UNIVERSITY OF WATERLOO

Faculty of Engineering

Department of Electrical and Computer Engineering

Analysis of Cloud based IoT Platforms

Connected Lab, Inc.

370 King Street West, Unit 300

Toronto, Ontario, Canada

Prepared by

Chang Bok Lee

ID 2031001

userid cb3lee

2B, Computer Engineering

3 January 2017

Chang Bok Lee

N2306-330 Phillip Street

Waterloo, Ontario

N2L 3W9

January 3, 2017

Vincent Gaudet, chai

Electrical and Computer Engineering

University of Waterloo

Waterloo, Ontario

N2L 3G1

Dear Sir:

This report “Analysis of Cloud based IoT Platforms”, was prepared for my 3A work report. This report is intended for the WKRPT 201 course and it was written during my work term at Connected Lab, where I worked for Arcwright Product Team as a developer.

Connected Lab provides wide range of resources for software development for new hardware. Typically Connected Lab would be contracted to appliance and hardware developing companies that has less expertise and experiences in developing software. Connected Lab helps those companies to make their appliances and hardware smarter by providing the software for mobile devices and the web infrastructure. I was in role to participate in development of the Arcwright which provides a connected IoT devices management platform. The purpose of this report is to analyze and compare the cloud based IoT platforms that are available in the market. This report may be used to help the decision making for the future roadmap and the demanded features for Arcwright.

I would like to thank Yi Zhang for helping with understanding the current situation of the different IoT platforms. Mr. Zhang also provided me with the great details about the general plan for the Arcwright product which helped me to extract important topics over many different IoT platforms. I would also like to express my appreciation to David Hong and M. Heo for proofreading and advising for this report. I hereby confirm that I have received no further help other than what is mentioned above in writing this report. I also confirm this report has not been previously submitted for academic credit at this or any other academic institution.

Sincerely,

Chang Bok Lee

ID 20361001

# Contributions

The Team that I worked for was relatively small. It consisted of four full time employees and three coops including myself. The team was effectively separated from the contracted work that Connected Lab was in part of. The team was essentially developing and researching a new product rather than working as a contractor for other companies. And the product was named Arcwright after an English inventor Sir Richard Arkwright.

The team’s main goals were the design and the development of the product Arcwright. Arcwright is a software product to provide IoT device manufacturers with the end-user devices management platform. It was planned to be built on top of the existing cloud platforms that many clients will be familiar with, such as AWS IoT, IBM Watson IoT, or Microsoft Azure IoT. Arcwright is designed to help consumer electronic device manufacturers. Often those manufacturers have numerous iterations of the same product making it difficult to manage them. Also, there are not much expertise or experience in building a software that connects the Things or the consumer electronic device to the Internet. Arcwright will help these manufacturers make those Things to be connected to the Internet securely with ease. Also, Arcwright provides a robust management platform for the iterations of devices and for each end-user devices. In the roadmap, it is planned to support analytics and visualization of data flow. The project launched in early 2016 and the actual development started in the near end of August 2016. Since the project is in such early stage, the detailed directions of the project and the features of the product are still extremely flexible.

My main tasks were to develop the product in agile fashion. As a side duty, I was the scrum master to lead the daily stand-up meetings and make sure every developer is on track with the weekly updated task list. The agile development methods were well introduced by many full time employees of Connected Lab. I spent most of the time working on developing the software development kit (SDK) for Node.js. The SDK is intended to be used by clients, the manufacturer of consumer electronic devices, to help them connecting their devices to the IoT platform of their choice without pain. The SDK also helps registering their devices to the Arcwright device management platform. I was also involved in developing the front-end of the device management tool which allows clients to create device entities and manage the devices and the end-user devices.

The relationship between this report and my job is that the report involves different cloud based IoT platforms that I needed to develop SDKs for. Although Arcwright only supports AWS IoT as of December 2016, it has a plan to expand the SDK to support numerous IoT platform providers. This report is an useful resource to the team to understand the differences among the IoT platform providers. This report will also help the team to recommend different IoT platforms to the future clients if the clients do not have existing IoT platform. In order to develop the SDK for Node.js devices, I researched and learned about AWS IoT platform and its SDK. That helped me to understand the data flow of an IoT platform which was helpful to analyze different IoT platform providers. Writing this report helped me to organize and evaluate information around the different IoT platform providers. It was important to have a clear recommendation since this report may be a reference to a recommendation to the clients of Arcwright and Connected Lab.

In the broader scheme of things, the work that I performed helped creating the very first own product that is developed by Connected Lab. It is still not clear if Arcwright will be released as a full product. However, as a software company that claims “We Design & Build Software for New Hardware”, it is a valuable asset to understand the different IoT platforms in the market.

# Summary

The main purpose of this report is to analyze and compare three different cloud based Internet of Things(IoT) platform providers; Amazon Web Services(AWS) IoT, Microsoft Azure IoT, and Google Cloud Platform. Currently Arcwright only supports AWS IoT. This report will help verifying the validity of the expansion plan to support the other platforms. This report is intended for readers with knowledge of general programming and web application development.

There are two major sections to the body of this report. The first section investigates and analyzes the common factors of the three providers; the cost of using the products, or how security is handled. The second section analyzes strength and weakness of each platform.

The conclusion is that AWS IoT provides the most suitable solution for Arcwright with the straight forward pricing model, various options for authentication, and the easy integration with other products. Microsoft Azure IoT also comes very close with the highly optimizable pricing model and the exclusive product suite from Microsoft. Google Cloud Platform came as the least favourable option for the team. However, the platform offers a high flexibility and scalability. Also, it exposes good potential market to the team.

The recommendation in this report is to expand Arcwright’s SDK to Google Cloud Platform. The AWS IoT SDK is under development and Google Cloud Platform leaves the team space to initiate the development of SDK for the platform.

Table of Contents

[Contributions iii](#_Toc481961913)

[Summary v](#_Toc481961914)

[List of Figures vii](#_Toc481961915)

[List of Tables viii](#_Toc481961916)

[1. Introduction 1](#_Toc481961917)

[1.1 Internet of Things 1](#_Toc481961918)

[1.2 Arcwright 1](#_Toc481961919)

[1.3 Scope 2](#_Toc481961920)

[1.4 Outline 2](#_Toc481961921)

[2. Basic Components 2](#_Toc481961922)

[2.1 Security 3](#_Toc481961923)

[2.2 Supported Protocols 4](#_Toc481961924)

[2.3 Software Development Kit 4](#_Toc481961925)

[2.4 Pricing 5](#_Toc481961926)

[3. Provider Specific Features 8](#_Toc481961927)

[3.1 Amazon Web Service IoT 8](#_Toc481961928)

[3.2 Microsoft Azure 9](#_Toc481961929)

[3.3 Google Cloud Platform 10](#_Toc481961930)

[4. Conclusions 11](#_Toc481961931)

[5. Recommendations 11](#_Toc481961932)

[Glossary 12](#_Toc481961933)

[References 13](#_Toc481961934)

[Appendix A Amazon Web Service IoT Details 14](#_Toc481961935)

# List of Figures

Figure 2-1. AWS IoT Authentication 3

Figure 3-2. AWS IoT overview 8

Figure 3-3. Azure IoT Hub 9

Figure 3-4. Google Cloud Platform for IoT 10

# List of Tables

Table 2-1. Supported Protocols of each Service provider…………………………………………4

Table 2-2. Azure IoT Hub Pricing…………………………………………………………………6

# Introduction

Connected Lab is a software engineering company that designs and develops software for new hardware. The targeted clients are often consumer electronics manufacturers. As of 2016, Internet of Things (IoT) is one of the most important keyword in the consumer electronics market. Countless number of consumer electronic devices are present in the market, and often those devices lack of connectivity with other devices. Although, traditional consumer electronic device manufacturers are eager to be involved in the current IoT trend, it is difficult to allocate resources for them especially due to the lack of experience. Also these manufacturers tend to have conventional engineering process which is not suitable for the fast paced agile software development. Connected Lab offers these manufacturers with a critical design process suitable for the modern software development process as well as great resources to create specialized applications and software.

## Internet of Things

At this point, it is important to define what IoT is. It is “broadly-deployed aggregate computing/communication application and application-consumption system consisting of (*i*) dispersed instrumented objects (“Things”) with embedded one- or two-way communications and some or no computing capabilities, (*ii*) where objects are reachable over a variety of wireless or wired local area or wide area networks, and, (*iii*) whose inbound data and outbound commands are pipelined to or issued by a system with a degree of human or computer-based intelligence.” [**1**] Today’s Internet provides network between people. The Internet of Things, in contrast, “the IoT enables things to access data and communicate with one another.” [**2**] The Internet is a pipeline for non-human devices triggering machine-to-machine communication.

## Arcwright

Arcwright is a product developed by Connected Lab that provides flexible methods to help designing and developing new IoT ready devices. Arcwright also acts as a device management platform that offers a robust feature set and a pain-free end-user device management. Software Development Kit (SDK) included in Arcwright product helps designing and developing a new hardware to be IoT ready.

Currently, Cloud based IoT market is already crowded and provides numerous well-designed infrastructures. Using a cloud based IoT platform helps engineering a new IoT device. It will help manufacturers to worry about the Things part rather that the Internet part.

This report is concerned with initial investigation and comparison for three different major cloud based IoT platforms available in the market.

## Scope

The purpose of this report is to analyze the differences and compare three different major cloud based IoT platform services; Amazon Web Services IoT, Microsoft Azure IoT, and Google Cloud Platform. This will help deciding the future roadmap for Arcwright product. Furthermore, this report may be used as a reference for the recommendations to the potential clients.

## Outline

The following sections introduce each platform and investigate the platform in different components. The basic components include categories and factors that are present in every platform such as security features, documentations, supported protocols and pricing. Furthermore, the platform specific features are analyzed and investigated in subsections. This helps identifying the weakness and the strength of each platform.

# Basic Components

There are a number of factors and features to be considered. First of all, the platform security is one of the top concerns for any manufacturer. While all of the service providers have secured interconnection between “Things”, it is important to note the differences between the providers.

## Security

The AWS IoT is Transport Layer Security (TLS)-based and all the messages between the nodes and the message broker are encrypted. Various TLS Cipher suites are supported, and detailed list is available in Appendix A of this report. AWS IoT supports X.509 certificates, IAM service, Amazon Cognito identities, and Federated identities. [**3**] Customers can create and use their own certificate attached to the device and the policies using AWS Command Line Interface (CLI).

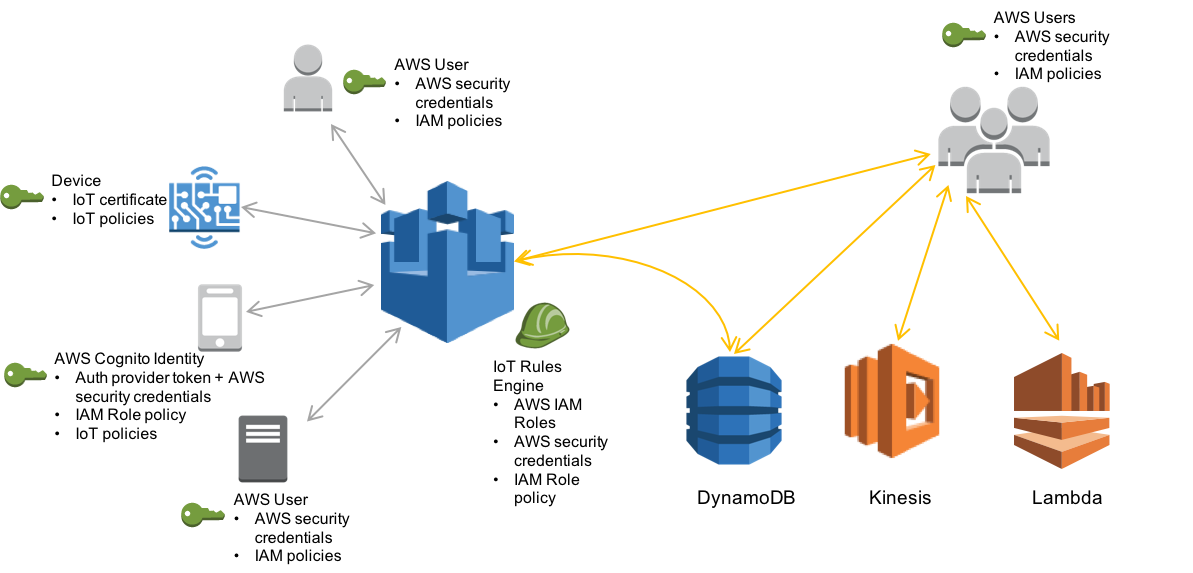


Figure 2-1. AWS IoT Authentication [**3**]

Microsoft Azure IoT also relies TLS protocol. A notable difference is that Azure leaves us less options to authenticate devices; using the X.509 certificate provided by Azure. [**4**] On the other hand, Azure provides a great documentation around security architecture and best practices of any IoT system development.

Google Cloud Platform does not have exclusive IoT packages. Although basic TLS protocol is supported by GCP, it lacks device-level authentication and activation capabilities. For Arcwright, this should be included in the future SDK for the client who decides to use GCP as their provider.

All three providers are TLS-based and all the messages are encrypted. However, AWS offers the most flexible options for device-level authentication and activation followed by Azure. The device-level authentication component in GCP requires development resources.

## Supported Protocols

Communication protocol is another basic component of an IoT platform. The nature of “Things”, commonly being low powered with limited computation power, has led the industry to utilize a lighter protocol than HTTP. Message Queue Telemetry Transport (MQTT) is an ISO standard publish-subscribe-based “lightweight” messaging protocol for use on top of the TCP/IP protocol. [**5**] MQTT is one of the industry standard protocols for the IoT industry. However, that is the case for smaller devices such as light bulbs or thermometers. The majority of consumer electronics usually have enough computing power to process HTTP protocol. While all three providers support REST API using HTTP, there are small differences in the supported protocols.

Table 2-1. Supported Protocols of each Service provider

|  |  |
| --- | --- |
| **Service provider** | **Supported protocols** |
| Amazon Web Service IoT | HTTP, MQTT, MQTT using WebSocket |
| Microsoft Azure IoT | HTTP, MQTT, AMQP |
| Google Cloud Platform | HTTP, gRPC |

Table 2-1 shows which protocols are supported by the different service providers. Both of AWS and Azure come with the industry standard HTTP and MQTT. The AWS IoT has a slight advantage since they allow using WebSocket to support MQTT communication. Meanwhile, GCP only supports HTTP and gRPC. The initial impression of gRPC is a high performance publish-subscribe protocol with a high throughput. Although a further investigation is required for gRPC, it is rarely used in the target client’s product at the moment. Advanced Messaging Queuing Protocol (AMQP) is a newer technology applied to the application layer of OSI model. It could be considered as an alternate option to MQTT. Currently, only Azure supports AMQP. Not only AMQP has a built-in support for MQTT, it has additional features like message transformation, custom authentication, compression/decompression, and encryption/decryption of traffic. [**7**]

## Software Development Kit

“Thing” is usually a piece of physical hardware that has some interoperability computing capability. The service providers offer software development kits (SDK) to help designing an IoT device that uses their service.

Amazon Web Service IoT Cloud is accessible by the end-devices using standard protocols like HTTP and MQTT. However, to help developing the end-device, AWS offers a device SDK in popular languages such as Java, JavaScript, Python, and Embedded C. The majority of the IoT devices runs on low power, and it has a limited computing power. It is helpful to have many options for the device SDKs. One notable language for AWS IoT is Arduino Yún SDK, since the other providers do not offer any Arduino SDK. All of the SDKs are open-source and available in GitHub.

AWS IoT Button is a programmable Wi-Fi button to provide a simple start for an IoT project. It was heavily used during development of Arcwright as a testing device. AWS IoT is in partnership with some hardware manufacturers, such as Intel, Broadcom, and Avnet, to provide the AWS IoT Starter Kits. These physical kits allow quick prototyping and designing.

The Azure IoT SDKs also come in popular languages; C, .NET, Java, JavaScript (Node.js), and Python. Of course, being a Microsoft platform, they offer SDK in .NET. as well as an operating system for the devices; Windows 10 IoT Core. Azure also issues “Microsoft Azure Certified” certification to the available boards in the market.

Google Cloud Platform has a slightly different approach. It does not have an exclusive end-to-end IoT SDK for devices. However, it is worthwhile to consider Android Things. Android Things is supported on several hardware platforms including Intel Edison and Joule, and Raspberry Pi 3. One of the most demanded solution for Arcwright to have is Over-The-Air (OTA) updating software solution. Android Things comes with built-in OTA solution. Also, Connected Lab already has plenty resources for Android platform and Android Things is essentially a lightweight OS for end-user devices.

## Pricing

Amazon Web Service has a pretty straight forward pricing with pay-as-you-go and no upfront cost model. AWS charges per million messages. Each message is 512 byte and counted multiple of 512-bytes. For example, 1200 byte message will be considered as three messages. Price per million messages varies from $5 to $8 depends on the region. Both of the inbound (publish) and the outbound (delivery) messages are charged.

Microsoft Azure has a slightly more complicated pricing model. It charges monthly per its IoT Hub unit. Each Hub unit can connect to unlimited number of devices but the total number of messages is limited per day base. There are three different tiers and higher the tier is more of messages can be processed. Also, Azure has 4 KB message size, eight times larger than the AWS message size.

Table 2-2. Azure IoT Hub Pricing

|  |  |  |
| --- | --- | --- |
| **Edition Type (Tier)** | **Price Per Unit (monthly)** | **Total Number of Messages/Day Per Unit** |
| S1 | $60.80 | 400,000 |
| S2 | $607.95 | 6,000,000 |
| S3 | $6079.50 | 300,000,000 |

Unfortunately, Google Cloud Platform has no direct way of calculating the cost of the IoT solutions as they do not have exclusive IoT packages.

Arcwright team should be aware of these price models since it could directly impact on how we build our solution. The following is a few of the scenarios and comparisons of pricing model for Azure and AWS.

**Scenario I**

This scenario is referenced from the example from the AWS IoT pricing page. A separate calculation in Azure has been done for a direct comparison.

100 sensors each publish one 400-byte message per minute to AWS IoT. AWS IoT then delivers all of the messages to one other device.

In the calculation, 4.3 million messages are produced in each way per month; publishing and delivery. AWS claims that this will cost $21.50/month for publishing and $21.50/month for delivery; total of $43 monthly. ($5/1 million messages applied)

With Azure,

100 sensors \* 2 messages/minute \* 24 hours/day \* 60 minutes/hour = 28,800 messages / day

28,800 messages per day will be enough to be covered by Tier S1 of Azure which is $60.80/month. However, Azure allows the message size to be up to 4KB. What if those sensors produce 4KB messages instead of 400 bytes? It will increase the total number of messages by eight times in the AWS IoT since a single message block in AWS IoT is 512-byte. Monthly, in AWS, it would be considered as 34.4 million messages per month in each way costing $172/month in the AWS IoT. Therefore, in total, it would cost $344/month with the AWS IoT.

**Scenario II**

A sensor publishes one 512-byte message per second and 5000 sensors subscribe to the published message every minute.

In AWS,

**Publish**

1 message/second \* 30 days \* 24 hours/day \* 3600 seconds/hour = 2.6 million   
 messages/month

**Subscribe**

5000 messages/minute \* 30 days \* 24 hours/day \* 60 minutes/hour = 216 million  
 messages/month

**Total** 218.6 million messages \* $5/million = $1093 monthly

In Azure, total number of messages per day,

**Publish**

1 message/second \* 3600 seconds/hour \* 24 hours/day = 86,400 messages/day

**Subscribe**

5000 messages/minute \* 24 hours/day \* 60 minutes/hour = 7.2 million messages/day

It comes close to 7.3 million messages. In this case, Azure’s tier S2 price model would be used to calculate, which costs $607.95/month. The tier S2 supports up to 6 million messages/day per IoT hub unit. Since the scenario suggests that there are more than six million messages per day, it is required to have total of two units. The total cost will be $607.95 \* 2 = $1215.90/month. Which is slightly more expensive than using the AWS IoT solution. However, the size of each message could be up to 4KB for the Azure IoT hub. If it is possible to reduce the total number of messages per day by taking advantages of the larger message size, then it is significantly considerable to use Azure.

To summarize, the AWS IoT pricing model offers reasonable price for general cases. On the other hand, the Azure IoT Hub pricing model could be advantageous when the design of the payload is flexible in terms of the size and the frequency. Azure would cost less if the system utilizes maximum quota for each tier compared to the AWS IoT. Google Cloud Platform does not have per message billing model but it has an online cost estimation calculator available. GCP is useful for Arcwright to build a highly custom SDK for potential clients since there is no preset pricing model constraining the design.

# Provider Specific Features

In this section, each provider will be investigated separately to show its strength and weakness.

## Amazon Web Service IoT

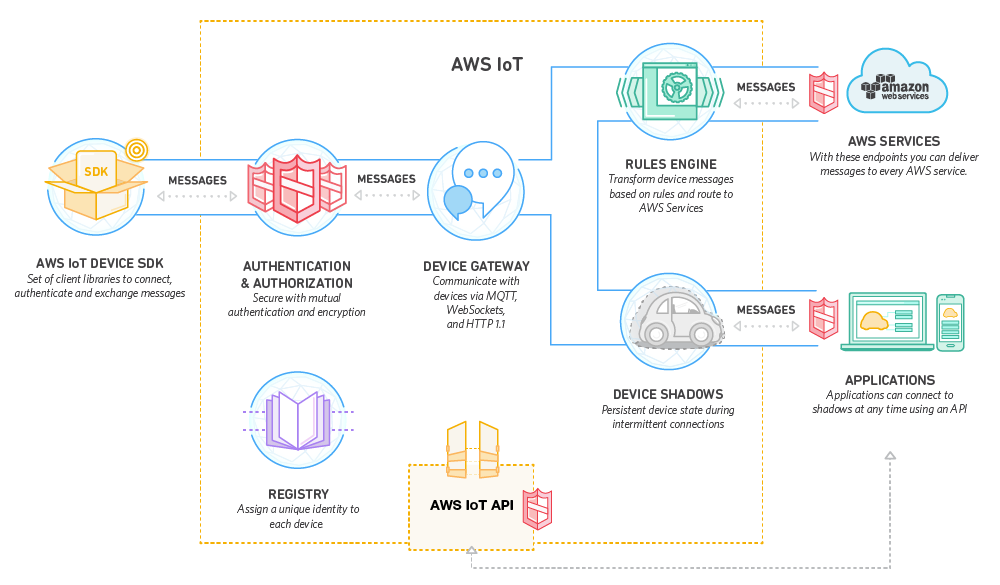


Figure 3-2. AWS IoT overview [**8**]

AWS IoT has Device Gateway directs messages to two different destinations; Rules Engine and Device(Thing) Shadows. Device(Thing) Shadow is a concept to communicate with the offline devices. The end-devices will follow its latest Device Shadow state in case the end-device was not online persistently. Rules Engine transform and route device messages to different destinations including other AWS services. One of the benefits of using AWS product is that they offer various other functional web services. For example, AWS Lambda offers simple way of processing logic without having a complicated infrastructure or AWS Kinesis provides pipeline for the stream of data generated from high amount IoT devices making it easy to analyze and collect the data. The entire AWS IoT suite is able to scale automatically. Another selling point for the AWS IoT platform is that all the messages delivers to the other AWS service are free of charge. Millions of messages generated by the end IoT devices can be stored in AWS S3 storage service without costing extra for the delivery to the storage system.

## Microsoft Azure

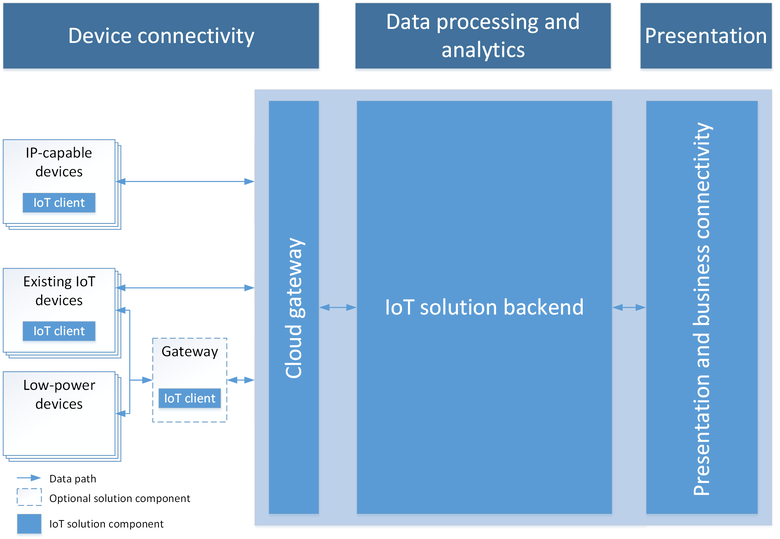


Figure 3-3. Azure IoT Hub [**9**]

Microsoft and Amazon shares the objectives for the IoT solutions. However, Azure’s approach to IoT is slightly different from Amazon’s. Azure IoT Hub, rather than using publish-subscribe communication model, enables bi-directional communication between devices and the cloud. Each device connected to Azure IoT Hub has two endpoints to interact with IoT Hub; Device to Cloud(D2C) and Cloud to Device(C2D). The former is used to send messages to the cloud from the device and the latter is used for the devices to receive commands. On the cloud side, the Hub operates similarly with two endpoints.

Being part of Microsoft product family, the Azure IoT Hub has great support for .NET framework and other Windows based product such as Windows 10 IoT Core. This is definitely a benefit towards the suite, since they also offer other programming languages to configure and use their product.

## Google Cloud Platform

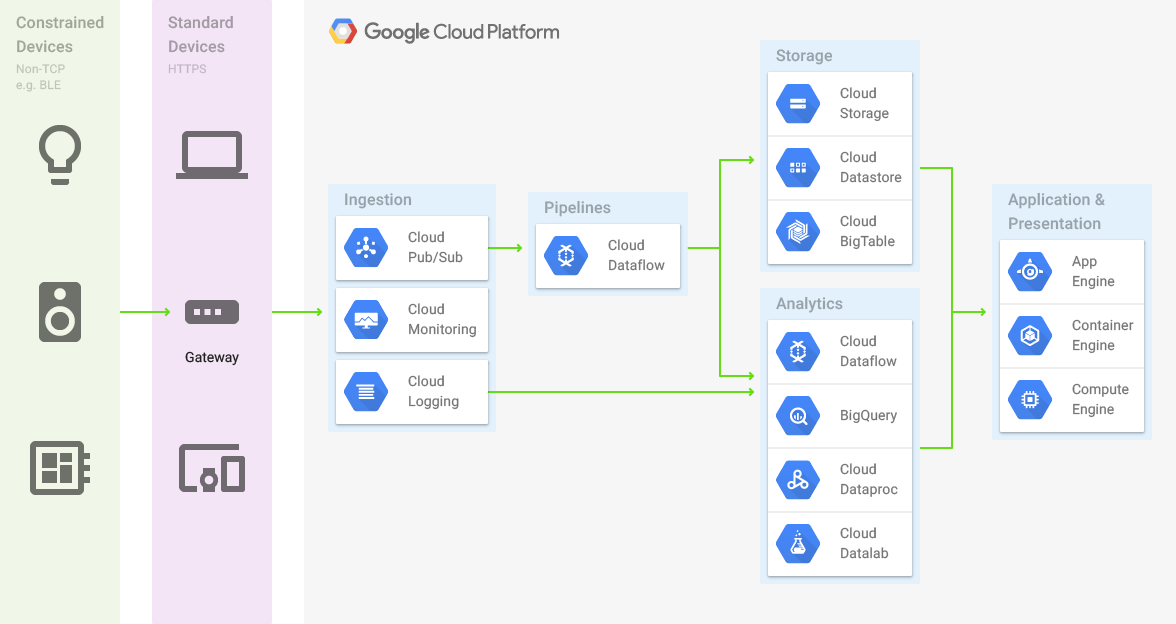


Figure 3-4. Google Cloud Platform for IoT [**10**]

Google’s approach to the IoT solution is by supporting communications between devices and cloud through Cloud Pub/Sub. Cloud Pub/Sub is highly scalable service adequate for the high volume IoT missions. It is worthwhile to note that GCP supports gRPC as the communication protocol.

Google has a number of other product that can be used for the IoT solution. Firebase is a mobile and IoT development platform owned by Google. It comes with SDKs for C++, JavaScript, iOS, and Android. Database, monitoring service, and operational logging also take parts in Firebase. Firebase is an alternative for some IoT solutions. Android Thing is an OS for the IoT end-devices which has a neat solution for over-the-air software and firmware update. Android Thing comes with built-in support for Weave, a communication platform designed to control and manage IoT devices. It seems like Google is putting efforts to make Weave the standard for IoT communication.

# Conclusions

From the investigation and the analysis of the three IoT platform providers, Google Cloud Platform has the least mature IoT ecosystem within their product suite. However, it definitely has space for flexibility with the potential market for Arcwright.

Meanwhile, Azure and AWS come very close to each other in term of the functionalities and the feature set. They both have good integration with the other products and offer a functioning analyzing tool for the stream data. Generally, AWS has a slightly cheaper pricing but the cost of using Azure could be highly optimized to lower than the AWS’s. AWS has an excellent authentication system that can reduce the workload for the developers using the AWS IoT suite. Azure offers integration with .NET framework which may be the only available option for some of the potential clients.

Since Connected Lab and Arcwright team has expertise and experience, it is concluded that the AWS IoT platform provides the team a fine solution.

# Recommendations

For the future roadmap for the Arcwright team, it is recommended to expand the SDK to Google Cloud Platform. GCP provides a number of other products that can be used for the team to build a customizable and scalable solution for the future potential clients. Firebase suits the team’s vision of having a mobile phone as the hub of the many IoT devices.

Recently, an engineer from Amazon visited Connected Lab to present AWS IoT platform. It is also recommended to invite an engineer from Azure to introduce the team to their suite.

# Glossary

**IoT:** Internet of Things

**AWS:** Amazon Web Services

**MQTT:** Message Queue Telemetry Transport

**REST:** Representational State Transfer

**HTTP:** Hypertext Transfer Protocol

**AMQP**: Advanced Message Queuing Protocol

**GCP**: Google Cloud Platform

**SDK**: Software Development Kit

# References

[1] D. Minoli, “Internet of Things Definitions and Framworks,” in *Building the Internet of Things  
 with IPv6 and MIPv6,* New Jersey: Wiley, 2013, pp. 33-35

[2] M. Miller, “Smart Connectivity: Welcome to the Internet of Things,” in *The Internet of   
 Things,* Indianapolis: Que, 2015, pp. 7

[3] AWS IoT Developer Guide. Security and Identity. [Online].  
 http://docs.aws.amazon.com/iot/latest/developerguide/iot-security-identity.html

[4] Microsoft Azure IoT Hub. Security from the ground up. [Online].  
 https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-security-ground-up

[5] Wikipedia. MQTT. [Online]. https://en.wikipedia.org/wiki/MQTT

[6] AWS IoT Developer Guide. Protocols. [Online]  
 http://docs.aws.amazon.com/iot/latest/developerguide/protocols.html

[7] DevExperience. Protocols: AMQP, MQTT, HTTP, and others. [Online].   
 https://paolopatierno.wordpress.com/2015/10/13/an-iot-platforms-match-microsoft-azure-iot-  
 vs-amazon-aws-iot/

[8] AWS IoT Platform. How it works. [Online]  
 https://aws.amazon.com/iot-platform/how-it-works/

[9] Microsoft Azure Blog. Developer’s Introduction to Azure IoT. [Online].  
 https://azure.microsoft.com/en-us/blog/developer-s-introduction-to-azure-iot/

[10] Google Cloud Platform. Overview of Internet of Things. [Online].  
 https://cloud.google.com/solutions/iot-overview

# Appendix A Amazon Web Service IoT Details

**TLS Cipher Suite Support**

* ECDHE-ECDSA-AES128-GCM-SHA256 (recommended)
* ECDHE-RSA-AES128-GCM-SHA256 (recommended)
* ECDHE-ECDSA-AES128-SHA256
* ECDHE-RSA-AES128-SHA256
* ECDHE-ECDSA-AES128-SHA
* ECDHE-RSA-AES128-SHA
* ECDHE-ECDSA-AES256-GCM-SHA384
* ECDHE-RSA-AES256-GCM-SHA384
* ECDHE-ECDSA-AES256-SHA384
* ECDHE-RSA-AES256-SHA384
* ECDHE-RSA-AES256-SHA
* ECDHE-ECDSA-AES256-SHA
* AES128-GCM-SHA256
* AES128-SHA256
* AES128-SHA
* AES256-GCM-SHA384
* AES256-SHA256
* AES256-SHA