**Customer Requirements Specification**

**(Lastenheft)**

(TINF19C, SWE I Praxisprojekt 2020/2021)

Project: OPC UA Server Farm

Customer: Rentschler & Holder

Rotebühlplatz 41

70178 Stuttgart

Supplier: Team 3 (Niclas Hörber, Kay Knöpfle, Nico Fischer, Daniel Zichler, Niklas Huber, Phillip Förster)

Rotebühlplatz 41

70178 Stuttgart

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comment** |
| 0.1 | 16.10.2020 | Niklas Huber | created |
| 0.2 | 17.10.2020 | Niklas Huber | Copied goal from the task |
| 0.3 | 22.10.2020 | Niklas Huber, Kay Knöpfle | Added Product Environment |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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.

#### ****Offene Punkte****

In diesem Abschnitt sollen alle Probleme und offenen Fragen gesammelt werden. Bei einem fertigen Lastenheft sollte er leer sein, aber bei Zwischenversionen kommt diesem Abschnitt besondere Bedeutung zu!

CONTENTS

*1.* Goal 3

2. Product Environment 4

*3.* Product Usage 5

*3.1.* Business Processes 5

3.1.1. <BP.001>: <Name> 5

3.2. Use Cases 5

3.2.1. <UC.001> Use Case Name 6

*3.3.* Features 7

3.3.1. /LF10/ ….. 7

3.3.2. /LF20/ ….. 7

*4.* Product Data 8

4.1. /LD10/ ….. 8

4.2. /LD20/ ….. 8

*5.* Other Product Characteristics 9

5.1. /NF10/ ….. 9

5.2. /NF20/ ….. 9

5.3. System Environment 9

6. References 10

# Goal

The goal of this project is to develop a Server farm which supports the testing of OPC UA Clients. The Server farm should provide multiple virtual OPC UA Server via Network. These virtual OPC UA Server profiles should be parameterizable via an AutomationML configuration file (in CAEX 3.0). In conclusion, the software should simulate multiple OPC UA Servers on one computer for testing OPC UA Clients. The target group are developer and tester of applications with OPC UA Client-Interface. The documentation of the software as well as the documentation of the development and project is also part of the goal.

# Product Environment

OPC UA (Open Platform Communications United) is a standardized and safe exchange of data and information for industry automation between machines, devices, computers and services from different industry sectors. Therefore, it is independent of programming languages, operating systems and system suppliers. (<https://www.industry-of-things.de/was-ist-opc-ua-definition-architektur-und-anwendung-a-727188/>)

The main parts of an OPC UA Environment are an OPC Server and Client. The Server is the foundation of the OPC communication, it implemented the OPC interfaces. The Client is the logical counterpart to the Server and can get data from the UPC Server. An OPC Test Client is a special Client that can test the function and configuration of a UPC Server. (<https://www.opc-router.de/was-ist-opc-ua/#OPC-Foundation-Video>) In the context of this project, OPC UA Client-Expert (<https://www.unified-automation.com/products/development-tools/uaexpert.html>) will be used as Test Client.

With CEAX 3.0 in AutomationML (AML) an OPC Server can be configured. AML is short for Automation MarkUp Language and it is the connecting element for the seamless automation planning. For that AML describes automation plans as objects and use various standards to describes them. The relevant standard in the context of this project is CEAX 3.0. It is used as to describe hierarchical structures and links. It is also XML based, a meta data format and it is standardized in IEC 62424.

This UPC UA Server Backend will be implemented via open62541-Stack. It is an open source implementation of OPC UA in the programming languages C99 and C++98, it´s platform independent. For that open62541 provides tools to implement OPC UA clients and servers. (<https://open62541.org/doc/open62541-current.pdf>)

The server farm will be running via one system. That means that all Server profiles will be simulated via one device. And this Server farm will be tested via the already mentioned Test Client.

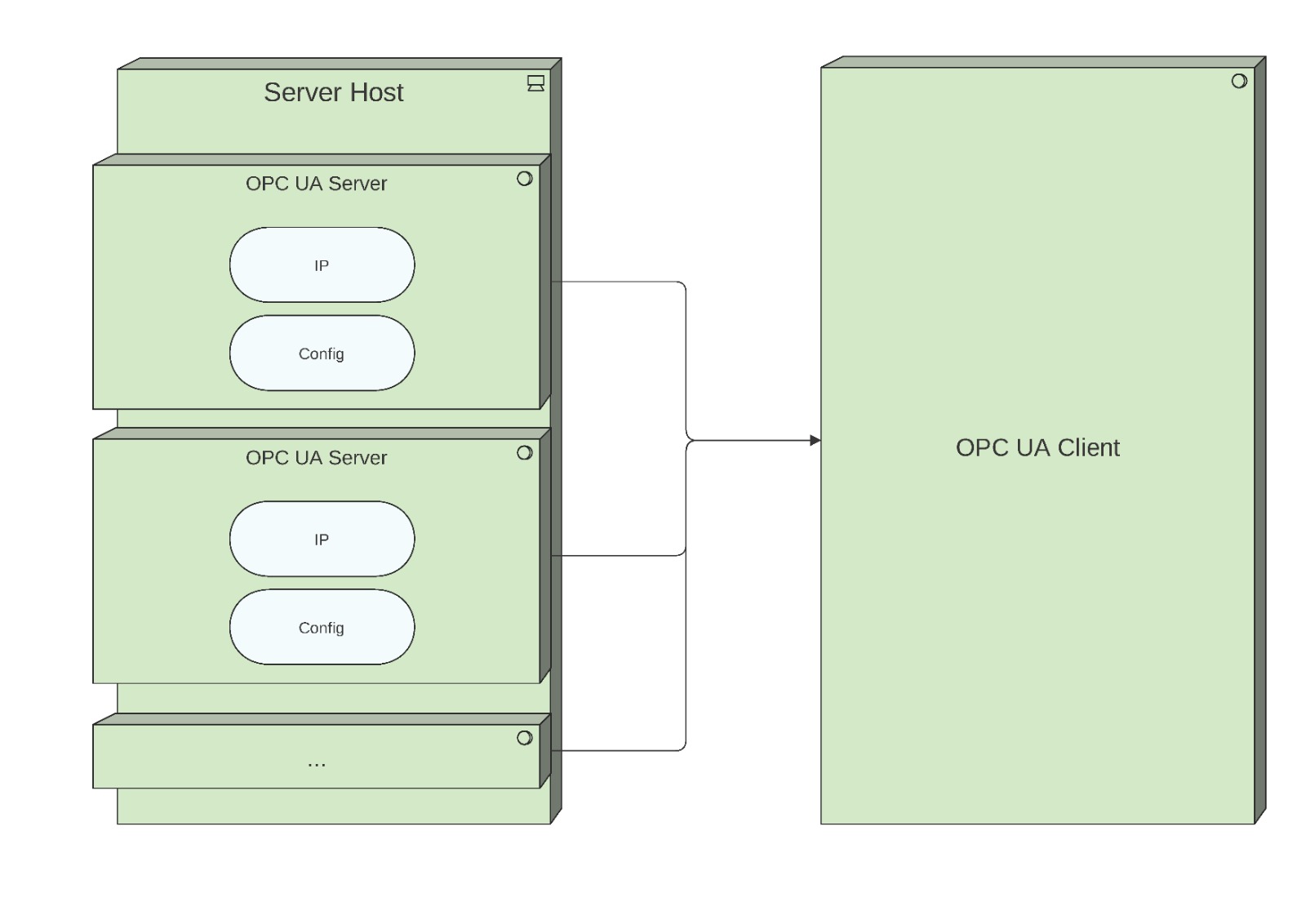


Figure 1 Product Enviornment

Dieser Abschnitt hat die Aufgabe den Einsatzbereich des zu entwickelnden Systems klarzustellen. Dazu gehören Erläuterungen der notwendigen Fachbegriffe und deren Zusammenhänge ebenso wie die Darstellung der systemrelevanten Abläufe im Einsatzbereich.

Unter dem Produkteinsatz versteht man sowohl den direkten Problembereich, in dem das zu entwickelnde System eingesetzt werden soll, als auch die umgebenden Geschäftsprozesse.

Hier also den Problembereich des Projektes benennen und erläutern, ob es zu unterstützende Abläufe im Einsatzbereich (Geschäftsprozesse) gibt und wo sie zu finden sind.

Dieser Abschnitt muss so geschrieben sein, dass er, den Laien mit der Terminologie und den Zusammenhängen im Problembereich vertraut macht. Daher muss die Beschreibung möglichst allgemein sein. Außerdem sollte der Text gut strukturiert sein. Auch der Einsatz von erläuternden Graphiken ist manchmal sinnvoll.



Figure x: Product Environment

Wichtig ist es auch noch, gemachte Annahmen sauber von den oben beschriebenen Fakten getrennt aufzulisten. Dies erleichtert eine spätere Fehlersuche, wenn das System nicht die Erwartungen erfüllt.

# Product Usage

The main purpose of the software will be the simulation and testing of multiple virtual OPC UA Servers which are simulated on one system. Each Server can be configured via sperate profiles and will start after finishing the configuration. From a user perspective, the main use case of this software is the testing of the simulated Servers.

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

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.

## Business Processes

This section will handle the starting of the Server Farm, respectively of one Server.

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.

Kunde erstellt profil, lädt das in Server, Server started sich

### <BP.001>: Server start

|  |  |
| --- | --- |
| Triggering Event: | User want to start the UPC UA Server farm, respectively one Server with a profile.  <Handlung oder Zeitpunkt, die Geschäftsprozess auslöst bzw. zu dem er beginnt> |
| Result: | The Sever starts with the profile.  <Was im Falle einer erfolgreichen Ausführung des Geschäftsprozesses erreicht werden soll> |
| Involved Roles: | User and UPC UA Server  <Rollenname derjenigen, die an der Durchführung des Geschäftsprozesses beteiligt sind. Das können auch existierende Systeme sein.> |

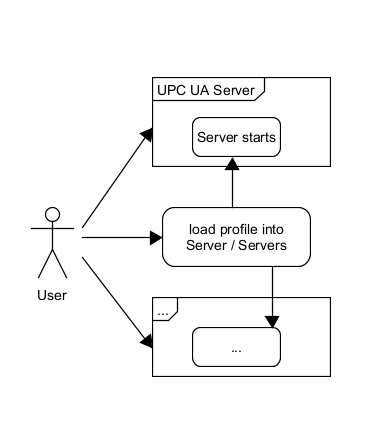


Figure 2: <BP.001> Server start (Unfertiges Diagramm, nur ob Prinzip richtig)

## Use Cases

This Project will implement a UPC UA Server farm, which runs on one device. The virtual OPC UA Servers will be parameterizable with the Server profiles. Each virtual Server will be able to tested via OPC UA Client UA-Expert from the user.

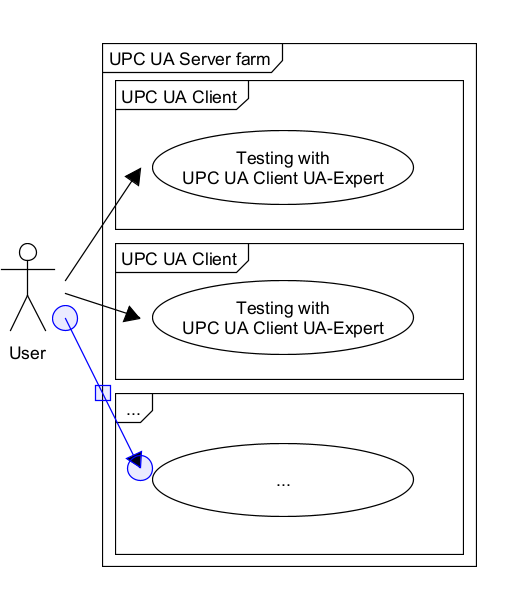
**

Figure 3: Use Case Overview Diagram (Unfertiges Diagramm, nur ob Prinzip richtig)

OPC UA Clients Testen (steht in Anforderungen)

Aufgabe dieses Abschnittes ist es, einen Überblick über die Produktfunktionen zu geben. Dazu wird ein Use Case Diagramm eingesetzt, das eine abstrakte Sicht auf die Produktfunktionen und die externen Beteiligten an diesen Funktionen gibt.

### <UC.001> Testing UPC UA Client(s)

Dieser Abschnitt muss für jeden Use Case wiederholt werden. Hier soll der Use Case mit einer ID versehen und kurz textuell sowie ggf. mit einem Aktivitätendiagramm erläutert werden.

|  |  |
| --- | --- |
| **Related Business Process:** | <BP.001>: Server start Prozess-ID: <elementarer Geschäftsprozess > |
| **Use Cases Objective:** | User want to test multiple UPC UA Clients.  Ausführliche Beschreibung des Zieles des Use Cases |
| **System Boundary:** | The Server farm is the system boundary.  Systemgrenze, die betrachtet wird (aus Diagramm des vorigen Abschnittes) |
| **Precondition:** | * OPC UA Servers must be started with the fitting profiles * The Servers must run without errors * Installation of Test Client   Was muss garantiert werden, damit der Use Case durchgeführt werden kann? |
| **Postcondition on success:** | ????  Was muss sichergestellt werden für eine erfolgreiche Ausführung des Use Case |
| **Beteiligte Nutzer:** | User, Test Client and UPC UA Servers (Server farm)  Rollenname: Beschreibung des Nutzers, der mit dem System interagiert. Nutzer können auch andere Systeme sein. |
| **Triggering Event:** | Start of the testing with the Test Client.  Handlung oder Zeitpunkt, die Use Case auslöst bzw. zu dem er beginnt |

## Features

In diesem Abschnitt sollen die bereits definierbaren funktionalen Features in „Balzert-Notation“ aufgelistet werden und nach den Regeln der Anforderungsschablone ausformuliert werden. à (<https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf>)

### /LF10/ …..

The [Component] shall {[adverb]} check if [Condition] and, {in case of violation trigger the appropriate reaction as specified in the fault database} {perform the following reactions:[List of Reactions]}

### /LF20/ …..

The [Component] shall {[adverb]} check if [Condition] and, {in case of violation trigger the appropriate reaction as specified in the fault database} {perform the following reactions:[List of Reactions]}

# 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.

Mögliche Product Data?

Konfigurationdatei

Open62541-Stack

10 Server Profile

Test Client

## /LD10/ …..

…

## /LD20/ …..

….

# Other Product Characteristics

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

…

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/Puplikationen/Broschueren/SOPHIST\_Broschuere\_MASTeR.pdf*](https://www.sophist.de/fileadmin/SOPHIST/Puplikationen/Broschueren/SOPHIST_Broschuere_MASTeR.pdf)) ausformuliert werden.

## /NF10/ …..

The software/system shall support …

## /NF20/ …..

….

## System Environment

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

…

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

# References

[1] …

[2] …