System Requirements Specification

(Pflichtenheft)

(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	01.11.2020	Fischer	created
0.2	06.11.2020	Fischer	Update of the whole document
0.3	11.11.20202	Knöpfle	Product Requirements

CONTENTS

1. Introduction

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.

1.1. 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 [1].

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 it. An OPC test client is a special client that can test the function and configuration of an OPC server [2]. In the context of this project, OPC UA Client-Expert [3] will be used as test client.

With CAEX 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 CAEX 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 OPC 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 [4].

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.
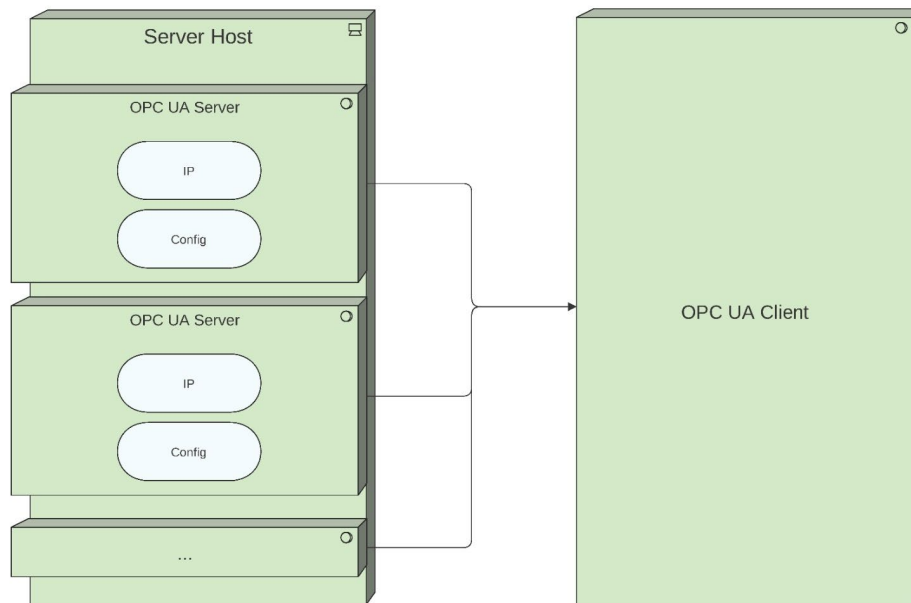
Figure 1- Product Environment

1.2. Use Cases

The software will implement an OPC 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 be tested via OPC UA Client UA-Expert from the user.
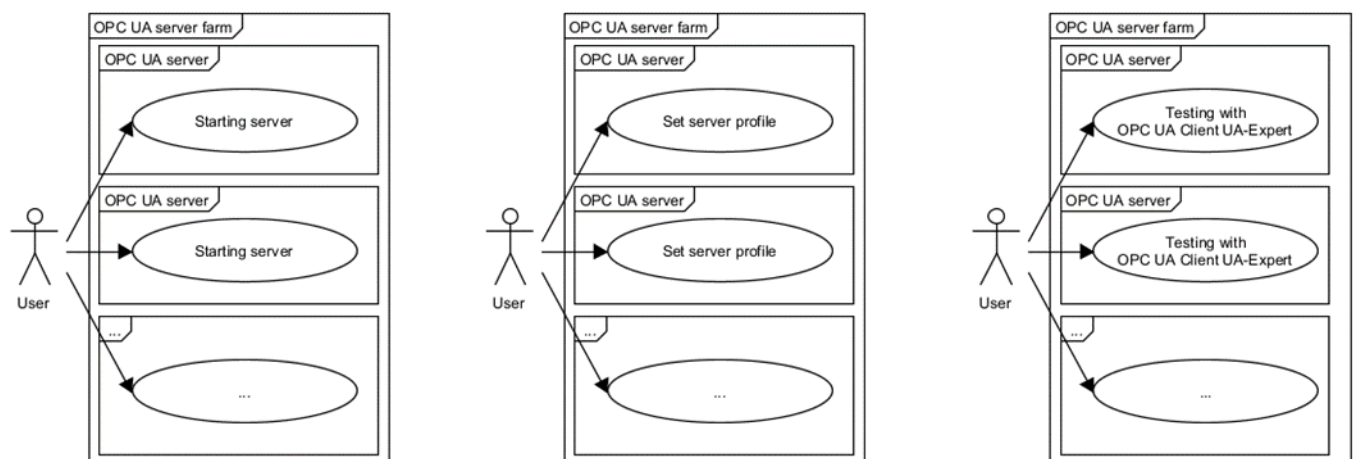

Figure 2 - Use Case Overview Diagram

1.2.1. <UC.001> Set server profiles

Related Business Process:	<BP.001>: Simplified testing of OPC UA clients
Use Cases Objective:	User wants to set server profiles before starting the server farm.
System Boundary:	Server farm and profiles itself are the system boundary.
Precondition:	- OPC UA server profiles must be defined - User know what kind of profile he needs
Postcondition on success:	Server profiles must work with the server farm
Involved roles:	User, OPC UA server profiles and OPC UA servers (Server farm).
Triggering Event:	User want to select the server profiles before starting the server farm.

1.2.2. <UC.001> Starting server farm

Related Business Process:	<BP.001>: Simplified testing of OPC UA clients
Use Cases Objective:	User wants to start the OPC UA server farm.
System Boundary:	The server farm is the system boundary.
Precondition:	The selected profiles must be loaded into the OPC UA servers
Postcondition on success:	The servers must be starting without errors.
Involved roles:	User, OPC UA server profiles and OPC UA servers (Server farm).
Triggering Event:	User want to start the server farm with already selected profiles.

1.2.3. <UC.001> Testing OPC UA Client(s)

Related Business Process:	<BP.001>: Simplified testing of OPC UA clients
Use Cases Objective:	User wants to test multiple OPC UA clients.
System Boundary:	The running server farm is the system boundary.
Precondition:	- OPC UA servers must be started with the selected profiles - Installation of test client
Postcondition on success:	The servers must run without errors
Involved roles:	User, OPC UA server profiles and OPC UA servers (Server farm).
Triggering Event:	Start of the testing with the test client. And starting the server(s) with the fitting profile(s).

2. Product Requirements

The following functionalities shall be supported by the system.

2.1. /LF10/Command Line Inteface

The User can interact with the tool via Command Line for ease of use and automatability. The User can specify a configuration file and a port or default settings will be used.

2.2. /LF20/File validation

The specified configuration file is checked for validity. If errors occur they will be logged.

2.3. /LF30/Server configuration

The Server shall be configured according to the specified configuration file and port.

2.4. /LF40/Server startup

The server starts after the configuration finished successfully.

2.5. /LF50/Server shutdown

The server stopps after external interupt.

2.6. /LF60/Logging

Errors and Events shall be logged to the comman line or a logfile.

3. Product Data

3.1 /LD10/Configuration File in CAEX 3.0

The system, respectively the multiple virtual servers shall be able to be configured via an AutomationML-configuration file in CAEX 3.0

3.2 /LD20/OPC UA Server Profilers

The software should provide 10 meaningful and high bandwidth OPC UA server profiles.

4.   Non-Functional Requirements

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

4.1 /NF10/Open62541-Stack

The system shall be implementing the OPC-UA-Backend with open62541-Stack.

4.2 /NF20/System with CAEX 3.0 parser

The system shall be using a CAEX 3.0 parser.

4.3 /NF30/OPC UA Client-Expert

The software should support the OPC UA Client UA-Expert as test client.

4.4 /NF40/Logging

The software should write into a separate logfile and in the command line.

4.5 /NF50/Command Line Interface

The software shall support at least a command line interface.

4.5 /NF60/License

The software is licensed under the MIT License [5].

5. References

[1] https://www.industry-of-things.de/was-ist-opc-ua-definition-architektur-und-anwen-dung-a-727188/

[2] https://www.opc-router.de/was-ist-opc-ua/#OPC-Foundation-Video

[3] https://www.unified-automation.com/products/development-tools/uaexpert.html)

[4] https://open62541.org/doc/open62541-current.pdf

[5] https://github.com/ghNico/TINF19C-Team-3-OPC-UA-Server-Farm/blob/master/LICENSE