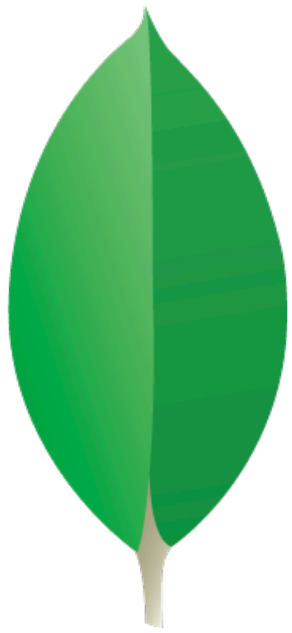
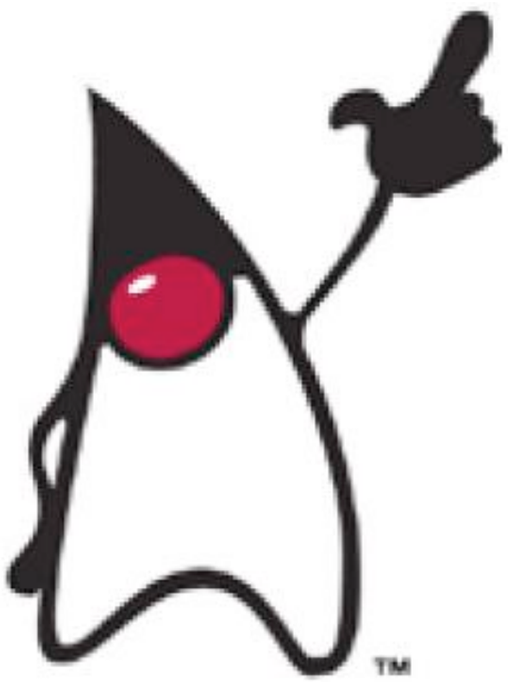


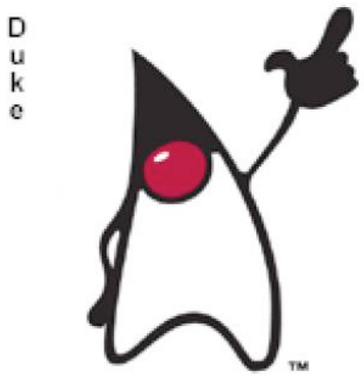
mongoDB

für Java Entwickler und Architekten - Schema Evolution und Maintenance



```
{
    "name" : "Timmo Freudl-Gierke",
    "twitter" :  "@timmo_gierke",
    "blog" : "http://blog-it.hypoport.de/"
}
```

mongoDB


Duke


Straßenbahn 901
Zoo/Uni
AG
Horn-von Hoegen
BEREICH M
MA
MB
MC
MD
ME
MF
MG
MM
Mensa
NETZ
ZBT
LT
LS
SE
Gästehaus
LR
Gerhard-Mercator-Haus
LG
inHaus 1
inHaus 2
Sonderlinie
Bus 926
Uni-Nord
Bus 933
Bus 924, 926
Uni-Nord/Lotharstr.
Lotharstr.
901
Schweizer Str.
SG
Universität
Bus 924, 933
Sonderlinie
Mühlenher Str.
Am Waldcasium
Carl Benz Str.
Sternbusch


mongoDB

humongous

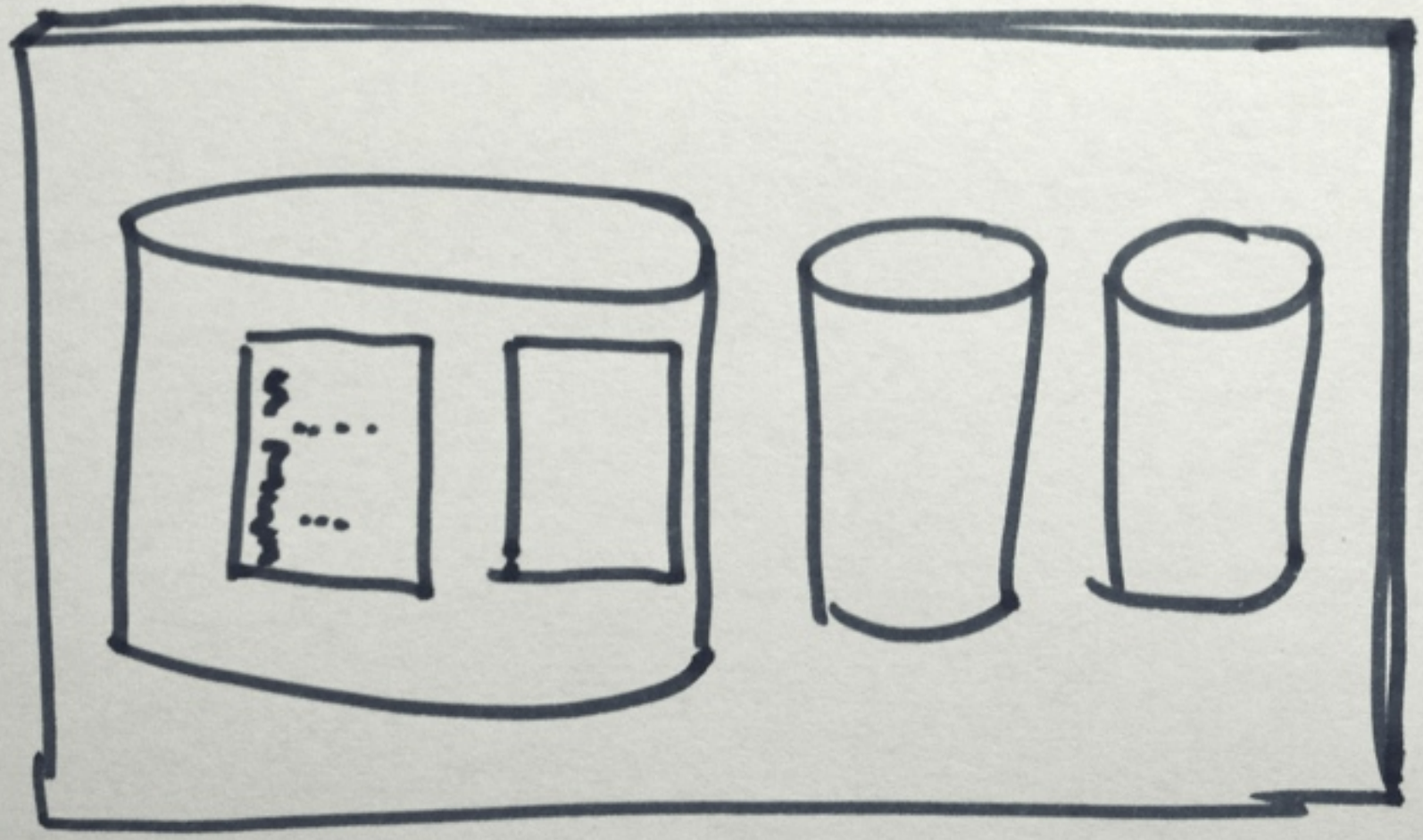
enorm

gigantisch

riesig

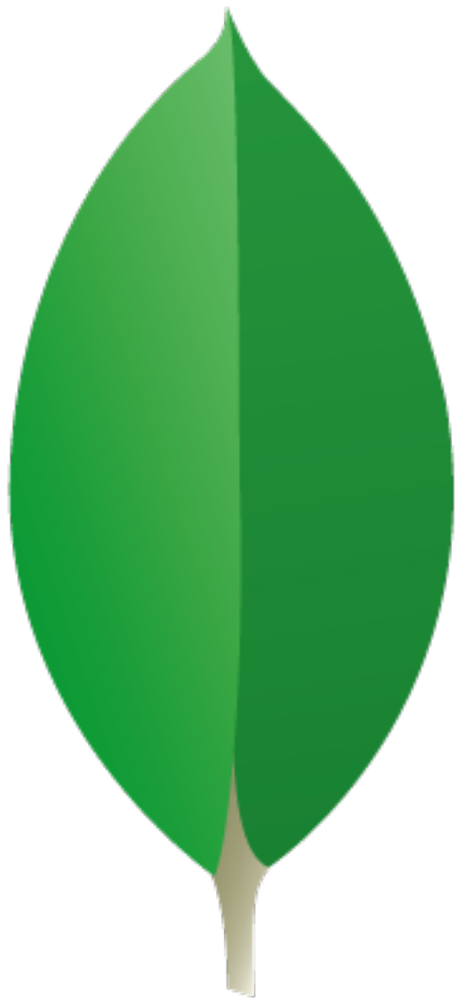
Schema Free

Hierarchie

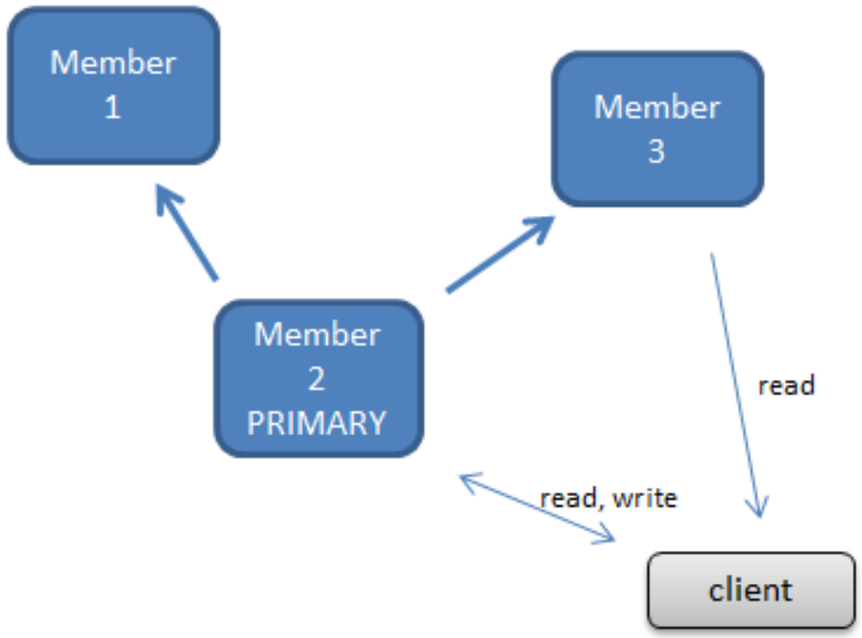


Mongo
DB
Collection
Document
Field : Value

BSON { 01010100 11101011 10101110 01010101 }



MONGO DOCS
MongoDB Documentation »



10gen

Open Source

GNU AGPL v3.0.

Apache License v2.0.

Ziel Platformen

MongoDB Downloads

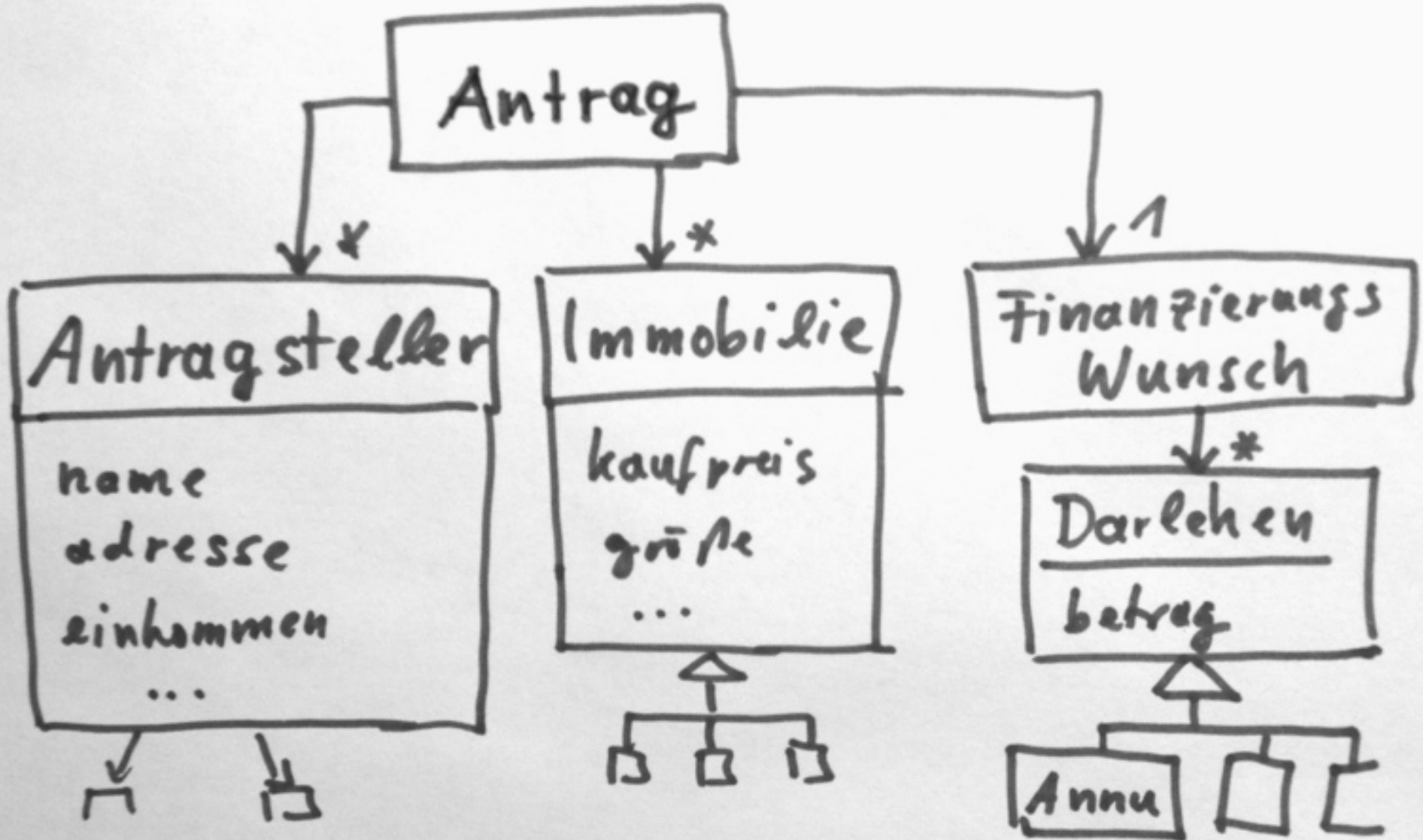
This table lists MongoDB distributions by platform and version. We recommend using these binary distributions, but there are also packages available for various package managers.

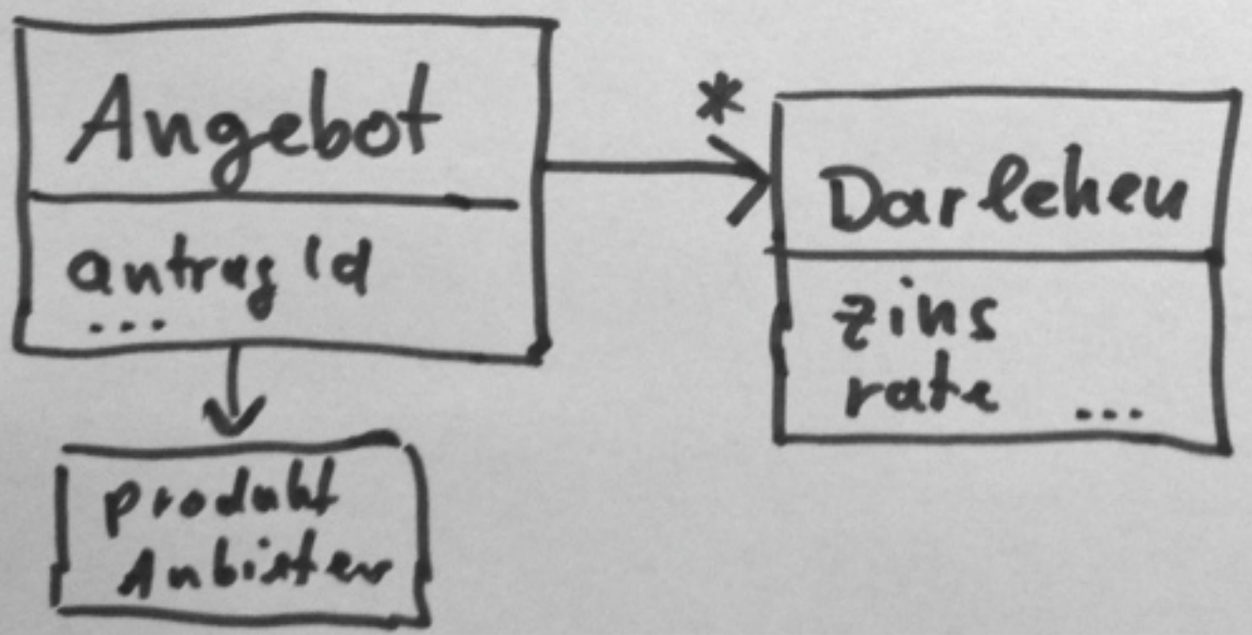
	OS X 32-bit note	OS X 64-bit	Linux 32-bit note	Linux 64-bit	Windows 32-bit note	Windows 64-bit	Solaris i86pc note	Solaris 64	Source
Production Release (Recommended)									
2.0.4 *3/20/2012* Changelog Release Notes	download	download	download *legacy-static	download *legacy-static	download	download *2008+	download	download	tgz zip
Nightly			download	download		download			tgz

Konsole

Mongo

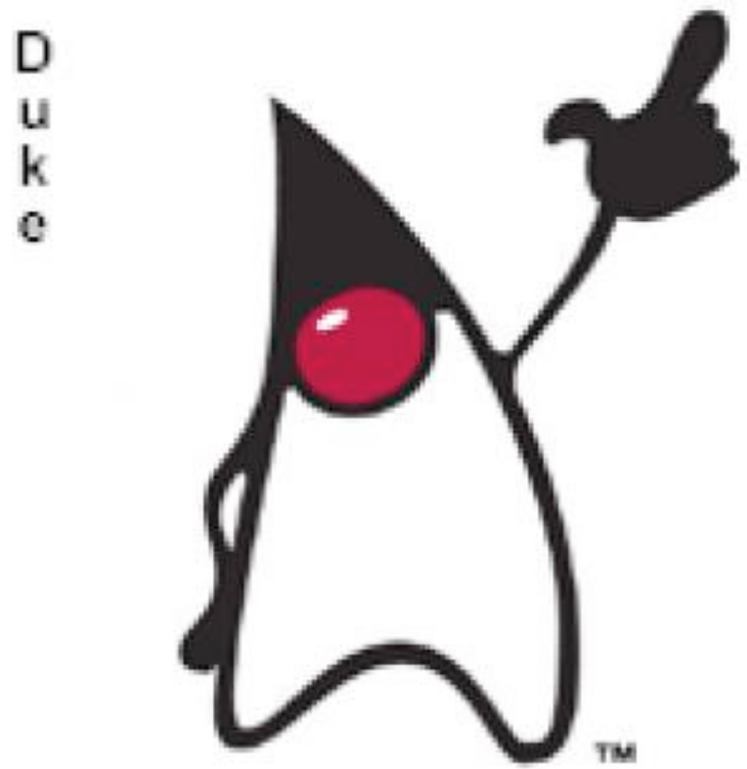
```
timmo@Io:~/Desktop/MongoDB-Java/demo> mongo
MongoDB shell version: 2.0.4
connecting to: test
> use bedcon
switched to db bedcon
> db.personen.save(
... {
...    name : "Schulze",
...    strasse : "Sonntagstr. 20",
...    plz : "10245",
...    ort : "Berlin",
... })
> db.personen.findOne()
{
        "_id" : ObjectId("4f73ff5570583fb228733f27"),
        "name" : "Schulze",
        "strasse" : "Sonntagstr. 20",
        "plz" : "10245",
        "ort" : "Berlin"
}
>
```


Antrag
*
*
1
Antragsteller
name
adresse
einkommen
...
Immobilie
kaufpreis
größe
...
Finanzierungs Wunsch
*
Darlehen
betrag
Annu

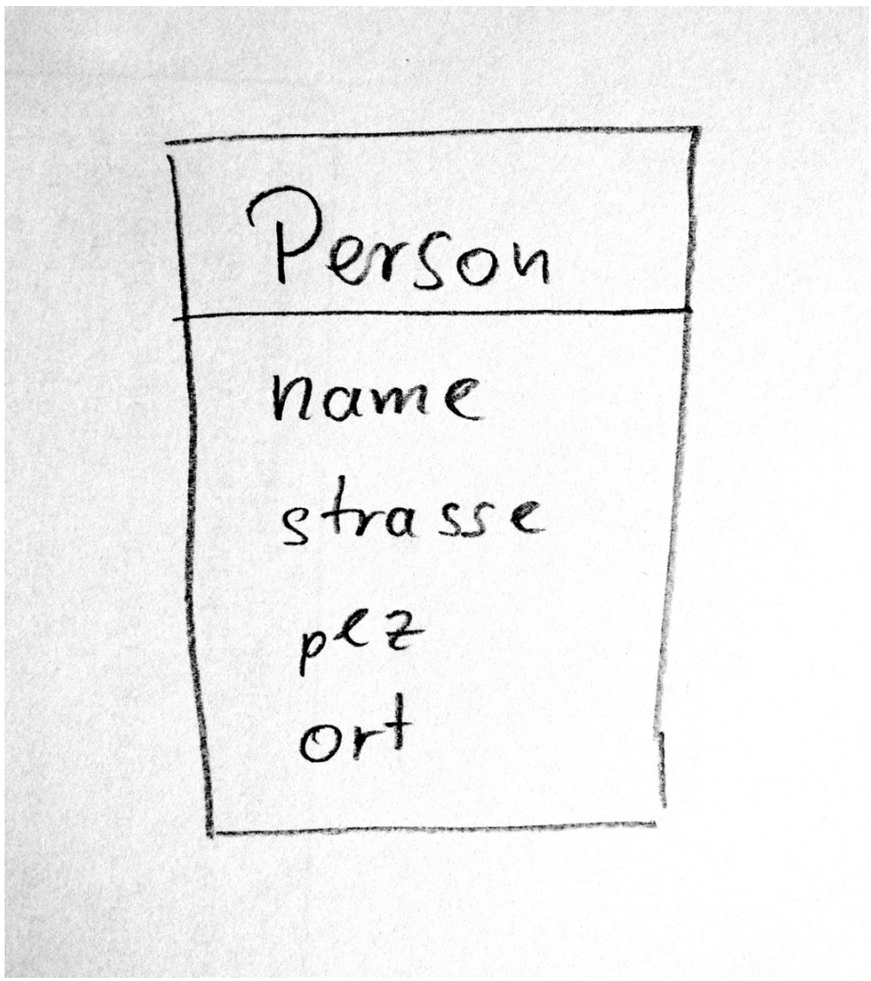

Angebot
antrag Id
...
*
Darlehen
zins rate ...
produkt Anbieter

Continuous Deployment

Später mehr ...


Duke
™

Application Domain Model

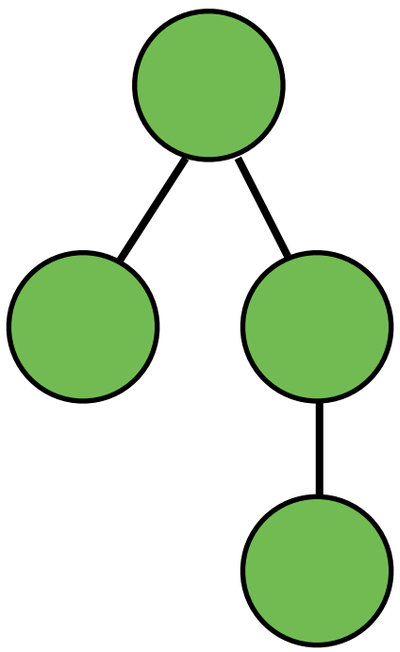


Schema / JSON Structure

```
{
  name : "Schulze",
  strasse : "Sonntagstr. 20",
  plz : "10245",
  ort : "Berlin",
}
```

MongoDB Java Driver

```
DBObject document = new BasicDBObject("name", "Schulze");
document.put("strasse", "Sonntagstr. 20");
...
```



```
{
        "_id" : ObjectId("4f7258da30046355dfe6a1a8"),
        "name" : "Schulze",
        "strasse" : "Sonntagstr. 20"
}
```

Unit Test - Setup

```
17
18  public class MongoDBLearningTest {
19
20      private Mongo mongo;
21      private DB db;
22      private DBCollection personenCollection;
23
24      @BeforeMethod
25      public void setupMongoDB() throws UnknownHostException {
26          mongo = new Mongo();
27          db = mongo.getDB("bedcon");
28          personenCollection = db.getCollection("personen");
29      }
30
31      @AfterMethod(alwaysRun = true)
32      public void tearDownMongo() {
33  //          personenCollection.drop();
34          mongo.close();
35      }
36
```

Unit Test - Mongo Driver

```
36
37      @Test
38      public void mongo_Treiber_API__save_and_find_again() {
39
40        DBObject person = new BasicDBObject("name", "Schulze");
41        person.put("strasse", "Sonntagstr. 20");
42        person.put("plz", "10245");
43        person.put("ort", "Berlin");
44
45        personenCollection.save(person);
46
47        BasicDBObject query = new BasicDBObject("name", "Schulze
48        DBObject found = personenCollection.findOne(query);
49
50        assertThat(found).isNotNull();
51        assertThat(found.get("name")).isEqualTo("Schulze");
52      }
53
```

Object - Document Mapper

Spring Data

- MongoTemplate
- Exception Translation
- Rich Object Mapping
- Lifecycle Events
- Repository Support

MongoDB Jackson Mapper

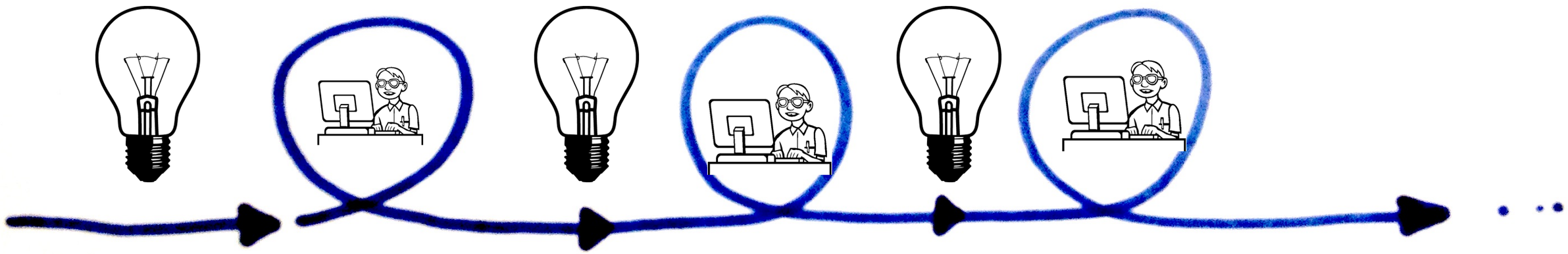
- Wraps Mongo DB + Collection API
- Full Jackson Extensibility
- Fast

- javax.persistence style
- Annotation Based Mapping
- DAO Support

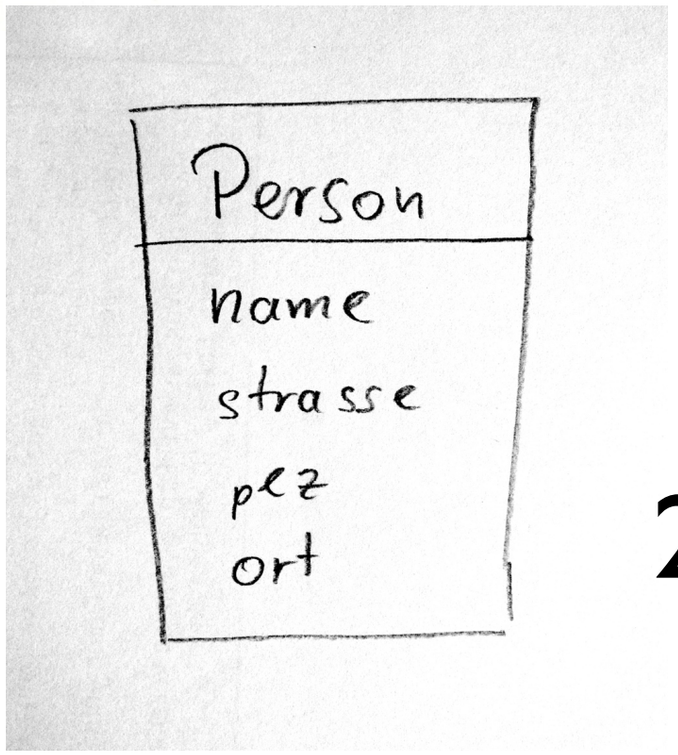
FLEXJSON

Unit Test

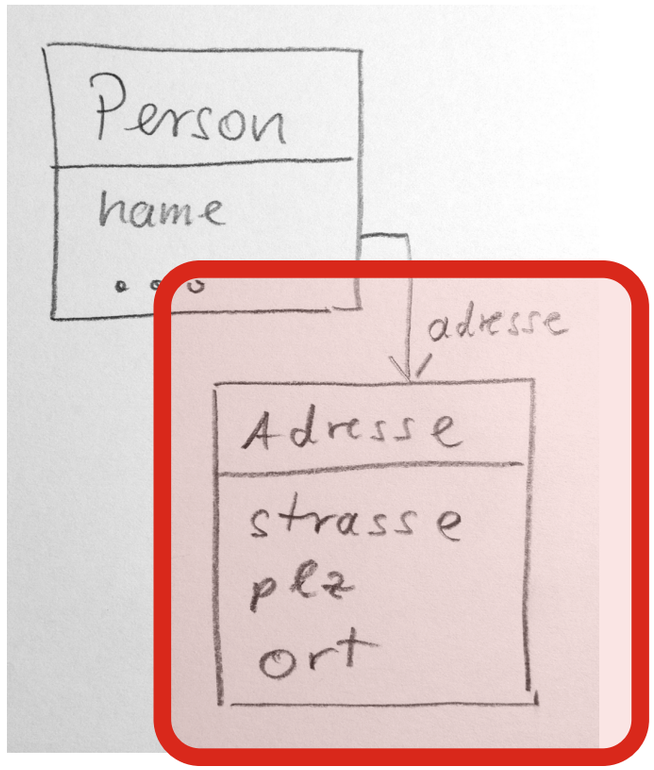
```
54    @Test
55    public void flexJson_Schritte() {
56       Person person = new Person();
57       person.setName("Schulze");
58       person.setStrasse("Sonntagstr. 20");
59       person.setPlz("10245");
60       person.setOrt("Berlin");
61
62       String json = new JSONSerializer().deepSerialize(person);
63
64       DBObject dbObject = (DBObject) JSON.parse(json);
65
66       personenCollection.save(dbObject);
67    }
```



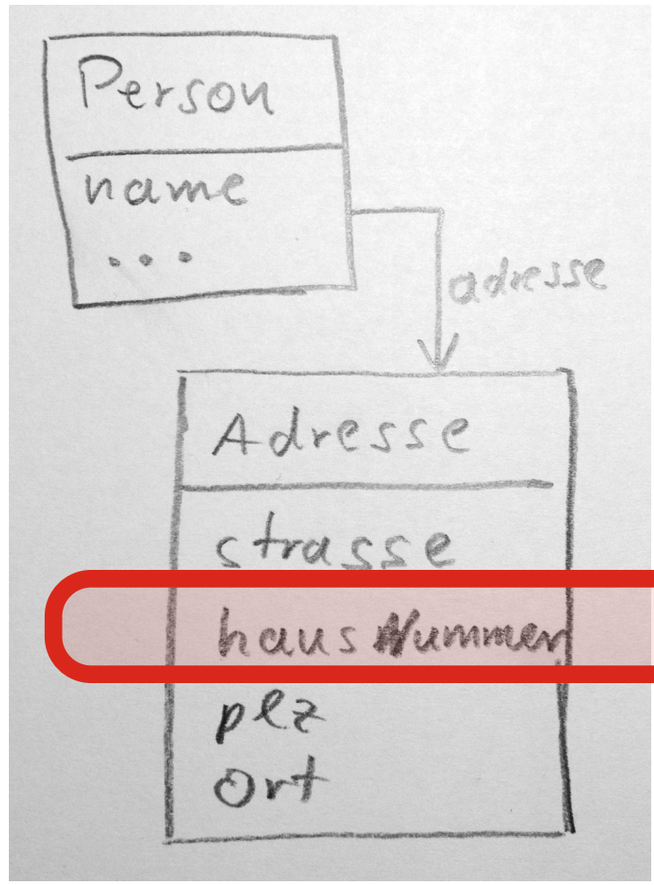
1)

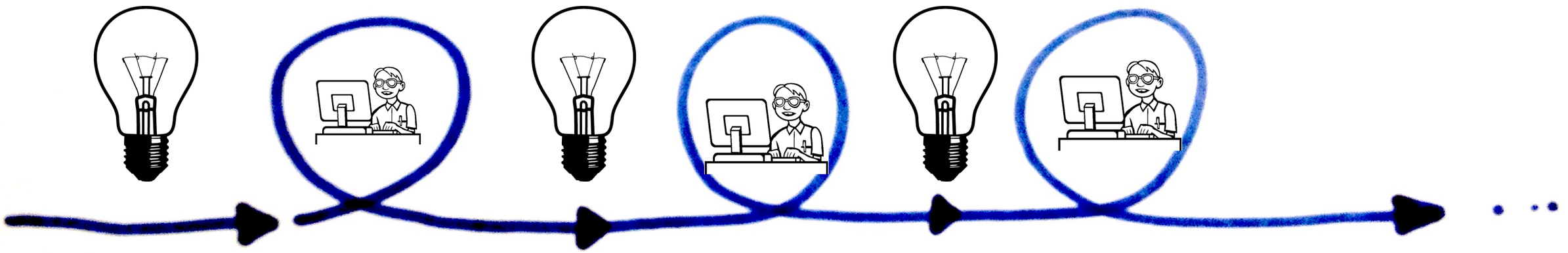

Person
name
strasse
plz
ort

2)


Person
name
adresse
Adresse
strasse
plz
ort

3)


Person
name
adresse
Adresse
strasse
haus Nummer
plz
Ort



1)

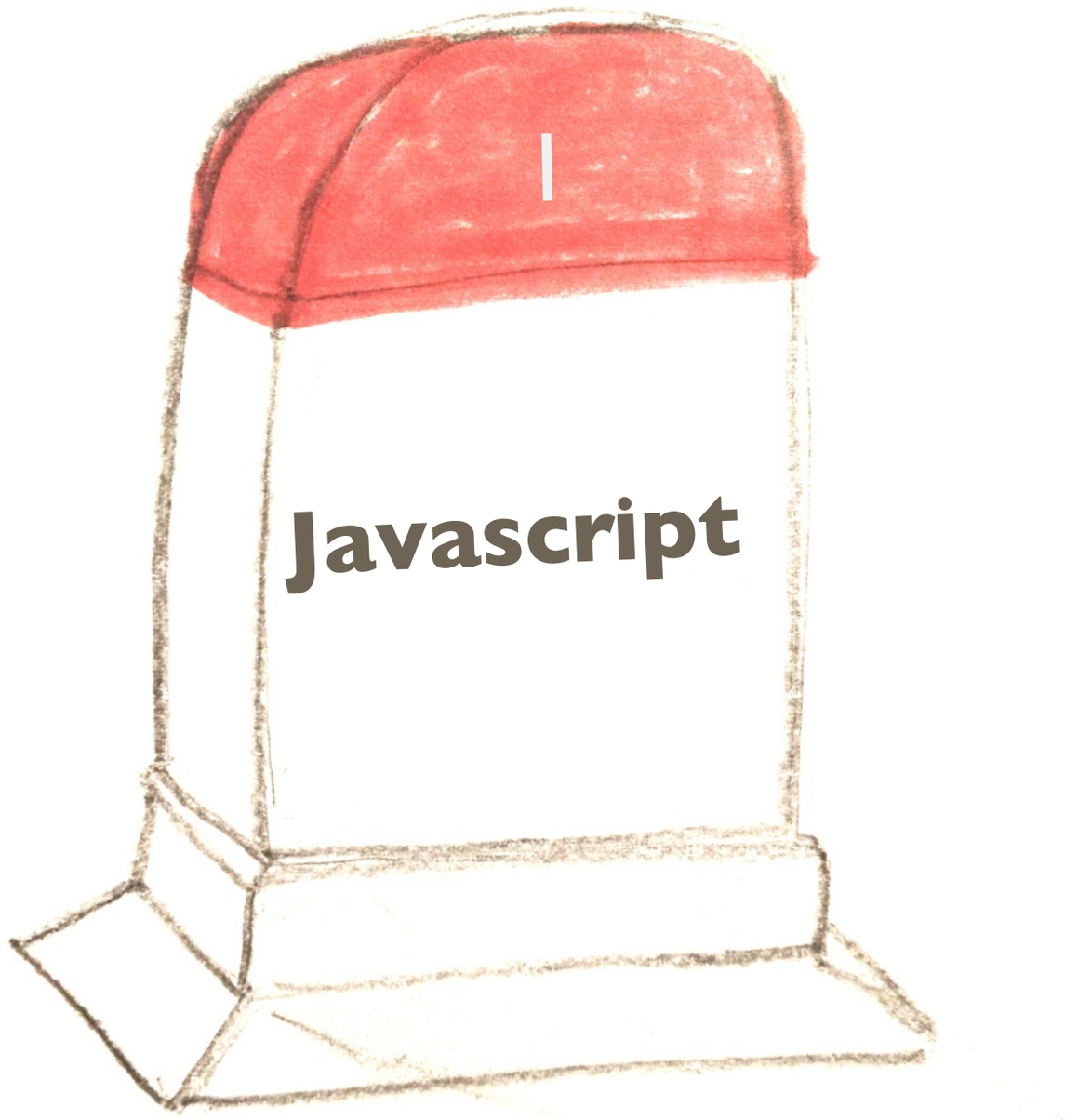
```
{
  name : "Schulze",
  strasse : "Sonntagstr. 20",
  plz : "10245",
  ort : "Berlin",
}
```

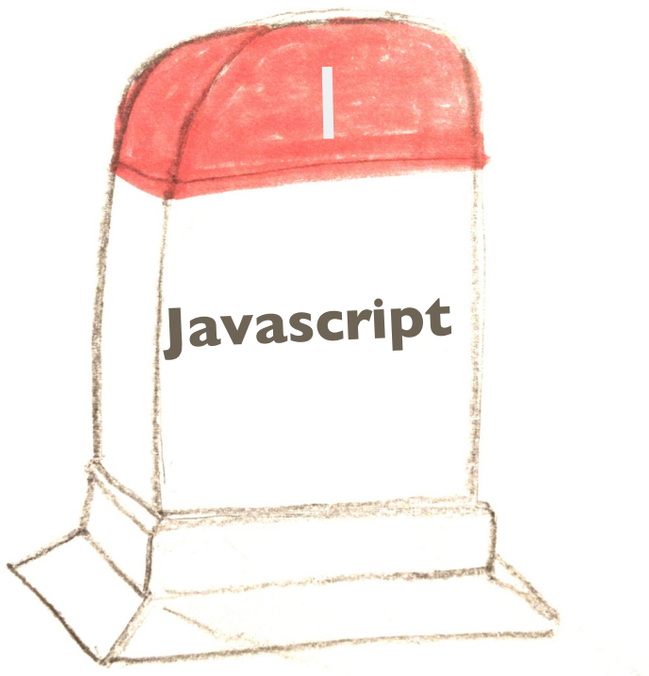
2)

```
{
  name : "Schulze",
  adresse : {
     strasse : "Sonntagstr. 20",
     plz : "10245",
     ort : "Berlin",
  }
}
```

3)

```
{
  name : "Schulze",
  adresse : {
     strasse : "Sonntagstr.",
     hausnummer : "20",
     plz : "10245",
     ort : "Berlin",
  }
}
```

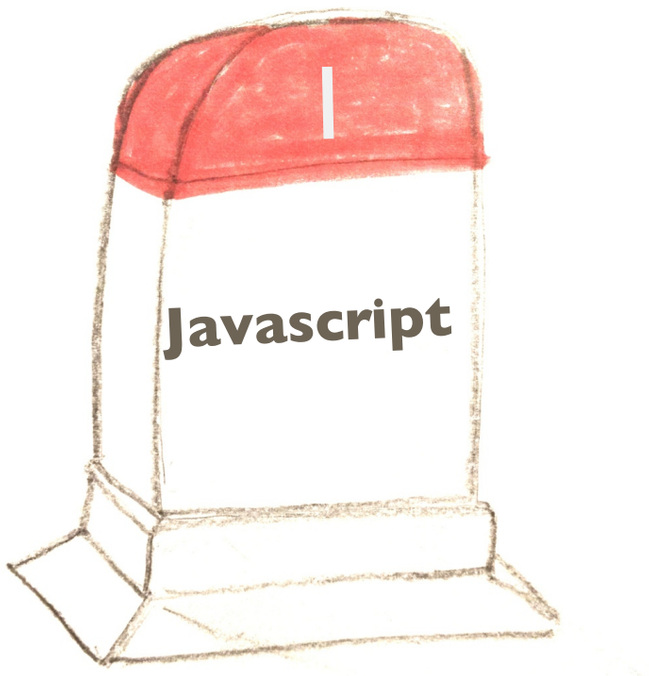

I
Javascript



Field Renaming

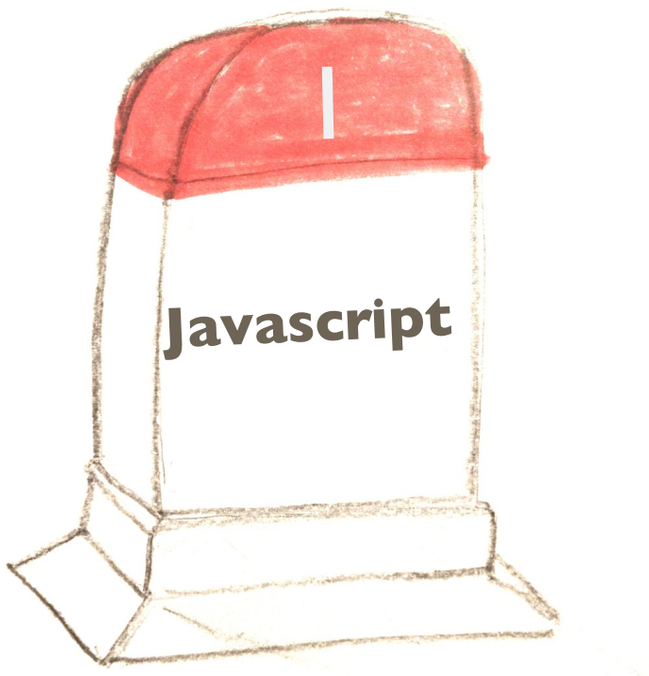
```
{$rename : {"oldFieldName" : "newFieldName"} }
```

```
db.personen.update({},{$rename:{"strasse":"adresse.strasse"}},0,1)
db.personen.update({},{$rename:{"plz":"adresse.plz"}},0,1)
db.personen.update({},{$rename:{"ort":"adresse.ort"}},0,1)
```



Field Splitting

```
var personen = db.personen.find({})
personen.forEach( function(p) {
   if (p.adresse != null) {
        var original = p.adresse.strasse
        var strasse = original.match(/[A-Za-z\.]+/)[0]
        var nummer = original.match(/[0-9]+[a-z]*/)[0]
        p.adresse.strasse = strasse
        p.adresse.hausnummer = nummer
        db.personen.save(p)
}})
```

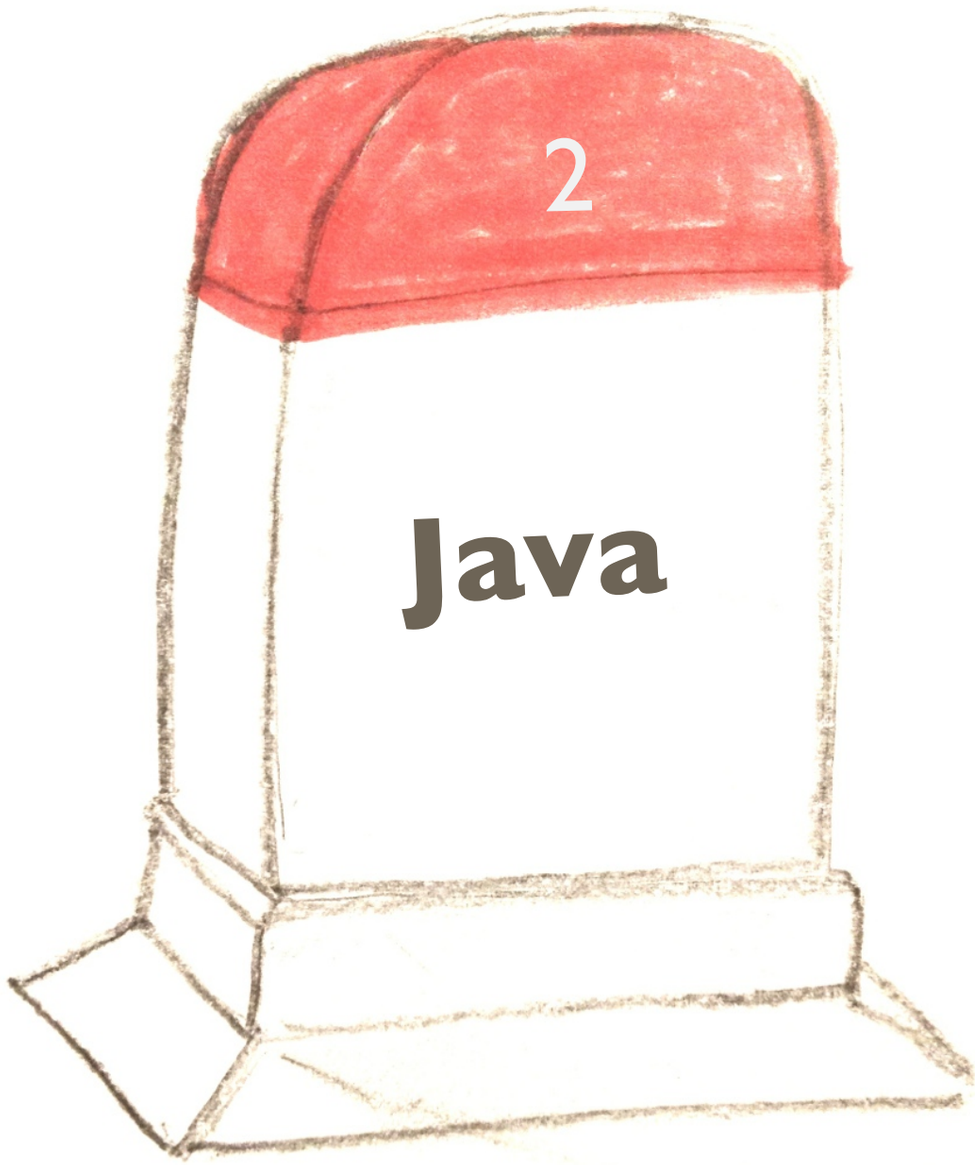


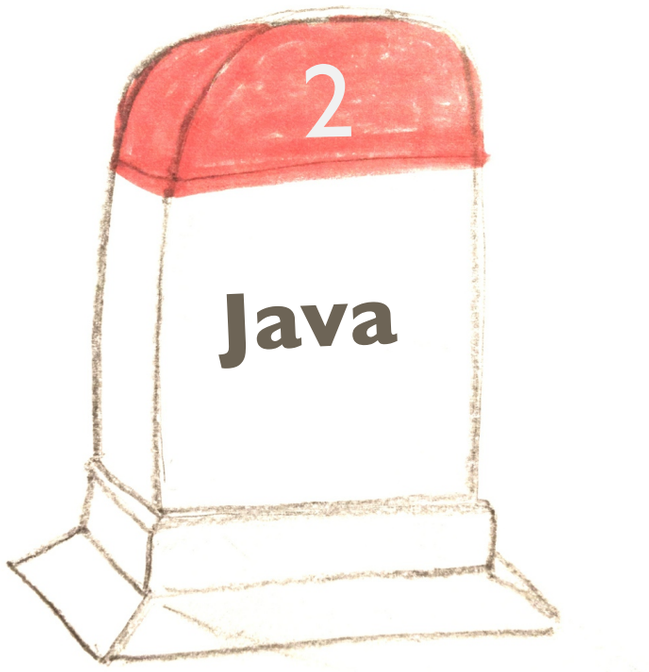
Console Skripting

```
> mongo setup.js migrate.js teardown.js
```

Console Skripting

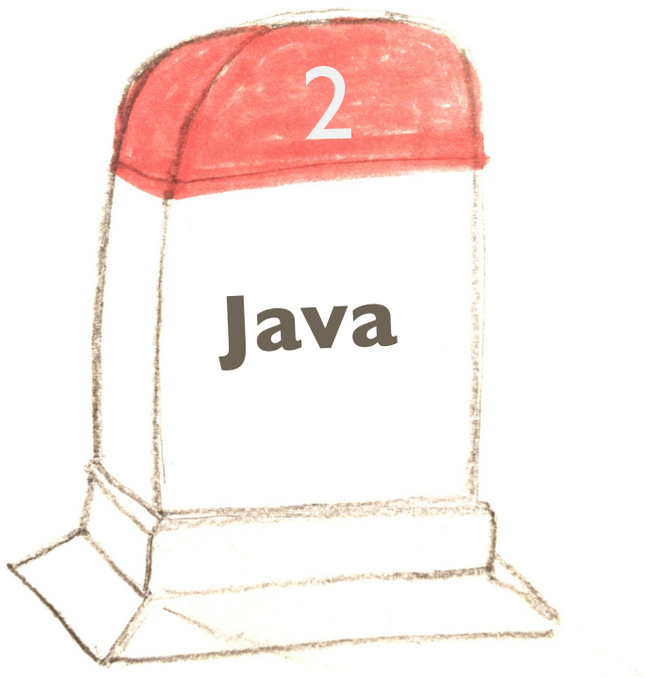
DEMO


2
Java



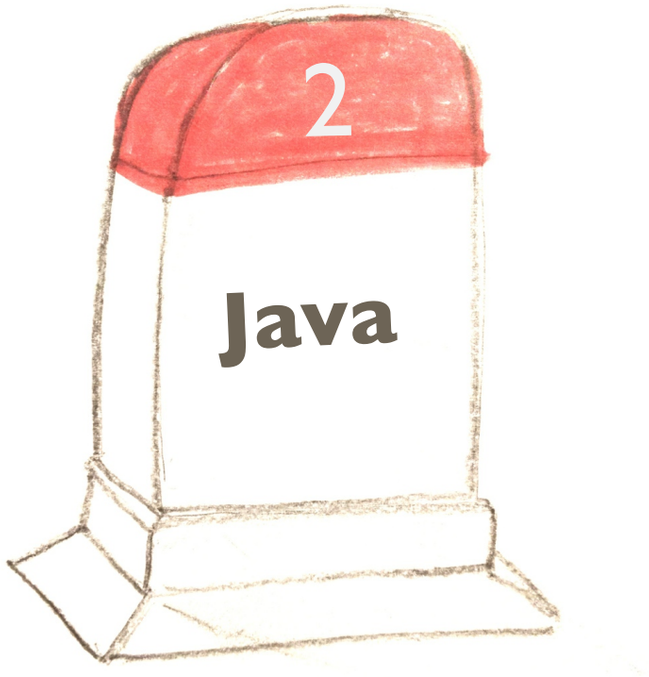
MigrateAdresse.java

```
DBObject person = ...

DBObject adresse = new BasicDBObject();
adresse.put("strasse", person.get("strasse"));
adresse.put("plz", person.get("plz"));
adresse.put("ort", person.get("ort"));

person.put("adresse", adresse);

person.removeField("strasse");
person.removeField("plz");
person.removeField("ort");

collection.save(person);
```



MigrateAdresse.java

```
copy(person, "strasse", "adresse.strasse")
copy(person, "plz", "adresse.plz")
copy(person, "ort", "adresse.ort")

collection.save(person);
```

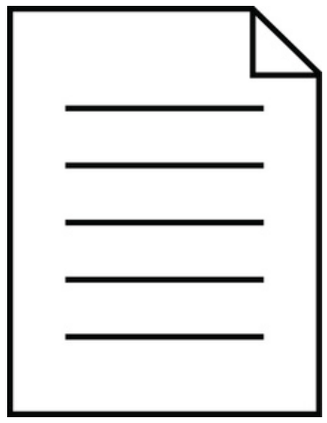


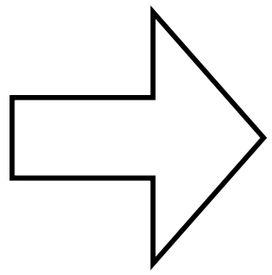
MigrateAdresseTest.java

```
@Test
public void testMigrateAdresse() {
  // given
  String json = getFileAsJson(this.getClass(), "person_v1.json");
  DBObject person = (DBObject) JSON.parse(json);
  // when
  new MigrateAdresse().migrate(person);
  // then
  DBObject adresse = (DBObject) person.get("adresse");
  assertThat(adresse.get("plz")).isEqualTo("10245");
}
```

Migration vor Deployment

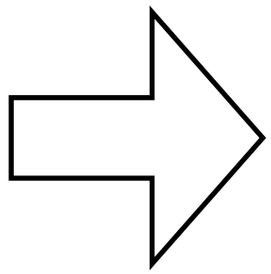
(1)





mongoDB

(2)



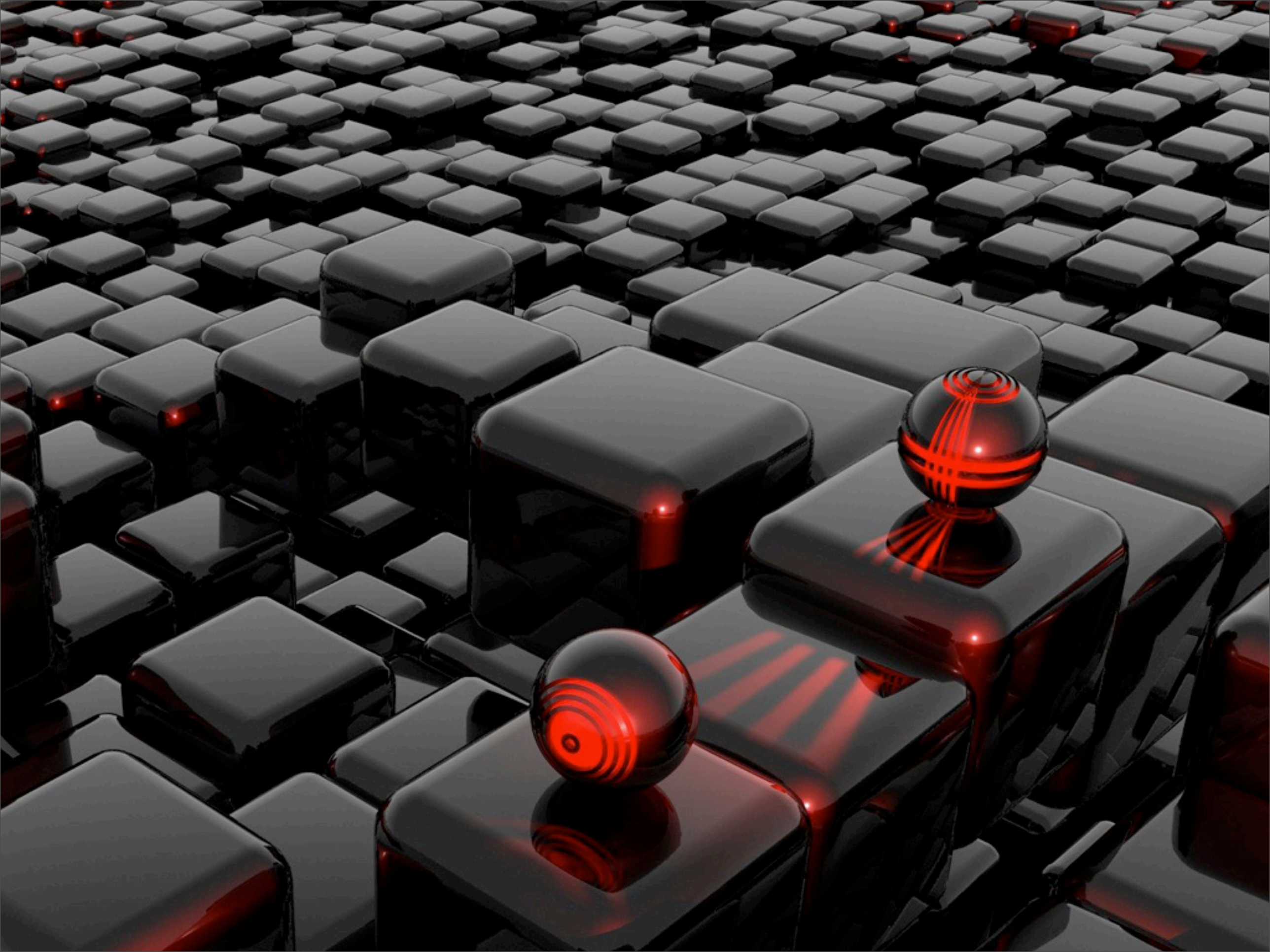
Produktiv System

Continuous Deployment

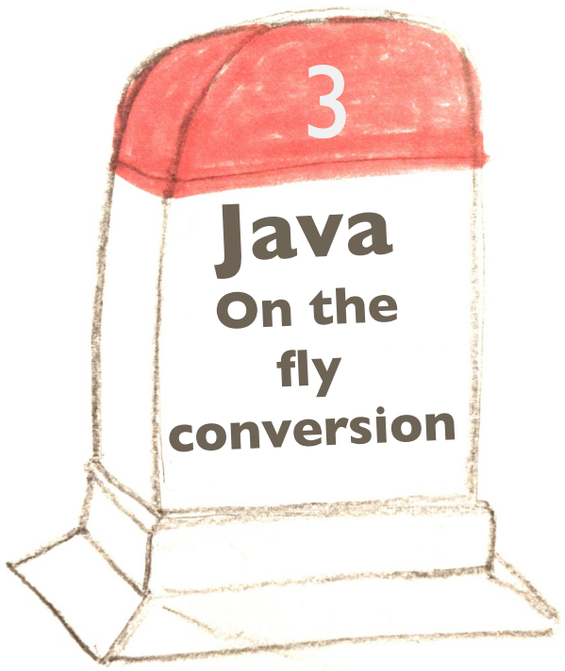
25 / Tag

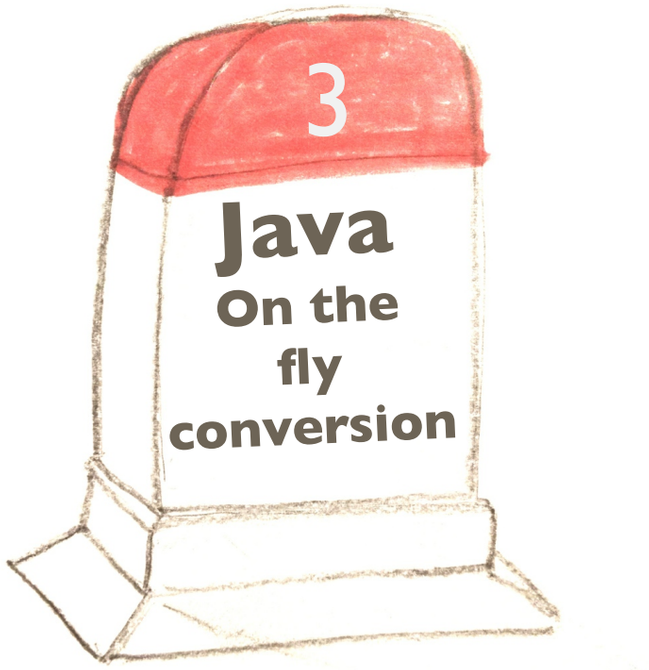
Produktiv System

Continuous Deployment



On the fly version conversion

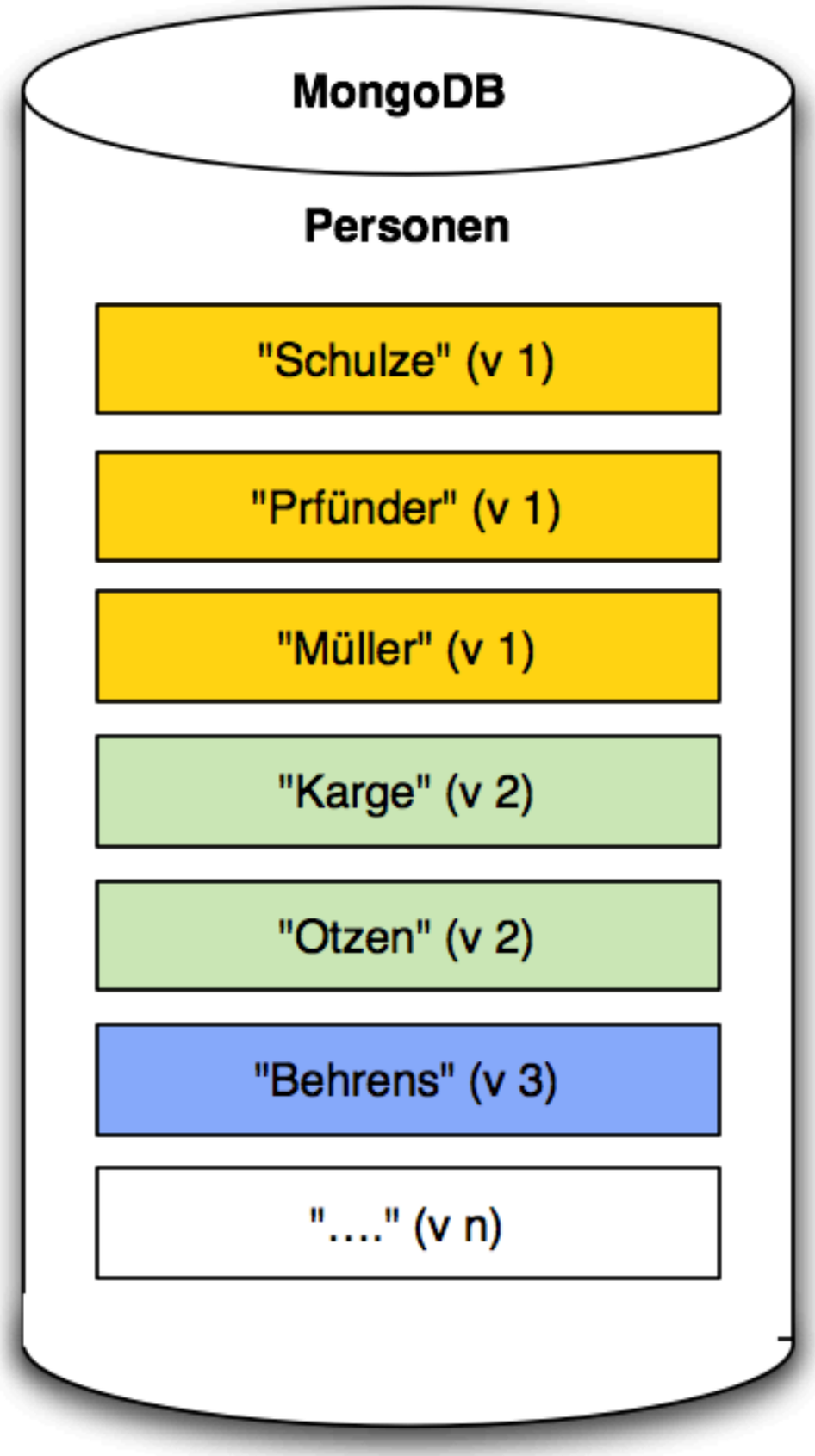


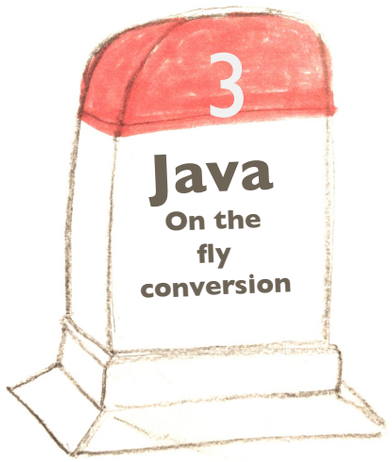



```
{
  "class" : "Person",
```
v1

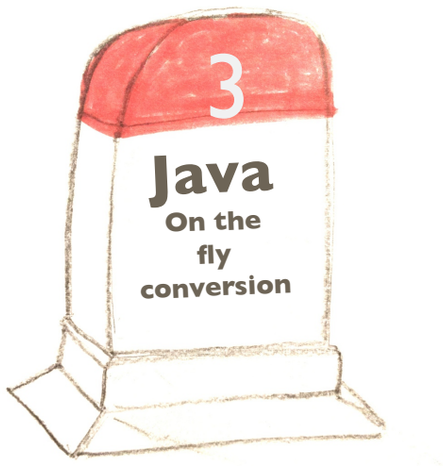
```
{
  "class":"Person",
  "name":"Schulze"
```
v2

```
{
  "class":"Person",
  "name":"Schulze",
  "adresse":{
    "strasse":"Sonntagstr.",
    "hausnummer":"20",
    "plz":"10245",
    "ort":"Berlin"
  }}
```
v3



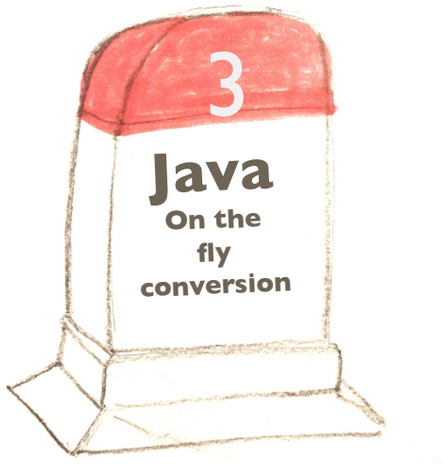


Version Converter
get Targed Version()
convert(DBObject): DBObject
{ordered}
*
Composite
Version Converter
registerConverter(VC)
model
version
.txt
Integration Test
Version Converter
Abstract impl
1->2
2->3
...
1->2
Test
2->3
Test
Test



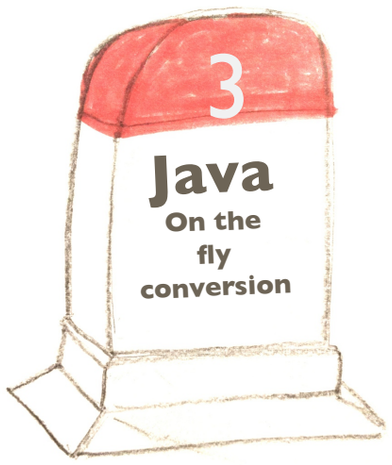
Version Converter Beispiel

```
5
6   public class VersionConverter_1to2_Adresse {
7
8     public DBObject convert(DBObject person) {
9       DBObject adresse = new BasicDBObject();
10      adresse.put("strasse", person.get("strasse"));
11      adresse.put("plz", person.get("plz"));
12      adresse.put("ort", person.get("ort"));
13
14      person.put("adresse", adresse);
15
16      person.removeField("strasse");
17      person.removeField("plz");
18      person.removeField("ort");
19
20      return person;
21    }
22  }
```



Version Converter Unit Test

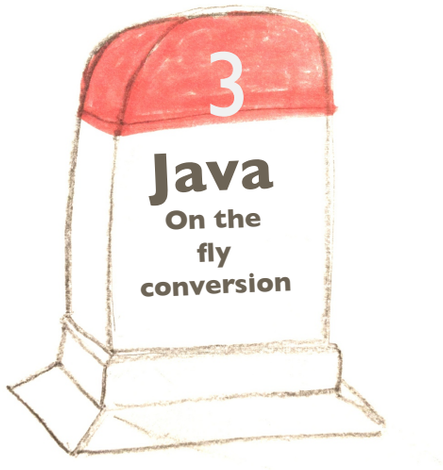
```
 9
10  public class VersionConverter_1to2_AdresseTest {
11
12      @Test
13      public void testMigrateAdresse() {
14          String json = getFileAsJson(this.getClass(), "person_v1.json
15          DBObject person = (DBObject) JSON.parse(json);
16
17          new VersionConverter_1to2_Adresse().convert(person);
18
19          DBObject adresse = (DBObject) person.get("adresse");
20          assertThat(adresse.get("plz")).isEqualTo("10245");
21      }
22  }
23
```



Composite Version Converter Beispiel

model
Version.txt
3

```
public class PersonVersionConverter
      extends CompositeVersionConverter {

  public PersonVersionConverter() {
      setCurrentVersion(getFileContent("/domainModel-version.txt"));
      registerConverter(new VersionConverter_1to2_Adresse());
      registerConverter(new VersionConverter_2to3_HausnummerExtraktion());
      // ...
  }
```



Composite Version Converter Integrations Test

person_v1.json
person_v2.json
person_v3.json
PersonVersionConverter
PersonVersionConverterTest

v1

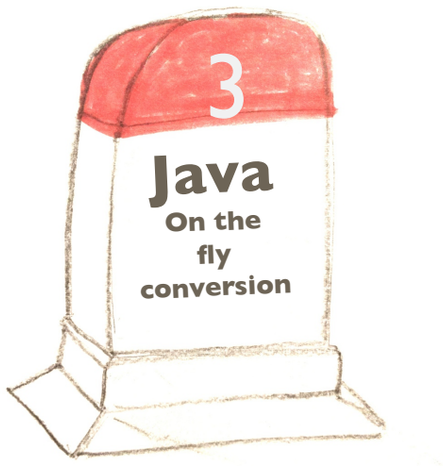
```
{
  "class" : "Person",
  "name":"Schulze",
  "strasse":"Sonntagstr. 20",
  "plz":"10245",
  "ort":"Berlin"
}
```

v2

```
{
  "class":"Person",
  "name":"Schulze",
  "adresse":{
    "strasse":"Sonntagstr. 20",
    "plz":"10245",
    "ort":"Berlin"
  }}
```

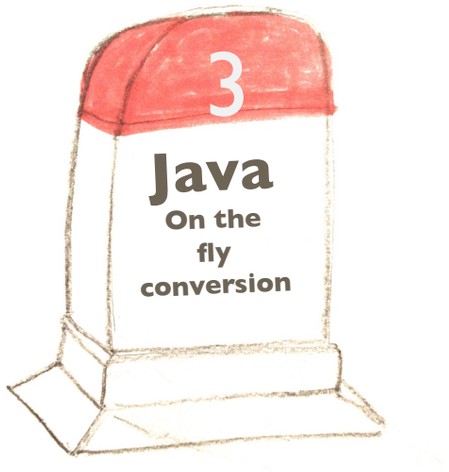
v3

```
{
  "class":"Person",
  "name":"Schulze",
  "adresse":{
    "strasse":"Sonntagstr.",
    "hausnummer":"20",
    "plz":"10245",
    "ort":"Berlin"
  }}
```



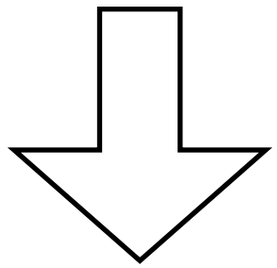
Version Conversion Integrations Test

```
11
12  public class PersonVersionConverterTest {
13
14      private JSONDeserializer<Person> flexJson;
15
16      @DataProvider
17      public Object[][] alleVersionen() {
18          return new Object[][] {
19                  {"person_v1.json"},
20                  {"person_v2.json"},
21                  {"person_v3.json"}
22          };
23      }
24
25      @Test(dataProvider = "alleVersionen")
26      public void versionConversion_and_deserialization(String filename) {
27          // given
28          String json = getFileAsJson(this.getClass(), filename);
29          DBObject dbObject = (DBObject) JSON.parse(json);
30
31          // when
32          DBObject currentVersion = new PersonVersionConverter().convert(dbObject);
33
34          // then
35          flexJson = new JSONDeserializer<Person>();
36          Person person = flexJson.deserialize(currentVersion.toString());
37          assertThat(person).isNotNull();
38          assertThat(person.getName()).isEqualTo("Schulze");
39      }
40  }
```



Queries

```
personen = getCollection().find(queryBuilder().put("name").is("Schulze"))
```



```
personen.addAll(
  getCollection().find(queryBuilder().put("name").is("Schulze"))
)
personen.addAll(
  getCollection().find(queryBuilder().put("neuerName").is("Schulze"))
)
```

Straßenbahn 901
Zoo/Uni

Mülheimer Str.

Am Waldessaum

Carl-Benz-Str.

Richtung Autob... Duisbu...

Sonderlinie
Bus 926
Uni-Nord
Bus 933

AG
Horn-von Hoegen

BEREICH M

MA
MB
MC
MD
ME
MF
MG
MM
Mensa

NETZ
ZBT
LT
LS
inHaus 1
inHaus 2
SE
Gästehaus
LR
Gerhard-Mercator-Haus
LG

Bus 924, 926
Uni-Nord/Lotharstr.

Lotharstr.

Richtung Zentrum/Hbf.

901
Schweizer Str.

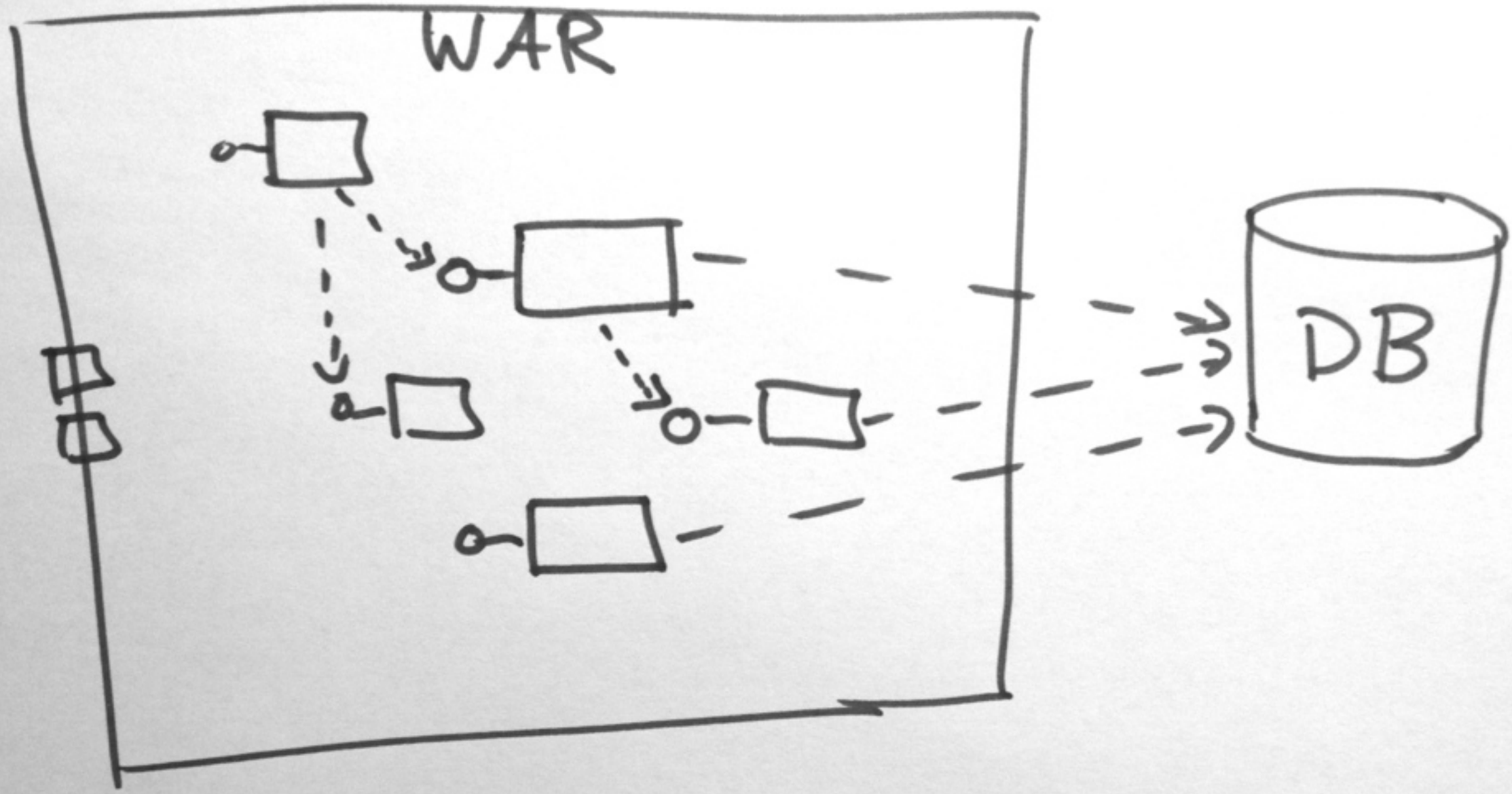
SG

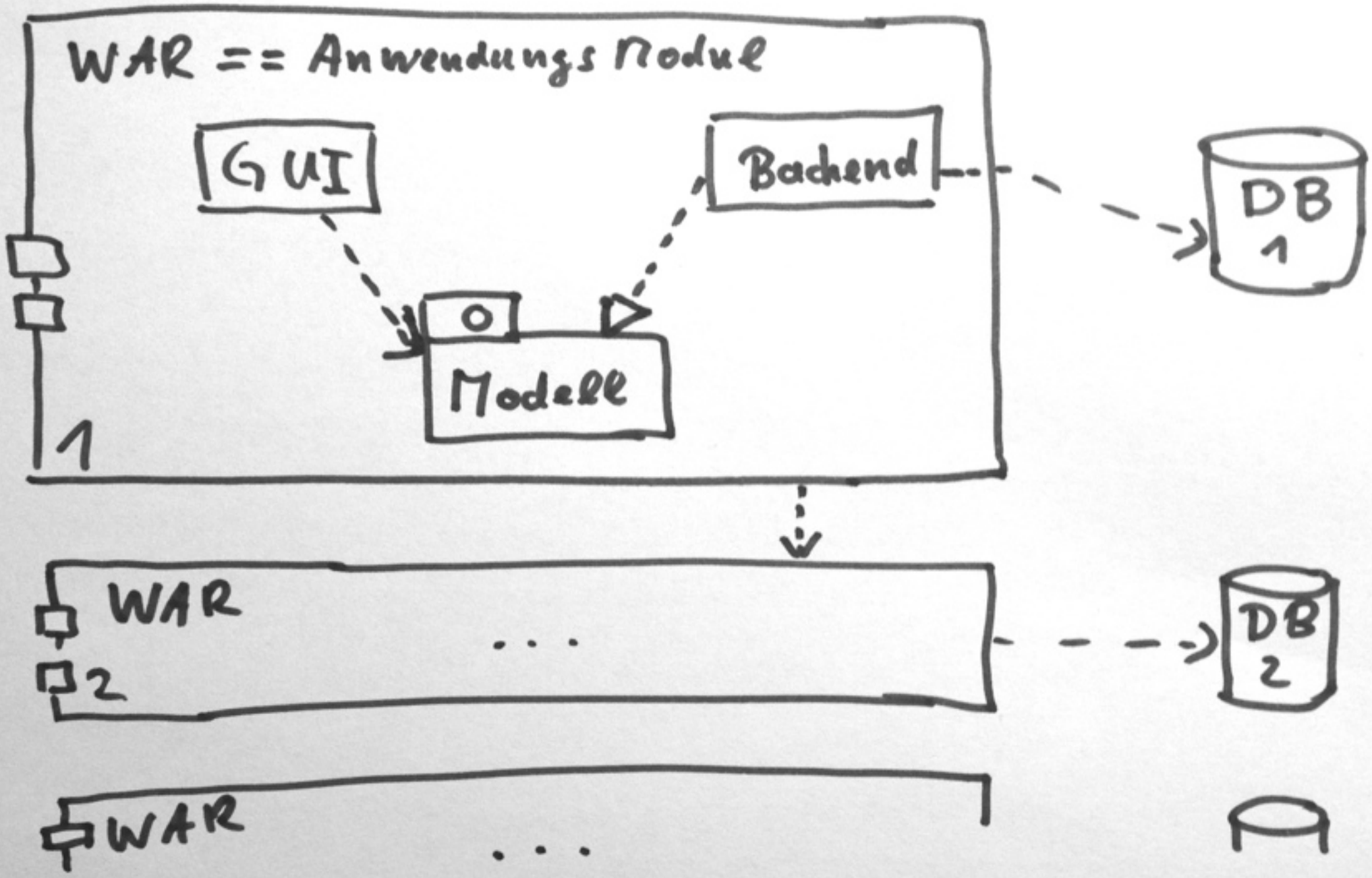
Geibelstr.

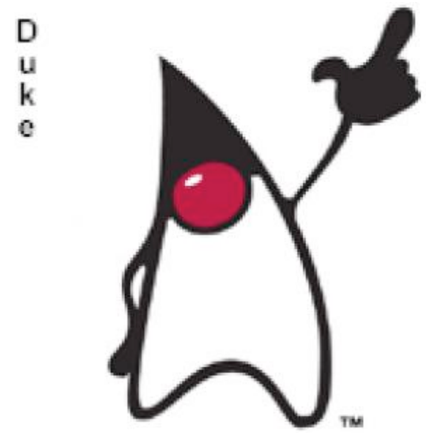
Sternbusch

Holteistr.

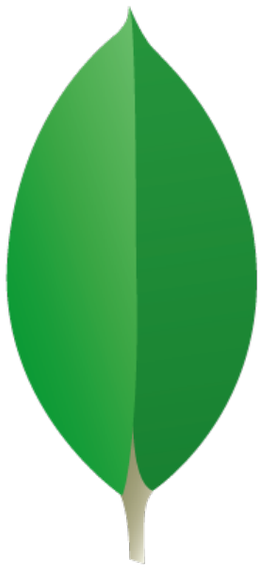
Universität
Bus 924, 933
Sonderlinie


WAR
DB


WAR == Anwendungs Modul
GUI
Backend
DB
1
Modell
1
WAR
2
. . .
DB
2
WAR
. . .

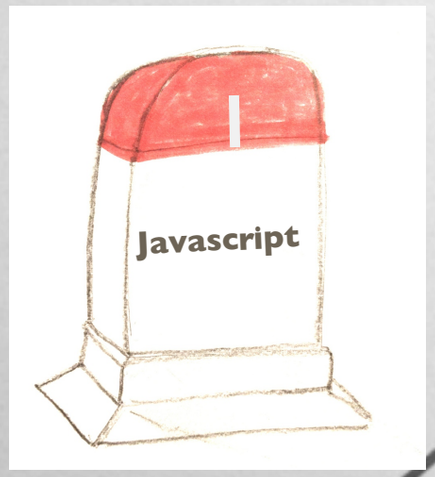


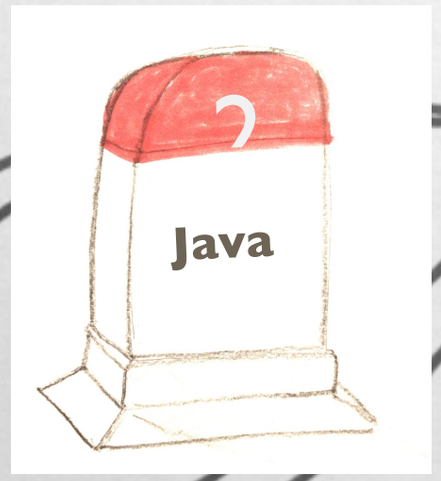
Domain Model

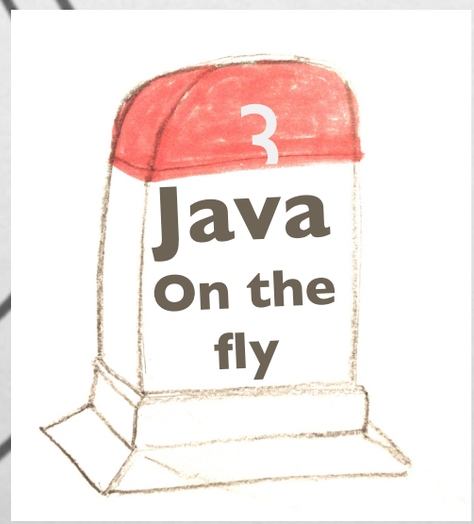


JSON Structure

Schema Evolution



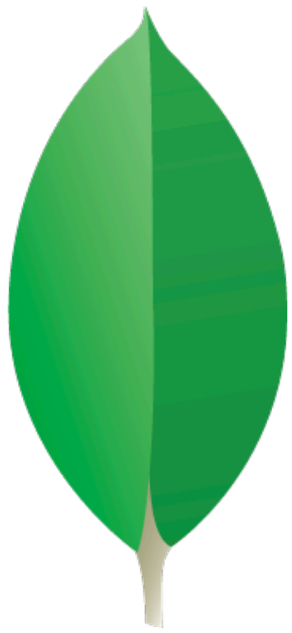




Ausblick

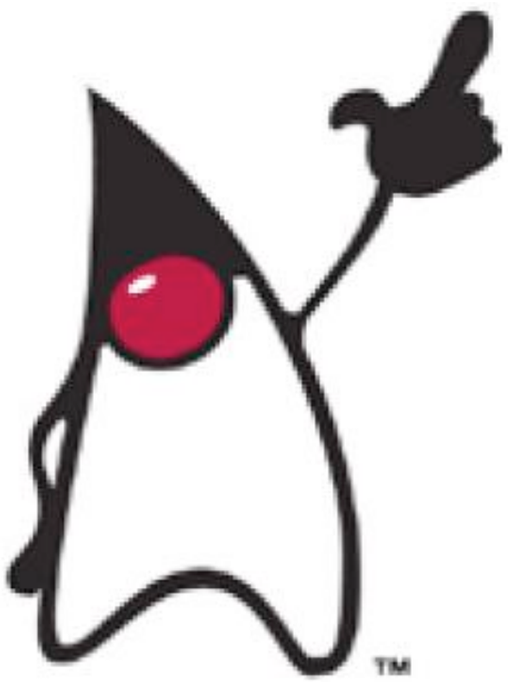
VersionConversion integration in OD Mapper

MongoDB User Group Berlin
in Gründung
@MUGBerlin



mongoDB

für Java Entwickler und Architekten - Schema Evolution und Maintenance



```
{
    "name" : "Timmo Freudl-Gierke",
    "twitter" :  "@timmo_gierke",
    "blog" : "http://blog-it.hypoport.de/"
}
```