

Case Study

Solar Tech Company Achieves Efficient Data Transfer to Improve Operations at Solar Farms



A solar technology company aimed to gather a large amount of data from 200+ solar farms to improve operations. The data would be analyzed and used for energy management, to increase productivity, and to improve efficiency across the business. Facilities were remote and data loads were extreme, so the solution needed to handle data with a small, efficient footprint with the ability to send data upon change to reduce the amount of data being sent and improve the speed of data transmission.

Project Goals and Challenges

Not only did the customer have more than 200 solar farms, but they had also more than 100,000 data tags to collect at each site, resulting in more than 2 million total tags to collect. Some data was changing every second. They also had multiple consumers of the data - a control center and cloud services using Apache Kafka for high-performance data pipelines and integration.

The customer was not getting the results they needed collecting data from the solar farms with typical poll/response protocols such as OPC UA. The current system was inefficient for the large amounts of data they were trying to consume - bandwidth usage was extremely high, and the data couldn't be retrieved fast enough to meet their requirements.

The goal was to adopt a solution that could poll for data locally and deliver the data needed rapidly, with context, and with greater efficiency. The customer wanted to reduce bandwidth usage, improve data throughput and reduce processing requirements.

Implementation

The customer chose to build their architecture around open standard tools for a future-proof solution that would keep them from being locked in to any one vendor. They chose the MQTT protocol to transfer large amounts of data with greater efficiency.

Specifically, MQTT could do two things for them:

1. Standardize the data and decouple data for multiple consumers

2. Provide contextualization on the data with MQTT Sparkplug

MQTT is a proven, standard machine-to-machine data transfer protocol that is quickly becoming the leading messaging protocol for Industrial IoT. MQTT was ideal for this use case as a publish/subscribe, extremely simple and lightweight messaging protocol ideal for constrained networks.

Sparkplug B is an open-source software specification that defines how to use MQTT in a mission-critical, real-time environment. The Sparkplug B specification provides the data model needed to define a tag value for use with OT, also providing data to IT, making it 100% self-discoverable and easy to consume. Sparkplug was also ideal for this use case because it has a built-in store and forward mechanism to send data across the network to be consumed not only by the control center but also the cloud service via Kafka.

Kafka is an Apache open-source distributed event streaming platform for high-performance data pipelines and data integration. A Kafka connector subscribes to the MQTT Sparkplug data and enables it for tools like Power BI and Grafana so the customer can do visualizations and more advanced analytics.

They chose the Ignition SCADA platform as the central control system for its connectivity capabilities in unison with Cirrus Link MQTT modules. Ignition connects to all the sensors at the solar farms, then uses the Cirrus Link MQTT modules to transfer that data to a central location and integrate to Kafka so it can then be used to optimize production with machine learning and other advanced analytics.

Results

The new architecture built around MQTT and Sparkplug gave the customer a standardized, efficient platform for data movement. The solution allowed the customer to move data securely and efficiently from the OT layer upstream to Kafka. Data can now be sent by exception, meaning only on change, which improved throughput over the network by more than 80 percent. Instead of sending 3 million tags with a round robin poll they send 5 to 10 percent of that by sending only on change.

The solar technology company also greatly reduced central processing time since the data producers and consumers are now decoupled via MQTT, so the data can go to multiple consumers at once. All the data is also self-discovered automatically following the Sparkplug data format.

The project scaled rapidly, and they rolled the solution out to 150 sites in the first year alone, with the other half set to roll out in year two.