



# Smart City Cockpit

Team Herr Oku

# Agenda

---

1

Ausgangslage und Anwendungskonzept

2

Architektur

3

Metriken

4

Live-Demonstration

5



Team

## Zentrale Herausforderungen und Ziele

### Anforderungen



# Zentrale Herausforderungen und Ziele



1

**Integration von Verkehrsdaten**  
aus vielfältigen Datenquellen

2

**Schnelle Reaktion** auf aktuelle Verkehrslage

3

**Show Case Betrieb** und Erweiterbarkeit

4

**Hohe Vernetzung** verschiedener Anwendungsbereiche

Zielsetzung des SCCs	
Übersicht	...auch bei komplexen Daten!
Dynamik	...in der Verkehrsverwaltung!
Konsistenz	...trotz verschiedener Datenquellen!
Skalierbarkeit	...über den Show Case hinaus!
Konfiguration	...des SCCs und der Simulation!

## Eckdaten

### 1 Projektdaten

**Zeitraum:** 4 Monate

**Team:** 6 Entwickler (3 FE, 3 BE)

**Vorgehensmodell:** Unified Process

### 2 Source Code

**Größe:** ca. 21.000 LOC

**Tests:**

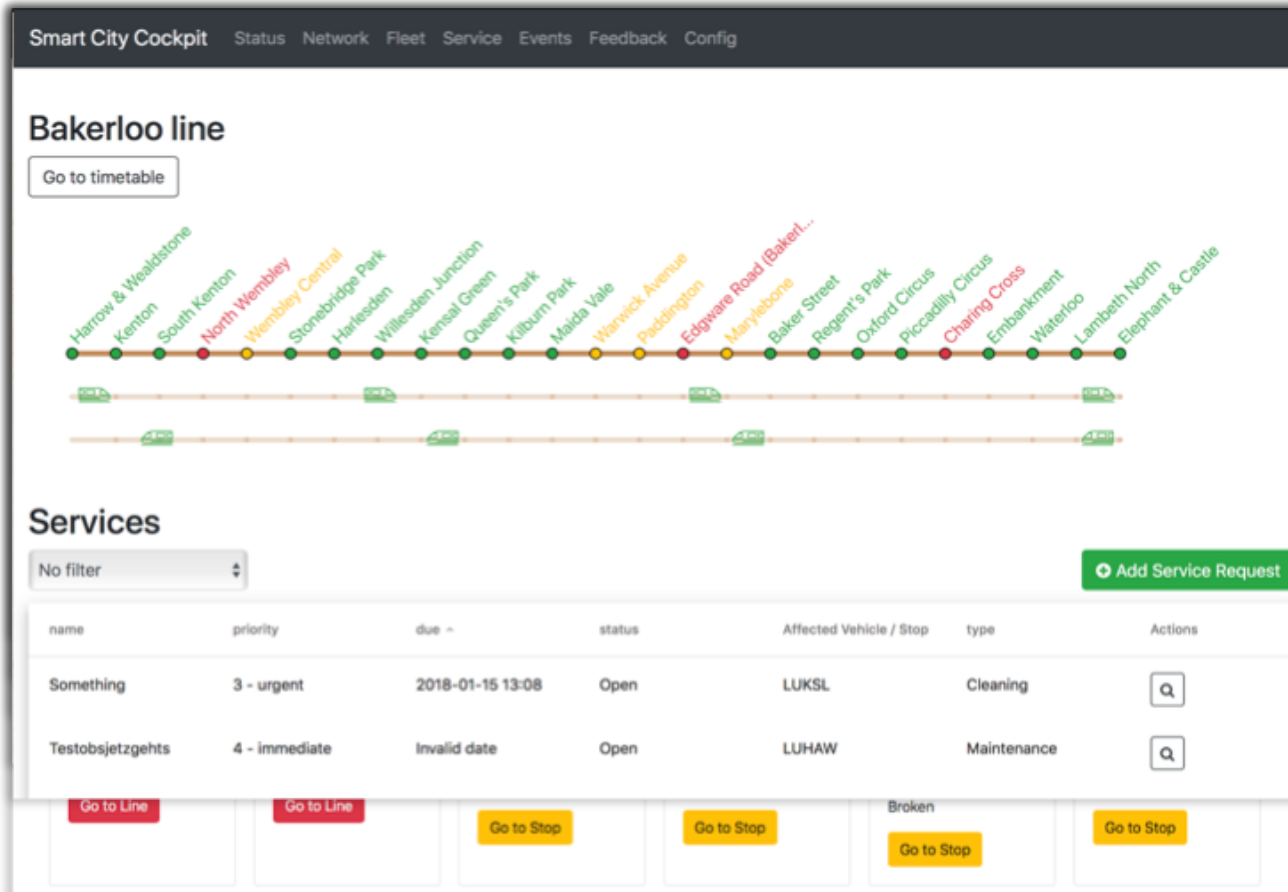
- Klassen-Abdeckung: 95%
- Methoden-Abdeckung: 84%
- Zeilen-Abdeckung: 81%

### 3 Dokumente

- Use Cases
- Glossar
- Domain Model
- Kontrakte & SSD
- Komponentendiagramm
- Designklassendiagramm

## Überblick der Zielerreichung

### SCC



### Highlights (Auszug)

#### Status

- Schneller **Gesamtüberblick**
- **Quick Links** direkt zum Problemherd

#### Interaktive Netzwerkvisualisierung

- **Geographisch korrekte** Stationsorte
- Support für verschiedene **Netzwerkoverlays**

#### Linienvisualisierung

- **Live** Statusanzeige und Positionsangabe

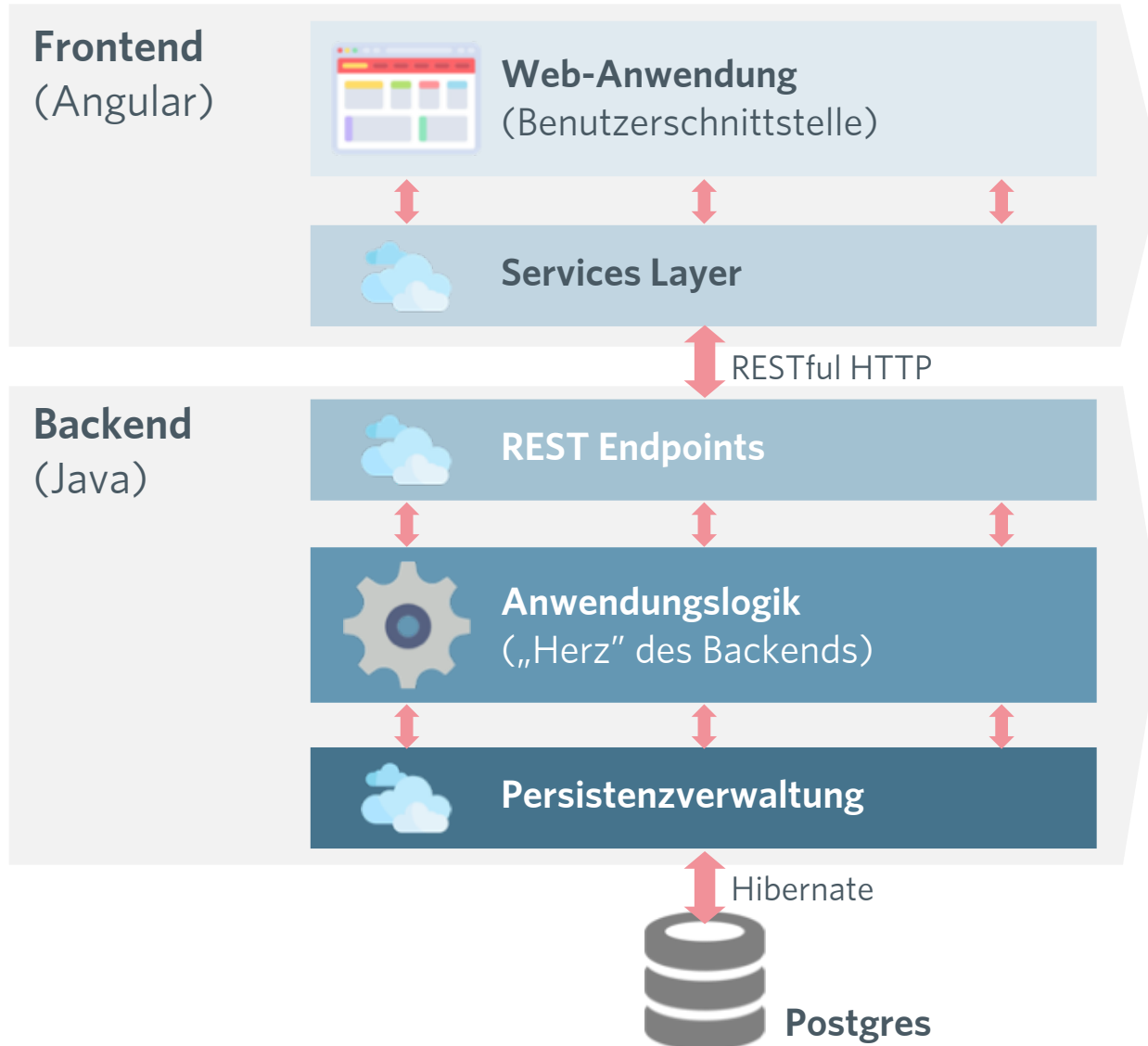
#### Shift Plan

- **Interaktive** Tourenplanung

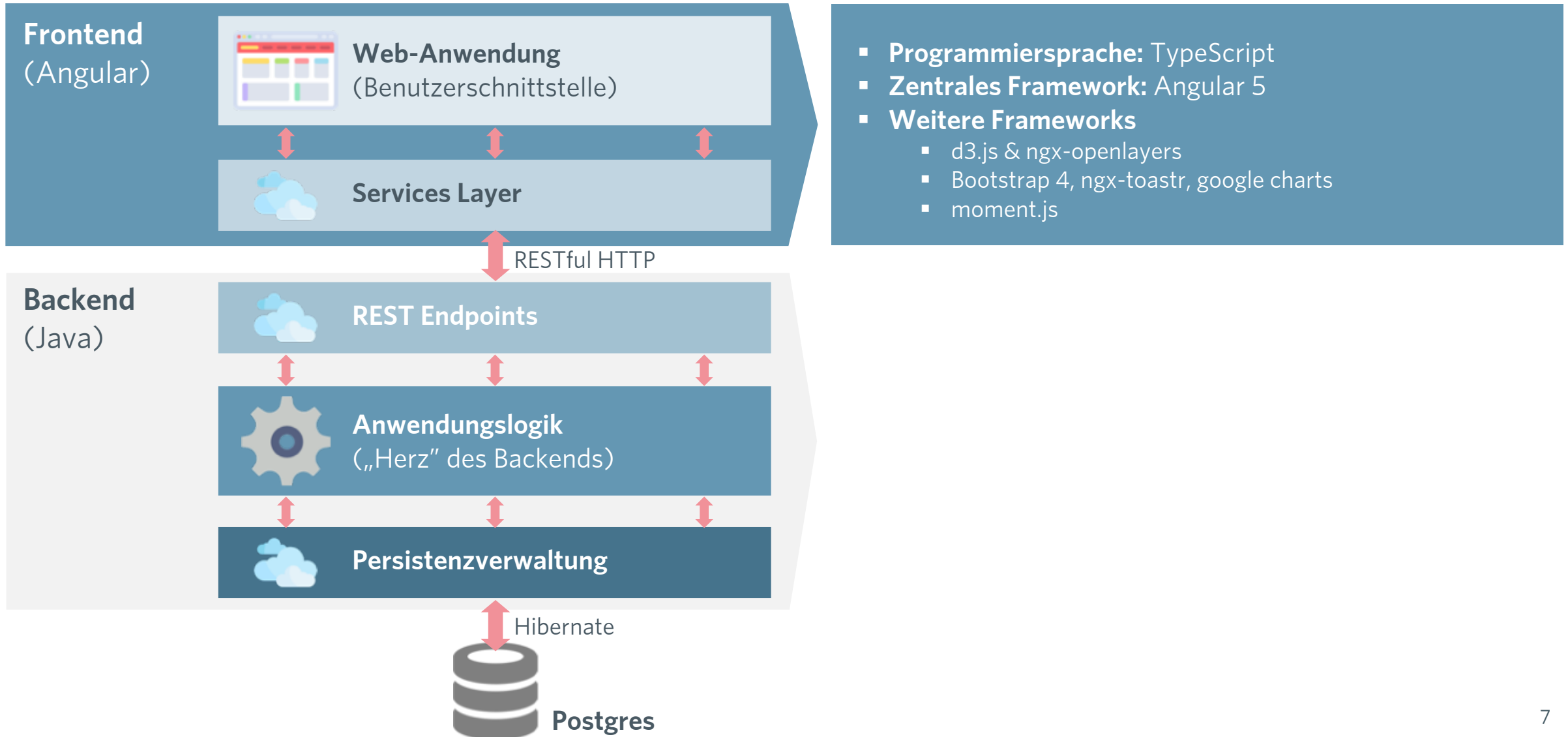
#### Customization

- Umfangreiche **Live Engine Konfiguration**

# Stabile Full-Stack Systemarchitektur

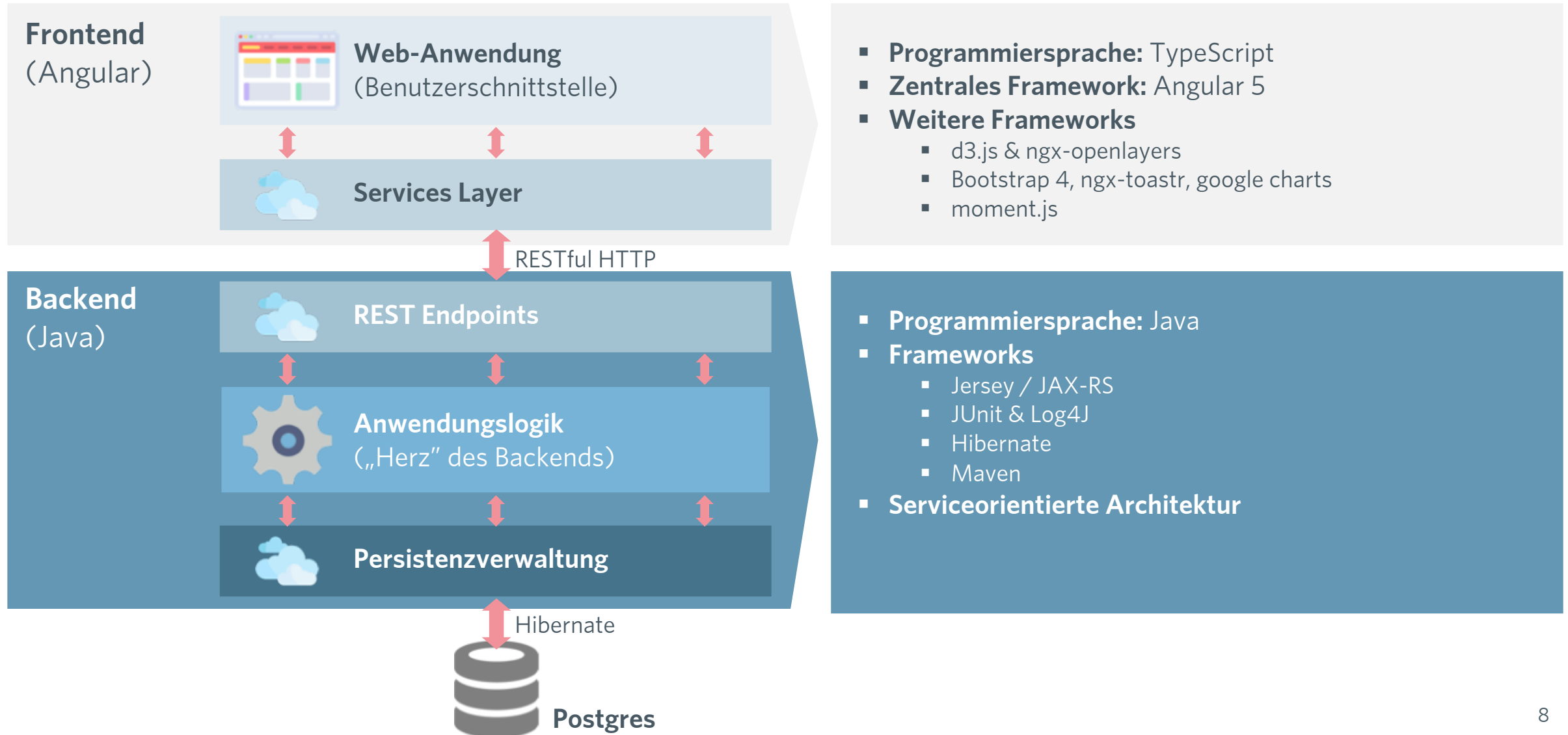


# Moderne Frontend-Technologien

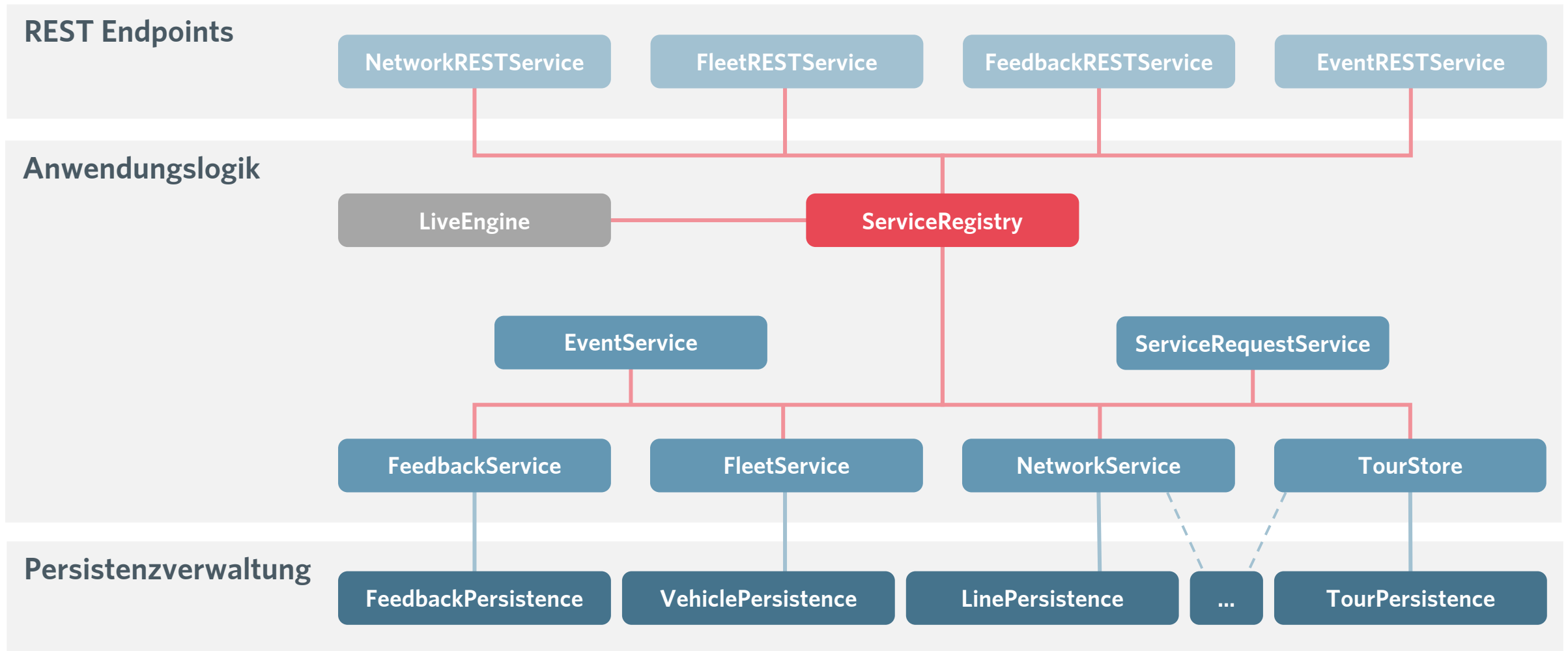




# Konfigurierbares Backend-Konzept



# Skalierbare Serviceorientierte Architektur



# Codeauszug: Service Registry

### ServiceRegistry

**Initialisierung:**

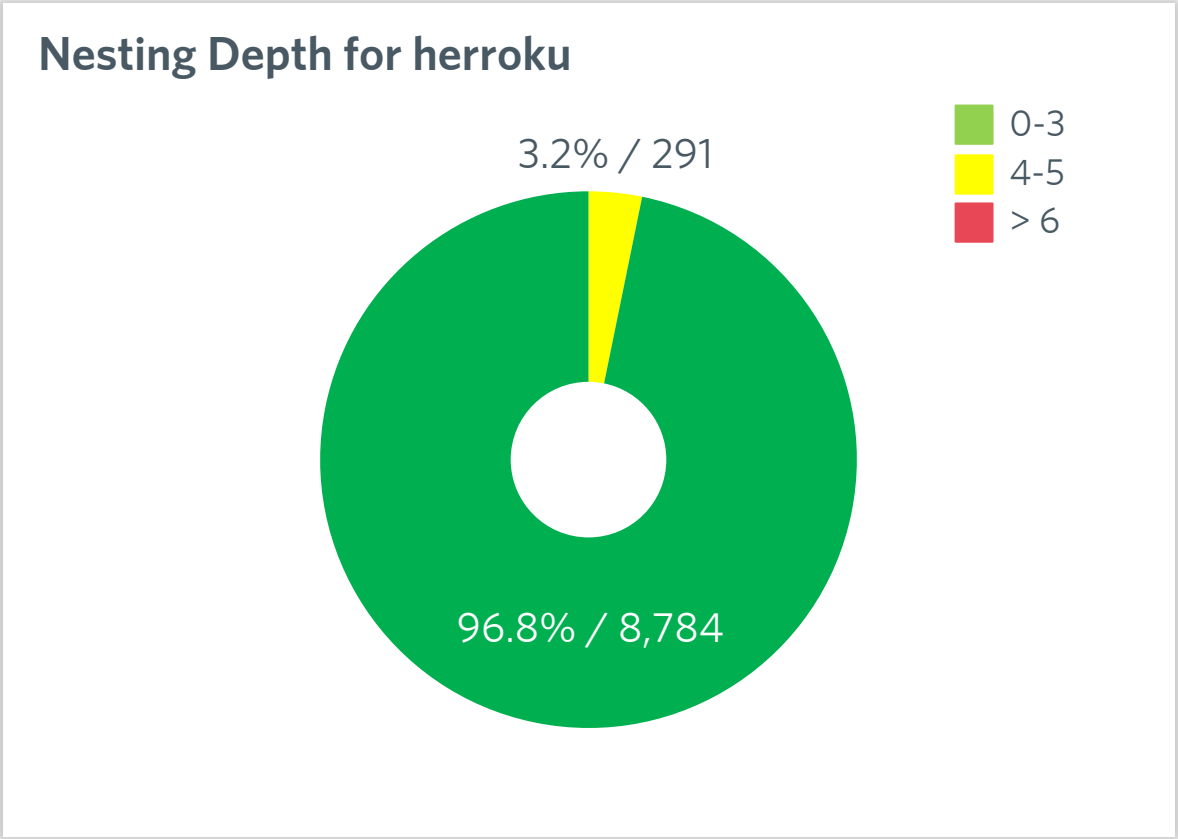
```
registry = new LinkedHashMap<Class, ParentService>();

registry.put(Network.class, new NetworkImpl());
registry.put(FleetService.class, new FleetServiceImpl());
registry.put(LiveEngine.class, new LiveEngineImpl());
registry.put(TourStore.class, new TourStoreImpl());
registry.put(FeedbackService.class, new FeedbackServiceImpl());
registry.put(EventService.class, new EventServiceImpl());
registry.put(ServiceRequestService.class, new ServiceRequestServiceImpl());

for (ParentService service : registry.values()) {
    service.initialize();
    logger.info("Service " + service.getName() + " is initialized!");
}
```

**Aufruf:** FleetService fleet = ServiceRegistry.getService(FleetService.class);

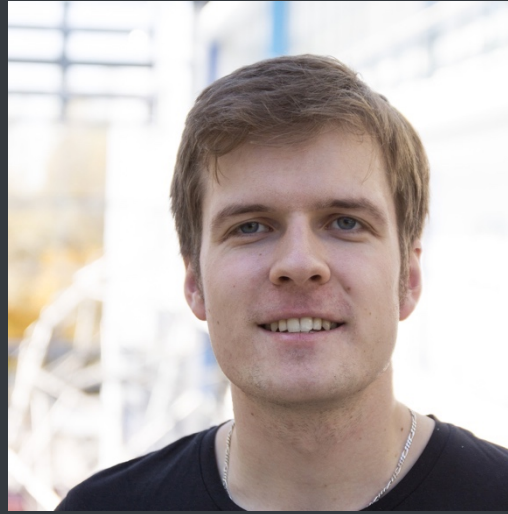
# Metriken: Geschäftslogik (MCCC) & Nesting Depth







Franzi Geiger



Marcel Henrich



Martin Keppner



Jonas Müller



Malte Sandstedt



Felix Schwarzmeier

---

# Quellen

Icons Tech Stack: Icon made by [Freepik](#) and Smashicons from [www.flaticon.com](http://www.flaticon.com)