

IT UNIVERSITY OF COPENHAGEN

SOFTWARE DEVELOPMENT BSc

BACHELOR PROJECT

ReRide IoT platform

Author:

Anders Edelbo LILLIE

Supervisor:

Tomas SOKOLER

February 20, 2017

Abstract

Lorem ipsum

Contents

1	Introduction	1
1.1	Background	1
1.2	Prerequisites	1
2	Problem description	1
3	Problem analysis	1
3.1	Internet of Things	1
3.2	Cloud based IoT platform	1
3.2.1	Google Cloud Platform	1
3.2.2	Amazon Web Services	3
3.3	Microsoft Azure	5
3.4	IBM Watson IoT	6
3.5	Alternatives	8
4	Method	8
5	Example	8
6	Tehcnical descreption	8
7	Discussion	8
8	Conclusion and reflection	8
9	References	8
10	Appendix	9

1 Introduction

This section introduces the paper, its background and prerequisites. The project is build as an extension to ReRide[2].

1.1 Background

1.2 Prerequisites

2 Problem description

3 Problem analysis

3.1 Internet of Things

3.1.1 Origin

3.1.2 Spime

3.1.3 Cloud-based computing

3.2 Cloud based IoT platform

3.2.1 What is a platform?

3.2.2 What is a service?

3.2.3 Google Cloud Platform

Overview *Cloud Platform brings scale of infrastructure, networking, and a range of storage and analytics products you can use to make the most of device generated data.*[4] This is Google's own description of their IoT Cloud solution, and the references article is used for analysis in this section.

The platform can be divided into three basic components, the device, gateway, and cloud, each of which are characterized by the following: A *device* is defined as hardware and software that directly interacts with the world, and are connected to a network and the Internet. A *gateway* lets devices *not* directly connected to the Internet reach a cloud service. The *Cloud* is the *endpoint* of device data, where it can be processed.

Several terms are used to describe what devices constitute, but the most important seems to be metadata and state information. Device metadata contains information such as ID and the date manufactured. The state information is basically the device status, such as "on"/"off". Furthermore devices must be associated with some security credentials for authentication. *Telemetry* is the term for device data gathering, and is described as the *eyes-and-ears data* the device collects in its environment. Once a device has been defined, telemetry can be sent to the Cloud Platform. As mentioned, a gateway can provide

connection to the cloud, when e.g. protocol translation is needed. The Cloud Platform is the infrastructure for data management, which is depicted in figure 1.

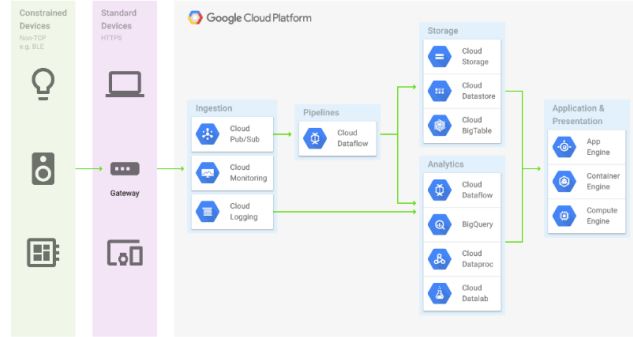


Figure 1: Diagram of the Google Cloud Platform IoT infrastructure

Google calls each of these components *services*, and are categorized by their functionality and role in the infrastructure. Communication between the services are called *pipelines*, which includes transforming, computing, combining, and moving data. The Cloud Platform also provides means for storing data in Google Firebase[3] and is especially used for continuously mirroring device states and exposing them to clients (See figure 2).

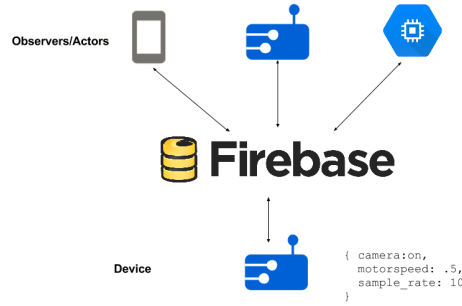


Figure 2: Abstract functionality of Google Firebase in the IoT Cloud Platform

The most interesting services are the "rule processing" and "streaming analytics", which allows for programmers to write custom logic for handling data triggered by device events. What you can do is to essentially trigger alerts, filter data, and invoke other APIs. This property of quick data processing is described as *key* in the Cloud Platform. The "Cloud Dataflow" service provides means for *more sophisticated* analytics.

Services Google Cloud Platform is not just for IoT, but provides a plethora of products. The following is a list of the categories and *some* of the services:

- Compute: Compute Engine, App Engine, Cloud Functions
- Storage & Databases: Cloud Storage, Cloud SQL
- Networking: Cloud Virtual Network
- Big Data: Cloud Pub/Sub, BigQuery
- Machine Learning: Cloud Machine Learning Platform, Cloud Natural Language API
- Identity & Security: Cloud IAM, Cloud Resource Manager
- Management Tools: Monitoring, Logging, Trace, Cloud APIs
- Developer Tools: Cloud SDK, Cloud Deployment Manager

Some of the above mentioned services are self-explanatory, and some are just listed for reference. The IoT solution is a combination of only some of these services, and form the infrastructure explained in the overview.

3.2.4 Amazon Web Services

Overview *AWS IoT is a managed cloud platform that lets connected devices easily and securely interact with cloud applications and other devices.*[1]. Aside from the huge and popular web shop Amazon hosts, they have developed a huge library of web services, called "Amazon Web Services" (AWS). Above citation is from their website and describes one of their products; The "AWS IoT Platform".

AWS IoT Platform constitutes of several components, labeled "AWS IoT Device SDK", "Device Gateway", "Authentication and Authorization", "Registry", "Device Shadows", and "Rules Engine". The infrastructure is depicted in figure 3.

The *AWS IoT Device SDK* provides software development kits to connect hardware devices or mobile applications, and supports C, JavaScript, and Arduino. Protocols like MQTT, HTTP, and WebSockets are supported for connectivity and data exchange. Furthermore Amazon has partnered up to offer a wider variety of SDKs, including Google's Android and Apple's iOS. The SDKs are meant for making integration with AWS easier for the programmer by encapsulating operations on the IoT Platform in library function calls.

The *Device Gateway* provides before mentioned protocols for communication between devices and the IoT Platform. It can be configured to work as a publication/subscription model for one-to-one and one-to-many communication, and it even scales automatically with *over a billion* devices connected.

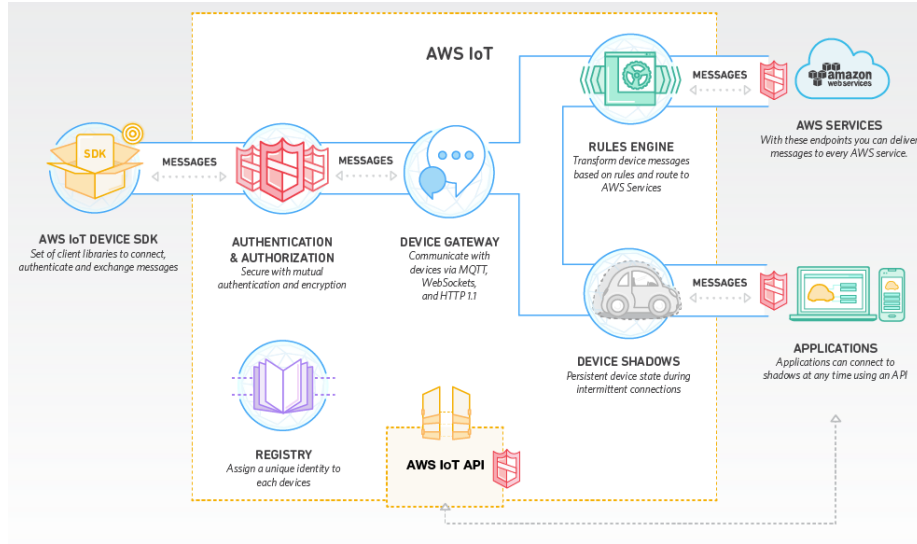


Figure 3: Components of AWS IoT Platform and the infrastructure

Security is taken seriously by Amazon, and this is what *Authentication and Authorization* takes care of. Data between device and IoT Platform is never exchanged without proven identity, by use of "SigV4" and X.509 certificates. It's possible to fully manage mapping of authentication policies for connected devices, as well as revoking access rights at any time. Another feature is to use Amazon Cognito, which basically takes care of authentication for mobile applications.

The *Registry* tracks device metadata by establishing unique identities for connected devices.

Device Shadows are persistent, virtual versions of connected devices and represent their states. This allows for other devices to interact with other devices through a REST API. In other words, it is a convenient way of updating devices, as well as getting information about them.

The data processing happens in the *Rules Engine*, which evaluates messages passed to IoT Platform by user defined rules. The data can simply be transformed and passed to another device, or routed to other AWS cloud services for further processing. This includes several of the AWS services like AWS Lambda, S3, and DynamoDB.

Figure 4 is an example of the IoT Platform and illustrates how a light bulb can be controlled through the above mentioned services. A mobile application sends instructions to a controller, which communicates with the IoT Platform in the cloud. Two rules are defined to act upon triggered events. One rule routes the data to other AWS services for evaluation, and the other changes the light color of the bulb. In either case the device shadow is updated with a new state. If

the light bulb is turned off, requested device states are still saved in the device shadow.

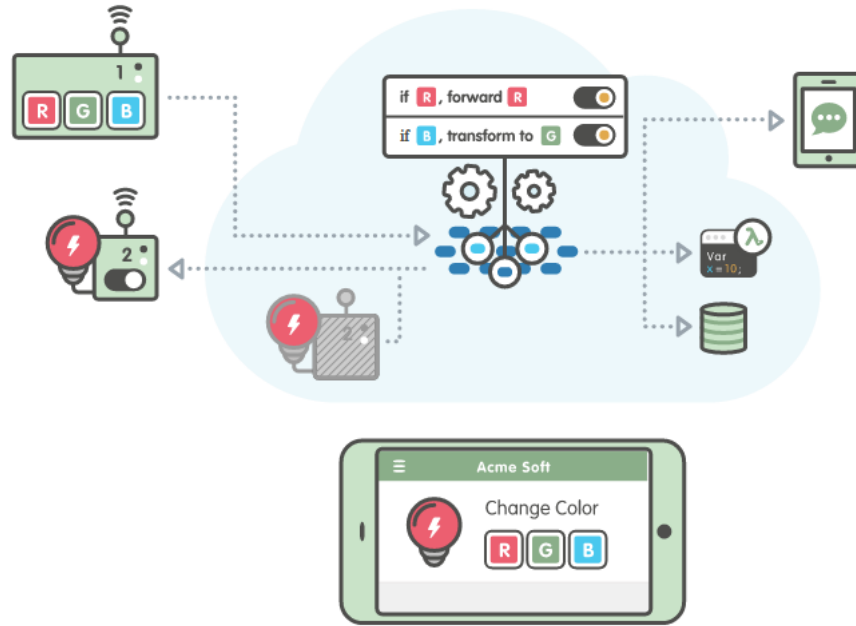


Figure 4: An AWS IoT Platform example illustrating the workings of the services.

Services As mentioned in the overview, the Rules Engine allows for inbound data to be routed to other services. One worth mentioned is AWS Lambda, which executes computations in the cloud. It is even possible to write code in e.g. C# and upload code snippets to the project. This service has the property of removing the need for a server, as well as automatically scaling with the size of the workload. Other services like "DynamoDB" offers a NoSQL cloud database, and "Kinesis" for data analytics.

3.2.5 Microsoft Azure

Azure IoT Hub is a fully managed service that helps enable reliable and secure bi-directional communications between millions of devices and a solution backend. REF(<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide>)

Microsoft is another big player in the field of IoT Cloud computing, and offers their own IoT platform. Much confusingly they offer an IoT Suite, as well as an IoT Hub. The difference is that the IoT Suite is a collection of pre-configured solutions REF(<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-devguide>)

hub-what-is-azure-iot). This means that they have actually made customizable solutions based on common IoT scenarios, such as remote monitoring and asset management. The IoT Suite is however also the platform including the IoT Hub, which is the most important service in the IoT infrastructure, as it takes care of bidirectional connection with devices and the cloud. Figure REF(<https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-what-is-iot-hub>) illustrates the infrastructure of the platform from a device to the back end services.

The IoT Hub keeps records of so called "Device twins", which are JSON documents containing device state and metadata information, for all connected devices. An identity registry enables devices to authenticate securely to the IoT Hub, which allows for full control of connecting devices. Devices communicate with the cloud through a device SDK or *gateway*. The gateway is meant to be used when the offered SDKs do not support the device. Two types of gateways are offered, one being the *protocol* gateway and the other being the *field* gateway. The former is deployed in the cloud and performs protocol translation for MQTT, AMQP, and HTTP. The latter is deployed locally, i.e. physically close to the device(s), and enables also protocol translation, but also has the property of being able to actively manage devices, and the information mediated to the cloud.

Services Returning to the IoT Suite, a handful of services like Azure Stream Analytics, Storage, DocumentDB, Web Apps, and Power BI are available for integration with the IoT Hub. Together they provide means for data analytics, storage, visualization, and integration with back-office systems. REF(<https://docs.microsoft.com/da-dk/azure/iot-suite/iot-suite-overview>). Telemetry can be routed to an Azure service by user-defined rules in the IoT Hub, and requires no code. As mentioned the IoT Suite contains preconfigured solutions, which are labeled "Remote monitoring" and "Predictive maintenance". These solutions are the way to start building an IoT platform, and are customizable and extensible.

3.2.6 IBM Watson IoT

Watson IoT Platform is designed to simplify cognitive IoT development so you can harness the full potential of the Internet of Things REF(<https://www.ibm.com/internet-of-things/>).

IBM has a lot of different domain specific products and solutions in regards to IoT. They offer different solutions based on the business application, e.g. "Asset management", "Facilities management", and "Product development" REF(<https://www.ibm.com/internet-of-things/iot-solutions/>). They are all hosted on the IBM Bluemix cloud platform, which essentially is a big collection of offered products and services for cloud computing. One of them is the Watson IoT Platform, which provides a "powerful application access to IoT devices and data to help you rapidly create analytics applications, visualization

dashboards, and mobile IoT apps” REF(<https://console.ng.bluemix.net/docs/starters/IoT/iot500.html>). The infrastructure of the platform is depicted in figure 5.

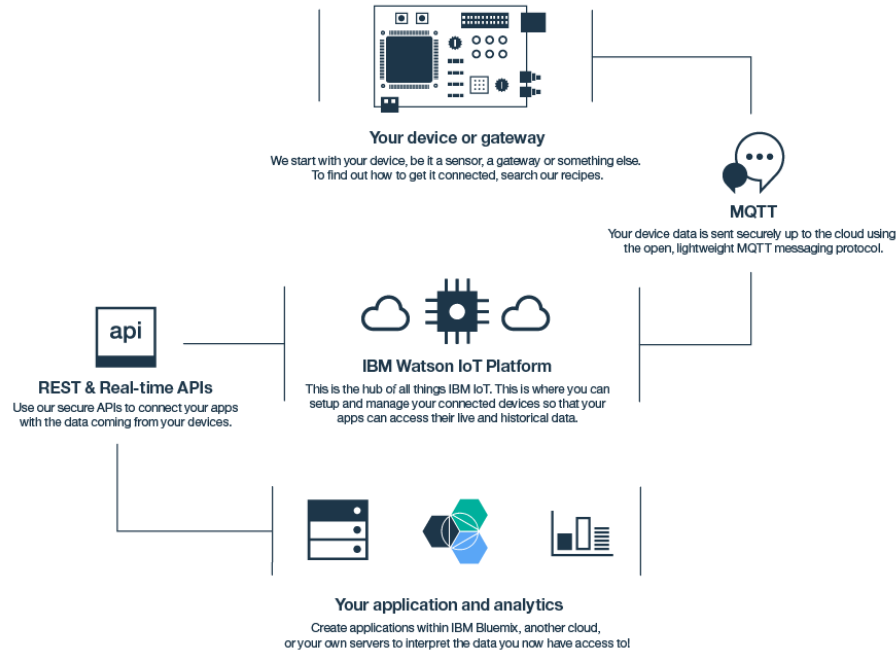


Figure 5: Infrastructure of the Bluemix service "Watson IoT Platform"

The Watson IoT Platform is described through six different concepts: organizations, devices, gateways, applications, events, and commands REF(<https://console.ng.bluemix.net/docs/service>). An organization is a way of separating systems and ensure data security through a unique identifier. This way, devices and applications are logically bound to a single organization and communication with other organizations can only happen with an explicit application for this purpose. Devices are divided into two classes; *managed* and *unmanaged* devices. The former contains a device management agent, which allows interactions with the Watson IoT Platform Device Management service. Such devices can perform operations which unmanaged devices cannot. This includes device management request and device management operations such as updates, downloads and reboots. Devices that cannot connect directly to the Watson IoT Platform can do so through a gateway. A gateway is both a device and an application that serves as an access point for other devices. Watson IoT Platform uses the HTTP and MQTT protocols for all communication. All devices and gateways must be registered in the Watson IoT Platform before they can connect to a service. Device manipulation and data interaction is controlled through applications, which do so with an API key and a unique application ID, all tied to a specific organization. Any application can subscribe to an event, which is triggered by a device and published to the

Watson IoT Platform. Conversely, applications can communicate with devices through commands. The overall architecture of the Watson IoT is depicted in figure 6, which resembles figure 5, but is more elaborate

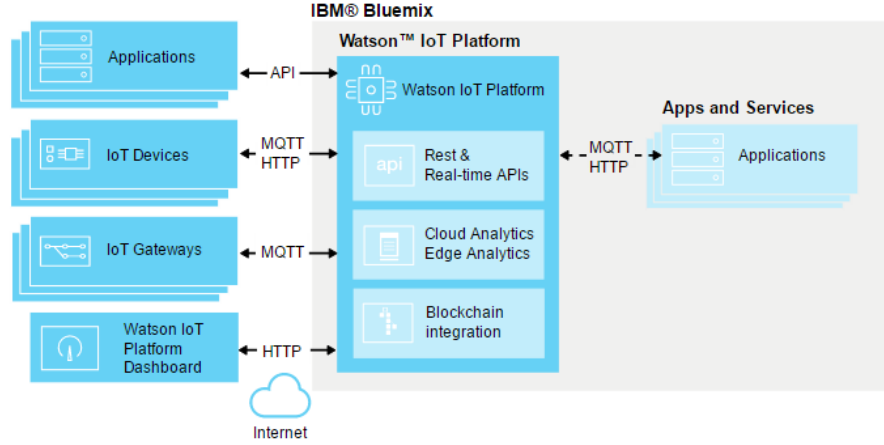


Figure 6: Architecture of the Watson IoT platform

Services Watson IoT Platform provides a way of representing data set values of different devices in the GUI for a quick overview, through "boards and cards". Analytics rules are defines to specify the conditions that trigger actions. *Cloud* rules trigger rules for devices connected directly to Watson IoT Platform, while *edge* rules trigger rules for devices connected through gateways. Rules can trigger actions, which can be either of the following types: "Send email", "IFTTT", "Node-RED", "Webhook".

Developing for Watson IoT Platform is supported by several SDKs and languages, such as C, C++, C#, Node.js, and Java. Furthermore "Node-RED" is supported for the platform, which is an integrated visual tool to develop applications, devices, and gateways.

As Watson IoT Platform is one of the IBM Bluemix services, it can be integrated with other *compatible* services that are hosted on Bluemix, e.g. for storage with "Cloudant NoSQL DB for Bluemix" or data to blockchain transactions with "Bluemix Blockchain" REF(<https://console.ng.bluemix.net/docs/services/IoT/reference/extensions/index>)

3.2.7 External IoT services

PubNub

IFTTT

3.2.8 Comparison

What is offered?

3.3 Alternatives

4 Method

5 Example

6 Tehcnical descreeption

7 Discussion

8 Conclusion and reflection

9 References

References

- [1] Amazon. AWS IoT Features, February 2017. <https://aws.amazon.com/iot-platform/how-it-works/>.
- [2] Naveen L. Bagalkot, Tomas Sokoler, and Suraj Baadkar. ReRide: Performing Lower Back Rehabilitation While Riding Your Motorbike in Traffic. *PervasiveHealth*, 2016.
- [3] Google. Firebase, February 2017. <https://firebase.google.com/>.
- [4] Google. Overview of Internet of Things, February 2017. <https://cloud.google.com/solutions/iot-overview>.

10 Appendix