EEL: OCPP Over Cloud

Robert Brown, Amnah Hussain, Brooke Kindleman, Nicholas Knapton, Anand Patel

Advised by Dr. Huaqing Wu
Schulich School of Engineering

Abstract

Our team has worked alongside FLO, a Canadian Electric Vehicle (EV) charging station manufacturer and network operator on this project. Connectivity of charging stations to network operators is a fundamental component to EV charging stations and networks. Some of FLOs chargers currently utilize a proprietary networks protocol to facilitate communication between the charging stations and network operators. The EV charging industry has adopted the Open Charge Point Protocol (OCPP) which defines a standard protocol for communicating between charging stations and network operators.

FLO has adopted the OCPP standard in some of their newer models, but this leaves older models communicating on an older protocol unless retrofitted. Our project develops a cloud-based solution to allow existing hardware to communicate with network operators using the OCPP standard without any on-site modifications. OCPP over Cloud provides a solution for existing EV charging network operators to utilize a single backend server and protocol to communicate with FLO charging stations. Additionally, this allows for better compatibility with third-party EV charging stations and software solutions for EV charging networks.

Introduction

As the adoption of EVs increases and international adoption targets draw closer, the infrastructure required to power EVs will become more important. Increases in the number of EVs on the road will lead to an increase in the number of EV chargers being installed. Early adopters of EV chargers may be utilizing hardware that can only interact with the manufacturer's own network or require hardware upgrades to communicate with third party servers. A cloud solution to allow for interoperability will allow for EV charge point operators (CPOs) to invest in new charging infrastructure instead of retrofitting old stations. By utilizing the cloud solution and the OCPP protocol for communication, CPOs can utilize the same charging network software resulting in lower development costs by not upgrading existing hardware.

Glossary

Open Charge Point Protocol (OCPP) - An open standard for communicating between EV chargers and EV charging networks.

Charge Point Operator (CPO) - An entity that is operating a public EV charger.

Host Pod – A container hosted in a Kubernetes cluster that is responsible for holding OCPP WebSocket connections between the service and CPO servers.

Router – A container hosted in a Kubernetes cluster that is responsible for receiving and routing messages to host pods.

Monitor – A container hosted in a Kubernetes cluster that is responsible for monitoring resource utilization of host pods and dynamically scaling them up or down based on predefined thresholds.

WebSocket Secure (WSS) Protocol – Network protocol for sending information bi-directionally using an encrypted TLS connection.

Service Bus – A message broker that allows different applications to communicate with each other with message queues and publish-subscribe topics.

Dynamic Scaling – The system automatically adjusts its resources based on the load on the system

Topic – A messaging entity that allows for one-to-many messaging, where all subscribers receive the message sent to the topic.

Cluster – A set of nodes that run containerized applications.

Diagrams

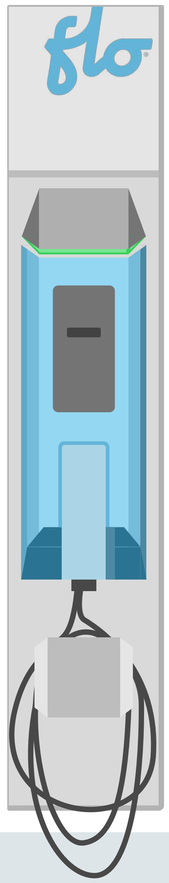

FIGURE 1: SERVICE DESIGN

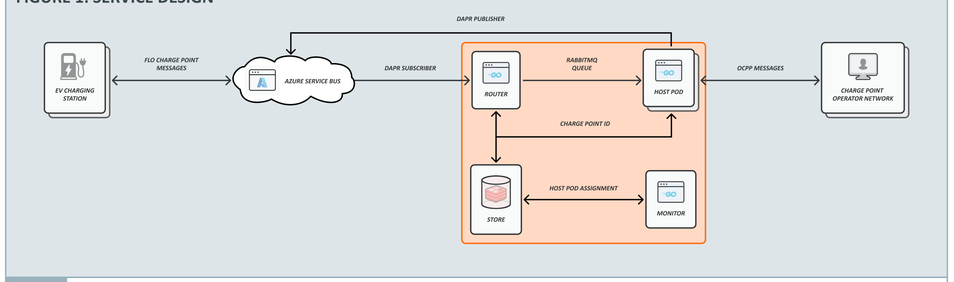

Figure 1: High-level diagram illustrating the information flow through the OCPP Over Cloud service. Components developed by EEL are highlighted in orange.

FIGURE 2: MESSAGE FLOW

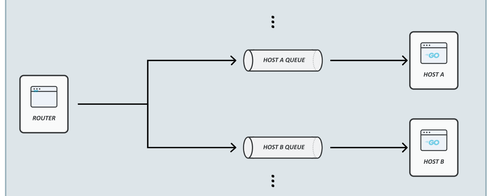

Figure 2: Messages flow from the router to the host pod queues. Messages in the queue until they are held until handled by the EEL host pods.

FIGURE 3: DAPR SUBSCRIBER

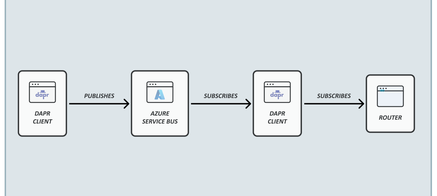

Figure 3: Chain of subscriptions used for by EEL for tracking message events from the routers.

Results

The OCPP over Cloud service developed as part of the project successfully satisfied the specifications outlined in the design phase. The solution can dynamically scale the number of hostpods based on system demand, allowing the service to accommodate up to 40,000 chargers that currently exist within FLOs network. Utilizing Dapr our service can subscribe to messages on an Azure service bus, with flexibility to be configured to use comparable applications on other cloud providers, allowing us to swap them out.

Upon receiving the message (FLO proprietary), the router routes the message to the designated hostpod if one is already assigned to that charger; otherwise, it assigns one based on performance metrics. Subsequently the message is packaged into an OCPP message and dispatched to the CPO network. The system supports the core OCPP 1.6 protocol and can be further expanded to accommodate additional profiles within the 1.6 standard. Most importantly, integration with FLO's systems allows seamless communication between hardware utilizing proprietary FLO protocols and OCPP compliant CPOs.

Conclusion

This project allowed our team to learn more about the inner workings of EV chargers and the networks that CPOs use to operate them. Hands-on experience with technologies such as Dapr, Kubernetes, Azure, RabbitMQ and Redis has developed our skills that are sought after in the industry. By leveraging these technologies, our team was able to deploy and manage our cloud-based solution that is scalable for up to 40,000 chargers.

This cloud-based solution signifies a step towards international adoption of EVs. By eliminating the need for retrofitting old stations, our solution allows for both interpretability and reduction of development costs.

Methods

Distributed Cluster — We are utilizing individual containers instances (pods) for each component of the service. The containers are deployed on an Kubernetes cluster allowing us to manage communication and scaling of the system.

Dynamic Scaling — A monitoring component was implemented to check the utilization of each hostpod and dynamically increase or decrease the number of hostpods currently running. The monitoring service also handles transferring chargers from one hostpod to another.

Service Bus — We are utilizing Dapr to wrap an Azure service bus which is used to subscribe to incoming messages from chargers and route them to the proper hostpod. Responses are published back onto a different topic on the service bus to be sent back to the chargers.

Modern Programming Language — We are primarily using the Golang programming language given its flexibility in cloud environments and high performance in distributed workloads.

Monitoring — We are utilizing New Relic to monitor logs and errors from our system. This allows us to leverage the data processing features provided from New Relic to better understand the large quantity of data sent in from our system.

Cloud Deployment — The service is currently deployed on the Azure cloud platform. By using Dapr to wrap platform dependent services and utilizing Kubernetes our system is cloud agnostic and could be deployed to any of the cloud platforms.

Technologies

References

[1] "Open charge point protocol," Open Charge Alliance, https://openchargealliance.org/protocols/open-charge-point-protocol/.
[2] A. Melnikov and I. Fette, "The WebSocket Protocol," Internet Engineering Task Force (IETF), https://datatracker.ietf.org/doc/html/rfc6455.
[3] spelluru et al., "What is Azure Service Bus?," Microsoft Learn, https://learn.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview.