FH Salzburg

BACHELORARBEIT 1

Erarbeitung eines Systems zum Laden von Elektrofahrzeugen mit Überschussenergie aus Energiegemeinschaften

durchgeführt am Studiengang
Informationstechnik und System-Management
Fachhochschule Salzburg GmbH

vorgelegt von
Muhammed Ali Tosun, Bernhard Dürnberger

Studiengangsleiter: FH-Prof. DI Dr. Gerhard Jöchtl

Betreuer: Ing. Ernst Forsthofer

Puch/Salzburg, Januar 2023

Eidesstattliche Erklärung

Ich erkläre hiermit eidesstattlich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt habe. Weiters versichere ich hiermit, dass ich die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission weder im In- noch im Ausland vorgelegt und auch nicht veröffentlicht.

10. Januar 2023

Muhammed Ali Tosun, Bernhard Dürnberger

Abstract

To support the transition to renewable energies, the Austrian Government introduced local energy communities (LEC). LECs enable power grid users to share self produced energy (i.e. energy produced through solar panels). This provides financial benefits to the LEC members and promotes the usage of renewable energies. To maximize these benefits, it is important to maximize the consumption of the produced energy within an LEC. One possible strategy to accomplish this, is to increase the loading power of an electric vehicle, when renewable energy is present and decrease the loading power, when no renewable energy is present. In this work, a system will be developed, which collects data about the produced and consumed amounts of energy of multiple LEC members. From this data, the amount of available energy will be determined. This energy is then consumed, by increasing the loading power of electric vehicles. To test the functionality of the developed system, the system interacted with a simulated LEC. The results of the simulation showed, that the system was able to detect and consume the surplus energy wihtin the simulated LEC.

Keywords: *surplus charging, EVCC, photovoltaics, local energy communities*

Inhaltsverzeichnis

Abkürzungsverzeichnis

ITS	Informationstechnologie und Systemmanagement
EEG	Erneuerbare-Energie-Gemeinschaften
YAML	Yet Another Markup Language (Serialisierungssprache für Datenstrukturen)
API	Application Programming Interface
M-Bus	Meter-Bus
REST	Representational State Transfer

Abbildungsverzeichnis

Tabellenverzeichnis

Quelltextverzeichnis

1 Einleitung

Energiegemeinschaften sind Vereinigungen, welche dazu berechtigt sind, eigen erzeugte Energie zu verbrauchen, zu speichern und zu verkaufen. Diese Rechte können genutzt werden, um beispielsweise finanzielle Vorteile für die Teilnehmer der Energiegemeinschaft zu generieren. So ist es möglich, dass Teilnehmer von Energiegemeinschaften von niedrigeren Energiekosten profitieren. Dies ist dann möglich, wenn die Energiegemeinschaft Energie an ihre Teilnehmer günstiger zur Verfügung stellt als konventionelle Energieanbieter.
Um diese finanziellen Vorteile nutzen zu können, ist es notwendig, den Energieverbrauch der Teilnehmer anzupassen. Dazu muss der Anteil an (günstiger) Energie aus der Energiegemeinschaft erhöht und der Anteil an (teurer) Energie von konventionellen Energieanbietern gesenkt werden. Um dies zu ermöglichen, wird in dieser Arbeit der Ladeprozess von Elektroautos gesteuert. Dabei wird die Ladeleistung von Elektroautos abhängig von der verfügbaren (= überschüssigen) Menge an selbst erzeugter Energie aus der Energiegemeinschaft erhöht oder vermindert (siehe Abschnitt 2.1).
Zusammengefasst sollen folgende Fragen innerhalb dieser Arbeit beantwortet werden:

- Wie ist es möglich, die Energiemesswerte einer Energiegemeinschaft zu messen?
- Wie können die Energiemesswerte aller Haushalte einer Energiegemeinschaft zentral gesammelt werden?
- Wie können die gesammelten Energiemesswerte genutzt werden, um einen Energieüberschuss innerhalb der Energiegemeinschaft festzustellen?
- Wie kann die überschüssige Energie innerhalb der Energiegemeinschaft durch das Laden von Elektrofahrzeugen verbraucht werden?

2 Theoretische Grundlagen

Im folgenden Abschnitt sollen die Technologien, welche im praktischen Teil dieser wissenschaftlichen Arbeit verwendet werden, auf Grundlage der existierenden Standards analysiert werden. Ebenso werden die wichtigsten Eigenschaften und Rahmenbedingungen erneuerbarer Energiegemeinschaften recherchiert.

2.1 Erneuerbare Energiegemeinschaften

Eine Energiegemeinschaft ist eine Vereinigung von mindestens zwei juristischen oder natürlichen Personen. Es wird unterschieden zwischen Erneuerbare Energiegemeinschaften und Bürgerenergiegemeinschaften. Für diese Arbeit sind nur Erneuerbare Energiegemeinschaften relevant.

2.1.1 Pflichten einer EEG

Folgende gesetzliche Pflichten der erneuerbaren Energiegemeinschaft sind im Rahmen dieser Arbeit relevant:

- EEGs dürfen nicht gewinnorientiert arbeiten. Sie müssen ökologische, wirtschaftliche oder sozialgemeinschaftliche Vorteile für ihre Mitglieder oder für die Gebiete, in denen sie tätig sind, bringen [1].
- Private Unternehmen dürfen nicht an EEGs teilnehmen, wenn deren Teilnahme ihre gewerbliche oder berufliche Haupttätigkeit darstellt [1].
- Die Verbrauchs- und Erzeugungsanlagen der Teilnehmer müssen auf den Netzebenen fünf bis sieben im Konzessionsgebietes desselben Netzbetreibers verbunden sein [2].
- Treffen von Vereinbarungen betreffend der Verwaltung und Bearbeitung der Energiedaten der Erzeugungs- und Verbrauchsanlagen der teilnehmenden Netzbenutzer mit den Netzbetreiber [2].

2.1.2 Rechte einer EEG

Folgende gesetzliche Rechte der erneuerbaren Energiegemeinschaft sind im Rahmen dieser Arbeit relevant:

- EEGs haben das Recht, eigen erzeugte Energie aus erneuerbaren Quellen zu verbrauchen, zu speichern und zu verkaufen [1].
- Sie kann im Bereich der Aggregierung[1] von erneuerbaren Energien tätig werden und auch andere Energiedienstleistungen anbieten [1].

2.2 EVCC

EVCC ist eine Open-Source-Software , mit der eine Netzteilnehmer*in Elektrofahrzeuge laden kann. Der Ladevorgang wird automatisch angepasst, abhängig von der Art und Menge der aktuell bezogenen Energie. Die Netzteilnehmer*in kann beispielsweise ihren Eigenverbrauch erhöhen, indem an EVCC die Vorgabe gegeben wird, dass das Elektrofahrzeug geladen werden soll, wenn ein Überschuss an eigen erzeugter Energie vorhanden ist [4].

2.2.1 Funktionalitäten

EVCC besitzt folgende für diese Arbeit relevanten Funktionalitäten [4]:

- EVCC kann Energiemesswerte aus unterstützten Energiemessgeräte auslesen und somit verbrauchte, erzeugte und gespeicherte Energiemengen messen.
- EVCC ermöglicht die Steuerung von Ladegeräten wie Wallboxen oder steuerbaren Steckdosen, um den Ladevorgang von Elektrofahrzeugen zu regulieren.
- EVCC kann den aktuellen Ladezustand von unterstützten Fahrzeugen auslesen und ermöglicht somit Zielladen.
- EVCC bietet Schnittstellen (Modbus, HTTP, MQTT, etc.) um externe Datenquellen als Ladegerät, Energiemessgerät oder Fahrzeug zu nutzen.

2.2.2 Konfigurationsmöglichkeiten

Um unterstützte Geräte der Heiminstallation (Energiemessgeräte, „Ladegeräte“, Fahrzeuge) mit dem Programm EVCC zu verbinden, müssen diese mithilfe einer Konfigurationsdatei registriert werden. Außerdem können in der Konfigurationsdatei Parameter gesetzt werden, welche das Verhalten des Programms beeinflussen. Die Konfigurationsdatei

[1] Mehrere kleine Kundenlasten/Erzeuger können zu einem Marktteilnehmer zusammengefasst werden, um als Gesamtheit am Elektrizitätsmarkt aufzutreten [3].

besteht aus Schlüssel-Wert Paaren im YAML Ain't Markup Language-Format (YAML-Format) [4]. Genauere Beschreibungen der EVCC-Konfigurationsdatei sind der EVCC-Dokumentation zu entnehmen.

2.2.3 REST API

Eine wichtige Funktionalität von EVCC ist die angebotene REST-Schnittstelle. Es besteht die Möglichkeit, durch REST-Abfragen für diese Arbeit wichtige Parameter abzurufen als auch zu setzen. Folgend werden die zwei relevantesten API Endpunkte beschrieben [4]:

- **/api/state**
  Dieser Endpunkt stellt alle Parameter mit einem API-Abruf zur Verfügung. Die im JSON-Format übertragenen Daten können in einer Software durch einen JSON-Parser verarbeitet werden, um sie in einem lesbaren Format darzustellen oder für weitere Verarbeitungen zu nutzen. In Abb. 1 ist dargestellt, welche Parameter in einem API-Response enthalten sind.

Abbildung 1: Darstellung des Response eines API-Requests auf /api/state

- **/api/residualpower**
  Der Parameter »ResidualPower« ist ein lesbarer und schreibbarer Wert. Das beschreiben des Wertes erfolgt mit einem POST-Request. Er kann dazu genutzt werden, den erlaubten Netzbezug im PV-Modus (nur Energie aus der PV-Anlage wird zum Laden verwendet) zu regulieren. So kann beispielsweise mit einer ResidualPower von -1000 ein Netzbezug von 1000W, zusätzlich zur erzeugten Energie der PV-Anlage, eingestellt werden.

2.3 MQTT

MQTT ist ein offenes Netzwerkprotokoll. Es ist in der Anwendungsschicht des OSI-Referenzmodelles einzuordnen. Das Protokoll ist simpel, weshalb es gut geeignet ist, um die Kommunikation zwischen ressourcenarmen Geräten zu realisieren. Außerdem eignet sich das Protokoll gut für den Einsatz in Netzwerken mit geringer Bandbreite.

MQTT tauscht Informationen mithilfe des Client-Server-Modells und des Publish-Subscribe Modells aus. Clients senden zusätzlich zum Inhalt einer Nachricht einen Topic Filter an den Server (publish). Der Server sendet Nachrichten, welche einem Topic Filter zugeordnet sind, an alle Clients, welche sich für den Empfang der Nachrichten des Topic Filters angemeldet haben (subscribe).

MQTT-Nachrichten beinhalten die Information über das QoS Level. Das QoS Level bestimmt, wie die Zustellung der Nachricht sichergestellt wird. Wird das QoS Level 0 angewandt, erfolgt die Sicherstellung der Zustellung der Nachricht ausschließlich durch das darunterliegende Transportprotokoll. QoS Level 1 garantiert, dass die Nachricht mindestens einmal zugestellt wird. Dabei werden die Nachrichten einfach bestätigt. QoS Level 2 garantiert, dass die Nachricht genau einmal zugestellt wird. Dabei werden die Nachrichten zweifach bestätigt [5].

2.4 Meter-Bus

Der Meter-Bus (kurz M-Bus) beschreibt ein standardisiertes Feldbussystem, welches hauptsächlich zur Datenübertragung von Energiemesswerten genutzt wird. Es handelt sich um ein hierarchisch aufgebautes Kommunikationssystem, welches aus mindestens einem Client und einem oder mehreren Servern besteht [6].

Die Übertragung kann lt. [7] über eine verpolungssichere Zweidrahtleitung (siehe Abb. 2) oder per Funktechnologie erfolgen. Wie in Abb. 3 dargestellt werden Informationen bei der

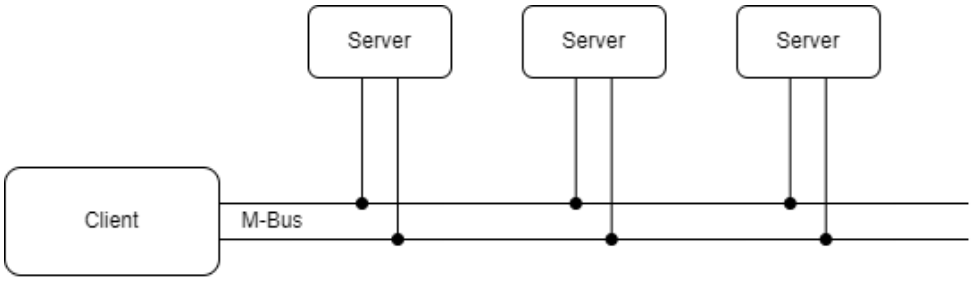

Abbildung 2: Darstellung einer M-Bus Topologie.

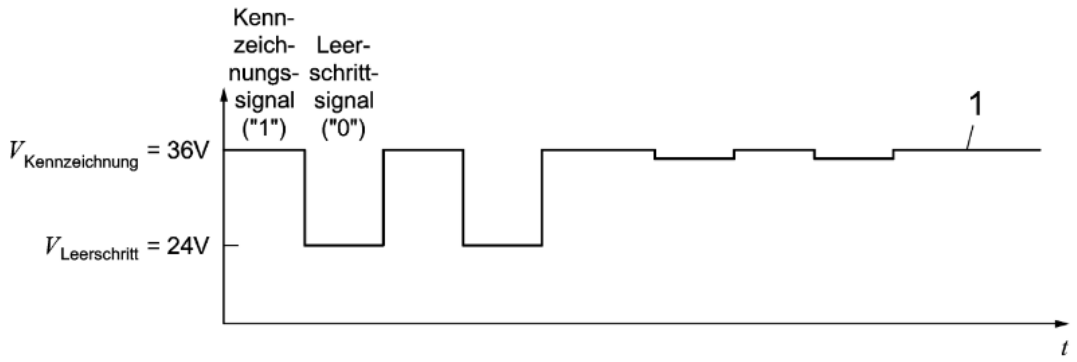

Abbildung 3: Bus Spannung am Repeater, Client überträgt an Server[7, S. 10]

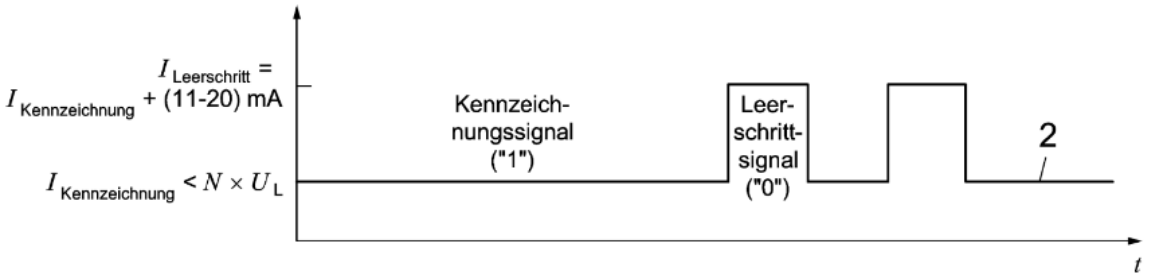

Abbildung 4: Stromverbrauch eines Servers, Server überträgt an Client[7, S. 10]

drahtgebundenen M-Bus Kommunikation vom Client zum Server über die Spannungspegeländerungen zwischen 36V (Kennzeichnungszustandsspannung U_{MARK}) und 24V (Wirkspannungspegel) übertragen. Hingegen erfolgt die Kommunikation vom Server zum Client über die Veränderung seines Konstantstroms (siehe Abb. 4). Server können durch die Spannungspegel im Bereich von 36V…24V oft mit Spannung versorgt werden, wodurch eine gesonderte Versorgung über separate Leitungen wegfällt. Als zu unterstützende Baudrate schreibt der Standard 300 Baud vor. Empfohlen werden außerdem die Baudraten 2400 Baud, 9600 Baud und 19200 Baud.

Von der Salzburg AG werden Smart-Meter der Marken Kaifa und Honeywell bei den Endverbrauchern als Stromzähler eingesetzt. Die Kundenschnittstelle dieser Smart-Meter

kommuniziert mit dem Protokoll M-Bus bei einer Baudrate von 2400 baud. Es muss beachtet werden, dass das Smart-Meter als Client konfiguriert ist. In einem Intervall von 5s werden Messwerte am Bus zur Verfügung gestellt [8].

2.5 Existierende Lösungen

Zum behandelten Thema existieren bereits mehrere Lösungsansätze, welche jedoch nur innerhalb eines Haushalts anwendbar sind. Folgend werden diese Lösungen kurz beschrieben und aufgezeigt, weshalb die in dieser Arbeit entwickelte Lösung einen Mehrwert darstellt.

- **EVCC**
  Die in Abschnitt 2.2 beschriebene Open-Source Software EVCC bietet die Möglichkeit, Elektroautos mit selbst erzeugter Überschussenergie aus dem eigenen Haushalt zu laden. Es steht jedoch keine Konfigurationsmöglichkeit zur Verfügung, um mehrere Haushalte an eine EVCC-Instanz anzubinden. Somit kann gesagt werden, dass sich diese Software zwar sehr gut für den Einsatz in einzelnen Haushalten eignet, jedoch nicht dafür ausgelegt ist, Überschussenergie in EEG's zu verteilen.

- **Heimspeicher**
  Als Alternative zum direkten Verbrauch der Überschussenergie steht der Einsatz eines Heimspeichers. Durch die Möglichkeit der Zwischenspeicherung kann Energie zu einem späteren Zeitpunkt genutzt werden, als sie produziert wurde. Dies würde zwar verhindern, dass Überschussenergie in das Netz eingespeist wird, kommt jedoch mit dem Nachteil eines geringen Wirkungsgrades von ca. 80% einher [9].

3 Praktische Umsetzung

Die Regulierung der Ladeleistung von Wallboxen ist ein wichtiger Faktor, um die Überschussenergie in Energiegemeinschaften bestmöglich zu nutzen. Durch die Flexibilisierung der Ladeleistung von Elektrofahrzeugen könnte der Energiebedarf in Haushalten und Gemeinschaften an die jeweilige Erzeugung von erneuerbarer Energie angepasst werden. Dies trägt dazu bei, den Ausgleich von Erzeugung und Verbrauch von Energie in Energiegemeinschaften zu verbessern und somit die Effizienz der Energienutzung zu erhöhen.

Es soll ein System entwickelt werden, welches erzeugte und verbrauchte Energie in der Energiegemeinschaft ermittelt. Ein Algorithmus entscheidet bei Anliegen von Überschussenergie, welche Wallbox durch Erhöhung der Ladeleistung zusätzliche Energie vom Netz beziehen soll. Durch eine geregelte Verteilung der Überschussenergie wird versucht, die Erzeugung und den Verbrauch in Waage zu halten.

3.1 Überblick Gesamtsystem

Der Aufbau des Systems besteht aus mehreren Komponenten, die in hierarchischer Struktur miteinander verbunden sind. Abbildung 5 zeigt einen groben Überblick der Struktur. Die oberste Ebene bildet der Überschuss-Verteiler (engl. surplus distributor), der alle Messdaten des Systems sammelt und in Abhängigkeit dieser die Wallboxen reguliert. Unter dieser Ebene befinden sich die jeweiligen Instanzen von EVCC (siehe Abschnitt 2.2) in den Haushalten, die für die Konnektivität zu den Wallboxen, Smart-Metern und sonstigen Hardware-Installationen zuständig sind. Es ist zu erkennen, dass nur eine Instanz des Überschuss-Verteilers existiert, während mehrere EVCC-Instanzen dem System zugehörig sein können.

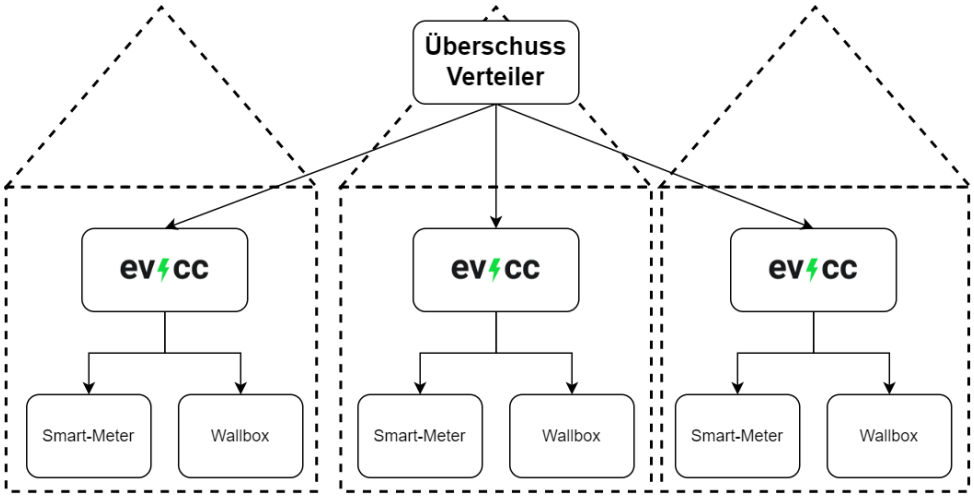

Abbildung 5: Hierarchische Struktur des Gesamtsystems

Abbildung 6 zeigt eine genaue Darstellung aller Komponenten in Form eines Prozessdiagramms. Es ist eine klare Kommunikationsstruktur mit definierten Schnittstellen zu erkennen, wodurch die Austauschbarkeit einzelner Komponenten gegeben ist.
Folgend werden alle Komponenten kurz erläutert:

- **Salzburg AG Smart-Meter**
  Das Smart-Meter misst die physikalische Leistung, welche von der Wallbox, der PV-Anlage und den Verbrauchern im Haus erzeugt bzw. verbraucht wird. In Form einer Kundenschnittstelle bietet das Smart-Meter der Salzburg AG zum Auslesen der Messwerte eine M-Bus Schnittstelle (siehe Abschnitt 2.4) an.

- **ESP32**
  Der ESP32 ist für das Auslesen der Messwerte über die Kundenschnittstelle des Smart-Meters zuständig. Die ausgelesenen Werte werden über das MQTT-Protokoll zur Verfügung gestellt.

- **MQTT Formatter**
  Die auf MQTT bereitgestellten Messwerte können aufgrund der falschen Formatierung nicht direkt von EVCC eingelesen werden. Deshalb muss der MQTT Formatter die Nachrichten des ESP32 in ein geeignetes Format bringen, sodass EVCC diese einlesen kann.

- **EVCC**
  Das Open Source Projekt läuft eigenständig in jedem einzelnen Haushalt und dient als Datensammler aller Messwerte eines Haushalts. Ebenso wird eine Schnittstelle

zum Anbinden verschiedenster Wallboxen angeboten, wodurch eine Steuerung der Ladeleistung von Elektroautos möglich wird.

- **Überschuss Verteiler**
  Diese Komponente ist, wie in Abb. 5 ersichtlich, den EVCC-Instanzen übergeordnet und liest über REST-Abfragen die wichtigsten Daten aus. Auf Basis dieser Daten entscheidet der Überschuss-Verteiler, ob die Ladeleistung einzelner Wallboxen erhöht oder verringert werden soll.

- **Influx DB**
  Die Messdaten aller EVCC Instanzen werden vom Überschuss-Verteiler ausgelesen und als Zeitreihen in eine InfluxDB geschrieben. Diese Datenbank wurde ausgewählt, da sie sich besonders für die Persistierung von Zeitreihen-Daten eignet und sie mit wenigen Schritten implementiert werden kann [10].

- **Grafana**
  Grafana ist eine Open-Source-Software zur Datenvisualisierung und -analyse. Sie wird häufig eingesetzt, um Daten aus verschiedenen Quellen in Form von Dashboards zu visualisieren und zu überwachen [11].

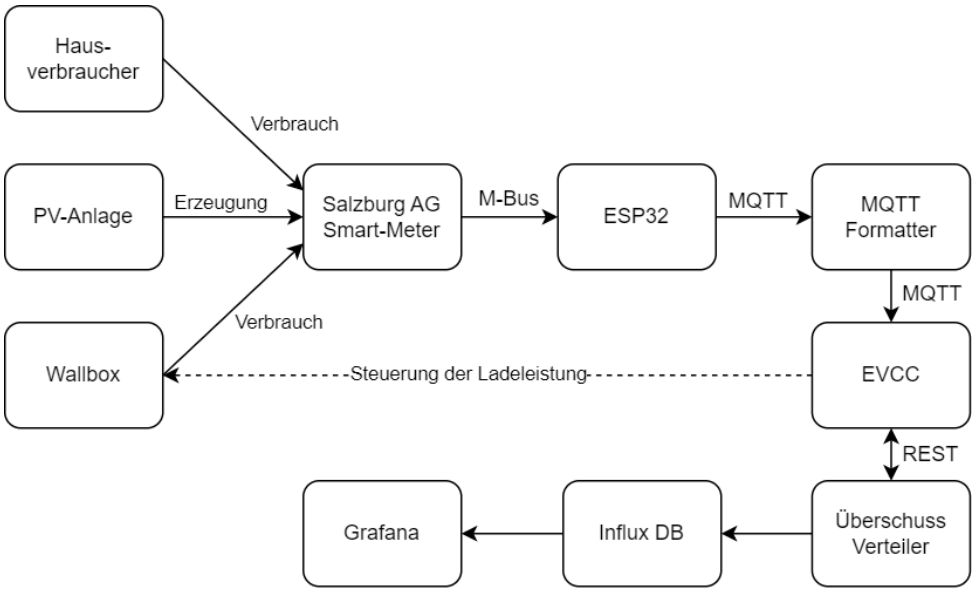

Abbildung 6: Prozessdiagramm des Gesamtsystems

3.2 Anbindung der Salzburg AG Smart-Meter

Für einen Betrieb des in dieser Arbeit entwickelten Systems ist ein Smart-Meter als Messstelle zwingend erforderlich. Da der Probebetrieb in einem Salzburger-Haushalt statt-

finden soll, muss das zur Verfügung gestellte Smart-Meter der Salzburg AG in das Gesamtsystem eingebunden werden. Über das M-Bus Protokoll (siehe Abschnitt 2.4) können sämtliche Messwerte über die Kundenschnittstelle des Smart-Meters ausgelesen werden. Als Hardware für diese Aufgabe kommt ein ESP32 zum Einsatz, welcher direkt an die Kundenschnittstelle angeschlossen wird. Der Vorteil dieses System-on-a-Chip (SoC) ist die integrierte WLAN-Funktionalität [12]. Dadurch lassen sich die ausgelesenen Messwerte drahtlos abfragen, wodurch keine Kabelverbindung zwischen ESP32 und EVCC (Abschnitt 2.2) benötigt wird.
Die auf dem ESP32 laufende Software wird von einem externen Dienstleister entwickelt. Diese liest die Messwerte vom Smart-Meter aus und führt sie in das MQTT-Protokoll (beschrieben in Abschnitt 2.3) über. Da sich EVCC ein bestimmtes Format der MQTT-Nachrichten erwartet, müssen diese umformatiert werden.

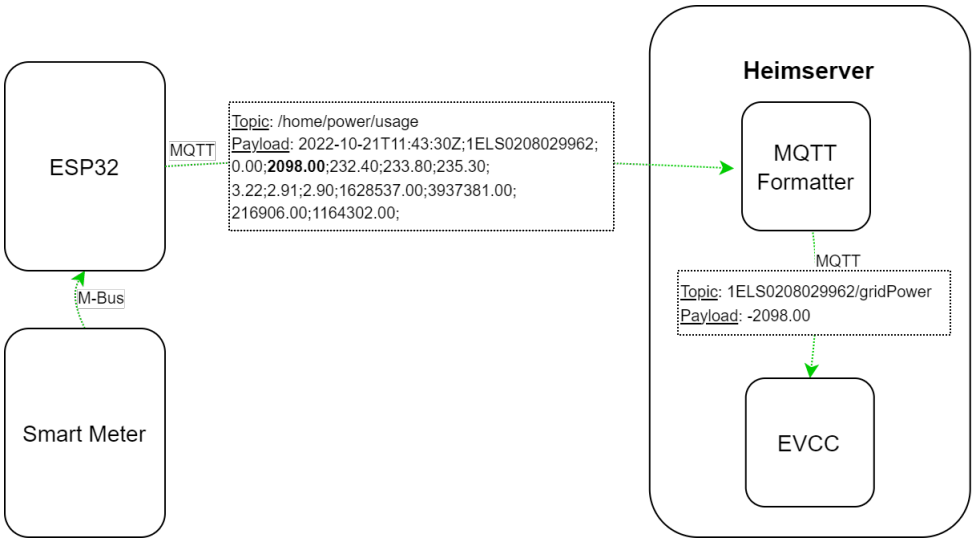

Abbildung 7: Anbindung des Smart-Meters an EVCC und Umformatierung einer MQTT-Nachricht

Eine Übersicht, wie ein Smart-Meter der Salzburg AG die Messwerte an EVCC überträgt, ist in Abb. 7 ersichtlich. Es ist zu erkennen, dass die Kommunikation zwischen Smart-Meter und ESP32 über den M-Bus stattfindet. Die vom ESP32 versendeten MQTT-Nachrichten werden am MQTT-Topic /home/power/usage vom MQTT-Formatter empfangen. Der MQTT-Formatter passt das Topic und die Payload an und sendet die Nachricht an EVCC weiter.

3.2.1 Umformatierung der MQTT-Nachrichten

Der ESP32 versendet die über M-Bus gesammelten Messwerte in folgendem Format per MQTT. Die Bedeutung der einzelnen Parameter ist in Tabelle 1 ersichtlich.

Rohes Nachrichtenformat:
2022-10-21T11:43:30Z;1ELS0208029962;0.00;2098.00;232.40;233.80;235.30;3.22;2.91;
2.90;1628537.00;3937381.00;216906.00;1164302.00;

Zeitstempel	**Geräte-ID**	**PowerPlus**	**PowerMinus**
2022-10-21T11:43:30Z;	1ELS0208029962;	0.00;	2098.00;
l1v	**l2v**	**l3v**	**l1c**
232.40	233.80	235.30	3.22
l2c	**l3c**	**EnergyPlus**	**EnergyMinus**
2.91	2.90	1628537.00	3937381.00

Tabelle 1: Format der MQTT-Nachrichten des ESP32

Die in Tabelle 1 dargestellten Parameter l1v, l2v und l3v beschreiben die Spannung der einzelnen Phasen des Stromnetzes, während l1c, l2c und l3c den Strom beschreiben. Diese werden nicht weiter verwendet. Die Parameter PowerPlus und PowerMinus sind in diesem Anwendungsfall am relevantesten, da nur diese Messwerte an EVCC weitergeleitet werden müssen. PowerPlus ist >0, wenn Leistung vom Netz bezogen wird. PowerMinus ist >0, wenn Leistung in das Netz eingespeist wird. Bei der Umformatierung werden beide Messgrößen zusammengeführt, da nur ein Leistungswert an EVCC übertragen werden kann (siehe Quelltext 2).

Es wurde ein Python Programm entwickelt, um die MQTT-Nachrichten des ESP32 einzulesen, in das spezifizierte Format zu bringen und anschließend auf das richtige MQTT-Topic rauszuschreiben. In Abb. 7 ist ersichtlich, wie eine Nachricht des ESP32 vom MQTT-Formatierer empfangen wird und in das für EVCC relevante Format gebracht wird.

Die in Quelltext 1 ersichtliche Funktion on_message() des MQTT-Formatters wird ausgeführt, wenn eine MQTT-Nachricht empfangen wird. In der Funktion werden alle Parameter aus der Nachricht extrahiert. Anschließend kann sofort die neue Nachricht erstellt und an EVCC weitergeleitet werden (siehe Quelltext 2).

```
1    def on_message(mosq, obj, msg):
2      # Prepare Data, separate columns and values
```

```
3       powerPlus, powerMinus, energyPlus, energyMinus, timeStamp, deviceid,
            l1v, l2v, l3v, l1c, l2c, l3c = 0,0,0,0,0,0,0,0,0,0,0,0
4       now = datetime.datetime.utcnow()
5       nowStr = now.strftime('%Y-%m-%d %H:%M:%S.%f')
6       try:
7         payload = str(msg.payload).removesuffix("'")
8         payload = payload.removeprefix("b'")
9         values = payload.split(";")
10        timeStamp = values[0] #unused
11        deviceid = values[1]
12        powerPlus = float(values[2])
13        powerMinus = float(values[3])
14        l1v = float(values[4])
15        l2v = float(values[5])
16        ...
```

Quelltext 1: Einlesen der MQTT-Nachricht des ESP32

```
1 #Send mqtt message to evcc with specific device id
2   try:
3     if(powerPlus > 0):
4       mqttc.publish(deviceid + MQTT_TOPIC_EVCC, str(powerPlus))
5     else:
6       mqttc.publish(deviceid + MQTT_TOPIC_EVCC, str(-powerMinus))
```

Quelltext 2: Senden der MQTT-Nachricht an EVCC

3.3 Sammeln der Energiemesswerte

Für dieses Projekt ist eine selbstständig laufende EVCC-Instanz in jedem Haushalt nötig. EVCC dient für jeden Haushalt somit als eigenständiger Datensammler. Unter Verwendung der in Abschnitt 2.2.3 beschriebenen, von EVCC angebotenen REST-Schnittstelle ist ein einfaches auslesen der Messwerte möglich.

3.3.1 Konfiguration des Überschuss-Verteilers

Um eine möglichst einfache Verwaltung der Mitglieder zu realisieren, wird eine config.yaml (Quelltext 3) zur Verfügung gestellt. Für jeden Teilnehmer muss ein Eintrag mit einer einmaligen ID, dem Namen des Teilnehmers, der EVCC IP-Adresse, dem Port und dem spezifischen Pfad zur Abfrage der Messwerte erstellt werden.

```
EEGMembers:
  - ID: 0
    Username: Benutzer0
    Evcc_Host: 10.0.0.56
    Evcc_Port: 7000
    Evcc_StatePath: /api/state
  - ID: 1
    Username: Benutzer1
    Evcc_Host: 10.0.0.67
    Evcc_Port: 7000
    Evcc_StatePath: /api/state
```

Quelltext 3: Beispielkonfiguration für die Datei config.yaml

Beim Start des Überschuss-Verteilers wird die Datei eingelesen. Für jeden Eintrag instanziert die JavaScript Software ein Objekt der Klasse EEGMember mit den Parametern aus der Konfigurationsdatei. Der Überschuss-Verteiler ruft anschließend im Betrieb die Daten aller in der config.yaml angegebenen Teilnehmer über die REST-API von EVCC ab.

3.3.2 Abfrage der Messwerte über die REST API

Die Funktion updateState() in der Klasse EEGMember ist dafür zuständig, die benötigten Parameter über die REST API abzurufen (ersichtlich in Quelltext 4). Da für jeden Teilnehmer der Energiegemeinschaft ein eigenes EEGMember-Objekt instanziert wird, muss diese Funktion für jeden Teilnehmer in konstanten Zeitintervallen ausgeführt werden. Da bei jedem API-Abruf alle Messwerte einer EVCC-Instanz abgerufen werden, existiert am System des Überschuss-Verteilers jeweils ein exaktes Abbild der Messwerte.

```

updateState() {
  const url = `http://${this.Evcc_Host}:${this.Evcc_Port}${this.
      Evcc_StatePath}`
  return fetch(url)
  .then(parseToObject)
  .then((data) => {
    this.state = data.result
    this.stateValid = true;
    persistence.writeMemberState(this);
  })
  .catch((error) => {
    this.stateValid = false;
    global.logger.error(`Error while fetching state from ${url} for user
        ${this.Username}: ${error}`)
```

```
14     })
15 }
```

Quelltext 4: Abruf der Messwerte von EVCC über die REST API

3.4 Datenverarbeitung

Das Ziel der Datenverarbeitung ist es, die Überschussleistung innerhalb der EEG festzustellen und anschließend durch die Steuerung der Ladeleistung von Elektroautos diese Überschussleistung zu verbrauchen. Dabei ist zu beachten, dass dieser Verbrauch möglichst exakt gesteuert werden soll, um einen übermäßigen Leistungsbezug aus dem Stromnetz zu vermeiden.
Die Datenverarbeitung besteht aus folgenden Schritten, welche periodisch durchlaufen werden:

1. Persistierung von Messdaten
2. Bestimmen der Überschussleistung
3. Verteilung der Überschussleistung
4. Steuerung der Ladeleistung

3.4.1 Persistierung der Messdaten

Zur Persistierung der Messdaten wird das Datenbankmanagementsystem InfluxDB (siehe Abschnitt 3.1) genutzt, weshalb die Daten in einer Zeitreihendatenbank gespeichert werden. Die Daten sind dabei in Messpunkten organisiert. Ein Messpunkt besteht aus einem Zeitstempel und den zu speichernden Daten.
Pro Messpunkt wird die ID aus Abschnitt 3.3.1 gespeichert. Außerdem wird folgende Auswahl an Werten aus Abb. 1 gespeichert: gridPower, homePower, pvPower, residualPower und die Summe aller chargePower-Werte.

3.4.2 Bestimmen der Überschussleistung

Zur Bestimmung der Überschussleistung wird der Messwert gridPower, welcher über einen API-Request auf /api/state von EVCC zur Verfügung gestellt wird, ausgewertet. Dieser

Wert gibt die Differenz zwischen verbrauchter und erzeugter Leistung für einen Haushalt an:

$$gridPower = P_{verbraucht} - P_{erzeugt} \tag{1}$$

Ist dieser Messwert positiv, gibt er an, dass Leistung vom Stromnetz bezogen werden muss, um den Leistungsbedarf des Haushalts zu decken (Netzbezug). Ist der Messwert negativ, bedeutet dies, dass Leistung in das Stromnetz eingespeist wird, da die erzeugte Leistung (von diesem Haushalt) nicht vollständig verbraucht werden kann (Überschuss). Es wird die Summe aller gridPower-Messwerte gebildet. Abhängig vom Vorzeichen beschreibt diese Summe, wie groß die Überschuss- bzw. die Netzbezugsleistung innerhalb der EEG ist.

3.4.3 Steuerung der Ladeleistung

Die zuvor ermittelte Überschussleistung kann verbraucht werden, indem die Ladeleistung von Elektrofahrzeugen erhöht wird. Die Regelung der Ladeleistung pro Haushalt wird durch EVCC realisiert. Die Regelgröße gridPower wird dabei standardmäßig auf die konstante Führungsgröße 0 W geregelt. Durch den Parameter residualPower kann diese Führungsgröße beeinflusst werden. Dies ermöglicht eine externe Steuerung der Ladeleistung.
Ist ein Überschuss von beispielsweise 1000 W innerhalb der EEG vorhanden, kann bei einem Haushalt der Wert residualPower um 1000 W vermindert werden. Somit nutzt dieser Haushalt (unter gewissen Voraussetzungen) 1000 W mehr zum Laden von Elektrofahrzeugen.
Bei der Steuerung der Ladeleistung muss außerdem beachtet werden, dass die zugeteilte Leistung wieder freigegeben werden muss, wenn innerhalb der EEG keine Überschussleistung mehr vorhanden ist. Würde in dem vorher gegebenen Beispiel keine Überschussleistung mehr verfügbar sein, so müsste der Wert residualPower um 1000 W erhöht werden.
Wie vorhin beschrieben müssen bestimmte Voraussetzungen erfüllt sein, damit ein Haushalt die mit residualPower zugeteilte Leistung tatsächlich nutzt. So darf die zugeteilte Leistung die maximal verbleibende zuschaltbare Leistung nicht überschreiten. Um dies zu berücksichtigen, wird vor der Steuerung die zuschaltbare Leistung pro Haushalt ermittelt.

3.4.4 Verteilung der Überschussleistung

Die Verteilung der Überschussleistung funktioniert nach dem Prinzip First Come First Serve. Es wird durch eine Liste aller EEG Teilnehmer iteriert. Unter Berücksichtigung der

vorhandenen Überschussleistung wird jedem Haushalt die maximal zuschaltbare Leistung zugeteilt.

3.5 Simulation einer Energiegemeinschaft

Um die Funktionalität des entwickelten Systems zu testen, wurde ein Programm entwickelt, welches das Verhalten eines Haushaltes simulieren kann. Ein simulierter Haushalt besteht dabei aus einem Netzanschlusszähler, einer Wallbox und einem Fahrzeug. Für diese Geräte spezifiziert EVCC Schnittstellen (siehe [13]). Das Simulationsprogramm implementiert diese Schnittstellen.
Die Interprozesskommunikation zwischen dem Simulationsprogramm und EVCC wird mithilfe des HTTP-Protokolls realisiert. Dazu implementiert das Simulationsprogramm zusätzlich einen HTTP-Server, mit welchen die von EVCC vorgeschriebenen Funktionalitäten über ein REST-Interface genutzt werden können. Mithilfe von HTTP-Plugins kann EVCC dieses REST-Interface nutzen, um mit den simulierten Geräten zu interagieren.
Es wird eine Energiegemeinschaft mit zwei Haushältern simuliert. Die Parameter der Simulation sind folgende:

Anzahl Fahrzeuge	1
Mindestladeleistung pro Fahrzeug	1,38 kW
Maximalladeleistung pro Fahrzeug	7,36 kW
Erzeugte Leistung	$P(t) = 0\,\mathrm{kW}$
Verbrauchte Leistung exklusive Ladeleistung	$P(t) = 0\,\mathrm{kW}$

Tabelle 2: Simulationsparameter für Haushalt 0

Anzahl Fahrzeuge	1
Mindestladeleistung pro Fahrzeug	1,38 kW
Maximalladeleistung pro Fahrzeug	7,36 kW
Erzeugte Leistung	$P(t) = 12\,\mathrm{kW}\sin^2(\frac{2\pi t}{500\,\mathrm{s}})$
Verbrauchte Leistung exklusive Ladeleistung	$P(t) = 2{,}6\,\mathrm{kW}$

Tabelle 3: Simulationsparameter für Haushalt 1

3.6 Ergebnisse der Simulation

Zunächst wird die Simulation ausgeführt, ohne die Ladeleistung zu steuern. In Abb. 8 ist zu sehen, dass zu Beginn mehr Leistung in der EEG verbraucht als produziert wird.

Zu dieser Zeit beträgt die Ladeleistung 0 W (Abb. 9). Die Überschussleistung steigt, bis die Überschussleistung gleich groß wie die minimale Ladeleistung ist. Ab diesem Zeitpunkt wird die Ladeleistung so geregelt, dass die Überschussleistung in etwa 0 W beträgt. Nach dem Erreichen der maximalen Ladeleistung des Haushalts 0, bleibt die Ladeleistung konstant. Da die produzierte Leistung weiter wächst, steigt die Überschussleistung. Diese Leistung könnte von Haushalt 1 genutzt werden. Dies geschieht allerdings nicht, da in diesen Fall die Ladeleistung noch nicht gesteuert wird.

Anschließend wird die Simulation nochmals ausgeführt, wobei diesmal die Ladeleistung gesteuert wird. Der in Abb. 10 gezeigte Leistungsverlauf ist sehr ähnlich mit dem zuvor betrachteten Verlauf in Abb. 8. Der wesentliche Unterschied besteht darin, dass nun die überschüssige Leistung zu jeder Zeit auf etwa 0 W geregelt wird. In Vergleich zu Abb. 9 ist in Abb. 11 erkennbar, dass nun auch der Haushalt 1 Leistung für das Laden von Elektrofahrzeugen nutzt. Dies geschieht, nachdem Haushalt 0 die maximale Ladeleistung erreicht hat und die Überschussleistung innerhalb der EEG größer als die Mindestladeleistung für das Elektrofahrzeug in Haushalt 1 ist.

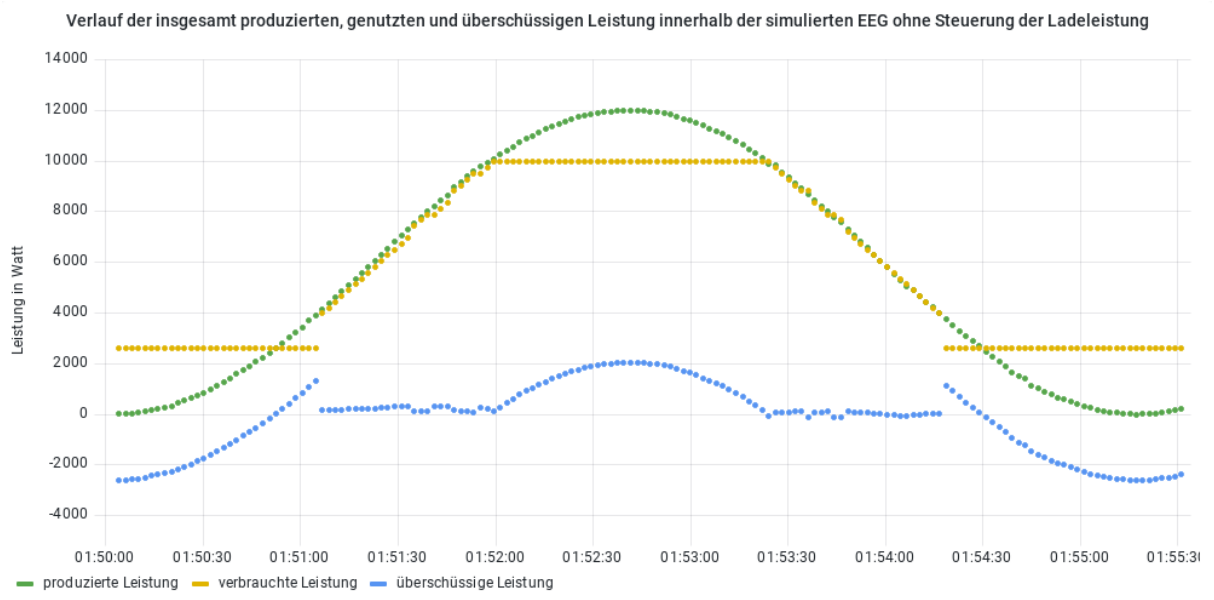

Abbildung 8: Verlauf der insgesamt produzierten, genutzten, und überschüssigen Leistung innerhalb der simulierten EEG ohne Steuerung der Ladeleistung.

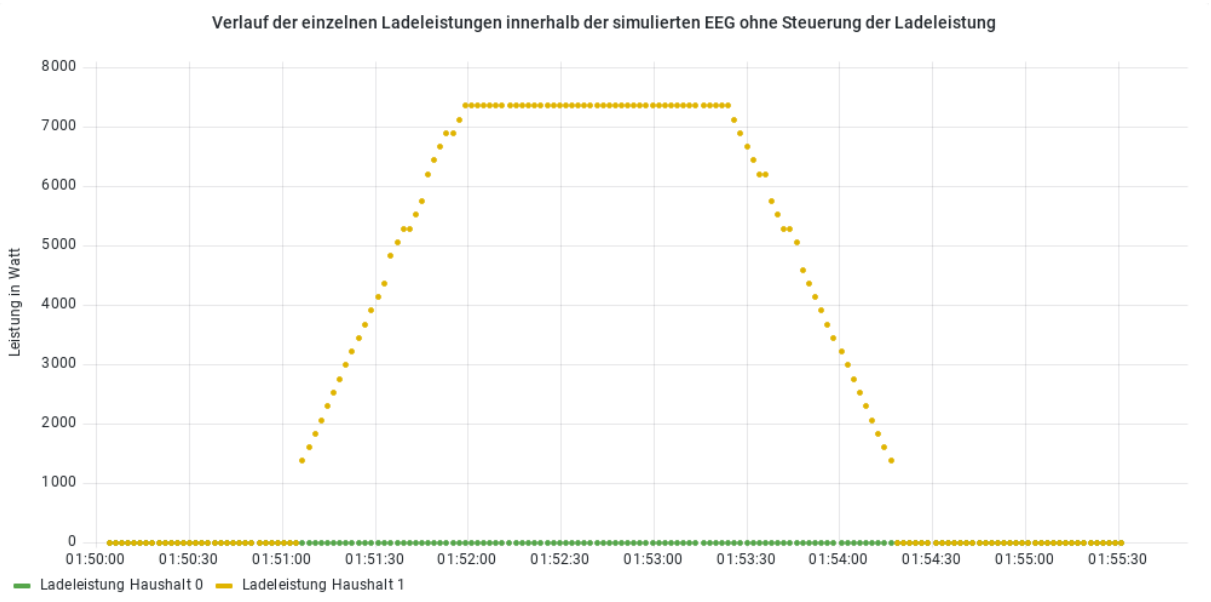

Abbildung 9: Verlauf der einzelnen Ladeleistungen innerhalb der simulierten EEG ohne Steuerung der Ladeleistung.

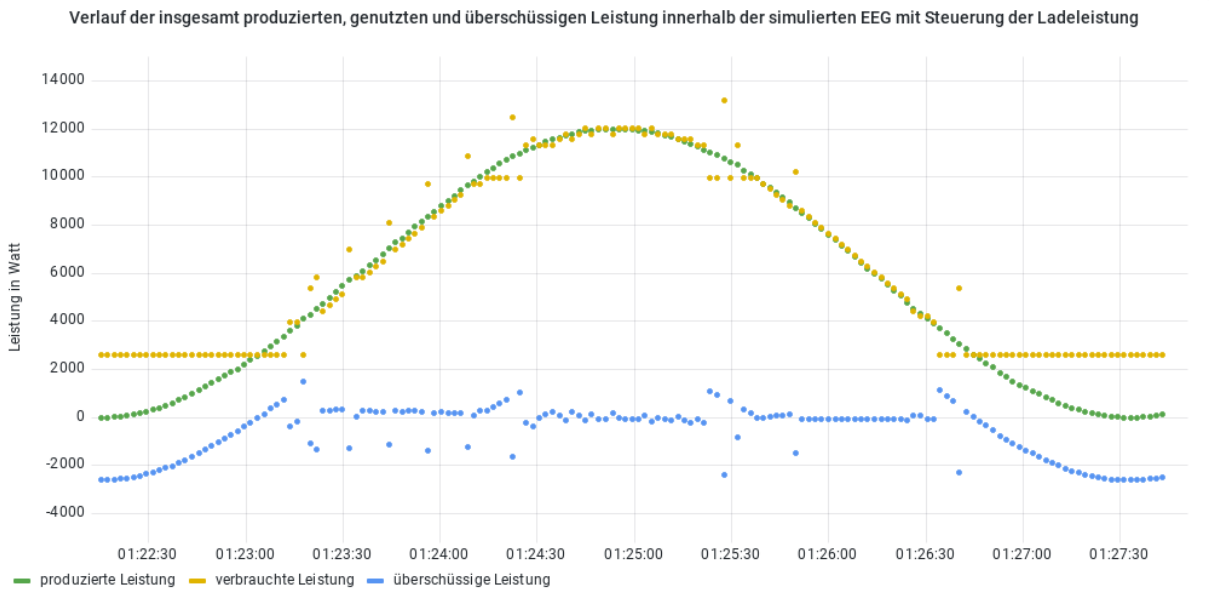

Abbildung 10: Verlauf der insgesamt produzierten, genutzten, und überschüssigen Leistung innerhalb der simulierten EEG mit Steuerung der Ladeleistung.

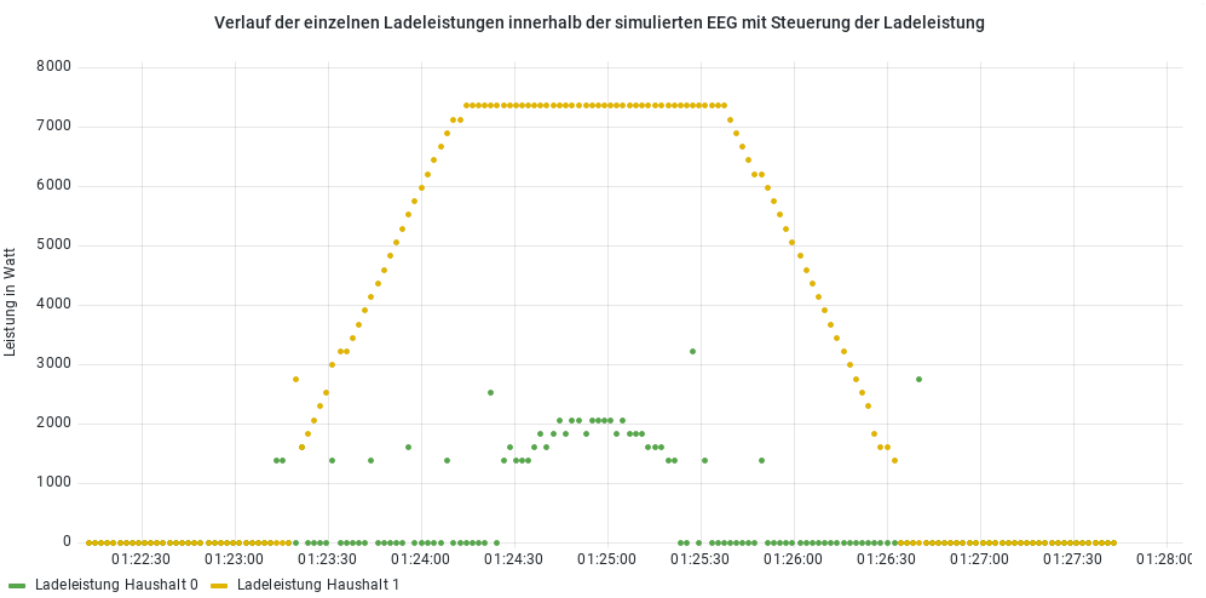

Abbildung 11: Verlauf der einzelnen Ladeleistungen innerhalb der simulierten EEG mit Steuerung der Ladeleistung.

4 Conclusio

In der vorliegenden Bachelorarbeit wurde untersucht, wie überschüssige Energie aus erneuerbaren Energiequellen (bspw. Photovoltaikanlagen) innerhalb von Energiegemeinschaften verteilt werden kann. Dies wurde erreicht, indem alle Energiemesswerte an einem zentralen Punkt gesammelt wurden und infolge dessen die Ladeleistung einzelner Wallboxen in Abhängigkeit der erzeugten Energie reguliert wurde. Das entwickelte System zeigte sich als stabil und effektiv. So konnte in einer simulierten Energiegemeinschaft, in welcher die Open Source Software EVCC in jedem Haushalt zum Einsatz kam, ein annähernder Ausgleich von erzeugter und verbrauchter Energie erreicht werden.
Um das System in einem Salzburger Haushalt verwenden zu können, war die Anbindung eines Smart-Meters der Salzburg AG nötig. Durch den Einsatz eines ESP32 als M-Bus Konverter und einem selbst entwickelten MQTT-Formatters konnten Messwerte an EVCC weitergeleitet werden.
Als Erweiterung zum entwickelten System könnte eine Strategie eingeführt werden, erzeugte Überschussenergie, welche innerhalb der EEG verbraucht wird, zu vergüten. Diese Strategie geht mit dem Algorithmus zur Aufteilung der Überschussenergie einher. Eine Anpassung des First Come First Served Algorithmus an das Tarifmodell des jeweiligen Teilnehmers würde eine wesentliche Verbesserung bringen.
Ebenso sind hinsichtlich der Sicherheit noch Verbesserungen möglich. Aufgrund der Kommunikationsstruktur zwischen EVCC und dem Überschuss-Verteiler ist derzeit die Kommunikation innerhalb eines privaten Netzwerkes zwingend erforderlich. Durch das aktive Versenden von Messdaten von EVCC könnte der Überschuss-Verteiler global betrieben werden, während die EVCC-Instanzen in privaten LAN-Netzwerken sicher vor Angriffen sind.

Literaturverzeichnis

[1] *Erneuerbaren-Ausbau-Gesetz §79*, 14. Feb. 2022. Adresse: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20011619&FassungVom=2022-02-14 (besucht am 01. 12. 2022).

[2] *Elektrizitätswirtschafts- und -organisationsgesetz 2010 §16d*, 1. Dez. 2022. Adresse: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20007045 (besucht am 01. 12. 2022).

[3] O. Okpako u. a., »Optimization of Community Based Virtual Power Plant with Embedded Storage and Renewable Generation,« in *Wireless and Satellite Systems*, I. Otung u. a., Hrsg., Cham: Springer International Publishing, 2017, S. 95–107, ISBN: 978-3-319-53850-1. DOI: 10.1007/978-3-319-53850-1_11.

[4] »EVCC Dokumentation.« (2022), Adresse: https://docs.evcc.io/docs/Home (besucht am 03. 12. 2022).

[5] »Spezifikation MQTT Version 5.0.« (2019), Adresse: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf (besucht am 03. 12. 2022).

[6] »Kommunikationssysteme für Zähler - Teil 1: Datenaustausch.« (2022), Adresse: https://lesesaal-austrian-standards-at.ezproxy.fh-salzburg.ac.at/action/de/private/details/717699/OENORM_EN_13757-1_2022_07_01 (besucht am 01. 12. 2022).

[7] »Kommunikationssysteme für Zähler - Teil 2: Drahtgebundene M-Bus-Kommunikation.« (2018), Adresse: https://lesesaal-austrian-standards-at.ezproxy.fh-salzburg.ac.at/action/de/private/details/633456/OENORM_EN_13757-2_2018_06_01 (besucht am 01. 12. 2022).

[8] »Technische Beschreibung Kundenschnittstelle Smart Meter,« Salzburg AG. (1. März 2022), Adresse: https://www.salzburgnetz.at/content/dam/salzburgnetz/dokumente/stromnetz/Technische-Beschreibung-Kundenschnittstelle.pdf (besucht am 01. 12. 2022).

[9] N. Munzke, »Im Großen wie im Kleinen–Wirtschaftlichkeit von stationä-ren PV-Speichersystemen nun auch bei Mehrfamilienhäu-sern?!« In *31. Symposium Photovoltaische Solarenergie, Kloster Banz*, März 2016.

[10] »InfluxDB Dokumentation.« (2022), Adresse: https://docs.influxdata.com/influxdb/v2.6/ (besucht am 10. 01. 2023).

[11] »Grafana Dokumentation.« (2022), Adresse: `https://grafana.com/docs/` (besucht am 10. 01. 2023).

[12] »ESP32 Datenblatt.« (2022), Adresse: `https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf` (besucht am 10. 01. 2023).

[13] »EVCC Programmierschnittstelle.« (2022), Adresse: `https://github.com/evcc-io/evcc/blob/master/api/api.go` (besucht am 10. 01. 2023).