**9**

**Creating a Foundation for IoT**

Market researchers such as Gartner expect that in 2030, over 30 billion devices will be connected through the **Internet of Things (IoT)**. This would mean a skyrocketing growth from the current number of connected devices: at the time of writing, the number of IoT-connected devices is around 14 billion, which is already an incredible figure. We have connected devices in our homes, in factories, and even in cars. All these devices produce and receive data.

The challenge that companies face is how they can monitor and manage these devices, preferably from one platform. Cloud providers offer centralized IoT platforms as a solution to this challenge. In this chapter, we will study the architectural principles of an IoT ecosystem and discuss how the cloud can help in managing IoT devices. We will explore some of these cloud solutions and also look at crucial elements of IoT, such as connectivity and security.

We will cover the following topics in this chapter:

* Choosing the right platform for IoT
* Monitoring IoT ecosystems
* Designing for connectivity to the cloud
* Connecting IoT with IPv6, LoRa, and 5G

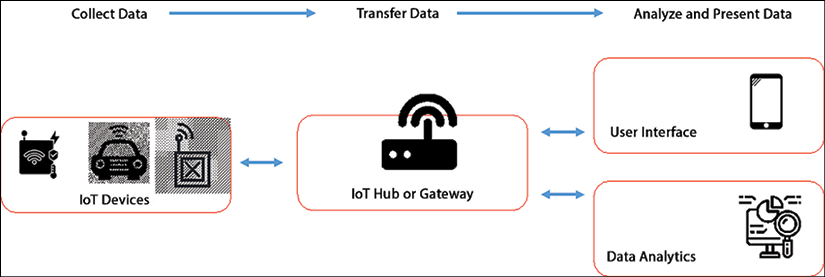
**Choosing the right platform for IoT**

There’s a lot of talk about the IoT, the internet that connects everything: every device, perhaps even a large amount of humans. Let’s first define a “thing.” Basically, a thing can be anything, but typically, we talk about devices that have a connection to a network through, for example, Bluetooth or Wi-Fi and, through this, a connection to the internet.

A device can be a machine or someone carrying a device with a **Unique Identifier** (**UID**). The device is capable of autonomously transferring data over a network, so without human interference. In literature, this is referred to as **Machine-To-Machine** (**M2M**).

A thing can thus also be a device that is “attached” to a person. You can think of implants that monitor the health status of patients. But you may also think of cars with all sorts of sensors alerting the driver when something is wrong with the car and sending a message to the dealer to book an appointment to fix the problem. All these devices and the applications in the devices use the **Internet Protocol** (**IP**) for communication and transferring data. The IoT has great benefits for both users and companies.

How does it all work? First, we are talking about a massive amount of devices. All of these devices are web-enabled and use the internet to transfer data. They collect the data through sensors and connect to the internet through an IoT gateway or edge computer. The data is collected in a centralized data platform where it is analyzed. The following diagram shows a high-level architecture of IoT environments.



*Figure 9.1: High-level architecture of IoT environments*

The concept of IoT is used in almost every industry, and in our homes. Think of sensors that automatically turn the lights on when it gets dark.

It’s essentially the same concept that is used in smart cities where streetlights automatically turn on when light levels get low enough, just as a very simple example. Sensors in smart cities can also analyze traffic and start rerouting traffic at rush hour, preventing heavy congestion. The data that these sensors collect can be used to analyze patterns and help design new street plans.

There are risks, however. Poorly protected devices are a gold mine for hackers. Once they infiltrate a device, they can virtually hop on to the next device and eventually end up in the main systems holding valuable data.

The number of devices is growing rapidly, leaving companies that have to manage these devices with some real challenges. How do you manage millions of devices? To start with: how do you keep track of these devices?

The public cloud might be of good help in observing, monitoring, and managing IoT ecosystems. In the next sections, we will briefly explore some of these platforms.

**Azure IoT Hub**

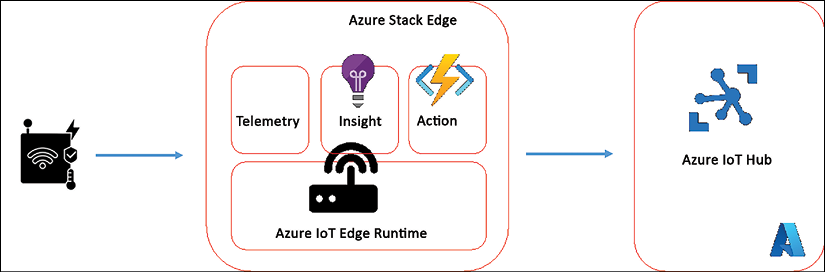
Azure offers **IoT Hub** as a platform to connect devices and receive and process data from these devices. IoT Hub is basically a suite of services, such as over-the-air device updates and integration with Event Grid, IoT Edge, and other Azure services such as Azure Logic Apps, Azure Machine Learning, and Azure Stream Analytics. Device Update allows for group-wise updates of IoT devices, either package-based or image-based. Packages target only specific components of the device, whereas image-based copies the entire updated image to the devices. Device Update was introduced in September 2022 and offers an extensive toolkit to manage devices through IoT Hub, including management, reporting, detailed control of update processes, diagnosis and troubleshooting, and automatic grouping of devices.

It is also worth mentioning IoT Central at this point. This service allows us to develop and manage IoT solutions using an IoT Application **Platform as a Service (aPaaS)**, heavily reducing effort and costs. The simple **User Interface (UI)** lets you connect devices, monitor device conditions, create rules, and manage devices and their data throughout their life cycle in a very comprehensive way.

A good starting point for Azure IoT is <https://azure.microsoft.com/en-us/products/iot-hub/>. More information about IoT Central can be found at <https://learn.microsoft.com/en-us/azure/iot-central/core/overview-iot-central>.

How does IoT Hub communicate system changes to the pool of devices and the applications running on top of these devices? As a message hub, IoT Hub allows for bi-directional communication with the devices. In addition, Azure IoT Hub integrates with Event Grid, allowing more than 500 service endpoints to route events. Event Grid is a serverless broker that is able to communicate events to destinations such as devices. An example would be changing the configuration of an application.

Lastly, IoT Edge must be mentioned. This is a solution where the analysis of IoT data must be closer to the actual devices and data sources. Azure services are wrapped in containers and transferred to the IoT Edge appliances. Workloads such as telemetry but also runtime for artificial intelligence applications are now executed locally on the edge appliance. IoT Edge is also used to send configurations to a specific group of devices. These devices are then connected to a particular edge appliance. The following diagram shows a high-level architecture.

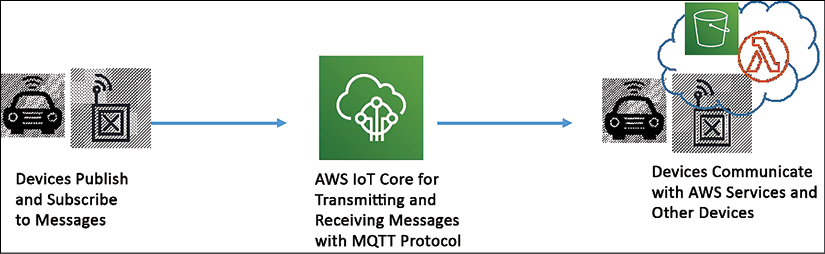


*Figure 9.2: Architecture for Azure IoT Edge*

Azure IoT Hub includes device management as a standard feature and on top offers a suite of services, including integration with security suites such as Defender for IoT for Sentinel, announced in late 2022. Sentinel is Microsoft’s cloud-based **Security Information and Event Management** (**SIEM**) solution. Defender for IoT, the security endpoint solution for IoT, now integrates into Sentinel, providing visibility of all **Operational Technology** (**OT**) and IoT devices and network connections, allowing for the fast detection of and response to potential vulnerabilities in the entire ecosystem.

**AWS IoT Core, Edge Manager, and IoT Greengrass**

AWS IoT Core is comparable with Azure IoT Hub. The service provides secure communication for IoT devices, allowing for the processing of data coming from these devices. However, users will need to add AWS IoT Device Management to IoT Core to enable the remote monitoring and management of devices themselves. In other words, IoT Core lets you connect devices to a central environment where IoT Core serves as a message broker between devices using the lightweight **MQTT** (**MQ Telemetry Transport**) protocol. *Figure 9.3* shows the principle of IoT Core.



*Figure 9.3: Architecture for AWS IoT Core*

Like Azure, AWS offers various services that enable the full-stack management of IoT ecosystems. Important services to mention are IoT Greengrass and Edge Manager. The IoT Greengrass cloud service in the AWS cloud helps build, configure, and deploy software to IoT devices. Greengrass communicates with Greengrass client software on edge appliances or core devices. The IoT devices themselves connect to these edge appliances.

The Greengrass architecture makes it easy to manage apps on the devices, operate the devices including anomaly detection, and locally process data. IoT Greengrass is also able to act as a gateway, allowing devices to connect and communicate even when they are not connected to the internet.

The next step would be to predict the behavior of devices and services running on these devices in case of specific events, allowing for the continuous optimization of the services and performance of the devices. This can be done through **Machine Learning** (**ML**). AWS offers SageMaker as an ML engine and with SageMaker Edge Manager, the models used for ML can be deployed to IoT devices.

The best way to get started with IoT in AWS is by reading the documentation at <https://aws.amazon.com/iot/>.

In most cases, IoT will be used in industrialized environments with industrial equipment. You can think of sensors in production lines and manufacturing robots. Collecting, modeling, and analyzing industrial data from industrial IoT devices is done through AWS SiteWise.

Be aware that we only listed a few basic services of AWS IoT. Obviously, AWS also offers solutions that can be added to enhance IoT management, including, for instance, IoT Device Defender as endpoint protection for devices and specific industry solutions such as IoT FleetWise for IoT in vehicles.

**Google Cloud IoT Core**

In 2015, Google introduced two important services for implementing applications across IoT devices. The first one was Brillo, an operating system based on Google’s Android, but scrubbed down to make it suitable for low-power devices. Weave was a communication protocol integrated into Brillo that allowed for communication between devices and the cloud. Both Brillo and Weave were rebranded to Android Things, which was depreciated at the beginning of 2022.

What does Google have to offer in terms of IoT nowadays? The main service is Google Cloud IoT Core, a suite of partner-led solutions hosted on the GCP platform. Some examples of these are Aeris and ThingsBoard. Aeris is a software platform for IoT and ThingsBoard is a solution for device management.

Google can integrate these partner solutions with data analytics and machine learning systems such as Cloud Dataflow and BigQuery for data insights.

**Alibaba IoT Platform**

Alibaba Cloud offers IoT Platform for connecting and managing devices and data. The platform has connection, communication, security, and device management capabilities, including over-the-air updates. IoT Platform is comparable to Azure IoT Hub and AWS IoT, with a full-stack service for entire IoT ecosystems.

Extensive documentation on Alibaba’s IoT solutions can be found at <https://www.alibabacloud.com/help/en/iot-platform>.

**Monitoring IoT ecosystems**

The most important challenge that architects should address in IoT is, not surprisingly, security. The security estate of IoT devices must be monitored continuously. To understand the risks better, we can have a look at the top risks that are listed by the **Open** **Web Application Security Project (OWASP)**. We’ve only listed the top five:

* Weak passwords
* Poorly protected network services
* Poorly secured interfaces
* Lack of update mechanisms for security rules and patches
* Use of outdated components

The top risk, however, is the lack of device management and leaving data transfer unmonitored. However, this starts with knowing where the devices are. But knowing where devices sit is not sufficient; we must also know what these devices are doing, what sort of data the devices collect, and how they collect it. Observability is the key principle that any IoT architecture must comply with.

This is already the biggest challenge in IoT. Think of the fact that an IoT ecosystem might consist of hundreds or thousands of devices. A good example might be a water company that has sensors in every pipe that transports water. There are sensors in the production facilities as well, collecting data about the water quality in the various stages. All these sensors collect data about the quality and distribution of water through a complex system of pipes. Important parameters such as water pressure and the functioning of pumps are crucial.

The question that must be raised is: does the company need to monitor every single sensor? Or do we need to determine what is really important to monitor, instead of monitoring and responding to every single error or failure of a sensor? The data that all sensors collect must be accurate and relevant to decide whether the entire system works as designed and predicted. In other words, data analysis is the key to monitoring IoT.

IoT devices collect data, but they also produce data. That data is important to monitor the status of the device itself, including its security posture.

Let’s recap the requirements for businesses to monitor IoT ecosystems:

* **IoT hardware**: Typically, IoT devices have a **CPU** (**central processing unit**), memory, and perhaps disks to store data. To ensure that the device is running and performing in an optimal way, we must monitor these components in the device, just as we would do with any other piece of hardware. Monitoring will be about checking that thresholds on usage on the CPU, memory, or disks are not exceeded, which could lead to slow performance or even failures. IoT monitoring tools that have been integrated into IoT suites will be able to send out real-time alerts in case of performance degradation or malfunction of devices.
* **IoT software and applications**: On top of the hardware, there will be software and applications running. This software will vary from operating systems up to small apps or functions that are hosted on lightweight containers with K3s, although K3s itself is developed to run workloads in unattended remote locations. Keep in mind that unattended doesn’t mean “not monitored”: these are different things. Whatever is running on IoT devices, this software must be monitored too. Monitoring tools must be able to detect issues in the malfunctioning of software since it will prevent devices from working correctly altogether.
* **Data**: As we have discussed, IoT devices collect and transfer data. The devices must be able to send that data to a central environment, such as an IoT hub or gateway. We will be discussing the gateway in a later section of this chapter. Just as important is that devices must be able to receive data, for instance, updates and patches. A non-interrupted flow of data is crucial for the optimal functioning of IoT devices. A glitch in that data flow might already cause severe problems. Monitoring the connectivity by monitoring the data flow between devices and between devices and the central managing environment is an important aspect of the monitoring tool. Interruptions must be detected in real time.
* **Security posture**: Last but not least, IoT devices must be secured and hardened. Hardening is the process of disabling redundant features and/or security risks and encrypting connections. This is to make it as difficult as possible for attackers to gain access to a system. In the occasion that the attacker does get access, it should be as difficult as possible to use obtained data. This also applies to IoT devices. Unnecessary accounts must be deleted, access restricted with least privilege, and connections encrypted. Also, an active policy must be applied for software and firmware updates, with the highest priority for security patches.

Monitoring tools must send alerts whenever security postures are breached. An IoT device is often an open door to a network leading to many other systems with valuable data. This is also true for home IoT: a non-protected door camera can be hacked, providing entry to the home network and computers that are connected to this network—a PC that holds the financial data of the owner, for instance. In companies, the consequences can be even more severe and lead to major data leaks.

The architecture for the IoT ecosystem must cover these topics and address the challenges. Next, we can define how the public cloud can help in providing solutions to overcome these challenges.

Implementing monitoring systems for IoT devices will present several challenges, such as:

* **Observability**: An IoT ecosystem will contain a huge number of distributed devices. This is the first problem that organizations will face: having a complete view of the entire landscape.
* **Detection and response**: It’s crucial to detect performance or security issues as quickly as possible to prevent the distribution of the issue through the ecosystem. The speed of detection and response is key. Issues must be detected before critical processes and, with that, services to customers are impacted.
* **Integration**: From time to time, new devices will be added to the landscape. These devices must be integrated into the ecosystem and promptly connected to monitoring systems. The latter is important for security reasons: the new device must be compliant with the security policies. A new device might introduce the risk of vulnerabilities when security policies are not applied instantly.

How can public cloud environments like the ones that we discussed previously help in setting up IoT ecosystems and monitoring these ecosystems? The answer: cloud providers are agnostic and provide a central solution that is not restricted to specific IoT technology or manufacturers of devices. Cloud platforms allow companies to have a single-pane-of-glass view of the entire landscape with central management capabilities.

Solutions such as IoT Hub of Azure, AWS IoT, and Google’s IoT Core all provide tools and solutions for:

* The auto-discovery of IoT devices in a network
* Centralized remote device monitoring including alerting
* The centralized monitoring and management of IoT security
* The centralized monitoring and management of connections

The biggest advantage, however, of implementing a cloud solution to manage IoT ecosystems is that the public cloud can offer all of this at speed and scale. The question is: what are the architectural requirements for an IoT system? First, we need to make sure that devices can connect with each other and to our cloud of choice. We will discuss this in the next sections, starting with connecting IoT ecosystems to the cloud.

**Designing for connectivity to the cloud**

Before we get to monitoring and managing an IoT ecosystem, there’s one crucial step that we must take first: connecting the IoT devices to our cloud. Gateways and edge computing are solutions to this.

We can’t simply connect thousands of IoT devices one by one to systems in the cloud. First, every connection in itself poses a risk of intrusion and with that the risk of a security breach. The solution to this is to have all connections targeting one system, machine, or instance that sits in front of the cloud environment, before the data of IoT devices is entered into systems that are hosted inside our cloud.

The IoT gateway is such as system. It can be a virtual machine or a service, but in all cases, it serves as the connection point between the cloud and the IoT devices. All data that flows between devices and the cloud environments will have to pass the gateway. The IoT gateway is the central connectivity and data controller. The diagram in *Figure 9.1*shows the principle of the IoT gateway.

Sometimes IoT gateway and edge computing get mixed up, but these are different things. Edge computing is about having compute power at the periphery—the edge—of a network. You could say that it’s a small piece of cloud that sits closer to the data source. A typical use case for edge computing is to prevent latency when processing power needs to be extremely close to the data source. Other use cases include privacy and compliance, when data is not supposed to be processed in a central cloud environment.

Edge computers can serve as gateways. Data from, for instance, IoT sensors is collected in edge computers where the data is processed. This is particularly useful when IoT devices are not capable of sending data over large distances, which is true for almost every IoT device. Keep in mind that sending data requires a network and power. IoT devices often only have a small battery. When the device constantly has to send data over long distances, the battery will be worn out fast. Shortening the data distance can save a lot of power. Edge computers used as gateways are a solution. Edge computers will process the data of the devices and also send relevant data to the central cloud environment for further analysis.

Cloud providers offer various solutions for edge computing. The two leaders in this space are Azure Stack Edge and AWS Outposts:

* **Azure Stack Edge**: This is a device that is completely managed by Azure. The appliance can be ordered through the Azure portal. There are four versions of Stack Edge, one with 32 vCPUs (Pro) and one with 40 vCPUs (Pro 2). The 1U rack version can hold up to 4.2 Tb of storage and is also equipped with a **Graphics Processing Unity (GPU)**. Stack Edge performs as a cloud storage gateway, enabling data transfers to Azure while retaining local access to files on the appliance itself. There are two additional versions: Pro R, providing a GPU, and Mini R, with a **Vision Processing Unit (VPU)** for edge processing.
* **AWS Outposts**: Outposts is an AWS-managed appliance that can be hosted on-premises. It is delivered and installed either as a 1U or 2U rack server or as an AWS Outposts 42U rack. Customers can run native AWS services on-premises with AWS Outposts, including Edge Manager and IoT Greengrass.

Let’s recap the challenges in managing IoT: it’s about massive numbers of devices that we need to connect to a centralized platform for, among others, monitoring and data analysis. IoT devices are extremely distributed and typically only run with a small battery as a power source. Next, we need these devices to send and receive data in real time, at high speed, and across quite some distances. This requires different connection types and communication protocols. We will explore these in the next section.

**Connecting IoT with IPv6, LoRa, and 5G**

In the previous section, we discussed how we can connect IoT ecosystems to public cloud environments through gateways and edge computing. It’s important to understand that IoT devices use different protocols to communicate with each other and eventually with the cloud. The reason for this is the massive number of devices that must be connected and with that the amount of data that is transferred from these devices to the cloud. The infrastructure must be capable of absorbing this, but it requires different means of communication.

We need connectivity that is able to connect machines with machines and continuously transport real-time, small chunks of data between the devices and cloud environments.

There are several emerging IoT standards, including the following:

* **IPv6 over Low-Power Wireless Personal Area Networks** (**6LoWPAN**): The **Internet Engineering Task Force** (**IETF**) defines this as an open standard for connecting and communicating between low-power IoT devices and the internet. The 6LoWPAN standard includes 804.15.4, **Bluetooth Low Energy**(**BLE**), and Z-Wave for home automation.
* **Zigbee**: This is likely the best-known standard for IoT, mainly used in industrial settings as a low-power, low-data rate wireless network. Zigbee is based on the common 802.15.4 standard of the **Institute of Electrical and Electronics Engineers** (**IEEE**).
* **OneM2M**: This is a global standard for M2M communication, embedded in software and hardware for devices. Another standard for real-time M2M communication is **Data Distribution Service**(**DDS**), developed by the Object Management Group.
* **Advanced Message Queuing Protocol** (**AMQP**): This open-source standard is used for asynchronous, encrypted messaging by wire across various applications. This protocol is widely used in IoT device management and supported by all IoT solutions that we have discussed in this chapter.
* **Constrained Application Protocol** (**CoAP**): This protocol was designed by the IETF to define how constrained, low-powered devices such as wireless sensors can communicate in networks. It uses a very simple syntax for messaging. The smallest CoAP message can be just 4 bytes.
* **Long-Range Wide Area Network** (**LoRaWAN**): This is designed to support huge networks with IoT devices, such as smart cities.

One technology that we must mention here is **5G/LTE** (**Long-Term Evolution**). The LoRaWAN network that we mentioned in the list is suitable for devices that occasionally need to be connected online. For devices that must be continuously connected and require a higher network speed and bandwidth, LTE-M and L4G M2M are better solutions.

The introduction and roll-out of 5G networks is the next phase with a huge potential to accelerate IoT. The biggest advantage of 5G is the ultra-low latency on the network, allowing for extremely fast response times, making it suitable for communication between services on the internet—the cloud—and fast-moving vehicles.

Some cloud providers have already developed services that use 5G. Azure Network Function Manager is a service that allows you to deploy specialized network functions—such as mobile packet core to enable a private LTE/5G solution on Azure Stack Edge.

Another example is AWS Wavelength, which allows for deploying ultra-low-latency applications. With Wavelength, parts of an application can be transferred to a Wavelength Zone, which is an extension of a virtual private cloud in AWS. IoT devices connected to 5G—think of moving vehicles—can connect to these applications in the Wavelength Zone without leaving the mobile network of a telecommunications provider. Typically, the device would communicate over different hops over the internet to reach the destination in the cloud, causing latency. Wavelength offers a solution for seamless communication.

More information on Azure Network Function Manager can be found at <https://azure.microsoft.com/en-us/products/azure-network-function-manager/>. For more details on AWS Wavelength, refer to <https://aws.amazon.com/wavelength/>.

**Summary**

In this chapter, we discussed the basic architecture principles for IoT ecosystems. It’s fair to say that IoT architecture is extremely complex, exposing a lot of challenges. The first challenge is the scale of an IoT ecosystem that can easily hold hundreds to thousands of devices. All these devices must be secured and monitored. Next, these devices send data that must be analyzed. This requires a centralized platform.

We explored the IoT solutions of Azure, AWS, GCP, and Alibaba and what they have to offer. The services vary from the update management of devices to security monitoring to detect and respond to potential vulnerabilities in IoT ecosystems. An important element of IoT architecture is connectivity. In the final sections, we studied connectivity to the cloud using IoT gateways and edge appliances and the various protocols to enable low-powered devices to connect and communicate with each other. Last, we explored the possibilities of 5G and how this will impact the growth of IoT.

This concludes the section about setting up foundational environments for several use cases such as data analytics and IoT. The next part of this book will be about keeping financial control over cloud assets, using the principles of FinOps.

**Questions**

1. What AWS solution would you use to build, configure, and deploy applications to IoT devices?
2. Both Azure and AWS offer appliances that can be used for edge computing. Name the two solutions.
3. What does LoRaWAN stand for?

**Further reading**

* *IoT and Edge Computing for Architects*, by Perry Lee, Packt Publishing, 2020