcpd /usr/local/src/phpunit/trunk/PHPUnit
phpcpd 1.1.0 by Sebastian Bergmann.
```

```
Found 4 exact clones with 131 duplicated lines in 7 files:
```





- Extensions/Database/DataSet/AbstractTable.php:156-190  
Extensions/Database/DataSet/ReplacementTable.php:172-206
- Samples/BankAccountDB/BankAccountDBTest.php:84-128  
Samples/BankAccountDB/BankAccountDBTestMySQL.php:84-128
- Tests/Extensions/Database/DataSet/XmlDataSetsTest.php:71-98  
Tests/Extensions/Database/DataSet/YamlDataSetTest.php:70-97
- Tests/Extensions/Database/DataSet/XmlDataSetsTest.php:71-97  
Tests/Extensions/Database/DataSet/CsvDataSetTest.php:70-96

```
0.21% duplicated lines out of 61720 total lines of code.
```

# Dokumentation

Enterprise Tools








-  **immer notwendig**
-  **besonders in der Agilen Entwicklung**
-  **der Entwickler muss entlastet werden**
-  **solte aktuell gehalten werden**

# PHP Documentor (phpdoc)

Automatisierte Dokumentation



-  **erstellt Dokumentation**
-  **spezielle phpdoc Kommentare im Quelltext**
  - keine Trennung von Quellen und Dokumentation
-  **phpdoc entspricht javadoc**
-  **Ausgabe als html, pdf, chm oder docbook**
-  **Integration in CI-Server vorhanden**

# Deployment

Enterprise Tools



- 📌 **Die Applikation läuft auf einem Server**
- 📌 **verschiedene Stufen bei der Entwicklung**
  - Development Server
  - Staging Server
  - Live Server

# Phar - PHP Archive

Deployment



- **seit 5.3.0 verfügbar, daher noch geringe Verbreitung**
- **entspricht JAR-Datei in der Java-Welt**
- **kein externes Tool ist für die Verarbeitung notwendig**
- **direkter Zugriff auf Phar-Datei**
  - per include
- **verschiedene Kompressionsvarianten**

# Phing

Deployment



- **in PHP geschriebenes Build System**
- **angelehnt an Apache ant (java build tool)**
- **viele „Tasks“ bereits vorhanden**
- **Quasistandard in der PHP-Welt**
- **nutzt PEAR**
- **kann auch PHARs bauen**

# Alternativen

Deployment



## ANT

- java notwendig



## PEAR Server

- eigener PEAR-Server notwendig
- PEAR Server muss vom Zielsystem erreichbar sein



## eigene Shellskripte

- Pflege möglicherweise aufwendig
- Wissenstransfer notwendig






## Versionsverwaltung

- svn, git usw.
- Versionsverwaltungssystem muss vom Zielsystem erreichbar sein

# Continuous Integration



-  **Monitor für das Code Repository**
-  **zentrale Stelle für genormte Codechecks**
-  **optische Darstellung der Codequalität**

# Xinc

Ein PHP-CI-Server



- **in PHP geschriebener CI-Server**
- **Support für Phing und svn integriert**
- **Installation per PEAR**
- **Unterstützt phpDoc, phpUnit und Xdebug**
- **Betriebssystem unabhängig (auch als Win-Service)**
  
- **Vorteile:**
  - alles in PHP - keine erhöhte Komplexität
  
- **Nachteile:**
  - keine Unterstützung der QA-Tools
  - Entwicklung scheint eingeschlafen (Mai 2008)

# CruiseControl

Ein JAVA-CI-Server



- 🔊 **Java CI-Server**
- 🔊 **mit phpUnderControl-Plugin kompatibel zu PHP**

## 🔊 Vorteile:

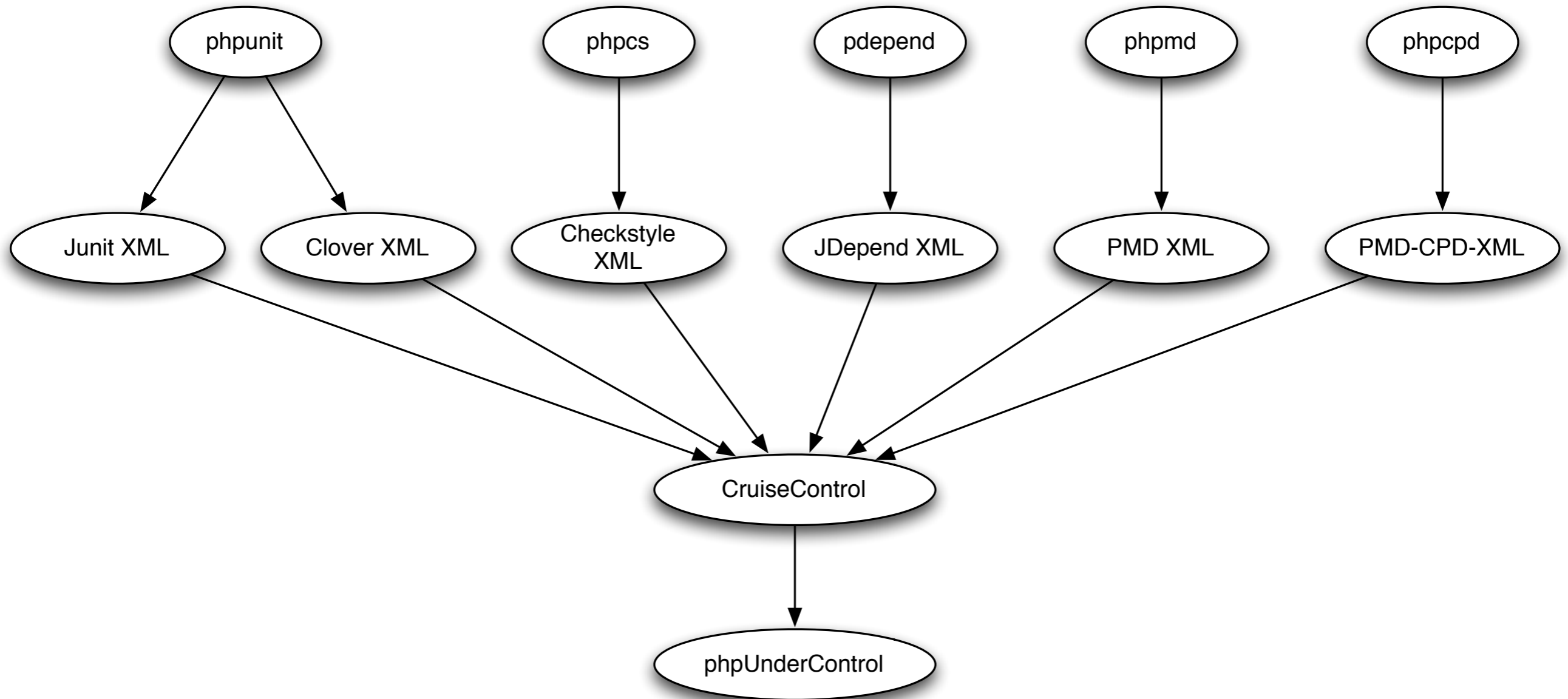
- erprobter CI-Server
- Support der QA-Tools
- letzte dev-Version vom Oktober 2009

## 🔊 Nachteile:

- Komplexität durch Java

# CI und die QA-Tools

Am Beispiel von CruiseControl



# phpUnderControl

Screenshot „Dashboard“



Browser address bar: <http://192.168.13.13:8080/buildresults/phpUnderControl?tab=metrics>

Project: **phpUnderControl** Build: **More builds** *waiting for next time to build since 02/03/2008 00:32:25*

By Manuel Pichler

Overview | Tests | XML Log File | **Metrics** | Coverage | Documentation | CodeSniffer | PHPUnit PMD

### Project Metric Summary

Number of Build Attempts	124
Number of Broken Builds	15
Number of Successful Builds	109

#### Breakdown of build types

Good Builds: 109 (87.9%)  
Broken Builds: 15 (12.1%)

#### Breakdown of build timeline

● Good Builds ● Broken Builds

#### Unit coverage

● Executable ● Covered

#### Unit Tests

● Total ● Failures

#### Test to Code ratio

● Classes ● Methods ● Test Classes ● Test Methods

#### Coding Violations

● PHP CodeSniffer ● PHPUnit PMD ● PHPDoc

phpUnderControl 0.3.5 is Copyright (C) 2007-2008 by Manuel Pichler hosted on [phpunit.de](http://phpunit.de).  
phpUnderControl is an extension for [CruiseControl](http://cruisecontrol.sourceforge.net).

# Debugging mit Xdebug

- 🔍 **Xdebug Extension für PHP ist der Standard**
- 🔍 **Debugger mit Anbindung an gängige IDEs**
  - Eclipse und Netbeans unterstützen Xdebug
- 🔍 **Profiler mit Ausgabe einer Cachegrind Datei**
  - kcachegrind für Linux
  - WinCacheGrind für Windows
  - MacCallGrind für MacOS X
  - Webgrind für Analyse im Browser
- 🔍 **Code Coverage Analyse (nutzt bspw PHPUnit)**
- 🔍 **ersetzt die PHP Fehlerausgabe**
  - stack- und functiontraces

# Caching

Überblick

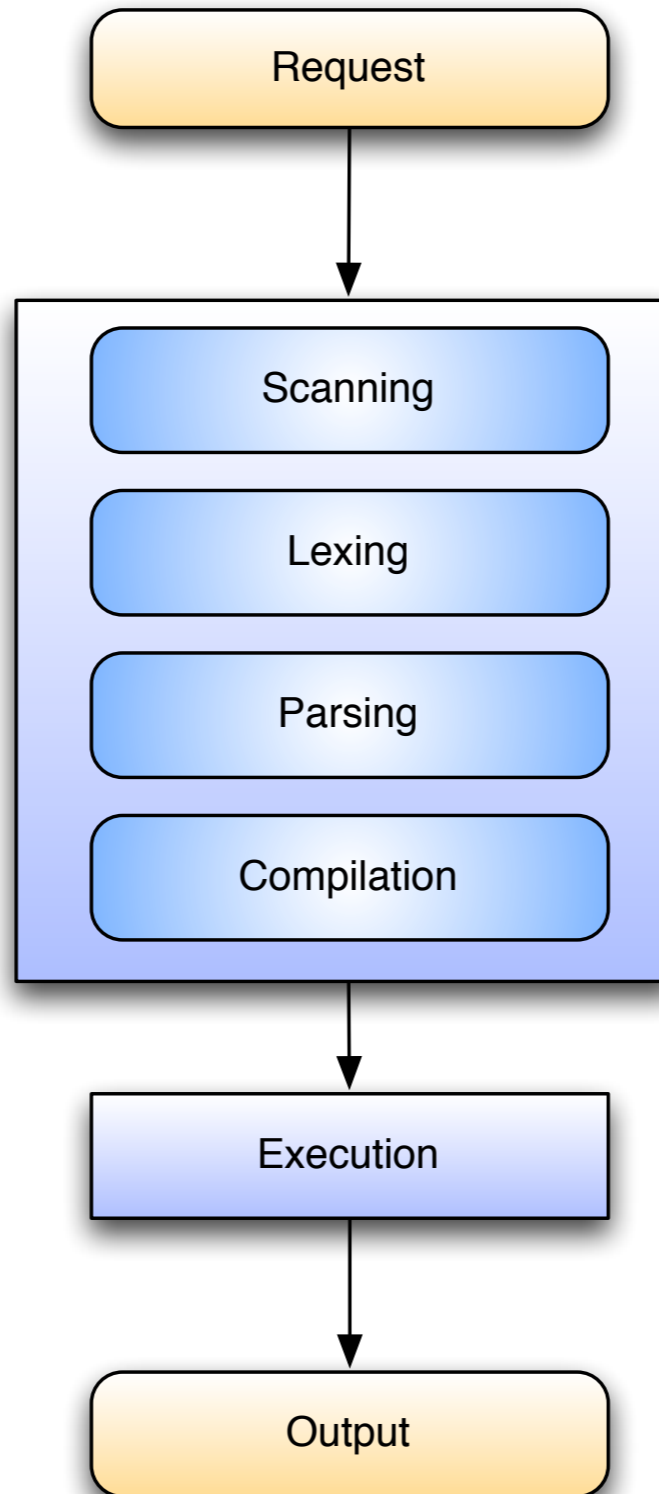


 **OpCode Caching**

 **Memcached**

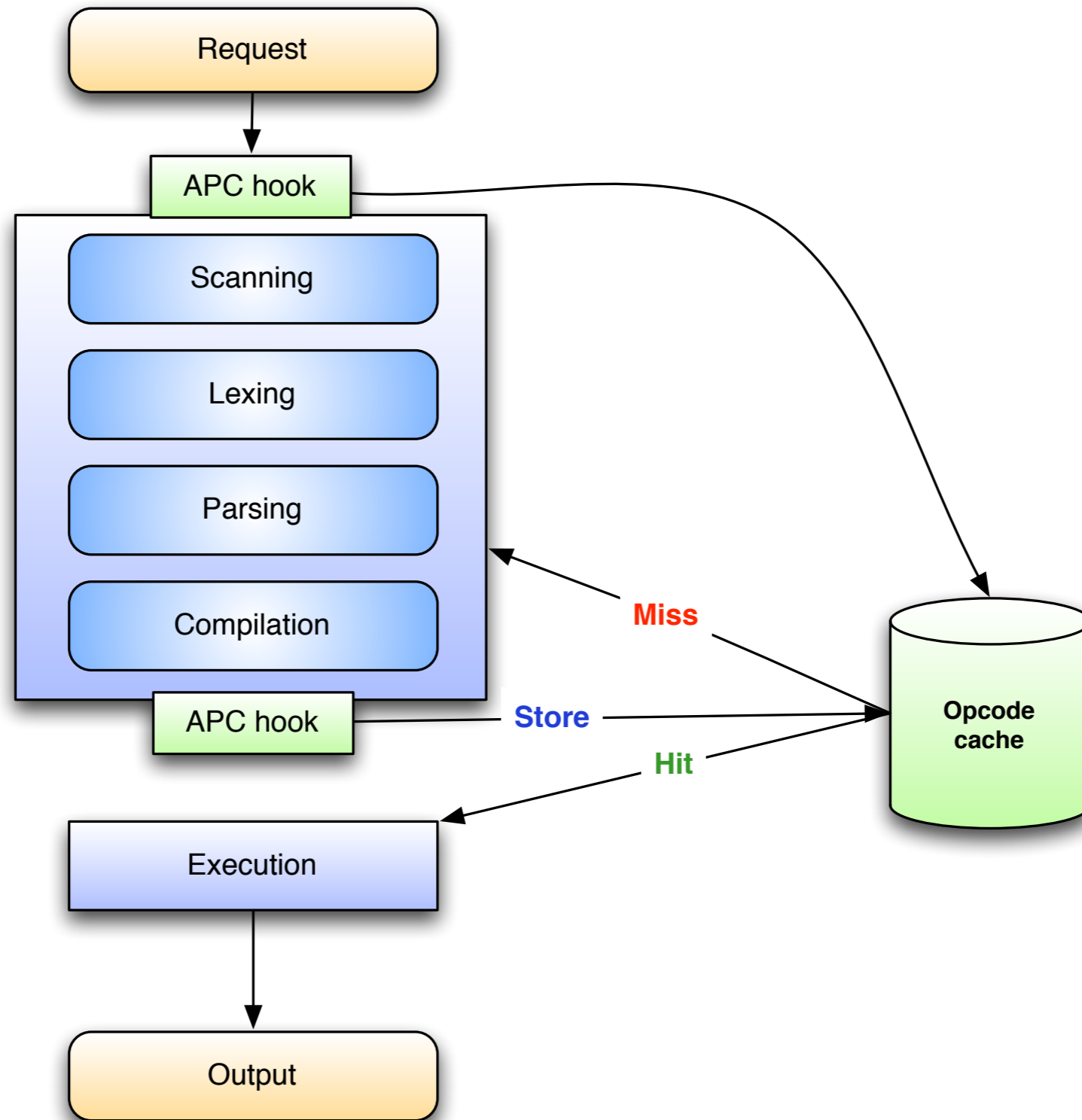
# Caching APC

Scriptverarbeitung bei PHP



# Caching APC

Scriptverarbeitung bei PHP mit APC



# Memcached



- **„distributed memory object caching system“**
- **Entlastung der Datenbank-Server**
- **nutzt nur Arbeitsspeicher des Servers**

## • **Was wird hier gecached?**

- Ergebnisse einer DB Query
- Thumbnails
- Objekte, die durch viel Rechenzeit erzeugt wurden
- Eigentlich alles, was viel Rechenzeit benötigt

## • **Zu beachten:**

- nicht persistent, keine Sicherheit, keine Dumps und keine Sessions
- Cache nur benutzen wenn notwendig - erhöht Komplexität

# Closed Source



 **PHP Quelltexte immer offen**

 **Verschlüsselung von PHP-Dateien vielleicht gewünscht**

- rechtlicher Schutz
- Schutz vor illegalen Kopien

# IonCube PHP Encoder



- **benötigt Decoder Plugin auf dem Webserver**
- **kostengünstig (ab \$199)**
- **Bindung an Lizenzdatei, MAC oder IP**
- **Verschleierung der Namen (gegen Decompiler)**
- **Zugriff auf IonCube API (verschlüsselte Templates)**

# Zend Guard

## **Wie ionCube Encoder**

## **Aber:**

- Integration in Zend Studio
- erkennt Code Bausteine die nicht verschlüsselt werden sollten
- Zend Optimizer bei vielen PHP Installation vorhanden

## **Nachteil: 600 EUR p.a.**

# Frameworks



- **Sammlung von Basiskomponenten**
- **Ansatzpunkt bei der Suche nach neuen Mitarbeitern**
- **geprüfte Qualität des Codes**
- **Community - neue Features, Problemlösungen**

# Zend Framework

- **Framework von Zend**
- **sehr gute Integration in die Zend Produkte**
- **umfangreiche Komponentensammlung**
  - MVC
  - Anbindung an DOJO Toolkit
  - Lucene Anbindung
  - PDF Generator
  - u.v.m
- **setzt primär auf Eigenentwicklungen**
- **Zertifizierung möglich**

# Symfony

PHP meets Ruby on Rails



- **PHP Version von Ruby on Rails**
- **relativ starre Verzeichnisstruktur**
- **viele 3rd Party-Komponenten**
- **nutzt Bibliotheken aus der PHP Welt**
  - Doctrine oder Propel als ORM
- **„umgebende“ Komponenten als Bibliothek**

# ezComponents

Enterprise PHP Lib



- 🔍 **nur eine Komponentensammlung**
  - Enterprise PHP Library
- 🔍 **Module unabhängig vom Rest nutzbar**
- 🔍 **sehr gute Diagrammkomponente**
- 🔍 **bietet Workflowkomponente**

# Nachteile

von Frameworks



## Nachteile

- **Steilere Lernkurve**
  - komplexer als reines PHP
  - Umstieg von Framework A auf B kommt einer Reimplementierung gleich
- **Code Effizienz**
  - Zeit- und Platzkomplexität nicht kontrollierbar
  - keine Kontrolle über den Ressourcenverbrauch
- **Zeit Effizienz**
  - langsamer Code
- **„advanced web features“-Unterstützung**
  - AJAX Support
  - CSS Sprites

# PHP und JAVA








- Objektorientierte Sprache
- professionelle Entwicklungsumgebung
- große Community
- QA/QS - Tools
- CI-Server
- erprobte Serverinfrastrukturen
- Zertifizierungen
- Namenhafte Supporter

# Vorteile von Java

- **Application Server**
- **Java Specification Request (JSR)**
- **sehr viele externe Komponenten**
- **Bytecode (class-Dateien)**
- **Desktop-Applikationen**

# Vorteile von PHP

-  **Skalierbarkeit**
-  **geringe Einstiegshürde**
-  **für Web-Applikationen entwickelt**
-  **großer Fundus an Web-Applikationen**
-  **PHP bereits bei Lowcost-Webhostern vorinstalliert**

Fragen?



**Vielen Dank für die Aufmerksamkeit**



## Bücher

- Programming PHP (ISBN: 978-0596006815)
- PHP in Action (ISBN: 978-1932394757)
- Enterprise PHP Tools (ISBN: 978-3935042932)
- PHP Design Patterns (ISBN: 978-3897218642)



## Links

- Geschichte und Grundlagen
  - <http://www.php.net/>
  - <http://en.wikipedia.org/wiki/PHP>
  - <http://groups.google.com/group/comp.infosystems.www.authoring.cgi/msg/cc7d43454d64d133>



## Links

### ● Entwicklung

- <http://www.eclipse.org/pdt/>
- <http://netbeans.org/features/php/>
- <http://www.zend.com/de/products/studio/>
- <http://www.vim.org/>
- <http://macromates.com/>
- <http://de.wikipedia.org/wiki/Linux>
- <http://httpd.apache.org/>
- <http://www.mysql.de/>
- <http://www.zend.com/de/products/server/>
- <http://www.iis.net/expand/wincacheforphp>



## Links

### ● Enterprise Tools

- <http://www.phpunit.de/>
- [http://pear.php.net/package/PHP\\_CodeSniffer/](http://pear.php.net/package/PHP_CodeSniffer/)
- <http://pdepend.org/news.html>
- <http://phpmd.org/about.html>
- <http://github.com/sebastianbergmann/phpcpd>
- <http://www.phpdoc.org/>
- <http://phing.info/trac/>
- <http://php.net/manual/en/book.phar.php>
- <http://ant.apache.org/>
- <http://www.pirum-project.org/>
- <http://code.google.com/p/xinc/>
- <http://cruisecontrol.sourceforge.net/>
- <http://phpundercontrol.org/about.html>
- <http://xdebug.org/>



## Links

### ● Enterprise Tools

- <http://php.net/manual/de/book.apc.php>
- <http://php.net/manual/de/book.memcache.php>
- <http://memcached.org/>
- [http://www.ioncube.com/sa\\_\\_encoder.php](http://www.ioncube.com/sa__encoder.php)
- <http://www.zend.com/de/products/guard/>

### ● Frameworks

- <http://www.symfony-project.org/>
- <http://ez.no/ezcomponents>
- <http://framework.zend.com/>