HTWK

Hochschule für Technik,
Wirtschaft und Kultur Leipzig

Fakultät Ingenieurwissenschaften

6010 - Praxisprojekt

Praxisbericht bei Siemens AG

Author Ivan Agibalov
Betreuer Prof. Dr.-Ing. Andreas Pretschner

28. Juni 2022

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Aufgabenstellung

Hier steht iwas zur Aufgabestellung

Bedarf:

- Testframework für die Integrationstest von der Ladesäule
- Eigenständiges OCPP1.6 und später OCPP2.0 Server für die manuellen Tests
- ERK (Eichrechtkonformität) automatisiert überprüfen

Lösung: Beide Anforderungen sind sehr ähnlich zueinander = soweit es geht gleiches Code benutzen, nur die unterschiedlichen Teile jeweils für den Bedarf schreiben.

1.1 Anforderungen an Standalone Server

Der Server wird großtenteils für die manuellen Tests benutzt.

- Der Zustand des Servers soll für den Nutzer erreichbar sein (z.B. eine REST Schnittstelle)
- Der Nutzer soll in der Lage sein den Server zu parametrieren

1.2 Anforderungen an Testframework

- Der OCPP Server soll den Port selber auswählen können und dann ihn zurückgeben =¿ um mehrere Tests parallel starten zu können.
- Das Verhalten von dem Testserver soll geändert werden können (auch während der Tests)
- Das Defaultverhalten von dem Testserver soll parametrierbar sein (z.b. einen Benutzer hinzufügen)
- Alle Events, die den Zustand der getesteten Ladesäule aufdecken, sollen beobachtbar sein

1.3 Anforderungen an ERK Teil

Damit die Ladesäule ERK ist, muss dies für jeder Ladesäule überprüft werden. Dies ist immer das gleiche Prozess, das sich entsprechend automatisieren lässt. Der Teil mit OCPP Server wird gebraucht um nachzuweisen, dass die gemessenen Daten nirgendwo in der Software geändert wurden und werden entsprechend genauso an der Server übertragen und dort abgerechnet. Dafür muss man für jede Ladesäule einen Ladevorgang starten (Transaction), währendderen Strom fließt und gemessen wird. Die Ladesäule übertragt dabei die Sichertsschlussel an den Server um dann sicherstellen zu können, dass die Messdaten nicht manipuliert wurden. Nach dem Beenden der Transaction werden die gemessenen Daten mit der Transactionsdaten mittels eine Drittsoftware überprüft.

2 Umsetzung

2.1 OCPP Standalone Server und Testframework

Alle gestellten Aufgaben benötigen OCPP Server um OCPP Kommunikation mit der Ladesäule auswerten zu können.

Um den Server so flexibel wie möglich zu planen, wurde von meiner Seite entschieden, ihn nach "Clean Architectureßu implementieren.

2.2 ERK Automatisierungstool

Das Automatisierungstool sollte laut der Aufgabenstellung in einer anderen bestehenden Tool integriert werden. Das bereits existierte Tool ist mit Electron geschrieben, daher um das Integrationsprozess so einfach wie möglich zu machen, wurde von meiner Seite entschieden auch mit Electron zu arbeiten. Das ERK Tool soll 2 vorgegebenen Schnittstellen benutzen:

- Die Kommunikationsschnittstelle zu der Autosimulator
- Die Kommunikationsschnittstelle zu dem Messgerät

In dem bestehenden Tool wurde zur Begin der Umsetzung nur die Kommunikationsschnittstelle zur Autosimulator implementiert, die Kommunikationsschnittstelle zum Messgerät wurde von dem Messgerätehersteller vorgegeben. Die Schnittstelle wurde mit einem Pythonscript implementiert. Damit man die Schnittstelle nicht anpassen soll um verschiedene Fehler zu vermeiden, wird das Pythonscript vom Electronprogramm mit entsprechenden Parametern gestartet, und danach wird nur die Konsoleausgabe empfängt und entsprechend ausgewertet. Am Ende des Ablaufs soll ein PDF Report entstehen, das alle wichitge Informationen über den Testablauf beinhaltet. Neben dem Report entstehen auch 2 weitere Ausgaben:

- Logs von dem Pythonscript
- Alle Messdaten

Die Logs können später benutzt werden um die Fehler im Pythonscript zu finden und um den Testablauf später nachmachen zu können.

Die Messdaten werden aufbewahrt um später die Richtigkeit des Reports nachweisen und mit anderen Werkzeugen überprüfen zu können.

Benutzten Werkzeuge:

- Typescript als Programmiersprach
- Svelte als Frontendframework
- Bulma als CSS Framework

Der komplette Testablauf:

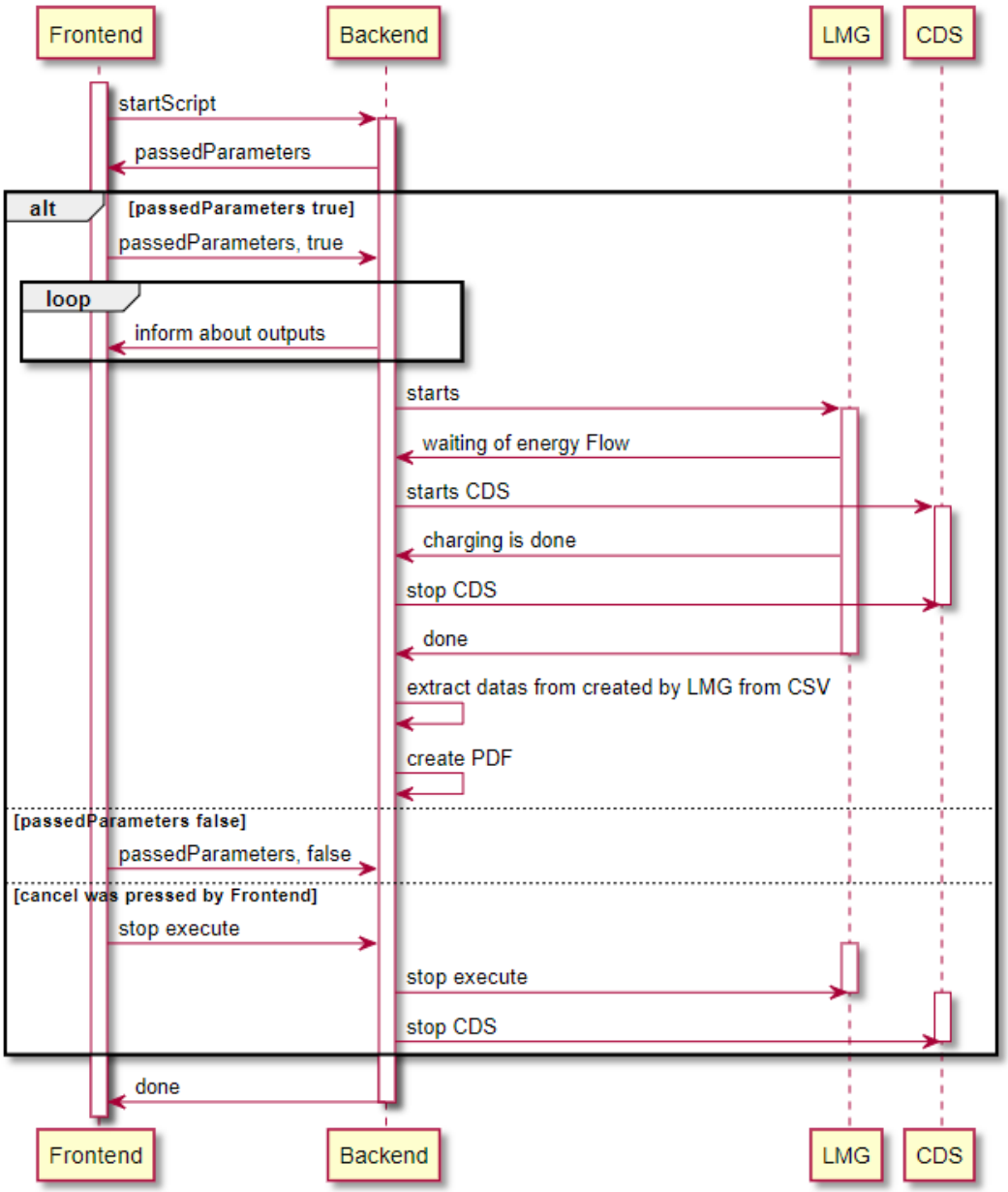

Abbildung 1: Sequenzdiagramm ERK Tool

Source: Eigene Quelle

Beispiel des Reports:

SIEMENS

Accuracy test according to MessEG

DC left

Tested Specifications

Date and Time	21.06.2022 09:52
Name of tester	1
SN of tested Meter	1
FA Charger	1
tested operating point	4A, 1000V

Measurement setup

ZES ZIMMER Electronic Systems GmbH LMG671 ATE 00022104 3.071-R67112

Channel	Group	Type	In Use	I Range	U Range	Sensor
1	1	L60-CH-A2	no	32.0	1000.0	no
2	1	L60-CH-A2	no	32.0	1000.0	no
3	1	L60-CH-A2	no	32.0	1000.0	no
4	2	L60-CH-A2	yes	0.005 (7.5 scaled)	1000.0	PCT600 (SN: 7100060045)
5	3	L60-CH-A2	no	0.6 (200.0 scaled)	1000.0	PCT200 (SN: 0480070039)
6	4	L60-CH-A2	no	32.0	1000.0	no

Measurement results

The 2.00% allowed deviation is used by 98%

Total energy	Total energy: 113.87 Wh
Total deviation	Total deviation: 2.23 Wh (1.96%)
Result	passed

Source: Eigene Quelle