

OI4-SERVICE- REGISTRY

Gruppe 4





GLIEDERUNG

- ▶ Team-Vorstellung
- ▶ Produktübersicht
- ▶ Master Use-Case
- ▶ Architekturübersicht und Module
- ▶ Testen
- ▶ Live Vorstellung
- ▶ Fazit



Projektleiter & Produktmanager

Goran Erdeljan
(inf19170@lehre.dhbw-
stuttgart.de)



Technischer Redakteur

Daniel Baumann
(inf19188@lehre.dhbw-
stuttgart.de)



Systemarchitekt / 1. Entwickler

Benedict Wetzel
(inf19052@lehre.dhbw-
stuttgart.de)



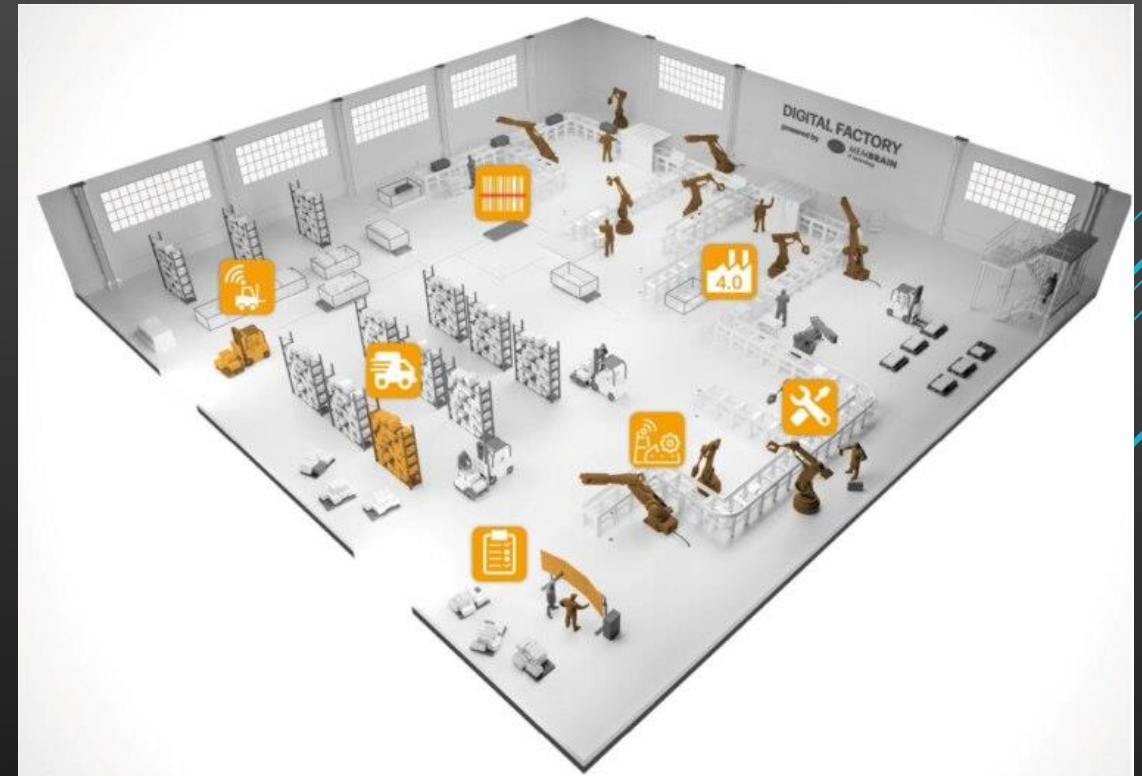
Testmanager / 2. Entwickler

Serdar Ilhan
(inf19223@lehre.dhbw-
stuttgart.de)

TEAM-VORSTELLUNG

PRODUKTÜBERSICHT

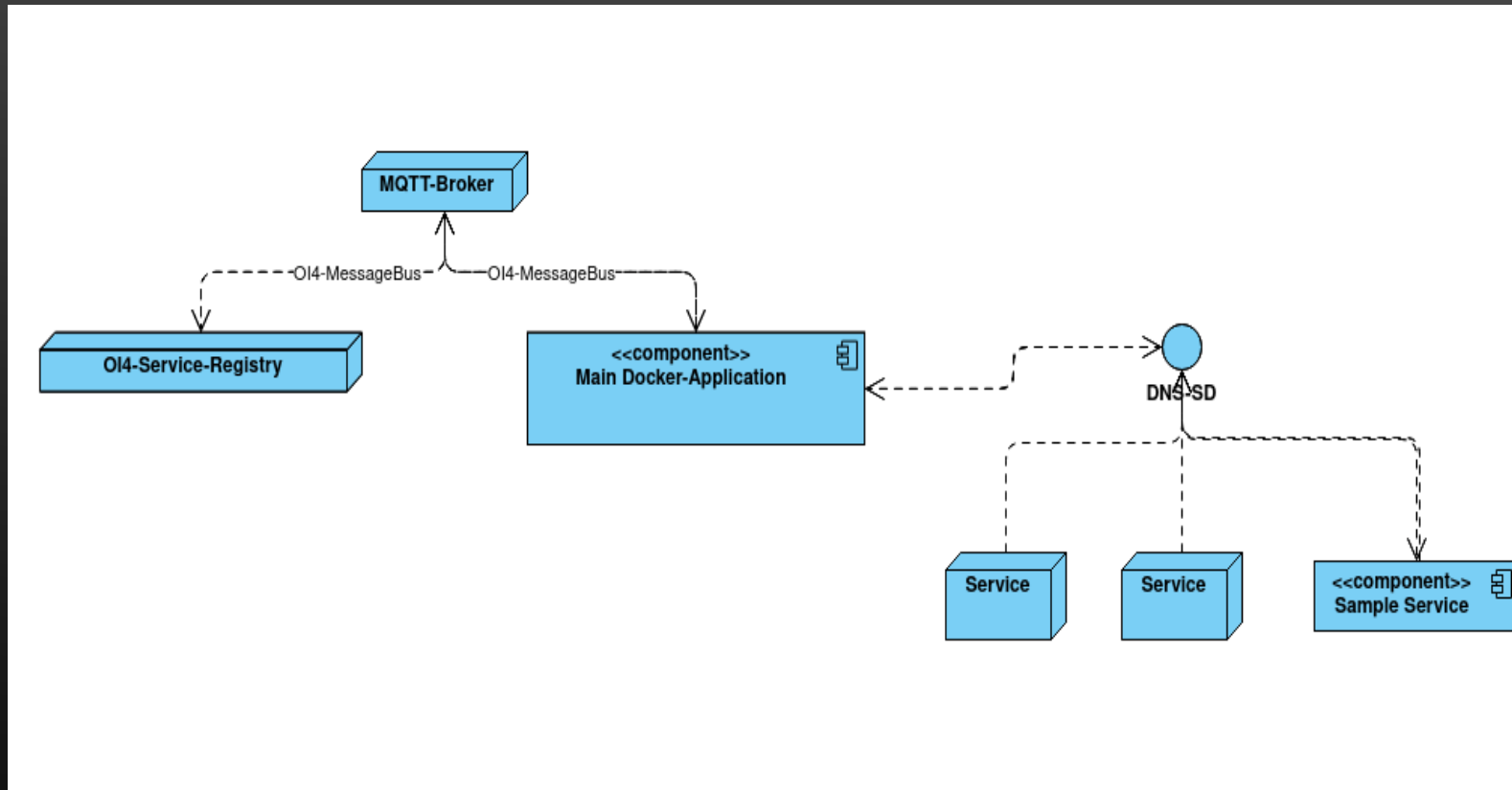
- ▶ Erweiterung der OI4-Service Registry
- ▶ Registrierung von Geräten und Diensten



MASTER USE-CASE

- ▶ Hinzufügen der *Service Discovery-Funktionen* zur OI4-Registry
- ▶ Registrierung von Geräten via DNS-SD
- ▶ Übernahme der registrierten Dienste (aus dem OI4-System) und deren Übermittlung an das Netzwerk über den DNS-SD-Mechanismus
- ▶ Ausführung der Anwendung in Docker Container

ARCHITEKTURÜBERSICHT UND MODULE



MOD10: Service-Registry Schnittstelle

- ▶ Importiert und startet Submodule
- ▶ Konfiguriert Submodule auf Basis der Umgebungsvariablen

Submodule von MOD10:

MOD11: DNS-SD-Listener

- ▶ Sendet mDNS-Queries um Dienste zu finden
- ▶ Informationen der Dienste über MQTT-Broker veröffentlicht --> an der OI4 Registry registriert

MOD12: Service-Registry-Listener

- ▶ Wartet darauf, dass Dienste ihr Master-Asset-Model auf dem MQTT-Broker veröffentlichen
- ▶ Beantwortet mDNS-Queries mit den Daten dieser Dienste sodass diese mit DNS-SD gefunden werden können

MOD13: Konformitätsvalidator

- ▶ Überprüft Daten auf Konformität mit den Spezifikationen der OI4
- ▶ Baut Master-Asset-Model aus TXT-Records

MOD20: Testanwendung

- ▶ Startet Test-Anwendung
- ▶ Konfiguriert Umgebungsvariablen

Submodule von MOD20:

MOD21: Web-Interface

- ▶ Steuert Backend mit Express
- ▶ Arbeitet HTTP-Anfragen für das Web-Interface ab

MOD22: Announce-Service

- ▶ DNS-SD-Mechanismus wird verwendet
- ▶ Wartet auf mDNS-Queries
- ▶ Antwort von der Testanwendung

MOD23: Registrieren bei OI4

- ▶ Veröffentlicht Dienste über MQTT

MOD24: DNS-SD-Listener

- ▶ DNS-SD-Mechanismus wird verwendet
- ▶ Sendet mDNS-Queries An alle dienste im Netzwerk
- ▶ Die betroffenen Dienste antworten auf die Anfrage
- ▶ Listet diese im Webinterface auf

TESTEN

Was muss getestet werden und warum?

- ▶ Dienste werden gesucht über DNS-SD → mDNS Queries
- ▶ Wenn Suche erfolgreich → Dienste antworten mit Dienst-Infos
- ▶ Falls Antwort mit Spezifikationen übereinstimmt
- ▶ Weitere Kommunikation über MQTT-Broker
- ▶ Veröffentlichen auf der OI4-Service-Registry

Diese Funktionen müssen richtig ausgeführt werden, um den Anforderungen des Projekts zu erfüllen !!!

Basierend auf wichtige Funktionalitäten:

- ▶ Suchen und Finden von Services mit der Hauptanwendung:
 - ▶ Die Registrierung der gefundenen Dienste in der OI4-Registry ("oi4=true"-TXT-Records)
 - ▶ Keine Registrierung der gefundenen Dienste in der OI4-Registry(ohne "oi4=true"-TXT-Records)
 - ▶ Ob ein Dienst am OI4-Messagebus über DNS-SD gefunden wird

LIVE-VORSTELLUNG

FAZIT

- ▶ Ständig neue Herausforderungen
- ▶ Unerwartete Änderungen
- ▶ Ungenaue Anforderungen

- ▶ **Aber:** Am Ende Projekt erfolgreich abgeschlossen
- ▶ Aktuelles Thema (Industrie 4.0)
- ▶ Gute Teamarbeit trotz Verlust von Teammitgliedern

DANKE FÜRS ZUHÖREN!

GIBT ES NOCH FRAGEN?