jamk.fi

Cloud Computing vs Edge Computing: Managing dataflow in intelligent robot systems

Corey Johnson

Bachelor's thesis
September 2020
Information and Communication Technology
Degree Program in Information Technology

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Description

Author(s) Johnson, Corey	Type of publication Bachelor's thesis	Date August 2020
		Language of publication: English
	Number of Pages 48	Permission for web publication: x

Title of publication
Cloud Computing vs Edge Computing: Managing dataflow in intelligent robot systems

Degree program
Information and Communications Technology

Supervisor(s)
Manninen, Pasi & Salmikangas, Esa

Assigned by
Solteq Oyj

Abstract

Solteq is a leading software solutions company based in Vantaa, Finland that specializes in many different sectors, including the retail sector. Solteqs latest project was to produce a robot that can go into retail stores to gather information in real-time about the product inventory, locations, and prices.

A comparative study between Cloud, Edge, and Fog computing platforms was conducted to help Solteq identify which computing platform is best suited in the development of the project to address bottlenecks and provide information as quickly as possible for their customers.

Through studying and understanding the robot systems, how they transfer data, and by testing different computing platforms on the robot or in an isolated environment which will simulate how the robot operates, the advantages and disadvantages of each computing platform were identified.

The results showed that it is in the best interest of Solteq to adopt and edge or fog computing strategy as the robot will work more effectively and more importantly, safely if the computation is done near the robot. Issues with latency, file transfer speed, and expensive computing plans offered by cloud services make using edge computing and fog computing a more viable choice, which may better satisfy customer need for up-to-date information on their inventories and price data.

Keywords/tags (subjects)
Cloud Computing, Edge Computing, Retail Robot, Solteq Oyj

Miscellaneous

Contents

Figures

Tables

Terminology

AWS – Amazon Web Services: an example of a cloud service

DPI – Dots Per Image: used to measure image quality. A high DPI is usually better.

EAN – International Article Number: used to identify goods in a computer system

FTE – Full-Time Employee

GB – Gigabyte, 1,000 Megabytes. A little over one hour of video fits in one GB

Gbps – Gigabyte per second: A high-speed rate of transfer

IT – Information Technology

MB – Megabyte: 1,000,000 bytes

Mbps – Megabyte per second, a good rate of transfer, or a good download speed measurement

OCR – Optical Character Recognition: A technology that can transform handwriting or photos to editable text on a computer

PIP – PIP Installs Packages: A package manager run on python

RAM – Random Access Memory: a form of computer memory that can be read and changed in any order, typically used to store working data and machine code

TB – Terabyte: 1,000 Gigabytes

1 Introduction

1.1 Thesis Background

Robotics is becoming more a part of our lives than ever before. In the beginning, robots were mainly used in repetitive manufacturing processes and any improvements took years or even decades to research and discover. The first industrial robot was put into use by General Motors in 1961, and it was mainly used for welding. Several years later, the first jointed robot was made by the same company. (Latxague 2013)

As the years went by, newer and newer technologies were invented, various sensors entered the robotic industry which allowed robots to “see”, “hear” and “touch”, thus robotics began slowly spreading into other areas in the industrial sector, replacing more repetitive tasks, and giving the factory workers more time and freedom to focus on more skilled labor. As these technologies progressed, Robotics soon began spreading into the consumer market on all levels, from entertainment and marketing to doing daily chores like vacuuming the floor and mowing the lawn. However, for many years, robots were only able to strictly follow given instructions. (Latxague 2013)

The invention of artificial intelligence opens a lot of possibilities in robotics advancement. With simple AI programs, robots can now have some ability to make their own decisions that might alter an otherwise repetitive task to either be more efficient, or avoid an obstacle, or complete calculations much faster than before, much like a human would in the real world. This new advancement secures the eventual spread of robotics into more challenging tasks such as keeping track of inventory, navigating, and even advancing medicine in ways such as being able to read x-rays and other medical-related images much better than an experienced doctor. For example, A doctor might be able to reference a few thousand images from memory to identify cancer growth, but an AI program with machine vision capabilities can reference millions of images and target cancer much earlier than a human can (Kann, Thompson, Thomas, Dicker, Aneja 2019).

Solteq is one of these companies that strives to lead the development of intelligent robotics systems in creating a robot that can be released into one of the most challenging and versatile environments that a robot has ever seen before: A retail store. In this environment, the robot must be able to navigate very dynamic layouts, identify and avoid unpredictable obstacles such as supply pallets and customers, make informed decisions about its surroundings, and report back to the retailer about what it 'sees' in near real-time. This creates a unique challenge in which the robot must be able to make millions of calculations very quickly while simultaneously collecting more data to process. This creates a very tight bottleneck in the processing of data as the robots need to be as efficient as possible to keep their clients on top of their market. This study aims to give an insight into this unique bottleneck and explore possible computing solutions to the issues caused by heavy data flow, from cloud computing to edge computing, and beyond. This study aims to provide a foundation for Solteq in deciding on which route it will take.

1.2 Company Profile

From the company website: "Solteq is a Nordic provider of IT services and software solutions specializing in the digitalization of business and industry-specific software. The key sectors in which the company has long-term experience include retail, industry, energy, and services. In addition to Finland, the company operates in Sweden, Norway, Denmark, Poland, and the UK. The company employs 600 IT professionals." (Solteq 2020)

Solteq operates primarily in Finland, with several offices in six cities around the country. Solteq has three offices in Denmark, two offices in Sweden, and an office in Norway, Poland, and a new office that just opened in London, England. (Solteq 2020)

In 2019, the Solteq group reported total revenue of 58,3 million euros, an operating profit of 5.7 million euros, and an adjusted operating profit of 3,5 million euros. Comparatively in 2018, the total revenue was 56,9 million euros, an increase of 1,4 million euros 2019, with an operating profit of 2,5 million euros (56% increase in 2019), and an adjusted operating profit of 2,3 million euros (34% increase in 2019). (Solteq 2020)

Solteq employs 600 professionals as of 31.12.2019, which is up from 567 from the end of 2018.

From their website, Solteq claims that they work in many different sectors in public and private business. Some of these sectors include Car Sales, Energy, Hotels and Restaurants, Wholesale, Manufacturing, Retail, and Public Sectors.

Solteq is very proud of many of their development projects in these sectors, and work with a lot of very well-known businesses. The projects Solteq is most well-known for are

- BI and analytics
- Customer Point of Sale Systems
- Enterprise Resource Planning systems
- eCommerce
- Online Marketing
- And Solteqs newest focus, and focus of this thesis: Robotics

1.3 A brief history of Solteq

Tampereen Tiedonhallinta (Tampere Data Management) was founded in 1982 as a company that specialized in Retail, Auto Retail, Manufacturing, Health, and Social data management. TH Tiedonhallinta was founded by Seppo Aalto, Chairman of the Board, and Ali U. Saadetdin, President and CEO until 2015. TH Tiedonhallinta Oyj was listed on the Helsinki Stock Exchange in August of 1999. On November 15th, 2000, TH Tiedonhallinta changed their name to Solteq Oyj. The current CEO as of 2017 is Olli Väätäinen, Kari Lehtosalo as CFO, Ilkka Brander as EVP of the Solteq Software team, and Juha Rokkanen as EVP of the Solteq Digital team. (Solteq 2020)

1.4 Research Questions

This research focused on a few key bottlenecks that are impacting the data flow in the robot systems. Thus, the questions that have come up, and will be the focus of this thesis are:

1. Where are the bottlenecks in the data flow from Robot to Client?

2. What are the challenges in sending large amounts of data wirelessly in real-time?
3. What are the pros and cons of each available computing platform?

1.5 Research methods

The methods which will bring about the best possible solution to the research questions lie within the observation of how the robot works and researching the solutions to the bottlenecks that come up during normal observation.

The most important clues towards the answer to the questions lie in a very thorough understanding of the robot systems:

- How each system works
- How the systems work together
- How the data gathered by the robot systems is processed internally
- How the data is sent elsewhere to be either processed further or published.

Next, a cost/benefit analysis for each computing system will be conducted, and if possible, the best system, in theory, will be tested on the robot to analyze how the robot performs using the computation services. These methods should be enough to come to a reasonable conclusion whether one computing platform is better than the other, if a mix of computational platforms should be used, or if there is a previously unknown or newly developing platform that Solteq may consider as a more attractive solution to their needs.

1.6 Targets

The target of this thesis is simple: the computing service that would theoretically perform most effectively under high demand will be considered or chosen as the computing platform used in the future of the Solteq Retail Robotics development project. This is nothing that the most effective computing platform is not necessarily the fastest, but that which produces the highest quality of data within the time constraints given by the customer, at the lowest cost possible. The robot will be expected to perform numerous calculations from each dataset it receives and output the result of these datasets continuously while more data is constantly flowing in and being queued up for processing. The computing platform which will be chosen will be the one that performs the best under high pressure, with minimal data loss, with a few program errors and other failures as possible. The criteria for choosing the computing platform in order of priority are as follows:

1. The speed at which the data is processed
2. Amount of data which can be handled at once
3. Quality of data processed
4. The visual output of the data after computation

There also may be a more mixed solution available, in which one computing platform is tasked to do take a certain load of the computing, while another is tasked to take care of the rest of the computational load. There is no real reason for the answer to be either black or white, in which the computational platform the robot uses can only be one or the other. This project is still in the prototype phase, which gives the development process the unique advantage to be very flexible in which directions it takes as it progresses.

2 Solteq Retail Robotics

2.1 Background

In 2017, Employees at Solteq were brainstorming some ideas for a new field of technology for Solteq to venture into. They discussed with their current customers and traveled the globe to various technology conferences in their search for solutions to their customer’s needs. As a result of the brainstorming session, the Solteq Incubator team was born. This team’s goal was to build and test different projects that may prove worthwhile for Solteq to invest in as a product. Amongst a few ideas, Solteqs retail customers noted that their retail store costs to check their inventory levels and keep the price tags up to date cost the retailers a lot of time, money, and valuable human resources. In some cases, they may need to hire one or two full-time employees just to keep track of all of this information. If Solteq could develop a robot that can do this repetitive work, then the employer could invest the employee's attention on other tasks that make them more valuable to the retailer. Thus, the Retail

Robot project idea was presented to the executive team, and the project idea was accepted. The Solteq incubator team was funded and the Retail Robot project was launched. (Solteq 2020)

The idea behind the Retail Robot was simple: to save a grocery retail company time and money it spends to check the shelves and products for emptiness and keep the price tags and EANS up to date. According to an advertisement found on YouTube, The Robot is advertised to be able to make a map of the store and pinpoint the products in their exact location in a 3D environment. The robot is also able to take street view photos of its surroundings and stitch them into a 360-degree panorama that is uploaded to the internet, where a retailer, executive, or vendor can check up on the layout of necessary displays without having to physically visit the store or send someone to do it for them, saving countless hours in traveling or asking someone at the location to verify the layout through photos or video. The robot can scan the shelves, and collect price tag information, and tell the retailer if the shelves are empty, give live feedback on the status of an EAN, and other product information on the EAN. The robot is advertised to be fully integrated into the POS and ERP systems in the store which allows the robot to fetch the latest information instantly from the store's supply chain and back-end systems. The Retail Robot is advertised to potentially save a retailer millions of dollars per year in reduced labor costs, and improved sales due to the improved availability of the store's products. From the website, the robot is advertised to forecast demand more accurately, and this can provide many benefits for a retailer including:

- Sales growth
- Less waste
- More efficient use of working capital
- More efficient purchasing and supply chain management
- Better customer experience as a bonus

The use of robotics in a retail chain is advertised to save a retailer money in hiring staff, offers real-time space management and shelf-space planning, allows precise identification of product locations in a 3D map, checks real-time price information, allows monitoring store displays remotely, and can guide online-store collections.

The number of full-time employees needed to check the shelf availability of products and inventory management can reduce by as much as one FTE per mid-sized store or more depending on the store size. (Solteq 2020)

2.2 Physical Characteristics

The Retail Robot as advertised, is approximately 2,5 meters high and has a sleek modern design. It is white by default but can be ordered in different colors to suit a customer’s environment. It is wider and heavier at the base, and the tower is thinner and lighter. This gives the robot stability as it moves and navigates through the tricky store environment while being able to view the highest shelves without tipping over. The robot features several types of cameras capable of doing the different tasks at hand, such as taking photos of the products, and very bright LED lights spanning the length of the tower. The robot can detect and differentiate between obstacles and customers, and it can navigate the store using a LIDAR attached to the front of the base. The robot has two long brass plates on the backside of the base, which is used to charge the robot. The robot backs into a charging base and the copper plates connect with the charging mechanism. The emergency stop button and power buttons are also located here. In summary, the design of the robot is physically very simple (see Figure 1).

Figure 1. Solteqs Retail Robot prototype

2.3 Retail robotics elsewhere

The idea of having a smart retail robot in a working environment is not an entirely new concept. There are many approaches that robots are taking in the retail world, such as handling checkout procedures and customer service. (Yakowicz 2016; Fukushima 2016) The use of robots in the inventory management sector of retail work is a relatively new concept, with the first robots hitting the stores as recently as 2018. Bosa Nova Robotics and Simbe Robotics for example have had their inventory robots in stores since early 2018 and many other companies quickly followed suit worldwide. (Smith 2018)

These companies have many common goals: to help their customers keep their shelves stocked, to reduce losses in missed sales and missing products, and free up the time used by workers in checking shelves. Some robots can identify the products on the shelves and determine if something is in the wrong place, others can take stitched photos so their customers can see the store without having to physically visit them. Solteq is the first known company to offer the ability to create a 3D model of the store shelves and place the products in the model in real-time.

2.4 Criticism against Retail Robotics

While Retail Robots are new and are advertised to help retailers 'save millions', Robotics has not been an entirely welcome sight to many. According to one article from Wbur, many workers are not seeing these robots as opportunities, but more like replacements. Robots in retail are not all that they're cracked up to be, noting that the newest AI robots are being left unattended, they're finicky and slow, and are constantly requiring human assistance. In some cases, the robots are not able to navigate isles as advertised and will shut down because the robot cannot take a corner. While the robots are working out their kinks, there has been a notable improvement in their performance and workers are starting to worry if these machines will eventually take over their jobs altogether. (Bauman, Chakrabarti, Schroeder 2019)

While many places of work claim that their robots will not replace their workers, it has been shown time and time again throughout modern history that robots have replaced as many as 3.3 workers per robot in the US (Dizikes 2020), and that as many as 25% of the entire labor force may be replaced as early as 2030 (Carson 2019). Robots are slowly taking over the most seemingly unlikely of jobs such as retail floor work and courier services and it is only expected to grow from there.

3 Challenges

3.1 Inventory Management Issues

In the past, a lack of up-to-date data in retail stores has caused losses of billions, losses which can cause even the largest retail chains to close their doors. In 2015, Target's supply chain suffered chaos when their Barbie toy cars piled high in their distribution centers. The mistake was caused by a mismatching EAN in Target's computer system, resulting in empty shelves, frustrated customers, a loss of $2 billion, and Target's eventual exit from Canada (Sengupta 2018). Similar losses affected Walmart when misplaced goods, overstock situations, and lack of immediate replacement cost the brand $3 billion.

These kinds of losses are the prime examples that motivate retailers to turn more to robotics, with 57% of retailers planning on making some sort of investment in AI by 2021. The implementation of robotics promises increased sales through product availability, better able to make accurate predictions, and adapt more quickly to customer needs.

Finnish grocery store chains have been making investments in AI as early as 2018, noting that people are spending too much time performing manual labor. AI is a more sensible solution lately as robotics can make more logical decisions and lower mistakes made by workers. (Kupsa 2018)

3.2 Bottlenecks

As the retail robot nears the end of the prototype phase and enters into the customer environment, the fleet of robots must be able to handle the data it collects as effectively and as efficiently as possible. The amount of data that the robots will collect will increase as they become more efficient, and the number of robots that will enter full service will also increase. This means that there could be several bottlenecks that will hinder the flow of data from the robots to the customer. According to PC Builds, A Computer Bottleneck occurs when there is a restrictive element holding back the performance that could otherwise be achieved. In simpler terms, it is anything that slows down the flow of data and keeps the computer running at maximum performance. (Bruno 2019) In other words, the computer system can only perform as fast as its slowest component. In the Retail Robot, it is believed that the main bottlenecks lie in the ability for the robot to perform all functions necessary in converting the photos that the robot collects into usable data that the customer can use, and in the bandwidth speeds in which the robot sends the data to the cloud service. In a retail robot, the data flow would be fairly straightforward:

1. The Robot moves along an aisle and takes photos of the shelves (see Figure 2).
2. The photo processing program and OCR checks the photos for any readable data, such as barcodes, EANS, and other characters.
3. A trained algorithm checks for shelf emptiness.
4. The shelf emptiness value and the price tag information are paired together
5. The exact location of the price tag is calculated and recorded in a 3D map.
6. This data is compiled together and sent to a computing service for storage and gathering.
7. The computing service then checks the accuracy of the price tag information with the store's systems.
8. The results are sent to the customer, and they receive information on any price tags that are inaccurate or outdated, any shelves that are missing products, the exact location of the product in a 3D map, and a series of panoramic photos of each shelf.

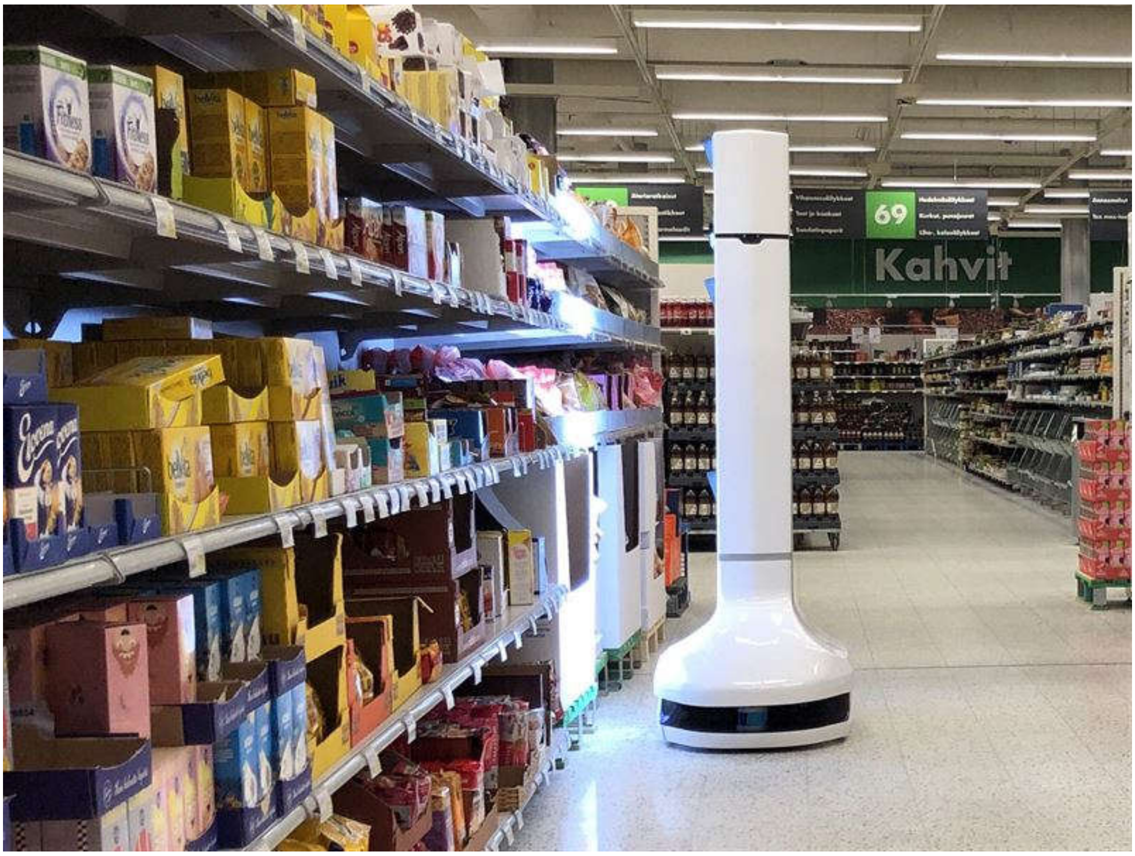

Figure 2. Retail Robot in action (Solteq Group 2020)

The robot can take as many as 5000 photos of a shelf in a single run, and each photo must be checked for the data mentioned above and sent to the customer in as little time as possible. The robot must then repeat this process for every shelf in the store, so the robot must be able to handle the data it collects as effectively as possible and return it to the customer promptly. Thus, this thesis aimed to determine the best course of action to take in managing this data flow as quickly as possible. Should the robot process the information onboard or close by before sending it to the cloud service and customer, or should it send the data to a cloud service and have it processed there before being sent to the customer?

4 Cloud Computing vs Edge Computing

4.1 Scenario

The Retail Robots task, as stated earlier, is to navigate the isles of their customer's retail stores and gather data from the images that the robot captures of the shelves and its surroundings. The robot may gather thousands of images from its cameras from even a single run, and these images must be of high enough quality that the data from the shelves is legible by the cameras and sensor. This means that the size of each photo can be large, and thousands of these photos mean that the amount of data that should be processed and sent are even in the terabyte range. In a fictional setting, a retail robot takes 40 Mb images and gathers 50,000 images, totaling 2 Tb of high-quality photographs. Each photograph must be analyzed, and the data from each photo must be extracted before it can be sent to the customer for review. In this scenario there are a few choices: In which the robot can process the data via a cloud computing platform, where the raw data is sent first to the cloud to be processed and then sent to the customer, or an edge computing platform, in which the data is processed on board before being sent to the cloud service for further processing to be sent to the customer.

4.2 Cloud Computing

According to the Microsoft Azure Cloud service, a cloud computer is, in simple terms, a platform in which resources are available for delivery on demand, especially data storage and computing power (see Figure 3). (Microsoft 2020) Cloud computers offer a host of other services such as servers, databases, networking, software, analytics, and intelligence that can be delivered over the internet. These are used to help a business lower their operating costs, and run a company's infrastructure more efficiently while allowing them to scale the services as they change. A cloud computing service is attractive in many cases.

1. Cloud computing services cut out the hardware and eliminates the need to cut out on-site datacenters, which in turn cuts the costs of electricity use for

power and cooling, and cuts the need for IT experts for managing the infrastructure, which results in a much lower cost for the software company.

2. Cloud computing services can be set up and provisioned in a few clicks and can be serviced and scaled on-demand, which take the pressure off of businesses in planning for their capacity.
3. IT professionals no longer need to take part in IT management chores, such as hardware setup, patching, maintenance, updating, and upgrading their software.
4. Cloud services are usually more reliable than in-house solutions as businesses can back up their data and recover from disasters more easily and cheaper since the data can be mirrored across multiple sites instead of in-house.
5. Cloud services are usually very secure, saving a business from investing in a private data security business

Many companies offer cloud computing services. According to Guru99, some of the companies that offer these services are

- Amazon Web Services, or AWS
- Kamatera Performace Cloud
- Cloudways
- DigitalOcean
- RackSpace
- MassiveGrid
- AlibabaCloud
- LiquidWeb
- Microsoft Azure
- Google Cloud Platform

Each having its niches and specialties which allow them to stand out from the others, such as free SSL certificates, unlimited website hosting, 24/7 support, Free trials, and more. (Guru99 2020)

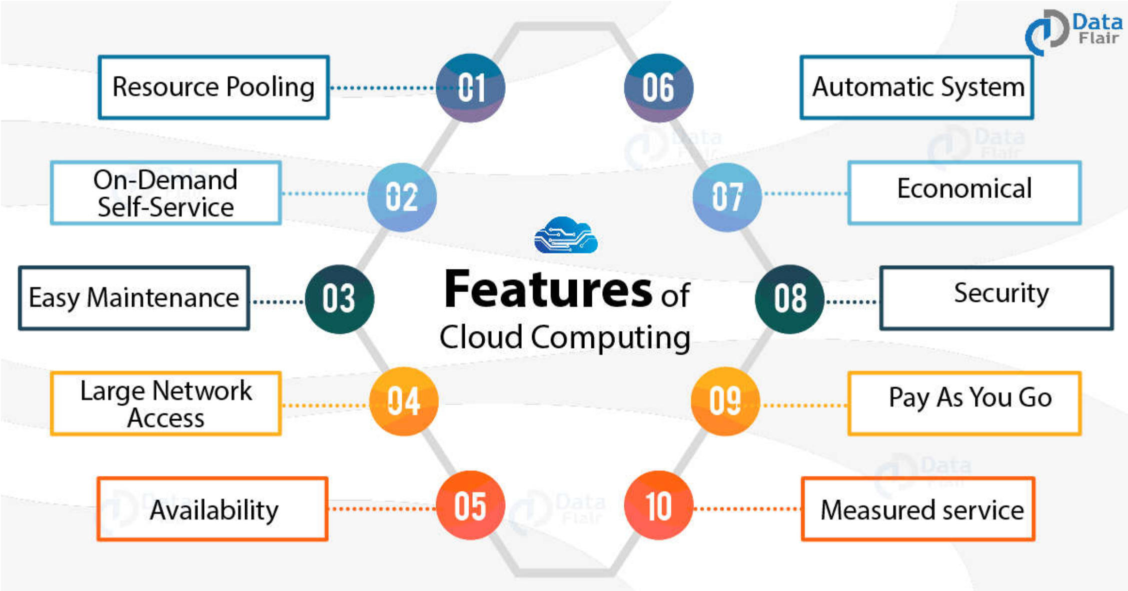

Figure 3. Cloud Computing Diagram

The major cons of cloud computing are that its use requires a consistently high-speed connection between the cloud service and the customer. Also, the use of cloud computing services is becoming more popular, which may slow down the cloud services' ability to perform as required. While security for most cloud services is a high priority, it may not be enough for extremely sensitive data which is safest solely in the hands of the owner and disconnected from the internet.

4.3 Edge Computing

According to Gartner Solutions and NetworkWorld, edge computing is, in simple terms, any information processing that happens “close to the edge” – where things and people produce or consume that information.

“Edge Computing brings computation and data storage closer to the devices where it's being gathered, rather than relying on a central location that can be thousands of miles away. This is done so that data, especially real-time data, does not suffer latency issues that can affect an application's performance. Also, companies can save money by having the processing done locally, reducing the amount of data that needs to be processed in a centralized or cloud-based location. (see Figure 4)

Edge computing was developed due to the exponential growth of IoT devices, which connect to the internet for either receiving information from the cloud or delivering data back to the cloud. And many IoT devices generate enormous amounts of data during their operations." (Shaw 2020)

A few reasons edge computing is used is to save on bandwidth costs and decrease latency for critical processes. Sending tremendous amounts of computing being processed and sent to a cloud service is expensive. Instead, all of the heavy processing can be done locally, and the cloud service can be reserved for lighter processes. This saves the business time and money compared to relying solely on cloud service technology to conduct all of the processing. Another major advantage of edge computing is that this gives the company more power to prioritize which data gets handled first, whereas cloud services may only process the data in the order it comes in.

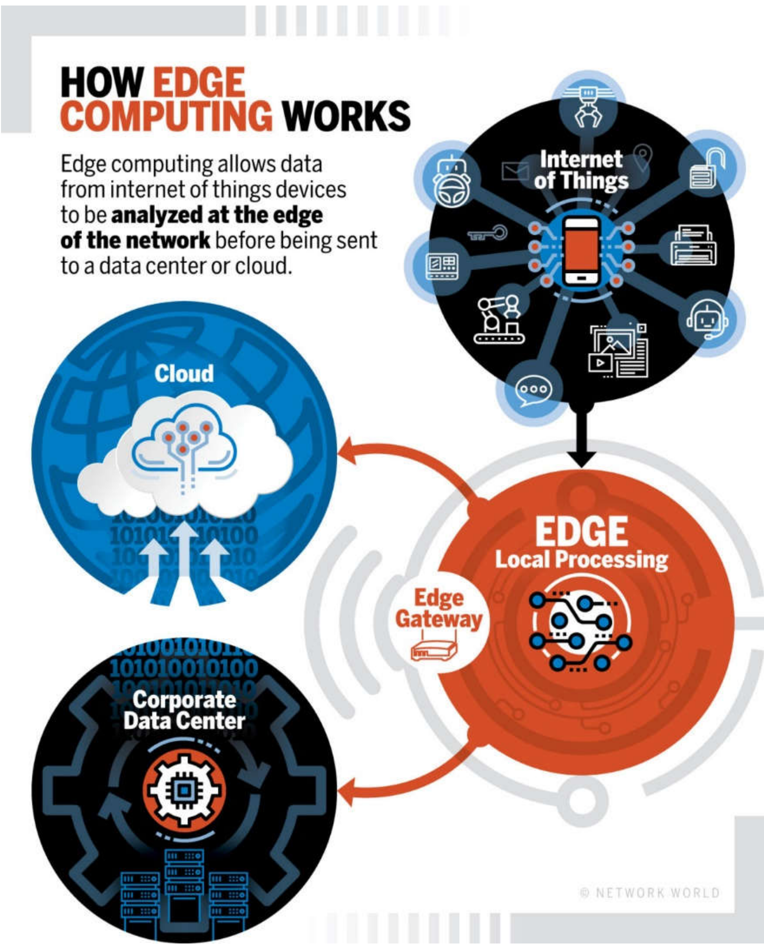

Figure 4. How Edge Computing Works (NetworkWorld 2019)

One study from IEEE Communications Magazine suggests that Edge computing devices can have up to 50% lower response time compared to core (cloud) computing systems, along with up to 40% lower delay and 30% throughput time respectively (see Figure 5). “Compared to the traditional core-based cloud computing infrastructure, mobile edge computing leads to lower delays and response times, and high throughput for many applications because most of the devices used in mobile edge computing need various services at the edge of the network.” (Kumar, Seadally, Rodrigues 2016)

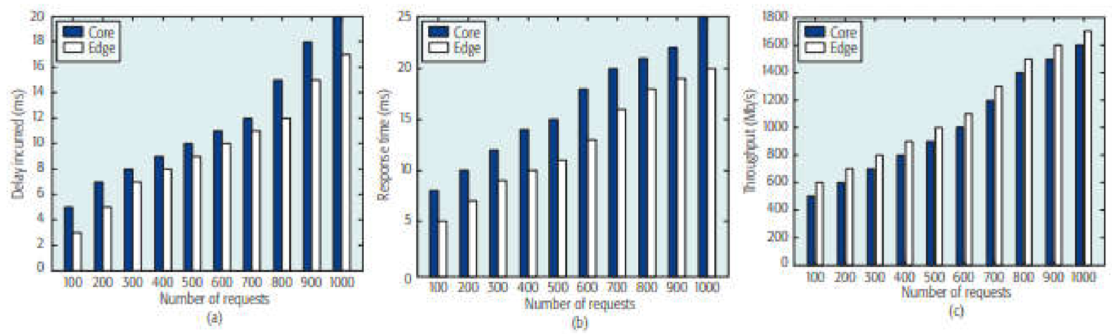

Figure 5. Delay, Response, and Throughput comparisons between core and edge computing devices (Kumar, Seadally, Rodrigues 2016)

This suggests that if Solteq were to use more edge devices in their robotics network, they may experience a much shorter latency period compared to using core or cloud computing services with their robot fleet.

According to w.media, The major cons of edge computing are that companies must invest in their ability to process, secure, and store the data. This is very expensive for many companies to do and should be considered if:

- The amount of data that requires processing would be more expensive to send to the cloud service to process than to process locally
- The data needs to be processed and returned with little latency.

Other cons include the lack of redundancy, which in turn causes issues with longer downtimes if there is data loss, corruption, or hardware failures. (w.media 2018)

4.4 Fog Computing

Fog computing is a relatively new platform compared to cloud and edge computing, in which computing happens over a distributed network and is collected in a place where it can be accessed without having to send it to a cloud service. The term fog computing was used first in 2011 (Bonomi 2011).

According to Tim Keary from ITPRC, a fog computing system enables an organization to be able to collect and access data from a central point without having to send it to a cloud service (see Figure 6).

> *Devices from controllers, switches, routers, and video cameras can act as fog nodes. These fog nodes can then be deployed in target areas such as your office floor or within a vehicle. When an IoT device generates data this can then be analyzed via one of these nodes without having to be sent back to the cloud. The main difference between cloud computing and fog computing is that the former provides centralized access to resources whereas the latter provides decentralized local access. (Keary 2018)*

Fog devices are best suitable for companies that need to be able to react to data in less than a second, for example in automobiles. Fog computing allows you to generate real-time data and insights at the edge of the network without sending information back to the center. (Keary 2018)

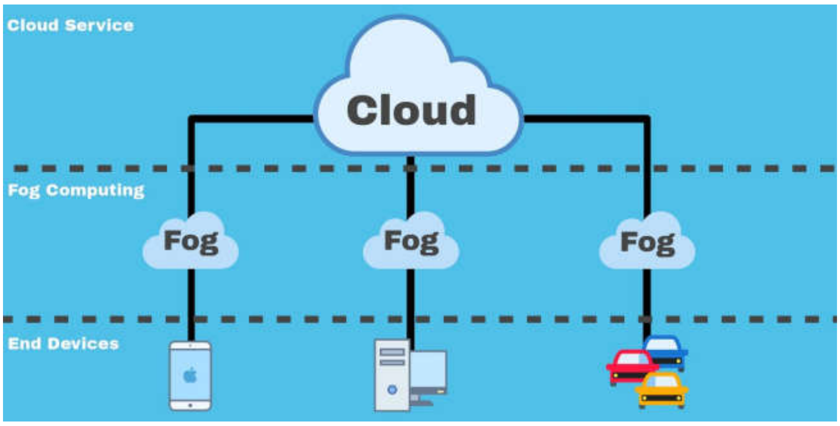

Figure 6. Fog computing diagram (Keary 2018(

In short, the main purpose of fog computing is to reduce the amount of data sent to a cloud service for processing as much as possible, while keeping the data as close to the end-user as possible.

4.5 Bottleneck analysis

Using the scenario laid out earlier, each cloud computing platform will be considered for the same set of fictional data. In the first scenario, the robot will send the data to a cloud service. According to a file transfer calculator provided by expedient, even with the fastest unhindered advertised internet speeds available in Finland today, at 600Mbps, it could take a robot network over 2 hours to transfer 2 TB of photo data to a cloud service to be processed. It would be impossible for Solteq to consider a pure cloud computing solution for a single retail robot, let alone an entire fleet of them. Thus, the bottleneck that would take place in a cloud computing strategy would be clearly in the bandwidth speeds available to transfer the data to the cloud service.

In an edge computing model, robots in a local area would send the photos to a local server to process the images. Afterward, the data extracted from the images would be sent to the cloud service to be further processed if needed. This could lower the processing time to potentially hours instead of days. If a robot could send 4 images per second to an edge computer, the computing platform would be able to process 1000 images in just over 4 minutes. If the robot takes 50,000 images throughout several runs of the shelves in a grocery store, and a local edge device processes the data between each run, the processing would still take over 2 hours. Even though, the time saved compared to a pure cloud computing strategy is significant. Since the size of the extracted data is negligible compared to the size of a photo, the network would be able to handle a fleet of retail robots sending data to the cloud service at once. Here, the bottlenecks would lie in the robot’s ability to process the photos and extract data from them instead of a network's ability to send it.

However, this may cause data integrity issues if any systems become corrupted during the processing phase. A lack of redundancy and risk of breakdown would delay the transfer of data from the robot to the customer severely, plus storing large amounts of data onboard may hinder the robot’s function if there is no way for the data to be stored externally. In turn, a mixture of cloud computing/edge computing services are considered to take advantage of the strengths of both services.

In a fog computing strategy, Each robot is responsible for its processing, and the processed data is sent to a local service which then a customer can access without having to log into the cloud service to see updates as close to real-time as possible. This eliminates the need for a central computer to handle the photo processing, but instead, each robot must have enough computing power onboard to process the photos that they take. This may make each robot more expensive but it may save money overall from having to invest in local processing power. This also eliminates the need for the cloud service for most needs, but the cloud service can still be considered if there is data that needs to be accessed globally.

5 Testing theories

5.1 Set up

To determine which method would be best for the Retail Robot project, a series of small simple tests can be set up and carried out to help estimate how data handling might function in the real world. The most important qualities a successful retail robot prototype needs are simple. Firstly, a high computation rate which can process images quickly is most important. Next, a high rate of data transfer needs to be possible for data to flow freely from the robots to destinations. Finally, low latency is necessary to ensure that data is delivered as quickly as possible from source to destination. Using some of the numbers and figures stated earlier in chapter 4.1, we can run a set of simulations to test which computing platform may return results faster.

The first test that will be run is a simple latency test, which tests the latency between A computer and cloud services, and also with other local devices. This should simulate the cloud computing and edge computing network. The test should also simulate sending large numbers of requests that the robots will be making to their respective hosts. The goal was to see how each computing platform can handle the request and what happens to the latency during each phase.

The next test would be a file transfer test. The robot may need to be able to transfer large file sets to a cloud service or edge device, and the test will determine how much time it is expected to take for a large file to be transferred to either a cloud service or a local storage service.

The final test will be a data extraction test, to determine how quickly available OCR software can extract data from a set of photos both on a cloud service and locally. This will help determine if it is best for image processing to happen locally or in a cloud service.

5.2 Latency Testing

Latency is the time it takes for a request to be sent from the source to a destination. This is calculated in milliseconds, the best target latency is as close to 0 as possible (see Figure 7). (DNSStuff 2019) With a latency of close to 0, the possibility to receive a fast response to a calculation is possible.

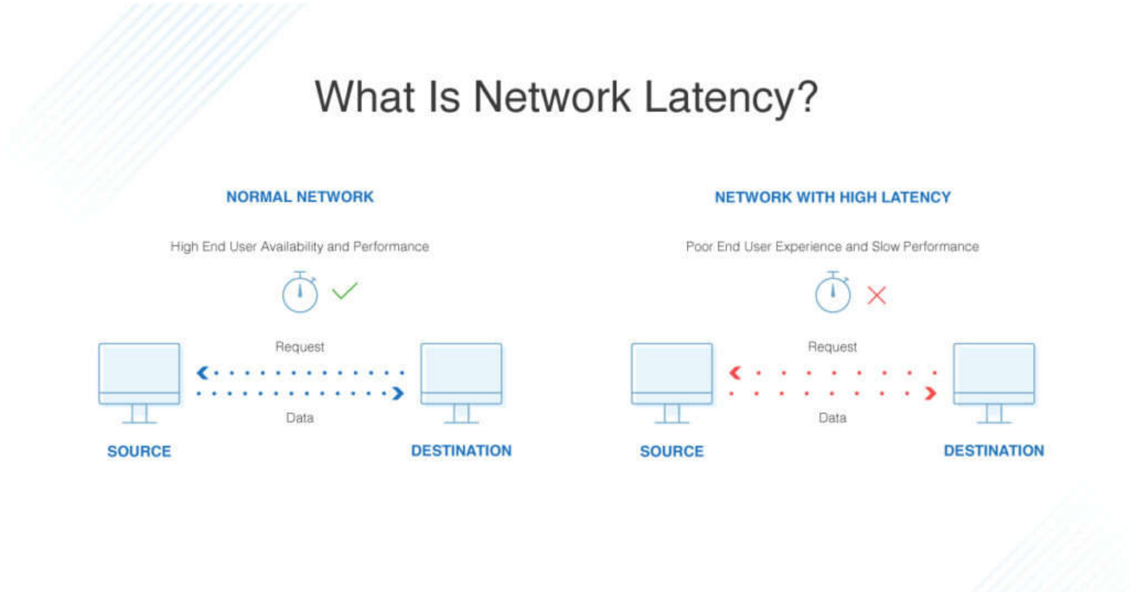

Figure 7. What is Latency? (DNSStuff 2019)

Latency can be determined by sending a tracert command to a set address. Some cloud services have latency testing available online. AWS, Google, and Azure have a latency program available that anyone can access by visiting their respective pages online. First, A cloud service latency test was requested from AWS, Azure, and Google to measure the range of latency these services have. The tests took between 10 minutes each, and each test sent several queries to each several different locations worldwide. The results are summarized in Table 1. The latency tests were provided by cloudpingtest.com (see Appendix 1).

Table 1. cloudpingtest showing cloud services latency range

Cloud service	Number of queries	Range of Latency
AWS	78	69-2371 ms
Google Cloud	87	38-1882 ms
Azure	110	71-306ms

The results of the cloud service latency tests show that servers closer to the source tend to have lower latency (Europe region). While servers further away have higher latencies (Asia) (see Figure 8).

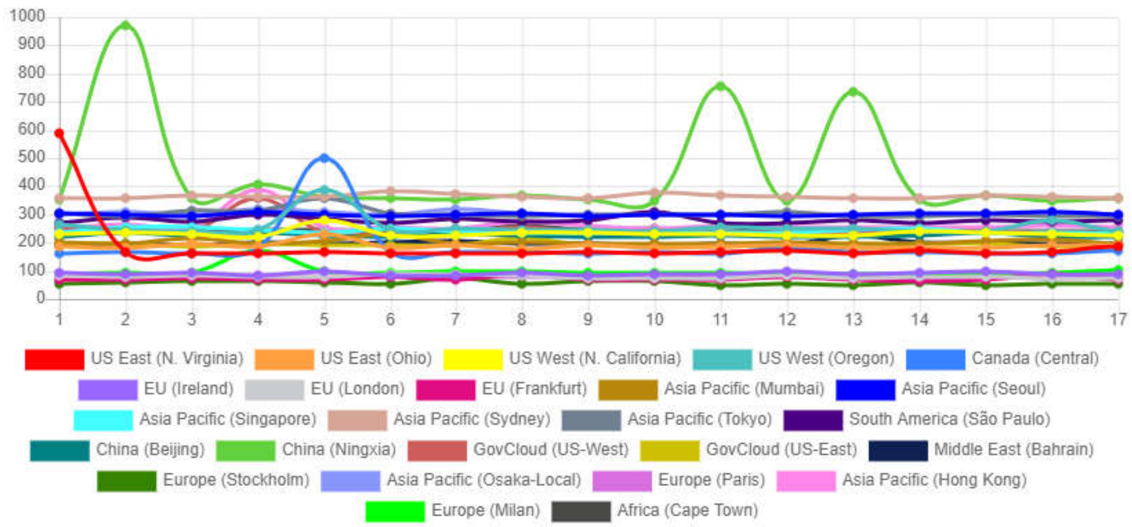

Figure 8. AWS Latency test (ping.psa.fun)

Next, the same latency tests were performed again and the areas with the best mean latency are selected. It is shown in table 2 that the areas with the best latency are usually within the European continent or Schengen territory.

Table 2. cloudpingtest results for mean latency

Cloud service	Area	Mean Latency
AWS	Europe (Stockholm)	219ms
Google Cloud	Finland (Hamina)	160ms
Azure	Germany/West Europe	74ms

In the next phase of testing, local latency was checked using a tracert command to a nearby location. A local area latency test can simulate an edge environment. To simulate a local area edge device, a tracert command was sent to jamk.fi from a nearby location. (see Figure 9).

```
$ tracert jamk.fi

Tracing route to jamk.fi [195.148.129.49]
over a maximum of 30 hops:

  1    17 ms    11 ms    31 ms  192.168.43.165
  2     *        *        *     Request timed out.
  3    61 ms    43 ms    37 ms  10.64.205.77
  4    52 ms    28 ms    59 ms  elisa-as719.ficix1.ficix.fi [193.110.226.16]
  5    41 ms    29 ms    30 ms  csc.ficix1.ficix.fi [193.110.226.14]
  6    55 ms    34 ms    41 ms  espoo2-et-0-1-7-1.ip.funet.fi [86.50.255.233]
  7    53 ms    34 ms    48 ms  hameenlinna1-et-0-0-0-1.ip.funet.fi [86.50.255.222]
  8    35 ms    47 ms    41 ms  hameenlinna2-et-0-0-1-1.ip.funet.fi [86.50.255.225]
  9    31 ms    47 ms    29 ms  tampere1-et-0-1-0-1.ip.funet.fi [86.50.255.221]
 10    34 ms    32 ms    37 ms  jyvaskyla1-et-0-1-0-1.ip.funet.fi [86.50.255.161]
 11    99 ms    72 ms   130 ms  193.166.5.142
 12    33 ms    29 ms    30 ms  host49.guest.jamk.fi [195.148.129.49]

Trace complete.
```

Figure 9. Tracert command to Labranet from a local location

Next, a ping command was sent to jamk.fi, and it shows similar results of latency being close to 30ms. A ping command simply calls the target and waits for a reply (see Figure 10). The reply shows the number of bytes received, the time in milliseconds for the target to receive the call, and the TTL (Time to Live) in which a packet is to be dropped if the latency exceeds that time.

```
$ ping jamk.fi

Pinging jamk.fi [195.148.129.49] with 32 bytes of data:
Reply from 195.148.129.49: bytes=32 time=31ms TTL=247
Reply from 195.148.129.49: bytes=32 time=28ms TTL=247
Reply from 195.148.129.49: bytes=32 time=44ms TTL=247
Reply from 195.148.129.49: bytes=32 time=31ms TTL=247

Ping statistics for 195.148.129.49:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 28ms, Maximum = 44ms, Average = 33ms
```

Figure 10. ping to jamk.fi

Another latency test from within the JAMK Labranet network can simulate if the robot does not need to send data to an external source (see Figure 11).

```
$ tracert jamk.fi

Tracing route to jamk.fi [195.148.129.49]
over a maximum of 30 hops:

  1    <1 ms    <1 ms    <1 ms  192.168.50.1
  2    <1 ms    <1 ms    <1 ms  192.168.100.2
  3     1 ms     1 ms     1 ms  195.148.39.49
  4     1 ms    <1 ms    <1 ms  host49.guest.jamk.fi [195.148.129.49]

Trace complete.
```

Figure 11. tracert from within JAMK network

After several tests, the latency rarely exceeded 1ms. It can be concluded that using the local network is the best possible solution in regards to latency. Table 3 shows the local area latency result.

Table 3. Local network latency tests

Tracert to	Latency
jamk.fi	29ms-33ms
jamk.fi (from inside the network)	<1ms

The result of the latency tests is heavily in favor of using local or edge networks in a computing platform.

On the retail robot, some critical systems require a near-zero latency to work successfully. One of these systems for example would be obstacle avoidance. The robot needs to be able to know how to react to possible obstacles in its path, for example, a human. A human may suddenly step in front of the robot to retrieve an item, and the robot should not have to send the data to a remote data center and have to wait even a few seconds to know how to react, instead an immediate reaction is necessary, and the robot should know to stop when there is a person in its path. On the other hand, a static obstacle, such as a box, is not moving. The robot should be able to know the difference between a human and an object and be able to change its course if the object is not moving from its path.

5.3 Data Transfer Tests

A data throughput test was conducted to cloud services and local services to determine how fast it would take to upload data to a specific destination. If a retail robot has to send even several gigabytes of data somewhere, the transfer must be completed as fast as possible. In this case, we can use some speed test tools available online such as cloudharmony to test how cloud services handle data transfer. Local data transfers to edge devices can be simulated through a file transfer done via the command line. In the case that the robot processes its data, we can assume that the time for file transfer is 0 since the data will stay at the place it is stored originally.

Cloudharmony network tests performed 6 tests from 256KB to 10Mb file sizes on each cloud service, testing the throughput speed on each service (see Figure 12). To keep data concise, one region from each service was chosen.

Figure 12. Example result from cloudharmony test on Azure

Table 4 shows the throughput speed for a specific region in the EU area. This may be inaccurate compared to real-world figures available to companies, however an accurate result when using a personal device.

Table 4. Throughputs in the EU region

Service	Region	Uplink Mb/s	Downlink Mb/s
Amazon EC2	Eu-west-1	10.79	34.03
Amazon Lightsail	Eu-west-1	8.66	16.7
Amazon S3	EU-west-3	-	39.36
Google Compute Engine	Europe-west3-a	8.83	44.37
Google Cloud Storage	EU	-	48.32
Microsoft Azure Virtual Machines	EU-west	7.55	52.08
Microsoft Azure Cloud Storage	EU-north	-	30.13

While Computing services showed both downlink and uplink speeds, cloud storage services provided only downlink speeds to be shown in these tests. (see Figure 13)

Figure 13. Amazon S3 Downlink speeds

In a local network, writing and reading files can happen much faster. In a local file transfer test conducted through a program available online, the 'LAN speed test', a file of 1 Gb size was created and transferred to a folder on the local device.

The results show that transferring files to a local location is much faster compared to transferring over the internet to a remote location. (see Figure 14)

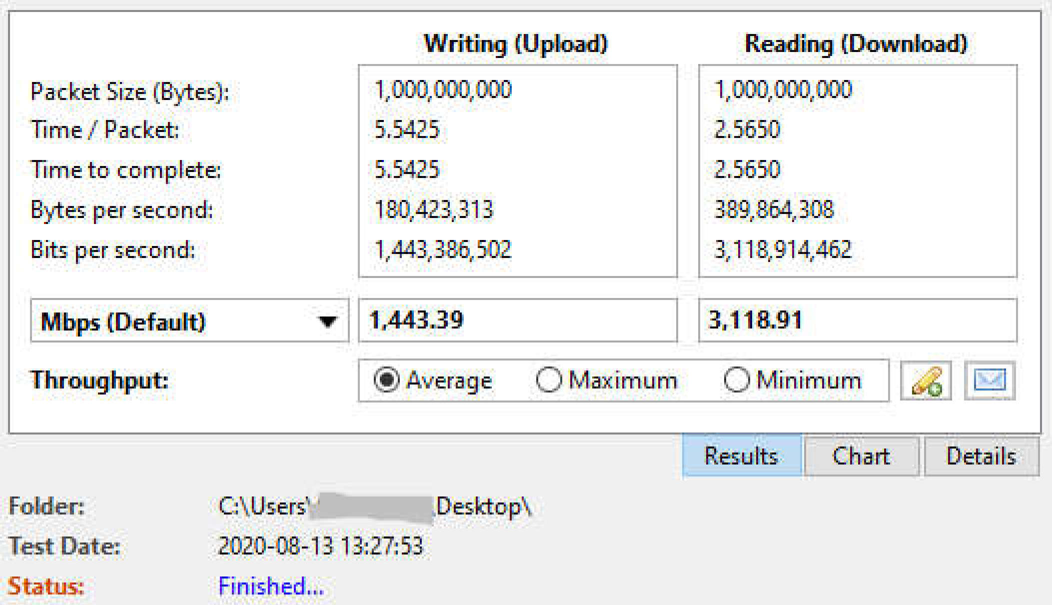

Figure 14. Local area network file transfer

Similarly, when the same file was transferred from within the JAMK network to a shared drive online simulating an edge device, the file transfer speed does drop somewhat but not significantly. (see Figure 15) This best simulates the retail robot environment, as all systems are contained since the robot is mobile, all non-local file transfers must be made wirelessly.

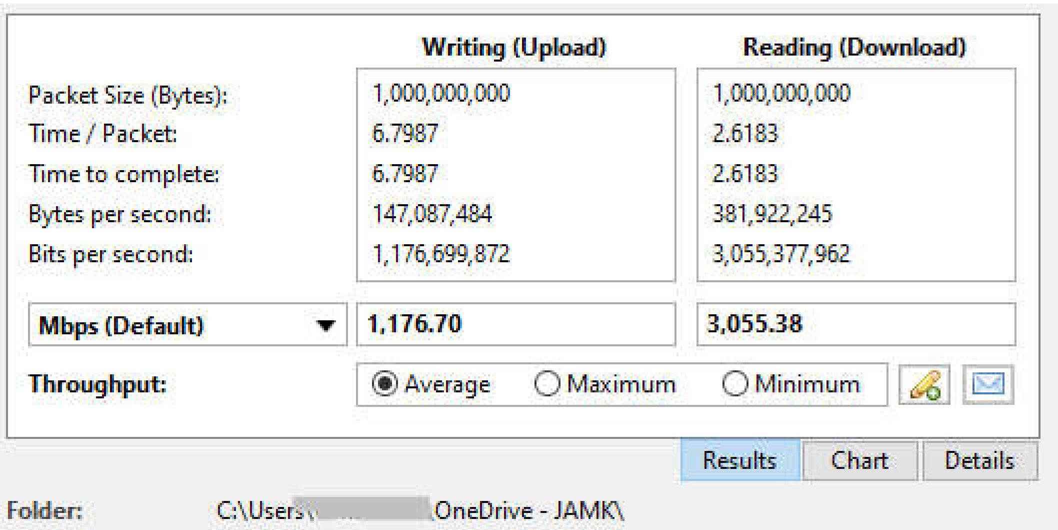

Figure 15. File transfer to JAMK shared drive

While the results of a transfer to an online drive in the same network are slightly slower compared to a file transfer to a different place on the same computer, it is still clearly much faster compared to transferring files over the open internet to a cloud service.

Using the figures from the scenario earlier in chapter 4.1, it could take a robot network perhaps several days to transfer 2 TB of photos to a cloud service even at the fastest available speeds, whereas a local connection can potentially transfer several gigabytes of photos every second, cutting the transfer time of large files to local and edge devices to less than an hour. Table 5 shows the results of the data transfer tests.

Table 5. Estimated time to transfer 2 TB of data over a network

2 TB Transfer to	Writing speed Mb/s	Estimated time
Cloud service (AWS)	10.79	4 days, 22 hours, 46 minutes, 33 seconds
Edge network (Local cloud)	1,176.7	1 hour, 5 minutes, 21 seconds
Local drive	1,443.39	0 hours, 53 minutes, 16 seconds

It can be concluded from this experiment that transferring large files within the local area network is more beneficial than sending files to a cloud service to be processed. Therefore, it adds weight to the decision to avoid cloud services to process large amounts of data.

The main issue with storing photos locally is the amount of space available on the retail robot. Since all systems are contained and the robot is mobile, the number of photos that can be stored is limited considering that they must be of high quality and enough photos must be taken as to not miss any important details. More storage space can always be purchased, but this also leads to increased costs, and purchasing extra storage for hundreds of units is very expensive.

5.4 Data Extraction

Successful data extraction using OCR and Machine Learning algorithms depends on relatively few factors. First, the cameras must be of high enough quality to take photos continuously while the platform the cameras mounted to are moving quickly. The photos also need to be able to be clear enough and focused on critical points to ensure high image quality. Secondly, the OCR program needs to be able to receive the images, extract the data it needs, and either store or discard the image as required by the task the robot is handling at that time quickly enough as to not cause a backup of queued images. Backups might cause an increase in processing time, lower image quality, and possibly loss of data due to overflow. The software which is used for data extraction specifically in-house is confidential, but there is some freely available OCR software available that can be used to test if data extraction is better to be done locally or through a cloud service. One such widely used software available online is Tesseract.

Tesseract is the most acclaimed open-source OCR engine of all and was initially developed by Hewlett-Packard. It's a free software under Apache license that's sponsored by Google since 2006 (Taylor). Tesseract will be used in conducting this next part of the test phase. A simple photo containing text will be selected and run through the tesseract program, and the time will be recorded. The test will be run using a local computer and if possible, on a cloud service. It is estimated that the cloud service will be able to perform the OCR function much more quickly given that cloud services have high performing computing hardware, but CPU usage is expensive especially if a high-powered service is used.

To install tesseract on a personal windows computer, it is as simple as downloading a package from a link on Windows machines, adding a PATH variable, and then using simple pip commands.

Assuming python and pip are installed and working:

1. Install the Tesseract package from the internet. (https://tesseract-ocr.github.io/tessdoc/Home.html)
2. Open the EXE file and follow the simple setup procedure.

3. Add the Tesseract source to the PATH variables on the computer (see Figure 16).

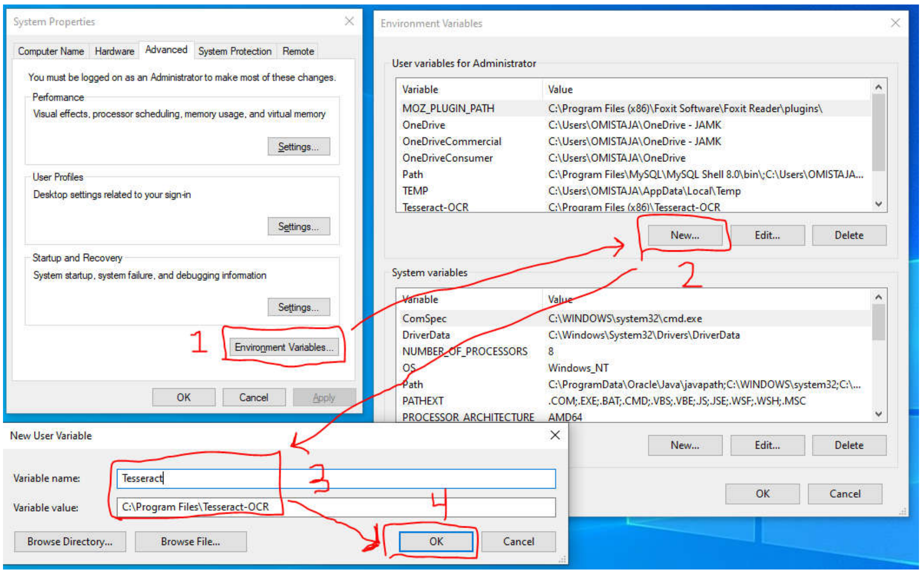

Figure 16. Path setup

If there are issues with exporting the path variable, usually by a failed command after setup is complete, a command to add the source to PATHS is possible:

```
setx path "%path%;c:\dir1\dir2"
```

4. Restart the computer
5. Confirm tesseract is installed by running the tesseract command on the cmd command line (see Figure 17)

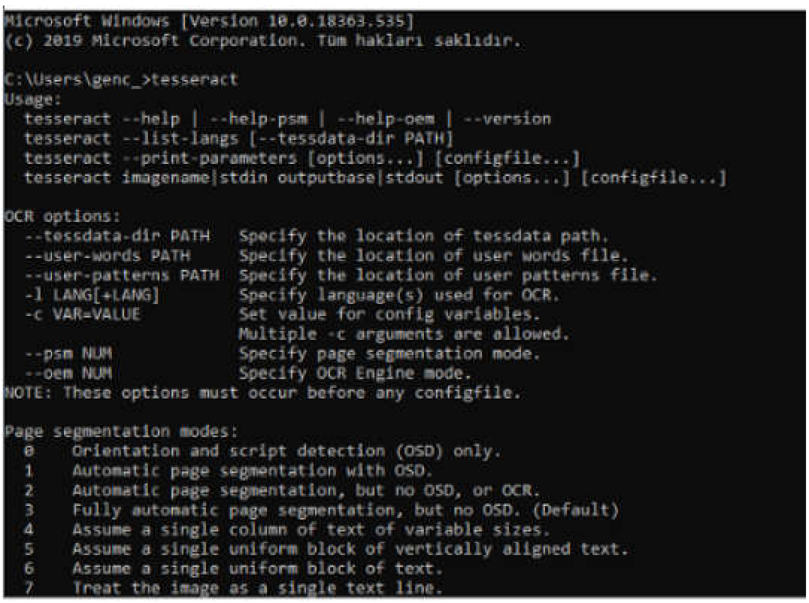

Figure 17. tesseract command on cmd

6. Finally, to use tesseract on python, a pytesseract package can be installed using a PIP, or Python Package Installer command:

```
pip install pytesseract or py -m pip install pytesseract
```

Tesseract can be tested by using the command

```
tesseract <image> <output file>.
```

In some cases, a dpi (dots per inch) parameter is needed. This can be retrieved by checking the image properties.

An open image of a receipt was saved off of the internet to be tested with Tesseract (see Figure 18). The image consists of some basic characters and numbers that Tesseract should be able to easily pick up and output to the destination file.

TARGET
EXPECT MORE PAY LESS.
OAHU HONOLULU ALA MOANA
05/12/2018 02:05 PM EXPIRES 08/10/18

GROCERY
267008011 BANANA FT $0.39
SUBTOTAL $0.39
T = HI TAX 4.7120% on $0.39 $0.02
TOTAL $0.41
CASH PAYMENT $0.41

REC#2-8132-2870-0169-3511-3 VCD#759-280-869

Everything come up roses?

Help make your Target Run better.
Take a 2 minute survey about today's trip:
informtarget.com
User ID: 7186 7713 0983
Password: 064 887
CUÉNTENOS EN ESPAÑOL
Please take this survey within 7 days.

Figure 18. Open source sample image for testing Tesseract

After the image is saved, the dpi of the image is retrieved from the image properties, and then a command is carried out to extract the characters from the image:

`tesseract --dpi 96 Target_Receipt.jpg target`. The command takes a moment to execute but it works without issues (see Figure 19).

```
C:\Users\Corey\OneDrive\Desktop>tesseract --dpi 96 Target_Receipt.jpg target
Tesseract Open Source OCR Engine v5.0.0-alpha.20200328 with Leptonica

C:\Users\Corey\OneDrive\Desktop>
```

Figure 19. Successful execution of a Tesseract command

From the cmd command, it is clear that the command was successful. The resulting text output was very close to what the image has.

```
EXPECT MORE. PAY LESS:
@TARGET
OAHU HONOLULU ALA MOANA
05/12/2018 02:05 PM EXPIRES 08/10/18
UNIAN
GROCERY
267008011 BANANA FT —- $0.39
SUBTOTAL 0.39
T = HI TAX 4.7120% on $0.39 Se
TOTAL 0.41
CASH PAYMENT $0.44.
REC#2-8132-2870-0169-3511-3 VCD#759-280-869
Everything «'
comeup  *i
roses?
Help make your Target Run better.
Take a 2 minute survey about today's trip:
Y
UUd
informtarget.com
User ID: Wigs 7718 0983
Password: 064 887
CUENTENOS EN ESPANOL
Please take this survey within 7 days.
```

While the command was successful, the text is not entirely clear and sometimes mixed. However, all the important information is there including items and prices. It is difficult to understand how long it takes to execute the command. PowerShell has a tool to measure the time it takes for a command to run, called Measure-Command. The previous test was then run using the available command via PowerShell:

```
Measure-Command {tesseract --dpi 96 Target_Receipt.jpg
target | Out-Default}
```

The results show the time it takes to execute the command in several different formats, including days, minutes, milliseconds, and ticks amongst others (see Figure 20).

```
PS C:\Users\Corey\OneDrive\Desktop> Measure-Command {tesseract --dpi 96 Target_Receipt.jpg target | Out-Default}
Tesseract Open Source OCR Engine v5.0.0-alpha.20200328 with Leptonica


Days              : 0
Hours             : 0
Minutes           : 0
Seconds           : 0
Milliseconds      : 803
Ticks             : 8032070
TotalDays         : 9.29637731481481E-06
TotalHours        : 0.000223113055555556
TotalMinutes      : 0.0133867833333333
TotalSeconds      : 0.803207
TotalMilliseconds : 803.207
```

Figure 20. Successful Measure-Command execution

Running the Tesseract command in several successions using various settings on a pc with 16 Gb of RAM produces results of around 800 milliseconds. A faster gaming computer can process the same image in about half the time.

Using Tesseract in a python program is also fairly simple, many guides online such as nanonets show examples of a successfully working Tesseract function in just a few lines (Zelic, Sable 2020):

```
import cv2

import pytesseract

img = cv2.imread('image.jpg')

# Adding custom options

custom_config = r'--oem 3 --psm 6'

pytesseract.image_to_string(img, config=custom_config)
```

There are guides available such as nanonets to preprocess the image to reduce background noise, correct skew, and enhance the colors to make the image easier to read for Tesseract. The example below shows a code using all of these features (Zelic, Sable import cv2

```
import numpy as np

img = cv2.imread('image.jpg')

# get grayscale image

def get_grayscale(image):

    return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# noise removal
```

```
def remove_noise(image):

    return cv2.medianBlur(image,5)

 #thresholding

def thresholding(image):

    return cv2.threshold(image, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)[1]

#dilation

def dilate(image):

    kernel = np.ones((5,5),np.uint8)

    return cv2.dilate(image, kernel, iterations = 1)

#erosion

def erode(image):

    kernel = np.ones((5,5),np.uint8)

    return cv2.erode(image, kernel, iterations = 1)

#opening - erosion followed by dilation

def opening(image):

    kernel = np.ones((5,5),np.uint8)

    return cv2.morphologyEx(image, cv2.MORPH_OPEN, kernel)

#canny edge detection

def canny(image):

    return cv2.Canny(image, 100, 200)

#skew correction

def deskew(image):

    coords = np.column_stack(np.where(image > 0))

    angle = cv2.minAreaRect(coords)[-1]

     if angle < -45:

        angle = -(90 + angle)

    else:

        angle = -angle

    (h, w) = image.shape[:2]
```

```
    center = (w // 2, h // 2)

    M = cv2.getRotationMatrix2D(center, angle, 1.0)

    rotated = cv2.warpAffine(image, M, (w, h), 
flags=cv2.INTER_CUBIC, borderMode=cv2.BORDER_REPLICATE)

    return rotated

#template matching

def match_template(image, template):

    return cv2.matchTemplate(image, template, 
cv2.TM_CCOEFF_NORMED) 2020).
```

Online services such as AWS Textract offer much more powerful OCR software than what is generally available for free. Textract claims it can extract tables, forms, and other structured data without custom code. Other OCR software available can do these things, but some training may be necessary. While testing Textract is possible, it is a paid service and requires sign-up with the credit card information to use. On the other hand, one source available online states that Amazon Textract is capable of scanning millions of pages of documents in just a few hours (Moon 2019). This indicates that using a cloud service to do image processing is much faster due to the seemingly limitless capabilities that the cloud has. Using the scenario in chapter 4.1, it can be calculated roughly how long it can take to process some images in a real-world environment. Table 6 shows a rough calculation using available resources.

Table 6. Comparing Local and Cloud CPU power

OCR Service	Time to process one image	Time to process 50000 images
Tesseract	800ms	100 hours
Textract	0.4ms	20 seconds

While Textract is faster here, it is notable that Cloud CPU usage can be expensive. A slower computing service may be available to suit the processing needs while saving money.

Processing power is not needed just for images. All other systems onboard a retail robot must share the CPU available to function properly. Lights, Cameras, Sensors, Displays, Modems, Motors for example are just as dependent on a well-functioning computer system with available processing power as image processers. While Tesseract is a good OCR example, Other OCR systems or one specifically designed for the retail robot needs may cut the time down enough to compete with some paid services like Textract.

6 Recommendations

Considering the data gathered earlier, it would be in the best interest of Solteq to eventually adopt a fog computing approach, with some edge and cloud computing where it best benefits the company. Fog computing has several pros that take the qualities from both cloud computing and edge computing which fill in key gaps and weaknesses that the other platform lacks.

- Smaller devices are placed over a distributed network
- Each device processes its own collected data
- Processed data is sent to a more centralized edge device or cloud service which will compile the data, prepare the results, and handle the release
- A centralized database or cloud service will store the data as needed

There are several main reasons that fog computing should be considered in Solteq.

1. The robot network will be widely distributed
2. The heavy computing power required to process the images would be most beneficial to both Solteq and their customers if it is done in the vicinity of each robot.
3. The photos a robot takes are large in terms of memory space, and it would be cheaper to keep the photos close to the customer vs. sending them to a cloud service and back.
4. Sending the data to a cloud service may result in high bandwidth costs and potentially cause unnecessary delays between data gathering runs and release.

5. It is safest for customers and objects around the robot if it can make its own decisions without having to wait for a remote device to instruct the robot how to react to an obstacle.

Cloud computing services can still be used as a central location for data gathering and storage, but all other heavy processes should eventually be handled using local edge devices in a fog network.

Another fact to consider is that the use of cloud computing services is becoming more heavily used worldwide, and issues with latency are increasing. Thus they are becoming more expensive for companies to invest in, which will make edge computation and storage more attractive in the long run. Likewise, a pure edge computing system would be more suitable if all of the data came from one source.

In practical terms, the robot network which is distributed over a large network should handle their image processing and send the lightweight data onwards to be collected and released. First, the robot will run scanning operations and take photos. After the photos are collected, they are processed onboard the robot next. Any data relevant to customer needs are extracted from the photos are collected in one location or list, and sent to a central database in a central location such as a cloud service, whether the cloud service is sourced in-house by Solteq or hosted elsewhere. Next, an application created by Solteq for their clients fetches the data in the cloud and processes the results to be released to the customer. Photos that are meant to be viewed by the customer should be sent directly to the user interface avoiding cloud service use as much as possible (see Figure 21). The user interface can also be used to monitor the robots themselves in tasks such as checking robot status, battery levels, and scheduling tasks to the fleet to be carried out.

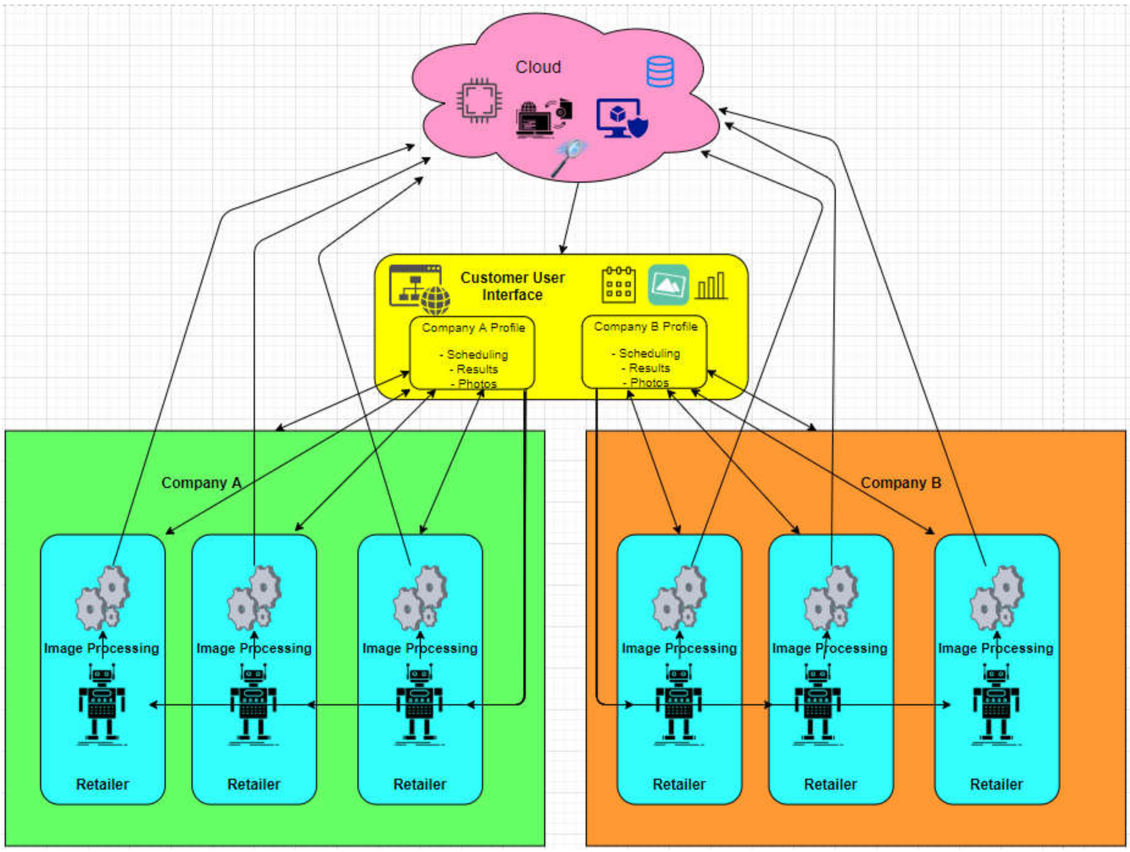

Figure 21. Suggested data flow

This would enable Solteq to better prioritize the data processing that needs to remain up to date as quickly as possible while saving the company valuable recourses that no longer need to be sent to cloud services to be processed. The focus of the cloud services would be mostly on storage of the processed data collected by the robot fleet during their runs, and the distributed network of edge devices would keep the heavier data such as photos and raw unprocessed data in the clients' vicinity, keeping bandwidth usage to a minimum.

An application and user interface for the robots used to store important photos, results and alerts, set up tasks and schedules, and see robot status, is most easily accessible if hosted by a cloud service but issues with latency and bandwidth use can be avoided if the application is hosted by a local data center.

7 Conclusion

As cloud computing services are becoming more widely used, they are also becoming more expensive, and many problems are beginning to surface as a result. For example, some services are starting to lag due to increasing amounts of traffic. Critical results are taking longer to return to customers, causing delays in their work and development. Thus, many businesses are turning away from cloud computing and to edge and fog computing platforms to cut cloud computing costs and data latency. Edge computing and fog computing are more useful for companies that need to handle high amounts of data and need to keep it up to date. These platforms also enable companies to keep important data close to their customers, allowing them quicker and more secure access than if the data was sent to a cloud service.

At Solteq, the high amounts of photo processing should discourage the use of cloud computing as much as possible. The amount of bandwidth needed to send the photos back and forth to the cloud would eventually become very expensive as the robot network expands. An Edge computing or Fog computing platform located in the vicinity, or onboard each robot could handle the photo processing and send the lighter processed data to the cloud service. The cloud service should only be used to receive and store the data captured by the robots to keep the data sent there as light as possible, and any photos that are meant to be seen by the clients should be sent directly to a more localized database and user interface that the customer will have access to, avoiding the use of cloud computing. The user interface created for customers would then fetch the data that the cloud service receives, and the results of the data would be gathered and published within the edge or fog computing platform.

Writing an effective thesis in this field for a technology company providing several unique challenges that had to be overcome to deliver an effective thesis for both the company and the institution where this thesis was written. On one hand, it was hoped that the topic of the thesis would be less common, avoiding subjects that were written over and over again. Given the authors' background in quality assurance and testing, it would have been an easy choice, but instead, it was hoped that this thesis would be more unique, thus more interesting to the institution and potential readers, potential clients, and even competitors. The next challenge to be overcome was the fact that this thesis had to be publishable while maintaining a level of secrecy that the company expects from a very high-profile project in the prototype phase. This thesis also had to maintain a more technical viewpoint hoped by the institution, and not one that is more of a marketing standpoint. The idea of the topic came from a colleague at Solteq who had planned to study some of the different computing platforms but had no time to carry out the study.

Furthermore, the Coronavirus pandemic that was ongoing at the time this thesis was written further complicated the situation as the authors' contract with the host company was terminated due to financial complications caused by the pandemic. This changed the focus of the thesis from one that was aiming to carry out experiments to gather physical data to use in the reports, to one that had to be more theoretical.

Many of the ideas presented in this paper may or may not have been considered or decided upon at the time this thesis is submitted. The goal of this thesis was to help Solteq gain a little deeper understanding of what direction they may be able to take when the topic of computation platforms are brought up in the future. Any experimental work which was originally planned to be carried out for the thesis was severely delayed or canceled as a result of the coronavirus pandemic. The experimentation could be carried out as part of a future thesis topic, but the results of the experimentation may not be publishable due to company confidentiality practices.

Working at Solteq Oyj was a great experience and a great opportunity to grow as a developer. The working atmosphere was relaxed, the people are great to work with, and there are several opportunities for any young developer to grow in their field of expertise. It was a great honor to be able to work with leading-edge technology and to be a part of something that will become common in everyday lives soon. The retail robot project is expected to begin use in stores across Finland within a few years.

8 References

Bonomi, F. The Eighth ACM International Workshop on VehiculAr Inter-NETworking. VANET 2011. Referenced 08.11.2020. https://www.sigmobile.org/mobicom/2011/vanet2011/program.html

Bruno. 2019. Everything You Need To Know About PC Bottlenecks. PC-Builds 11.1.2019. Referenced 1.3.2020. https://pc-builds.com/everything-you-need-to-know-about-pc-bottlenecks/.

Bauman, A. Chakrabarti, M. Schroeder, A. 'I Could Scrub Rings Around It': Walmart Employee Slams Grocery Store Robots. Wbur On Point 04.11.2019. Referenced on 08.06.2020. https://www.wbur.org/onpoint/2019/04/11/cleaning-robots-grocery-stores-walmart

Carson, E. Robots could replace humans in a quarter of US jobs by 2030. cNet 24.1.2019. Referenced on 11.8.2020. https://www.cnet.com/news/robots-could-replace-humans-in-a-quarter-of-us-jobs-by-2030/

Dataflare Team. Features of Cloud Computing – 10 Major Characteristics of Cloud Computing. Dataflare Training 5.3.2019. Referenced 19.4.2020. https://data-flair.training/blogs/features-of-cloud-computing/.

Dizikes, P. How many jobs do robots really replace?. MIT News Office 14.5.2020. Referenced 11.8.2020. http://news.mit.edu/2020/how-many-jobs-robots-replace-0504

DNSstuff. 2020 Guide to Network Latency – How to Check, Measure, and Reduce Network Latency. DNSstuff 19.08.2019. Referenced on 08.11.2020. https://www.dnsstuff.com/network-latency

Fukushima, K. Panasonic introduces robotic checkout at a grocery store in Osaka. CNBC 14.12.2016. Referenced 08.06.2020. https://www.cnbc.com/2016/12/14/panasonic-introduces-robotic-checkout-at-a-grocery-store-in-osaka.html

Gartner. 2020. Gartner Glossery: Edge Computing. Referenced on 5.12.2020. https://www.gartner.com/en/information-technology/glossary/edge-computing.

Guru99. 2020. Top 21 Cloud Computing Service Provider Companies in 2020. Publish date unknown. Referenced on 18.4.2020. https://www.guru99.com/cloud-computing-service-provider.html.

IFS Group. 2016. IFS ACQUIRES MAINIOT SOFTWARE OY. IFS Group 1.3.2016. Referenced 19.4.2020. https://www.ifs.com/corp/news-and-events/newsroom/2016/03/01/08/01/geofabrics-australasia-deploys-ifs-applications-9-to-benefit-from-embedded-crm-capabilities/.

Kann, B. Thompson, R. Thomas C. Dicker, A. Aneja, S. Artificial Intelligence in Oncology: Current Applications and Future Directions. Cancernetwork 15.2.2019. Referenced 27.7.2020. https://www.cancernetwork.com/view/artificial-intelligence-oncology-current-applications-and-future-directions

Karttunen, M. 1999. Tiedonhallinta aikoo pörssiin. Tivi Uutiset 6.8.1999. Referenced on 19.4.2020. https://www.tivi.fi/uutiset/tiedonhallinta-aikoo-porssiin/6154e853-e80d-3a29-a06c-144f6728c880.

Keary, T. 2018. What is Fog Computing? ITPRC 06.10.2018. Referenced on 03.07.2020. https://www.itprc.com/fog-computing/

Kumar, N. Seadally, S. Rodrigues, J. Vehicular Delay-Tolerant Networks for Smart Grid Data Management Using Mobile Edge Computing. IEEE Communications Magazine 54.10 14.10.2016, 60-66. Referenced 08.06.2020. https://ieeexplore.ieee.org/abstract/document/7588230.

Kupsa, T. 2018. Tekoäly muuttaa kaupan alan – otetaan rohkeasti mallia Walmartista. Kauppalehti 01.06.2018. Referenced on 06.08.2020. https://blog.kauppalehti.fi/solteq-yksinkertaisempaa-digitalisaatiota/tekoaly-muuttaa-kaupan-alan-otetaan-rohkeasti-mallia-walmartista

Latxague, F. 2013. A brief history of robots. Paris Innovation Review 18.7.2013. Referenced on 5.3.2020. http://parisinnovationreview.com/articles-en/a-brief-history-of-robots.

Microsoft Azure. 2020. What is cloud computing? A beginner's guide. Referenced on 5.12.2020. https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/.

Moon, M. Amazon's Textract AI can read millions of pages in a few hours. Engadget 30.05.2019. Referenced on 19.08.2020. https://www.engadget.com/2019-05-30-amazons-textract.html

Saadetdin, A. 2000. Solteq Oyj Vuosikertomus 2000. Published 19.2.2001. Referenced on 19.4.2020. https://vuosikertomukset.net/resources/Solteq/fin/vuosikertomukset/Solteq_vuosikertomus_2000.pdf.

Sengupta, Y. 2018. Taking stock: How technology is transforming inventory management in retail. Programmable Business 17.04.2018. https://programmablebusiness.com/taking-stock-how-technology-is-transforming-inventory-management-in-retail-c5664ea9875b

Shaw, K. 2019. What is edge computing and why it matters. Network World 13.11.2019. Referenced on 5.12.2020. Referenced 06.08.2020. https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html.

Smith, K. 2018. Automaton on aisle 6! Walmart is tracking its inventory with high-tech retail robots. Los Angeles Daily News 21.3.2018. Referenced 06.08.2020. https://www.dailynews.com/2018/03/21/automaton-on-aisle-6-walmart-is-tracking-its-inventory-with-high-tech-retail-robots/

Solteq Oyj. 2020. Acquisitions: Our most significant acquisitions. 2020. Referenced on 19.4.2020. https://www.solteq.com/en/investors/solteq-as-an-investment/acquisitions/.(IFS group sale)

Solteq Oyj. 2020. Video: Solteq Retail Robot. YouTube 8.1.2020. Referenced on 27.2.2020. https://www.youtube.com/watch?v=nda7bTLNcQs.

Solteq Retail Robot. 2020. Referenced on 19.4.2020. https://www.solteq.com/en/robotics/solteq-retail-robot.

Tavan, C. 2019. Pressikuva Solteq Retail Robot-2. Cision News 25.9.2019. Referenced on 12.5.2020. https://news.cision.com/fi/solteq/i/pressikuva-solteq-retail-robot-2,c2633884

Taylor, K. List of Top Open Source OCR Tools. HighTechNectar. Publish date unknown. Referenced 14.8.2020. https://www.hitechnectar.com/blogs/open-source-ocr-tools/#GOCR

Väätänen, O. 2019. Solteq Oyj Vuosikertomus 2018. Solteq 27.2.2019 Referenced 18.4.2020. https://cdn2.hubspot.net/hubfs/484933/wp-content/uploads/2019/03/1010173.pdf.

Väätänen, O. 2020. Tilintarkastuskertomus Solteq Oyj:n yhtiökokoukselle. Published 31.3.2020. Referenced 18.4.2020. https://www.solteq.com/fi/sijoittajat/tiedotteet/solteq-oyjn-tilintarkastuskertomus-2019.

w.media. 2018. Edge computing: Advantages & Disadvantages. w.media 5.2018. Referenced 21.5.2020. https://w.media/wp-content/uploads/2018/05/Advantage-and-Disadvantage-of-Edge-Computing.pdf.

Yakowicz, W. Pepper the Robot Is Coming to Work in U.S. Retail Stores, and Then Your Home. INC 19.10.2016. Referenced on 08.06.2020. https://www.inc.com/will-yakowicz/pepper-the-robot.html

Zelic, F. Sable, A. A comprehensive guide to OCR with Tesseract, OpenCV and Python. Nanonets 02.08.2020. Referenced 20.8.2020. https://nanonets.com/blog/ocr-with-tesseract/

9 Appendices

Appendix 1. Latencies from Google Cloud Platform

GCP Region Name	Region Code	Mean	Median	Min	Max	Test 1	Test 2	Test 3	Test 4	Test 5
USA (**Nevada**)	*us-west4*	**239 ms**	**230 ms**	**206 ms**	**505 ms**	505 ms	233 ms	212 ms	248 ms	241 ms
USA (**Utah**)	*us-west3*	**239 ms**	**227 ms**	**207 ms**	**877 ms**	394 ms	229 ms	222 ms	261 ms	224 ms
Finland (**Hamina**)	*europe-north1*	**241 ms**	**220 ms**	**138 ms**	**1084 ms**	1084 ms	138 ms	196 ms	281 ms	201 ms
Germany (**Frankfurt**)	*europe-west3*	**275 ms**	**258 ms**	**214 ms**	**723 ms**	364 ms	542 ms	218 ms	307 ms	215 ms
Netherlands (**Eemshaven**)	*europe-west4*	**275 ms**	**260 ms**	**89 ms**	**646 ms**	646 ms	507 ms	236 ms	351 ms	258 ms
Switzerland (**Zürich**)	*europe-west6*	**279 ms**	**263 ms**	**222 ms**	**480 ms**	480 ms	335 ms	244 ms	306 ms	268 ms
Belgium (**St. Ghislain**)	*europe-west1*	**281 ms**	**258 ms**	**227 ms**	**862 ms**	407 ms	285 ms	227 ms	291 ms	244 ms
UK (**London**)	*europe-west2*	**291 ms**	**278 ms**	**238 ms**	**499 ms**	305 ms	439 ms	260 ms	393 ms	263 ms
Canada (**Montreal**)	*northamerica-northeast1*	**518 ms**	**502 ms**	**166 ms**	**797 ms**	567 ms	790 ms	476 ms	627 ms	472 ms
USA (**Northern Virginia**)	*us-east4*	**524 ms**	**503 ms**	**464 ms**	**714 ms**	619 ms	494 ms	483 ms	473 ms	571 ms
USA (**Iowa**)	*us-central1*	**580 ms**	**562 ms**	**243 ms**	**1029 ms**	793 ms	258 ms	567 ms	601 ms	583 ms
USA (**South Carolina**)	*us-east1*	**582 ms**	**541 ms**	**179 ms**	**1543 ms**	564 ms	657 ms	575 ms	563 ms	555 ms
Brazil (**São Paulo**)	*southamerica-east1*	**621 ms**	**850 ms**	**288 ms**	**1323 ms**	1323 ms	369 ms	871 ms	291 ms	902 ms
USA (**California**)	*us-west2*	**653 ms**	**651 ms**	**218 ms**	**1047 ms**	350 ms	691 ms	649 ms	681 ms	728 ms
USA (**Oregon**)	*us-west1*	**658 ms**	**653 ms**	**215 ms**	**1007 ms**	372 ms	796 ms	216 ms	698 ms	698 ms
Japan (**Osaka**)	*asia-northeast2*	**659 ms**	**898 ms**	**305 ms**	**1342 ms**	1005 ms	354 ms	944 ms	314 ms	1018 ms
Japan (**Tokyo**)	*asia-northeast1*	**662 ms**	**891 ms**	**301 ms**	**1374 ms**	1013 ms	848 ms	929 ms	333 ms	1004 ms
Taiwan (**Changhua County**)	*asia-east1*	**705 ms**	**958 ms**	**331 ms**	**1345 ms**	1063 ms	386 ms	1028 ms	343 ms	1009 ms
South Korea (**Seoul**)	*asia-northeast3*	**719 ms**	**972 ms**	**328 ms**	**1487 ms**	1045 ms	468 ms	985 ms	337 ms	1040 ms
Hong Kong (**Hong Kong**)	*asia-east2*	**748 ms**	**1012 ms**	**340 ms**	**1577 ms**	1151 ms	589 ms	1087 ms	379 ms	1113 ms
Australia (**Sydney**)	*australia-southeast1*	**759 ms**	**1026 ms**	**346 ms**	**1561 ms**	1333 ms	411 ms	1040 ms	371 ms	1153 ms
Singapore (**Jurong West**)	*asia-southeast1*	**784 ms**	**1076 ms**	**361 ms**	**1826 ms**	1298 ms	457 ms	1127 ms	371 ms	1167 ms
India (**Mumbai**)	*asia-south1*	**918 ms**	**1253 ms**	**427 ms**	**1882 ms**	1352 ms	561 ms	1276 ms	427 ms	1341 ms