



Groovy und Grails Quo vadis?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de

Falk Sippach

Trainer, Berater, Entwickler



Co-Organisator

Schwerpunkte

Architektur

Agile Softwareentwicklung

Codequalität



Java und XML

) Software Factory)

- Schlüsselfertige Realisierung von Java Software
- Individualsoftware
- Pilot- und Migrationsprojekte
- Sanierung von Software
- Software Wartung

) Object Rangers)

- Unterstützung laufender Java Projekte
- Perfect Match
- Rent-a-team
- Coaching on the project
- Inhouse Outsourcing

) Competence Center)

- Schulungen, Coaching, Weiterbildungsberatung, Train & Solve-Programme
- Methoden, Standards und Tools für die Entwicklung von offenen, unternehmensweiten Systemen

*Das Jahr 2015 begann turbulent für die beiden bekanntesten Projekte aus dem Groovy Universum. Von der bisherigen "Mutter" **Pivotal den Laufpass erhalten**, musste sich Groovy auch noch auf die Suche nach einem neuen Zuhause begeben und ist letztlich **bei Apache fündig** geworden. All diese Unsicherheiten haben die neuen Features der **Releases 2.4 (Groovy) bzw. 3.0 (Grails)** ziemlich in den Hintergrund gedrängt. Dabei sind die Projekte lebendiger denn je und vor allem schon längst **reif für den produktiven Einsatz**.*

*Wir werden uns die wichtigsten und interessantesten Neuerungen der vergangenen Releases anschauen und natürlich auch einen **Ausblick auf die Zukunft** und Roadmaps wagen.*

- Motivation + Politisches
- Groovy
- Grails
- Ausblick

Warum Groovy und Grails?

- Groovy-Fan seit 1.0 (2006)
- 5+ Jahre Projekterfahrung mit Grails

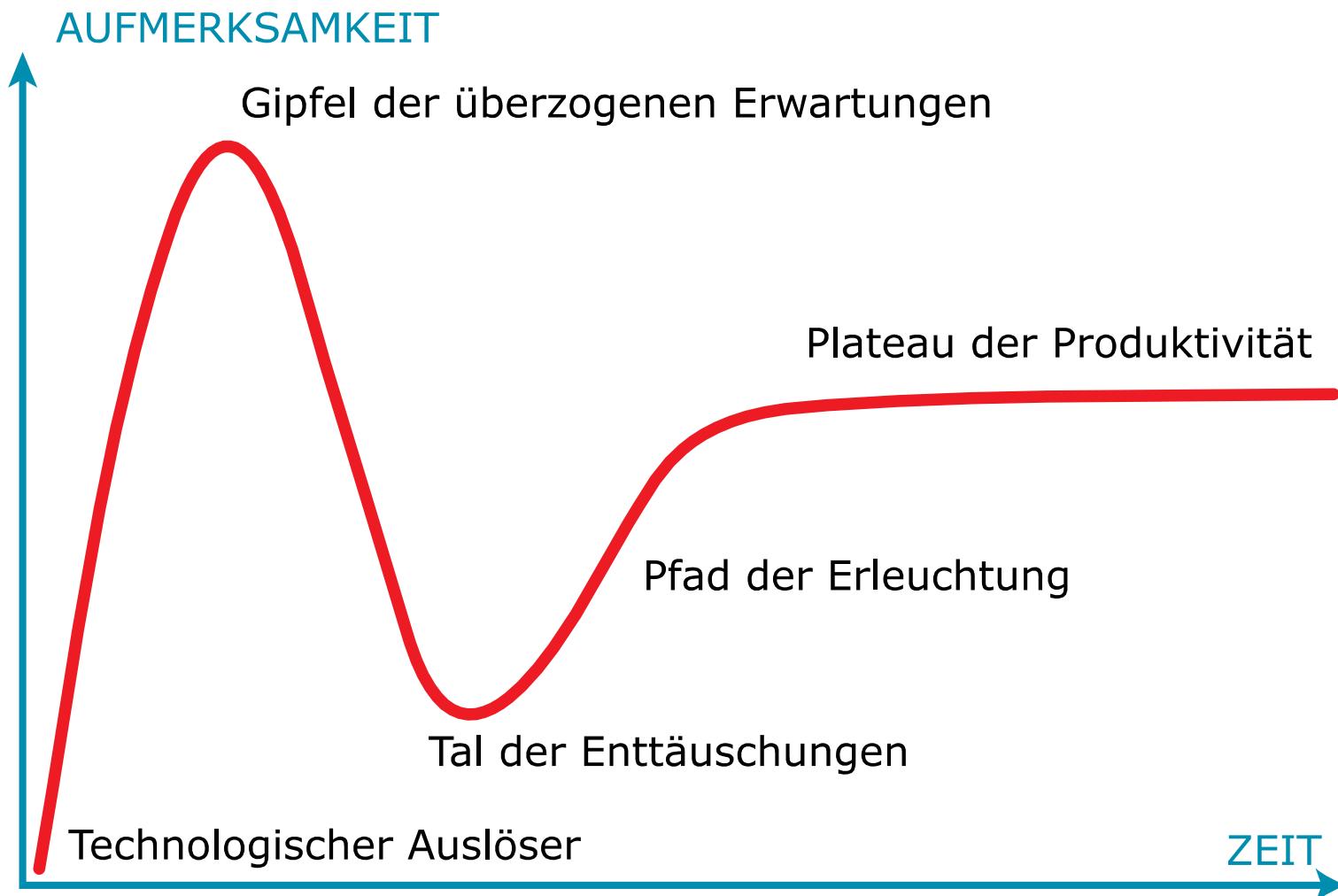


Foto von alankotok, available under a [CC0 Public Domain license](#).

Aber es ist ruhig geworden ...



Öffentliche Aufmerksamkeit
tendiert gefühlt gegen Null!



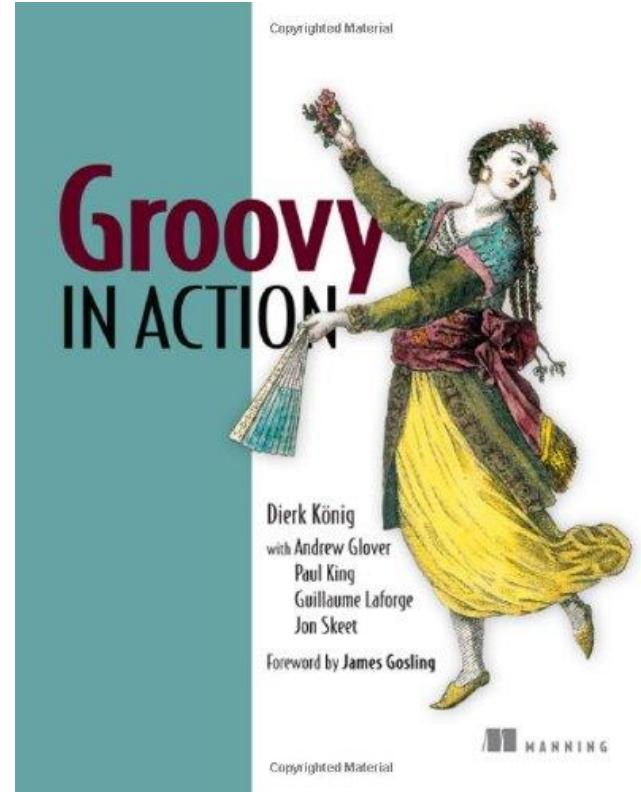
[Grafik von Idotter](#), available under a [Creative Commons Attribution-ShareAlike 3.0 Unported \(CC BY-SA 3.0\) license](#).

Sind Groovy und Grails nicht mehr hip genug?

Regina – "Gut Ding will Weile haben"

Who the fuck is Regina?

Gehen wir zurück ins Jahr 2007



Planung einer zweiten
Auflage von GinA:

Beginn einer
"Never-Ending-
Story"

ReGinA war geboren

*Hi all,
announcing the start of MEAP for June (2009)...*

*Anyway, this summer is the time where we will do the majority
of the work on the second edition and you can expect the
MEAP progressing rather quickly. ...*

*thanks for your understanding
Dierk*

<https://forums.manning.com/posts/list/18629.page>

6 Jahre und viele Tweets später ...

2009

Dierk König @mittie · 22 Aug 2009
soooo many new things in #regina
them all in #reg
 

Dierk König @mittie · 20 Aug 2009
I guess I have to rewrite the #groovy type system section in #regina a third time - tomorrow, *sigh*

2010

gt groovytweets @groovytweets · 12 Jan 2010
RT @the_very: Finally! The book is out.
Chapters 1-6 is out.
 

Dierk König @mittie · 18 Apr 2010
#groovy in action 2nd ed will now cover groovy 1.8 (due in oct/nov 2010).
#regina (via @wmacgyver) dec 2010 I would say...

2011

Dierk König @mittie · 31 Mar 2011
#regina MOP explanations and
explanations about
 

Erik Pragt @epragt · 30 Nov 2011
Please RT if you also would like #Groovy in Action 2 to be ready for Christmas.
#regina I'll give out 5 books among RT-ers!
 19 ⭐ ...

2012

?

2013

Dierk König @mittie · 21 Mar 2013
#ReGina MEAP
@ManningBooks
 

Wolfgang Schell @jetztgradnet · 21 Mar 2013
Welcome back, #reGINA, long time no see! Thanks for the update and finally
with mobile formats, yeay! (@mittie, @ManningBooks) #Groovy
 3 ⭐ 1 ...

2014

?

Suche: #regina #groovy(lang)

JUNI 2015



Groovy in Action, 2. Auflage (ReGinA)

Mal anders betrachtet

Eine Ursache:

hohes Tempo bei Groovy-Releases

immerhin: MEAP = Dauer Abo



Foto von [RickToxik](#), available under a [CC0 Public Domain license](#).

Januar 2015



[Foto von StepanFoto](#), available under a [CC0 Public Domain license](#).

Groovy 2.4 And Grails 3.0 To Be Last Major Releases Under Pivotal Sponsorship

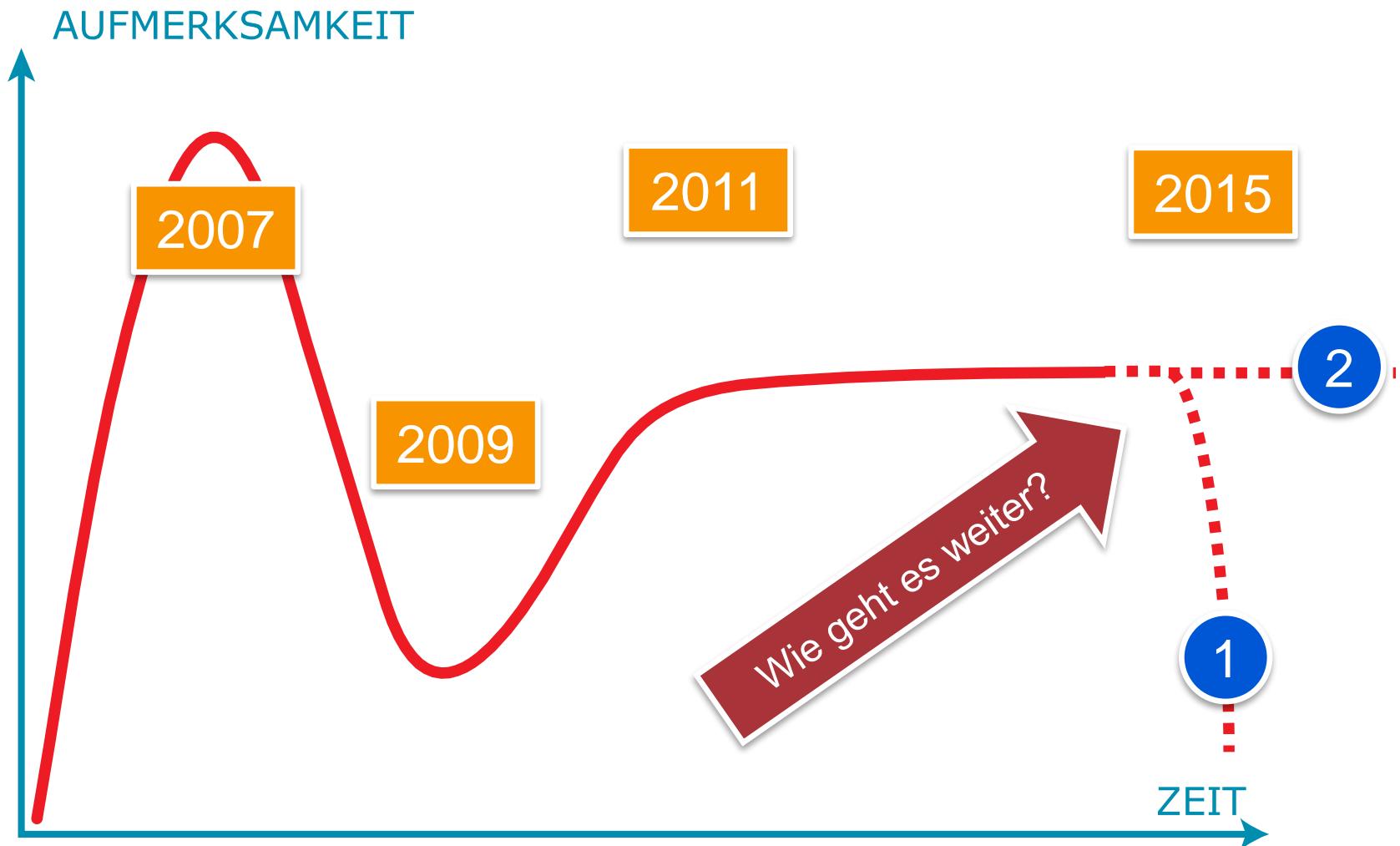
The decision ... is part of Pivotal's larger strategy to concentrate resources on ... its **growing traction in Platform-as-a-Service, Data, and Agile development.**

Pivotal has determined that the **time is right** to let further development ... be led by **other interested parties** ... who can best serve the goals ...

<http://blog.pivotal.io/pivotal/news-2/groovy-2-4-and-grails-3-0-to-be-last-major-releases-under-pivotal-sponsorship>

	Groovy	Grails
2003	Projektstart	
2005		Projektstart
2006	1.0	
2007		G2One
2008		1.0
2008		SpringSource
2009		VMware
2013		Pivotal
2015	???	???

Zeitliche Einordnung Groovy/Grails



Grafik von Idotter, available under a Creative Commons Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0) license.

Pivotal versucht gut aus der Sache rauszukommen



Frist bis

31.03.2015

Schonfrist

Unterstützung bei Sponsorsuche

Hosting von grails.org

Folge des Vert.x Desasters?

<http://www.heise.de/developer/meldung/VMware-beansprucht-Copyright-an-Vert-x-Projekt-1779511.html>

Groovy/Grails Toolsuite (Eclipse) eingestellt

aber weiterhin <https://github.com/groovy/groovy-eclipse>
Grails 3.0 benötigt kein spezielles Eclipse-Plugin mehr

Gradle entwickelt jetzt eigenes Eclipse-Plugin

Wenn man sowieso am Boden liegt ...

- Codehaus schließt
- gestartet 2003
- 2015 der Übermacht von Github und Co. gebeugt
 - <http://www.codehaus.org/history/>
- Groovy braucht einen neuen Hosting Service
 - Source-Repo sowieso schon bei Github
 - aber Jira, Homepage, Wiki, ...

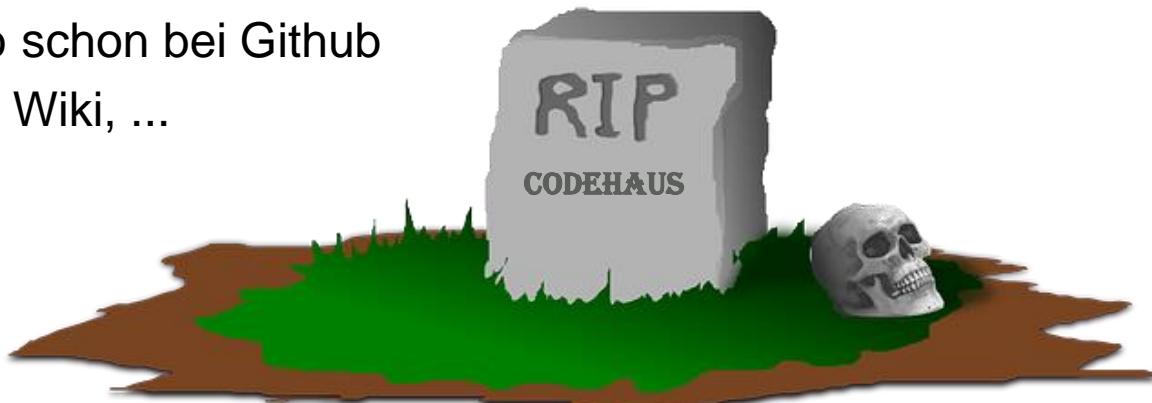


Foto von OpenClipartVectors, available under a [CC0 Public Domain license](#).

- Eclipse Foundation
- Apache Foundation
- Software Freedom Conservancy



And the winner is ...

Rege Diskussion auf Mailingliste

Schwere Entscheidung



24.03.2015

Aufnahme im Inkubator

- **5 Mentoren**
- **5 initiale Committer**
- **neue Mailinglisten**
- **Jira-Tickets verschoben**
- **neues Git-Repo**
- **weitere Committer**

By kOchstudiO [Public domain], via Wikimedia Commons

	Groovy	Grails
2003	Projektstart	
2005		Projektstart
2006	1.0	
2007		G2One
2008		1.0
2008		SpringSource
2009		VMware
2013		Pivotal
2015	2.4 (Apache)	???

Grails has a New Home at OCI



oci @ObjectComputing · 9. Apr.

We have announced some BIG news!
{@grailsframework} and #Grails have a
New Home at OCI!!
interact.stltoday.com/pr/business/PR ...



OCI

“ With the support and backing of OCI, an established IT solutions engineering firm, with deep roots in Open Source, Grails is moving forward with maximum momentum. ”

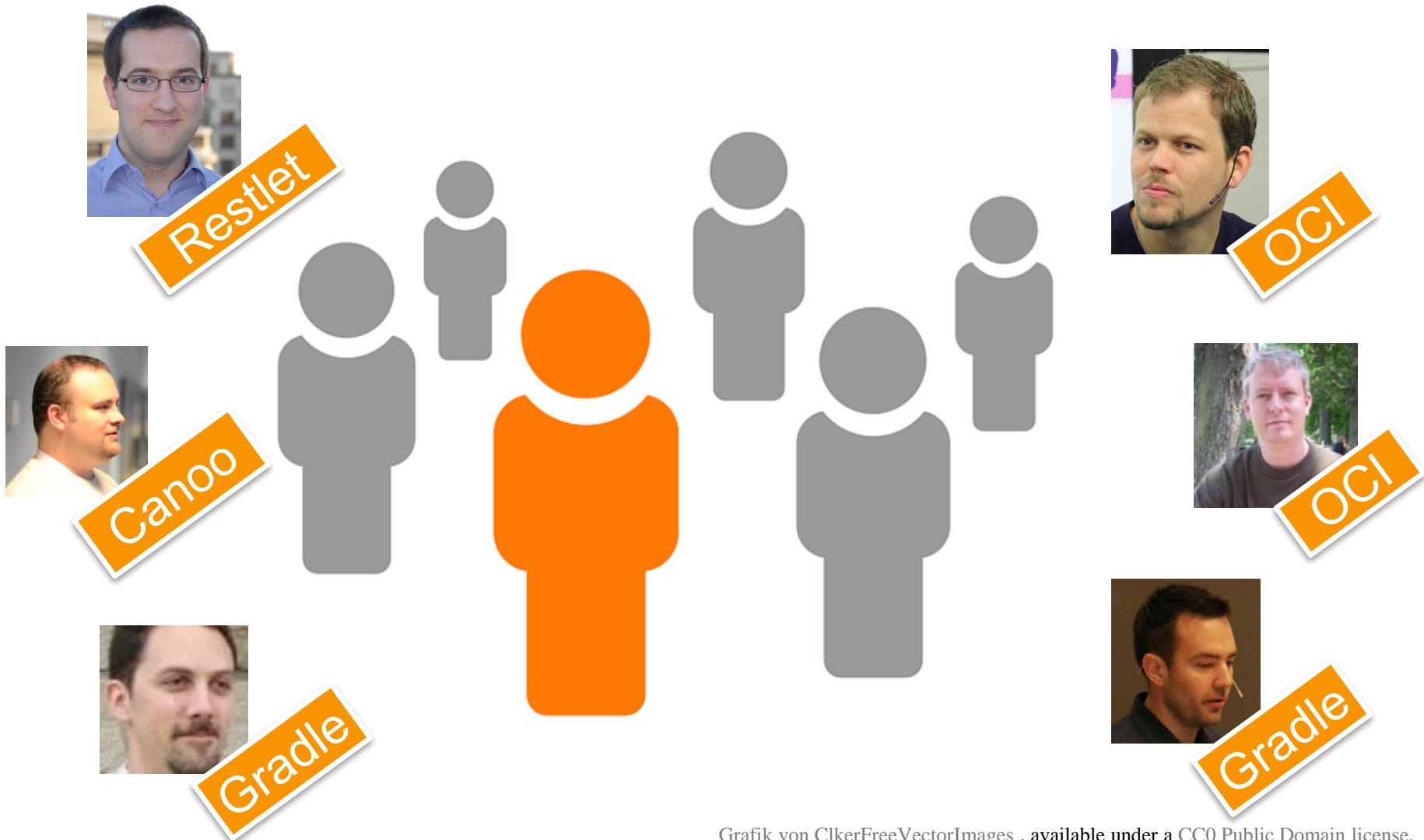
Object Computing, Inc.

St. Louis, MO

<http://www.ocieweb.com>

	Groovy	Grails
2003	Projektstart	
2005		Projektstart
2006	1.0	
2007		G2One
2008		1.0
2008		SpringSource
2009		VMware
2013		Pivotal
2015	2.4 (Apache)	3.0 (OCI)

Jeweils 3 Core Committer waren bei Pivotal



Grafik von ClkerFreeVectorImages , available under a [CC0 Public Domain license](#).

Who is Groovy?

- interessante Statistik der Committer
- insgesamt 100+ Committer seit 2003
- Sieger: **Paul King** (nicht bezahlt)



<http://melix.github.io/blog/2015/02/who-is-groovy.html>



2008

"I can honestly say if someone had shown me the **Programming in Scala** book by Martin Odersky, Lex Spoon & Bill Venners back in **2003** I'd probably have **never created Groovy.**"

James Strachan

<http://macstrac.blogspot.de/2009/04/scala-as-long-term-replacement-for.html>

Hauptprinzipien von Groovy

1



Featurereich

3



Dynamisch

2



Java-freundlich

4



Robuste Plattform

~~schwache~~ dynamische Typisierung

fehlende Tool-Unterstützung (Refactoring)

Fehler erst zur Laufzeit

Performance

Ignorieren der Kapselung



[Foto von ashishacaway](#), available under a [CC0 Public Domain license](#).



Dierk König

@mittie



Following

In OO, static types limit your choice. That makes dynamic languages attractive. (1/2)



Dierk König @mittie · Nov 24

In pure FP, almost any code that works can be super-strongly type-inferred and the pain is carried by the compiler. (2/2)



7



5



Dynamische Typisierung ist ein Feature

ermöglicht Runtime-Metaprogrammierung

Alternativen

@TypeChecked und @CompileStatic

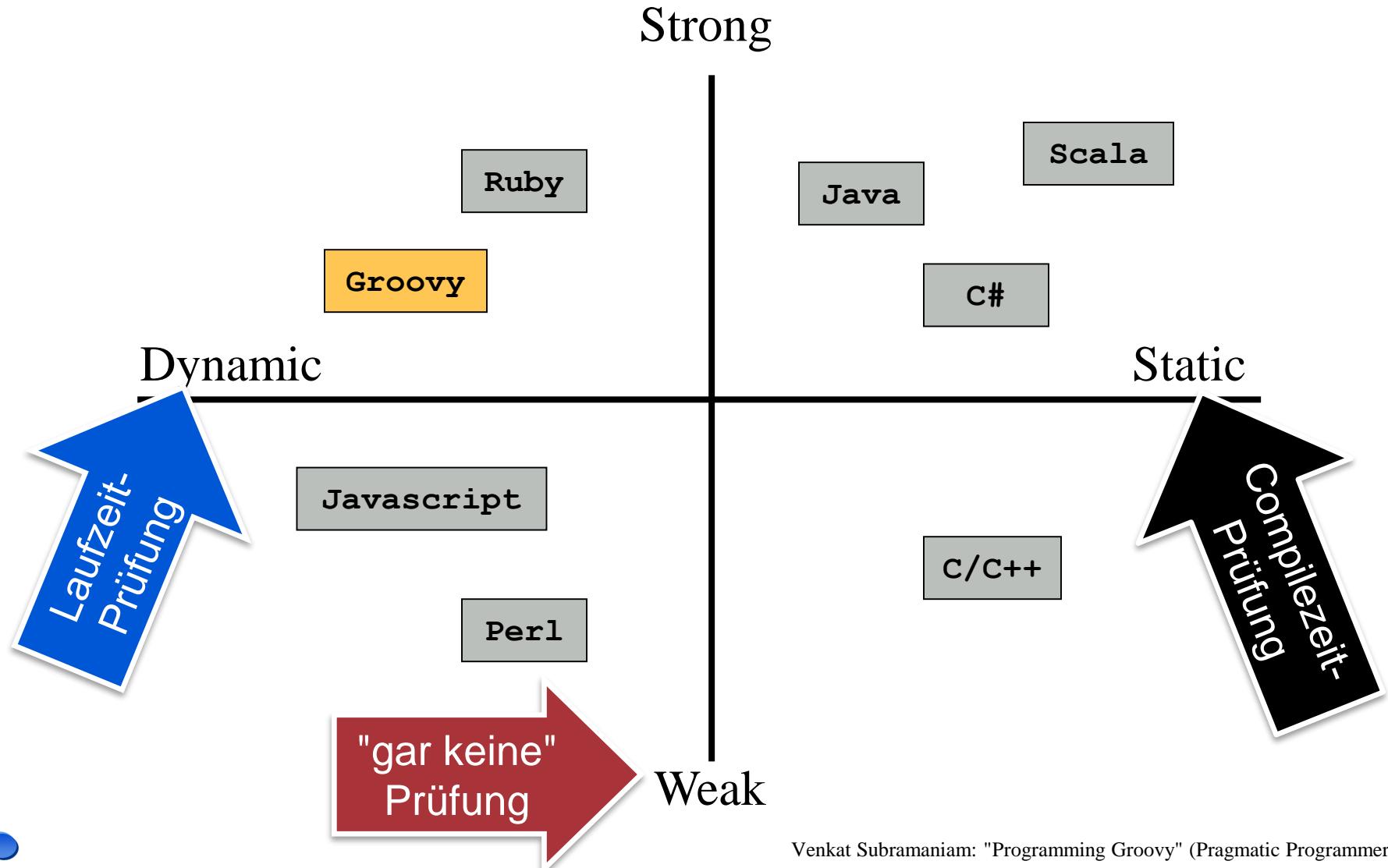
AST-Transformation

Traits



Foto von [Unsplash](#), available under a [CC0 Public Domain license](#).

Weakly != Dynamic Typing



Venkat Subramaniam: "Programming Groovy" (Pragmatic Programmers)

@TypeChecked

```
class MeineKlasse {  
    def meineMethode1() {  
        // nur statisch getypter Code erlaubt  
    }  
}
```

@TypeChecked(TypeCheckingMode.SKIP)

```
def meineMethode2() {  
    // dynamisch getypter Code möglich  
}  
}
```



Sicherheitsnetz durch Tests

Chancen

Syntax wunderbar geeignet

Mocking-Framework eingebaut

DSLs: Spock, Geb, ...



Foto von carloscuellito87, available under a [CC0 Public Domain license](#).

Performance: stetige Verbesserungen

```
@TypeChecked  
@CompileStatic  
class MeineKlasse {  
    [...]  
}
```



	Fibonacci	Pi quadrature	Binary trees
Java	191 ms	97 ms	3.6 s
Groovy 2.x: Static Compilation	197 ms	101 ms	4.3 s
Groovy 1.8: Primitive optimizations	360 ms	111 ms	23,7 s
Groovy 1.7: No primitive optimizations	2590 ms	3220 ms	50 s

Konzentration auf kleine, nützliche Funktionen

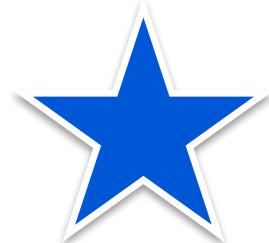


Foto von Ben Kerckx, available under a [CC0 Public Domain license](#).

Meine Top 5 Groovy Features

① Multiline Strings/GStrings

② Elvis Operator

③ Objektnavigation/Dereferenzieren

④ XyzSlurper/Parser/Builder

⑤ Power Asserts



Foto von Ben Kerckx, available under a [CC0 Public Domain license](#).

2.1

Januar 2013

2.2

November 2013

2.3

Mai 2014

2.4

Februar 2015

3.0

???

neue AST-Transformationen

2.x

Traits

2.3

Android Support

2.4

"Built-in" Lombok (schon seit 1.6)

viele fertige

@Singleton

@Immutable

@Lazy

@TypeChecked

@Compilestatic

@Synchronized

@Grab

@Synchronized

...

selbst erweiterbar

2.x

Beispiel: @Builder

@Builder

```
class Person {  
    String firstName, lastName  
    int age  
}  
  
def person = Person.builder()  
    .firstName("Dieter")  
    .lastName("Develop")  
    .age(21)  
    .build()
```

```
assert person.firstName == "Dieter"  
assert person.lastName == "Develop"
```

2.3

```
trait Fahrbar {  
    int geschwindigkeit  
    void fahren() {  
        println "Fahren mit " +  
            "${geschwindigkeit} km/h!"  
    }  
}
```


Zustand
~~Java 8~~
Verhalten
Java 8

```
class Bobbycar implements Fahrbar {}  
// Fahren mit 100 km/h  
new Bobbycar(geschwindigkeit:100).fahren()
```


2.3

Konflikte bei Mehrfachvererbung

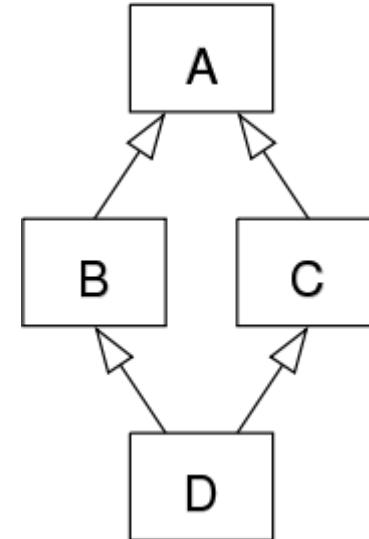
```
trait A {  
    String exec() { 'A' }  
}
```

```
trait B extends A {  
    String exec() { 'B' }  
}
```

```
trait C extends A {  
    String exec() { 'C' }  
}
```

```
class D implements B, C {}
```

```
def d = new D()  
assert d.exec() == 'C'
```



Last wins!

2.3

Manuelles Auflösen Mehrfachvererbung

```
class D implements B, C {  
    String exec() { B.super.exec() }  
}  
  
def d = new D()  
assert d.exec() == 'B'
```

2.3

schlanker weniger Boilerplate-Code als mit Java

prägnantere Syntax als Java

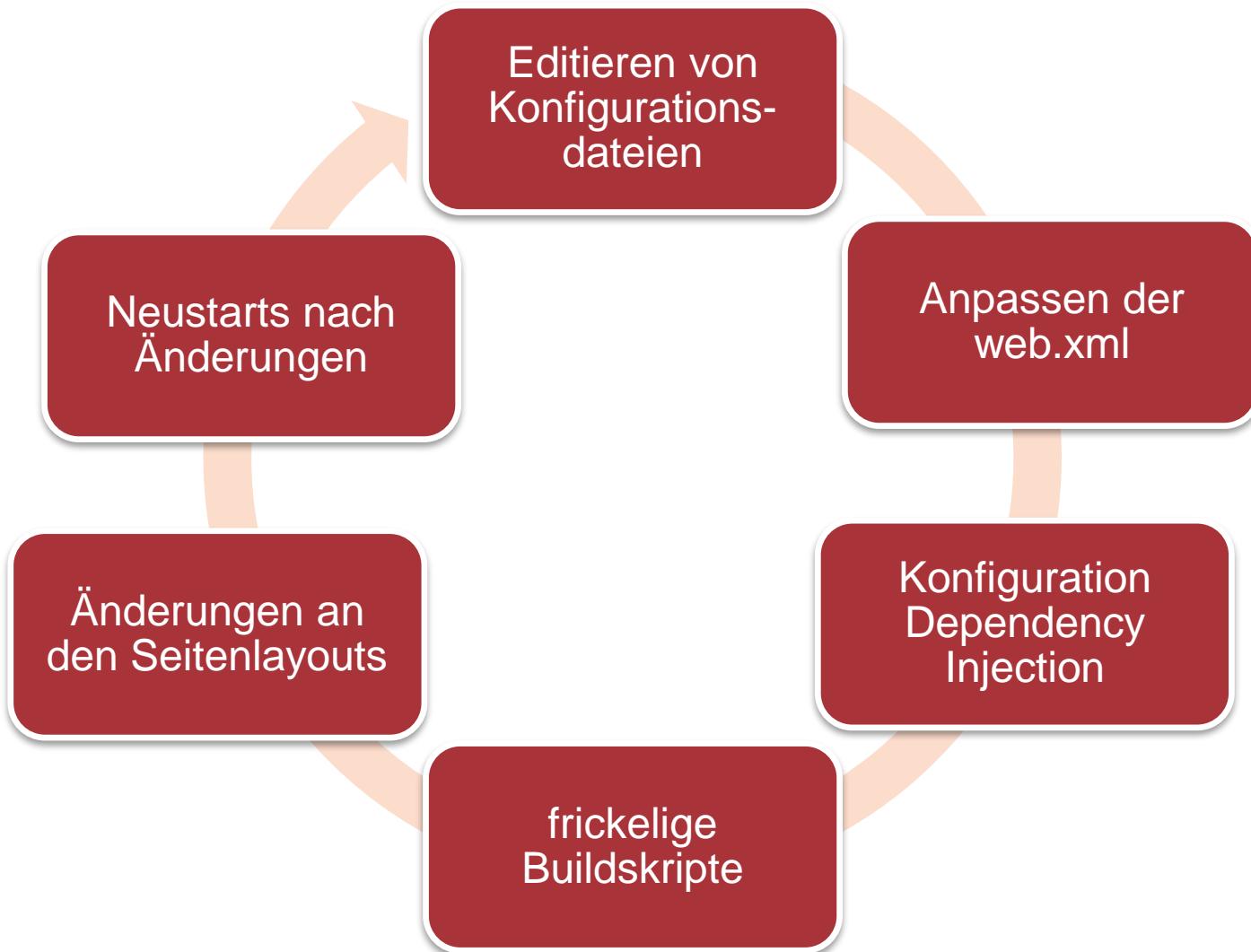
SwissKnife-Bibliothek (AST-Transformationen)

2.4



Aber mittlerweile
eigenständiges,
gestandenes
Framework!

Inspiriert durch
Ruby on Rails



Hauptprinzipien von Grails

1



Produktivität

2



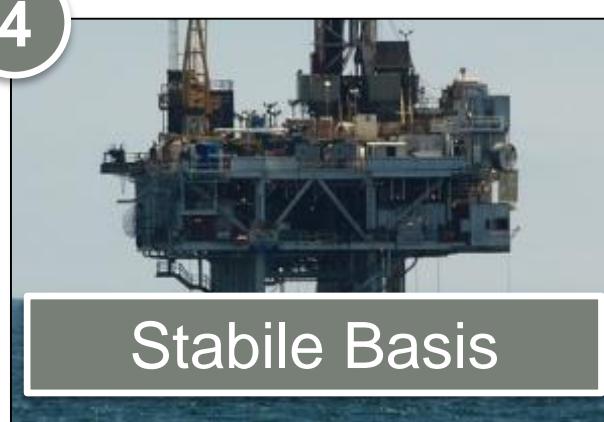
Einfachheit

3

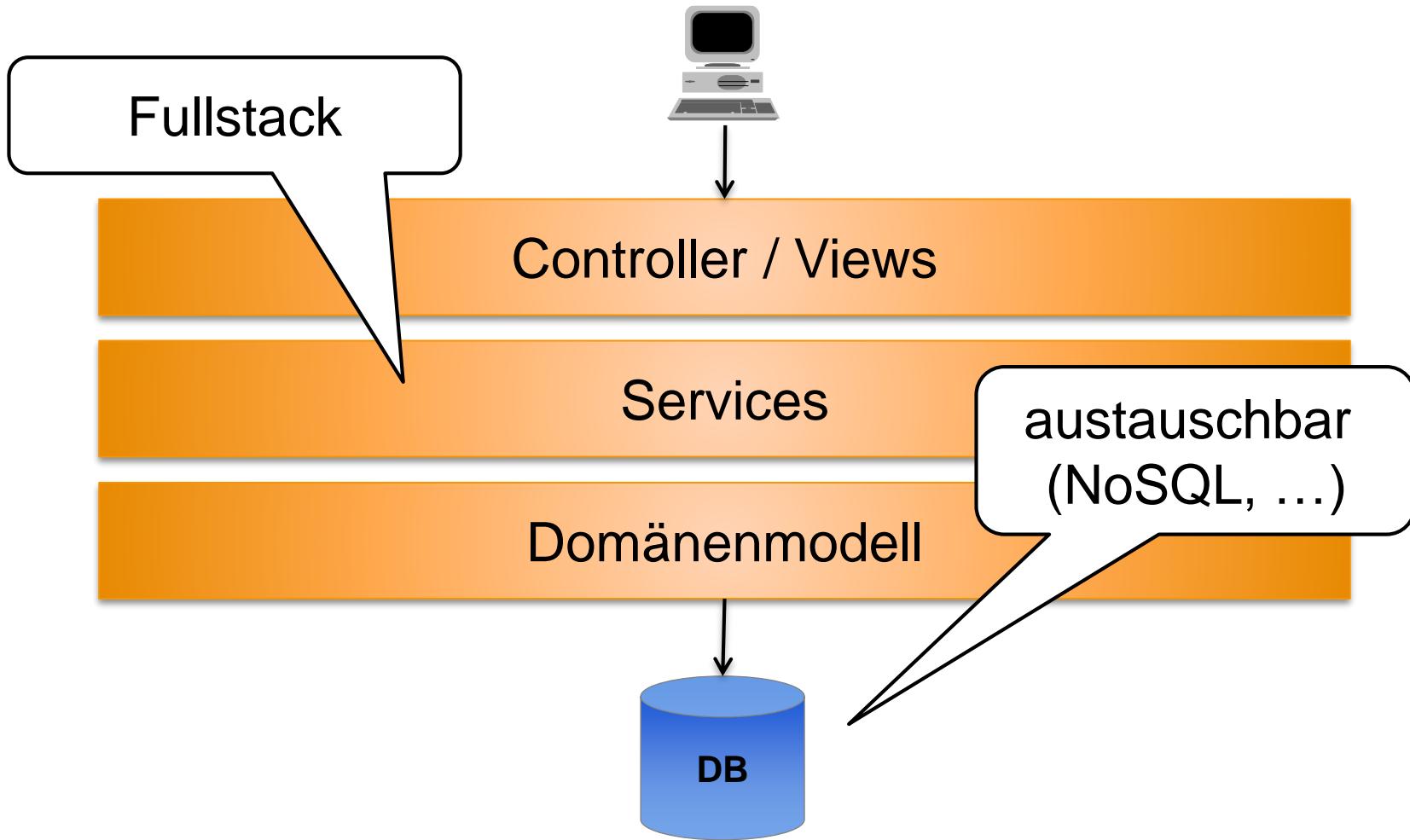


Erweiterbarkeit

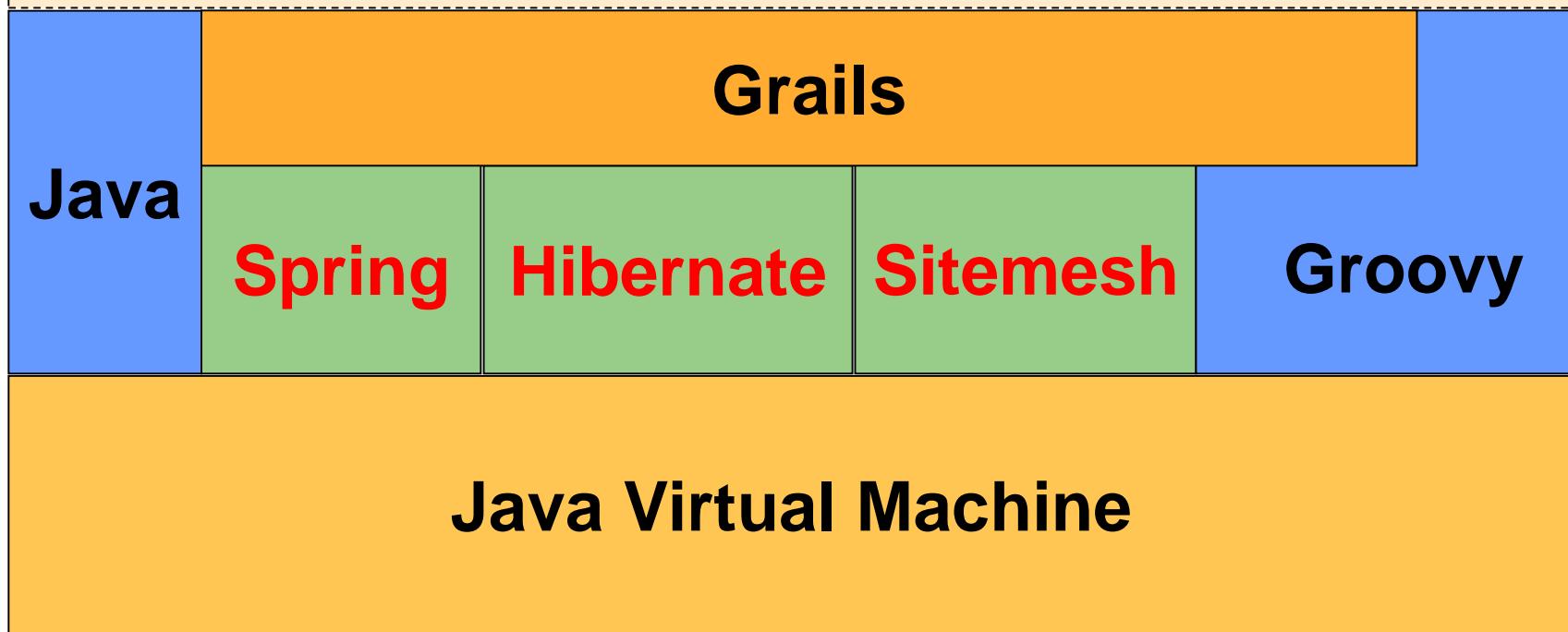
4



Stabile Basis



Anwendung



Problematisch in großen Projekten 1

Abwärtskompatibilität/Upgrades 2

Aktualität der Plugins 3

Stacktraces 4

schwierig bei Misch-Masch
von Java und Groovy- Entwicklern

Festlegen projektübergreifender Code-Conventions

hohe Testabdeckung

80 % funktionieren in Grails super (CRUD, ...)
20 % sind umständlicher, aber nicht unlösbar

80/20 Regel

1

leider ja

Aber es gibt einen Migrationsguide!

Möglichst frühzeitig upgraden!

2

Plugins der Core-Entwickler sind gut

- werden gewartet
- schnelle Upgrades bei neuen Grails-Versionen

Hände weg von 3rd-Party-Plugins ...

- mit vielen offenen Bugs
- ohne nennenswerte Aktivitäten

Grails 3 macht einige Plugins obsolet

3

Administrator: C:\Windows\System32\cmd.exe - grails

Stacktraces



Administrator: C:\Windows\System32\cmd.exe - grails

c:\ Administrator: C:\Windows\System32\cmd.exe - grails

© 2015 Orientation in Objects GmbH

Groovy und Grails – Quo vadis?

Meine Top 5 Grails Features



like

① Automatische DI

② Validation

③ Tag Libs

④ CRUD-Methoden

⑤ Where Queries

Foto von [geralt](#), available under a [CC0 Public Domain license](#).

2.2

Dezember 2012

2.3

September 2013

2.4

Mai 2014

2.5

3.0

März 2015

3.1

???

Basis Spring Boot

3.0

Interceptor API

3.0

Anwendungsprofile

3.0

Gradle als Buildsystem

3.0

API Redesign mit Traits

3.0

Spring 4.1 + Spring Boot 1.2



lauffähige JAR (Container eingebettet)

ohne Container start- und debugbar

keine IDE-Unterstützung notwendig

Interceptor API löst Grails Filter ab

- moderner, basiert auf Traits
- Interceptor erbt Funktionalität über Trait
- 3 Callbacks: before, after, afterView
- Convention over Configuration: Namenskonvention
 - BookInterceptor für BookController

- ähnlich Java EE Profilen (Web, Full, ...)
- Profil kapselt die Anwendungsstruktur
 - Kommandos
 - Plugins
 - Skeletons, Templates
 - Ressourcen

```
grails create-app myapp --profile=web-plugin
```

- Default: Web-Profil
 - Projektstruktur für Webanwendung

- Verwaltung in einem Repository (USER_HOME/.grails/repository)
- Profil = Verzeichnis mit folgender Struktur

```
web
  * commands
    * create-controller.yml
    * run-app.groovy
    ...
  * skeleton
    * grails-app
      * controllers
      ...
      * build.gradle
  * templates
    * artifacts
      * Controller.groovy
  * profile.yml
```

Kein selbstgeschriebenes Buildsystem mehr

frickeliges Gant und Ivy sind Geschichte
Build-Integration in bestehende Projektlandschaft wird einfacher

Keine IDE mit Grails-Plugins mehr nötig

nur Gradle-Support notwendig
theoretisch reichen Sublime/Atom/Vi/Emacs/... plus Kommandozeile

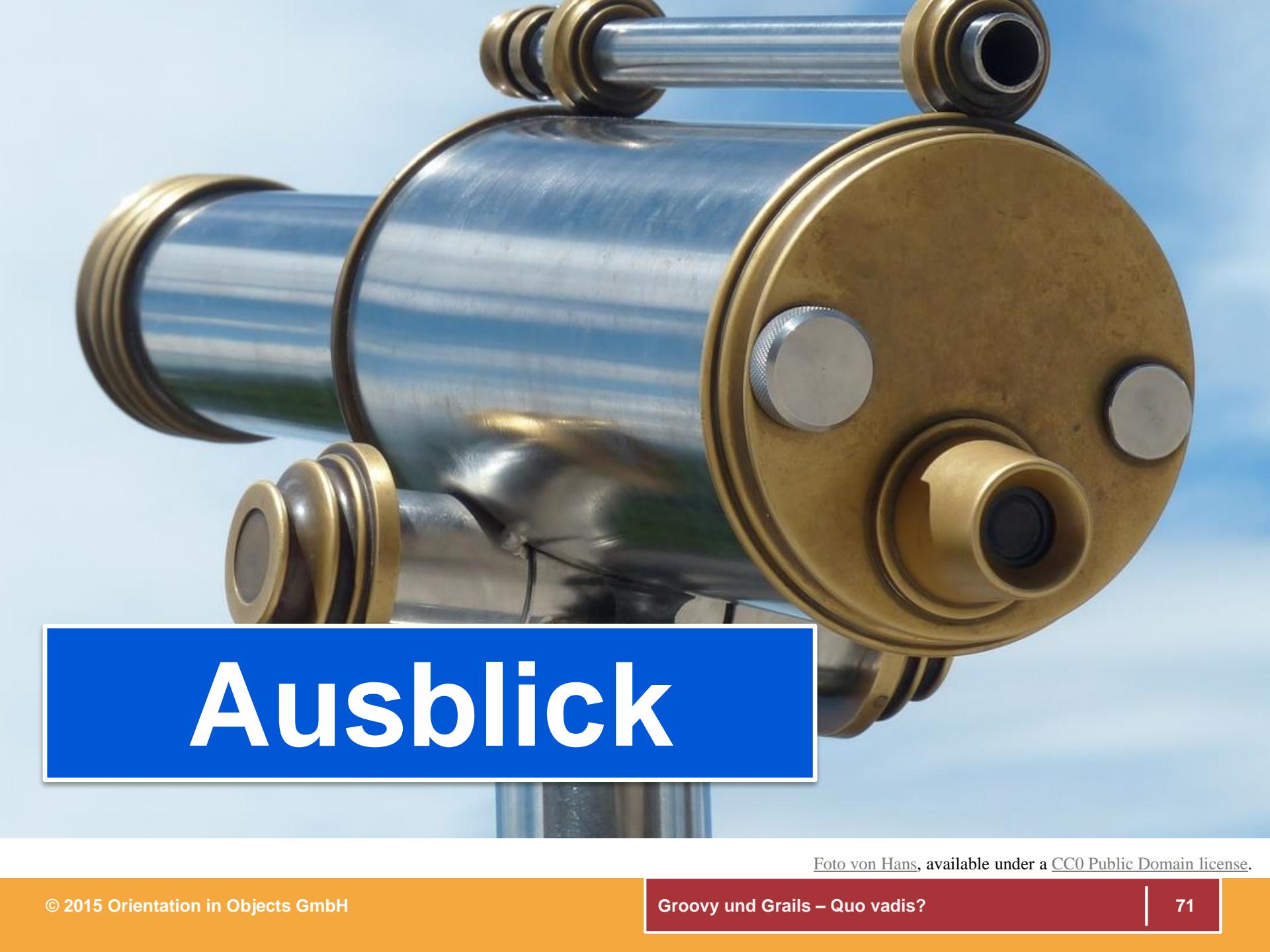


Compieltime-Metaprogrammierung

Stabilität

weiterhin Flexibilität

Schnittstellen aufgeräumt (grails.* vs. org.grails.*)



Ausblick

[Foto von Hans](#), available under a [CC0 Public Domain license](#).



Schon reif für den Java Framework Friedhof?

Foto von PublicDomainPictures, available under a [CC0 Public Domain license](#).



Aussage vom Groovy-Projektleiter



<https://speakerdeck.com/glaforge/groovy-state-of-the-union-gr8conf-europe-2015>



Stärken und Grenzen von Groovy und Grails kennen

Wahl haben – bewusst entscheiden

[Foto von Efraimstochter](#), available under a [CC0 Public Domain license](#).

Scripting

Testen (Spock, Geb, ...)

DSLs (Gradle, ...)

Admin-Konsole (Java EE Apps)

Grails

Prototyping

(kleine) Intranetanwendungen

RIA/SPA (GWT, AngularJS) mit Grails Backend

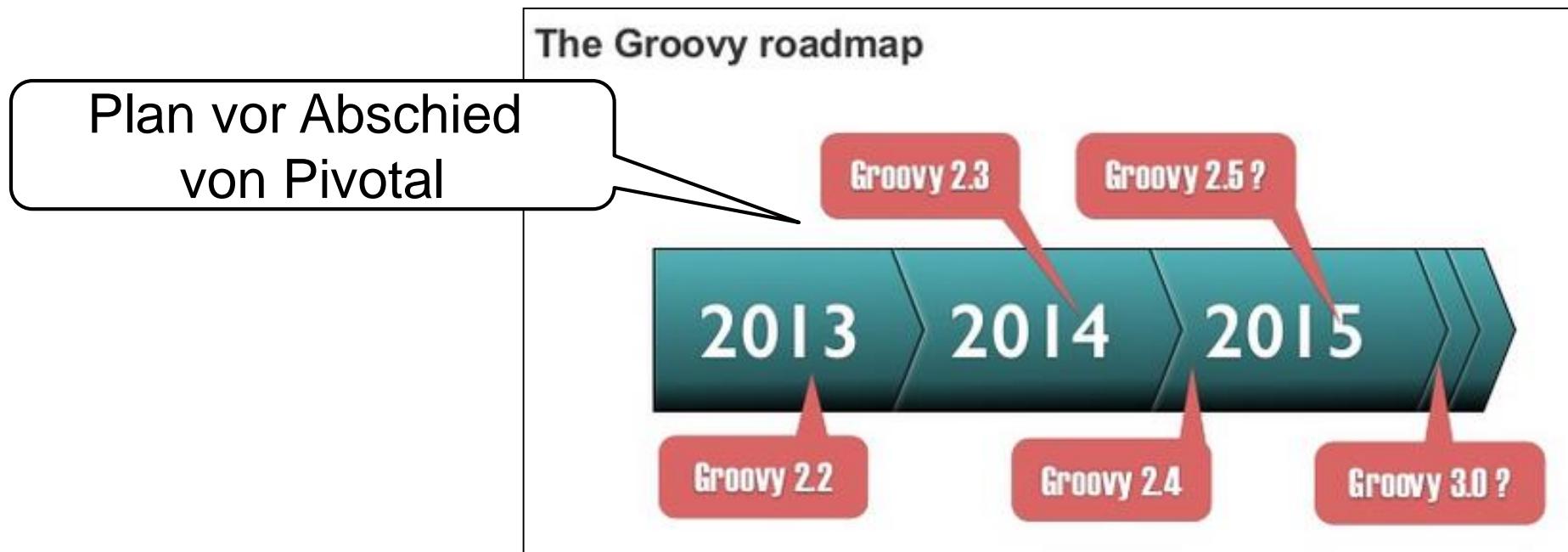
Microservices

Roadmaps



[Foto von fancycrave1](#) available under a [CC0 Public Domain license](#).

- neues Meta-Object Protokoll
- Laufzeit-Metaprogrammierung auf Basis von Invoke Dynamic
- Sprachgrammatik neu in Antlr v4



<http://de.slideshare.net/SpringCentral/groovy-in-2014andbeyond>

erstes Release nach Apache Richtlinien

- 2.4.4 vom 16.07.2015



(enthält wichtiges Sicherheitsupdate)

kleinere Änderungen für 2.5 geplant

- AST-Transformationen
- GDK-Verbesserungen

Rewrite MOP in 3.0



[Foto von skeeze](#), available under a [CC0 Public Domain license](#).

The background image shows a path through a dense forest. The scene is heavily shrouded in fog or mist, which obscures the background and creates a sense of depth. Bare, skeletal tree branches are silhouetted against the light, particularly on the left side. The ground appears to be a mix of dirt and fallen leaves.

eigentlich angekündigt für Juni 2015



[Foto von Unsplash](#), available under a [CC0 Public Domain license](#).

Roadmap Grails – es tut sich etwas ...



Steve Bowman

@sbowman96



+ Follow

Cool hearing Graeme speak about Grails
3.0 & 3.1 roadmap in STL w/key OCI client
@ObjectComputing @grailsframework



RETWEETS
5

FAVORITES
5



11:42 PM - 22 Jul 2015

3.1 (Q1 '16)

Profile-Support ausbauen Repos, AngularJS-Profil (SPA)

REST-Support ausbauen REST-Profil, JSON-Erweiterungen

Hibernate 5, MongoDB 3.0, Neo4j 2.3 in GORM

3.2 (Q2 '16)

Non Blocking GORM

Lightweight GORM-SQL, GORM Rest

3.3 (Q2 '17)

Non Blocking IO Netty

Postgres Async support

OCI gibt Gas



oci @ObjectComputing · 16. Apr.

Our very own [@jeffscottbrown](#) is speaking at [#GIDS2015](#). Check out updates [@greatindiandev](#)



oci @ObjectComputing · 1. Juni

We are pleased to announce that [@daveklein](#) & [@colinharrington](#) have joined The Grails Team at OCI. Welcome! [#grailsfw](#)

Sponsor und Auftritte auf Konferenzen

Anstellung von weiteren Leuten



oci

@ObjectComputing



Following

Would you like to join The Grails Team at
OCI? Project work, plugins, GORM, core,
etc. - Send resume to
grailsjobs@ociweb.com. #grailsfw

RETWEETS
18

FAVORITES
12



6:06 PM - 3 Aug 2015

*Sie suchen
noch Leute*

Groovy und Grails

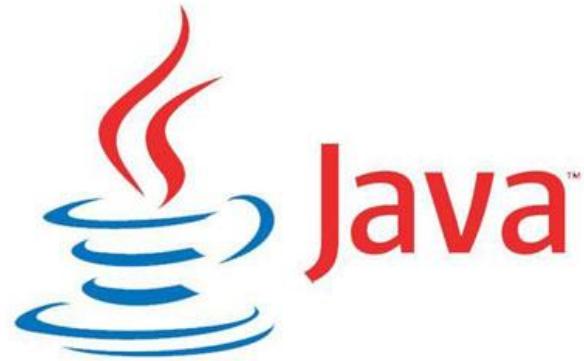
werden weiter leben,

nicht zuletzt wegen ...

Großes Ökosystem



Groovy als Treiber für andere Sprachen, z. B.



Konferenzen

- GR8Conf Europe (Kopenhagen) + GR8Conf US
- Greach (Spanien)
- SpringOne 2GX

Podcasts

- <http://groovypodcast.podbean.com/>

Weekly Newsletter

- <http://www.groovy-lang.org/groovy-weekly.html>

22,350
questions tagged

grails

Stackoverflow und aktive Mailinglisten

Synonyms

groovygrails

Können sich diese Firmen irren?

They all use Groovy!



<http://groovy-lang.org/>

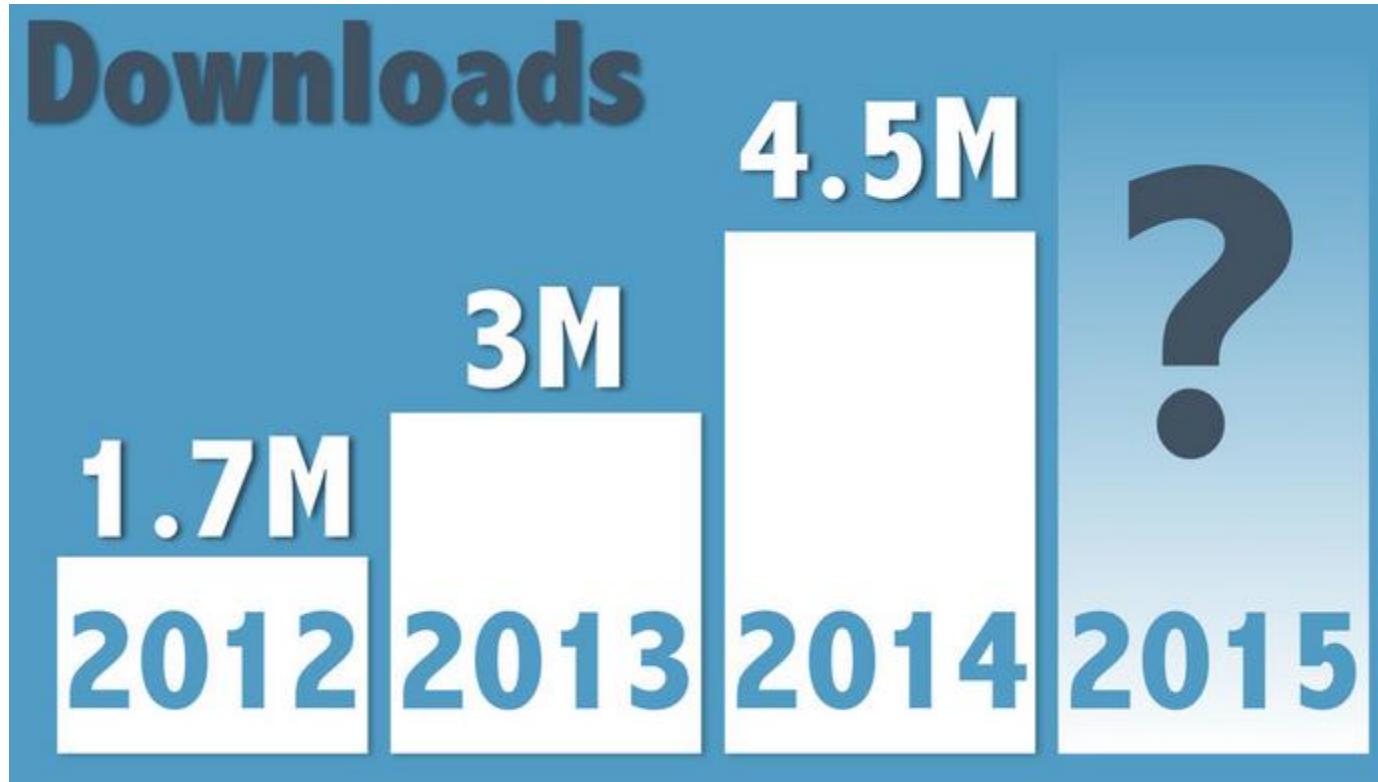
Studie von ZeroTurnaround/RebelLabs
von 2013/2014 ?

IN THE END
GRAILS & VAADIN
ARE THE BIG WINNERS,

FOLLOWED REASONABLY CLOSELY BY GWT, JSF, AND PLAY.

THERE YOU HAVE IT. NOW YOU CAN GO HOME ;-)

<http://de.slideshare.net/hamedhatami2012/curious-coders-java-web-frameworks-comparison>



<https://speakerdeck.com/glaforge/groovys-history-and-current-status>

- nur Maven Central
- es fehlen Codehaus + Bintray

**2.1 MILLIONS
FOR FIRST 4 MONTHS
OF 2015**

<https://speakerdeck.com/glaforge/groovys-history-and-current-status>

Explodierende Downloadzahlen



Guillaume Laforge

@glaforge



Following

Interesting stat of the day: in 6 months, **#groovylang** has been downloaded 4.5M times, as much as the whole year of 2014!



Eberhard Wolff @ewolff · Jul 4

@glaforge wow - any idea why?



[View other replies](#)



Guillaume Laforge @glaforge · Jul 4

@ewolff more seriously, I think the move to **#apache** helped greatly too. People have more confidence in Apache projects I think.



[View other replies](#)



Eberhard Wolff @ewolff · Jul 4

@glaforge That is the main reason you think? I thought you'd come up with some technology like Gradle, Grails, GPars etc.



Guillaume Laforge @glaforge · Jul 4

@ewolff true, and Gradle (thanks to Google and Android) is the key driver for adoption these days

**JULI
2015
4,5 MILLIONEN
DOWNLOADS**

Groovy "Rockstars"



Jens Rossa
@ReussRunner



Follow

@mittie wir sollten es sagen "einfach geil" ;)
Hat Spass gemacht die Groovy-Session!
nxt year dann evtl nicht im Keller, sondern
im Ballsaal



Dierk König @mittie · 23h
@reussrunner einfach @jaxcon sagen :-)



Jens Rossa @ReussRunner · 22h
hej @jaxcon , bitte für den groovy talk von @mittie 2016 den Ballsaal frei
machen. groovy rocks! (== im mainstream angekommen ;))



Martin @PStibbons · 22h
@ReussRunner @jaxcon Genau das! @mittie 4 Ballsaal!

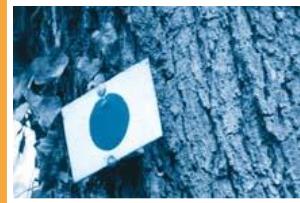


JAXCON @jaxcon · 22h
@pstibbons wir geben bei der nächsten Planung unser bestes ;)





[Foto von tpsdave](#), available under a [CC0 Public Domain license](#).



Fragen ?

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de



Orientation in Objects



**Vielen Dank für ihre
Aufmerksamkeit !**

Orientation in Objects GmbH

Weinheimer Str. 68
68309 Mannheim

www.oio.de
info@oio.de