



# CoronaCheckIn

## Abschlusspräsentation

Bilal Alnaani

Daniel Depta

Tom Käßpler



# Gliederung

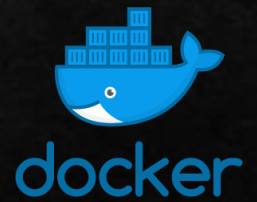
1. Einführung
2. Technologien
3. Architektur
4. Nx + Docker + Github
5. Grafana + Prometheus + NestJS
6. Lessons Learned
7. Demo

# 1. Einführung

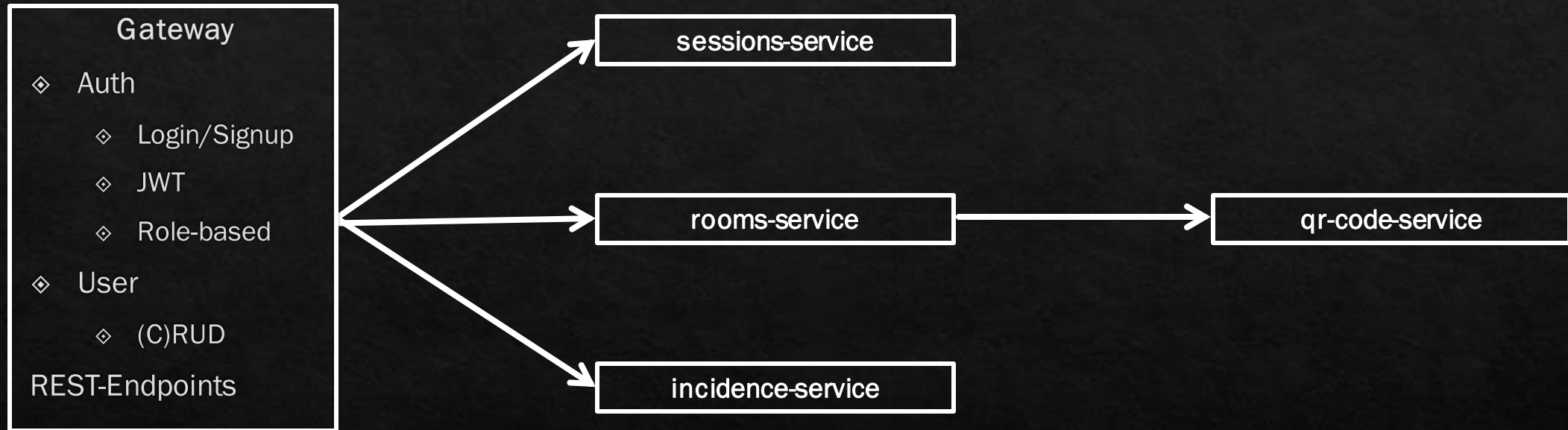
- ◆ Authentifizierung + Autorisierung
- ◆ Frontoffice
- ◆ Backoffice – Stammdatenverwaltung (Räume, Profile, Session)
- ◆ einchecken und auschecken aus Räumen via QR-Code
- ◆ QR-Code Generierung

## 2. Technologien

- ◊ Nest
  - ◊ TypeScript
  - ◊ Gateway
  - ◊ Microservices
- ◊ Redis
- ◊ Nx
- ◊ Docker
- ◊ PostgreSQL
- ◊ Prometheus
- ◊ Grafana

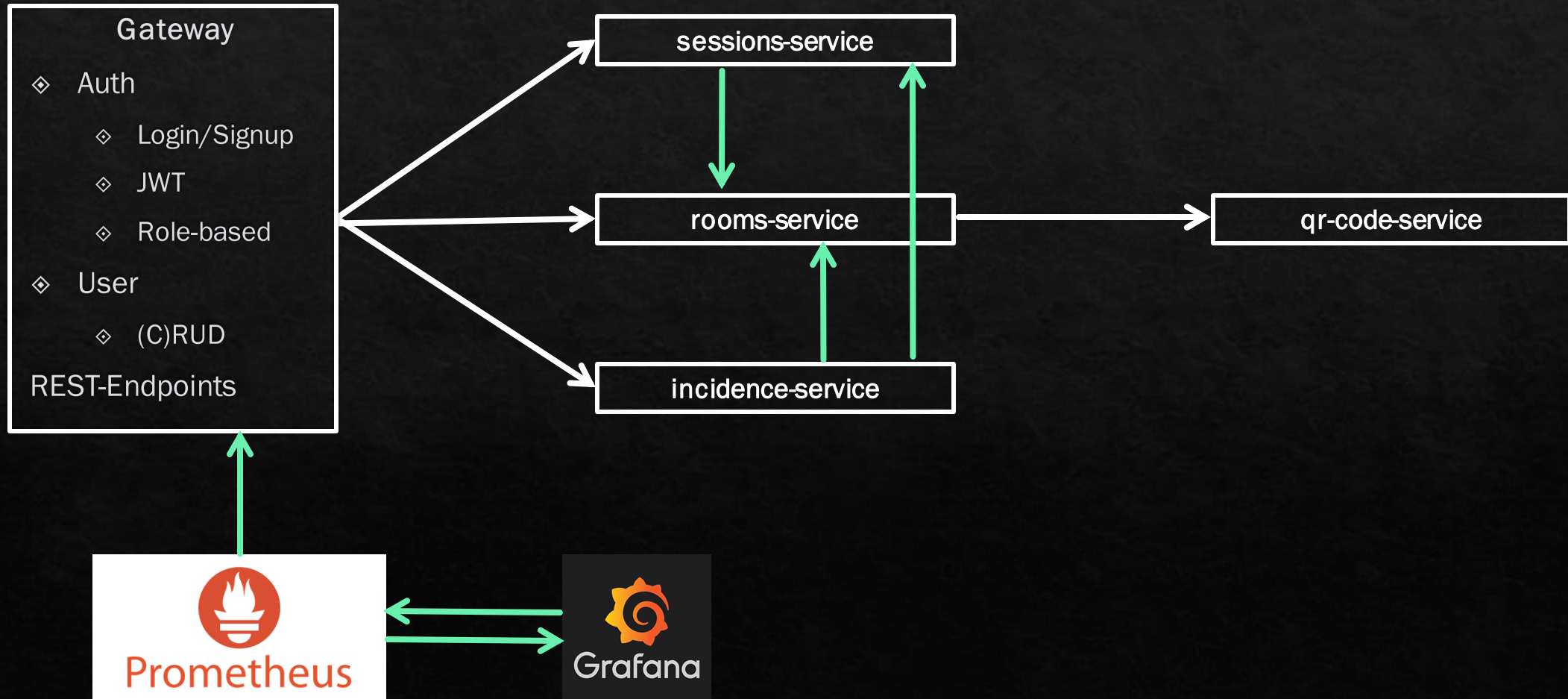


## 3.1 Architektur (alt)



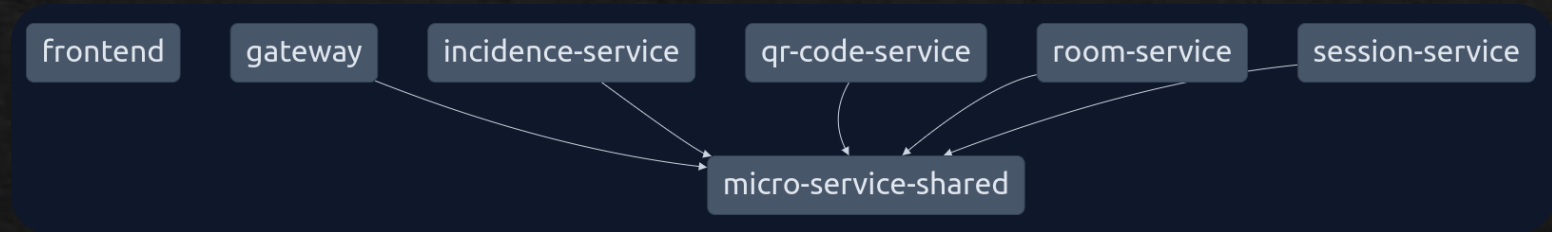


## 3.2 Architektur (neu)



## 4. Nx + Docker + Github = ❤️

- ◆ Nx Graph/Nx Affected

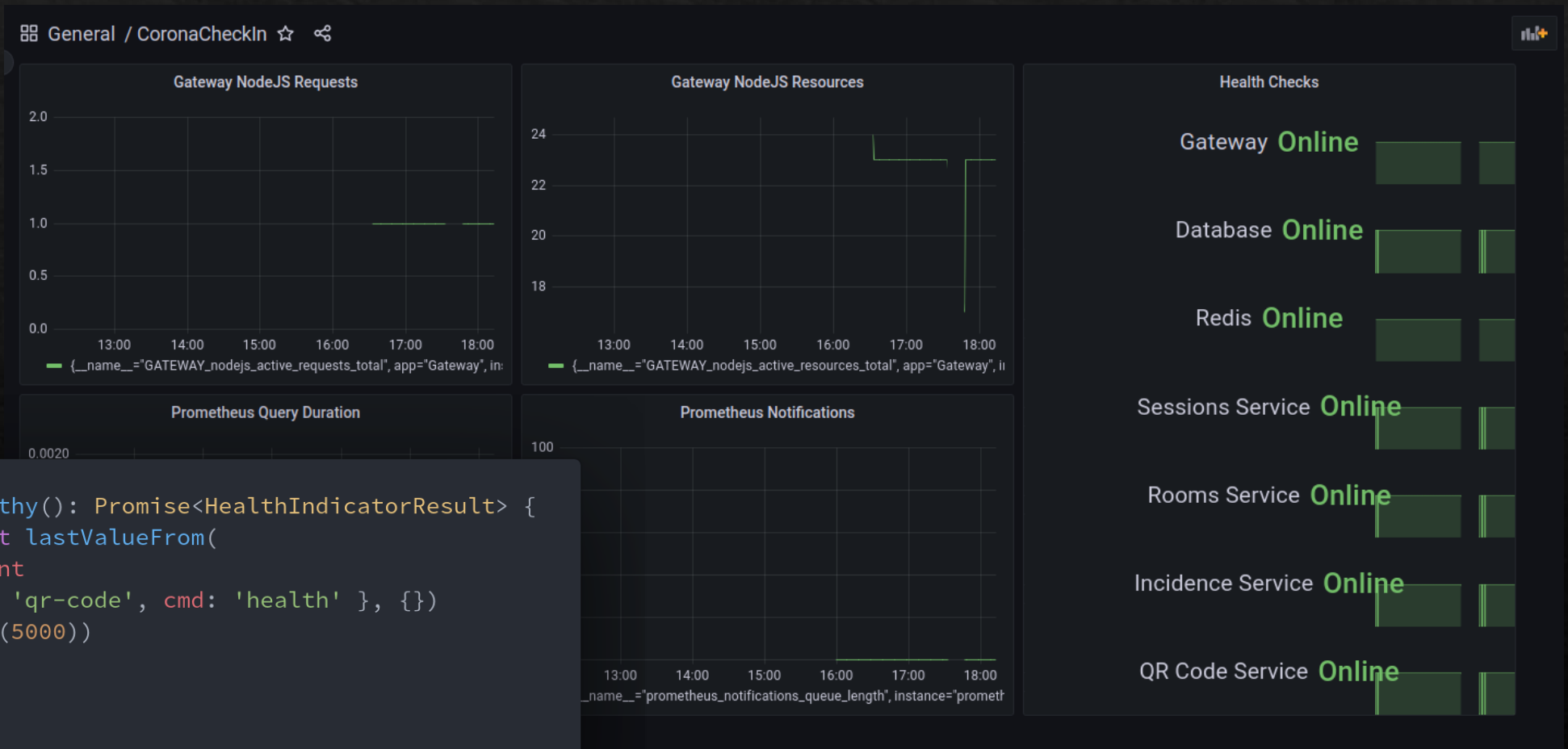


- ◆ Produktions-Umgebung per Docker Compose
- ◆ Nx Task System -> Wegkapseln von Docker Commands
  - ◆ 13 Commands hinter einem npm run

```
1  "build_docker": {  
2    "executor": "nx:run-commands",  
3    "options": {  
4      "commands": [  
5        { "command": "nx build frontend" },  
6        { "command": "docker build -f ./docker/frontend.dockerfile . -t ccn-frontend" }  
7      ],  
8    }  
9  }
```

```
1  "scripts": {  
2    "docker:prod": "docker compose -f docker-compose.prod.yml up",  
3    "docker:prod:build": "nx run-many --target=build_docker --all=true --parallel=100"  
4  }
```

# 5. Grafana + Prometheus + NestJS





# 6. Lessons Learned

- ◆ User in eigenen Microservice auslagern
  - ◆ Workaround: User-Entity ist Shared -> Zugriff in anderen Services per Repository
- ◆ Relationen nicht vergessen
- ◆ Nx kann manuelle Arbeit verringern
- ◆ 2-Wochen Sprints durch WE2 (Review, Retro und Planung)

## 7. Demo



POSTMAN

CoronaCheckIn