UNIVERSITI TUNKU ABDUL RAHMAN

ACADEMIC YEAR 2022/2023

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY (PERAK)

UCCD3113 DISTRIBUTED COMPUTER SYSTEMS

TOPIC: SMART GARBAGE COLLECTION SYSTEM

Name	Student ID	Course
Cheow Yue Chen	1902365	IA
Gan Win Sian	1702594	IA
Kesavan a/l Muniandy	1700867	IA
Yap Jheng Khin	1800224	CS

Table of Contents

Chapter 1: Introduction

1.1 Background of Smart Garbage Collection System

The smart garbage collection system is a specially designed approach to dispose of the garbage effectively that solves the social problems of hygiene in the country. The system uses sensors for the identification of personnel and measuring garbage levels. The smart garbage bin offers an automated lid to personnel. The display of the rubbish bin provides staff with continuous data on garbage levels. Personnel has access to an automatic cover on this smart waste bin. When workers are detected within 30 cm of the waste bin, the lid opens if the rubbish bin is not complete, but it stays closed. Inversely, when the garbage bin is full or is 100% filled up, it also sends a message to the appropriate authorities, instructing them to collect the rubbish. Arduino Uno synchronizes the whole automated system. In a small-scale bin, an actual prototype of the smart waste management system was constructed and proven adequate.

Moreover, the bin is designed using various sensors and modules. For instance, it consists of a waste collection tray at the top where the user throws the rubbish into the bin. The garbage collection tray consists of a moisture sensor identifying garbage types. Two servo motors are fitted with the waste collection tray to fill the tray towards the wet collection bin or dry collector bin. When the users throw the waste on the tray, the waste will be categorized using the Moisture Sensor, and the appropriate servo motors begin to tilt the tray to dump the waste in dry/wet bins. Then, the bin is fitted with an IR proximity sensor on its side at a moderately full level. When the trash is collected to this level, the sensor reading activities the GPS module, and through the Wi-Fi-Module, the GPS module interacts with the cloud server and informs its current location.

Nowadays, one of the most significant issues in highly populated metropolitan areas is garbage management (Sohag and Podder, 2020). Because of environmental pollution, it is becoming increasingly impossible to live a healthy, sustainable life in metropolitan settings. Due to a lack of an appropriate waste management strategy, issues such as garbage overflow arise, causing severe environmental damage. Polluted environments facilitate the spread of different illnesses in epidemic form. Nonetheless, open dumping and rubbish pickup at open dumpsites pose major health dangers, such as skin infections and chronic ailments (Sohag and Podder, 2020). As a result, appropriate smart garbage management should be a top concern to decrease pollution and protect public health. The management or government must emphasize proper waste management to guarantee a healthy environment and long-term growth.

In summary, the garbage system is important and must be implemented effectively to provide a relatively basic, eco-friendly management system. The technique is far more user-friendly than traditional garbage management methods. It helps to reduce trash overflow and spread using IoT-based applications. In addition, the device has an automatic cover that prevents any personal interaction with the waste container. It may also cause a significant shift in people's attitudes regarding chaotic urban living.

1.2 History of Smart Garbage Collection System

In current world, many countries adopting technology-based garbage collection systems. The term "Smart Garbage Collection System" refers to all the actions involved in tracking waste generated in a city, from when citizens produce waste to collection, transportation, and final disposal. However, there has been a lot of evolution in the waste management field prior to the smart garbage collection system. The first rubbish can was developed in England in the year 1875. The ashes of burning rubbish were placed in this container. This receptacle was emptied on a weekly basis, and each resident was charged a price regardless of whether their trash was empty or full (Czermak, 2020). The creator of the plastic garbage can is Charles Harrison. Harrison downplayed the significance of this discovery, preferring to focus on the development of the plastic waste can. He set out to create a more robust, easier-to-transport plastic garbage can on wheels

The smart system's recent development in developed country such as China, the government released a "zero waste" plan with the goal of having 35 percent of garbage recycled and a ban on plastic bags in place by 2022. The "Internet Plus Recycling Model" which incorporates modern technology in garbage sorting promoted in this plan. Firstly, technologies and gamification were utilised to teach people about waste sorting. Users are encouraged to participate in order to earn points and incentives. Virtual reality video game machines with free access were deployed in public locations in Shanghai to imitate sorting experiences. Users wear a virtual reality headset and are presented with four trash recycling bins, with the goal of throwing the waste that appears in the correct bin to earn points (SQLI Digital Experience, 2020). Second, Alibaba has created a waste recognition system based on artificial intelligence through its e-commerce platform Taobao. The software allows users to take a photo of trash and have it matched to the appropriate container. By recycling their packaging at one of Alibaba's recycling stations, users of Alibaba's e-commerce platforms can earn points that can be used towards future orders. Customers scan a QR code, which turns into "green points."

The Xiaohuanggou (XHG) Group, for example, has installed intelligent waste-management recycling bins around China to encourage individuals to sort and recycle in exchange for rewards. The smart garbage bins are simple to use; consumers simply scan a QR code on the container with the XHG app before discarding their stuff (Lin, 2020). Their rubbish will be weighed by the intelligent bin, and points will be awarded accordingly.

1.3 Future trend of garbage collection system

First of all, first future trend of the garbage collection system is there will no more regular garbage bin, and everything will turn to self-service kiosk (BigRentz, Inc., 2021). The wastes like paper, plastic, food, and exacta, or some of the waste like light bulb, electronic device, batteries, and others will need the people to self-collect and recycle themselves. Some of the regular items like tins and plastic can change into money, the system may get monetary to the user who use garbage collection system.

For the second future trends is there will no more garbage trucks collect garbage one by one house, the underground waste collection systems will replace it (Envac, 2021). The underground waste system can help government or country save budget on purchasing trucks and the petrol. For example, Envac group an innovative global automated waste collection industry. They set out to design a garbage waste system that can handle 50 tons of the waste per week by using airflow. This system will no need require any presence of waste collection operatives. Envac's pneumatic trash system are now default infrastructure in 44 cities (Rivero, 2021). This system is solving the trash problem in these 44 cities.

And the third future trend is recycling for garbage will fully automated. As the technology are improving all the time, Artificial intelligence (AI) and robot become more mature, because the error rate are reducing and the cost for produce the AI and labours are also decrease, so it makes this trend easier to widespread. These AI and robots will help garbage facilities like recycling centre to recycling materials in longer time, less error recycling and reduce worker's working pay rate. Another advantage is it can prevent most worker injury problem and sick problem (Barker, 2021). Last, the AI and robots can real time monitoring all the data to prevent the loss of recyclables to landfill, it also can diagnose the gaps to processing more efficiency (Barker, 2021).

1.4 Problem Statement

The garbage collection system is inefficient in Danish House Student Hostel. Each garbage bin is only pick up by employees once per week even though the bin is already full for a few days. Most students are forced to put the trash outside of the garbage bins since the bins are always full. As a result, the area becomes a breeding ground for pests like cockroach and Aedes. Numerous maggots crawl around the bin, dead cockroaches can be found everywhere in the stairway, surgical masks get thrown outside of bin and flown away... This inevitably will cause health problems to the students.

1.5 Objective and Proposed Solution

To solve the problem, smart garbage collection system is implemented for Danish House Student Hostel in Kampar. The system will update the records in the cloud to inform which bin is already full on an hourly basis. As a result, the garbage workers will be spared from manually checking the bins one by one and only collect garbage from bins that are full. Besides, the system will also collect the garbage bin’s fill level data on a daily basis and send it to the cloud. The telemetry is processed to extract insights that can be used to improve the garbage collection system. Furthermore, the garbage bin can internally sort the trash items into partitions using a machine learning model. The automation of trash sorting can reduce the burdens of recycling workers in the material recovery facilities.

Chapter 2: System Design

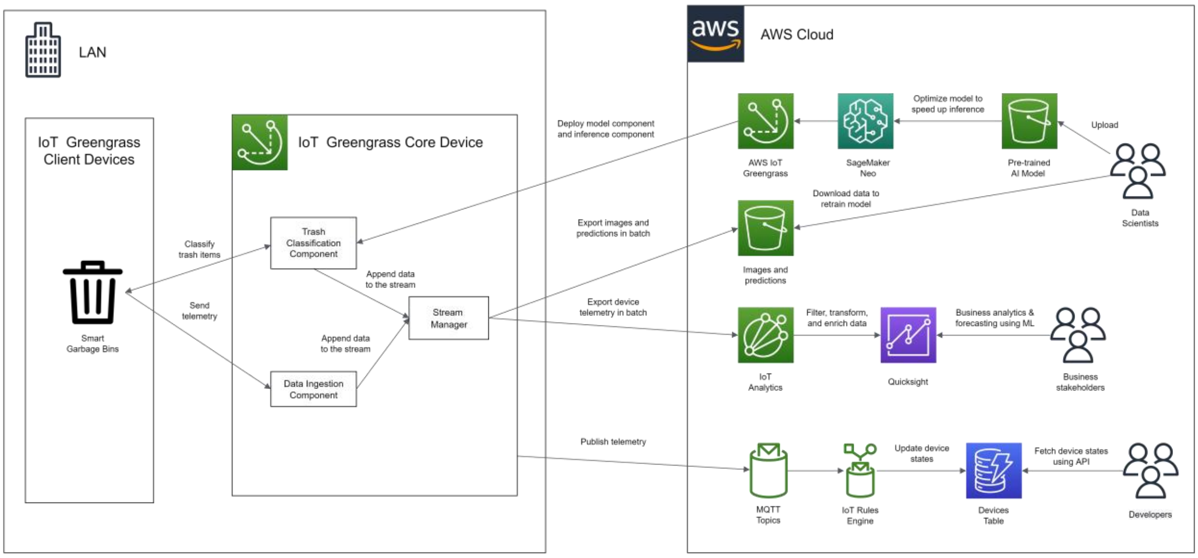

Figure 1: The full view of architecture diagram.

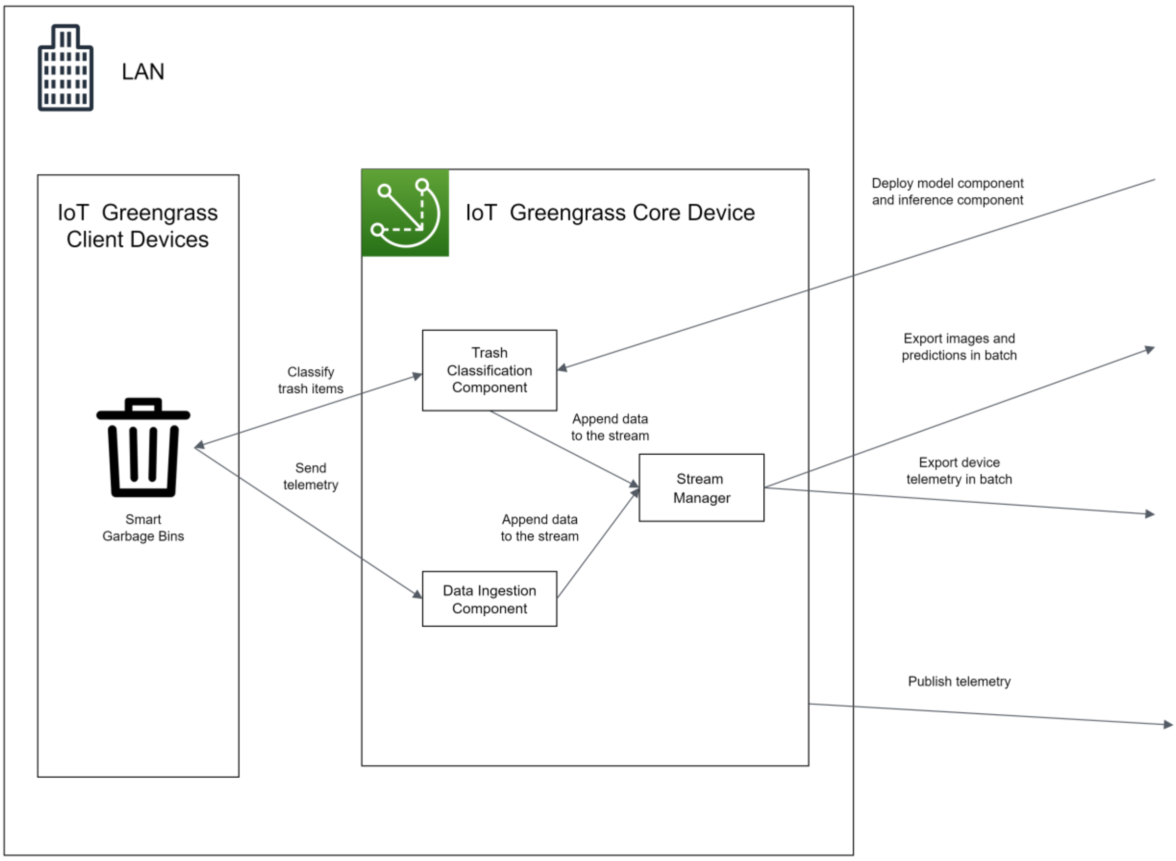

Figure 2: The enlarged left-side view of architecture diagram.

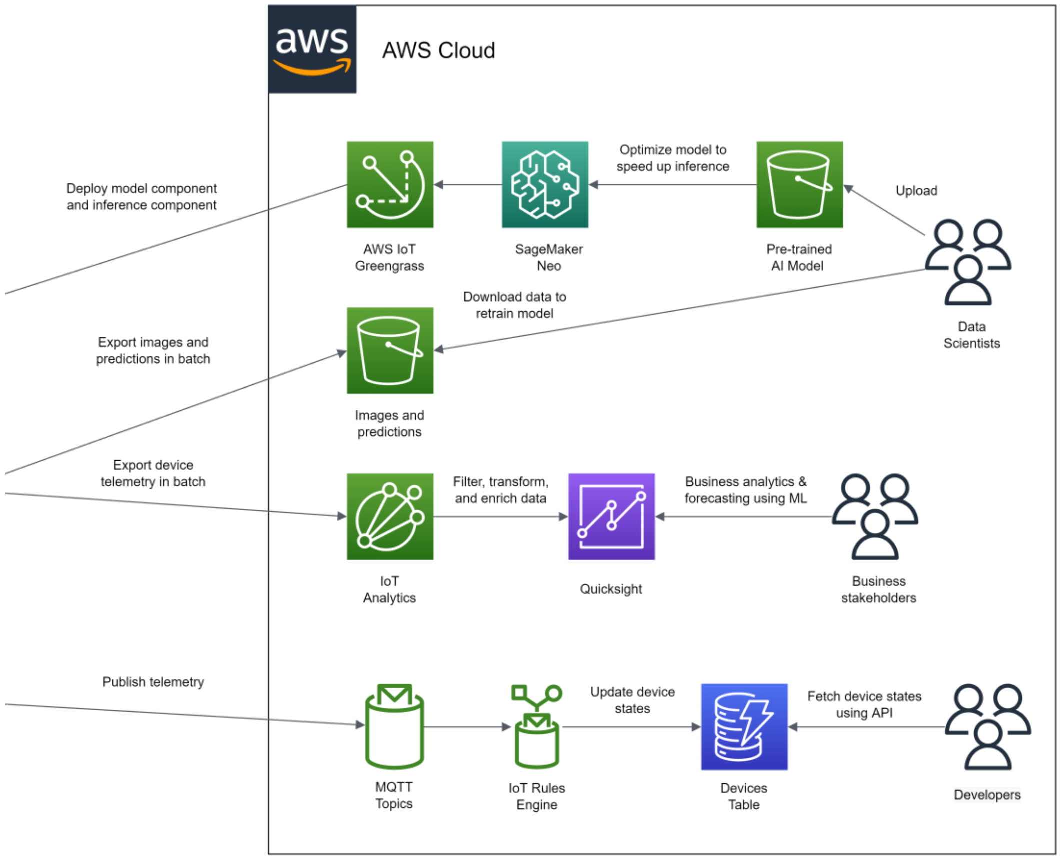

Figure 3: The enlarged right-side view of architecture diagram.

2.1 AWS IoT Greengrass

AWS IoT Greengrass comprises of cloud services and an open-source edge runtime named AWS IoT Greengrass Core software. The local network environment can be comprised of edge gateway, devices that are installed with core software (also known as core device), and client downstream devices. Normally, client downstream devices act as the low-spec sensors and actuators that only capable of collecting IoT telemetry and performing specific actions, while the edge gateway or core devices handle heavier loads such as model inference, model retraining, and exporting telemetry to the cloud. With AWS IoT Greengrass Core software, each functionality can be deployed to the core devices and installed in an isolated component and the cloud service is used to monitor the deployment and health status of the core devices. These Greengrass components can either be provided by AWS or custom built by the IoT developers. Taking one of the AWS-provided components as an example, the MQTT bridge component allows the local MQTT communication between client devices and core devices to send and receive lightweight telemetry, command request, and command response. In this solution, two main AWS-provided components are required, which are stream manager component and MQTT bridge component. In addition, two custom components are developed to classify trash at the edge and process raw telemetry data at the edge before sending to the cloud, respectively.

2.2 Trash Classification Component

The first custom component is the trash classification component. When a garbage bin detects a trash item, it will send the captured image and the request payload to the trash classification component in the core device. Then, the trash classification component will fetch the image, perform the inference, and send the prediction result back to the garbage bin. The component will also filter and aggregate the images and model predictions before appending to the stream in order to be exported to the S3 bucket. In order to perform the inference, the trash classification component will need the inference code, the neo-compiled trash classification model and execution environment. First, a custom component is created to install and manage the package dependencies that are required to run the inference. Second, another custom component is created to load the model from the S3 bucket. Third, the trash classification component is created to run the inference code in the execution environment that is loaded with the required package dependencies.

2.3 Data Ingestion Component

The second custom component is the data ingestion component. The garbage bins will send the fill level telemetry to the data ingestion component on an hourly basis. The data ingestion component then receives the telemetry, preprocess and filter the telemetry, before appending the telemetry to the stream in order to be exported to the AWS IoT Analytics channel. A custom component is required to install and manage the package dependencies that are required to run the data ingestion code.

2.4 Stream Manager Component

The first AWS-provided component used is the stream manager component. The stream manager component is an AWS-provided component that allows efficient, reliable, and cost-efficient transfer of IoT data to the AWS cloud. The transfer is efficient since the data is aggregated and send in batches. The transfer is reliable since the component can guarantee the data delivery even if the internet connectivity is poor by configuring options like bandwidth use, timeout behaviour, and data persistence (Amazon Web Services, Inc., 2022). The transfer is cost efficient since the data can be analysed, preprocessed, and compressed at the edge to optimize the network payload size and cloud storage cost. For example, the component can choose to only append the compressed images and predictions with low confidence scores to the stream for model retraining. In this solution, the stream manager component is used to export trash images and inference results to the Amazon S3 bucket. Then, the data scientists can download the files from the S3 bucket to evaluate or retrain the model. Besides, the stream manager component is used to export the fill level telemetry streams in bulk to the AWS IoT Analytics for further analysis.

2.5 MQTT bridge component

The second AWS-provided component used is the MQTT bridge component. As the name suggests, the MQTT bridge component bridges multiple MQTT brokers to allow communications that otherwise will not happen. For a client device to communicate with AWS IoT Core via the core device using MQTT, the MQTT messages must be routed to and from the Local Moquette MQTT broker and AWS IoT Core MQTT broker. For a client device to communicate with Greengrass components installed in the core device using MQTT, the MQTT messages must be routed to and from the Local Moquette MQTT broker and Local publish/subscribe messaging IPC service. To enhance the security, the MQTT bridge component also support the authentication and authorization of the client devices using another component called client device auth component. In this solution, the bridging between Local Moquette MQTT broker and Local publish/subscribe messaging IPC service is set up such that the client devices, in this case, the garbage bins can send the fill level telemetry, send the inference request, and receive the inference request to and from the core device. Besides, the bridging between Local Moquette MQTT broker and AWS IoT Core MQTT broker is also set up so that the client devices can send the fill level telemetry on an hourly basis.

2.6 AWS IoT Analytics

In this solution, the stream manager Greengrass component exports the fill level telemetry to the AWS IoT Analytics for further analysis. AWS IoT Analytics streamlines the process of analysing data from IoT devices. Before saving IoT data in a time-series data repository for analysis, AWS IoT Analytics filters, converts, and enhances it. The service may be configured to gather only the data required from the devices, process it using mathematical transformations, and enrich it with device-specific metadata like device kind and location before storing it. The data can then be analysed using the built-in SQL query engine, or more complicated analytics and machine learning inference can be performed. In addition, data visualisation is also possible with AWS IoT Analytics because to its interaction with Amazon QuickSight. Amazon QuickSight is directly integrated with AWS IoT

Analytics. Amazon QuickSight is a business analytics solution that allows users to quickly create visualisations, do ad-hoc analysis, and extract business insights from the data. It allows businesses to scale to hundreds of thousands of users while maintaining responsive performance thanks to a powerful in-memory engine (SPICE).

2.7 Amazon QuickSight

In this solution, business stakeholders use Amazon QuickSight to extract high level business insights that are useful in optimising the garbage collection process to reduce cost and improve profits through recycling. Amazon QuickSight uses Amazon's proven technology to execute ML-powered anomaly detection on millions of metrics and billions of data points on a continuous basis. This anomaly detection can obtain deep insights that are often buried in aggregates, hidden from view, and not scalable with human analysis *(Amazon Web Services, 2019).* When abnormalities are discovered, Amazon QuickSight sends out timely email warnings to stakeholders, outlining the data in the cloud and informing them which bins are already full and need to be collected on an hourly basis. The data is analysed to derive insights that can be used to improve the garbage collection system. Stakeholders, for example, obtain business knowledge into altering the size of the rubbish bin based on how frequently the bin fills. They can see which housing areas or floor numbers require regular waste collection. Stakeholders can now easily estimate their critical business metrics with just a few clicks thanks to Amazon QuickSight's ML-powered forecasting and what-if analysis. Amazon QuickSight's built-in machine learning technology is designed to handle difficult real-world scenarios.

Amazon QuickSight employs machine learning to produce more accurate forecasts than traditional methods. For example, using garbage classification inference with various levels of seasonality and a time window of a week or month, they may forecast the type of trash objects sorted. Besides that, stakeholders can use ML-powered forecasting to conduct interactive what-if assessments to predict the growth trajectory required to meet demand for bins and optimise operations (Amazon Web Services, 2019). Amazon QuickSight uses auto-narratives to automatically read the charts and tables in the stakeholders' dashboard and deliver a number of natural language suggestions. They may receive ideas such as what the day-over-day changes look like, what the highest type of garbage collected was, what the growth rate of using smart garbage bins is, and when the forecast indicates the highest garbage collection date, depending on the shape and form of their data.

2.8 Amazon SageMaker Neo

Amazon SageMaker Neo reduces the burden of developers by automatically optimizes the machine learning models for cloud and supported device that at the edge on SageMaker. Amazon SageMaker Neo model compilation can ensure any model to optimally perform inference across multiple cloud instances and edge devices regardless of different hardware and software specification (Amazon Web Services, Inc., 2022b). In this solution, the trash classifier is directly obtained from the GitHub repository that is owned by Woo (2022). The trash classifier is uploaded to S3 bucket and the SageMaker Neo service will fetch it during the Neo compilation. Then, the trash classifier is optimized and compiled and stored it back in the S3 bucket. To use the compiled model for inference, a custom Greengrass component is developed and deployed to the core device to download the model from the S3 bucket to a local directory in the core device. Finally, the trash classifier component imports the model into the program to perform the inference.

2.9 Amazon DynamoDB

Amazon Dynamo DB is a fully managed NoSQL database service that offers seamless scalability and consistent performance. DynamoDB offloads the administrative responsibilities of running and managing a distributed database, so users don't have to worry about hardware provisioning, setup, configuration, replication, software patching, or cluster scalability. DynamoDB also offers encryption at rest, which cuts down on the time and effort needed to protect sensitive information.

There are several functions of Amazon DynamoDB. In Dynamo DB, tables, items, and attributes are the main elements that the users need. Dynamo D utilizes primary keys to uniquely determine each item in the table and secondary indexes to offer more flexibility. For example, it aids in the creation of database tables that can save and retrieve data, at any level of request traffic. Therefore, the users can scale up or down the table's output capacity without downtime or performance degradation. Further, the users can observe resource utilization and performance metrics by adopting the system. Also, the users can enjoy creating full backups of tables in a long-term relationship and archival for regulatory compliance needs because Dynamo DB offers the on-demand backup ability to users.

In this solution, a DynamoDB table is created to store the latest garbage bin’s fill level. Every hour, the garbage bins will send the fill level telemetry to the MQTT topics in the AWS IoT Core via bridging. The AWS IoT Rules Engine is configured such that a rule is triggered to write the received fill level telemetry from the corresponding MQTT topic to the DynamoDB table.

Next, the developers can use API to retrieve the bins that are maximum filled and alert the worker on duty in the web/mobile application. The level of debris in the bins is detected with the assistance of sensors and sensed data is delivered to the server. The ultrasonic sensor will detect the garbage levels and send a message to the truck's deliverer. As the truck driver must go to each bin every single day to check whether the bin is full or not. If the bin is not full at a certain level, it results in time consumption, waste of fuel and truck, as well as increased pollution due to smoke released from the truck. Nevertheless, while we are using API, when the bins are full or achieve more than 90% of the capacity, the devices will show the red warning in terms of text or sound that remind the person in charge to manage the garbage. At the same time, if the workers are not at the place, they can receive the notification or call related to this message so they can handle this issue on the spot. For instance, the API can access data and interpret the garbage level, thus notifying the nearest truck driver by updating the bin location. By doing this, it can forestall any absence of efficient waste management and cause severe environmental problems and cost issues.

Next, the developer can adopt data in API to measure bin size and raspberry pi to process the information. For example, once the garbage inside the bin crosses the specific threshold level, the bin and its location will illustrate in the web application by using google maps. It will calculate the fastest arriving truck and intimates the truck deliver in the location of the bin, then notifies the result in the web application. Each of the trucks can observe it easily and effortlessly. Besides, if the truck delivery is detected, the current location will be shown to make sure the garbage bins are empty before the dustbin overflow. It helps to reduce the environmental issues, save costs, and reduce unnecessary manpower.

Chapter 3: Comparison of Cloud Services

3.1 Amazon S3 vs Azure Blob storage

Amazon S3 is a simple storage service. AWS Simple Storage Solution is a cloud-based object storage service that provides industry-leading scalability, data availability, security, and performance. Additionally, a bucket is a storage container for Amazon S3 items. Each item is housed within a bucket. Objects are the most basic entities in Amazon S3 and are made up of object data and metadata. AWS S3 is a worldwide service for regions, although buckets are kept in a single area. The storage class determines the object's longevity, accessibility, and cost in an S3 bucket. Use lifecycle rules to specify actions users want Amazon S3 to do during the lifespan of an item, such as migrating it to a different storage class, archiving it, or removing it after a set amount of time. Bucket rules govern access to buckets from a central location.

Blob storage in the Azure account gives the data its namespace. Every item that users save in Azure Storage has a unique account name as part of its address. S3 is a bucket for Amazon, but Azure is Azure. Like a directory in a file system, Blob storage is a container that organizes a collection of blobs. An infinite number of containers can be added to a storage account, and each container can hold an unlimited number of blobs. The term "blob" refers to items that comprise things like photographs and multimedia files. Unstructured data is defined as data that does not adhere to any data model. Users may use Azure Blob Storage lifecycle management to transfer the data to the optimal access tier and expire data at the end of its lifespan using a rich, rule-based policy.

3.2 Amazon QuickSight vs Microsoft Power BI

Amazon QuickSight is a business intelligence (BI) solution offered by Amazon Web Services (AWS) that aids in the creation of data visualisations and interactive dashboards for in-depth research. This tool, which is powered by the SPICE super-fast parallel in-memory computation engine, can communicate with a variety of AWS sources, SQL databases, and even SaaS services such as Salesforce. The SPICE Engine does not require any prior configuration, making it incredibly simple to use. This also means that each user can work on the interactive analysis with multiple data sources in AWS at the same time. Overall, QuickSight provides a speedier environment as a result of this functionality (Amazon Web Services Inc, n.d.). QuickSight can integrate several data sources into AWS, such as Apache Spark, Amazon Redshift, Amazon Aurora, and so on. SQL servers, on the other hand, are also compatible. QuickSight even provides visualisation suggestions for the data. It has a function called 'AutoGraph,' which uses a collection of algorithms to learn data and recommend relevant graphs.

Microsoft's Power BI is a set of analytics tools created by the Microsoft. It has been one of the most used BI tools among analysts throughout time. When it comes to visualisation, Power BI is steadfast and offers a large range of customization choices. It can also import customisations from other business intelligence solutions. Power BI's data source integration and compatibility are outstanding *(Sukumar, n.d.)*. The user interface offers databases as a menu view, allowing users to quickly select their data source, connect to data, and begin analysing and reporting. The availability of APIs through which developers can integrate Power BI dashboards with other products is a standout feature. Power BI allows users to employ capabilities like Power Query, which make cleaning and previewing data for analysis easier.

Business intelligence is critical to improving an organization's day-to-day operations. By successfully utilising BI technologies, the obtained data can be transformed into useful insights (Sukumar, n.d.). Every firm eventually seeks for a solution that best fits the needs of all functional areas and devotes its resources to that solution. Despite the fact that Amazon QuickSight and Microsoft Power BI have similar goals, they differ significantly in terms of features and functionalities.

Features and Functionalities	Microsoft Power BI	AWS Quick Sight
Cloud capability	Connects to all leading cloud platforms	Connects to all leading cloud platforms
Implementation	Easy to Implement	Easy to Implement
Integration	Seamless integration with Microsoft Azure cloud platform services and also other sources.	Can connect to multiple data sources widely known Seamless integration with AWS cloud platform services
Self-Contained ETL (Extract, Transform, Load System)	No	No
Data Modelling	Supports simple data models	Supports basic data models using SQL
Data Blending	Supports data blending	Supports data blending
Security	Provides Row and User level security	Provides Row and User level security
Capability in Performing Advanced Analytics	Yes, Provides Analytics use cases using Azure ML web services	Yes, using ML insights supports web analytics. Amazon QuickSight provides major features ML-powered anomaly detection, ML-powered forecasting, and Auto narratives.
R & Python Integration	Extensive support to R and Python Integration	Supports integration. The use of AWS services can be an added advantage.
Data Visualization	Provides backend data manipulation features with access to Simple Visualization.	Provides Simpler Visualization
Embedded Analytics	Yes	Yes
Mobile App	Yes	Yes

Table 3.2.1: Comparison Table of Microsoft Power BI and AWS Quick Sight

3.3 Amazon DynamoDB vs Azure Cosmo DB

The advantage for the Amazon DynamoDB is performance and scalability. This advantage can give user the ability to auto-scale the usage by tracking usage of upper bounds. And it also can avoid the issue of performance while reducing cost and adjustment according to usage of amount data traffic. For the Azure Cosmo DB, it supports consistency levels, for instance, consistent prefix, session, and bounded-staleness that allows most flexibility and low cost-to-performance ratio.

Name	Amazon DynamoDB	Azure Cosmos DB
Description	Hosted, scalable database service by Amazon with the data stored in Amazons cloud	Globally distributed, horizontally scalable, multi-model database service
Primary database model	Document store Key-value store	Document store Graph DBMS Key-value store Wide column store
Secondary database models	None	Spatial DBMS
Cloud-based only	yes	yes
Typing	yes	yes
Secondary indexes	yes	yes
SQL	no	SQL-like query language
APIs and other access methods	RESTful HTTP API	DocumentDB API Graph API (Gremlin) MongoDB API RESTful HTTP API Table API

Table 3.3.1: compare Amazon DynamoDB and Azure Cosmos DB

3.4 AWS IoT Greengrass Core software vs Azure IoT Edge runtime

In this architecture, AWS IoT Greengrass Core software is installed in a device to provide the necessary supports to function as a core device. First, the software orchestrates the installation and execution of all the components deployed from the cloud service. The components are custom built to provide additional functionalities such as performing trash classification inference and ingesting fill level telemetry. Second, the Greengrass Core software monitors and reports the components' states to the cloud service. Third, the software enables local and cloud communication by allowing:

a) A client device to communicate with another client device.
b) A client device to communicate with the AWS IoT Core.
c) A client device to communicate with a core device's component.
d) A core device's component to communicate with another core device's component.
e) A core device's component to communicate with the AWS IoT Core.

Comparing with Azure, Azure IoT Edge runtime is a software that provides the exact functionalities as mentioned above. Despite the similarities, each software has few technical differences. Before discussing the differences, analogous technical terms that are used in both cloud service providers are shown below to avoid confusion.

AWS	**Azure**
AWS IoT Core	Azure IoT Hub
AWS IoT Greengrass	Azure IoT Edge
AWS IoT Greengrass Core software	Azure IoT Edge runtime
Greengrass Nucleus	IoT Edge agent & IoT Edge hub
Greengrass component	IoT Edge module
Client device	Leaf downstream device
Core device	IoT edge device

Based on the table above, it must be clarified that Greengrass nucleus offers the same functionalities as IoT Edge agent and IoT Edge hub combined. They are all compulsory component/module for their respective software. IoT Edge agent focuses on installing modules, executing modules, and monitoring modules' statuses. IoT Edge hub focuses on local and cloud communication (Microsoft, 2022a).

The first technical difference is that startup and failure recovery mechanism in the IoT Edge agent is less robust and efficient as compared to Greengrass nucleus. It is because Greengrass nucleus maintains the dependency graph between components while IoT Edge agent only maintains module's start up orders using arbitrary numbers (Microsoft, 2022b). For example, given that an inference component/module is dependent on a model component/module. For Azure, the inference module will not get notified when the model module suddenly shut down. As a result, the inference module might fail since the model module has not started. While for AWS, inference component will get notified and wait for the model component to restart when the model component suddenly shut down.

The second technical difference is that the local communication implementation of Greengrass nucleus is more mature than IoT Edge hub. It is because part of the IoT Edge hub is not production ready. Based on the table below, it is observed that the Azure IoT Edge MQTT broker is the only way to establish communication between client devices and modules. Unfortunately, as of March 2022, Azure IoT Edge MQTT broker is still in public review (Microsoft, 2022c). According to the terms of use, any services under public preview are excluded from the Service Level Agreement or any warranty. The services may be changed or even discontinued at any time without further notice (Microsoft, 2021). Thus, Azure IoT Edge runtime is not yet ready to be used for now since there is no alternative option to enable communication between client devices and modules.

Communication	Azure
Client device ⇔ Client device Client device ⇔ Module	• Use MQTT client libraries like Mosquitto to publish and subscribe messages to and from Azure IoT Edge MQTT broker. Can use user-defined topics.
Client device ⇔ Azure IoT Hub	• Use DeviceClient class from Azure IoT Device SDKs to send and receive messages to and from Azure IoT Hub. Cannot use user-defined topics. Must use IoT hub special topics instead.

Module ⇔ Module	• Use MQTT client libraries like Mosquitto to publish and subscribe messages to and from Azure IoT Edge MQTT broker. Can use user-defined topics. • Use ModuleClient class from Azure IoT Device SDKs to send and receive messages to and from another module. Cannot use user-defined topics. Must use IoT hub special topics instead.
Module ⇔ Azure IoT Hub Module ⇔ Azure IoT Hub	• Use ModuleClient class from Azure IoT Device SDKs to send and receive messages to and from Azure IoT Edge Hub and Azure IoT Hub. Cannot use user-defined topics. Must use IoT hub special topics instead.

Table 3.4.1: Implementation Details of Communication in Azure IoT Edge

Unlike AWS IoT Greengrass and Azure IoT Edge, other services like Google Cloud IoT Core, Cisco IoT Control Center, and IBM Watson IoT Platform are full managed services and do not provide development platform for managing and deploying custom components or modules from the cloud to the edge device. At the time being, AWS is the primary choice for application development in edge devices, followed by Azure. Azure IoT Edge will be equally as good as AWS IoT Greengrass if the Azure IoT Edge MQTT broker becomes generally available. Both of the AWS and Azure development platforms has healthy community support as can be seen at AWS official documentation and Azure marketplace, respectively.

References

Amazon Web Services, Inc., 2018. *AWS IoT Analytics Overview - Amazon Web Services*. [online] Amazon Web Services, Inc. Available at: <https://aws.amazon.com/iot-analytics/> [Accessed 13 April 2022].

Amazon Web Services, Inc., 2019. *Amazon QuickSight Announces General Availability of ML Insights*. [online] Amazon Web Services. Available at: <https://aws.amazon.com/blogs/big-data/amazon-quicksight-announces-general-availability-of-ml-insights/> [Accessed 13 April 2022].

Amazon Web Services, Inc., 2022. *Manage Data Streams on the AWS IoT Greengrass Core - AWS IoT Greengrass*. [online] docs.aws.amazon.com. Available at: <https://docs.aws.amazon.com/greengrass/v2/developerguide/manage-data-streams.html> [Accessed 26 March 2022].

AmazonWeb Services, Inc., n.d. *Amazon QuickSight*. [online] Amazon WebServices, Inc. Available at: <https://aws.amazon.com/quicksight/> [Accessed13 April 2022].

Amazon Web Services, Inc.(n.d.). *Compile and Deploy Models with Neo - Amazon SageMaker*. [online]docs.aws.amazon.com. Available at:<https://docs.aws.amazon.com/sagemaker/latest/dg/neo.html> [Accessed 13April 2022].

Barker, K., 2021. *AI Robotics Are the Future of centralized, data-driven Recycling Models*. [online] Recycling Product News. Available at: <https://www.recyclingproductnews.com/article/37211/artificially-intelligent-robotic-sorters-are-the-future-of-data-driven-recycling> [Accessed 13 April 2022].

BigRentz, Inc., 2021. *8 Innovative Smart Waste Management Technologies | BigRentz*. [online] https://www.bigrentz.com. Available at: <https://www.bigrentz.com/blog/smart-waste-management> [Accessed 13 April 2022].

Czermak, A., 2020. *Who Invented the Trash Can?* [online] PlasticMill. Available at: <https://www.plasticmill.com/blogs/plasticmill/who-invented-the-trash-can> [Accessed 13 April 2022].

Envac, 2021. *The Envac System - Envac*. [online] Envac. Available at: <https://envac.com.sg/how-it-works/the-envac-system/> [Accessed 13 April 2022].

Lin, L., 2020. *Digitizing China's Recycling efforts: Q&A with XHG Board Secretary Wang*

Yiru. [online] KrASIA. Available at: <https://kr-asia.com/digitizing-chinas-recycling-efforts-qa-with-xhg-board-secretary-wang-yiru> [Accessed 13 April 2022].

Microsoft, 2021. *Preview Terms of Use | Microsoft Azure*. [online] azure.microsoft.com. Available at: <https://azure.microsoft.com/en-us/support/legal/preview-supplemental-terms/> [Accessed 26 March 2022].

Microsoft, 2022a. *Learn How the Runtime Manages Devices - Azure IoT Edge*. [online] docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/azure/iot-edge/iot-edge-runtime> [Accessed 26 March 2022].

Microsoft, 2022b. *Properties of the Agent and Hub Module Twins - Azure IoT Edge*. [online] docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/azure/iot-edge/module-edgeagent-edgehub#edgeagent-desired-properties> [Accessed 26 March 2022].

Microsoft, 2022c. *Publish and Subscribe with Azure IoT Edge*. [online] docs.microsoft.com. Available at: <https://docs.microsoft.com/en-us/azure/iot-edge/how-to-publish-subscribe> [Accessed 26 March 2022].

Rivero, N., 2021. *City Garbage Collection Is Finally Getting the Disruption It Deserves*. [online] Quartz. Available at: <https://qz.com/1909600/the-future-of-garbage-collection-is-pneumatic-tubes/> [Accessed 13 April 2022].

Sohag, M.U. and Podder, A.K., 2020. Smart Garbage Management System for a Sustainable Urban life: an IoT Based Application. *Internet of Things*, 11, p.100255. https://doi.org/10.1016/j.iot.2020.100255.

SQLI Digital Experience, 2020. *New Technologies Help Raise Chinese citizens' Awareness of Waste Sorting | SQLI*. [online] www.sqli.com. Available at: <https://www.sqli.com/int-en/insights-news/blog/new-technologies-help-raise-chinese-citizens-awareness-waste-sorting> [Accessed 13 April 2022].

Sukumar, N., 2021. *Comparison of the Best Business Intelligence Platforms in 2021*. [online] Ideas2IT. Available at: <https://www.ideas2it.com/blogs/comparison-of-the-best-business-intelligence-bi-tools-of-2021> [Accessed 13 April 2022].