

# Built-Automation

Gradle

Stefan Kuhnert   Thurner Michael

8. Juni 2015



# Was ist Gradle

- Built-Automation Tool:
  - komplexe Projekte
  - built-by-convention
- Weiterentwicklung bestehender Systeme → dazu später mehr

# Was ist Gradle

- Built-Automation Tool:
  - komplexe Projekte
  - built-by-convention
- Weiterentwicklung bestehender Systeme → dazu später mehr

# Was ist Gradle

- Built-Automation Tool:
  - komplexe Projekte
  - built-by-convention
- Weiterentwicklung bestehender Systeme → dazu später mehr

# Was ist Gradle

- Built-Automation Tool:
  - komplexe Projekte
  - built-by-convention
- Weiterentwicklung bestehender Systeme → dazu später mehr

- DSL → auf groovy basierend
  - bessere Lesbarkeit
  - direkt ausführbar
- DAG:
  - Abarbeitungsreihenfolge der Tasks

- DSL → auf groovy basierend
  - bessere Lesbarkeit
  - direkt ausführbar
- DAG:
  - Abarbeitungsreihenfolge der Tasks



- DSL → auf groovy basierend
  - bessere Lesbarkeit
  - direkt ausführbar
- DAG:
  - Abarbeitungsreihenfolge der Tasks

- DSL → auf groovy basierend
  - bessere Lesbarkeit
  - direkt ausführbar
- DAG:
  - Abarbeitungsreihenfolge der Tasks

- DSL → auf groovy basierend
  - bessere Lesbarkeit
  - direkt ausführbar
- DAG:
  - Abarbeitungsreihenfolge der Tasks

- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung

- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung

- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung

- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung

- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung



- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung

- Parallelisierung:
  - Tasks können auf mehreren CPUs/Systemen laufen
- Incremental Build:
  - nur bei Veränderung
- Build-Prozess:
  - Konfiguration
  - Ausführung

# Aspekte anderer Systeme

- **ANT**
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- **ANT**
  - **Flexibilität**
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL



# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

# Aspekte anderer Systeme

- ANT
  - Flexibilität
  - Kontrolle
- Ivy
  - Abhängigkeitsmanagement
- Maven
  - Convention over configuration
  - Multimodulare Projekte
  - erweiterbar durch Plugins
- GANT
  - Groovy-DSL

- Gradle wird von folgenden Frameworks verwendet:
  - Hibernate (relationale DB)
  - Grails (Web App für Groovy)
  - Groovy
  - Spring (Java)
  - Android

- Gradle wird von folgenden Frameworks verwendet:
  - Hibernate (relationale DB)
  - Grails (Web App für Groovy)
  - Groovy
  - Spring (Java)
  - Android

- Gradle wird von folgenden Frameworks verwendet:
  - Hibernate (relationale DB)
  - Grails (Web App für Groovy)
  - Groovy
  - Spring (Java)
  - Android



- Gradle wird von folgenden Frameworks verwendet:
  - Hibernate (relationale DB)
  - Grails (Web App für Groovy)
  - Groovy
  - Spring (Java)
  - Android

- Gradle wird von folgenden Frameworks verwendet:
  - Hibernate (relationale DB)
  - Grails (Web App für Groovy)
  - Groovy
  - Spring (Java)
  - Android

- Gradle wird von folgenden Frameworks verwendet:
  - Hibernate (relationale DB)
  - Grails (Web App für Groovy)
  - Groovy
  - Spring (Java)
  - Android

# Beispiel

## Unser Server-Gradle Script

```
1  apply plugin: 'java'
2  apply plugin: 'application'
3  sourceCompatibility = 1.7
4  version = '1.0'
5  sourceSets.main.java.srcDirs = ['.']
6  mainClassName = 'server.Main'
7
8  sourceSets {
9
10 }
11
12 repositories {
13     mavenCentral()
14 }
15
16 jar {
17     from { configurations.compile.collect { it.isDirectory() ? it : zipTree(it) } }
18     baseName = 'SMRMTMS-SERVER-JAR'
19     manifest {
20         attributes 'Implementation-Title': 'Gradle Quickstart',
21                   'Implementation-Version': version,
22                   'Main-Class': 'server.Main'
23     }
24 }
25
26 uploadArchives {
27     repositories {
28         flatDir {
29             dirs 'repos'
30         }
31     }
32 }
33
34 dependencies {
```