



Karsten Silz

Technologien & Tools schneller
auswählen durch Coding mit JHipster

#ittage



FOLIEN & VIDEOS

ZUSATZ-INFOS

FEEDBACK

SELBST LOSLEGEN!

BPF.LI/RIB

**TECHNOLOGIEN & TOOLS
SCHNELLER AUSWÄHLEN
MIT JHIPSTER**

AGENDA

MEHR ALS NUR **JAVA BACK-END** FÜR UNS

JAVA: **MEHR** TECHNOLOGIEN **ÖFTER**

WIE WÄHLEN WIR TECHNOLOGIEN AUS?

WAS IST **JHIPSTER**?

WIE **HILFT** JHIPSTER?

MEHR ALS NUR **JAVA BACK-END** FÜR UNS

JAVA: **MEHR** TECHNOLOGIEN **ÖFTER**

WIE WÄHLEN WIR TECHNOLOGIEN AUS?

WAS IST **JHIPSTER**?

WIE **HILFT** JHIPSTER?

MEHR ALS NUR
JAVA BACK-END

FRÜHER

BACK-END

FRONT-END

APP SERVER

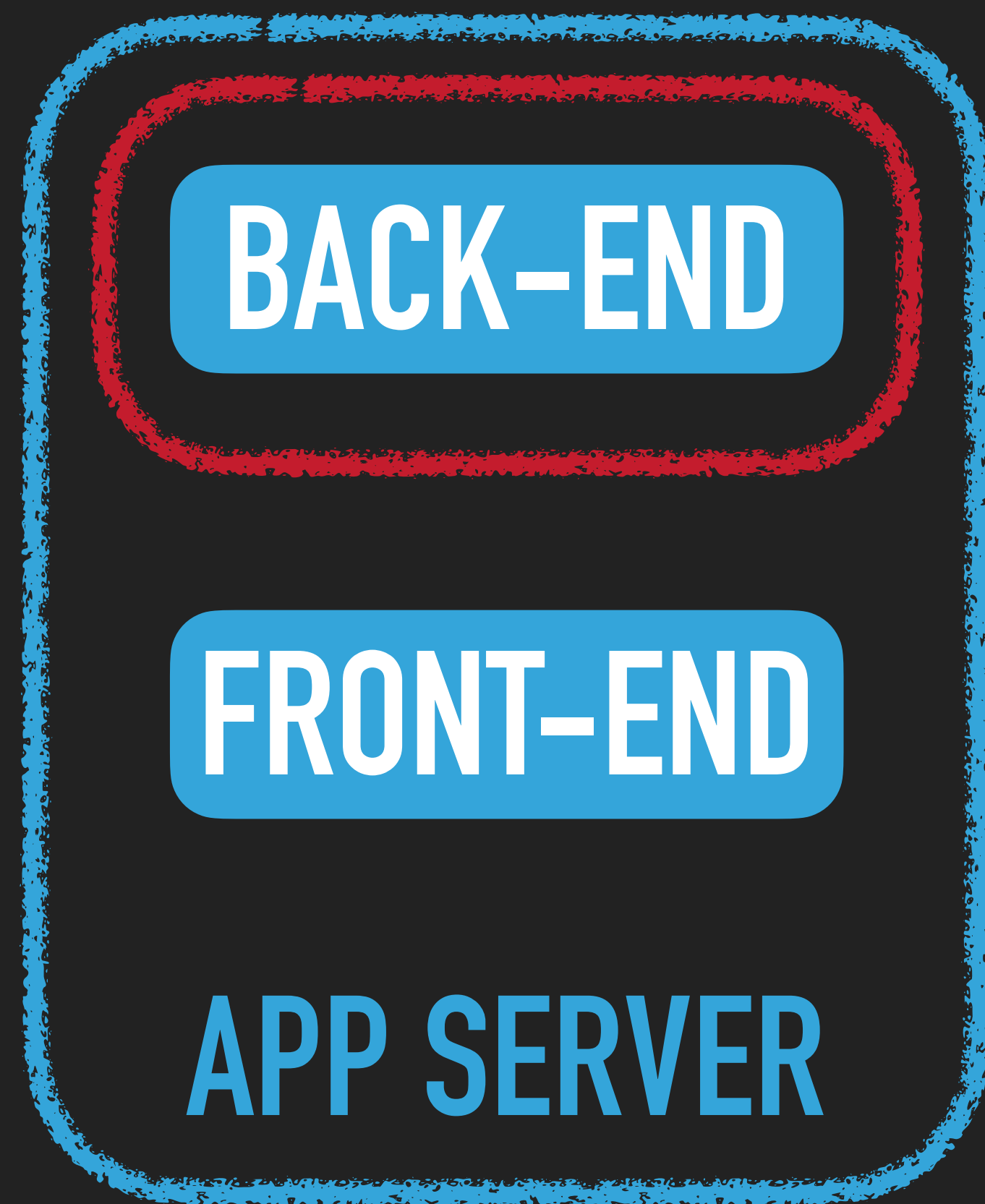
DB

QA

OPS

SUPPORT

ÄNDERUNGEN!



QA

OPS

SUPPORT

MICROSERVICES & CONTAINER

BACK-END

QA

OPS

SUPPORT

FRONT-END

MICROSERVICES & CONTAINER

BACK-END

QA

OPS

SUPPORT

FRONT-END

**FULL-STACK
DEVELOPER**

BACK-END

FRONT-END

QA

OPS

SUPPORT

**FULL-STACK
DEVELOPER**

BACK-END

FRONT-END

QA

OPS

SUPPORT

SHIFT LEFT

BACK-END

FRONT-END

QA

OPS

SUPPORT

SHIFT LEFT

BACK-END

FRONT-END

QA

OPS

SUPPORT

DEVOPS

BACK-END

FRONT-END

QA

OPS

SUPPORT

DEVOPS

BACK-END

FRONT-END

QA

OPS

SUPPORT

**SINGLE PAGE
APPLICATIONS**

BACK-END

FRONT-END

QA

QA

OPS

SUPPORT

SUPPORT

JAVA

JS

**SINGLE PAGE
APPLICATIONS**

BACK-END

FRONT-END

QA

QA

OPS

SUPPORT

SUPPORT

JAVA

JS

MOBILE APPS

BACK-END

FRONT-END

FRONT-END

FRONT-END

QA

QA

QA

QA

OPS

SUPPORT

SUPPORT

SUPPORT

SUPPORT

JAVA

JS

IOS

ANDR.

**WOFÜR NEHMEN WIR
CLOUD-KOMPONENTEN?**

FRÜHER



A diagram illustrating the roles and components in software development. On the left, a large light blue rounded rectangle contains three stacked rounded rectangles: a red one at the top labeled 'BACK-END', a blue one in the middle labeled 'FRONT-END', and the text 'APP SERVER' at the bottom. Below this stack is a blue cylinder labeled 'DB'. To the right of the 'BACK-END' box are three blue rounded rectangles labeled 'QA', 'OPS', and 'SUPPORT' respectively.

BACK-END

FRONT-END

APP SERVER

DB

QA

OPS

SUPPORT

HEUTE

BACK-END

FRONT-END

FRONT-END

FRONT-END

QA

QA

QA

QA

OPS

SUPPORT

SUPPORT

SUPPORT

SUPPORT

JAVA

JS

IOS

ANDR.

UNSERE PROJEKTE: IRGENDWO **ZWISCHEN**
"FRÜHER" UND "HEUTE"

MEIN PROJEKT: "HEUTE"

**MEHR ALS NUR
JAVA BACK-END**

**MEHR ALS NUR
JAVA BACK-END**

~~MEHR ALS NUR JAVA BACK-END FÜR UNS~~

JAVA: MEHR TECHNOLOGIEN ÖFTER

WIE WÄHLEN WIR TECHNOLOGIEN AUS?

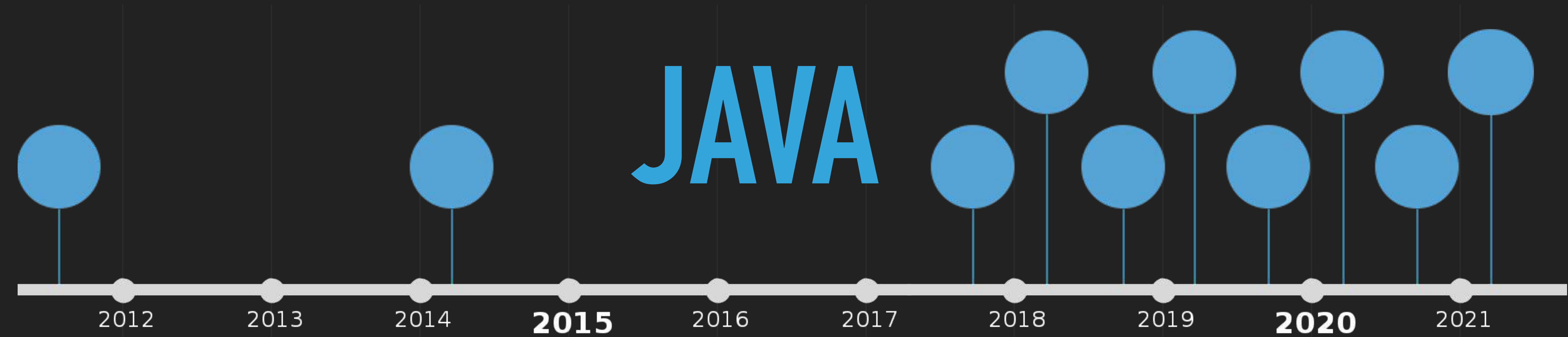
WAS IST JHIPSTER?

WIE HILFT JHIPSTER?

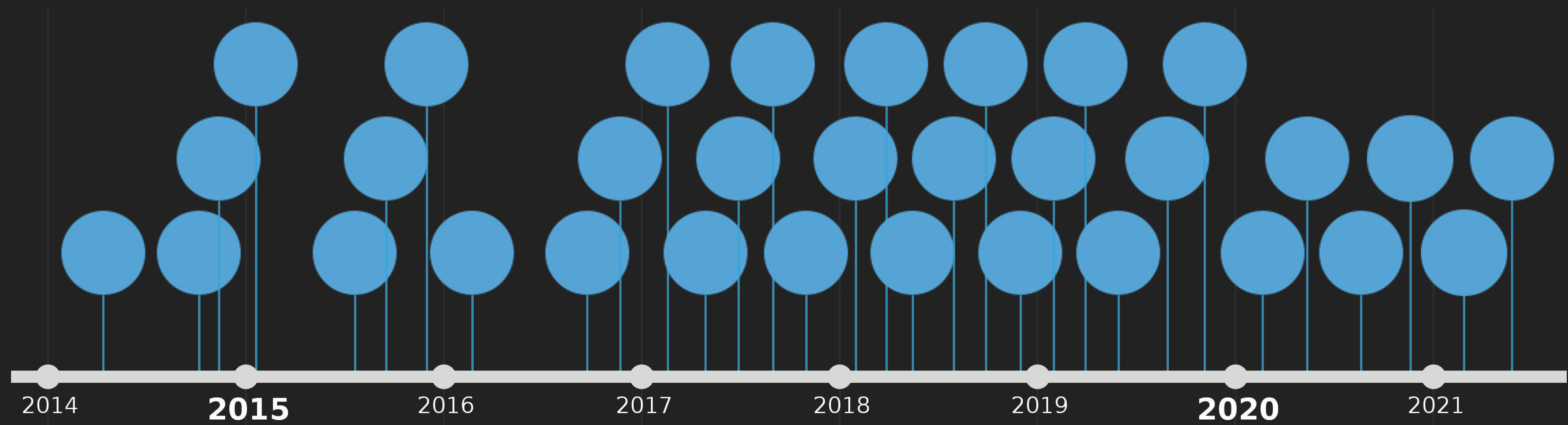
**MEHR TECHNOLOGIEN
ÖFTER**

RELEASES ÖFTER

JAVA

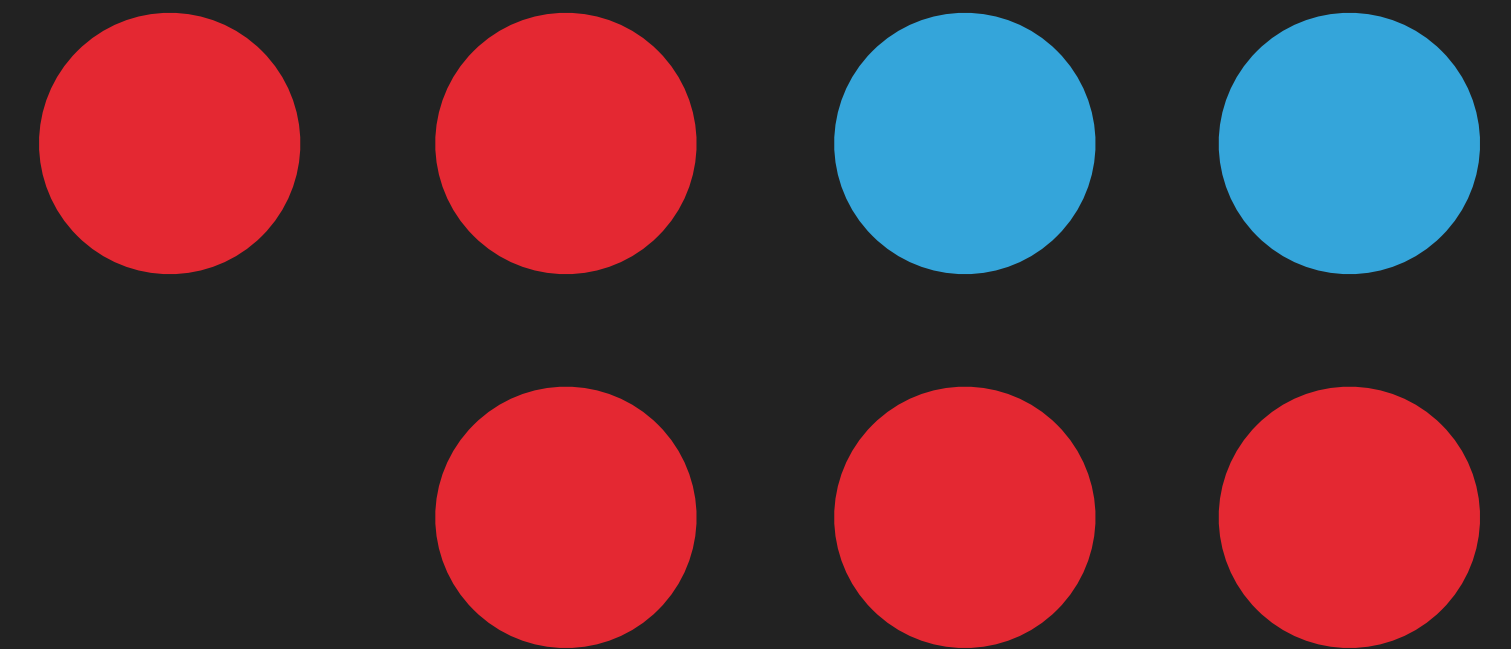


TYPE- SCRIPT



MEHR ALTERNATIVEN IN JAVA

FRAMEWORKS



FRÜHER

HEUTE

JAVA-DISTRIBUTIONEN



FRÜHER

HEUTE

**MEHR TECHNOLOGIEN
ÖFTER**

~~MEHR ALS NUR JAVA BACK-END FÜR UNS~~

~~JAVA. MEHR TECHNOLOGIEN ÖFTER~~

WIE WÄHLEN WIR TECHNOLOGIEN AUS?

WAS IST JHIPSTER?

WIE HILFT JHIPSTER?

VORAUSWAHL

GENAUE PRÜFUNG

WIE WÄHLEN WIR
TECHNOLOGIEN AUS?

KANDIDATEN SIEBEN

ERWEITERBAR?

FUNKTIONIERT?

SICHER?

SKALIERT?

RISIKO?

POPULÄR?

COMPLIANCE?

VORAUSWAHL

UNBEKANNTE TECHNOLOGIEN!

WELCHE KANDIDATEN?

WIE BEWERTEN?

WIE AUSSIEBEN?

VORAUSWAHL: PROBLEME

UNSERE UMGEBUNG

DEBUG

UNSERE DATEN

TEST

UNSERE IDE

CI

BUILD & DEPLOY

MONITORING

GENAUE PRÜFUNG

UNBEKANNTE TECHNOLOGIEN!

WIE PROGRAMM ERSTELLEN?

UNSERE DATEN?

GENAUE PRÜFUNG: PROBLEME

VORAUSWAHL

GENAUE PRÜFUNG

**SO WÄHLEN WIR
TECHNOLOGIEN AUS!**

**SO WÄHLEN WIR
TECHNOLOGIEN AUS!**

~~MEHR ALS NUR JAVA BACK-END FÜR UNS~~

~~JAVA. MEHR TECHNOLOGIEN ÖFTER~~

~~WIE WÄHLEN WIR TECHNOLOGIEN AUS?~~

WAS IST JHIPSTER?

WIE HILFT JHIPSTER?

WAS IST JHIPSTER?



SUPERSTAR-PROGRAMMIERER

KENNT ALLES

KONFIGURIERT ALLES

SCHREIBT VIEL CODE

JHIPSTER IST WIE..



OPEN-SOURCE CODE-GENERATOR

JAVA, NODE.JS, GIT

KEINE PROPRIETÄRE SCHICHT

PRODUKTIONS-REIFER CODE

JHIPSTER

LIVE DEMO

APLIKATION CRUD-CODE

JHIPSTER-WIZARD



APPLIKATION & CRUD- CODE GENERIERT

JHIPSTER:
GLEICHE APPLIKATION MIT
ANDEREN TECHNOLOGIEN!

SEIT 7.5 JAHREN

REGELMÄßIGE RELEASES (AUßER 2020)

CA. 150.000 DOWNLOADS PRO MONAT

GRÜNDER "JAVA ADVOCACY MANAGER" BEI MICROSOFT

SICHER?

TEXT-DATEI ZUR KONFIGURATION

APPLIKATION

DOMAIN ENTITIES

JHIPSTER DOMAIN
LANGUAGE (JDL)

```
entity Product {  
    name String required minlength(4) maxlength(100)  
    price BigDecimal required min(0)  
    size ProductSize required  
    image ImageBlob  
    specifications Blob  
    description TextBlob  
}
```

ENTITY

```
relationship OneToMany {  
    ProductCategory{product} to Product{productCategory(name)}  
}
```

RELATIONSHIPS

```
service * with serviceImpl  
dto * with mapstruct  
paginate * with pagination
```

KONFIGURATION

WIE SUPERSTAR-PROGRAMMIERER

KEINE PROPRIETÄRE SCHICHT

PRODUKTIONS-REIFER CODE

GLEICHE APPLIKATION MIT ANDEREN TECHNOLOGIEN

DAS IST JHIPSTER

DAS IST JHIPSTER

~~MEHR ALS NUR JAVA BACK-END FÜR UNS~~

~~JAVA. MEHR TECHNOLOGIEN ÖFTER~~

~~WIE WÄHLEN WIR TECHNOLOGIEN AUS?~~

~~WAS IST JHIPSTER?~~

WIE **HILFT** JHIPSTER?

WIE HILFT JHIPSTER?

UNBEKANNTE TECHNOLOGIEN!

WELCHE KANDIDATEN?

WIE BEWERTEN?

WIE AUSSIEBEN?

VORAUSWAHL: PROBLEME

UNBEKANNTE TECHNOLOGIEN!

WIE PROGRAMM ERSTELLEN?

UNSERE DATEN?

GENAUE PRÜFUNG: PROBLEME

JHIPSTER:
GLEICHE APPLIKATION MIT
ANDEREN TECHNOLOGIEN!

VORAUSWAHL: DURCH JHIPSTER

GENAUE PRÜFUNG: KOMPLETTE APPLIKATION,
VON UNBEKANNTEN CODE LERNEN?

WIE HILFT JHIPSTER?

DATENBANKEN: ALLE RELATIONALEN AUSSER DB2,
MONGODB, CASSANDRA, NEO4J

FRAMEWORK: SPRING BOOT, QUARKUS,
MICRONAUT, NODE.JS, .NET

FRONT-END: REACT, ANGULAR, VUE.JS

VORAUSWAHL: JHIPSTER

SPRACHE, ARCHITEKTUR, BUILD TOOL,
AUTHENTIFIZIERUNG, CACHING, TESTS,
CONTINUOUS INTEGRATION, DEPLOYMENT,
MONITORING, ...

VORAUSWAHL: JHIPSTER

APPLIKATION MIT ALLEN TECHNOLOGIEN GENERIEREN

UNSER DATEN-MODELL ALS JDL IMPORTIEREN

GENAUE PRÜFUNG: JHIPSTER

~~UNSERE UMGEBUNG~~

~~DEBUG~~

~~UNSERE DATEN~~

~~TEST~~

~~UNSERE IDE~~

~~CI~~

~~BUILD & DEPLOY~~

~~MONITORING~~

GENAUE PRÜFUNG: JHIPSTER

...UND WIR MÜSSEN DEN
JHIPSTER-CODE NOCH NICHT
MAL VERSTEHEN!

**VON UNBEKANNTEN
CODE LERNEN?**

SQL VS NOSQL: ENTITIES

ENTITY:

```
public class Product implements Serializable {  
  
    @Id  
    private Long id;  
  
    @NotNull  
    @Size(min = 2, max = 90)  
    private String name;
```

ENTITY: NOSQL

```
@Document(collection = "product")
public class Product implements Serializable {

    @Id

    private String id;

    @NotNull
    @Size(min = 2, max = 90)
    @Field("name")
    private String name;
```


ENTITY: SQL

```
@Entity
@Table(name = "product")
@Cache(usage = CacheConcurrencyStrategy.READ_WRITE)
public class Product implements Serializable {
```

```
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE,
                    generator = "sequenceGenerator")
    @SequenceGenerator(name = "sequenceGenerator")
```

```
    private Long id;
```

```
    @NotNull
    @Size(min = 2, max = 90)
    @Column(name = "name", length = 90, nullable = false, unique = true)
    private String name;
```

ANDERER TYP

ENTITY: NOSQL

```
@Document(collection = "product")
public class Product implements Serializable {

    @Id

    private String id;

    @NotNull
    @Size(min = 2, max = 90)
    @Field("name")
    private String name;
```

NOSQL & SQL ENTITIES
ÄHNLICH!

ANGULAR VS REACT

CRUD-CODE:

```
<div>
  <h2 id="jhipsterApp.product.home.title" class="page-heading">
    <span data-bbox="115 425 135 445" class="icon">
      <img alt="plus icon" data-bbox="115 425 135 445" />
    </span>
    <span data-bbox="145 425 915 445" class="page-heading">Products</span>
  </h2>
  <div id="jh-create-entity" data-bbox="115 525 915 855">
    <div data-bbox="115 575 915 655" class="float-right">
      <button data-bbox="115 575 915 655" class="btn btn-primary">Create new Product</button>
    </div>
  </div>
</div>
```

CRUD-CODE: ANGULAR – HTML

PRÄSENTATIONS-CODE IM TEMPLATE

STANDARD HTML-TAGS, PROPRIETÄRE ATTRIBUTE

```
<div>
  <h2>Products</h2>
  <span jhiTranslate    ="jhipsterApp.product.home.title">Products
</span>
  <button id="jh-create-entity" [routerLink]="['/product/new']"
    class="btn btn-primary float-right jh-create-entity
    create-product">
    <fa-icon          icon="plus"</fa-icon>
    <span jhiTranslate    ="jhipsterApp.product.home.createLabel">
      Create new Product
    </span>
  </button>
</div>
```

CRUD-CODE: REACT - JAVASCRIPT

VERMISCHT BUSINESS-LOGIK & PRÄSENTATION

KEINE STANDARD HTML-TAGS

CRUD-CODE: ANGULAR - HTML

```
<div>
  <h2 id="page    -heading">
    <span jhiTranslate    ="jhipsterApp.product.home.title">Products
  </span    >
  <button id="jh-create-entity" [routerLink]="['/product/new']"
    class=    "btn btn-primary float-right jh-create-entity
    create-product">
    <fa-icon    icon="plus"</fa-icon>
    <span jhiTranslate    ="jhipsterApp.product.home.createLabel">
      Create new Product
    </span    >
  </button>
</h2>
```


ANGULAR: TEMPLATES
REACT: DEKLARATIV

**WIR KÖNNEN SOGAR VOM
UNBEKANNTEN CODE LERNEN!**

LIVE DEMO

APPLIKATION BEGUTACHTEN



APPLIKATION BEGUTACHTEN

TECHNOLOGIE **NICHT** IN JHIPSTER

WIE **PRODUKTIV** MIT TECHNOLOGIEN?

WO **HILFT** JHIPSTER **NICHT**?

JHIPSTER SPART ZEIT...

**...ABER SIE MÜSSEN
ENTSCHEIDEN!**

JHIPSTER:
GLEICHE APPLIKATION MIT
ANDEREN TECHNOLOGIEN!

VORAUSWAHL: DURCH JHIPSTER

GENAUE PRÜFUNG: KOMPLETTE APPLIKATION,
LERNEN VON FREMDEN CODE!

WIE HILFT JHIPSTER?

WIE HILFT JHIPSTER?

~~MEHR ALS NUR JAVA BACK-END FÜR UNS~~

~~JAVA. MEHR TECHNOLOGIEN ÖFTER~~

~~WIE WÄHLEN WIR TECHNOLOGIEN AUS?~~

~~WAS IST JHIPSTER?~~

~~WIE HILFT JHIPSTER?~~

**TECHNOLOGIEN & TOOLS
SCHNELLER AUSWÄHLEN
MIT JHIPSTER**

WERBUNG

OKTOBER 2021

FULL-STACK **JAVA: SPRING BOOT, ANGULAR, FLUTTER**

FREIBERUFLER/BEFRISTETE ANSTELLUNG

MILTON KEYNES, LONDON, ODER **REMOTE**

SUCHE **PROJEKT!**

The End



FOLIEN & VIDEOS

ZUSATZ-INFOS

FEEDBACK

SELBST LOSLEGEN!

BPF.LI/CIN