

Projektzusammenfassung: EV-LKW-Lade-Microgrid

Phase 1

Jules (Software-Ingenieur)

13. August 2025

Zusammenfassung

Dieses Dokument fasst die erste Phase des Projekts „EV-LKW-Lade-Microgrid“ zusammen. Das Hauptziel dieser Phase war der Entwurf und die Entwicklung einer übergeordneten On-Site-Anwendung zur Optimierung der Energieflüsse, mit einem Fokus auf der Integration von On-Site-Energiemessungen. Das Projekt lieferte erfolgreich eine robuste und erweiterbare Anwendung, die Echtzeit-Sensorintegration, Edge-Steuerung für intelligente Entscheidungsfindung und eine transparente Visualisierung der Energieflüsse ermöglicht. Diese Ergebnisse tragen direkt zur Verbesserung der betrieblichen Effizienz und zur Senkung der Energiekosten für Siemens Energy bei.

Inhaltsverzeichnis

1	Einleitung	2
2	Systemarchitektur und Design	2
2.1	Sensor- & Zähler-Integration	2
2.2	Implementierung der Edge-Steuerung	3
2.3	Visualisierung des Energieflusses	4
3	Tests und Validierung	4
4	Geschäftlicher Mehrwert	4
5	Fazit und nächste Schritte	5

1 Einleitung

Der Übergang zu Flotten von Elektrofahrzeugen (EV), insbesondere bei schweren Lastkraftwagen, stellt eine erhebliche Herausforderung für die Energieinfrastruktur dar. Der hohe Leistungsbedarf für das Laden kann die lokalen Netze belasten, was zu Instabilität und erhöhten Kosten führt. Um diesem Problem zu begegnen, hat Siemens Energy das Projekt „EV-LKW-Lade-Microgrid“ initiiert, um eine Lösung für ein intelligentes On-Site-Energiemanagement zu entwickeln.

Phase 1 dieses Projekts, das Thema dieses Berichts, konzentrierte sich auf das Design und die Entwicklung einer grundlegenden Softwareanwendung. Die Hauptziele für diese Phase waren:

- **High-Level-Anwendungsdesign:** Schaffung einer erweiterbaren und robusten Softwarearchitektur zur Optimierung der Energieflüsse innerhalb eines Microgrids.
- **Fokus auf On-Site-Messungen:** Integration mit On-Site-Hardware, einschließlich EV-Ladegeräten, Batterien und dem öffentlichen Netz, um Echtzeit-Energiedaten zu sammeln.
- **Edge-Steuerungslogik:** Implementierung eines lokalen Steuerungssystems, das in der Lage ist, Echtzeitentscheidungen zu treffen, um einen zuverlässigen und effizienten Betrieb zu gewährleisten.
- **Systemtransparenz:** Bereitstellung eines Live-Dashboards zur Überwachung der Energieflüsse und des Systemstatus.
- **Wissenstransfer:** Angebot von fortlaufenden SDK-Schulungen für Mitarbeiter von Siemens Energy, um den langfristigen Projekterfolg und die Wartbarkeit zu sichern.

Dieser Bericht beschreibt die Architektur, Implementierung und Ergebnisse der Phase 1 und zeigt die erfolgreiche Erreichung dieser Ziele auf.

2 Systemarchitektur und Design

Die Anwendung basiert auf dem Frequenz SDK, das ein robustes Framework für die Entwicklung von Energiemanagement-Anwendungen bietet. Die Architektur basiert auf dem Aktor-Modell, einem Paradigma, das die Entwicklung von nebenläufigen und verteilten Systemen erleichtert. Jede Komponente des Systems ist ein „Aktor“, der unabhängig läuft und mit anderen Akteuren über asynchronen Nachrichtenaustausch kommuniziert. Dieses Design gewährleistet eine hohe Verfügbarkeit und Skalierbarkeit.

Das Herzstück der Anwendung ist der **TruckChargingActor**, der als zentraler Orchestrator dient. Er ist verantwortlich für:

- Das Abonnieren von Echtzeit-Datenströmen von verschiedenen Microgrid-Komponenten.
- Die Verarbeitung der eingehenden Daten und die Aufrechterhaltung des Systemzustands.
- Das Aufrufen der Steuerungslogik, um Entscheidungen auf der Grundlage des aktuellen Zustands zu treffen.
- Das Melden von Schlüsselmetriken an eine Zeitreihendatenbank zur Überwachung und Visualisierung.

Diese modulare, aktorbasierte Architektur macht das System hochgradig erweiterbar, sodass neue Funktionalitäten und Steuerungsstrategien mit minimalen Änderungen am bestehenden Code hinzugefügt werden können.

2.1 Sensor- & Zähler-Integration

Ein zentrales Ergebnis der Phase 1 war die erfolgreiche Integration mit On-Site-Energiemessgeräten. Die Anwendung nutzt das Frequenz SDK, um Verbindungen zu verschiedenen Microgrid-Komponenten herzustellen und deren Echtzeit-Datenströme zu abonnieren.

Der **TruckChargingActor** initialisiert und verwaltet Abonnements für die folgenden Datenquellen:

- **Netzleistung:** Die Hauptleistungszufuhr aus dem öffentlichen Netz, bezogen über `microgrid.grid().power`.
- **Batterie-Pool:** Die aggregierte Leistung und der Ladezustand (SoC) der On-Site-Batterien, verwaltet durch `microgrid.new_battery_pool()`.
- **EV-Ladegerät-Pool:** Der kombinierte Stromverbrauch aller EV-Ladegeräte, zugänglich über `microgrid.new_ev_char`.
- **Erzeugerleistung:** Die von On-Site-Anlagen wie Solarmodulen erzeugte Leistung, verfügbar über `microgrid.producer().power`.

Der Akteur verwendet eine asynchrone `select`-Anweisung, um effizient auf alle diese Datenströme gleichzeitig zu lauschen. Wenn eine neue Messung empfangen wird, aktualisiert der Akteur seinen internen Zustand und stellt so sicher, dass die Steuerungslogik immer mit den aktuellsten Daten arbeitet. Diese Echtzeit-Datenerfassung ist die Grundlage für die intelligenten Entscheidungsfähigkeiten des Systems.

2.2 Implementierung der Edge-Steuerung

Die Kernintelligenz der Anwendung liegt in der Klasse `TcControlLogic`, welche die Edge-Steuerungslogik implementiert. Diese Komponente ist dafür verantwortlich, Echtzeitentscheidungen zu treffen, um die Netzstabilität zu gewährleisten und den Energieverbrauch zu optimieren. Die Steuerungslogik wird immer dann ausgeführt, wenn eine neue Netzleistungsmessung empfangen wird.

Das Hauptziel der Steuerungslogik ist es, den Stromverbrauch aus dem Netz unter einem vordefinierten Zielwert (z.B. 25 kW) zu halten. Die Logik folgt diesen Schritten:

1. **Prüfung auf Überverbrauch:** Wenn die aktuelle Netzleistung den Zielwert überschreitet, ergreift das System korrigierende Maßnahmen.
 - Wenn die Batterie ausreichend geladen ist ($\text{SoC} > 10\%$), wird die Batterie angewiesen, zu entladen, um den überschüssigen Leistungsbedarf zu decken.
 - Wenn die Batterie erschöpft oder nicht verfügbar ist, wird die Stromversorgung der EV-Ladegeräte gedrosselt, um die Netzleistung wieder in den Zielbereich zu bringen.
2. **Prüfung auf überschüssige Leistung:** Wenn die Netzleistung negativ ist (d.h. Strom wird exportiert), deutet dies auf einen Überschuss an On-Site-Erzeugung hin. In diesem Fall wird die Batterie angewiesen, mit diesem überschüssigen Strom zu laden.
3. **Keine Aktion:** Wenn sich die Netzleistung im Zielbereich befindet, wird keine Steuerungsaktion durchgeführt.

Der folgende Code-Ausschnitt aus der Methode `TcControlLogic.run` veranschaulicht die Logik zur Behandlung von Überverbrauch:

```

1 # latest_grid_power, latest_battery_soc are provided as arguments
2 if latest_grid_power > self._target_power:
3     if latest_battery_soc > self._min_soc:
4         # Discharge Battery & Restrict EV
5         residual_power = (latest_grid_power + latest_battery_power) - self._target_power
6         await self._battery_pool.propose_power(residual_power)
7     else:
8         # Restrict EV Power only
9         residual_power = latest_grid_power - self._target_power
10        # await ev_charger_pool.propose_power(residual_power)

```

Listing 1: Steuerungslogik zur Behandlung von Netzüberlastung.

Diese Logik stellt sicher, dass das Microgrid effizient arbeitet und hohe Kosten für Spitzenlastbezug aus dem Netz vermieden werden.

2.3 Visualisierung des Energieflusses

Um Transparenz zu schaffen und eine Echtzeit-Überwachung der Leistung des Microgrids zu ermöglichen, enthält die Anwendung eine Komponente zur Visualisierung des Energieflusses. Dies wird durch die Integration von InfluxDB, einer Zeitreihendatenbank, und Grafana, einem beliebten Datenvisualisierungstool, erreicht.

Die Klasse `InfluxReporter` ist dafür verantwortlich, Schlüsselmetriken aus der Anwendung zu sammeln und an eine InfluxDB-Instanz zu melden. Die folgenden Metriken werden gemeldet:

- Netzleistung (Watt)
- Batterieleistung (Watt)
- Batterieladezustand (SoC)
- EV-Ladegerätleistung (Watt)
- Erzeugerleistung (Watt)

Diese Metriken werden mit einem Zeitstempel versehen und in der Datenbank gespeichert, wodurch ein historischer Verlauf des Systembetriebs erstellt wird. Ein vorkonfiguriertes Grafana-Dashboard verbindet sich mit dieser Datenbank, um Live-Visualisierungen dieser Metriken bereitzustellen. Dieses Dashboard ermöglicht es den Betreibern, den Zustand des Systems zu überwachen, Energieflussmuster zu verstehen und zu überprüfen, ob die Steuerungslogik wie erwartet funktioniert.

Die `README.md`-Datei des Repositorys enthält detaillierte Anweisungen zum Einrichten der Grafana- und InfluxDB-Dienste, was eine einfache Bereitstellung der Überwachungslösung ermöglicht.

3 Tests und Validierung

Der Projektumfang für Phase 1 betonte die Entwicklung einer erweiterbaren Anwendung mit einer Grundlage für robuste Software-Unit- und Integrationstests. Die modulare Architektur mit ihrer klaren Trennung der Verantwortlichkeiten zwischen dem Haupt-Aktor, der Steuerungslogik und der Datenberichterstattung ist so konzipiert, dass sie leicht testbar ist.

Obwohl umfassende Unit-Tests für die `TcControlLogic` ein wichtiger nächster Schritt sind, enthält das Repository ein Beispiel für einen fortgeschrittenen, systemweiten Testfall: `TestDynamicConditionOnGrid`. Dieser Test ist für den Betrieb in einer Hardware-in-the-Loop (HiL)-Simulationsumgebung ausgelegt und validiert die Reaktion des Systems auf dynamische Änderungen der Netzbedingungen, wie z.B. allmähliche oder sprunghafte Laständerungen.

Dieser bestehende Testfall dient zwei Zwecken:

- Er bietet ein Framework für fortgeschrittenere, systemweite Validierungen in zukünftigen Projektphasen.
- Er demonstriert die Fähigkeit der Anwendung, sich in eine größere Test- und Simulationsumgebung zu integrieren.

Zukünftige Arbeiten werden die Entwicklung einer vollständigen Suite von Unit-Tests umfassen, um die Korrektheit der Steuerungslogik unter verschiedenen Szenarien zu überprüfen, sowie die Erweiterung der systemweiten Tests zur Abdeckung komplexerer Anwendungsfälle.

4 Geschäftlicher Mehrwert

Die in Phase 1 gelieferte Anwendung für das EV-LKW-Lade-Microgrid bietet Siemens Energy einen erheblichen geschäftlichen Mehrwert, indem sie zentrale betriebliche und finanzielle Herausforderungen im Zusammenhang mit dem groß angelegten Laden von Elektrofahrzeugen adressiert.

Die primären geschäftlichen Vorteile umfassen:

- **Verbesserte betriebliche Effizienz:** Die automatisierte Steuerungslogik stellt sicher, dass das Microgrid optimal ohne manuelle Eingriffe arbeitet. Das System gleicht intelligent Stromangebot und -nachfrage aus und stellt sicher, dass die EV-Ladegeräte bei Bedarf verfügbar sind, während das lokale Netz vor übermäßiger Belastung geschützt wird.
- **Reduzierte Energiekosten:** Durch die aktive Steuerung des Energieflusses hilft die Anwendung, die Energiekosten zu senken. Sie minimiert den Verbrauch aus dem Netz während der Spitzenlastzeiten durch die Nutzung von On-Site-Batteriespeichern und kann hohe Lastspitzenentgelte vermeiden, indem sie bei Bedarf nicht wesentliche Lasten drosselt.
- **Erhöhte Zuverlässigkeit und Stabilität:** Die Edge-Steuerungslogik bietet eine schnell reagierende, lokalisierte Antwort auf sich ändernde Bedingungen, was die Zuverlässigkeit und Stabilität der Ladeinfrastruktur erhöht. Dies ist entscheidend, um sicherzustellen, dass die EV-LKW-Flotte betriebsbereit bleibt.
- **Zukunftssichere, skalierbare Lösung:** Die erweiterbare Architektur ermöglicht die Integration neuer Anlagen, komplexerer Steuerungsstrategien und die Teilnahme an zukünftigen Energiemärkten, was sie zu einer wertvollen langfristigen Investition macht.

5 Fazit und nächste Schritte

Phase 1 des Projekts „EV-LKW-Lade-Microgrid“ hat erfolgreich eine grundlegende Anwendung zur On-Site-Energieoptimierung geliefert. Das Projekt hat alle seine Hauptziele erreicht, einschließlich der Integration von On-Site-Sensoren, der Implementierung eines Echtzeit-Edge-Steuerungssystems und der Bereitstellung eines Live-Visualisierungs-Dashboards. Die resultierende Anwendung ist robust, erweiterbar und bietet durch die Verbesserung der Effizienz und die Senkung der Kosten einen sofortigen geschäftlichen Mehrwert.

Die folgenden nächsten Schritte werden für zukünftige Projektphasen empfohlen:

- **Entwicklung umfassender Unit-Tests:** Erstellung einer vollständigen Suite von Unit-Tests für die TcControlLogic, um deren Korrektheit und Zuverlässigkeit unter einer Vielzahl von Bedingungen sicherzustellen.
- **Verbesserung der Steuerungsstrategien:** Implementierung anspruchsvollerer Steuerungsalgorithmen, die Faktoren wie Energiepreise, Demand-Response-Signale und Wettervorhersagen berücksichtigen können, um den Energieverbrauch weiter zu optimieren.
- **Erweiterung des Hardware-in-the-Loop (HiL)-Testings:** Aufbauend auf dem bestehenden Testfall TestDynamicConditionOnGrid, um eine umfassende HiL-Testsuite zu erstellen, die die Leistung des Systems in einer simulierten, realitätsnahen Umgebung validiert.
- **Bereitstellung und Feldtests:** Einsatz der Anwendung in einer Live-Produktionsumgebung, um reale Leistungsdaten zu sammeln und die Steuerungslogik basierend auf betrieblichen Erfahrungen zu verfeinern.

Der erfolgreiche Abschluss der Phase 1 hat eine starke Grundlage für die weitere Entwicklung und den Einsatz dieser innovativen Energiemanagementlösung gelegt.