

Software Requirements Specification

(TINF19C, SWE I Praxisprojekt 2020/2021)

Project: **Service Registry**

Customer: *Rentschler & Holder*

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team x (Name1, Name2, Name3, Name4, Name5)

Rotebühlplatz 41

70178 Stuttgart

[illegible]

1. Introduction

The goal of the project is to add service discovery functionalities to the existing Oi4-Service-Registry, developed by the OI4-Alliance. To be added features are the registration of devices, which are announced via DNS-SD but also to take the services, which are already registered at the OI4-Service-Registry and announce them to the network using the DNS-SD mechanism. These features shall be implemented in an application running in a Docker-Container. The project shall also contain a Docker-Application for testing the functionalities of the system

2. Use Cases

2.1 UA-001 Registering Services, that are found using DNS-SD

Related Business Process:	-
Use Cases Objective:	Registering all available services at the OI4-Service-Registry
System Boundary:	OI4-Service-Registry, Services, which announce themselves using DNS-SD, Docker-Application
Precondition:	The to be registered services must use TXT-records conform with the specification published by the OI4-Alliance
Postcondition on success:	The services will be registered at the OI4-Service-Registry
Users:	-
Triggering Event:	Any new DNS-SD entries on the network.

2.2 UA-002 Announcing registered Services on the network

Related Business Process:	-
Use Cases Objective:	Announcing registered Services on the network
System Boundary:	Ol4-Service-Registry, Services registered at the Ol4-Service-Registry, Docker-Application
Precondition:	The services have to be registered at the Ol4-Service-Registry
Postcondition on success:	The services will be announced to the network using the DNS-SD mechanism.
Users:	-
Triggering Event:	Any new service registered at the Ol4-Service-Registry

3. Non-Functional Requirements

3.1 /NF10/ User Documentation

The Project should contain extensive Documentation for all parts of it. Using this Documentation a user should be able to install and use all components and features, the application provides. The documentation shall be distributed using a PDF file contained in the GitHub-Repository.

3.2 /NF20/ Easy Deployment

An easy deployment shall be targeted, as the system is based on Docker-Containers. The Docker-Images needed to start the Docker Container from shall either be built from the source-code that is contained in the GitHub-Repository or be downloadable from the 'docker-hub'.

4. Functional Requirements

4.1 /F10/ Running in Docker

The Application has to be able to run in the context of a Docker Container. The instructions to build and configure such a container shall be included in the User Documentation.

4.2 /F20/ Listening to DNS-SD Entries

The Main Docker Application should listen to any upcoming DNS-SD Services. It should then decide, whether the Entries are from Services, which need to be published to the OI4-Service Registry.

4.3 /F30/ Registering Services at the OI4-Registry

The Data that is collected from the DNS-SD entries should be taken and published on the OI4-MessageBus. Published Messages have to fulfill the Specifications put up by the OI4-Alliance.

4.4 /F40/ Exemplary Docker Test-Applications

The Project should contain simple Docker Applications, which use and test the basic functionalities of the system.

4.5 /F50/ Listening to Services registered at the OI4-Service-Registry

The Main Docker application shall listen to any changes in the Services registered at the OI4-Service-Registry. When there are new Services registered it shall announce them to the network via DNS-SD.

5. References
