

# Customer Requirements Specification

## *(Lastenheft)*

(TINF19C, SWE I Praxisprojekt 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)

Rotebühlplatz 41  
70178 Stuttgart

Version	Date	Author	Comment
0.1	07.09.2020		created

## Allgemeine Hinweise:

*Alles, was in dieser blauen Schriftart gesetzt ist, dient nur zur Erläuterung und sollte im fertigen Lastenheft nicht mehr auftauchen!*

*Der Umfang dieses Dokuments darf sechs Seiten nicht überschreiten.*

*Ein Lastenheft enthält eine grobe Beschreibung aller fachlichen Anforderungen, die das zu entwickelnde Produkt erfüllen muss. Die Inhalte des Lastenheftes (CRS) dienen als Grundlage für das Pflichtenheft und können -wenn sinnvoll- im Pflichtenheft (SRS) wieder verwendet werden.*

## CONTENTS

<b>1.</b>	<b>Goal .....</b>	<b>3</b>
<b>2.</b>	<b>Product Environment .....</b>	<b>4</b>
<b>3.</b>	<b>Product Usage .....</b>	<b>5</b>
3.1.	Business Processes .....	5
3.1.1.	<BP.001>: <Name> .....	5
3.2.	Use Cases .....	8
3.2.1.	<UC.001> Use Case Name .....	8
3.3.	Features .....	10
3.3.1.	/LF10/ .....	10
3.3.2.	/LF20/ .....	7
<b>4.</b>	<b>Product Data .....</b>	<b>11</b>
4.1.	/LD10/ .....	11
4.2.	/LD20/ .....	11
<b>5.</b>	<b>Other Product Characteristics .....</b>	<b>12</b>
5.1.	/NF10/ .....	13
5.2.	/NF20/ .....	13
5.3.	System Environment .....	13
<b>6.</b>	<b>References .....</b>	<b>14</b>

## 1. Goal

An asset in the meaning of Industry 4.0 can both be a single device and a Docker hosted application. In both cases there must be a way to announce the offered capabilities via a service discovery mechanism. A suitable mechanism for this is DNS-SD [1] [2].

The system will combine the capabilities of DNS-SD and the already existing Service-Registry developed by the OI4-Alliance.

A Service Registry is an application that offers a list of available Services in the network. It makes it possible for Users and other applications to find a service, that matches their requirements. Therefore, a Service-Registry shall also contain Data about the capabilities of the service.

The to be developed application must make it possible to, on the one hand use the DNS-SD mechanism to discover Services, that aren't already known to the Service Registry and on the other hand announce the already known Services via DNS-SD in the Network.

The new interface must be implemented and integrated into an Example project on Linux. The interface must, just like the provided Service Registry, run in a Docker environment. The Docker Environment will also contain a MQTT-Broker, which hosts the OI4-MessageBus. Additionally, there must be test applications, which also run in Docker Containers. These test-applications shall be able to list the services, that were announced over the network, via a simple GUI. The developed applications shall be made publicly available in form of an open-source project.

## 2. Product Environment

The usage environment of the software consists of the Docker-Application itself, which is connected to the OI4-Service-Registry via a MQTT-Broker. In this context the communication using the MQTT-Protocol is called the OI4-MessageBus. The to be developed Docker-Application shall act as an interface between the Service-Registry and a Service-Discovery mechanism called DNS-SD. This adds an additional way not only of registering new services, but also of announcing already known services to the network.

These services can for example be Industry 4.0 related. Having them registered at a Service-Registry makes them easily accessible.

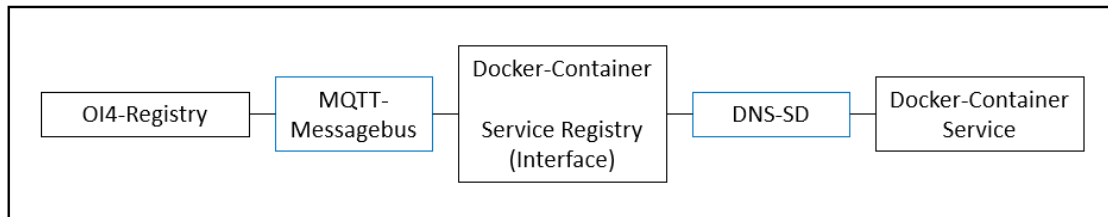


Figure x: Product Environment

### 3. Product Usage

The following business processes, use cases and features shall be supported by the system.

*Dieser Abschnitt hat die Aufgabe, die Anwendung des zu entwickelnden Systems sowohl überblicksartig als auch detaillierter aus Benutzersicht zu beschreiben. Abschliessend sollen die vom Produkt erwarteten Features beschrieben werden.*

#### 3.1. Business Processes

*Falls notwendig, sind hier die identifizierten Geschäftsprozesse näher zu beschreiben. Jeder von ihnen erhält einen eigenen Unterabschnitt gemäß dem Template. In diesem Abschnitt wird der Ablauf der Geschäftsprozesse des vorigen Abschnittes genauer beschrieben. Diese Abläufe sind es, die das zu entwickelnde System ausschnittsweise unterstützen soll.*

##### 3.1.1. <BP.001>: <Grundfunktion>

<i>Triggering Event:</i>	<i>Wird ausgeführt, sobald das Gerät o. Applikation konform mit der Open Industry 4.0 kommuniziert</i>
<i>Result:</i>	<i>OI4 Registry listet alle Applikationen und Geräte auf, Testet sämtliche Assets und gibt Feedback an den User</i>
<i>Involved Roles:</i>	<i>Geräte o. Applikationen, Mitarbeiter</i>

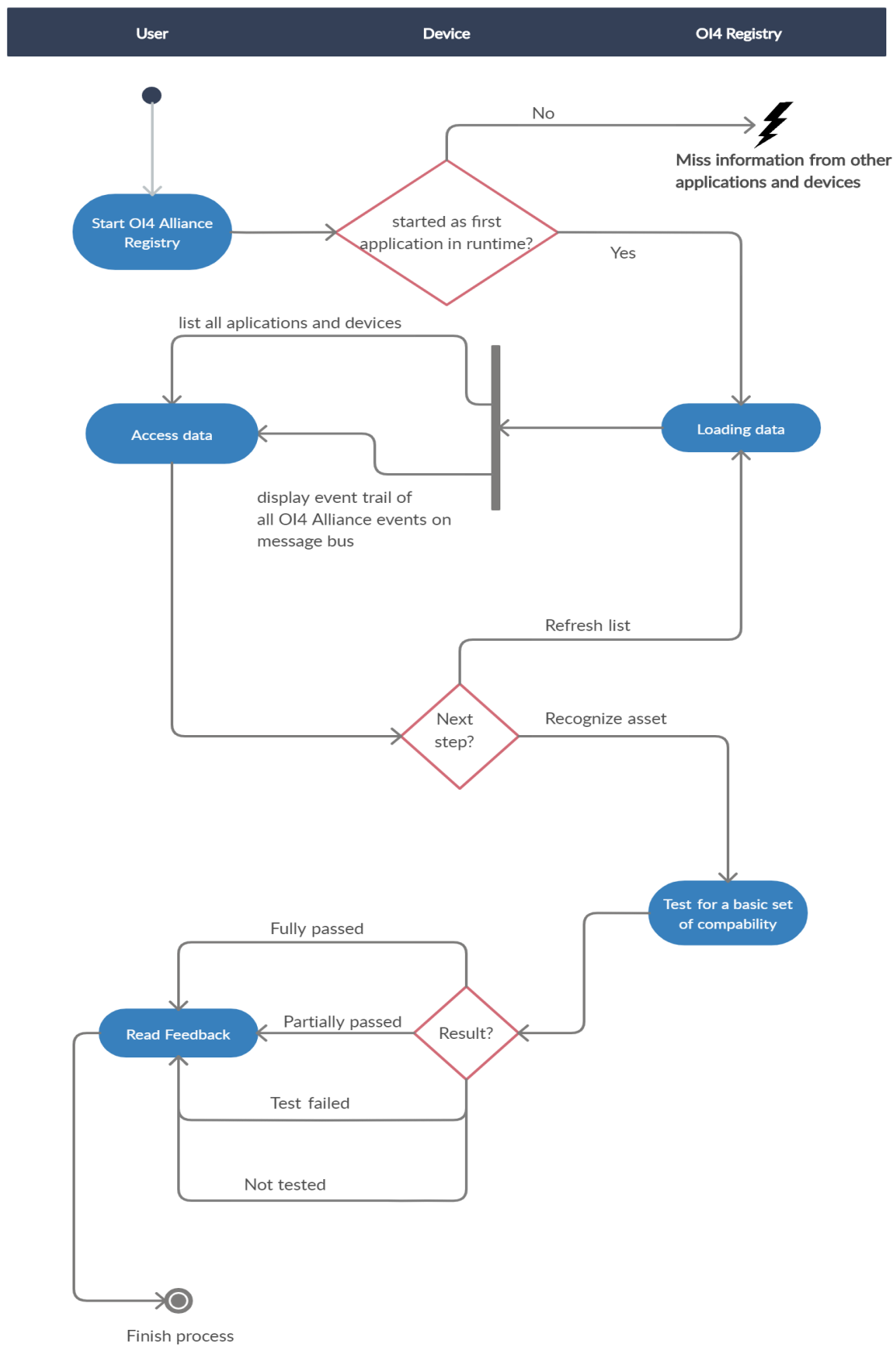


Figure 2.2: <BP.001> Workflow of general function

3.1.2. <BP.001>: <Publish on OI4 Messagebus>

<i>Triggering Event:</i>	<i>Wird ausgeführt, sobald ein neuer Input</i>
<i>Result:</i>	Es wird überprüft ob txt records den Spezifikationen entsprechen. Wenn ja, wird es auf dem OI4 Messagebus veröffentlicht, sonst ignoriert und wartet auf nächsten Eintrag
<i>Involved Roles:</i>	<i>Geräte o. Applikationen, Mitarbeiter</i>

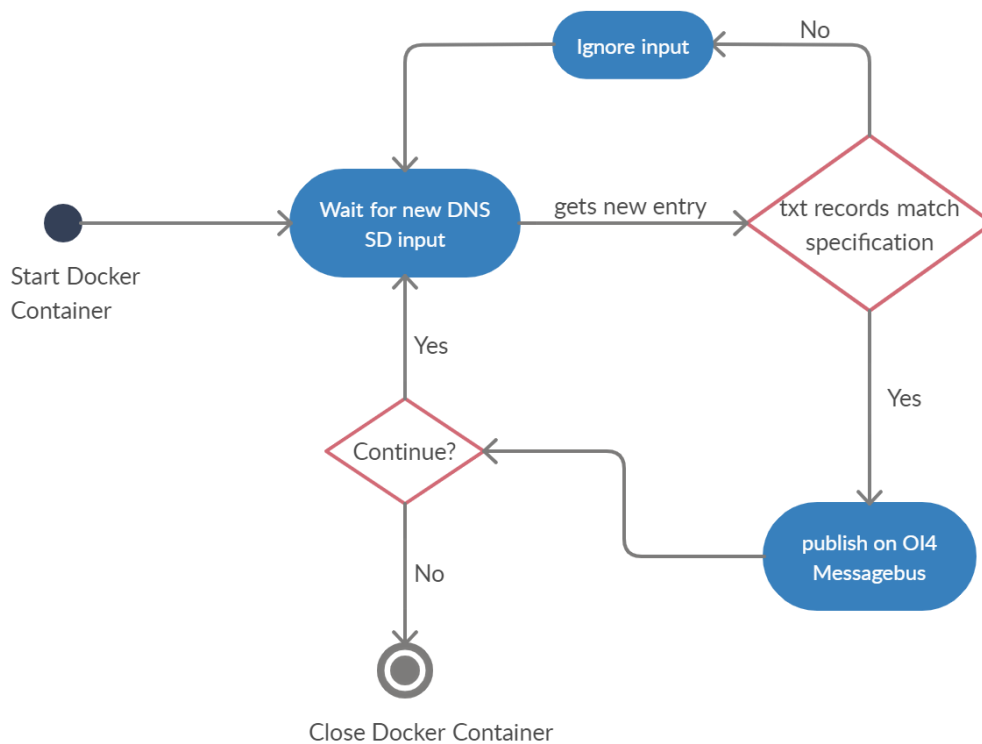
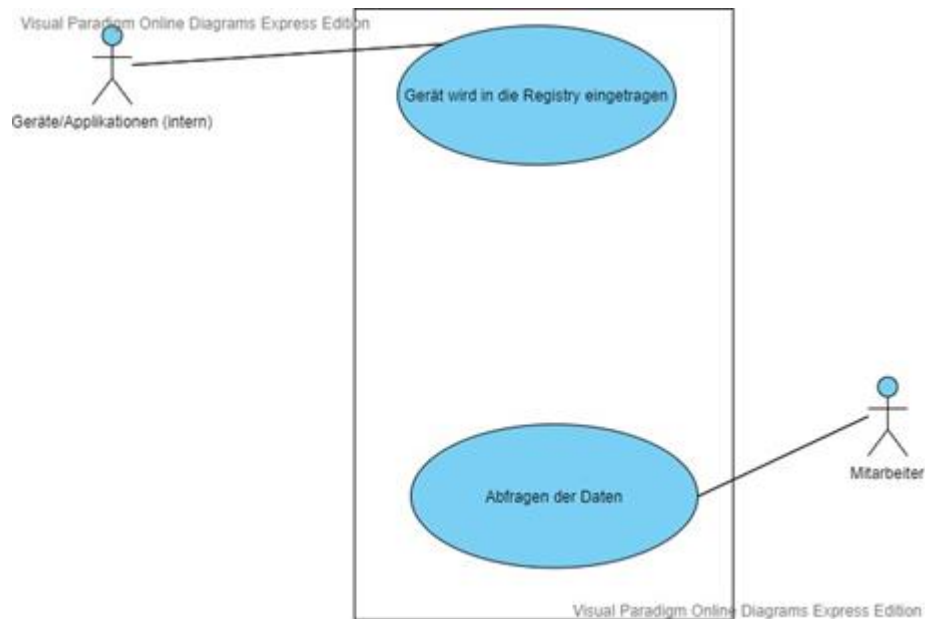


Figure 2.3: <BP.002> Workflow of input handling

### 3.2. Use Cases



<https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=UseCaseDiagram>

Figure 1: Use Case Overview Diagram

#### 3.2.1. UA-001

<b>Related Business Process:</b>	UA-001
<b>Use Cases Objective:</b>	Auflistung der Geräte- und Applikationsinformationen.
<b>System Boundary:</b>	Geräte und Applikationen innerhalb der Firmenumgebung und der Mitarbeiter.
<b>Precondition:</b>	Die beteiligten Nutzer müssen korrekt und konform mit der Open Industry 4.0 kommunizieren.
<b>Postcondition on success:</b>	Die beteiligten Nutzer müssen korrekt und konform mit der Open Industry 4.0 kommunizieren.
<b>Beteiligte Nutzer:</b>	Geräte o. Applikationen, Mitarbeiter
<b>Triggering Event:</b>	Eintrag in die Registry: <ul style="list-style-type: none"> <li>- Wird ausgeführt, sobald das Gerät o. Applikation konform mit der Open Industry 4.0 kommuniziert.</li> </ul> Anzeigen der Daten:



	<i>User greift via Webbrowser auf die Registry zu und fragt die Daten ab.</i>
--	---

### **3.3. Features**

#### **3.3.1. /LF10/ Forwarding Information to the Service-Registry**

The Docker Application shall periodically check if there are any new DNS-SD entries and, in case the entry is meant for the Service Registry perform the following reactions: read the DNS-TXT records of the DNS-SD entry, push the received Information to the OI4-MessageBus.

#### **3.3.2. /LF20/ Announcing Services via DNS-SD**

The Docker Application shall check if there are any new entries in the Service-Registry and, in case they are not already announced by DNS-SD, perform the following reactions: announce services via DNS-SD.

#### **3.3.3. /LF30/ Announce itself via DNS-SD**

The exemplary Docker Application shall check if the user has configured it to announce itself via DNS-SD and perform the following reactions: announce itself via DNS-SD.

#### **3.3.4. /LF40/ Register itself directly at the Service-Registry**

The exemplary Docker Application shall check if the user has configured it to register itself directly at the OI4-Service-Registry and perform the following reactions: register itself at the OI4-Service-Register.

#### **3.3.5. /LF50/ Show advertised Services**

The exemplary Docker Test Applications shall check if there are any Services registered in the Service Registry and perform the following reactions: show Services on a UI.

#### **3.3.6. /LF60/ Check Conformity**

The Docker Application shall check if the received data is conform with the specifications of the OI4-MessageBus.

## 4. Product Data

*In diesem Abschnitt werden die Hauptdaten und Datenschnittstellen beschrieben, mit denen das Softwareprodukt arbeiten soll und die bereits identifizierbar sind (siehe Abb. 1). Im Allgemeinen werden diese Hauptdaten eines Programms auch nonvolatil gespeichert.*

### 4.1. /LD10/ OI4-MessageBus

The OI4-MessageBus is based on the MQTT-Protocol. All Communication via the MessageBus must follow the specifications made by the OI4-Alliance. The OI4-Registry is reached via this interface. Therefore, the MessageBus is used when new DNS-SD entries are discovered and must be added to the Registry but also when the application checks whether there are new Services on the Registry, which might need to be announced via DNS-SD.

### 4.2. /LD20/ DNS-SD

The DNS-SD system is used to announce Services, that are registered on the Service-Registry, over the network. It is also used to look for other Services, which might not be registered on the Service-Registry. These services are then added to the Service-Registry using the OI4-MessageBus.

## 5. Other Product Characteristics

This section describes the already known non-functional requirements for the product.

*Die Schnittstellenbeschreibung (die Service Registry) ist eine Docker Applikation. Die Kommunikation des client-Server läuft über ein mqtt broker und registriert so die angebundenen Services. Die Service Registry wird über ein DNS-SD Server im Netzwerk veröffentlicht das Frontend der Applikation hat einen tabellarischen Aufbau und beinhaltet Informationen wie eine Anwendungstabelle, Gerätetabelle und ein Global Event Trail. Die Informationen in den Tabellen sind Hersteller, Modell und Geräteklasse, zusätzlich wird die Gesundheit und eine Konformität geprüft. Die Konformität ist die Anzeige für alle durchlaufenden Tests.*

*Jedes erkannte Asset wird auf eine grundlegende Kompatibilität mit der Open Industry 4.0 Alliance-Spezifikation getestet. Das Ergebnis wird wie folgt angezeigt:*

*Alle Tests für GET / PUB-Methoden und zugehörige Nutzlastformate vollständig bestanden: Namur*

*Teilweise bestanden, weil GET / PUB-Methoden beantwortet wurden, die zugehörige Nutzlast jedoch nicht korrekt war: (Grüner Haken)*

*Fehlgeschlagen, weil obligatorische GET-Methoden nicht beantwortet werden: (rotes Kreuz)*

*Noch nicht getestet (weder erfolgreich noch fehlgeschlagen): ?*

*Es befindet sich auch einen Expertenmodus um weitere Features zu nutzen.*

*Der Expertenmodus bietet folgende Optionen:*

*Laden und speichern Sie die Konfiguration in und aus einem JSON*

*Maintainer-Aktionen*

*Löschen Sie ein einzelnes Asset (in der Asset-Tabelle verfügbar).*

*Löschen Sie alle Assets. Warnung: Dadurch wird die Datenbank der Registrierung gelöscht.*

*Löschen Sie alle Protokolldateien. Wenn die Protokollierung in einer Datei aktiviert ist, werden alle Dateien auf der Festplatte gelöscht.*

*Frontend-Konfiguration*

*Bearbeiten Sie die Anzahl der angezeigten Überwachungselemente im globalen und lokalen Ereignispfad.*

*Backend-Konfiguration*

*Aktivieren Sie die Anzeige der Registrierung in der Anwendungstabelle*

*Bearbeiten Sie die Prüfstufe, die die Registrierung abhört. Dies wirkt sich darauf aus, welche Ereignisse in der Ereignistabelle angezeigt werden.*

*Aktivieren Sie die Protokollierung in einer Datei. Dies kann entweder den Dateilogger aktivieren oder aktivieren.*

*Bearbeiten Sie die Größe der Protokolldatei. Die Größe der Protokolldatei (wenn die Protokollierung aktiviert ist) kann von 500 KB bis 10 MB angepasst werden.*

*Die Aufgabe dieses Abschnittes ist die Beschreibung der nicht-funktionalen Anforderungen. Dabei handelt es sich um Charakteristiken oder Qualitäten, die das Produkt attraktiv machen und es von vergleichbaren Produkten unterscheiden.*

*In diesem Abschnitt werden die wesentlichen Eigenschaften des zu entwickelnden Produktes beschrieben, die nicht direkt die zu leistende Funktionalität betreffen.*

In diesem Abschnitt sollen diese bereits definierbaren Anforderungen in „Balzert-Notation“ aufgelistet werden und nach den Regeln der Anforderungsschablone ([https://www.sophist.de/fileadmin/SOPHIST/Publikationen/Broschueren/SOPHIST\\_Broschuere\\_MASTeR.pdf](https://www.sophist.de/fileadmin/SOPHIST/Publikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf)) ausformuliert werden.

### 5.1. /NF10/ User-Documentation

The project shall contain an extensive user-documentation. The documentation shall contain instructions on building the Docker-Images and deploying them as Docker-Containers and specifications on how TXT-records are formatted.

### 5.2. /NF20/ Deployment

The system shall be easy in Deployment.

### 5.3. System Environment

This section describes the system environment required to operate the product.

...

Ein Asset im Sinne der Industrie 4.0 kann sowohl ein einzelnes Gerät als auch eine über Docker gehostete Applikation sein, bspw. in einem Edge-Gateway. Das OI4-Registry ist eine Docker Applikation deshalb fehlt es am Projekt an keine Hardware. Die Anbindung über eine DNS-SD ist ein wichtiger Parameter für die Veröffentlichung der Services.

*Hier sollten alle wesentlichen und notwendigen Parameter der Systemumgebung (Hardware, Software) beschrieben werden, soweit diese bereits festlegbar ist.*

## 6. References

- [1] <http://www.ietf.org/rfc/rfc6763.txt>
- [2] <https://blog.stackpath.com/dns-discovery-edge-compute/>
- [3] <https://www.docker.com/>
- [4] [https://en.wikipedia.org/wiki/Edge\\_computing](https://en.wikipedia.org/wiki/Edge_computing)
- <https://www.edgelo.com/edgelo-device-service-discovery/>