



Smart-Meter-Toolkit

Open-Source-Lösung für Endkunden | Gemäss Stromversorgungsverordnung muss ein Netzbetreiber seinen Endkunden ermöglichen, Echtzeitmessdaten des Smart Meters zu erhalten. Das Smart-Meter-Toolkit ist eine schlanke, kostengünstige Open-Source-Lösung, die dies und beispielsweise die Einbindung der Messdaten in Heimautomatisierungsprozesse ermöglicht.

MARIO CORAY

Immer mehr Immobilien von Endkunden wurden in den letzten Jahren mit Smart Metern ausgerüstet, die den Stromverbrauch nicht nur messen, sondern die Werte auch regelmässig an den Netzbetreiber übermitteln. Typischerweise geschieht dies einmal pro Tag oder seltener. Endkunden können diese Daten meist über eine Weboberfläche rückwirkend einsehen. Für eine detailliertere Analyse oder gar Einbindung der Messdaten in Heimautomatisierungsprozesse ist dies jedoch unzureichend. Deshalb beinhaltet Art. 8a der Stromversorgungsverordnung eine Bestimmung, dass Endverbraucher Messdaten direkt

und lokal über eine Kundenschnittstelle am Smart Meter empfangen können. So lässt sich der aktuelle Stromverbrauch praktisch in Echtzeit mit einer Auflösung von 5 bis 15 s erfassen.

Kundenschnittstelle

Alle gängigen Smart Meter verfügen über eine Kundenschnittstelle, auch «Consumer Information Interface» (CII) genannt, die sich zu diesem Zweck einsetzen lässt. Nun werden für die Datenübertragung leider keine Protokolle eingesetzt, die die Messwerte gemäss Stromversorgungsverordnung in einer verständlich dargestellten Form zeigen. Die hiesigen

Zähler verwenden dafür meist den Standard DLMS/Cosem, der über dem High-Level Data Link Control (HDLC) Netzwerkprotokoll eingesetzt wird. Für den Physical Layer werden je nach Zählermodell der M-Bus- oder der DSMR-P1-Standard verwendet. Zudem sind die Mess-Samples aus Datenschutzgründen oft verschlüsselt und werden fragmentiert übermittelt. Somit werden seitens Smart-Meter-Hersteller zwar internationale Standards eingehalten, für Netzbetreiber wie EKZ bleibt aber die Frage, wie die Kunden gemäss Verordnung die Messdaten «in verständlich dargestellter Form» empfangen können.

Proof of Concept

An den Energy Hackdays im August 2020 sammelte EKZ dazu erste Erfahrungen. Ein Team entwickelte einen Proof of Concept (PoC), um mit zum Teil offenen, aber auch proprietären Softwarekomponenten die Smart Meter über die Kundenschnittstelle auszulesen und die Daten zu visualisieren. Daraufhin beauftragte EKZ die Firma Supercomputing Systems AG (SCS), auf der Basis dieses PoCs ein Produkt zu entwickeln, das interessierten Endkunden zur Verfügung gestellt werden kann. Die Lösung musste einfach und kostengünstig für die Endkunden sein, durfte aber auch keine proprietären Komponenten enthalten, da diese schlecht erweiterbar sind und häufig unter einschränkenden Lizenzbedingungen stehen.

Variante mit Raspberry Pi

In der Initialisierungsphase des Projekts wurde evaluiert, ob die definitive Lösung über eine eigene Hardware auf Basis eines Microcontrollers und der entsprechenden M-Bus- oder DMSR-P1-Schnittstelle verfügen soll. Dies hätte jedoch entweder viel Aufwand für die Produktion und den Vertrieb



Installation vor Ort - über WLAN oder Ethernet kann auf das Toolkit zugegriffen werden.

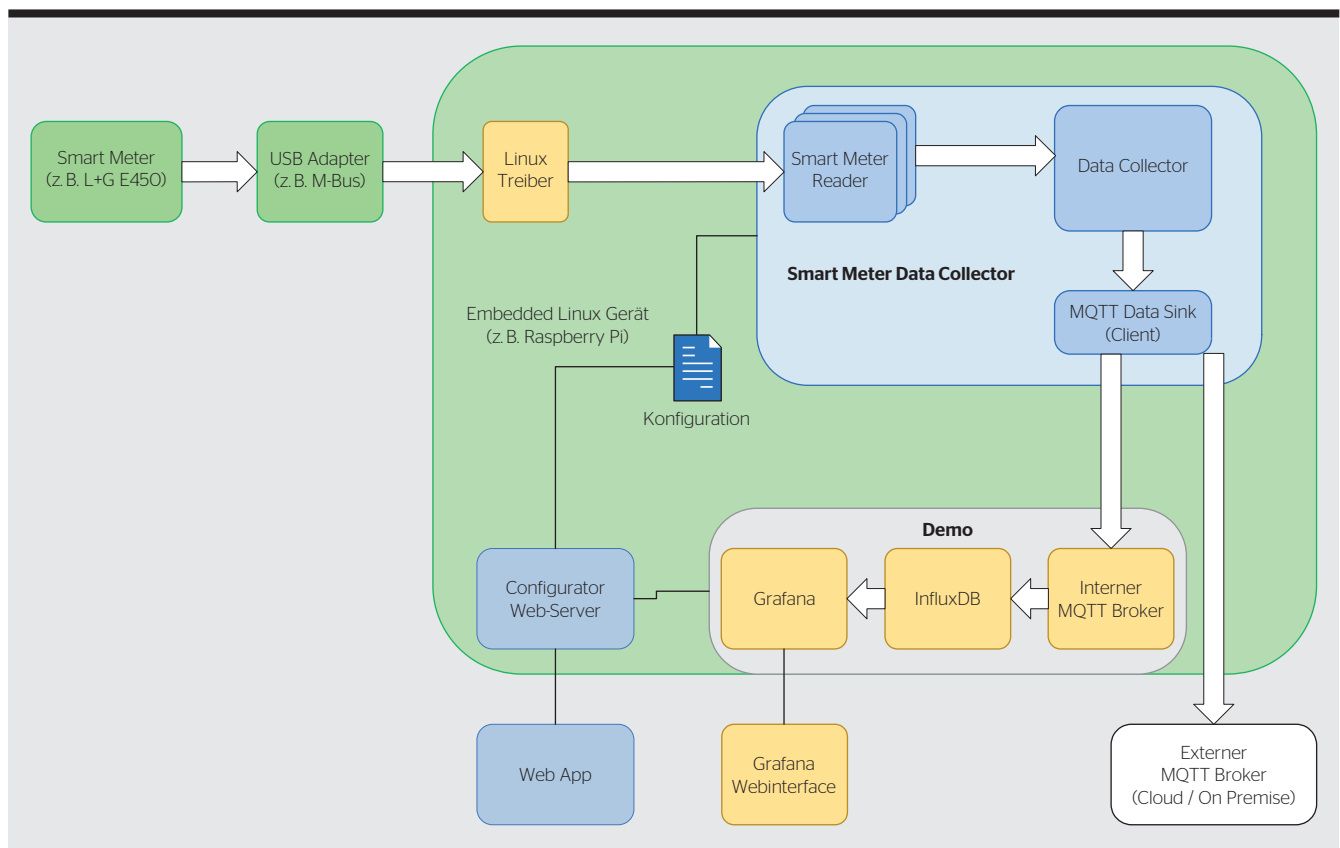
aufseiten EKZ bedeutet oder wäre bei einem Open-Hardware-Projekt eine grosse Hürde für Endkunden, um ihre Hardware selbst zusammenzustellen. Es wurde deshalb eine Variante mit einem Raspberry Pi gewählt, wobei die Endkunden je nach Zählertyp den passenden USB-Adapter bei einem Drittanbieter erwerben können.

Zudem stellte sich die Frage, wie die geparteten Zählerdaten weitergeleitet werden sollten. Hier wurde auf MQTT gesetzt - einem einfachen publish-sub-

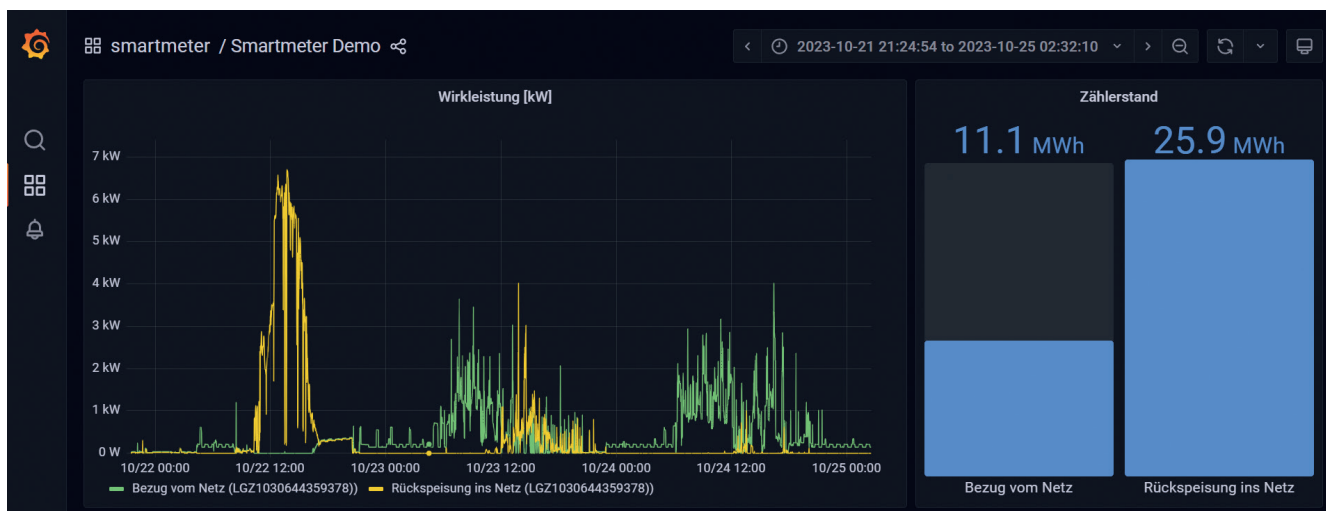
scribe-basierten Netzwerkprotokoll, das beispielsweise im IoT-Bereich verbreitet ist. Es bietet die Flexibilität, die Daten sowohl im Netzwerk als auch lokal auf dem Gerät, auf dem die Zählerdaten empfangen werden, weiterzuverwenden.

Open-Source-Softwarelösung

SCS entwickelte für EKZ eine reine Open-Source-Softwarelösung, die auf Github gehostet wird. Der Fokus lag dabei einerseits auf einem einfachen



Systemübersicht des Smart-Meter-Toolkits.



Grafische Darstellung der Messdaten direkt auf dem Raspberry Pi mit Grafana.

Einsatz der Software, andererseits auf der Zugänglichkeit für versierte Endbenutzerinnen und Endbenutzer, um die Software anzupassen oder weiterzuentwickeln. Deshalb wählte man für die Kernapplikation, den Data Collector, die Programmiersprache Python 3. Der Flexibilitätsgedanke wurde auch in der Architektur dieser Software weitergeführt, indem ein oder mehrere Smart-Meter-Lesegeräte als Datenquellen und eine oder mehrere Datenbanken instanziiert werden können. Eingangsseitig ist es also möglich, mehrere Zähler parallel an das gleiche Raspberry Pi anzuschließen und deren Daten im gleichen Programmprozess an die Datenbanken weiterzuleiten.

Flexible Softwarearchitektur

Als Datenbanken stehen aktuell eine MQTT-Client-Implementierung und ein einfacher Datenlogger ins Linux-Log-Journal zur Verfügung, die wiederum parallel eingesetzt werden können. Es ist aufgrund des erweiterbaren Softwaredesigns einfach möglich, nebst weiteren Zählern auch Datenbanken zu anderen Schnittstellen hinzuzufügen. Um die Messdaten innerhalb der Applikation weiterleiten zu können, ohne die Datenquellen und -senken zu stark miteinander zu koppeln, wurde auf das Observer Pattern zurückgegriffen. Dies passt gut zum asynchronen Ansatz, auch bekannt als `async/await`-Konzept, auf das die Applikation aufbaut. Asyn-

chrone Programmierung wird gerne in I/O-lastigen Applikationen eingesetzt, die wie hier einen Grossteil der Zeit auf Datenübertragung an den Ein- und Ausgängen warten. Umgesetzt wird dies mit pseudoparallelen Tasks, verwaltet in einem Python-eigenen Eventloop, der nur in einem Thread läuft. Tasks, die momentan nichts verarbeiten müssen, sind pausiert und blockieren somit nicht den Rest der Applikation. Aufgrund der Implementierung des Eventloop mit Betriebssystemfunktionen müssen die einzelnen Tasks auch nicht aktiv gepollt werden, was zu einer tiefen CPU-Last der Applikation führt.

Für das Dekodieren der von den Zählern stammenden DLMS-Nach-

RÉSUMÉ

Une « boîte à outils » pour compteurs intelligents

Solution open source pour les clients finaux

Selon l'ordonnance sur l'approvisionnement en électricité, le gestionnaire de réseau doit permettre au client final d'avoir accès aux données de mesure en temps réel générées par son compteur intelligent. Le Smart-Meter-Toolkit – une « boîte à outils » pour compteurs intelligents – est une solution open source simple et peu onéreuse qui permet cet accès ainsi que, par exemple, l'intégration des données de mesure dans des processus de domotique. Cette solution est basée sur un Raspberry Pi, les clients finaux pouvant acheter l'adaptateur USB approprié auprès d'un fournisseur tiers en fonction du type de leur compteur. Les données du compteur analysées sont transmises via le protocole MQTT, très répandu dans le domaine de l'IoT. Il offre la flexibilité de pouvoir réutiliser les données aussi bien sur le réseau que

localement, sur l'appareil sur lequel les données du compteur sont reçues.

Bien que la solution ait été conçue, à la demande d'EKZ, spécifiquement pour les compteurs et la configuration que le gestionnaire de réseau utilise, elle est désormais suffisamment générique pour être utilisée avec les compteurs intelligents d'autres gestionnaires de réseau de distribution. Pour pouvoir recevoir les données de mesure localement, il faut non seulement avoir un compteur intelligent compatible, mais aussi faire activer l'interface client par le gestionnaire de réseau de distribution. Les étapes de la mise en place de la solution, ainsi que d'autres informations, se trouvent, entre autres, sur: github.com/scs/smartmeter-datacollector/wiki.

richten wurde eine umfangreiche, ebenfalls open-source verfügbare Software-Library der finnischen Firma Gurux in die Applikation eingebunden. Sie bietet die nötigen Funktionen, wie das Entschlüsseln und Interpretieren der DLMS-Daten für den Einsatz mit den hierzulande eingesetzten Smart Metern.

Python-Paket und Demo-Image

Den Benutzern stellt SCS nicht nur ein Python-Paket mit der oben beschriebenen Data-Collector-Software auf dem offiziellen Python-Package-Index (PyPi) zur Verfügung, sondern auch ein fertiges Raspberry Pi Demo-Image, das direkt auf eine SD-Karte geflasht werden kann. Mit wenigen Konfigurationsschritten über ein lokales Webinterface des Raspberry Pi ist damit die Visualisierung der Messdaten über Grafana möglich. Weiter werden in dieser Demo die Daten per InfluxDB aufgezeichnet, damit sie rückwirkend über ein paar Monate angezeigt werden können. All dies findet lokal auf dem Raspberry Pi statt – es werden keine Verbraucherdaten in die Cloud geschickt.

In die Heimautomation integrierbar

Ambitionierte Endnutzerinnen und Endnutzer, die ihre Smart-Meter-Messdaten in eine Heimautomatisierungslösung einbinden möchten, können die Data-Collector-Software direkt mit einem externen MQTT-Broker verknüpfen. Da in Python entwickelt, ist es auch möglich, die Software auf einer anderen Linux-Plattform statt auf einem Raspberry Pi laufen zu lassen. Somit ist die Software flexibel einsetzbar.

Auf Github

Als Hosting-Standort für das Projekt wurde Github gewählt, weil es eine gut zugängliche und ausgebaute Plattform für Open-Source-Vorhaben bietet, wo sich User und Entwickler zur Lösung austauschen und sich gegenseitig helfen [1]. SCS pflegt die Community, nimmt Vorschläge und Bugreports entgegen und webt diese wiederum in die Lösung ein. Auch wurde das Projekt schon mehrfach geforkt, wo andere Personen eigene Varianten für ihre spezifischen Anwendungszwecke entworfen haben.

Offen für andere Verteilnetze

Obwohl die Lösung auf Wunsch von EKZ speziell für ihre eingesetzten Zähler und Konfiguration entworfen wurde, ist sie mittlerweile generisch genug, um auch mit Smart Metern von anderen Verteilnetzbetreibern eingesetzt zu werden, wie sich gezeigt hat. Aktuell werden die Zähler E450, E360 und bald E570 von Landis+Gyr, Iskraemeco AM550 und Kamstrup Omnipower mit dem HAN-NVE-Modul unterstützt. Neben einem kompatiblen Smart Meter muss man vom Verteilnetzbetreiber die Kundenschnittstelle freischalten lassen, um die Messdaten lokal empfangen zu können. Dies erreicht man bei EKZ über den Kundensupport, wo man auch eine ausführliche Installationsanleitung erhält. Die Schritte für das Aufsetzen der Lösung sowie weitere Informationen findet man auch im Wiki der Github-Projektseite.

Referenz

[1] github.com/scs/smartmeter-datacollector



Autor

Mario Coray ist Projektleiter bei Supercomputing Systems AG.
→ SCS, 8005 Zürich
→ mario.coray@scs.ch



pronutec AG

Starkstromkomponenten
von den Experten



Composants basse tension
par des experts

pronutec AG
Rosenweg 3
6234 Triengen

041 545 86 70
info@pronutec.ch
www.pronutec.ch

