



– Grails –  
agil und extrem

Stefan Rook  
[stefan.roock@akquinet.de](mailto:stefan.roock@akquinet.de)  
Bernd Schiffer  
[bernd.schiffer@akquinet.de](mailto:bernd.schiffer@akquinet.de)

11.12.2007  
XPUG-HH



# Herzlich Willkommen!



## Bernd Schiffer

- akquinet AG
- Softwareentwickler
- XP, Scrum, TDD, Groovy, Grails



## Stefan Rook

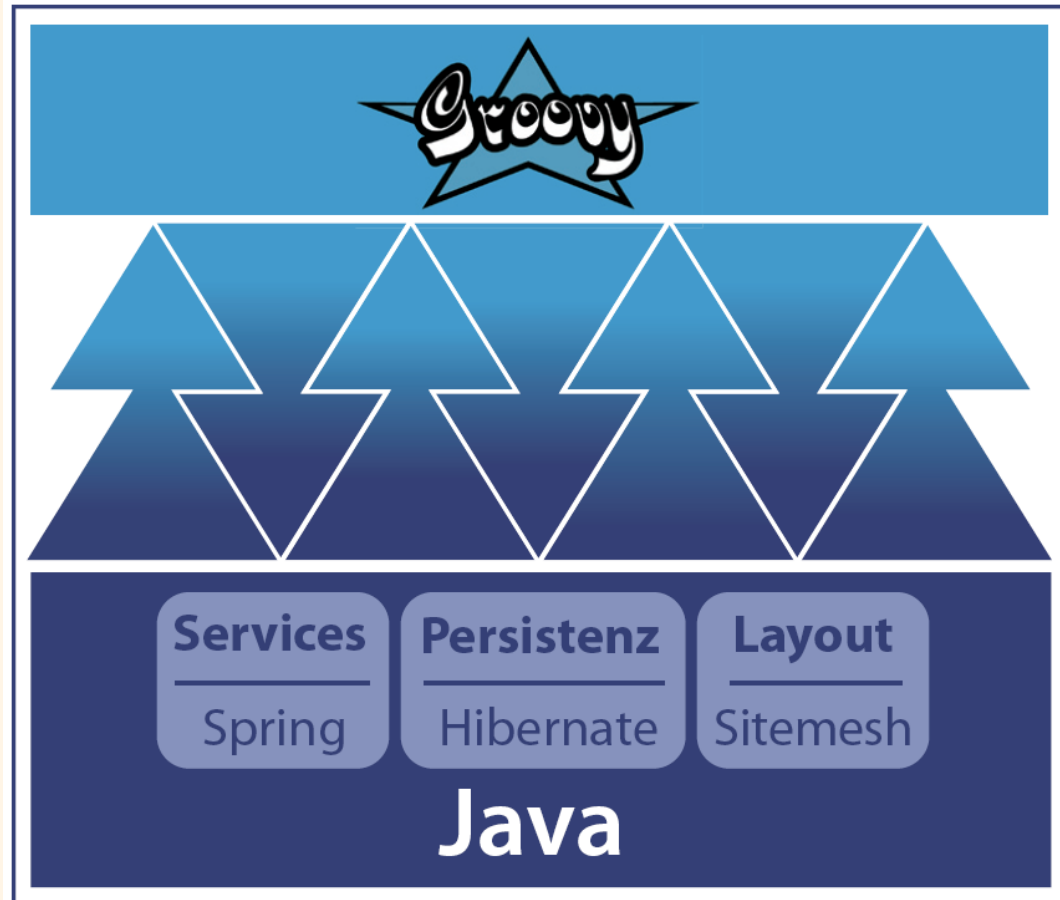
- akquinet AG
- Coach für agile Methoden, Projektleiter, Softwarearchitekt
- XP, Scrum, Akzeptanztests, TDD, Refactoring ...

# Was wir heute zeigen

- Vorstellung von Grails
- Live-Demo: Wir basteln uns eine Abendveranstaltung



# Grails – Der Zuckerguss 1/2



# Grails – Der Zuckerguss 2/2



**Grails-Prinzipien:**

- Konvention statt Konfiguration
- DRY: Don't Repeat Yourself

## Ajax

Dojo

Prototype

Yahoo UI

...

## Batch

Quarz

## Sicherheit

Acegi

...

## Testen

JUnit

Canoo  
Web Test

...



# Groovy – Grails' Dynamik 1/2

- Literale

```
def list = [1, 2, 3, 4]
def map = [a:1, b:2]
```

- GStrings

```
def vorname = 'Micky'
def nachname = 'Maus'
assert 'Mein Name ist Maus, Micky Maus.' ==
"Mein Name ist $nachname, $vorname $nachname."
```

# Groovy – Grails' Dynamik 2/2

- Closures

```
def list = [1, 2, 3, 4].collect{ it > 2 }  
assert list == [3, 4]
```

- RegExp

```
'23.12.2007, 23.09.2004'.eachMatch(/(\d{2}\.\d{2}\.\d{4})/) { alles, datum ->  
    println datum  
}
```

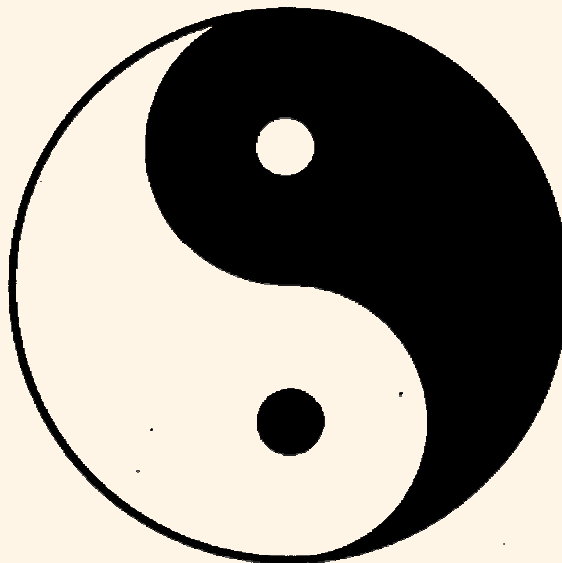
- IO

```
new File('file.txt').eachLine{  
    line -> println line  
}
```

23.12.2007  
23.09.2004

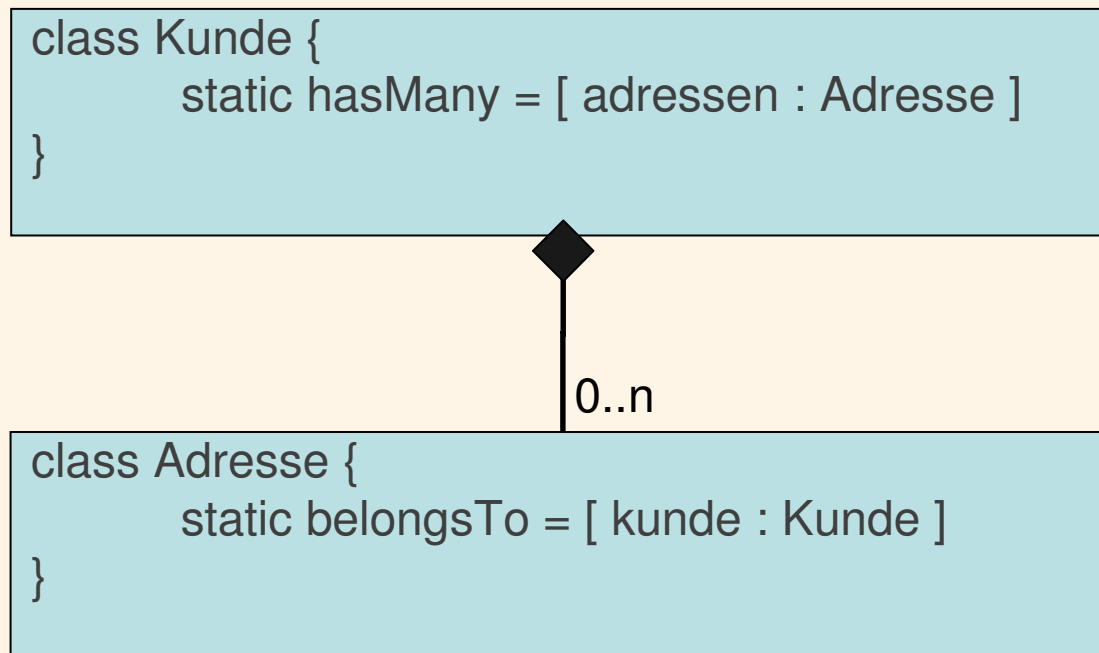
# Java – Die Basis

- Nahtlose Verschmelzung von Java und Groovy
- Nutzung aller vorhandenen Java-Ressourcen
- Objekte, Klassen, Threads, Security – keine Überraschungen



# Domain-Objects & GORM 1/2

- DOs werden automatisch gemappt (Hibernate)



# Domain-Objects & GORM 2/2

- Statische Typisierung zwecks Mapping

```
class Adresse {  
    String strasse  
    int hausnummer  
    String hausnummernzusatz  
    int postleitzahl  
    String ort  
    Date gueltigVon  
    Date gueltigBis  
}
```

# Constraints – Validierung der Domäne

```
class Adresse {  
    String strasse  
    int hausnummer  
    ...  
    Date gueltigVon  
    //...  
    static constraints = {  
        strasse(blank:false)  
        hausnummer(blank:false, min:1)  
        gueltigBis(validator: {  
            gueltigBis == null || gueltigVon <= gueltigBis  
        })  
    }  
}
```

# Dynamic Finder – jetzt wird's interessant

```
Adresse.findAllByPostleitzahl(21502)
```

```
Adresse.findAllByPostleitzahlBetween(21000, 21999)
```

```
Adresse.findAllByPostleitzahlAndGueltigBis(21502, null)
```

```
Adresse.findAllByPostleitzahlBetween(21000, 21999,  
    [sort:'postleitzahl', order:'desc'])
```

```
Adresse.list(max:10, offset:100, sort:'postleitzahl', order:'asc')
```

# Views – Wir wollen was sehen!

```
<html>
  <head>
    <title>${kunde.name}</title>
  </head>
  <body>
    <ol>
      <g:each in="${kunde.adressen}">
        <li>${it.postleitzahl} ${it.ort}</li>
      </g:each>
    </ol>
  </body>
</html>
```

# TagLibs – Vergessen ist JSP!

```
<html>
  <head>
    <title>${kunde.name}</title>
  </head>
  <body>
    <g:liste von="${kunde.adressen}">
      ${it.postleitzahl} ${it.ort}
    </g:liste>
  </body>
</html>
```

```
class MeineTagLib {
  def liste = { attrs, body ->
    out << '<ol>'
    attrs.von.each{
      out << '<li>'
      out << body(it)
      out << '</li>'
    }
    out << '</ol>'
  }
}
```

# Controller – Model und View verbinden

```
class KundeController {  
  
    def index = { redirect action:show }  
  
    def show = { [kunde:Kunde.get(params.id)] }  
  
    def list = { [kunden:Kunde.list()] }  
  
    def save = {  
        new Kunde(params).save()  
        redirect action:list  
    }  
}
```

# AJAX – Grails' Web-2.0-Tauglichkeit

- eingebaut
  - Prototype/Script.aculo.us
  - Yahoo! UI (Plugin)
  - Dojo (Plugin)

- eingebunden

```
<g:javascript library="prototype" />
```

- benutzt

```
<div id="message"></div>  
<g:remoteLink action="delete" id="1" update="message">  
  Lösche Kunde</g:remoteLink>
```

- JSON, XML-(Un-)Marshalling

```
render kunde as JSON
```

# Plugins

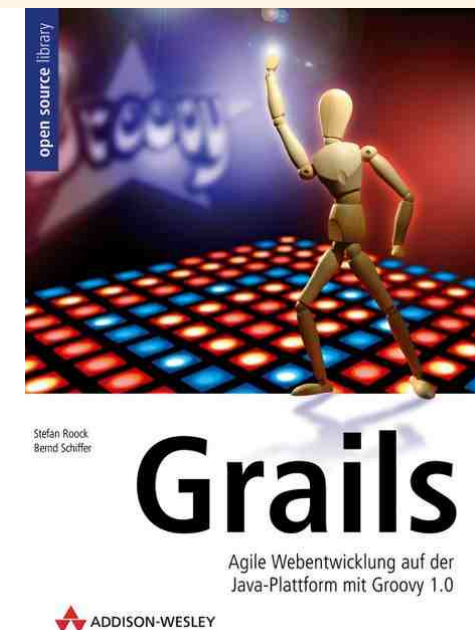
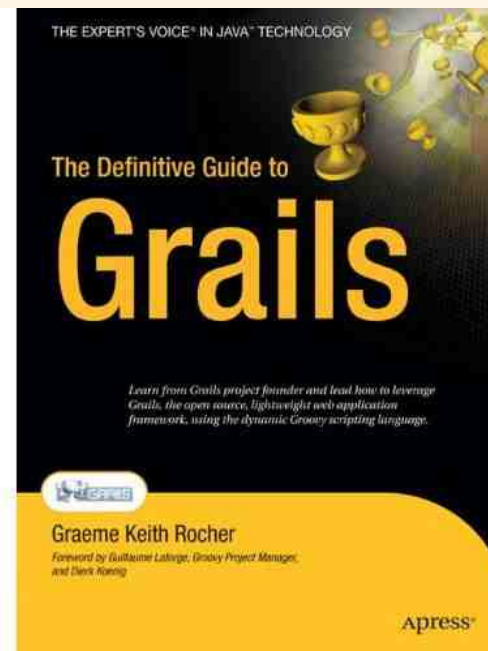
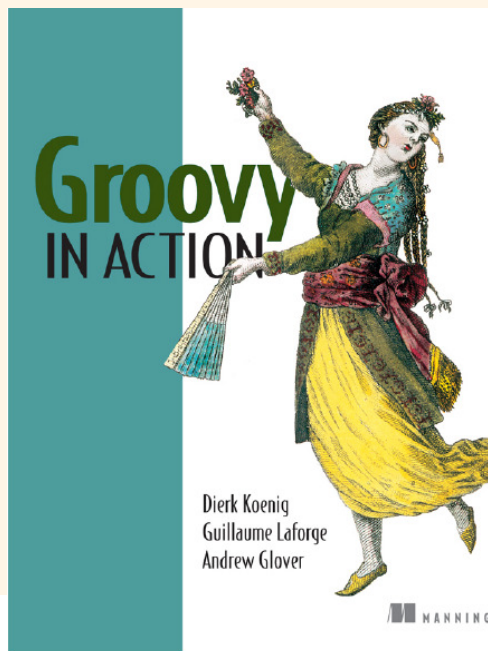
- Grails selbst aufbaut aus Plugins
- per Plugin einbindbare Techniken
  - Canoo Webtest, GWT, DWR, X-Fire, XML-RPC, Quartz, Spring Security (Acegi), JCapcha, Compass + Lucene, LiquiBase, dbmigrate, uvm.
  - <http://grails.codehaus.org/plugins>
- Modularisierung mit eigenen Plugins
  - pro Modul ein Plugin
  - technisch oder fachlich geschnitten
- Beeinflussung durch Plugins
  - Artefakte, Skripte, MetaObject-Programmierung, EventHandlerer, Bibliotheken, uvm.

# Testen

- Grails ist agil ...
  - Unittests, Integrationstests (inhärent)
  - Akzeptanztests (Plugin: Canoo Webtest)
- ... weil es Testen einbindet ...
  - Test-Skelette pro Domain-Class und Controller generiert
  - Ausführung der Tests per Skript (einschl. Reports)
- ... aber ist noch viel zu un-extrem
  - Zu langsam!

# Lust bekommen auf mehr?

- Groovy: <http://groovy.codehaus.org>
- Grails-Mailinglisten: <http://grails.codehaus.org/Mailing+lists>
- Grails: <http://grails.org/>



# Groovy-/Grails-Werbung (schamlos)

- Schulungen zu
  - Groovy: <http://www.akquinet.de/content/java-sap/landing-pages/groovy.html>
  - Grails: <http://www.akquinet.de/content/java-sap/landing-pages/grails.html>
- Beratung, Coaching, Inhouse-Schulung
- Entwicklungsunterstützung

Vielen Dank für die  
Aufmerksamkeit



<http://www.it-agile.de>

# Noch Fragen?



Schulung verlängerte Werkbank  
**agile Softwareentwicklung**  
Festpreisprojekte Coaching  
RCP **Systemintegration** Eclipse  
h3270 Hostintegration  
Scrum Refactoring testgetriebene Entwicklung  
Hibernate SAP-Netweaver **OpenSource**  
Ajax JBoss/JEMS Groovy  
 **eXtreme Programming**