

# DOCKER COMPOSE: BEISPIEL DB & UI MIT INTELIJ IDEA UND MAVEN

ANGULAR, SPRING BOOT & POSTGRESQL

SIMON SYMHOVEN

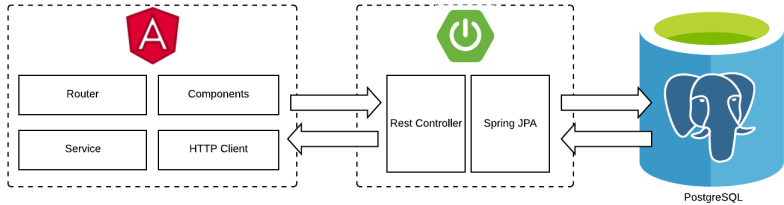
SIMON.SYMHOVEN@HM.EDU  
FAKULTÄT FÜR  
INFORMATIK UND MATHEMATIK



2. JUNI 2020

# MOTIVATION

# DAS PROBLEM



# **GRUNDLAGEN & VORRAUSSETZUNGEN (DOCKER, MAVEN & INTELLIJ)**

- Projektmanagement von Softwareentwicklungsprojekten
- **Fokus:** Builds, Dokumentation, Reporting, Abhängigkeiten, SCM (Software Configuration Management), Releases und Distribution
- Produktivität ↗, Wiederverwendbarkeit ↗
- **pom.xml** als zentrale Steuerungsdatei (5 Bereiche: Koordinaten, Projektbeziehungen, Projektinformationen, Projekteinstellungen und Projektumgebung)
- Basis Lifecycles (clean, compile, test, package, integration-test, verify, install, deploy und site.)
- Plugin Goals, Erweiterung der Lifecycles (z.B. **mvn spring-boot:run**)
- Lokales Repo `${user.home}/.m2/repository` > Remote-Repository

# MAVEN BUILD PROZESS

Project Object Model  
(pom.xml)



Projekt Dateien



**mvn [goal]**



eigenes Projekt  
(Komponente)

externe Abhängigkeiten  
(Komponenten & Bibliotheken)



- Isolierung von Anwendungen und Containervirtualisierung (Betriebssystemvirtualisierung)
- Vereinfachung der Anwendungsbereitstellung
- Image enthält alle benötigten Komponenten (Laufzeitumgebung, Betriebssystem, Libraries, ...)
- ausgeführtes Images ist ein Container
- Dockerfile beinhaltet alle Befehle für die Erstellung eines Images
- **Image erzeugen:** `docker build <Dockerfile> -t <image-name>`
- **Image starten:** `docker run -d <image-name>` (→ **Container**)
- Docker-Plugin<sup>1</sup> für IntelliJ IDEA

---

<sup>1</sup><https://plugins.jetbrains.com/plugin/7724-docker>

# ZURÜCK ZUM PROBLEM: SERVICES STARTEN



1. `docker run -p 5432:5432 --env POSTGRES_PASSWORD=123 --name dbpostgresql postgres`



1. application.properties anpassen (DB-URL: localhost)
2. `mvn spring-boot:run`
1. `docker build -t spring-boot-api .`
2. `docker run -p 8080:8080 --name spring-boot-api --link dbpostgresql:postgres spring-boot-api`



1. `ng serve`
1. `docker build -t angular-app-frontend .`
2. `docker run -p 4200:80 --link spring-boot-api angular-app-frontend`

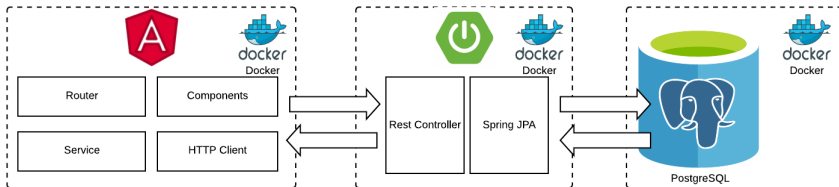


# **DIE LÖSUNG: DOCKER COMPOSE**



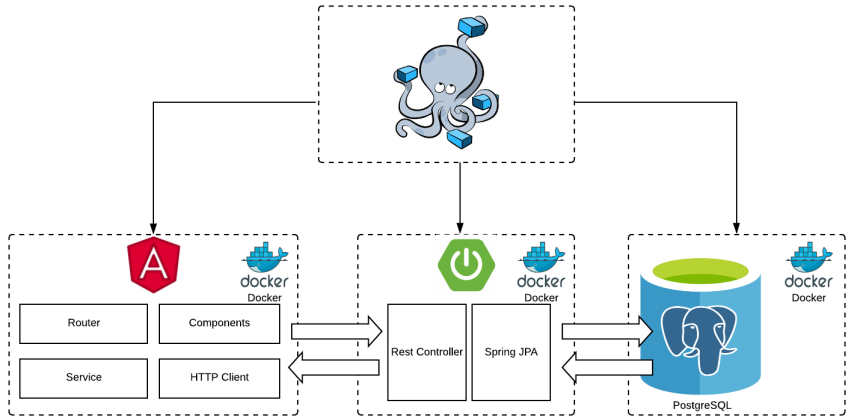
- Tool um mehrere Docker Container zu konfigurieren und auszuführen
- **docker-compose.yml** (Compose File) : Definition der Dienste für die Anwendung
- Alle Dienste laufen auf **einem** Host

# SCHRITT 1: DOCKER CONTAINER FÜR JEDE ANWENDUNG DEFINIEREN



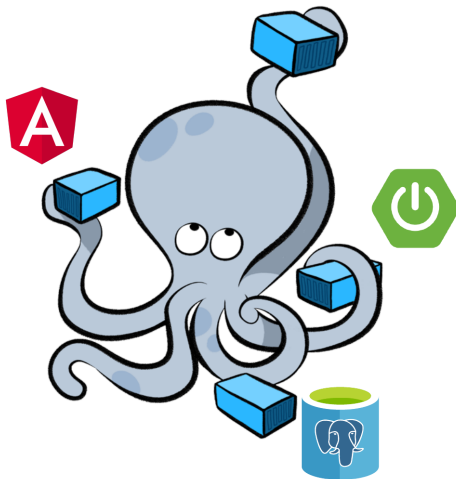


## SCHRITT 2: DOCKER COMPOSE DEFINIEREN





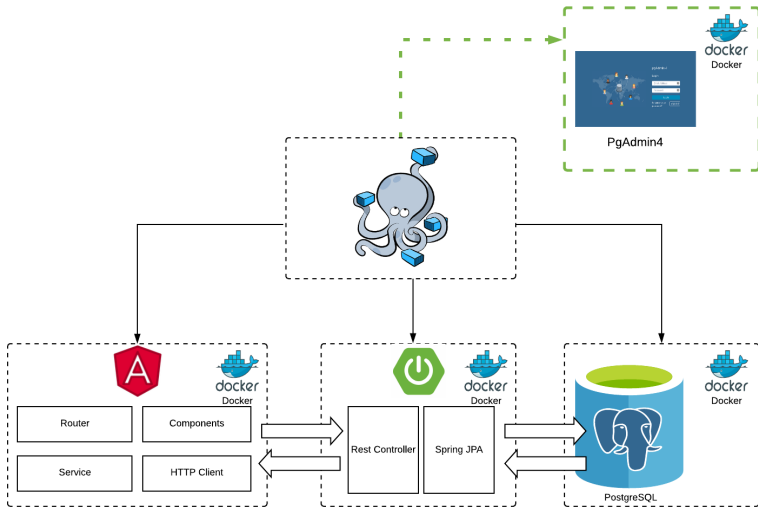
## SCHRITT 3: DOCKER COMPOSE STARTEN



- **docker-compose create:** Services erstellen
- **docker-compose start:** Services starten
- **docker-compose stop:** Services stoppen
- **docker-compose pause:** Services pausieren
- **docker-compose unpause:** Services aufwecken
- **docker-compose ps:** Container auflisten
- **docker-compose images:** Images auflisten
- **docker-compose up:** Services erstellen und starten
- **docker-compose down:** Services stoppen und löschen (inkl. Networks, Images, Volumes)

# LIVE CODING

# DOCKER COMPOSE SERVICE: PGADMIN4



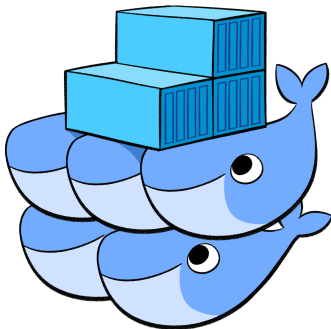


# **FAZIT & AUSBLICK**

- manuelle Verwaltung eines größeren Systems aus mehreren Services wird schnell unübersichtlich und kompliziert
- Docker Compose als Möglichkeit System innerhalb einer Datei zu definieren und zu verwalten
- Unterschiedliche Versionen lassen sich problemlos parallel installieren, starten, stoppen und entfernen
- Applikationen und Abhängigkeiten lassen sich strukturiert abbilden

# AUSBLICK: DOCKER SWARM

- Ausfallsicherheit und Deployment
- Container auf **mehreren** Hosts
- Container-Cluster Management Tool um Applikation in Produktion zu überführen
- → Konkurrenz zu Kubernetes<sup>2</sup> & Mesos<sup>3</sup>



---

<sup>2</sup><https://kubernetes.io/de/>

<sup>3</sup><https://mesos.apache.org>

DANKE FÜR IHRE AUFMERKSAMKEIT!

-  HORN, T.  
**DOCKER.**  
<https://www.torsten-horn.de/techdocs/Docker.html>,  
2015.
-  HORN, T.  
**MAVEN 3.5.**  
<https://www.torsten-horn.de/techdocs/maven.htm>,  
2015.
-  INC., D.  
**OVERVIEW OF DOCKER COMPOSE.**  
<https://docs.docker.com/compose/>, 2020.
-  ÖGGL, B., KOFLER, M.  
***DOCKER - DAS PRAXISBUCH FÜR ENTWICKLER UND DEVOPS-TEAMS.***  
Rheinwerk Computing, Januar 2020.



SYMHOVEN, S.

**SKILLBATZ, A FULLSTACK DOCKERIZED APPLICATION WITH ANGULAR,  
SPRING-BOOT AND POSTGRESQL.**

<https://github.com/simonsymhoven/skillbatz>, May 2020.

NOCH FRAGEN?

