IoT Partner Quickstart

Getting Started with IoT for Google Cloud Partners

Developing IoT Solutions for GCP

Contents

1. Introduction

No longer the stuff of science fiction, IoT is fast becoming mainstream. People want their cars to be smart, their homes to be under voice control, and their appliances to repair themselves. As well as changing everyday life for consumers, IoT has enormous potential to facilitate transformation of entire industries. “Transformation” means not just having robotic arms on an assembly line that can predict their own failure, but also enabling businesses to innovate around new revenue streams. Traditional hardware companies are now evolving into services companies as their customers change consumption patterns.

Transformation such as this depends on combining machine learning (ML) with the ‘things’ of IoT—all manner of devices that make up the infrastructures that support various public and private sector use cases—manufacturing, asset tracking, inventory management, oil rigs, power grid, water system, weather alert system, smart buildings, smart parking, smart cities, and so on.

Furthermore, ML itself depends on vast quantities of data—to train, test, implement, and improve upon the ML model that can be applied to any specific use case.

For example, an ML model that can predict component failures for a self-repairing manufacturing line requires data collected over time from hundreds or thousands of sensing devices, and the transformation of the manufacturing process might include not only rerouting the manufacturing line, but pulling the needed component from the warehouse and putting it on a conveyor belt that makes its way to the maintenance crew just as they’re receiving an alert that the component needs to be replaced.

Figure 1. A smart building can respond in meaningful ways

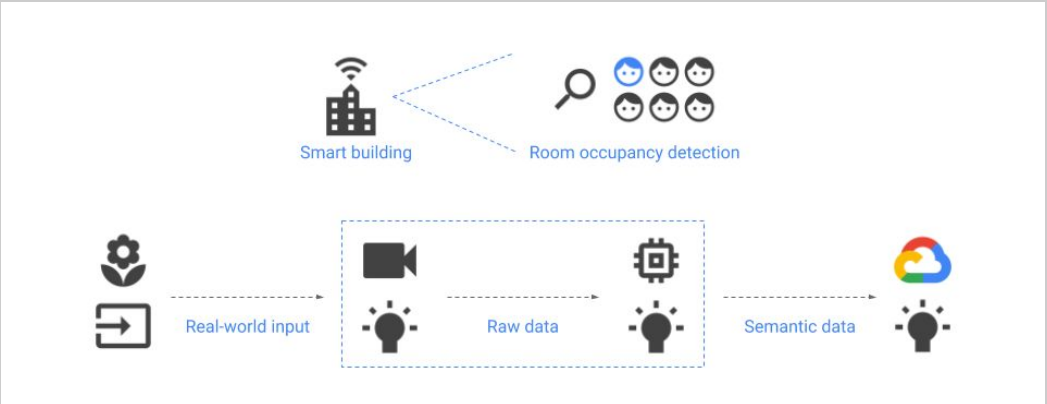


Figure 1 provides another example, a smart building that can adjust HVAC, lights, and other environmental settings for each room based on the number of people and other factors (ambient temperature, light levels) detected at any given time. In addition to the building’s environmental controls, smart cameras at its entrance use facial recognition logic to compare an employee’s photo in the HR database with the image taken by the camera. If these don’t match, the system immediately disables the employee’s (possibly stolen) badge before the person trying to enter the building gets to the security turnstile (and simultaneously sends an alert to the security guard station).

From predictive maintenance and asset tracking to smart buildings and visual intelligence, machine learning is at the center of many IoT scenarios. And although requirements for specific IoT use cases can vary widely, they have in common:

- The need to scale to support any number of devices around the globe.
- The need for increased layers of security.
- The need for real-time or near real-time processing to derive actionable insights that can be applied automatically.

Google Cloud Platform and Cloud IoT Core have been designed to meet these requirements. GCP leverages Google's globally distributed infrastructure that has been responding to billions of queries from billions of users for many years now. When you target Google Cloud Platform and Cloud IoT Core for your devices or your IoT solution, you gain the advantage of Google's unprecedented level of security and scale.

2. Google Cloud Platform and IoT overview

The centerpiece of Google Cloud Platform for IoT is Cloud IoT Core, the fully managed service for securely connecting and managing a global network of devices.

Cloud IoT Core

Google Cloud IoT Core is a fully-managed service for securely connecting and managing a global network of devices. Because Cloud IoT Core is a serverless component of the Google Cloud Platform, it can support any number of connected devices—from one to one million or more—without needing any server provisioning, sizing, tuning, reconfiguration or burdening IT with any other cumbersome administrative tasks.

Data from millions of globally dispersed devices can be ingested by Cloud IoT Core for processing by any number of other fully managed services running on GCP, as needed for the specific use case. In this way, Cloud IoT Core provides something of a gateway into Google Cloud Platform.

Cloud IoT Edge

Cloud IoT Edge is a set of software packages and services that turns Linux-based devices into full featured IoT edge devices capable of running and applying ML models at the data source. Devices can be anything for many different use cases—from patient-centric medical devices to robotic arms, oil rigs, wind turbines, and the like—any device for which it makes sense to apply machine learning in real-time, on board the device.

At the hardware level, Cloud Iot Edge devices can include one or more Edge TPUs (tensor processing unit). An Edge TPU is an extremely small hardware accelerator ASIC (application specific integrated

circuit) developed by Google for the TensorFlow framework. Open sourced by Google,[1] the TensorFlow API and framework is used for machine learning applications and neural networks. (Cloud ML Engine—the serverless, fully managed core component of the Google Cloud Platform that is the centerpiece of Google's machine learning capability for GCP—can be thought of as hosted TensorFlow.) Several Edge TPU chips can fit within the circumference of a penny, so even the smallest devices can be equipped with machine learning capabilities.

TensorFlow Lite is an implementation of the TensorFlow framework designed for mobile and embedded devices. The Edge ML runtime is built on TensorFlow Lite, which works by using a compressed or compiled version of a TensorFlow model generated by an optimizing engine and sent from GCP to the edge device where it is run locally in the context of the Edge ML runtime, using the Edge TPU when available on the device to speed-up processing.

The Edge IoT Core runtime interfaces to Cloud IoT Core and enables the device to connect securely Cloud IoT Core and exchange data, such as when sending data intermittently to GCP so that ML models can be improved (with additional training data, for example) and re-compiled for download back to the edge device as TensorFlow Lite models.

In short, Edge IoT Core comprises the components that enable the device to be just that—an edge device that does not require connectivity to the cloud to gain the benefit of machine learning.

Note For prototyping and development, Google offers an Edge TPU development board comprising a SoM (system on module) with quad-core CPU, Wifi, secure element (a crypto coprocessor chip), and Edge TPU. Another option is a USB-based TPU accelerator.

Reference architecture

Figure 2 shows a reference architecture for supporting IoT data as it emitted from sensors located remotely (eg., drilling platform) via device or edge device to Google Cloud Platform.

1) Telemetry data—typically, a measurement value detected by relevant sensors, such as a temperature reading, number of decibels, distance, on-off switch reading, and so on—is transmitted by the device to Cloud IoT Core. Devices transmitting over MQTT send the data to the same global endpoint (mqtt.googleapis.com) regardless of the device's source region or other location context. This global endpoint means that Cloud IoT Core solutions do not require configuring regions or replicating configurations across regions.

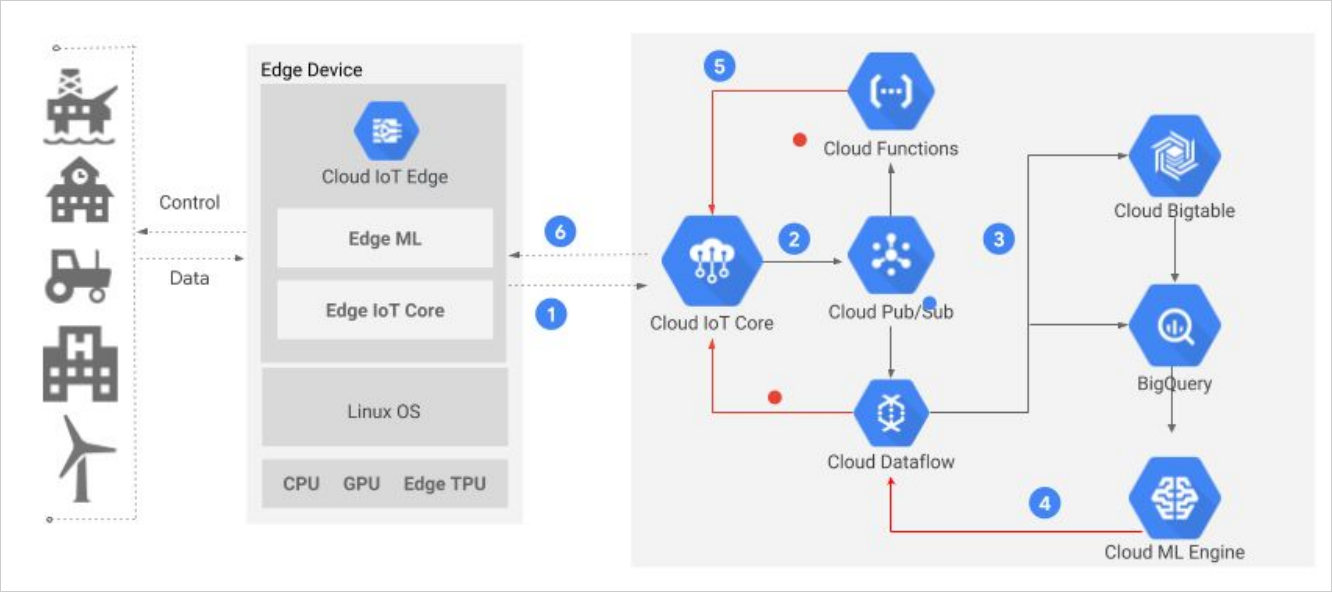
2) Data received by Cloud IoT Core is then sent to Cloud Pub/Sub. Cloud Pub/Sub is GCP's fully managed message queue and event broker. Data received by Cloud IoT Core is brokered as messages to Cloud Pub/Sub which sends the data to a queue as a notification topic. Topics are stored for 7 days (by Cloud Pub/Sub) but can be accessed immediately by other services as needed for the use case.

3) Processing from Cloud IoT Core or from Cloud Pub/Sub can take any number of different paths. For example, data sent from devices worn by heart patients might be anonymized (transformed) by Cloud

[1] TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2015)

Dataflow and added to historical data held on Google Cloud Storage, Google BigQuery, or in Cloud Bigtable. (Researchers could use Google BigQuery for ad hoc analysis or Cloud Data Studio for visualization but for this scenario, the accumulating data will be used to train an ML model and develop an end-to-end solution healthcare solution).

Figure 2. Google Cloud Platform and IoT reference architecture



4) Research scientists trying to develop smart healthcare systems that can predict heart attacks and trigger preemptive measures (activate the patient's cardiac resynchronization device) are using Cloud Machine Learning (ML) Engine to train and refine ML models using the anonymized data stored in buckets on Google Cloud Storage. In another bucket that collects data about actual heart attacks, also collected over time, the researchers can use the training data to refine their models and keep fine-tuning until the model correctly predicts the cardiac rhythms that precede an actual heart attack.

5) In this reference architecture, the Cloud Pub/Sub topic (received at step 2) had another subscriber as well, another fully managed GCP service, Cloud Functions. Cloud Functions is an execution environment for single-purpose functions that can respond to events. Several functions work together to quickly evaluate the actual patient data values in the message topic to determine if medical intervention is needed. If values don't meet certain parameters, another function sends an alert to the patient's medical team via a mobile phone text providing patient details.

6) Control (configuration) data can be sent back to the IoT device by Cloud IoT Core when needed. For example, this entire scenario could be processed locally a Cloud IoT Edge device running the trained ML models developed by the research scientists on Cloud ML Engine. Updates to the refined ML models are compiled using the TensorFlow Lite tools) and sent to the device by Cloud IoT Core. The wearable device monitors the patient in real-time, applies all logic locally, on the device, and sends alerts immediately without round trips to the cloud.

Note that Figure 2 shows the reference architecture in the context of Edge IoT Core and an edge device only for completeness. Neither is required and the processing flow could start from any IoT device

configured to support Cloud IoT Core. However, Google is expanding support to encompass all types of IoT use cases, and that includes supporting edge computing with Cloud IoT Edge, given edge computing's many benefits.

Benefits of edge computing

The benefits of edge computing include:

- **Fast response times**. Data storage and computation can be distributed to the cloud or performed completely locally. Reducing roundtrips to the cloud reduces latency and results in faster response times and real-time preemptive failure diagnosis and repair to ensure smooth running operations
- **Unconstrained by connectivity limitations**. Remote assets such as oil wells, farm pumps, solar farms, windmills, drilling rigs, and so on are difficult to monitor. Edge devices that can locally store and process data ensure reliable operations regardless of limited connectivity.
- **Compliance with strict privacy requirements**. Data transfer from an IoT device to the cloud is typically unavoidable, but edge technology can filter sensitive information locally and send to the cloud only the data needed to create or re-generate an effective ML model that gets sent back to the device.
- **Cost-effectiveness**. The upfront costs of adopting fully distributed IoT include network bandwidth, data storage, and computational power, and such costs can preclude some customers from deploying solutions. By utilizing edge computing, businesses can spread their computational load among cloud and local edge devices for an overall cost-effective IoT solution and good ROI.
- **Interoperability**. Interoperability between legacy and modern devices. Edge devices can serve as a communications liaison between legacy and modern machines, so legacy industrial machines can connect to modern systems or IoT solutions for immediate benefits of capturing insights from both legacy and new systems.

Cloud IoT Core internals

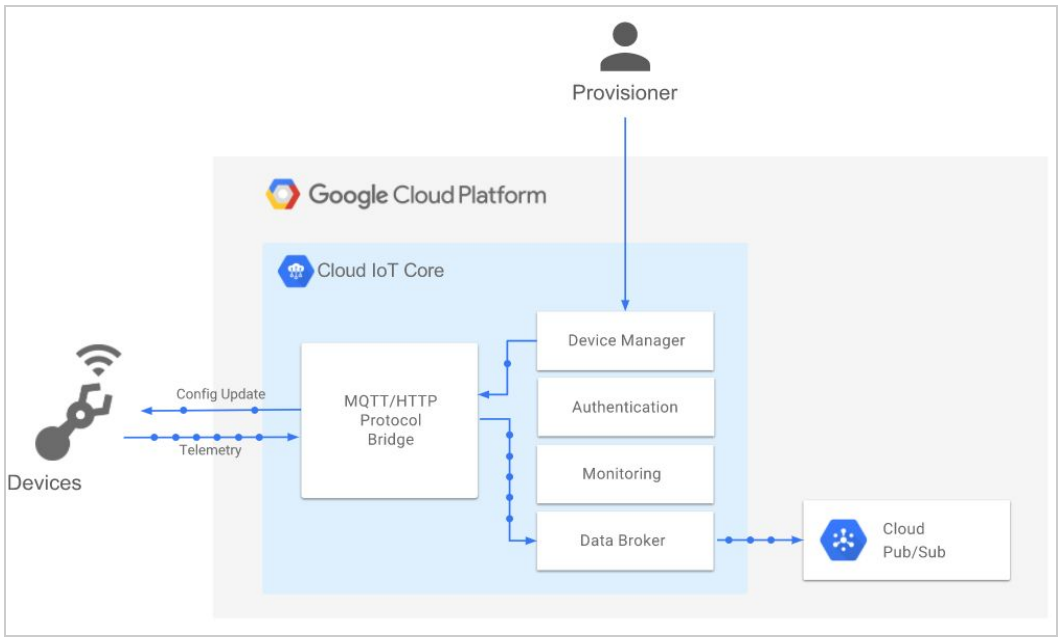
The data flow shown in the reference architecture shows Cloud IoT Core in the context of Google Cloud Platform. Figure 3 shows that internally, Cloud IoT Core comprises several subsystems, including a protocol bridge and device manager.

Devices transmit data securely to the Cloud IoT Core using TLS (TLS 1.2 with certificates) and the protocol bridge listens on the secure MQTT port (8883) and HTTP/S port (443). The endpoint used is a single global endpoint, mqtt.googleapis.com.

Behind the scenes, the protocol bridge also functions as a load balancer, fielding connections from devices and then routing them internally to the appropriate data broker for further processing.

The **device manager** handles configuration, management, monitoring, and deployment at the behest of a *provisioner*—typically, an administrator using the GCP Console or gCloud command line (or programmatically, by invoking the APIs).

Figure 3. Cloud IoT Core internals



The provisioner uses the device manager before deployment to configure device's identity and other details, such as associating the device with public key that will verify JWTs at runtime.

The device manager is also used to configure some of the behavior of Cloud IoT Core for any devices supported by the particular use case. For example, selecting the protocols to be supported by the protocol bridge (HTTP, MQTT, or both) and identifying the type of authentication required from any device (eg, RSA- or EC-encryption? X.509 certificate? and so on).

Thus, the device manager maintains a logical configuration of each device. After devices are deployed to the field, the device manager can also be used to control the device and change its behavior (for example, sending revised TensorFlow Lite models to edge devices).

Note Google is working with secure element (SE) vendors to develop a software service—Cloud IoT Provisioning (early access)—that will be offered to device partners to simplify the provisioning burden for customers so they can connect millions of devices with no provisioning overhead.

3. Partner use cases

Google Partners and Members of the Google Cloud Platform Partner Program can tap into the enormous potential of IoT by developing solutions for Cloud IoT Core. This section provides some ideas for device partners and for application developers.

Partner opportunities—Devices

Device manufacturers and OEMs run the gamut from semiconductor companies and microcontroller unit (MCU) manufacturers to companies that make sensors, developer kits, devices, and gateways as either

starting points for others or complete solutions for customers.

The first step on the path to partnership with Google for IoT is to integrate your device with Cloud IoT Core at the most basic level. For example, if your device is a cloud-enabled digital camera that supports MQTT or HTTP (and which has an underlying OS that can support the necessary libraries), setup a project on GCP to enable basic connection to Cloud IoT Core. See IoT Partner Device Integration Guide for more information.

With basic connection to Cloud IoT Core established for your device, start exploring other functionalities on GCP and start fleshing out your use case to take advantage of other fully managed services that would differentiate your device. For example, getting the images sent from the camera from Cloud IoT Core and into Cloud Bigtable, or getting text into Google BigQuery and processing it in some way to support use cases that would truly differentiate your device and add value for customers.

In addition, you can evaluate Cloud IoT Edge and Edge TPU to determine if your device would benefit from on-board processing logic and machine learning in any fruitful way. Cloud IoT Edge (with its Edge IoT Core and Edge ML runtimes) and Edge TPU in particular are poised to bring the benefits of edge computing directly to your device.

As well as considering Cloud IoT Edge and Edge TPU, you should also evaluate the secure provisioning service. This service will have application for any type of device (edge device or basic device) and should greatly reduce the overhead associated with setting up devices. This add-on service for Cloud IoT Core will simplify device provisioning for customers by using using tamper-resistant hardware-based security (see Example: secure hardware for devices for more information).

Developer kits for prototyping

Over a dozen developer prototyping kits are currently available for Cloud IoT Core for prototyping a variety of devices and gateways. For example:

- For prototyping IoT **devices**, Adafruit offers an ARM-based IoT Kit for Cloud IoT Core that includes a Raspberry Pi3, GPIO breakout cable, breadboard, cables, sensors, and actuators. Documentation is on GitHub in the ARM-software/Cloud-IoT-Core-Kit-Examples repository.

- For prototyping **SE-based devices**, Microchip offers the Microchip security development kit with ATECC608A. This kit includes the secure element (SE), ATECC608A, plus WINC1500 Wi-Fi b/g/n network controller, on a ATSAMD21 Cortex-M0 development board. This kit is aimed at developers who want to prototype industrial grade secure devices using an embedded system. Documentation is on GitHub in the MicrochipTech/gcp-iot-core-examples repository. See Example: Using secure hardware for devices for a closer look at this hardware-based approach.

These are just two examples of the developer kits available. See Get Started with Cloud IoT Core and scroll to find “IoT Developer Prototyping Kits” for the complete list.

Example: Using secure hardware for devices

Cloud IoT Core is secure. Each device is registered using asymmetric cryptography (public/private key pair to encrypt communications over TLS 1.2 (or later) between the device and GCP. The IoT Partner Device Integration Guide shows you how to create keys manually using OpenSSL.

However, secure element (SE) vendors offer hardware-based approaches for creating keys and certificates which removes the burden from the customer. Working in conjunction with a commercial CA (certificate authority) or an organization's internal CA, the digital keys are embedded in the secure element in the device during the manufacturing process.

Figure 4. Using hardware-based asymmetric cryptography to authenticate devices

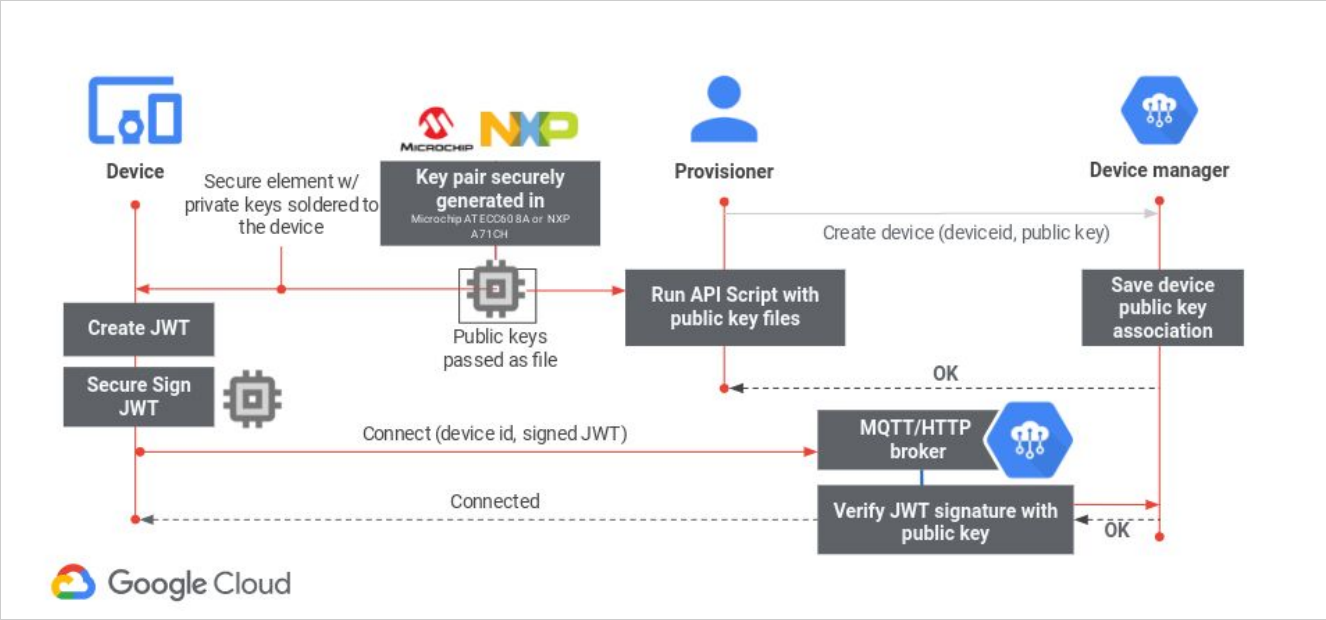


Figure 4 shows the interactions taking place between device and GCP, starting with uploading the device's public key to Cloud IoT Core's device registry (using an API script in the figure). After the public key is associated with the device in the registry, the device can securely and transparently connect to Cloud IoT by using the on-board private key to sign the JSON Web Token (JWT) that gets sent to authenticate the device.

There's a lot more to the story, but the gist is that this hardware-based approach removes all the burden of generating, distributing, and managing keys from the end customer (which also means it's a lot more secure--there is no possible way to physical hack into the hardware and the private key is created and safely stored in the device). For a detailed overview of hardware-based key handling, see the Cloud IoT Core Authentication Use Case video.

Partner opportunities—Applications and end-to-end solutions

Application developers can build software solutions to help provision and manage IoT devices and connections, and to provide insights into data generated from IoT devices. In addition, application developers that focus on **specific vertical markets** can leverage their expertise in those verticals to build

solutions specifically targeting customer needs around device management and analytics.

End-to-end solutions encompass both the hardware device and the application that solves a specific purpose or meets customer needs in some way. Device makers and application developers can look for opportunities to transform legacy products or to bring ML to industries or environments whose tools may be saddled by inefficiencies, facing obsolescence, or otherwise not delivering the benefits of modern technology.

For example, aging commercial buildings may be using hard-wired legacy systems to control the physical plant—HVAC (heating, ventilation, and cooling) system; plumbing; electrical; fire alarm and sprinkler system, and so on. Such systems may use proprietary networks and require building engineers to always be onsite to use the various systems' arcane dedicated tools, which cannot easily provide a coherent snapshot of the building's vitals.

Transforming a system like this should start from the needs of the building engineers who need the data. For starters, they don't want to be physically onsite 24/7--so they want to leverage the cloud and have access from anywhere at any time. They also want a single web-based interface that provides an at-a-glance snapshot of all the building's systems—mechanical, electrical, and so on—rather than a piecemeal approach.

To meet these needs:

- a **device maker** might create a gateway that converts the proprietary protocols of such legacy systems into industry standard IP (internet protocols, TCP/IP) for targeting one or more of the cloud providers;

- an **application developer** might develop a new user interface that aggregates and refines the data from each of the various services.

However, **when a device maker and an application developer join forces,** they can deliver a comprehensive state-of-the-art solution that plugs-into the existing system's network as a gateway to Cloud IoT Core that leverages the data from the legacy systems, sends it to Google Cloud Platform for processing, including applying analytics and machine learning to streamline and automate many of the tasks that occupy way too much of the building engineers' time.

In short, when device makers and application developers team-up to create end-to-end solutions, the end result is much more than the sum of its parts.

Joining forces to build end-to-end solutions

Device makers and application developers who are already Google partners and want to expand their reach by joining forces on a specific solution can find each other in the cross-product Google Cloud Partner directory, by visiting the Partner Lounge, or by talking with your Partner Manager.

Members of the program who don't yet have access to Cloud Partner Directory can reach out to us

through the Partner Program Support form. The Google Cloud Community for Google Cloud Platform is another great way to collaborate with other GCP technologists.

Another opportunity is for device manufacturers and application developers that already have IoT solutions targeting other cloud or network providers to extend their solutions to Google Cloud Platform. Providers that offer their own network infrastructures for various open source or proprietary technologies can offer products such as development kits, gateways, and routers themselves, or by partner with others in their industry to produce such devices.

Feedback...is a gift! Let us know what you think about this Quickstart so we can improve the content in this and other documents in the series.

4. Next steps

Your next steps depend on where you're at and where you want to go. For starters, be sure to take advantage of your benefits as a Google Partner or a Member of the program:

Table 1. Getting started checklist

Benefit	Member	Partner
Coursera credits for technical training	List of courses	Request Form
Qwiklabs credits	NA	Request Form
GCP Sandbox Credits	GCP Free Tier	Request Form
Orbitera Test Drive for your solution	NA	Program Overview
Establish regular meeting cadence to review pipelines, plan joint initiatives, etc.	Office Hours	Reach out to partner management team for help.

Next, focus on your solution:

- **Device partners** should step through the Device Partner Integration Guide and run one of the examples. The integration guide is intended to help Members of the Google Cloud Partner Program and Google Cloud Partners integrate devices with Cloud IoT Core.
- **Application developers** should read the community tutorial, Using IoT Core as Scalable Ingest for Hybrid Projects, which shows how to set up an ingestion layer that can handle data sent between on-premises and Cloud IoT Core—so the use case is practical and can be extended to many other environments. See 5. For more information for more tutorials.

If you're relatively new to Google Cloud Platform:

- Take advantage of the Google Cloud Platform Free Tier to get hands-on experience with the Google Cloud Platform, The GCP Free Tier offers a $300 credit to use with GCP products. See the Device Partner Integration Guide for more information.

Finally, if your development efforts are already underway:

- Use the Partner Growth Plan template to guide your progress as you develop and market solutions.

5. For more information

See the Documentation Quicklist for a complete list of conceptual overviews, tutorials, APIs, and reference documentation for Google Cloud Platform products. In addition to the Quicklist, here are some IoT-specific guides and overviews:

- Architecture: Real Time Stream Processing - Internet of Things—GCP provides the infrastructure to handle streams of data fed from millions of intelligent devices. The architecture for this type of real time stream processing must deal with ingest, processing, storage and analysis of hundreds of millions of events per hour.

- Automating IoT Machine Learning: Bridging Cloud and Device Benefits with Cloud ML Engine— A tutorial that shows how to automate a workflow that delivers new or updated Machine Learning (ML) models directly to IoT (Internet of Things) devices.

- Designing a Connected Vehicle Platform on Cloud IoT Core—Learn about several automotive use cases—usage-based insurance, predictive maintenance, freight tracking, and personalized in-vehicle experience— that bring the data storage and analytics capabilities of the cloud to connected vehicles.

- Google Cloud IoT Core Device to Device Communication—A community tutorial that shows how to send telemetry data from one device to Cloud IoT Core and Cloud Pub/Sub and then use Cloud Functions to receive the telemetry data and effect a change on a second device.

- Oil and Gas Equipment Monitoring and Analytics—Learn about a wide range of Google Cloud Platform services that can be applied to upstream oil and gas use cases. How to enable a system of globally distributed networks to communicate with and monitor on-site equipment; monitor equipment health in real time; analyze and predict equipment failure; and schedule preventative maintenance.

- Real time Data Processing With Cloud IoT Core—A community tutorial that emulates continuously streaming data (such as might be emitted from sensors at an industrial facility) and processing in parallel pipelines in real--time. The sensor data is simulated by a Java application, and the processing pipelines encompass Cloud IoT Core, Cloud Functions, Stackdriver Logging, Cloud Dataflow, and Google BigQuery.

- Using IoT Core as Scalable Ingest for Hybrid Projects—A community tutorial that shows how to create a secure and scalable ingestion layer comprising Cloud IoT Core and Cloud Pub/Sub combined with a small relay service over private networking to an on-prem IoT solution.

Last updated: 15-November-2018