Software Requirements Specification

(TINF19C, SWE I Practice Project 2020/2021)

Project: Service Registry

Customer: Rentschler & Holder
Rotebühlplatz 41
70178 Stuttgart

Supplier: Team 4 (Daniel Baumann, ~~Tim Diehl~~, Goran Erdeljan, Serdar Ilhan, Benedict Wetzel)

Version	Date	Author	Comment
0.1	07.09.2020		created
1.0	21.03.2021	Daniel Baumann	Überarbeitet
1.1	09.04.2021	Daniel Baumann	Überarbeitet
1.2	27.04.2021	Daniel Baumann	Überarbeitet
1.3	30.04.2021	Daniel Baumann	Überarbeitet

Table of Contents

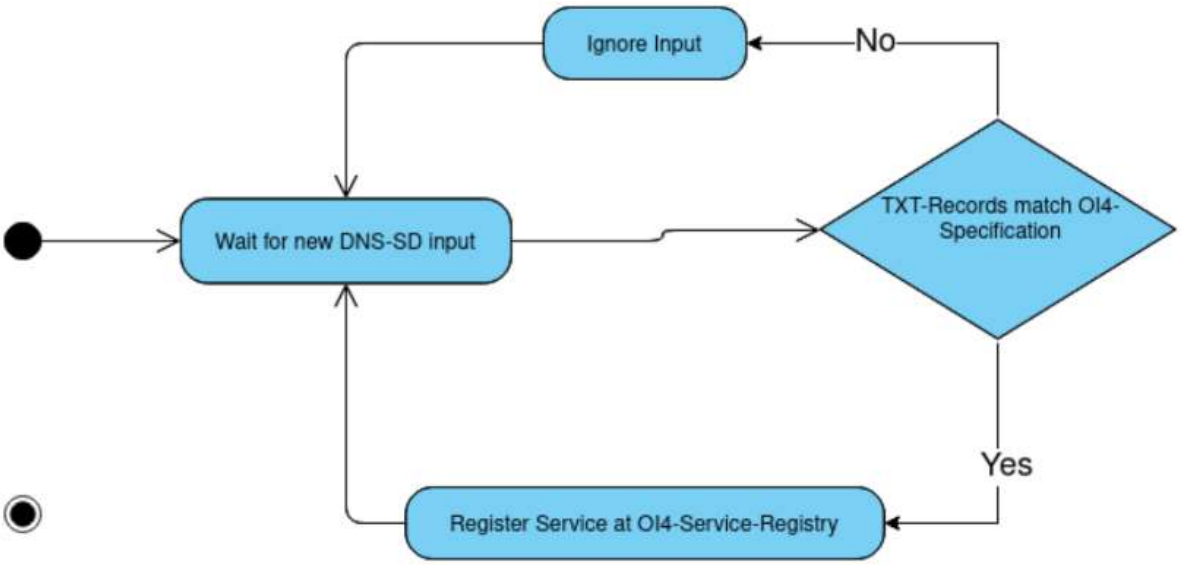
1. Introduction

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

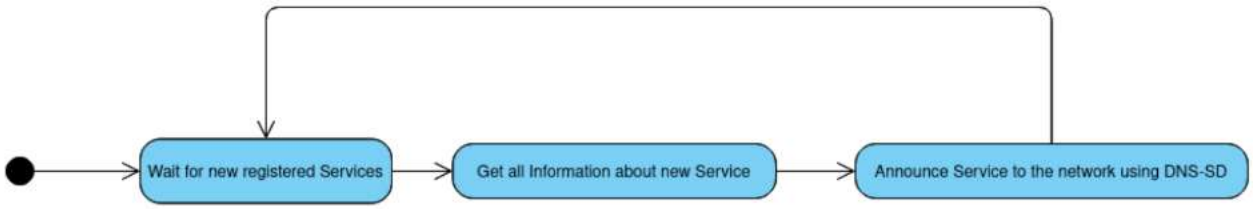
2.1 UA-001 Registering Services, that are found using DNS-SD

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:	OI4-Service-Registry, Services registered at the OI4-Service-Registry, Docker-Application
Precondition:	The services have to be registered at the OI4-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 OI4-Service-Registry



2.3 UA-003 Deployment

Related Business Process:	-
Use Cases Objective:	Simple software deployment as target.
System Boundary:	-
Precondition:	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'.
Postcondition on success:	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'.
Users:	-
Triggering Event:	-

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/ Exemplary Docker Test-Applications

The Project should contain simple Docker applications, which use and test the basic functionalities of the system. This is necessary in order to optimally check the system for faults and to eliminate them.

3.2 /NF30/ 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 /LF10/ Running in Docker

The Application has to be able to run in the context of a Docker Container for all functions to work properly. The instructions to build and configure such a container shall be included in the User documentation.

The master asset model described in the document (https://github.com/GoranErdeljan/TINF19C-Team-4-Service-Registry/files/5407856/Open.Industry.4.0.Alliance.-.Development.Guideline_v0.11.pdf) is mapped to the TXT records. The TXT records is the data model for DNS-SD entries. This data model is divided into several strings, since the length of a TXT records string may only have a maximum length of 255 bytes (see https://tools.ietf.org/html/rfc6763#section-6.1) section 4 "The format of each constituent string within the DNS TXT record is a single length byte, followed by 0-255 bytes of text data"). Each of these strings always consists of a key and a value. The key is the attribute name in the master asset model and the value is a JSON string for the corresponding object. The prerequisite for the system to work well is that all specifications and required attributes mentioned by the OI4 are set. In addition to the data contained in the MAM, "oi4=true" should be entered there as a key-value-pair. This value tells the interface that it should register the given MAM in the registry. The key "DataSetWriterId" is used to check whether the interace of this MAM was sent by itself via DNS-SD.

Example for TXT Records:

```
oi4=true
Manufacturer={"Locale":"en-US","Text":"Hilscher"}
ManufacturerUri="urn:hilscher.com"
Model={"Locale":"en-US","Text":"Registry Application"}
ProductCode="OI4-REG"
HardwareRevision=""
SoftwareRevision="0.11.8"
DeviceRevision=""
DeviceManual="Not available"
DeviceClass="Registry"
ProductInstanceUri="urn:hilscher.com/Registry%20Application/OI4-REG/undefined"
RevisionCounter=0
Description={"Locale":"en-US","Text":"OI4 Registry application"}
DataSetWriterId=urn:undefined.com/DNS_SD_INTERFACE/DNS_SD_INTERFACE
/undefined
```

INTERFACE Beschreibung, was muss beachtet werden damit gut läuft, OI4-Schnittstellenbeschreibung

4.2 /LF20/ 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 /LF30/ 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 /LF40/ 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. In this case, the service is entered in the OI4 service registry.

5. References

- https://github.com/GoranErdeljan/TINF19C-Team-4-Service-Registry/blob/master/PROJECT/CRS/TINF19C_CRS_Service-Registry_Team_4_0v1.pdf
- [1] https://openindustry4.com/
- DNS-SD https://www.rfc-editor.org/rfc/rfc8882.html