



2 Problemlöser: Groovy und Grails

Thomas Westphal
25. Juni 2015

Thomas Westphal
westphal@it-factum.de



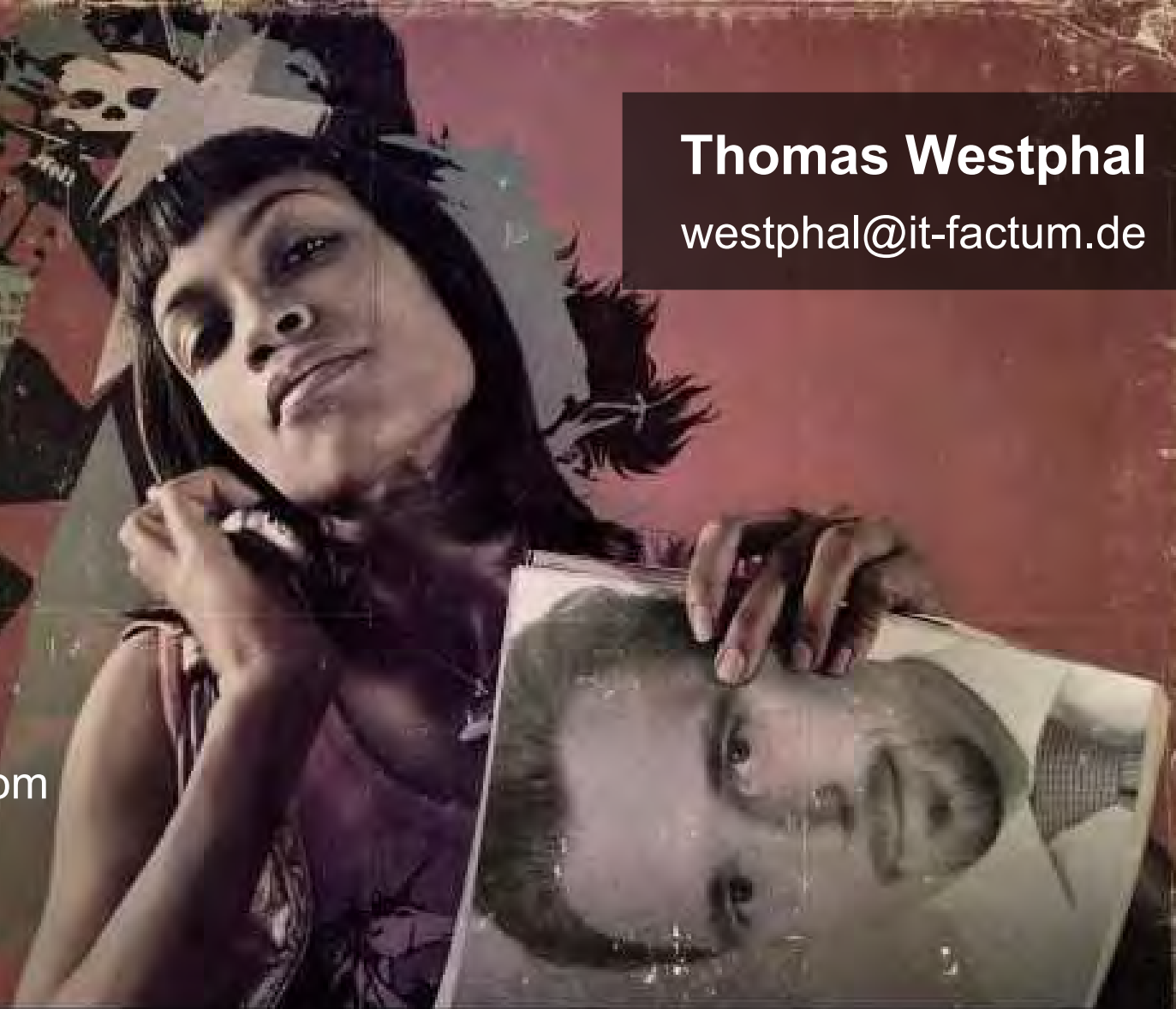
@twest72



twest-log.blogspot.com



gplus.to/twest



ROBERT RODRIGUEZ'S
**"DEATH
PROOF"**

APRIL 6, 2007

Prologue

„Why?“ (i)

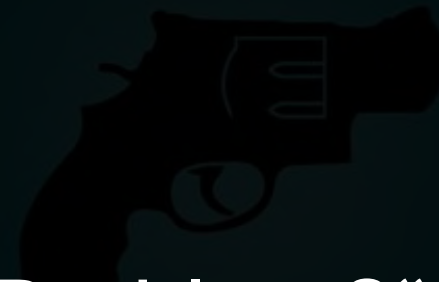




„Sie haben ein Problem?“

„Ich habe kein Problem.

Ich habe Probleme. Plural.“



Four Rooms

Welche Probleme?



- ▶ Ausdrucksschwacher Code
- ▶ Altlasten in Java
- ▶ Schnelle Releasezyklen
- ▶ Technologie-“Blumenstrauß“

Problemlöser?



Problemlöser?

„Du hast kein Problem, Jules. Ich kümmere mich um den Scheiß...“

Pulp Fiction

EXPLORER

A large red and white ship, named 'EXPLORER', is shown from a low angle, sailing on a dark, choppy sea. The ship's hull is red, and the upper part is white. The name 'EXPLORER' is written in black capital letters on the white section. The ship is tilted slightly to the right. In the background, there are dark, rocky mountains under a grey, overcast sky. The overall mood is somber and dramatic.

Problemlöser!



Episode I

„The Groovy Way“

Was ist Groovy?



Was ist Groovy?

„Groovy is like a super version of Java.“
„...If you are a developer, tester or
script guru, you have to love Groovy.“

<http://groovy.codehaus.org>



Was ist Groovy?



... ist eine dynamische Sprache für die JVM.

Was ist Groovy?



... ist eine ~~dynamische~~ Sprache für die JVM.

- ▶ dynamische und statische Typisierung

Was ist Groovy?



... eine vielschichtige Sprache für die JVM.

▶ dynamische und statische Typisierung

▶ <http://groovy-lang.org>

Was ist Groovy?



... eine vielschichtige Sprache für die JVM.

- ▶ dynamische und statische Typisierung
- ▶ Elemente aus Python, Ruby und Smalltalk
- ▶ seit 2003
- ▶ aktuelle Version 2.4.3

- ▶ <http://groovy-lang.org>

Was ist Groovy?

Vielschichtig

- ▶ Flache Lernkurve
- ▶ dynamisches und reiches Ökosystem
- ▶ Skriptfähigkeit und „testing glue“
- ▶ powerful features



Java Plattform

- ▶ läuft auf jeder JVM
- ▶ Zugriff auf jegliche Java-Bibliotheken

Wie löst Groovy Probleme?



Wie löst Groovy Probleme?

„Wenn Du es machst,
dann mach es cool!“

Pulp Fiction

Wie löst Groovy Probleme?

Java Plattform: Nahtlose Integration

Zugriff von Java auf Groovy-Objekte

Joint Compilation

Java Code ist Groovy Code!

Wie löst Groovy Probleme?

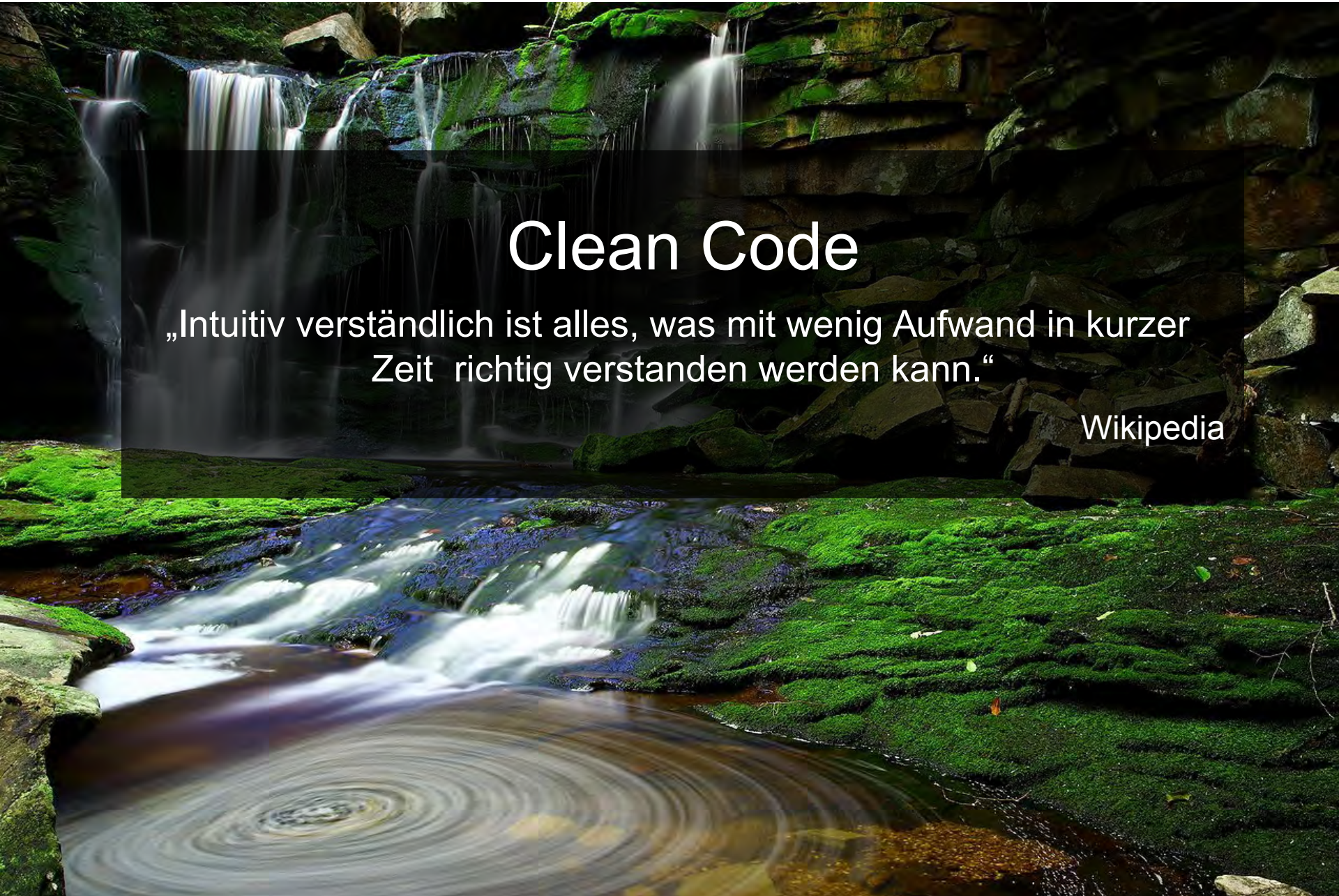


Wie löst Groovy Probleme?

Clean Code

„Intuitiv verständlich ist alles, was mit wenig Aufwand in kurzer Zeit richtig verstanden werden kann.“

Wikipedia

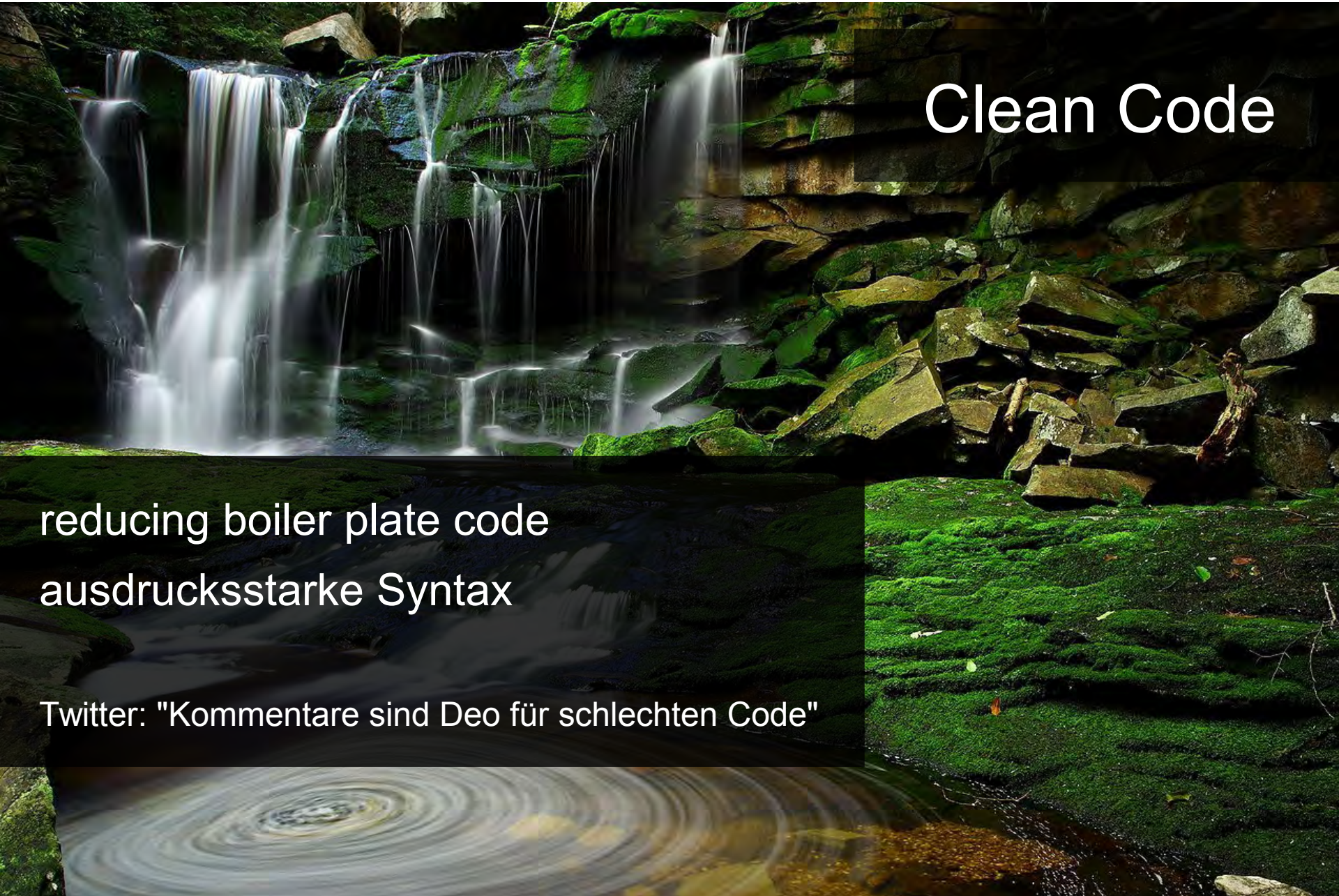


Wie löst Groovy Probleme?

Clean Code

reducing boiler plate code
ausdrucksstarke Syntax

Twitter: "Kommentare sind Deo für schlechten Code"



Eine ganz normale Java-Klasse...

```
package de.aonnet.groovy.start;  
  
public class JKontaktanfrage {}
```

...mit Imports...

```
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
import org.apache.commons.lang.builder.EqualsBuilder;
import
org.apache.commons.lang.builder.ToStringBuilder;
import org.apache.commons.lang.builder.ToStringStyle;
import org.codehaus.groovy.util.HashCodeHelper;
```

...mit Feldern...

```
private Long id;  
private Long version;  
private String unternehmen;  
private String name;  
private String vorname;  
private String ort;  
private String postleitzahl;  
private String produkt;  
private String anliegen;  
private List<String> telefonNummern = new  
ArrayList<String>();  
private Map<String, String> emailAdressen = new  
HashMap<String, String>();
```


...mit (nur) einem Konstruktor...

```
public JKontaktanfrage(Long id, Long version, String unternehmen,
String name,
    String vorname, String ort, String postleitzahl,
    String produkt, String anliegen, List<String> telefonNummern,
    Map<String, String> emailAdressen) {

    super();
    this.id = id;
    this.version = version;
    this.unternehmen = unternehmen;
    this.name = name;
    this.vorname = vorname;
    this.ort = ort;
    this.postleitzahl = postleitzahl;
    this.produkt = produkt;
    this.anliegen = anliegen;
    this.telefonNummern = telefonNummern;
    this.emailAdressen = emailAdressen;
}
```

...Getter und Setter...

```
public Long getId() {  
    return this.id;  
}
```

```
public void setId(Long id) {  
    this.id = id;  
}
```

```
public Long getVersion() {  
    return this.version;  
}
```

```
public void setVersion(Long version) {  
    this.version = version;  
}
```

```
public String getUnternehmen() {  
    return this.unternehmen;  
}
```

```
public void setUnternehmen(String unternehmen) {  
    this.unternehmen = unternehmen;  
}
```

```
public String getName() {  
    return this.name;  
}
```

...und mit Methoden

```
@Override
public int hashCode() {
    return hashCodeHelper.updateHash(hashCodeHelper.initHash(), this.id);
}

@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (obj == this) {
        return true;
    }
    if (obj.getClass() != this.getClass()) {
        return false;
    }
    final JKontaktanfrage kontaktanfrage = (JKontaktanfrage) obj;
    return new EqualsBuilder().append(this.id, kontaktanfrage.id).isEquals();
}

@Override
public String toString() {
    return ToStringBuilder.reflectionToString(this,
        ToStringStyle.SHORT_PREFIX_STYLE);
}
```


Das ist viel Quellcode!

```
package de.svenkrogh.groovy.start;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;

import org.apache.commons.lang.builder.EqualsBuilder;
import org.apache.commons.lang.builder.ToStringBuilder;
import org.apache.commons.lang.builder.ToStringStyle;
import org.codehaus.groovy.util.MultiCallback;

public class Kontaktanfrage {

    private Long id;
    private Long version;
    private String unternehmen;
    private String name;
    private String vorname;
    private String ort;
    private String produkt;
    private String postleitzahl;
    private String anliegen;
    private List<String> telefonnummern = new ArrayList<String>();
    private Map<String, String> emailAdressen = new HashMap<String, String>();

    public Kontaktanfrage(Long id, Long version, String unternehmen, String name, String vorname, String ort,
        String postleitzahl, String produkt, String anliegen, List<String> telefonnummern,
        Map<String, String> emailAdressen) {
        super();
        this.id = id;
        this.version = version;
        this.undernehmen = unternehmen;
        this.name = name;
        this.vorname = vorname;
        this.ort = ort;
        this.postleitzahl = postleitzahl;
        this.produkt = produkt;
        this.anliegen = anliegen;
        this.telefonnummern = telefonnummern;
        this.emailAdressen = emailAdressen;
    }

    public Long getId() {
        return this.id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Long getVersion() {
        return this.version;
    }

    public void setVersion(Long version) {
        this.version = version;
    }

    public String getUnternehmen() {
        return this.undernehmen;
    }

    public void setUnternehmen(String unternehmen) {
        this.undernehmen = unternehmen;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getVorname() {
        return this.vorname;
    }

    public void setVorname(String vorname) {
        this.vorname = vorname;
    }

    public String getOrt() {
        return this.ort;
    }

    public void setOrt(String ort) {
        this.ort = ort;
    }

    public String getPostleitzahl() {
        return this.postleitzahl;
    }

    public void setPostleitzahl(String postleitzahl) {
        this.postleitzahl = postleitzahl;
    }

    public String getProdukt() {
        return this.produkt;
    }

    public void setProdukt(String produkt) {
        this.produkt = produkt;
    }

    public String getAnliegen() {
        return this.anliegen;
    }

    public void setAnliegen(String anliegen) {
        this.anliegen = anliegen;
    }

    public List<String> getTelefonnummern() {
        return this.telefonnummern;
    }

    public void setTelefonnummern(List<String> telefonnummern) {
        this.telefonnummern = telefonnummern;
    }

    public Map<String, String> getEmailAdressen() {
        return this.emailAdressen;
    }

    public void setEmailAdressen(Map<String, String> emailAdressen) {
        this.emailAdressen = emailAdressen;
    }

    @Override
    public int hashCode() {
        return hashCodeBuilder().append(this.id).append(this.ort).append(this.name).append(this.vorname).append(this.postleitzahl).append(this.produkt).append(this.anliegen).append(this.undernehmen).append(this.version).append(this.id).hashCode();
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (obj == this) {
            return true;
        }
        if (obj.getClass() != this.getClass()) {
            return false;
        }
        final Kontaktanfrage kontaktanfrage = (Kontaktanfrage) obj;
        return new EqualsBuilder().append(this.id, kontaktanfrage.getId()).append(this.ort, kontaktanfrage.getOrt()).append(this.name, kontaktanfrage.getName()).append(this.vorname, kontaktanfrage.getVorname()).append(this.postleitzahl, kontaktanfrage.getPostleitzahl()).append(this.produkt, kontaktanfrage.getProdukt()).append(this.anliegen, kontaktanfrage.getAnliegen()).append(this.undernehmen, kontaktanfrage.getUnternehmen()).append(this.version, kontaktanfrage.getVersion()).append(this.id, kontaktanfrage.getId()).isEquals();
    }

    @Override
    public String toString() {
        return ToStringBuilder.reflectionToString(this, ToStringStyle.SHORT_PREFIX_STYLE);
    }
}
```

JKontaktanfrage.java



JKontaktanfrage.groovy

Was haben wir da?

Viel Gepäck!



Was kann gespart werden?



Tuning!

Was kann gespart werden?

```
package de.aonnet.groovy.start;
```

```
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;
```

```
import org.apache.commons.lang.builder.EqualsBuilder;  
import org.apache.commons.lang.builder.ToStringBuilder;  
import org.apache.commons.lang.builder.ToStringStyle;  
import org.codehaus.groovy.util.HashcodeHelper;
```

```
public class JKontaktanfrage {
```

```
...
```

```
}
```

Standard-Imports:

- ▶ java.io.*
- ▶ java.lang.*
- ▶ java.net.*
- ▶ java.math.BigDecimal
- ▶ java.math.BigInteger
- ▶ java.util.*
- ▶ groovy.lang.*
- ▶ groovy.util.*

Public per Default:

- ▶ Klassen
- ▶ Methoden

Was kann gespart werden?

```
private Long id;
private Long version;
private String unternehmen;
private String name;
private String vorname;
private String ort;
private String postleitzahl;
private String produkt;
private String anliegen;
private List<String> telefonNummern = new ArrayList<String>();
private Map<String, String> emailAdressen = new HashMap<String, String>();
```

- ▶ **ohne Modifier für Sichtbarkeit = Groovy-Property**
- ▶ **Semikolon ist optional**

Was kann gespart werden?

```
public JKontaktanfrage(Long id, Long version, String unternehmen, String name,  
    String vorname, String ort, String postleitzahl,  
    String produkt, String anliegen, List<String> telefonNummern,  
    Map<String, String> emailAdressen) {  
  
    super();  
    this.id = id;  
    this.version = version;  
    this.unternehmen = unternehmen;  
    this.name = name;  
    this.vorname = vorname;  
    this.ort = ort;  
    this.postleitzahl = postleitzahl;  
    this.produkt = produkt;  
    this.anliegen = anliegen;  
    this.telefonNummern = telefonNummern;  
    this.emailAdressen = emailAdressen;  
}
```

Konstruktor

- ▶ Default Konstruktor mit Named Parameter

Was kann gespart werden?

```
@Override
public int hashCode() {
    return hashCodeHelper.updateHash(hashCodeHelper.initHash(), this.id);
}

@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (obj == this) {
        return true;
    }
    if (obj.getClass() != this.getClass()) {
        return false;
    }
    final JKontaktanfrage kontaktanfrage = (JKontaktanfrage) obj;
    return new EqualsBuilder().append(this.id, kontaktanfrage.id).isEquals();
}

@Override
public String toString() {
    return ToStringBuilder.reflectionToString(this,
        ToStringStyle.SHORT_PREFIX_STYLE);
}
```

Was kann gespart werden?

```
@Override
public int hashCode() {
    return hashCodeHelper.updateHash(hashCodeHelper.initHash(), this.id);
}

@Override
public boolean equals(Object obj) {
    if (obj == null) {
        return false;
    }
    if (obj == this) {
        return true;
    }
    if (obj.getClass() != this.getClass()) {
        return false;
    }
    final JKontaktanfrage kontaktanfrage = (JKontaktanfrage) obj;
    return new EqualsBuilder().append(this.id, kontaktanfrage.id).isEquals();
}

@Override
public String toString() {
    return ToStringBuilder.reflectionToString(this,
        ToStringStyle.SHORT_PREFIX_STYLE);
}
```

Die Logik nicht!!!

Das ist Groovy!!!

```
package de.aonnet.groovy.start;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Maps;

import org.apache.commons.lang.builder.ToStringBuilder;
import org.apache.commons.lang.builder.ToStringStyle;
import org.apache.commons.lang.builder.ToStringStyle;
import org.codehaus.groovy.util.HashCodeHelper;

public class GKontaktanfrage {

    private Long id;
    private Long version;
    private String unternehmen;
    private String name;
    private String vorname;
    private String ort;
    private String postleitzahl;
    private String anliegen;
    private List<String> telefonNummern = new ArrayList<String>();
    private Map<String, String> emailAdressen = new HashMap<String, String>();

    public GKontaktanfrage(Long id, Long version, String unternehmen, String name, String vorname, String ort,
        String postleitzahl, String produkt, String anliegen, List<String> telefonNummern,
        Map<String, String> emailAdressen) {
        super();
        this.id = id;
        this.version = version;
        this.unternehmen = unternehmen;
        this.name = name;
        this.vorname = vorname;
        this.ort = ort;
        this.postleitzahl = postleitzahl;
        this.prdukt = produkt;
        this.anliegen = anliegen;
        this.telefonNummern = telefonNummern;
        this.emailAdressen = emailAdressen;
    }

    public Long getId() {
        return this.id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Long getVersion() {
        return this.version;
    }

    public void setVersion(Long version) {
        this.version = version;
    }

    public String getUnternehmen() {
        return this.unternehmen;
    }

    public void setUnternehmen(String unternehmen) {
        this.unternehmen = unternehmen;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getVorname() {
        return this.vorname;
    }

    public void setVorname(String vorname) {
        this.vorname = vorname;
    }

    public String getOrt() {
        return this.ort;
    }

    public void setOrt(String ort) {
        this.ort = ort;
    }

    public String getPostleitzahl() {
        return this.postleitzahl;
    }

    public void setPostleitzahl(String postleitzahl) {
        this.postleitzahl = postleitzahl;
    }

    public String getProdukt() {
        return this.prdukt;
    }

    public void setProdukt(String produkt) {
        this.prdukt = produkt;
    }

    public String getAnliegen() {
        return this.anliegen;
    }

    public void setAnliegen(String anliegen) {
        this.anliegen = anliegen;
    }

    public List<String> getTelefonnummern() {
        return this.telefonNummern;
    }

    public void setTelefonnummern(List<String> telefonNummern) {
        this.telefonNummern = telefonNummern;
    }

    public Map<String, String> getEmailAdressen() {
        return this.emailAdressen;
    }

    public void setEmailAdressen(Map<String, String> emailAdressen) {
        this.emailAdressen = emailAdressen;
    }

    @Override
    public int hashCode() {
        return HashCodeHelper.updateHash(HashCodeHelper.initHash(), this.id);
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (obj == this) {
            return true;
        }
        if (obj.getClass() != this.getClass()) {
            return false;
        }
        final GKontaktanfrage kontaktfrege = (GKontaktanfrage) obj;
        return new EqualsBuilder().append(this.id, kontaktfrege.getId()).isEquals();
    }

    @Override
    public String toString() {
        return ToStringBuilder.reflectionToString(this, ToStringStyle.SHORT_PREFIX_STYLE);
    }
}
```

```
package de.aonnet.groovy.start
```

```
import org.apache.commons.lang.builder.EqualsBuilder
import org.apache.commons.lang.builder.ToStringBuilder
import org.apache.commons.lang.builder.ToStringStyle
import org.codehaus.groovy.util.HashCodeHelper
```

```
class GKontaktanfrage {
```

```
    Long id
    Long version
    String unternehmen
    String name
    String vorname
    String ort
    String postleitzahl
    String produkt
    String anliegen
    List<String> telefonNummern = []
    Map<String, String> emailAdressen = [:]
```

```
    public int hashCode() {
        HashCodeHelper.updateHash(HashCodeHelper.initHash(), id)
    }
```

```
    boolean equals(def obj) {
        if (obj == null) {
            return false
        }
        if (obj == this) {
            return true
        }
        if (obj.getClass() != this.getClass()) {
            return false
        }
        new EqualsBuilder().append(this.id, obj.id).isEquals()
    }
```

```
    String toString() {
        ToStringBuilder.reflectionToString(this, ToStringStyle.SHORT_PREFIX_STYLE)
    }
}
```

POGO

GroovyBean

Das ist Groovy!!!

```
package de.aonnet.groovy.start;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;
import java.util.Maps;
import org.apache.commons.lang.builder.ToStringBuilder;
import org.apache.commons.lang.builder.ToStringStyle;
import org.apache.commons.lang.builder.ToStringStyle;
import org.codehaus.groovy.util.HashCodeHelper;

public class GKontaktanfrage {

    private Long id;
    private Long version;
    private String unternehmen;
    private String name;
    private String vorname;
    private String ort;
    private String postleitzahl;
    private String produkt;
    private String anliegen;
    private List<String> telefonNummern = new ArrayList<String>();
    private Map<String, String> emailAdressen = new HashMap<String, String>();

    public GKontaktanfrage(Long id, Long version, String unternehmen, String name, String vorname, String ort,
        String postleitzahl, String produkt, String anliegen, List<String> telefonNummern,
        Map<String, String> emailAdressen) {
        super();
        this.id = id;
        this.version = version;
        this.unternehmen = unternehmen;
        this.name = name;
        this.vorname = vorname;
        this.ort = ort;
        this.postleitzahl = postleitzahl;
        this.produkt = produkt;
        this.anliegen = anliegen;
        this.telefonNummern = telefonNummern;
        this.emailAdressen = emailAdressen;
    }

    public Long getId() {
        return this.id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public Long getVersion() {
        return this.version;
    }

    public void setVersion(Long version) {
        this.version = version;
    }

    public String getUnternehmen() {
        return this.unternehmen;
    }

    public void setUnternehmen(String unternehmen) {
        this.unternehmen = unternehmen;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getVorname() {
        return this.vorname;
    }

    public void setVorname(String vorname) {
        this.vorname = vorname;
    }

    public String getOrt() {
        return this.ort;
    }

    public void setOrt(String ort) {
        this.ort = ort;
    }

    public String getPostleitzahl() {
        return this.postleitzahl;
    }

    public void setPostleitzahl(String postleitzahl) {
        this.postleitzahl = postleitzahl;
    }

    public String getProdukt() {
        return this.produkt;
    }

    public void setProdukt(String produkt) {
        this.produkt = produkt;
    }

    public String getAnliegen() {
        return this.anliegen;
    }

    public void setAnliegen(String anliegen) {
        this.anliegen = anliegen;
    }

    public List<String> getTelefonNummern() {
        return this.telefonNummern;
    }

    public void setTelefonNummern(List<String> telefonNummern) {
        this.telefonNummern = telefonNummern;
    }

    public Map<String, String> getEmailAdressen() {
        return this.emailAdressen;
    }

    public void setEmailAdressen(Map<String, String> emailAdressen) {
        this.emailAdressen = emailAdressen;
    }

    @Override
    public int hashCode() {
        return HashCodeHelper.updateHash(HashCodeHelper.initHash(), this.id);
    }

    @Override
    public boolean equals(Object obj) {
        if (obj == null) {
            return false;
        }
        if (obj == this) {
            return true;
        }
        if (obj.getClass() != this.getClass()) {
            return false;
        }
        final GKontaktanfrage kontaktanfrage = (GKontaktanfrage) obj;
        return new EqualsBuilder().append(this.id, kontaktanfrage.getId()).isEquals();
    }

    @Override
    public String toString() {
        return ToStringBuilder.reflectionToString(this, ToStringStyle.SHORT_PREFIX_STYLE);
    }
}
```



```
package de.aonnet.groovy.start
```

```
import org.apache.commons.lang.builder.EqualsBuilder
import org.apache.commons.lang.builder.ToStringBuilder
import org.apache.commons.lang.builder.ToStringStyle
import org.codehaus.groovy.util.HashCodeHelper
```

```
class GKontaktanfrage {
```

```
    Long id
    Long version
    String unternehmen
    String name
    String vorname
    String ort
    String postleitzahl
    String produkt
    String anliegen
    List<String> telefonNummern = []
    Map<String, String> emailAdressen = [:]
```

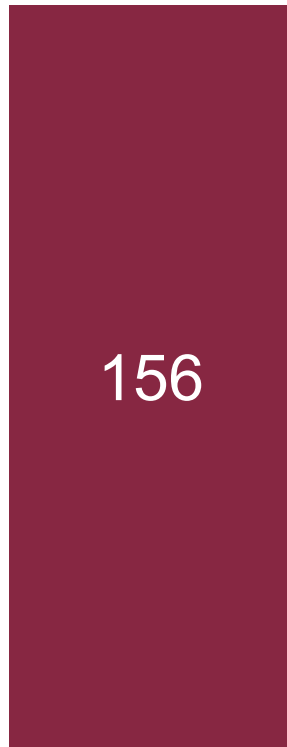
```
    public int hashCode() {
        HashCodeHelper.updateHash(HashCodeHelper.initHash(), id)
    }
```

```
    boolean equals(def obj) {
        if (obj == null) {
            return false
        }
        if (obj == this) {
            return true
        }
        if (obj.getClass() != this.getClass()) {
            return false
        }
        new EqualsBuilder().append(this.id, obj.id).isEquals()
    }
```

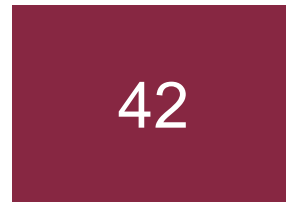
```
    String toString() {
        ToStringBuilder.reflectionToString(this, ToStringStyle.SHORT_PREFIX_STYLE)
    }
}
```


Das ist Groovy!!!

LOC



Java



Groovy



Groovy > 1.8

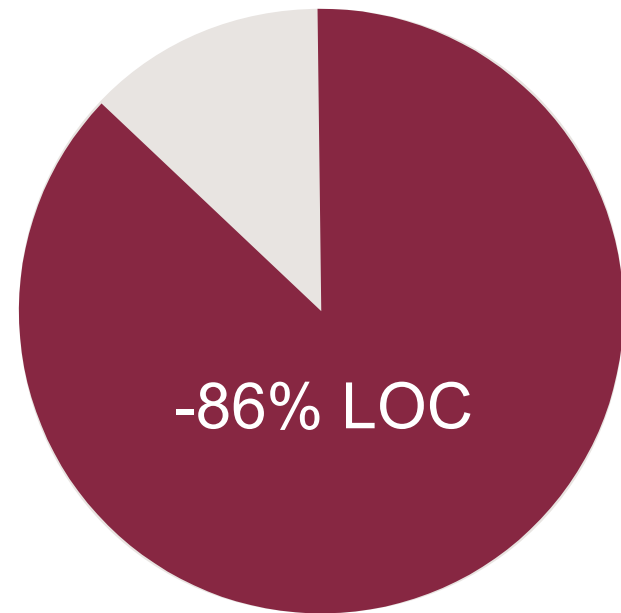
Das ist Groovy ab Version 1.8!!!

```
package de.aonnet.groovy.start

import groovy.transform.EqualsAndHashCode
import groovy.transform.ToString

@ToString
@EqualsAndHashCode(includes='id')
class GKontaktanfrage {

    Long id
    Long version
    String unternehmen
    String name
    String vorname
    String ort
    String postleitzahl
    String produkt
    String anliegen
    List<String> telefonNummern = []
    Map<String, String> emailAdressen = [:]
}
```



Das ist Groovy!!!



Wie löst Groovy noch Probleme?



Wie löst Groovy noch Probleme?

Ausdrucksstarke
Syntax



Wie löst Groovy noch Probleme?

Ausdrucksstarke Syntax

Groovy Markup und Builder
Closures
Operatorüberladung
Native Listen und Maps
Gleitkommaoperationen
Groovy DSL

Wie löst Groovy noch Probleme?



Wie löst Groovy noch Probleme?

Groovy Markup
und
Builder



Ausdrucksstarke Syntax: Groovy Markup und Builder

JSON Objekte in Java

```
JSONObject object = new JSONObject();  
object.put("firstName", "Don");  
object.put("lastName", "Draper");
```

```
JSONArray array = new JSONArray();
```

```
JSONObject kid = new JSONObject();  
kid.put("firstName", "Bobby");  
kid.put("lastName", "Draper");  
array.put(kid);
```

```
kid = new JSONObject();  
kid.put("firstName", "Sally");  
kid.put("lastName", "Draper");  
array.put(kid);  
object.put("kids", array);
```

Ausdrucksstarke Syntax: Groovy Markup und Builder

Groovy: native markup support

```
def root = new JsonBuilder().person {
    firstName 'Don'
    lastName 'Draper'
    kids ([
        [
            firstName: 'Bobby',
            LastName: 'Draper'
        ], [
            firstName: 'Sally',
            LastName: 'Draper'
        ]
    ])
}

// creates a data structure made of
maps (Json object) and lists (Json array)

assert root instanceof Map
```

Ausdrucksstarke Syntax: Groovy Markup und Builder

JSON

Ant tasks

HTML

JMX

W3C DOM

XML

FileTree

Swing

eigene Builder...

Wie löst Groovy noch Probleme?



Wie löst Groovy noch Probleme?

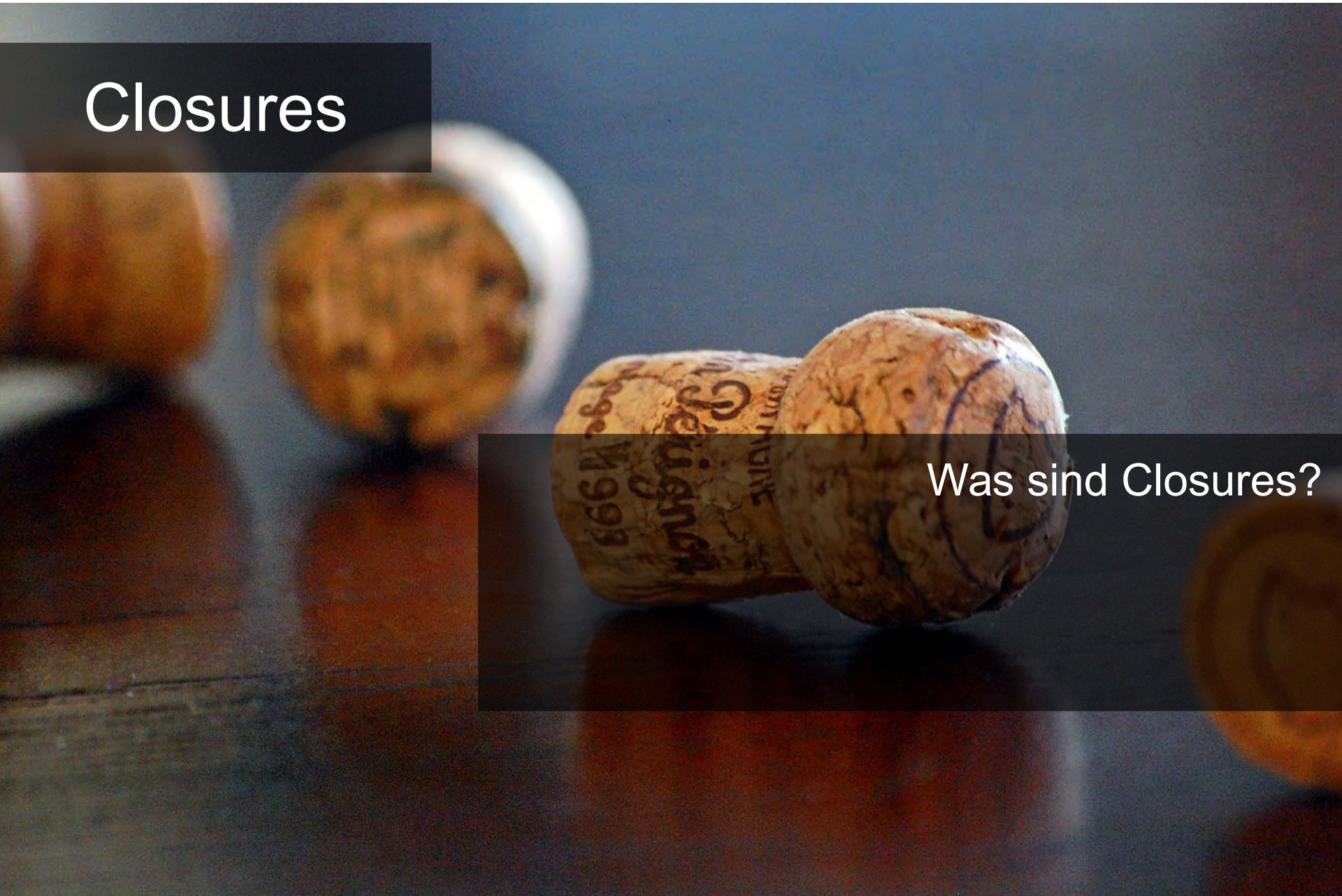
Closures



Wie löst Groovy noch Probleme?

Closures

Was sind Closures?



Wie löst Groovy noch Probleme?

Closures

spezielle Funktionen



Wie löst Groovy noch Probleme?

Closures

spezielle Funktionen

Was bewirken Closures?

Wie löst Groovy noch Probleme?

Closures

spezielle Funktionen

Machen komplexe Programmieraufgaben

elegant und lesbar!

Ausdrucksstarke Syntax: Closures

```
def square = { x ->  
  x * x  
}
```

Ausdrucksstarke Syntax: Closures

```
def square = { x ->  
  x * x  
}
```

Ausdrucksstarke Syntax: Closures

```
def square = { x ->
  x * x
}

assert square(4) == 16
```


Ausdrucksstarke Syntax: Closures

```
def square = { x ->
  x * x
}
```

```
def calculate(def x, def operator) {
  operator(x)
}
```

Ausdrucksstarke Syntax: Closures

```
def square = { x ->
  x * x
}
```

```
def calculate(def x, def operator) {
  operator(x)
}
```

```
assert calculate(4, square) == 16
```

Ausdrucksstarke Syntax: Closures

```
def square = { x ->
  x * x
}
```

```
def calculate(def x, def operator) {
  operator(x)
}
```

```
assert calculate(4, {y -> y ** y}) == 256
```

Wie löst Groovy noch Probleme?



Wie löst Groovy noch Probleme?



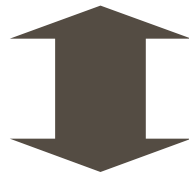
Operatorüberladung

Ausdrucksstarke Syntax: Operatorüberladung

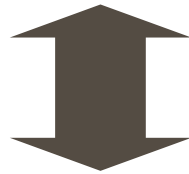
`a == b`

Ausdrucksstarke Syntax: Operatorüberladung

`a.equals(b)`



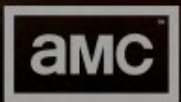
`a == b`



`a.compareTo(b) == 0`



$a == b$



MADMEN
Where the truth lies

Ausdrucksstarke Syntax: Operatorüberladung

`a == b`

`a.equals(b)` oder
`a.compareTo(b) == 0`

`a != b`

`! a.equals(b)`

`a <=> b`

`a.compareTo(b)`

`a > b`

`a.compareTo(b) > 0`

`a >= b`

`a.compareTo(b) >= 0`

`a < b`

`a.compareTo(b) < 0`

`a <= b`

`a.compareTo(b) <= 0`


Ausdrucksstarke Syntax: Operatorüberladung

<code>a + b</code>	<code>a.plus(b)</code>
<code>a - b</code>	<code>a.minus(b)</code>
<code>a * b</code>	<code>a.multiply(b)</code>
<code>a ** b</code>	<code>a.power(b)</code>
<code>a / b</code>	<code>a.div(b)</code>
<code>a % b</code>	<code>a.mod(b)</code>
<code>a b</code>	<code>a.or(b)</code>
<code>a & b</code>	<code>a.and(b)</code>
<code>a ^ b</code>	<code>a.xor(b)</code>
<code>a++</code> or <code>++a</code>	<code>a.next()</code>
<code>a--</code> or <code>--a</code>	<code>a.previous()</code>
<code>a[b]</code>	<code>a.getAt(b)</code>
<code>a[b] = c</code>	<code>a.putAt(b, c)</code>
<code>a << b</code>	<code>a.leftShift(b)</code>
<code>a >> b</code>	<code>a.rightShift(b)</code>
<code>switch(a) { case(b) : }</code>	<code>b.isCase(a)</code>
<code>~a</code>	<code>a.bitwiseNegate()</code>
<code>-a</code>	<code>a.negative()</code>
<code>+a</code>	<code>a.positive()</code>

Ausdrucksstarke Syntax: Operatorüberladung

Download mit Groovy:

```
String url = 'http://upload.wikimedia.org/...se.jpg'  
  
new File('pic.jpg').withOutputStream {  
    it << new BufferedInputStream(url.toURL().openStream())  
}
```

A photograph of three firefighters in full silver protective gear, including helmets and hoods, working at a large fire. They are positioned in front of a massive wall of orange and yellow flames. The firefighters are wearing backpacks with air tanks; one tank has the 'SCOTT' logo. They are holding a yellow fire hose. The ground in the foreground is covered in grey gravel.

```
String url = 'http://upload.wikimedia.org/...se.jpg'  
new File('pic.jpg').withOutputStream {  
    it << new BufferedInputStream(url.toURL().openStream())  
}
```


Wie löst Groovy noch Probleme?



Wie löst Groovy noch Probleme?



Native Listen und Maps

Ausdrucksstarke Syntax: Native Listen

`[1, 2, 3, 4] == (1..4)`

Ausdrucksstarke Syntax: Native Listen

`[1, 2, 3] << 1 == [1, 2, 3, 1]`

`[1, 2, 3] * 2 == [1, 2, 3, 1, 2, 3]`

Ausdrucksstarke Syntax: Native Listen

```
[1, 2, 3, 1].count(1) == 2
```

```
[1, 2, 3, 4].sum() == 10
```

```
[1, 2, 3].reverse() == [3, 2, 1]
```

Ausdrucksstarke Syntax: Native Listen

```
[1, 2, 3].collect { it + 3 } == [4, 5, 6]
```

```
[4, 2, 1, 3].findAll {it % 2 == 0} == [4, 2]
```


Ausdrucksstarke Syntax: Native Maps

```
def m = [a: 'A', b: 'B', c: 'C']
```

```
m.a == 'A'
```

```
m.get('a') == 'A'
```

Ausdrucksstarke Syntax: Native Maps

```
def m = [a: 'A', b: 'B', c: 'C']
```

```
def p = new StringBuffer()
```

```
m.each{ key, value ->
    p << "$key: $value; "
}
```

```
p.toString() == 'a: A; b: B; c: C; '
```

Wie löst Groovy noch Probleme?

LCARS ACCESS

LCARS 40274

02-654598

2385	8578232	9	5789	3882	5893	9865	3489	3465	0846	9798	9629	29
2064	2064962	7	9776	626	1276	7612	126	97	6165	6626	876	74
34	279		89	6589	6547	6587	3465	867	2347	5762	4588	05
4768	8967248	7	9798	8969	476	9847	8476	9749	0982	8969	0247	89
685	3478	8	867	346	34	48	49	8	89	897	38	
757	898990	8	200	285	923	9	387	238	578	875	87	9
484	947589	7	569	68	678	893	56	584	678	476	458	4

9886-234

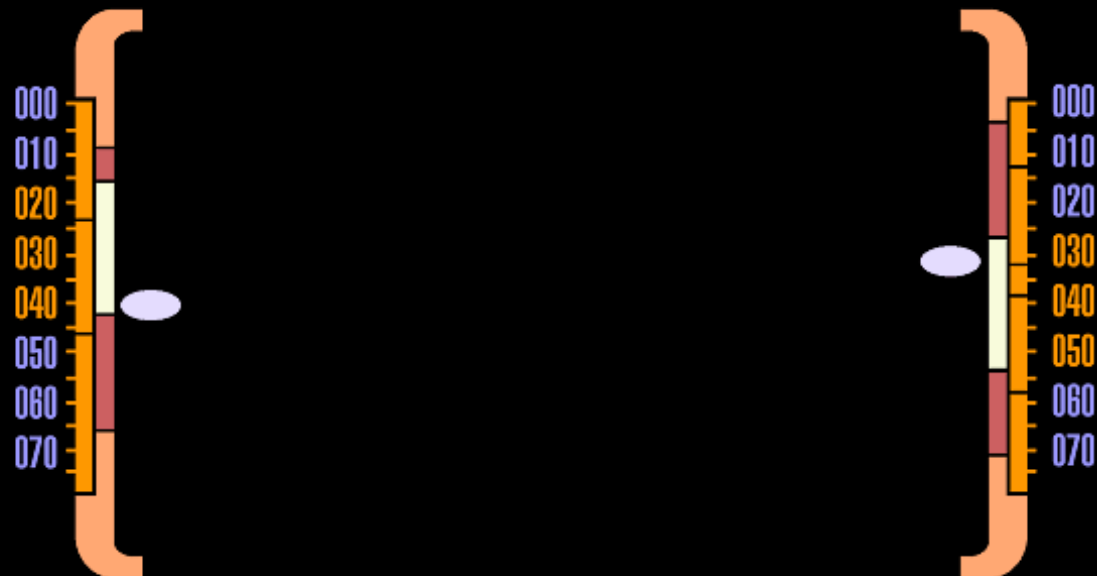
1014-819

03-975683

04-765466

05-224353

06-576565



Wie löst Groovy noch Probleme?

LCARS ACCESS

LCARS 40274

02-654598

2385	8578232	9	5789	3882	5893	9865	3489	3485	0846	9798	9629	29
2064	2064962	7	9776	626	1276	7612	126	97	6165	6626	876	74
34	279		89	6589	6547	6587	3465	867	2347	5762	4588	05
4768	8967248	7	9798	8969	476	9847	8476	9749	0982	8969	0247	89
685	3478	8	867	346	34	48	49	8	89	897	38	
757	898990	8	200	285	923	9	387	238	578	875	87	9
484	947589	7	569	68	678	893	56	584	678	476	458	4

9886-234

1014-819

03-975683

04-765466

05-224353

06-576565

GLEITKOMMAOPERATIONEN

000
010
020
030
040
050
060
070

000
010
020
030
040
050
060
070

Ausdrucksstarke Syntax: Gleitkommaoperationen

1.4 instanceof BigDecimal

1.7 == 1.3 + 0.4

1.700000000000000000002d == 1.3d + 0.4d

Ausdrucksstarke Syntax: Gleitkommaoperationen

```
1/2 == new java.math.BigDecimal('0.5')
```

```
// y = 2 x^3 + 5 x^2 - 3 x + 2
```

```
def x = 5.0
```

```
def y = 2.0*x**3 + 5.0*x**2 - 3.0*x + 2.0
```

```
y == 362.0000
```

Ausdrucksstarke Syntax: Gleitkommaoperationen

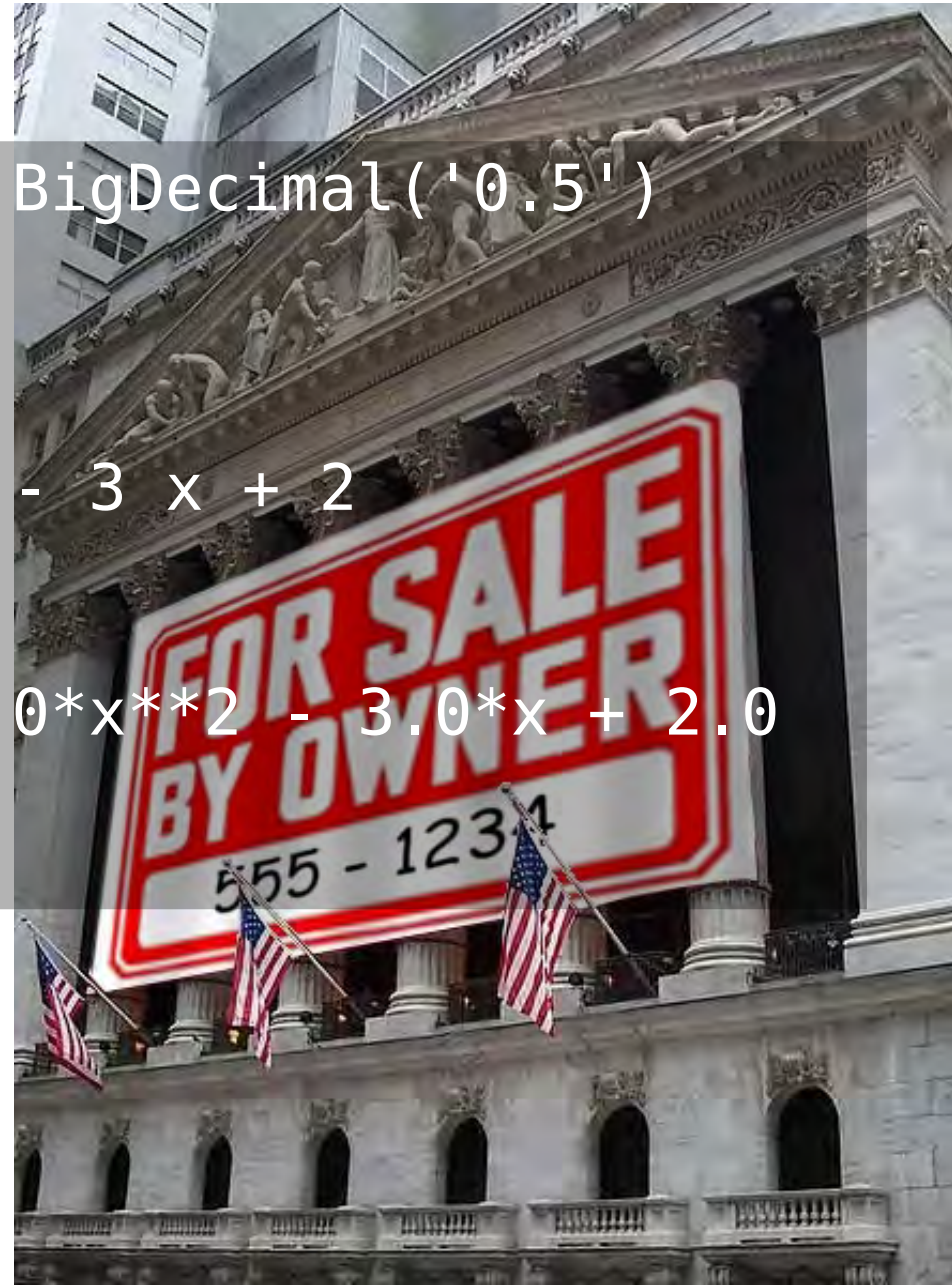
```
1/2 == new java.math.BigDecimal('0.5')
```

```
// y = 2 x^3 + 5 x^2 - 3 x + 2
```

```
def x = 5.0
```

```
def y = 2.0*x**3 + 5.0*x**2 - 3.0*x + 2.0
```

```
y == 362.0000
```



Wie löst Groovy noch Probleme?



Wie löst Groovy noch Probleme?

DSL

(domain-specific language)



Ausdrucksstarke Syntax: DSL

Optionale Klammern und Semikolon

```
move(left);  
move left
```

Ausdrucksstarke Syntax: DSL

Metaprogramming

```
Integer.metaClass.getPills = { ->  
    "$delegate Pills"  
}
```

```
assert 2.pills == '2 Pills'
```


Groovy DSL

take 2.pills of niaxilin after 6.hours

Los gehts!



Live Coding

„The Groovy programming language is sustained & led by SpringSource and the Groovy Community“



„The Groovy programming language
is sustained & led by SpringSource
and the Groovy Community“



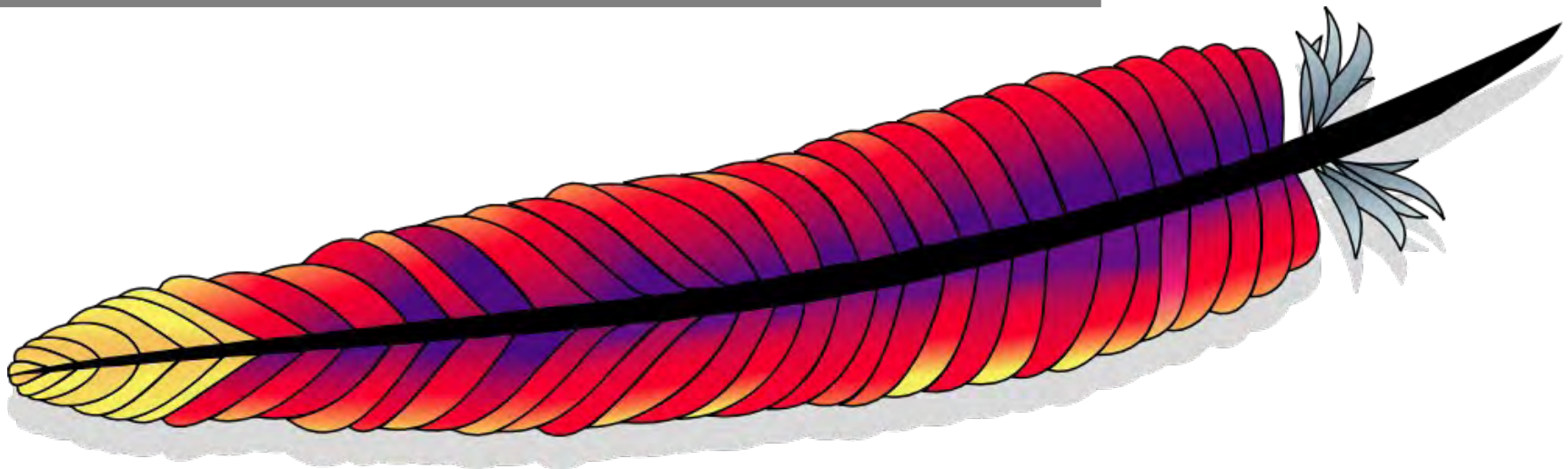
Rosige Aussichten



FIRED!

Rosige Aussichten

APACHE INCUBATOR



<http://incubator.apache.org/projects/groovy.html><http://groovy.codehaus.org>

Rosige Aussichten!



MAD MEN

Sundays 10/9c

amc

Episode II

„The Holy Grail“

Was ist Grails?



Was ist Grails?

A medieval manuscript illustration depicting a scene from the Grail legend. A knight in a white surcoat with a red cross and a black tunic is shown in profile, reaching out towards a glowing golden chalice (the Grail) that is suspended in the air. The chalice is surrounded by a bright, circular glow. Below the knight, two other knights are falling from a cliff. One knight is upside down, and the other is falling towards the bottom. The background is a textured, yellowish-brown surface, possibly representing a cliff face or a parchment page. A dark grey rectangular box is overlaid on the center of the image, containing the text "Die Lösung aller Probleme?".

Die Lösung aller Probleme?

Was ist Grails?



the search is over.



... ist ein Web Application Framework.

- ▶ angelehnt an Ruby On Rails
- ▶ seit 2006
- ▶ aktuelle Version 3.0.2
- ▶ <http://grails.org>

Was will Grails?

Java Web Entwicklung

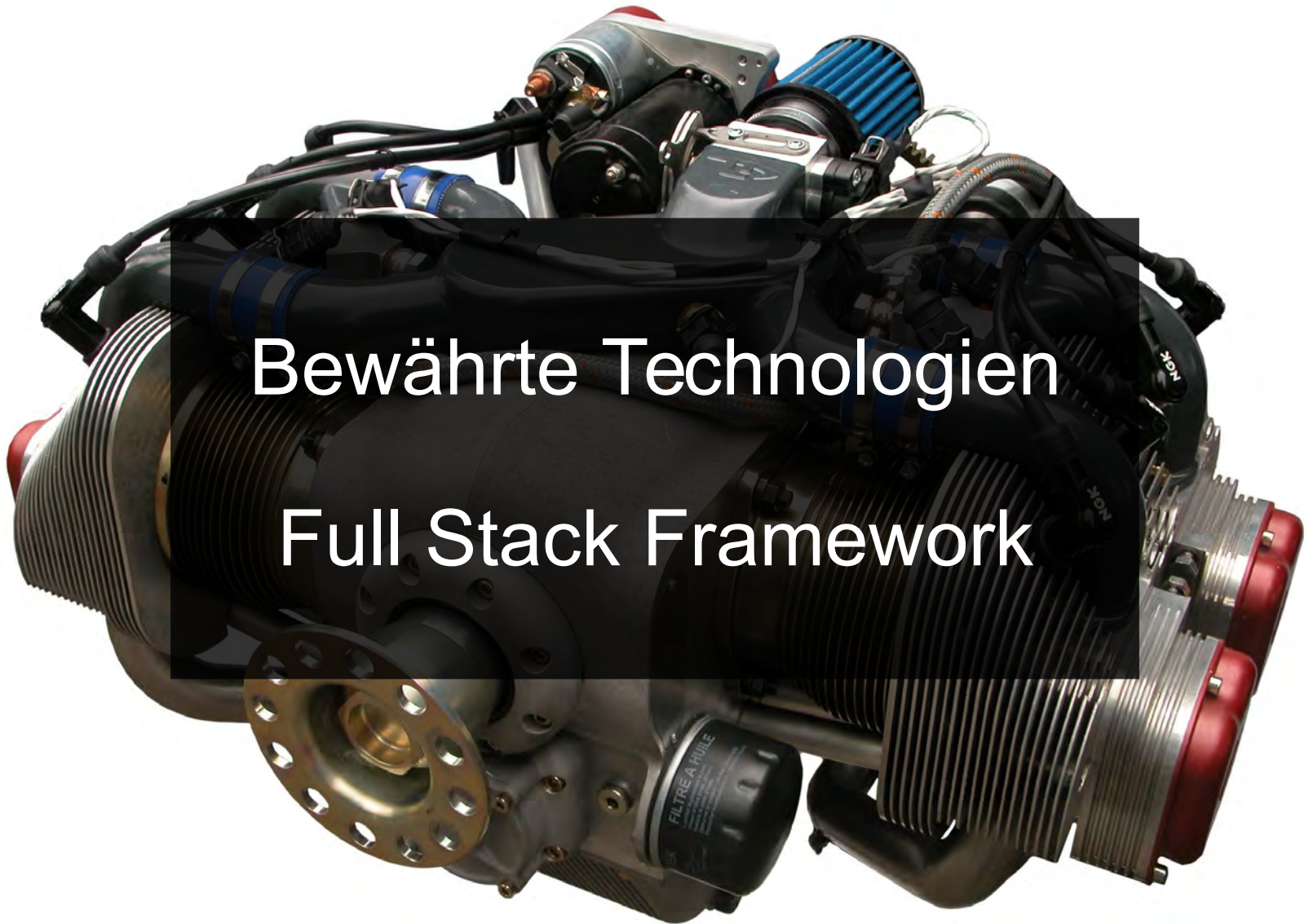
einfach, schnell und robust



Wie löst Grails Probleme?



Wie löst Grails Probleme?



Bewährte Technologien

Full Stack Framework

Bewährte Technologien - Full Stack

- ▶ ORM-Layer GORM (**Hibernate**)
- ▶ transaktionaler Service Layer (**Spring Transactions**)
- ▶ Controller (**Spring MVC**)
- ▶ Groovy Server Pages
GSP
- ▶ Layout mit **Sitemash**
- ▶ DI (**Spring**)



Bewährte Technologien – Grails 3.0

- ▶ Neue Basis: **Spring Boot**
- ▶ Neues Buildsystem: **Gradle**
- ▶ **Application Profiles**
- ▶ **Redesigned API**



Scaffolding



Scaffolding



Home



Motorrad Liste



Motorrad anlegen

Motorrad bearbeiten

Name *

Hersteller *

Herstellungs Datum

*

Gaenge *

Beschreibung *



Aktualisieren

Scaffolding



Home



Motorrad Liste



Motorrad anlegen

Motorrad anzeigen

Motorrad 1 wurde geändert

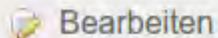
Name Monster 1200

Hersteller Ducati

Herstellungs Datum 23.06.2015 00:00:00 MESZ

Gaenge 7

Beschreibung Die Leistung und das begeisternde Ansprechverhalten des neuen Testastretta 11° DS-Motors, die Vielseitigkeit der drei Riding Modes, der Komfort der mehrfach verstellbaren Sitzhöhe machen die Monster 1200 zu einem faszinierenden Motorrad, das sich nach Ihre persönlichen Ansprüchen richtet. Es gibt Ihnen in jeder Situation die Souveränität, einfach Ihren Instinkten zu folgen und das Fahrvergnügen zu genießen. Das einzigartige Design der Monster hat sie zu einer Ikone gemacht und vereint Form und Funktion.



Bearbeiten



Löschen

Wie löst Grails noch Probleme?



Wie löst Grails noch Probleme?

The image features two large stone statues of King Arthur and Merlin. King Arthur is on the left, wearing a crown and a long, patterned robe, with his right hand raised in a gesture of blessing or command. Merlin is on the right, with a long white beard and a similar robe, holding a staff in his left hand and gesturing with his right. The statues are set against a backdrop of a cloudy sky and a rocky landscape.

Convention
over
Configuration

Wie löst Grails noch Probleme?



Convention
over
Configuration



Don't
Repeat
Yourself

Convention over Configuration



Convention over Configuration

Projektstruktur


A photograph of a line of soldiers in uniform, likely from a historical or military-themed film. The soldiers are wearing blue jackets with red lapels and gold epaulettes. They are also wearing blue hats with red bands and gold braiding. The soldiers are looking forward with serious expressions. The word 'Projektstruktur' is overlaid in white text in the center of the image.

Convention over Configuration



Projektstruktur
Namenskonventionen

Convention over Configuration



Projektstruktur
Namenskonventionen
Unified Configuration

Don't Repeat Yourself



Don't Repeat Yourself

Unified Configuration



Don't Repeat Yourself



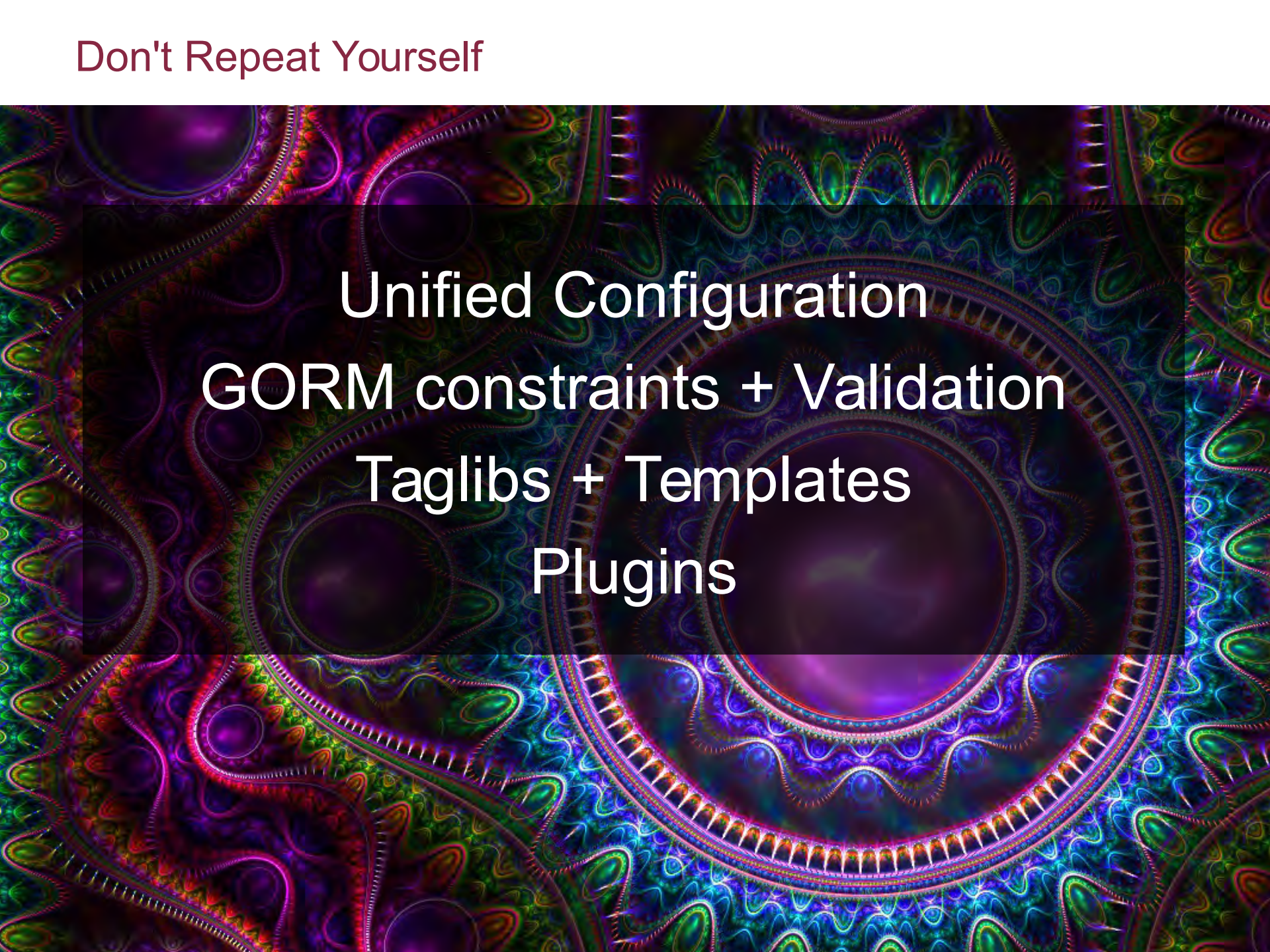
Unified Configuration
GORM constraints + Validation

Don't Repeat Yourself



Unified Configuration
GORM constraints + Validation
Taglibs + Templates

Don't Repeat Yourself



Unified Configuration
GORM constraints + Validation
Taglibs + Templates
Plugins

Grails Plugins



Grails Plugins

A photograph of the International Space Station (ISS) in orbit above Earth. The station's complex structure, including multiple large solar panel arrays and a central module, is clearly visible against the blue and white horizon of the planet. The text is overlaid on a dark, semi-transparent rectangular area at the bottom of the image.

umfangreicher Plugin Mechanismus

Grails Central Plugins Repository

Grails 2: ~~728~~ ~~1073~~ **1209** Plugins

Grails Plugins

A photograph of the International Space Station (ISS) in orbit above the Earth. The station's complex structure, including its large solar panel arrays and central modules, is clearly visible against the blue and white horizon of the planet. The text is overlaid on a dark, semi-transparent rectangular area at the bottom of the image.

umfangreicher Plugin Mechanismus
Grails Bintray Plugins Repository
Grails 3: 35 Plugins

Grails Plugins - Beispiele

Security Plugin

Commentable Plugin

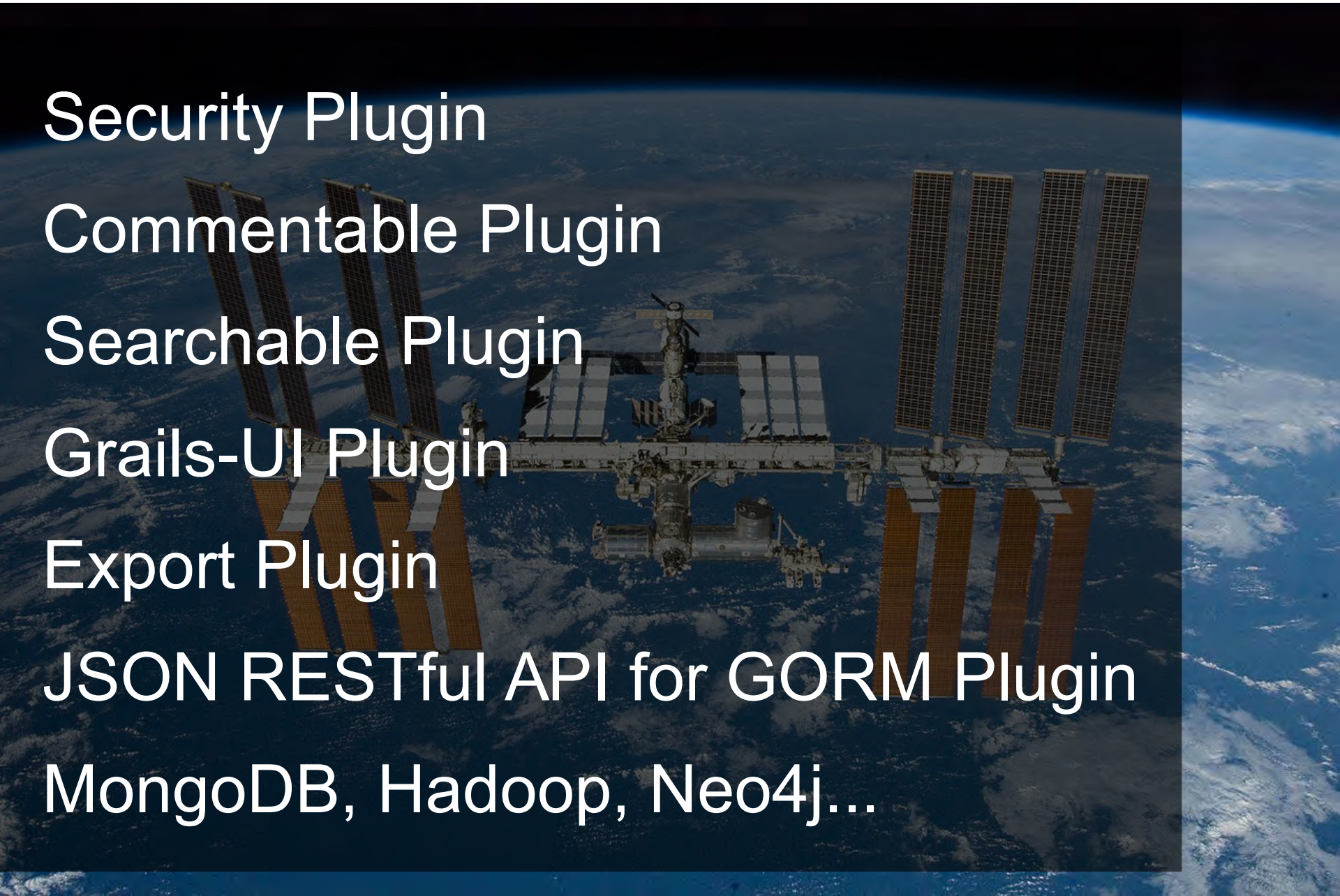
Searchable Plugin

Grails-UI Plugin

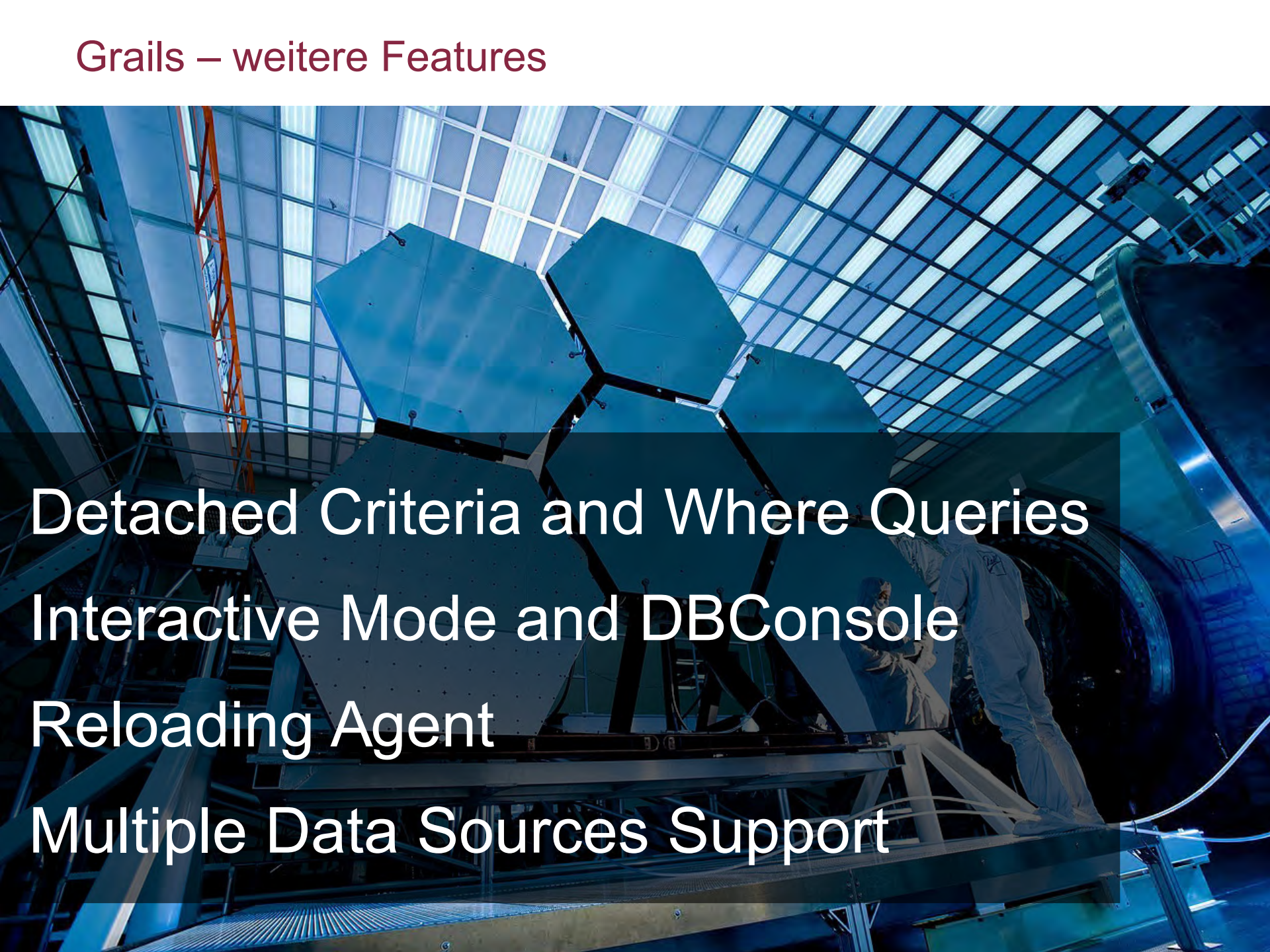
Export Plugin

JSON RESTful API for GORM Plugin

MongoDB, Hadoop, Neo4j...



Grails – weitere Features



Detached Criteria and Where Queries
Interactive Mode and DBConsole
Reloading Agent
Multiple Data Sources Support

Rosige Aussichten



FIRED!

Grails 3

„Grails 3.0 is a complete ground up rewrite of Grails and introduces new concepts and components for many parts of the framework.“

<https://grails.github.io/grails-doc/latest/guide/upgrading.html>

Grails 3

Kompatibel zu Spring Boot

Features von Spring Boot in Grails

Features von Grails in Spring Boot

Epilogue

„Why?“ (ii)

Was sind Groovy und Grails?

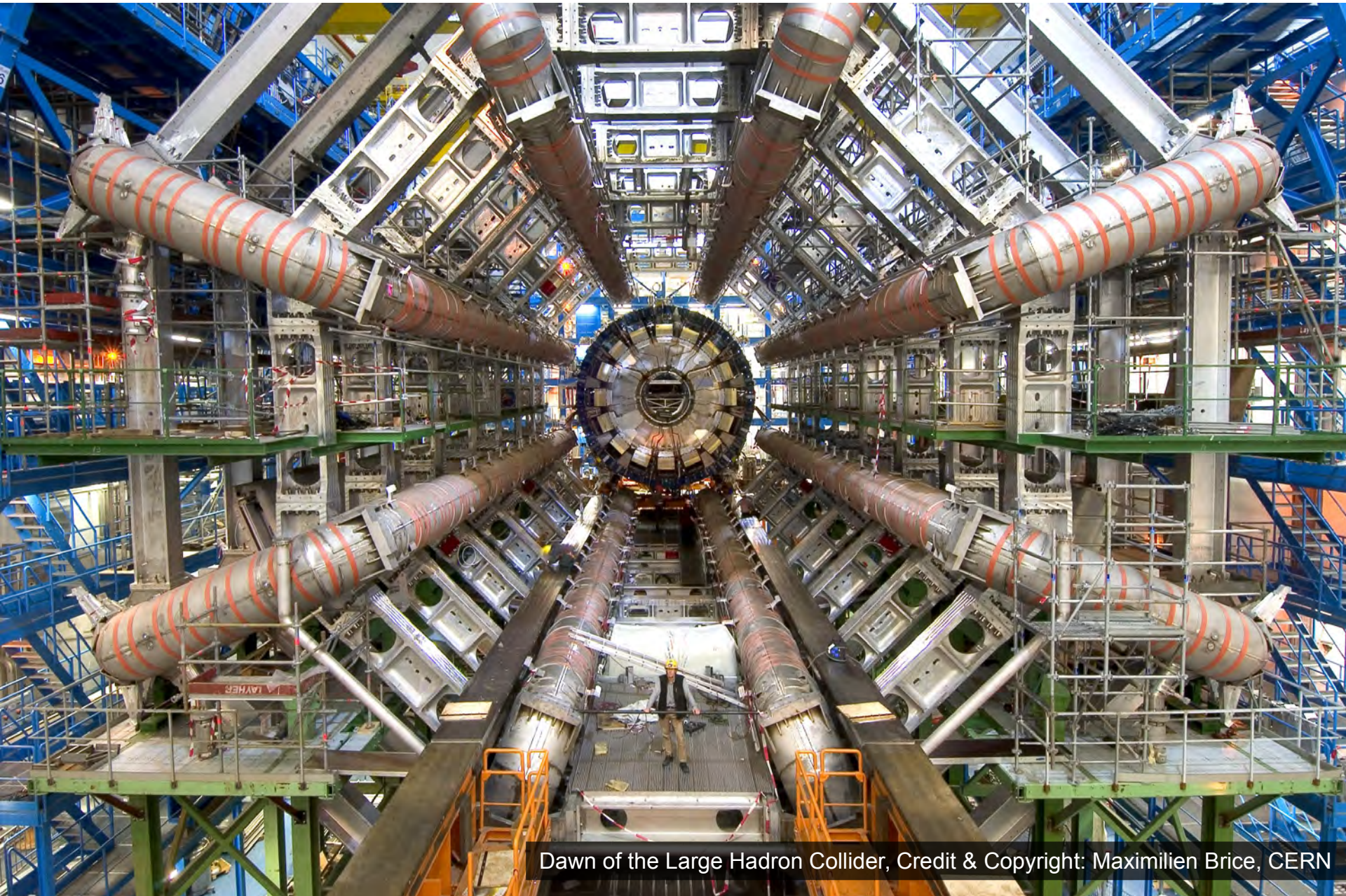


Was sind Groovy und Grails?



RAPID, DYNAMIC, ROBUST

Anwendungsfälle



Dawn of the Large Hadron Collider, Credit & Copyright: Maximilien Brice, CERN

Anwendungsfälle

Groovy

Listen+Maps

POJOs

XML/JSON

Generierung (Templating)

Testing

Filehandling

Gleitkommaoperationen

Scripting

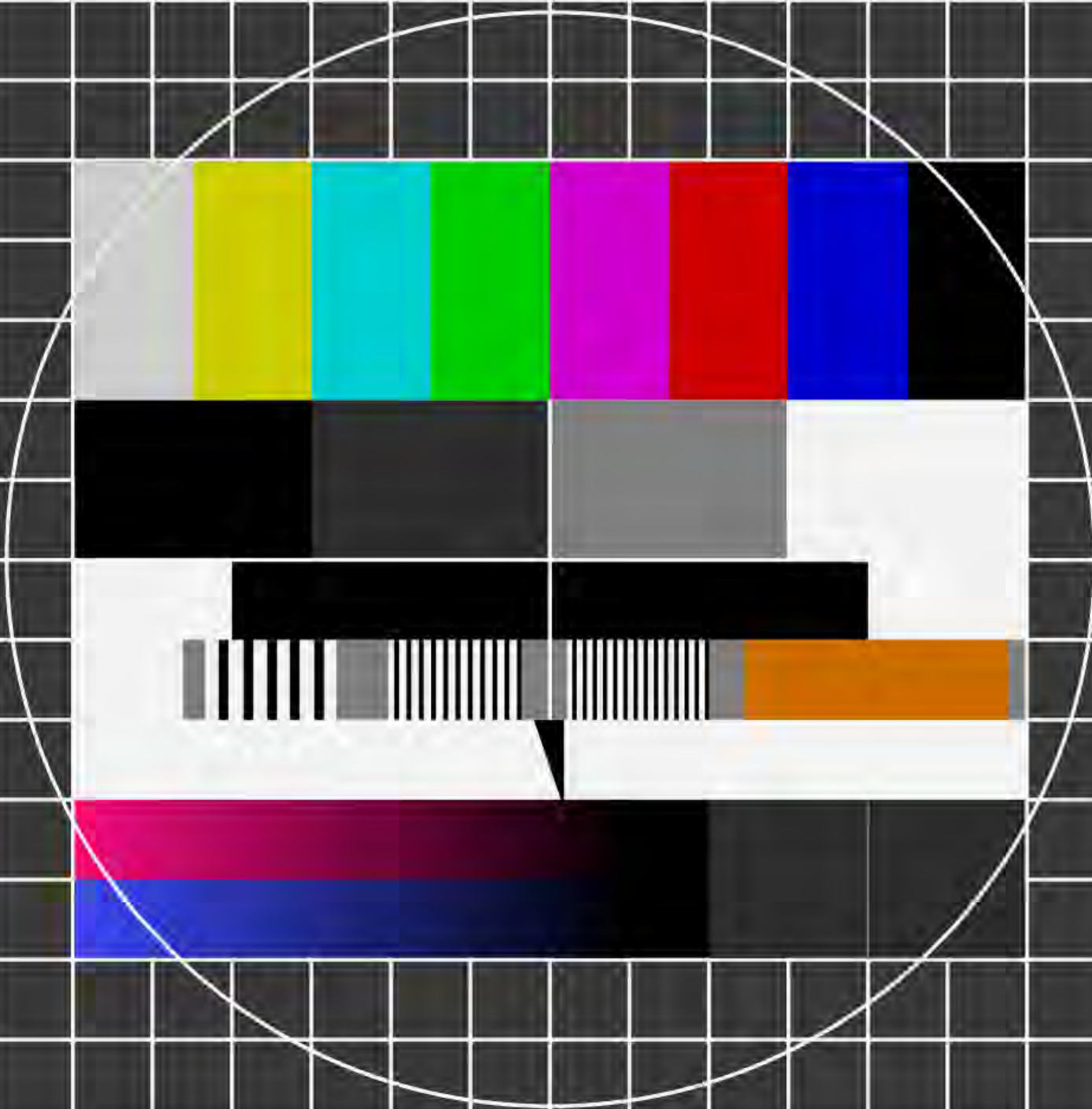
Anwendungsfälle

Groovy

- Listen+Maps
- POJOs
- XML/JSON
- Generierung (Templating)
- Testing
- Filehandling
- Gleitkommaoperationen
- Scripting

Grails

eignet sich grundsätzlich
für jede Art von
Web-Anwendung!



The Groovy Universe



The Groovy Universe



Gradle

gpars

Spock

geb

The Groovy Universe



GVM

Rapack

gealyk

Vert.X

GroovyServ

Griffon

GroovyFX

GContracts

GS

CRaSH

Netflix Asgard

Netflix Zuul

Groovy in ES

Was geben uns Groovy und Grails?



Was geben uns Groovy und Grails?

Clean Code

A large, textured rock formation, possibly a sea stack, is the central focus of the image. It is illuminated from the side, creating a strong contrast between the bright, sunlit side and the dark, shadowed side. The rock has a rough, crystalline texture. The background shows a sunset sky with soft colors and a dark sea with small waves. A dark rectangular overlay is positioned on the right side of the image, containing the text 'Clean Code' in white.

Was geben uns Groovy und Grails?

Clean Code

easy to build



Was geben uns Groovy und Grails?

Clean Code

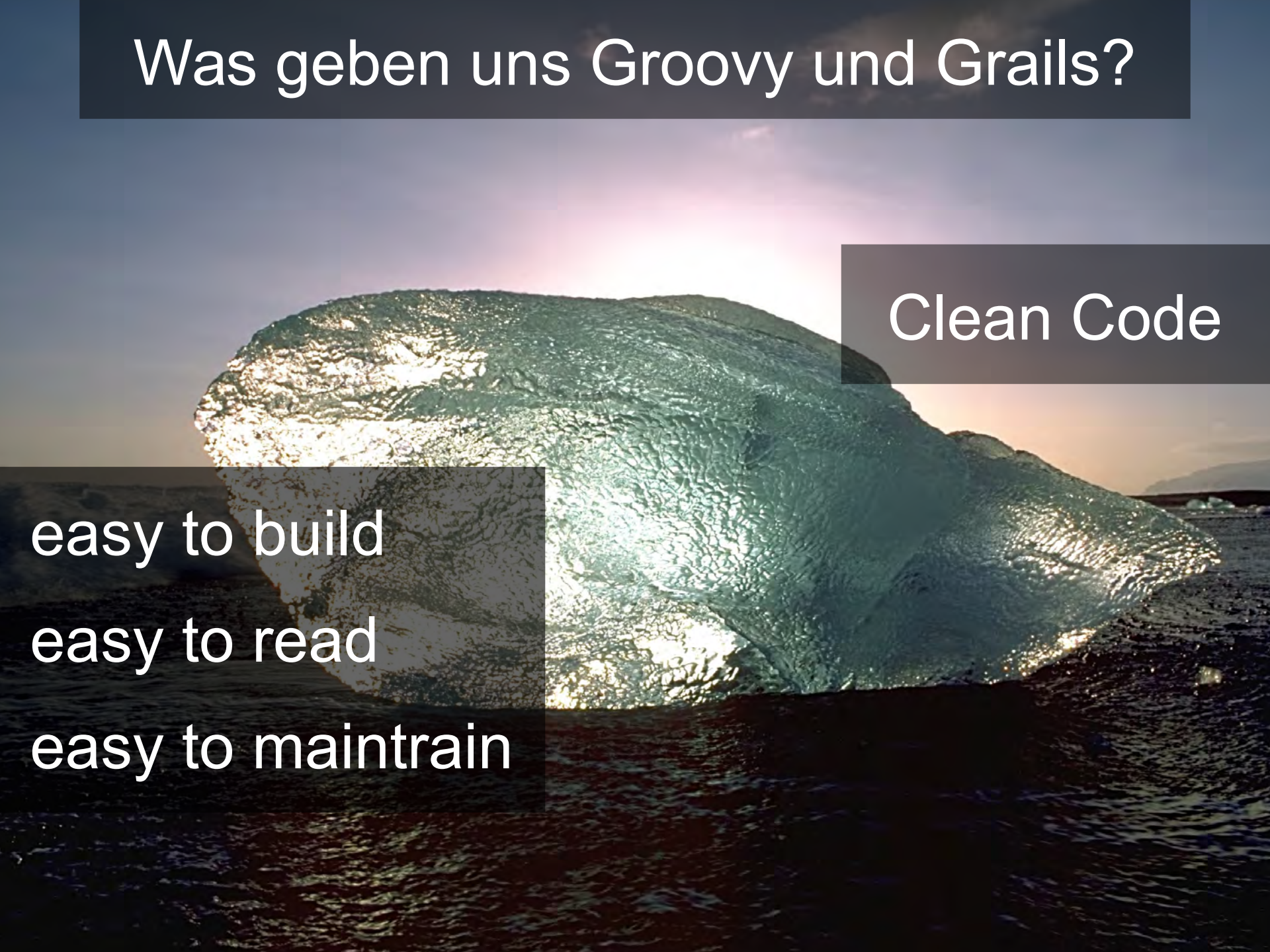
easy to build
easy to read



Was geben uns Groovy und Grails?

Clean Code

easy to build
easy to read
easy to maintain



„Vincent sind wir glücklich?“
„Ja wir sind glücklich!!“

Pulp Fiction



Bildnachweis 1

S 002

<http://photofunia.com>

S 004-005

http://fc07.deviantart.net/fs71/f/2010/339/0/6/four_rooms_movie_poster_by_leknives-d349rpl.jpg

S 007-008

<http://commons.wikimedia.org/wiki/File:Explorer-sinking-2.jpg>

S 011-012

http://upload.wikimedia.org/wikipedia/commons/4/4b/Shock_Wave_Jet_Truck.jpg

S 018-020

<http://www.freemovie-tvwallpaper.com/tv/madmen/images/Mad-Men-05.jpg>

S 021-023

http://commons.wikimedia.org/wiki/File:Elakala_Waterfalls_Swirling_Pool_Mossy_Rocks.jpg

S 032

<https://commons.wikimedia.org/wiki/File:S404-300TDI-erg-chebbi.jpg>

S 033

<http://commons.wikimedia.org/wiki/File:VWRod.jpg>

S 040

http://commons.wikimedia.org/wiki/File:Volkswagen_Hot_Rod.jpg

S 041-043

<http://images.wikia.com/memoryalpha/en/images/4/4c/Khan!!!.jpg>

S 044-045

http://commons.wikimedia.org/wiki/File:Paris_Parade_Navidad_2010-14.JPG

S 052-057

http://commons.wikimedia.org/wiki/File:Champagne_corks.jpg

S 064-065

<http://commons.wikimedia.org/wiki/File:Fire02.jpg>

Bildnachweis 2

S 068

<http://www.vincent-kartheiser.de/Wallpaper/MadMenWallpaper03.jpg>

S 072

http://upload.wikimedia.org/wikipedia/commons/5/5b/Firefighting_exercise.jpg

S 073-074

http://commons.wikimedia.org/wiki/File:Greek_Rich_List_Magazine.jpg

S 081-082

http://commons.wikimedia.org/wiki/File:Lcars_wallpaper.gif

S 085

http://commons.wikimedia.org/wiki/File:NYSE_for_sale.jpg

S 086-087

<http://imaximus.deviantart.com/art/Star-Trek-Klingon-Wallpaper-134194976>

S 090

<http://commons.wikimedia.org/wiki/File:Pills.JPG>

S 091

http://commons.wikimedia.org/wiki/File:Piers_Sellers_spacewalk.jpg

S 092-093

http://commons.wikimedia.org/wiki/File:Leon_hot_air_balloon_festival_2010.jpg

S 094+126

https://commons.wikimedia.org/wiki/File:Uss_iowa_bb-61_pr.jpg

S 096

<http://images.amcnetworks.com/amc.com/wp-content/uploads/2015/04/mm-S7-Wallpaper-1600x1200-E-post.jpg>

S 098-099

<http://www.therpf.com/f9/indiana-jones-grail-knight-painting-3-mega-high-res-pics-104256>

S 101

<http://commons.wikimedia.org/wiki/File:Sailing-yachts.Tuiga.Lulworth.Cambria.Cannes.2006-09-26.jpg>

Bildnachweis 3

S 102-105

http://commons.wikimedia.org/wiki/File:ULPower_UL260i.jpg

S 106-108

http://commons.wikimedia.org/wiki/File:Flickr_-_NewsPhoto!_-_paleis_op_de_Dam_in_de_steigers.jpg

S 109-111

http://www.lotrscenerybuilder.org/images/Argonath_Home_large.jpg

S 112-115

http://commons.wikimedia.org/wiki/File:Russian_honor_guard_at_Tomb_of_the_Unknown_Soldier,_Alexander_Garden_welcomes_Michael_G._Mullen_2009-06-26_2.jpg

S 116-120

http://commons.wikimedia.org/wiki/File:Oracul_1%29fractaldancer%28.png

S 121-125

http://commons.wikimedia.org/wiki/File:STS132_undocking_iss2.jpg

S 127-128

http://commons.wikimedia.org/wiki/File:STS-130_exhaust_cloud_engulfs_Launch_Pad_39A.jpg

S 130-131

http://commons.wikimedia.org/wiki/File:Peugeot_206_WRC.jpg

S 132-134

<http://apod.nasa.gov/apod/ap080225.html>

S135

https://upload.wikimedia.org/wikipedia/commons/thumb/3/32/Telefunken_FuBK_test_pattern.svg/1280px-Telefunken_FuBK_test_pattern.svg.png

S 136-138

https://commons.wikimedia.org/wiki/File:M104_ngc4594_sombrero_galaxy_hi-res.jpg

S 139-144

<http://commons.wikimedia.org/wiki/File:IceBlockNearJoekullsarlton.jpg>

The End