



**diva<sup>e</sup>**

Digital Value Excellence

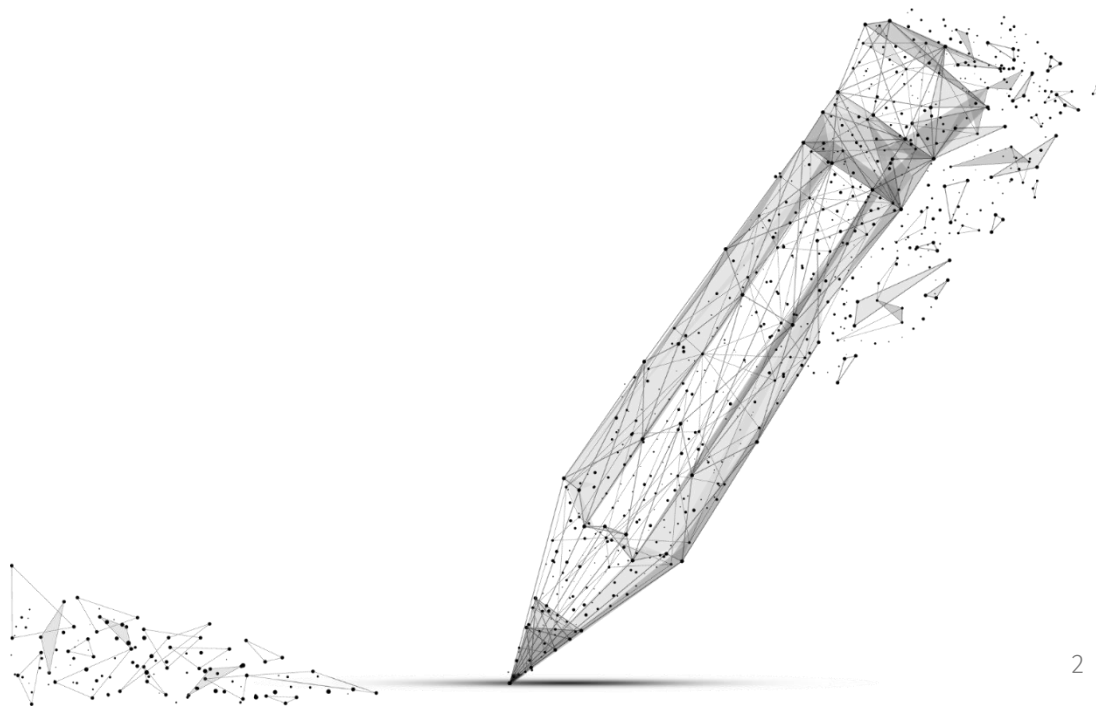
# **Bytecode-Manipulation leicht gemacht**

→ Christian Kumpe, JavaLand 14. März 2018

# Agenda

---

**Referent**  
**Einführung**  
**Motivation**  
**Werkzeuge**  
**Javassist & ClassFileTransformer**  
**Fazit**



# Referent

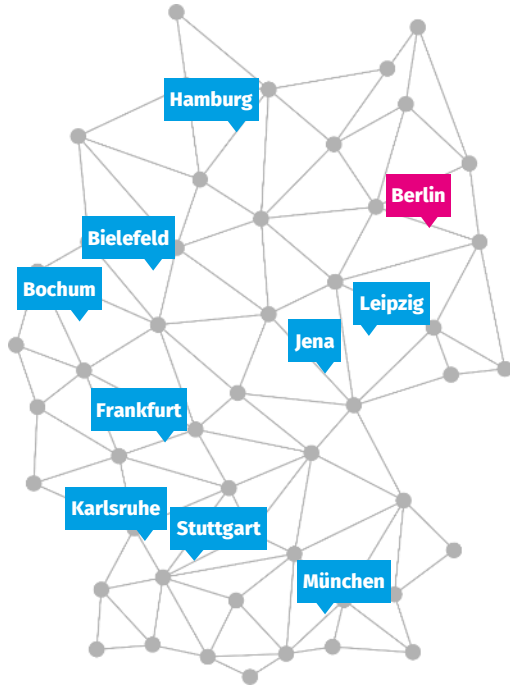
## Christian Kumpe

Expert Developer

- Informatikstudium am KIT (Universität Karlsruhe)
- Freelancer im Bereich Web und Java
- Seit Mai 2011 bei diva-e in Karlsruhe
- Seit 2002 in der Java-Welt unterwegs



# Unternehmensvorstellung



## 10 Standorte

Wir halten eine umfangreiche Vor-Ort-Präsenz bei unseren Kunden als einen wichtigen Erfolgsfaktor. Deshalb sind unsere Standorte über die gesamte Bundesrepublik verteilt.

## 500+ Mitarbeiter

Bei uns arbeiten derzeit über 500 Mitarbeiter.

## 100% Know-How

Von E-Commerce über Content- und Digital-Marketing-Services bis hin zur Retail-Kompetenz: Bei uns erhalten Sie Leistungen aus allen wichtigen E-Business-Disziplinen, vernetzt unter einem Dach.

# Einführung Was ist das?

ca fe ba be 00 00 00 34 00 1a 08 00 11 07 00 13  
07 00 14 07 00 15 07 00 16 09 00 05 00 0a 0a 00  
03 00 0b 0a 00 04 00 09 0c 00 0f 00 0c 0c 00 18  
00 12 0c 00 19 00 0d 01 00 03 28 29 56 01 00 15  
28 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74 72 69  
6e 67 3b 29 56 01 00 16 28 5b 4c 6a 61 76 61 2f  
6c 61 6e 67 2f 53 74 72 69 6e 67 3b 29 56 01 00  
06 3c 69 6e 69 74 3e 01 00 04 43 6f 64 65 01 00  
0c 48 65 6c 6c 6f 20 57 6f 72 6c 64 21 01 00 15  
4c 6a 61 76 61 2f 69 6f 2f 50 72 69 6e 74 53 74  
72 65 61 6d 3b 01 00 27 63 6f 6d 2f 64 69 76 61  
65 2f 74 61 6c 6b 73 2f 6a 61 76 61 73 73 69 73  
74 2f 65 78 61 6d 70 6c 65 73 2f 4d 61 69 6e 01  
00 13 6a 61 76 61 2f 69 6f 2f 50 72 69 6e 74 53  
74 72 65 61 6d 01 00 10 6a 61 76 61 2f 6c 61 6e  
67 2f 4f 62 6a 65 63 74 01 00 10 6a 61 76 61 2f  
6c 61 6e 67 2f 53 79 73 74 65 6d 01 00 04 6d 61

# Einführung Was ist das?

```

ca fe ba be 00 00 00 34 00 1a 08 00 11 07 00 13
07 00 14 07 00 15 07 00 16 09 00 05 00 0a 0a 00
03 00 0b 0a 00 04 00 09 0c 00 0f 00 0c 0c 00 18
00 12 0c 00 19 00 0d 01 00 03 28 29 56 01 00 15
28 4c 6a 61 76 61 2f 6c 61 6e 67 2f 53 74 72 69
6e 67 3b 29 56 01 00 16 28 5b 4c 6a 61 76 61 2f
6c 61 6e 67 2f 53 74 72 69 6e 67 3b 29 56 01 00
06 3c 69 6e 69 74 3e 01 00 04 43 6f 64 65 01 00
0c 48 65 6c 6c 6f 20 57 6f 72 6c 64 21 01 00 15
4c 6a 61 76 61 2f 69 6f 2f 50 72 69 6e 74 53 74
72 65 61 6d 3b 01 00 27 63 6f 6d 2f 64 69 76 61
65 2f 74 61 6c 6b 73 2f 6a 61 76 61 73 73 69 73
74 2f 65 78 61 6d 70 6c 65 73 2f 4d 61 69 6e 01
00 13 6a 61 76 61 2f 69 6f 2f 50 72 69 6e 74 53
74 72 65 61 6d 01 00 10 6a 61 76 61 2f 6c 61 6e
67 2f 4f 62 6a 65 63 74 01 00 10 6a 61 76 61 2f
6c 61 6e 67 2f 53 79 73 74 65 6d 01 00 04 6d 61

```

```

.....4.....
.....
.....
.....()V
(Ljava/lang/Stri
ng;)V..([Ljava/
lang/String;)V..
<init>..Code..
Hello World!..
Ljava/io/PrintSt
Ream;..'com/diva
e/talks/javassis
t/examples/Main
..java/io/PrintS
tream..java/lan
g/Object..java/
lang/System..ma
in..out..print
ln!.....
.....
.....*.....

```

# Motivation

## → Was war unser Problem?

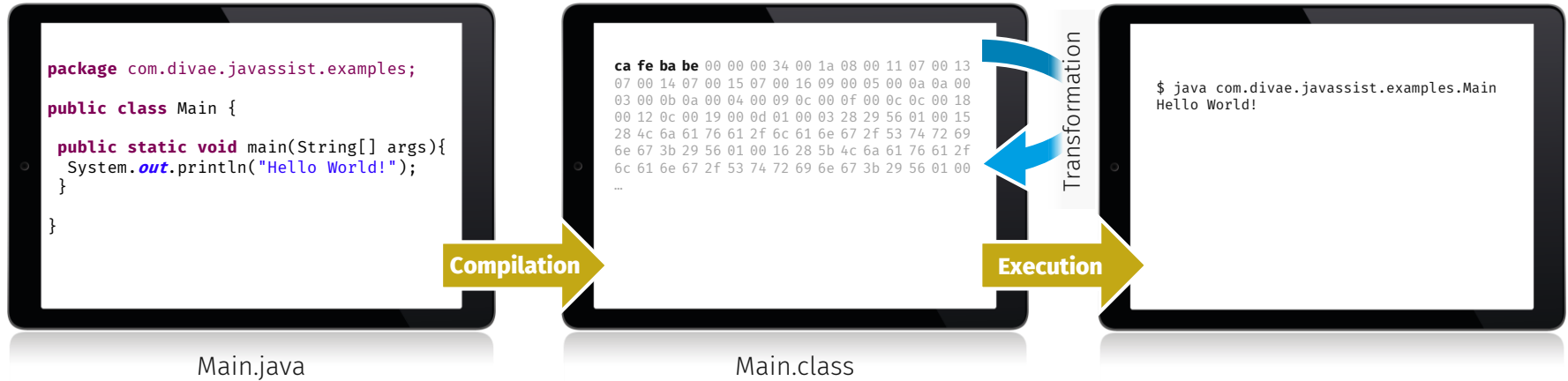
Third-Party Klassen ohne Code sollen serialisierbar gemacht werden.

## → Warum war das ein Problem?

Weil wir die Objekte mit Ehcache zwischenspeichern wollten.



# Vom Code zur Ausführung





# Bytecode

```
$ javap -v Main.class
...
public class com.divae.talks.javassist.examples.Main
...
Constant pool:
  #1 = Methodref      #6.#20      // java/lang/Object."<init>":()V
  #2 = Fieldref       #21.#22     // java/lang/System.out:Ljava/io/PrintStream;
  #3 = String         #23         // Hello World!
...
  #7 = Utf8           <init>
...
{
...
  public static void main(java.lang.String[]);
    descriptor: ([Ljava/lang/String;)V
    flags: (0x0009) ACC_PUBLIC, ACC_STATIC
    Code:
      stack=2, locals=1, args_size=1
         0: getstatic   #2          // Field java/lang/System.out:Ljava/io/PrintStream;
         3: ldc         #3            // String Hello World!
         5: invokevirtual #4          // Method java/io/PrintStream.println:(Ljava/lang/String;)V
         8: return
    LineNumberTable:
      line 5: 0
      line 6: 8
    LocalVariableTable:
      Start Length Slot Name Signature
         0      9      0 args [Ljava/lang/String;
  }
  SourceFile: "Main.java"
```

# Werkzeuge für Bytecode Transformation

## AspectJ

Deklarativ über Pointcuts & Joinpoints.  
Bietet Typsicherheit und  
Inter-Typ-Deklarationen.

<https://www.eclipse.org/aspectj/>

## Bytecode Transformation Tools

Können den Bytecode komplett manipulieren.  
Sind nicht unbedingt typsicher.

Javassist (gleich mehr)

ASM <http://asm.ow2.org>

CGLIB <https://github.com/cglib/cglib>

# Javassist

## → **Highlevel API**

Ist an die Java Reflexion API angelehnt

## → **Bytecode API**

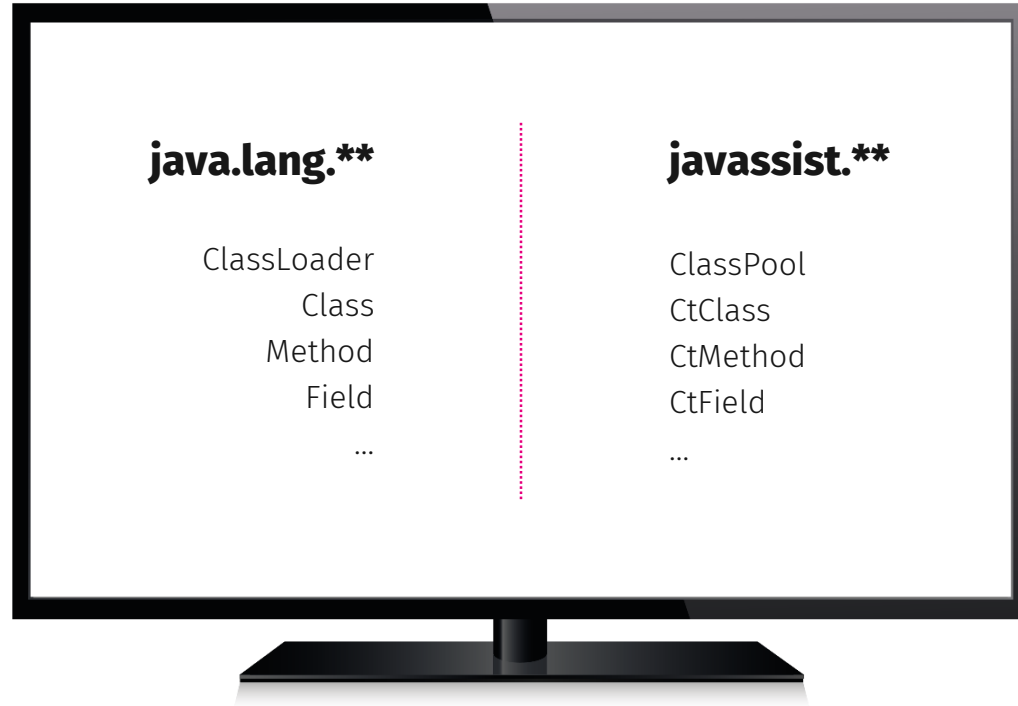
Arbeitet direkt auf dem Bytecode

## → **Homepage**

<http://jboss-javassist.github.io/javassist/>



# Javassist Grundbegriffe



# Hands On

## → High-Level-API

- Klasse öffnen
- Klasse untersuchen
- Klasse erstellen
- Klasse modifizieren

## → Low-Level-API

# Java Agent und ClassFileTransformer

## → Einstiegsklasse erstellen und ins Manifest eintragen

```
public static void premain(String options, Instrumentation instrumentation) {};
```

Registriert den ClassFileTransformer

## → Verpacke den Agent mit allen Abhängigkeiten in eine JAR

JAR kann z.B. mit dem Maven Assembly Plugin erstellt werden

## → Beim Start den Agent mitgeben

```
$ java -javaagent:agent.jar=options
```

# Hands On

- **ClassFileTransformer**
- **Das Serialization-Beispiel aus der Einführung**

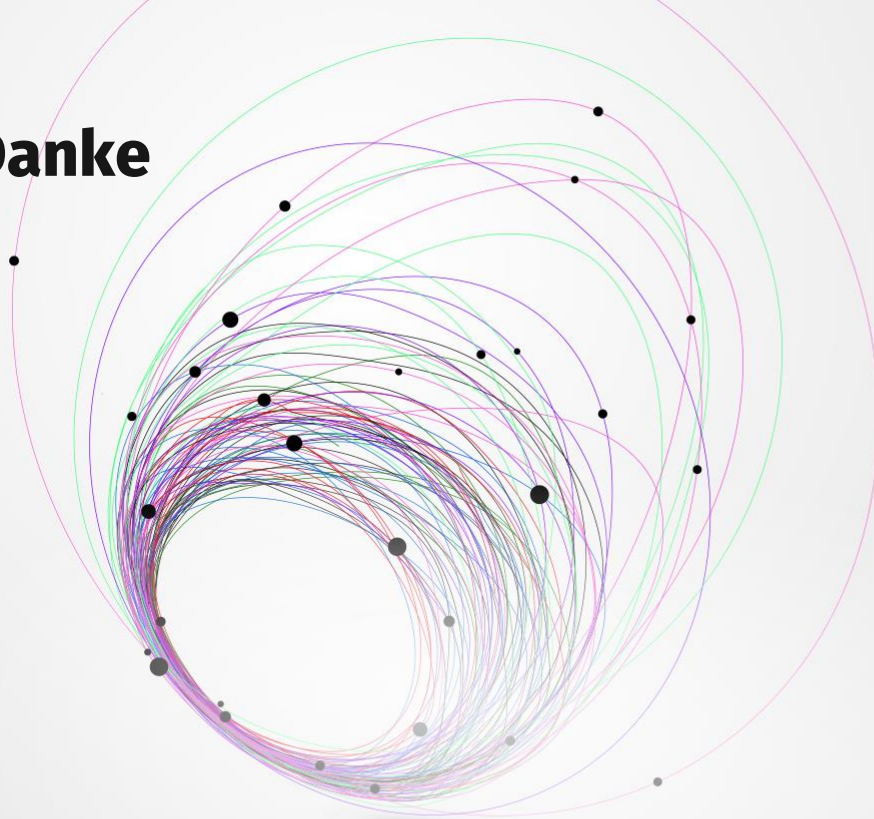
# Fazit

- **Wo AspectJ am Ende ist, beginnt die Bytecode-Manipulation ;-)**
- **Javassist bietet mit seinen beiden APIs viele alle Möglichkeiten**  
Die Highlevel-API senkt die Einstiegshürde deutlich
- **Code Beispiele in GitHub**  
<https://github.com/diva-e/javassist-talk-examples>





# Danke



Digital Value Excellence

## **diva-e Digital Value Excellence GmbH**

Office Karlsruhe

Ihr Kontakt:  
Christian Kumpe  
Expert Developer

Telefon +49 (721) 92060-710  
[christian.kumpe@diva-e.com](mailto:christian.kumpe@diva-e.com)  
[www.diva-e.com](http://www.diva-e.com)