

Anwendungen von Semantik MediaWiki

Thema Docker

Raphael Manke, Patrick Eisele

Institut für Angewandte Informatik und Formale Beschreibungslehre (AIFB)



Motivation

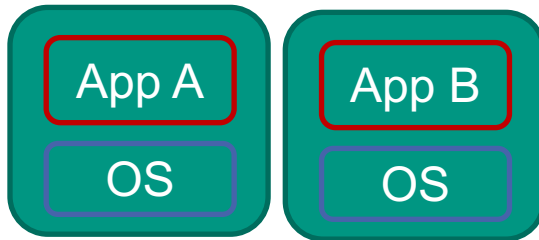
- Sehr zeitaufwändig MediaWiki, Semantic MediaWiki jeweils auf verschiedenen Servern zu installieren, sowie die benötigten Entwicklungsumgebungen aktuell zu halten
- Problem: ... funktioniert aber auf meiner Maschine, gelöst



Ziele

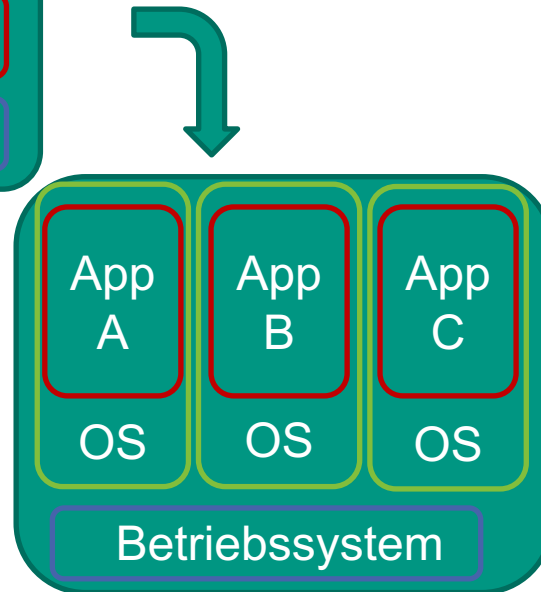
- einfache Möglichkeit ein Semantik MediaWiki aufzusetzen
 - zum Testen / Ausprobieren
 - Demonstrationen
 - Entwicklungsumgebung
 - Deployment
- „Baukastensystem“ für den Benutzer
 - Oberfläche zum konfigurieren
 - click & run
- Migration „alter“ Semantik Media Wikis
 - einspielen einer Datenbank
 - Update auf neuste Version

Bare Metal vs. VM vs. Docker



physischer Server

- hohe Betriebskosten
- ineffiziente Ressourcenauslastung
- schlecht zu skalieren

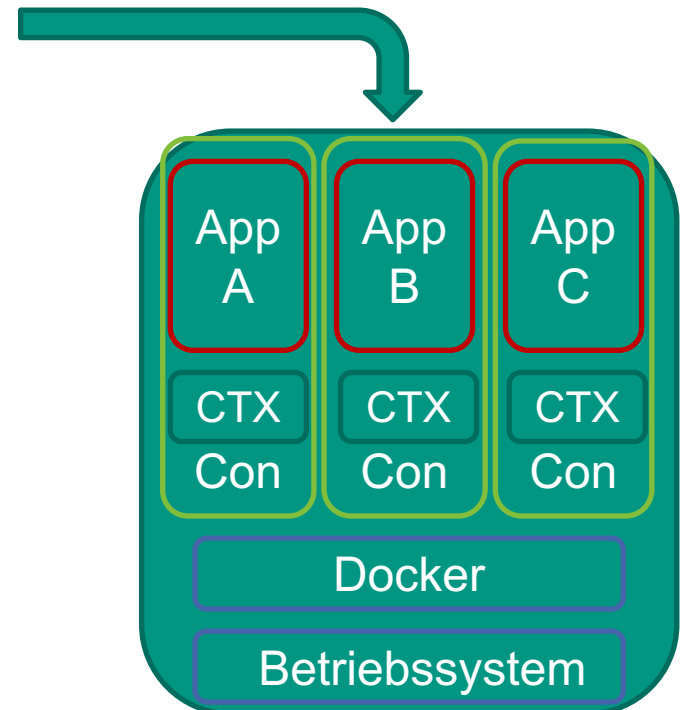


VM Server

- mehrere Anwendungen parallel
- leichtere Verteilung
- fixe Ressourcen-Verteilung
- Gast OS notwendig

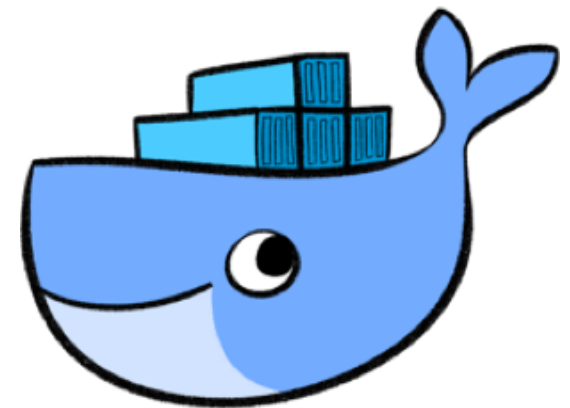
Docker Server

- gemeinsame Ressourcen
- kein OS zu starten
- starke Portabilität



Docker

- Docker ist eine Containerplattform die es erlaubt Anwendungen in Containern auszuführen
 - Container = Anwendung + minimaler Ausführungskontext
 - Verteilung der Anwendung in viele kleine „Container“
- Kapselt alle für Anwendung benötigten Abhängigkeiten in einem virtuellem Container



Docker Grundlagen

■ Image

- „unveränderlicher“ Zustand einer virtuellen Umgebung

■ Container

- Instanz eines Image
- Zusätzliche Schicht auf Image

■ Dockerfile

- Spezifiziert ein Image
- Basis für Container

■ Docker Build

- Baut ein Image aus einem Dockerfile
 - Bsp. `docker build -t smw .`

■ Docker Run

- Erstellt einen Container aus Image
 - Bsp. `docker run -p 8080:8080 smw`

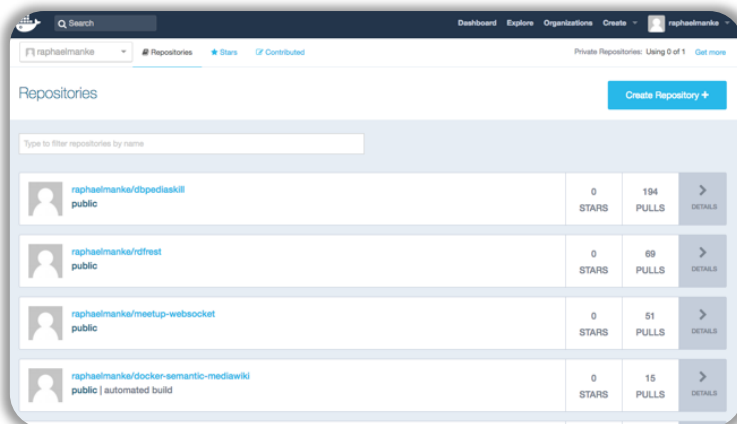
```

1 ARG tag
2 FROM mediawiki:${tag:-1.30}
3 MAINTAINER Raphael Manke, Patrick Eisele
4
5 ENV COMPOSER_ALLOW_SUPERUSER = 1
6
7 RUN apt-get update && apt-get install -y \
8     nano \
9     wget \
10    jq \
11    unzip \
12    libpng-dev
13
14 # install needed php extension
15 RUN docker-php-ext-install gd
16
17 # Copy scripts
18 COPY scripts ./scripts/check_db.sh ./
19 # Run the composer installation
20 RUN sh "install_composer.sh"
21
22 # Install semantic mediawiki
23 RUN php composer.phar require mediawiki/semantic-media-wiki "~2.5" --update-no-dev
24
25 # At containerstart run check script and then http server
26 CMD /var/www/html/check_db.sh && apache2-foreground
27

```

Docker Grundlagen

- Docker compose
 - Komposition von Containern
 - Spezifiziert Konfiguration
 - Bsp docker.compose.yml
 - Startet mehrere Container
 - Bsp. docker-compose up
- Docker Hub
 - Repository für Images
 - Autobuild Funktion

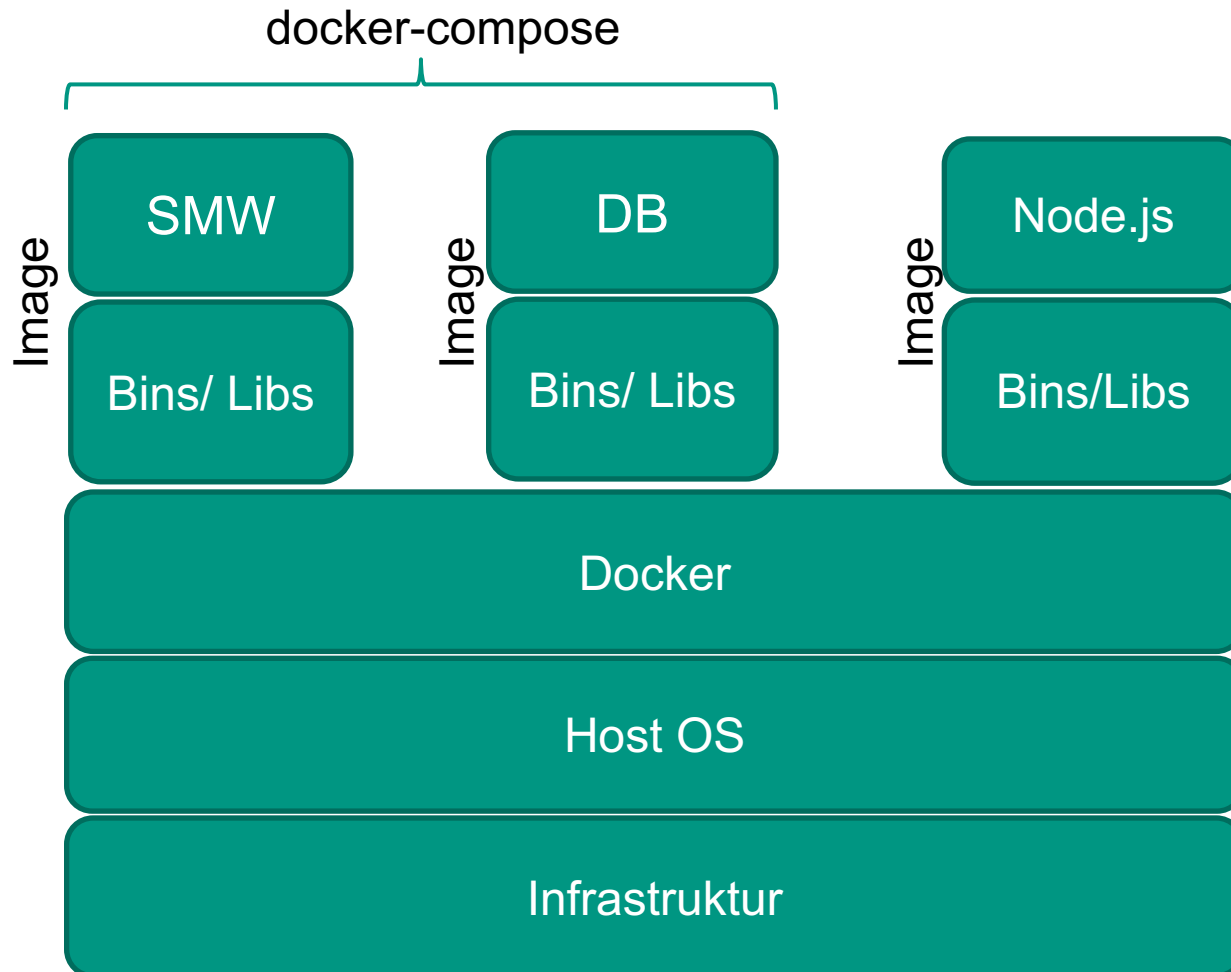


```

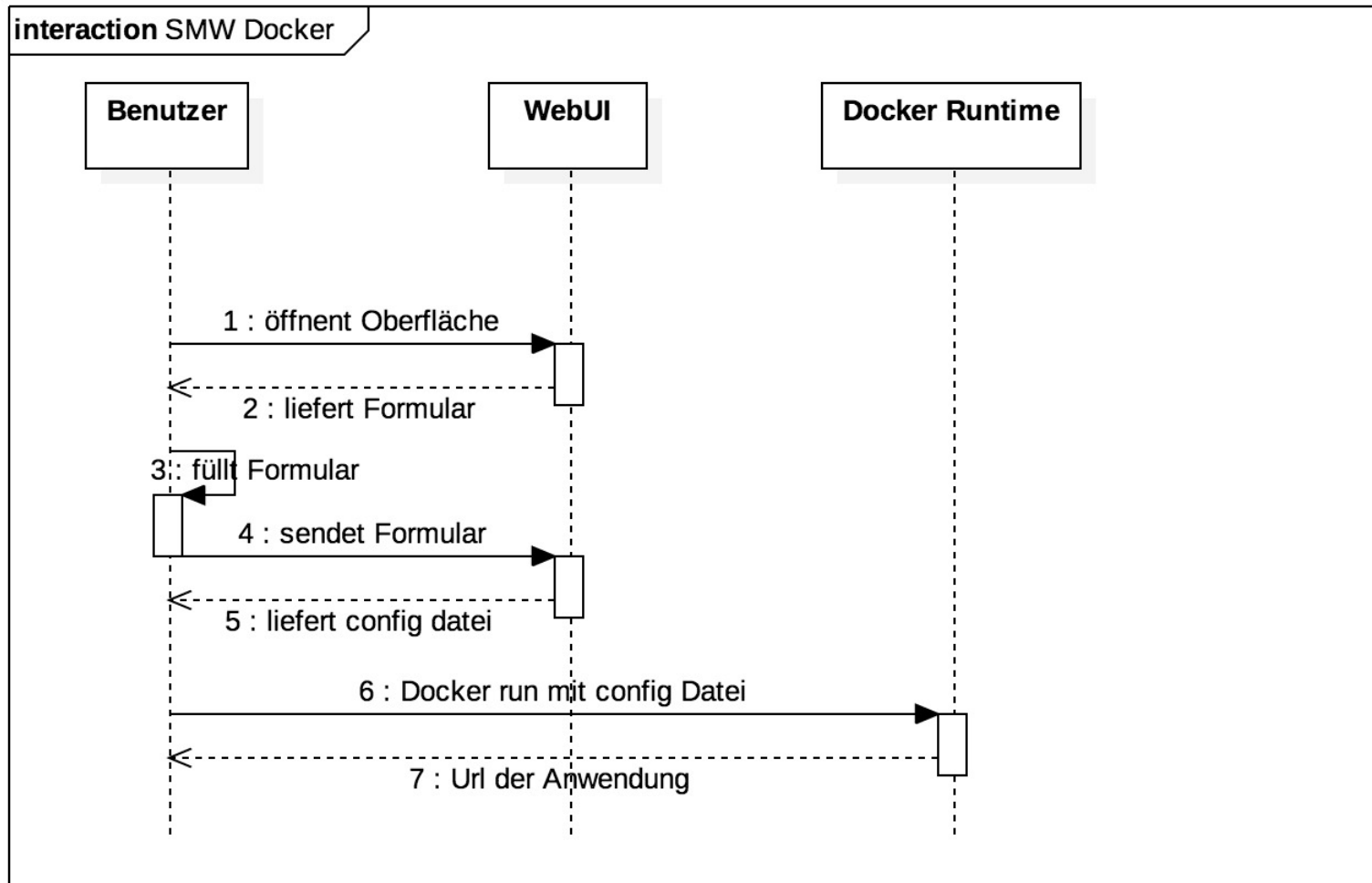
5  version: '3'
6  services:
7    mediawiki:
8      image: raphaelmanke/docker-semantic-mediawiki
9      restart: always
10     ports:
11       - ${HTTP_PORT:-8080}:80
12     links:
13       - database
14     volumes:
15       # After initial setup, download LocalSettings.php to the same directory as
16       # this yml and uncomment the following line and use compose to restart
17       # the mediawiki service
18       # - ./LocalSettings.php:/var/www/html/LocalSettings.php
19       - ./extensions.json:/var/www/html/extensions.json
20     depends_on:
21       - database
22  database:
23    image: mariadb
24    restart: always
25    ports:
26      - 3306:3306

```

Lösungsansatz schematisch

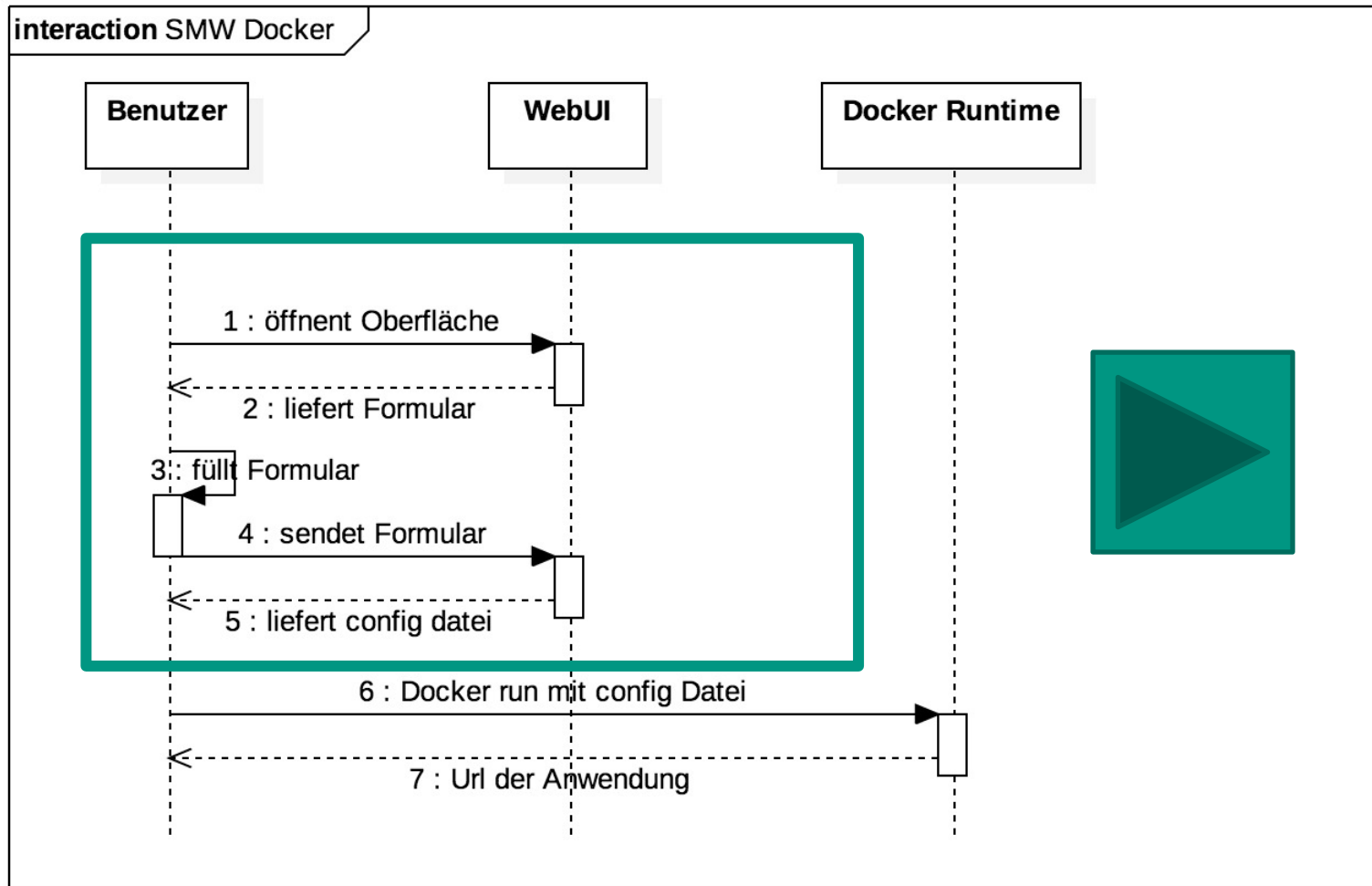


Lösungsansatz schematisch



Angular Grundlagen

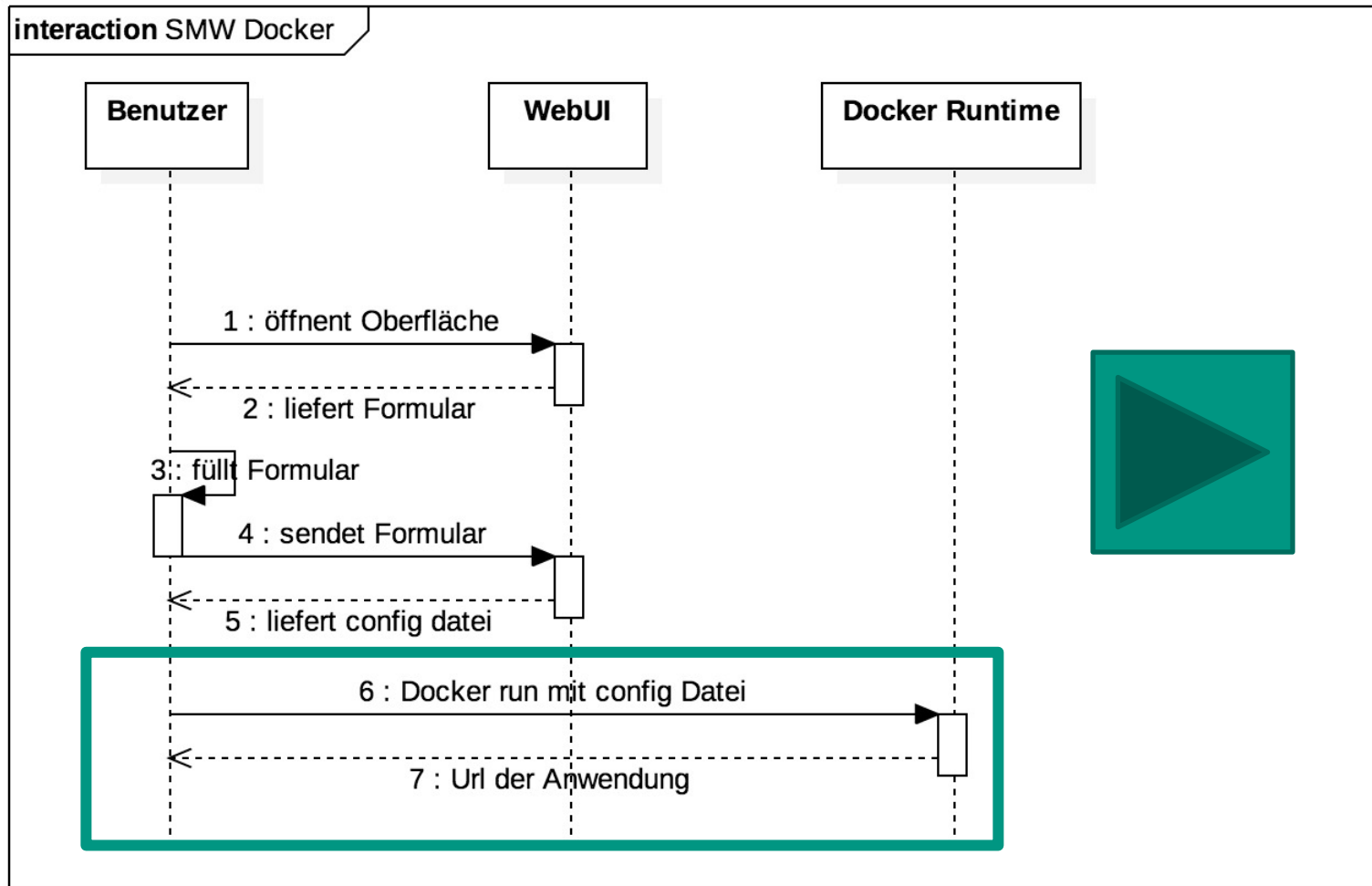
Weboberfläche Demo



Mediawiki + Extensions mit Docker

- Mediawiki als offizielle Image im Dockerhub
 - Wird von der Foundation/Community verwaltet
 - Verwendung als Basisimage
- Erweitern durch Composer und SemanticMediawiki
 - Installation des Composer
 - Installation SemanticMediawiki mithilfe des composer
 - Kopieren eines initialisierungs-Skript
 - Speichern als neues Docker Image
- Zur Laufzeit
 - Initialisieren der Datenbank
 - Installation der individuellen Extensions
 - Download + registrieren in LocalSettings.php
 - Composer optional

Docker Demo



Probleme

Installation und Auswahl von Erweiterungen

- Kein einheitlicher Marktplatz
 - Begrenzung auf offizielles MediaWiki Github-Repository
- Keine einheitliche Installationsmethode und nicht ohne weiteres ersichtlich, welche verwendet wird
 - Verwendung der „alten“ Installationsmethode

Probleme

Migration alter MediaWikis

Vorgehen

- Datenbank einspielen, Zugangsdaten in LocalSettings.php anpassen
- Bau einer minimalen LocalSettings.php

Erkenntnisse

- Liste mit installierten Extension, die Installationsmethode angibt ist unumgänglich
 - Änderungen an der LocalSettings.php müssen bekannt sein und für jeden Docker Container nachgeholt werden
 - Die referenzierten Repositories müssen zugänglich sein
- ➔ Migration möglich, aber mit Anpassungen verbunden

Weiterführende Arbeit

- Weiterentwicklung zu Produktivsystem mittels Docker Swarm, Kubernetes
 - Gute Skalierbarkeit
 - Ausfallsicherheit durch Redundanzen
- Einheitliche Verwaltung von Erweiterungen (und Skins)
 - Konfigurationsdateien automatisch mit der Auswahl generieren
- Mittelfristig einheitliche Installation mittels Composer
- Erstellen eines Marketplaces

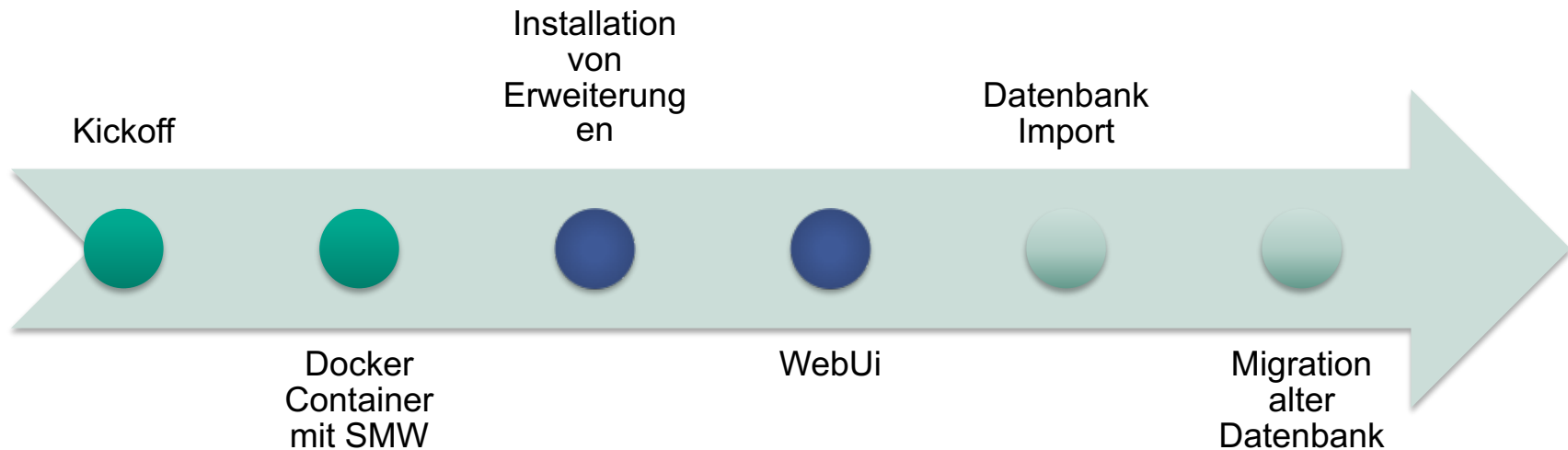


Vielen Dank für ihre Aufmerksamkeit

Fragen?

<https://github.com/RaphaelManke/docker-smw-all>

Roadmap



Aktuelle Probleme

- Welche MediaWiki Erweiterungen?
 - Installationsmethode nicht immer gleich
 - Kompatibilität mit Versionen
 - insbesondere beim Update
- Skins werden nicht in zentralem Repository verwaltet
 - Webcrawler? Händische Selektion (speichern in einer Datenbank)?
- Extensions zur build- oder runtime installieren

- Einfache und Intuitive Weboberfläche gestalten
 - Welche Extensions anzeigen? Wie sortiert?
 - Extension-index cachen? (Datenbank)

Was ist noch geplant?

- Node.js Container für UI ist noch nicht eingebunden
- Momentan muss
- Schwierig herauszufinden, welche Erweiterungen unterstützt werden können