Sicherheit in Microservice-Umgebungen

Tobias Tefke und Ralf C. Staudemeyer

Hochschule Schmalkalden

30. DFN-Konferenz "Sicherheit in vernetzten Systemen"
8.–10. Februar 2023

Inhalt

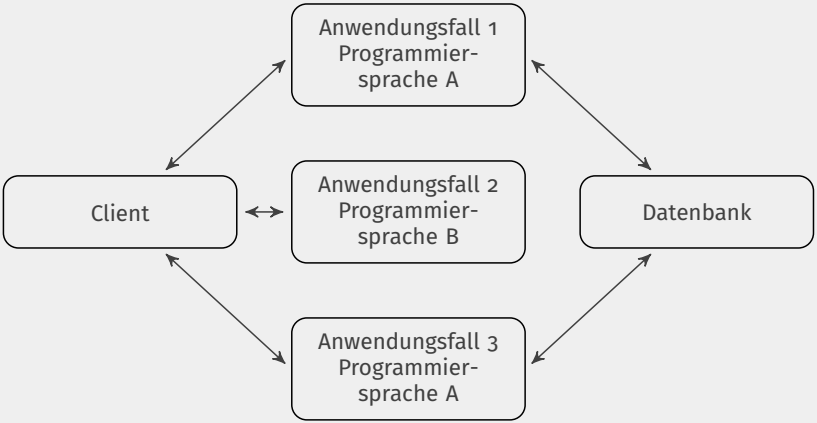
Microservices

Monolithische Software

Schematische Darstellung einer monolithischen Softwarearchitektur

Prinzipien von Microservices

- Begrenzter Kontext
- Geringer Codeumfang (Größe)
- Unabhängigkeit



Schematische Darstellung einer Microservice-Softwarearchitektur

Kommunikation mit und zwischen Microservices

- Intern vs. extern
- Synchron vs. asynchron

Oft Containerization, da

- Automatisierung
- Unabhängigkeit
- Portabilität
- Leichtgewichtigkeit
- Isolation (Sicherheit)

Nachteile von Microservices

- Zugriffsschutz
- Abhängigkeiten zwischen Microservices
- Ausfallkaskaden $\rightarrow$ Circuit Breaker

Sicherheit in Bezug auf Microservices

Authentifikation

- Durch HTTP ermöglichte Verfahren (z.B. Basic, Bearer)
- JSON Web Token (JWT)
- Sidecars $\rightarrow$ zentrale Implementierung

Isolation von Containern

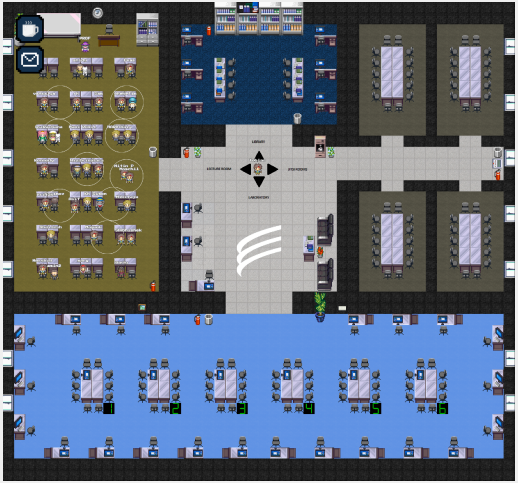
- Vom Host und untereinander
- Nutzung von Namespaces
- AppArmor und SELinux
- Netzwerkisolation

Gewährleistung der Vertraulichkeit

- Externe Schnittstellen: TLS
- Interne Schnittstellen: MTLS

Microservices am Beispiel einer Lehrumgebung

Bewegen von Charakteren über eine Karte



Karte unseres virtuellen Internet-der-Dinge-Labors

Gruppenkommunikation

Gruppenkommunikation in WorkAdventure

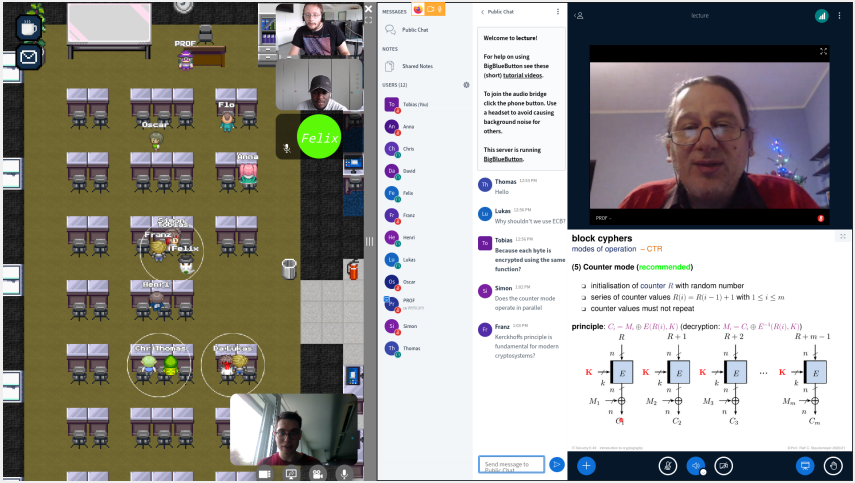
Aktionen

Aktionen werden nach Bestätigung durch den Nutzer ausgeführt

Aktionen (Forts.)

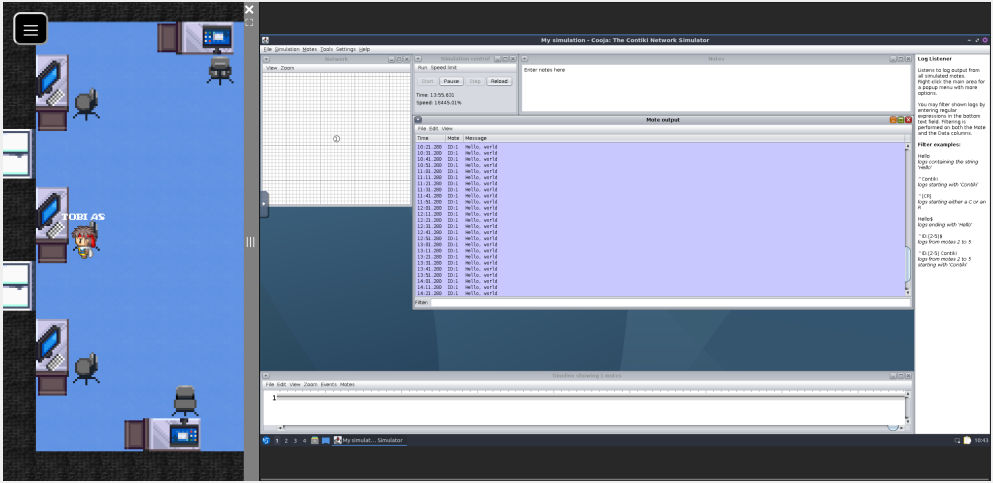
Aktionen sind erweiterbar

Integration von BigBlueButton für Online-Vorlesungen



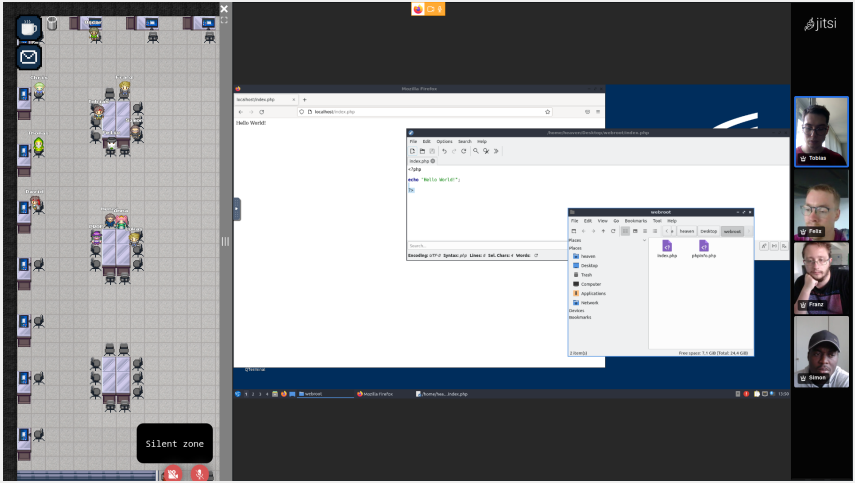
Studierende können während der Vorlesung miteinander kommunizieren

Arbeiten an virtuellen Rechnern



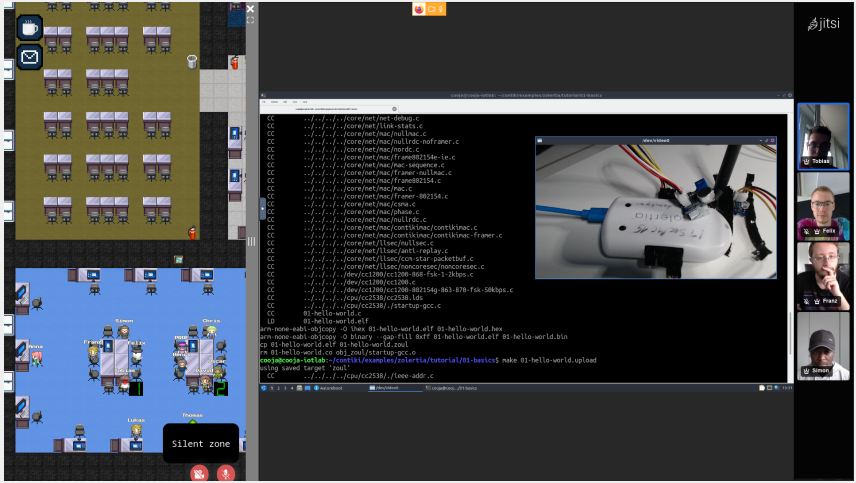
Ausführung eines Programms auf einem virtuellen Rechner

PC-Pool



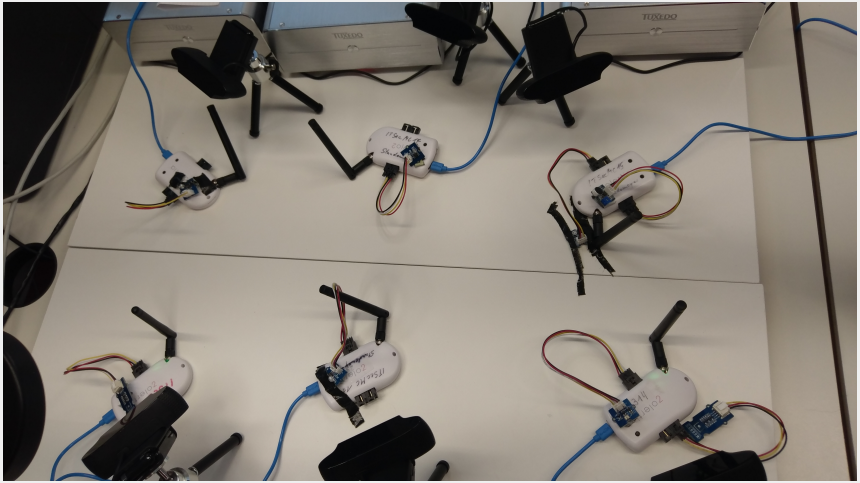
Gemeinsames Arbeiten im PC-Pool

Dezentrales Arbeiten an physischen Geräten



Gemeinsames dezentrales Arbeiten an Laborhardware

Laborhardware

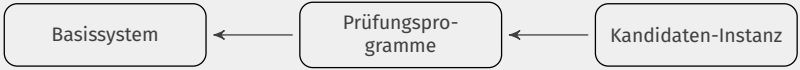


RE-Motes im Labor, auf welche online zugegriffen werden kann

Prüfungsmodus

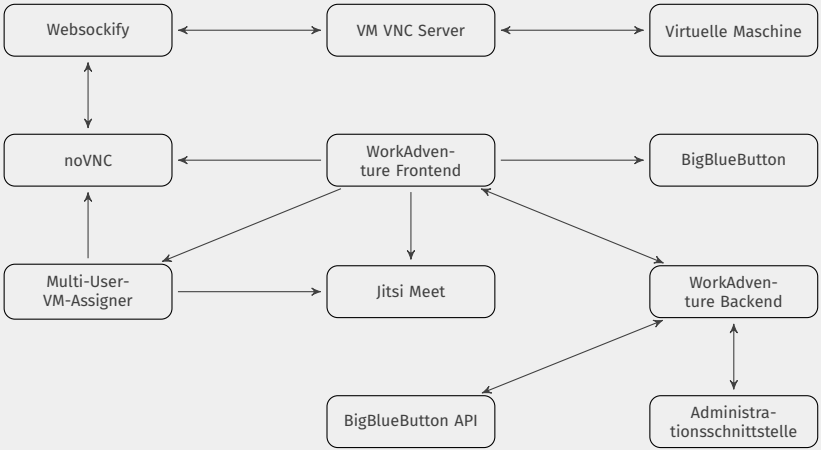
- Jeder Kandidat erhält einen Rechner, um Lösungen zu erarbeiten
- Benötigte Entwicklungsumgebung ist vorinstalliert
- Nach Ablauf der Zeit werden die Lösungen auf einen Server hochgeladen
- Dort können diese vom Korrektor eingesehen werden

Prüfungsmodus (Forts.)



Struktur eines virtuellen Rechners im Prüfungsmodus

Struktureller Aufbau der Lehrumgebung



Schematische Darstellung der Labor Basisarchitektur

Sicherheitsanalyse der Lehrumgebung

Sicherheitsanalyse

- Spezielle HTTP-Anfragen, die an Microservices geschickt werden
- Mittels HTTP-Client (curl) gefunden
- Standard-Administrationstoken in der Community-Version

Spezielle HTTP-Anfragen

Mehrere Lücken, z.B.:

- Überlastungsmeldung zeigen
- Zugang zu Debug-Informationen
- DoS durch inkorrekte Fehlerbehandlung

Beispiel: Abrufen von Debug-Informationen

Request:

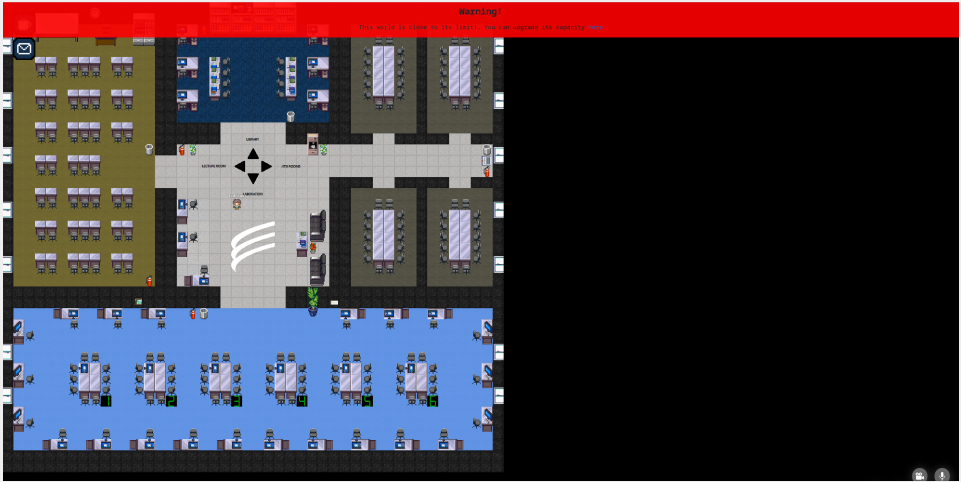
```
curl -i -k -X GET -H "Content-Type:  application/json"
https://your.instance/pusher/dump?token=123
```

Response:

```
HTTP/2 200
content-type:  application/json
uwebsockets:  18
content-length:  16327
date:  Fri, 22 Jul 2022 21:46:36 GMT

// Debug-Informationen
```

Beispiel: Überlastungsmeldung zeigen



Überlastungsmeldung in fast leerem Raum

Fazit der Sicherheitsanalyse

- Offene Standard- und Debugzugänge deaktivieren
- Authentifizierung von Nutzern immer erzwingen
- Sicherstellen, dass keine privaten Schnittstellen öffentlich angeboten werden
- Alle Eingaben prüfen

Kontakt

- SUASecLab
- t.tefke@stud.fh-sm.de
- r.staudemeyer@hs-sm.de