Literature study assignment

Coordinators: Saba Amiri, Adam Belloum,

Note:

Following are reports of the Literature study assignment part of course "Web Services and Cloud Systems"[1] given in the context of the Joint UvA-VU Computer Science program[2]. The literature assignment is worth 35% of the total course grade. Students have to read at least 17 papers and prepare a (8-10)-page report in a style of a scientific publication[3], and give 15 mn presentation at the end of the course. The literature topics are not covered during the lectures, students use the knowledge acquired during the lectures to perform the literature study. To introduce the students to scientific paper analysis, 4 scientific papers are analysed and discussed during the lecture hours. Reports are checked for plagiarism using Trinity tool integrated in Canvas (similarity score tolerated is max 20%).

[1] https://studiegids.uva.nl/xmlpages/page/2020-2021/zoek-vak/vak/79525
[2] https://masters.vu.nl/en/programmes/computer-science-big-data-engineering/index.aspx
[3] Formatting requirements: NeurIPS 2019 conference. More information and LaTeX templates can be found here: https://nips.cc/Conferences/2019/PaperInformation/StyleFiles

Distributed cloud based ML and DL workflows

Niels Galjaard
10465189
University of Amsterdam
Amsterdam
The Netherlands
niels.galjaard@student.uva.nl

Iliya Georgiev
13399640
University of Amsterdam
Amsterdam
The Netherlands
iliya.georgiev@student.uva.nl

Georgi Stoilov
12830313
University of Amsterdam
Amsterdam
The Netherlands
georgi.stoilov@student.uva.nl

Abstract

This study evaluates different machine learning and deep learning workflows in the cloud. The complexity of those is increasing in the past few years and therefore developers are trying to define some standardized processes to handle this. It considers the prescriptive process documentation of several cloud vendors and early process description and assesses them based on known problems that occur in machine learning systems. The authors set criteria that they based the evaluation upon and present an overview of how well the different architectures are performing. The study concludes that the ML/DL workflows that are based on the Kubernetes are performing the best according to the defined criteria.

Introduction

From the early 90s, there was a clear understanding among what we now call data scientists to standardize procedures. With funding from the European Union early standardized procedures were created. These claim to have several benefits for both service providers and customers of various data science services [1]. The common understanding can serve as a reference point for discussion, which will increase the understanding of crucial issues by all participants. Customers will feel more comfortable due to the impression of a well-established practice, which will reduce the need to educate customers on data science-specific topics. This allows the discussion to focus on where to apply data science. There are also several benefits to practitioners, since they can have a structure of the many tasks to be completed. Which will help prevent simple mistakes or wrongful omissions of tasks. It will also aid in the cooperation of people with a varied set of skills.

The need for a standardized process

As can be seen in the experience of past experts and current large companies, there is a clear need to get every member of a team an understanding of the whole process of ML/DL when integrating such systems into other software engineering enterprises. Moreover, due to the rarity of experts in these fields there is often a need for people in other roles to contribute to large ML/DL projects [2] [3]. Standardization of the process or workflow of exploring, developing training, deploying and monitoring ML/DL solutions can help create a common understanding and prevent disruptions during developments [1]. This will allow experts of certain subsets of ML/DL tasks to contribute to the

Preprint. Under review.

development of complex ML/DL systems. Another reason for standardization is the well-known need for a convention to maintain software systems [4]. In fact many ML/DL systems have their own software maintenance issues resulting directly from the use of ML/DL [5]

Evaluating frameworks

In this paper, we set out to judge and compare several different frameworks and a baseline of an early EU project. The reasoning for this is that unlike software development, which is overrun by terms like agile, DevOps, kanban and many other organizational workflows. In ML/DL the workflow is inherently related to the service provider. Many libraries, frameworks and other service providers supply their own tutorials, evaluation metrics and prescriptive multi-step processes. Because there are three main situations running software in general, your own servers, managed servers or a specific cloud vendor. We have chosen to also include the specific proprietary version by the world's largest cloud provider.

Criteria

Because of what is outlined in several sources [6][5] [7]. The issues during the development of ML/DL systems and their maintenance often arise from details of the frameworks and libraries used in those systems. The need for integrated data management solutions [6] is not captured only by a descriptive framework outlining technology-agnostic solutions. Yet the issue of hidden feedback loops [5] can only occur with an iterative process that fails to recognize hidden feedback and fails to correct for this. Moreover the lines between workflow, library, framework, deployment and data management is somewhat blurred by the creation of libraries [8], infrastructure [9], deployment [10], workflows [11], certification and courses [12] by the same party or closely related parties. Hence we have chosen to evaluate several different frameworks and their prescribed sequence of steps based on papers that outline common issues with DL/ML systems. And from those, we have chosen the problems that occur as a result of technology or process choices, while issues related to DL/ML can occur as a result of sub-optimal choices on the part of a data scientist. For example, fostering unrealistic expectations. It is incredibly hard to judge particular processes or frameworks based on these kinds of problems.

Integrated data management

According to internal research in Microsoft [6] there is a need for integrated data management when dealing with ML. This is due to the need to rapidly experiment and thereby change data often. Having the same tool manage both allows you to tag and explain what data leads to which model. This is required when retraining or evaluating models in production. Any model of which you cannot find the exact data version it was trained on, is essentially a non-reproducible black box, which might have diverged completely from its intended purpose. Moreover, without a record of data there is little reusability and knowledge sharing that can be done. Any framework or workflow that does not provide or specify the need for mappings between data and models is essentially waiting for this issue to occur.

Composability

In ML systems many models can be composed together using simple hard-coded rules, this can be used to better explain the models and isolate errors to their respective models and modules of a software solution. Because this is a common pattern used in ML systems we will judge frameworks based on their ability to compose many models together. Note that given a single data science machine this is a trivial consideration, where you can simply add several modules to the main program. However, in the cloud this is something where developers might have to take into account the performance of their models, and the ability to provide timely predictions. For example, several models are all hosted on different servers. In this case, the performance might suffer greatly due to the need for many network hops. Here we can evaluate frameworks and processes based on their documentation and technical capability of composing models.

Boundary Erosion (coupling)

The erosion of boundaries is something that happens in real-world software. An example might be to create a large sequence of events that penetrate many layers of your software. This happens as a result of introducing throwaway code. [13] Code intended to be used experimentally and kept too long. Incremental changes without taking into account the drift in architecture can lead to this problem as well. The fact that ML is both new, incremental and experimental in current systems makes systems that include ML prone to a situation described as a big ball of mud [13].

In ML there are models that ingest and correlate many data points at once, hence they might ingest from many sources and create a dependency between them where there was none before. However, more dangerously is the situation in which ML is applied to data produced by services and subsequently made available to many internal teams. But this dependency is not explicitly documented, hence a change to server logging might break many models, which might in turn cause failures in many parts of the system. Frameworks that consume live data and publish, but have no logging and monitoring built-in run the risk of creating these unknown dependencies. Ideally, you would set up explicit monitoring of data access on both the model inputs and outputs to prevent creating a big ball of mud at the system level.

Hidden feedback loops

A model can influence its own new features and target variables. A classic example is the algorithms used in searching and ranking content on the internet. It is well known that users are more likely to click on highly ranked results and thereby it can skew new data collected for its own training. There can also be more difficult to recognize feedback loops such as two independently developed automatic traders influencing each other's buy and sell patterns as well as training data. We can evaluate frameworks based on their prescription to take this possibility into account. or the ability to completely remove direct feedback loops due to a non-iterative nature of the whole process. [5]

Baseline - CRISP-DM

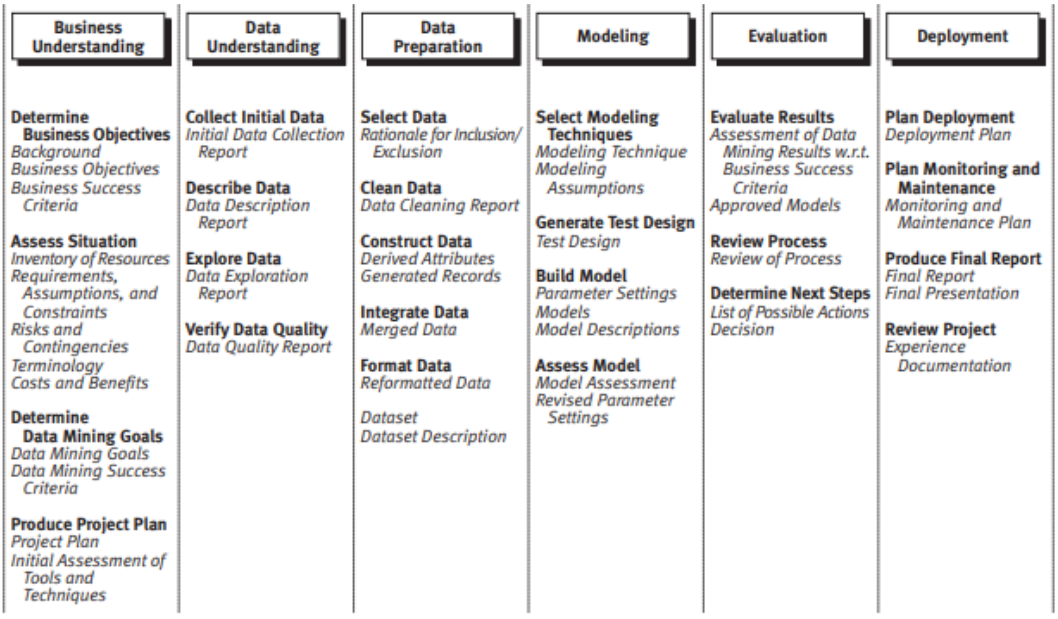


Figure 1: The CRISP DM V1.0

The CRoss Industry Standard Process for Data Mining is an early workflow for what was then called data mining, which was created with funding from the European Union in the 90s. [14] It outlines

the need for an industry-standard process of discovery in large datasets and is one of the earliest workflows that specifically aims to be a Datascience / ML workflow. The standard process describes phases, generic tasks, specialized tasks and process instances. The process also indicates that it's an idealized sequence of events and that in real-world use cases there may be a lot of backtracking and out-of-order events. The process also outlines a user guide for every step with a description of many pitfalls at every step. We consider this one of the first ML workflows. Earlier work is usually industry-specific rather than cross-industry [1]. We have chosen this as a baseline for comparison to other modern variants since it is an early development, which is based on the goal to generate reports, which are used to inform the business. However, it is still clearly a true ML workflow due to the usage of early ML models.

Process

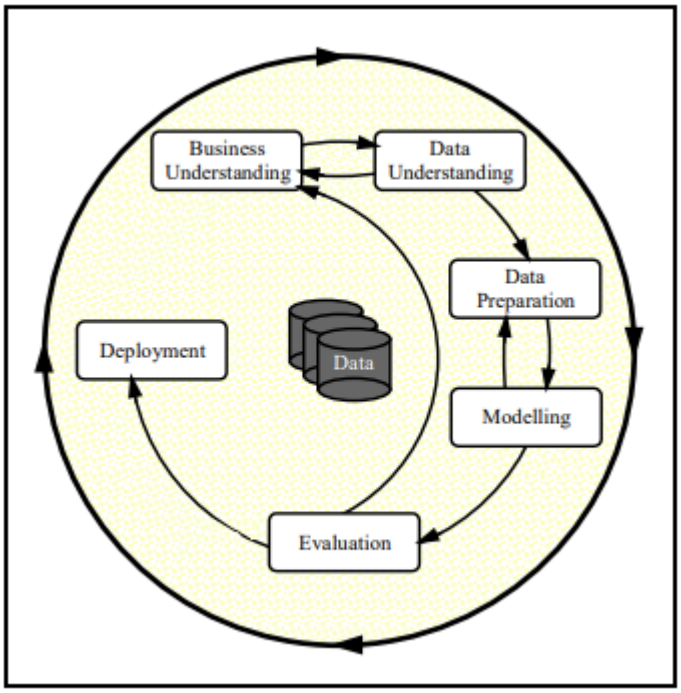


Figure 2: the crisp process summary

The CRISP-DM workflow identifies a six-step process of, Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. The old nature of this process is clear during the early phases, like Data Understanding, which aims to generate reports of data exploration, quality and data description. The process is linear towards a deployment, unlike later workflows there is a clear end phase of this process with Deployment, this is due to the early nature of this process. The specific end goal in mind is to generate a final report, final presentation and insight into the data. Meaning that the end result is not a real-time model, but a report or code to generate a report based on known formatted data. Moreover, there is no clear description of software engineering issues when creating systems to generate reports.

Criteria

In the pdf which outlines this workflow, there is an explicit mention of reporting the collection and cleaning of data, however, in step 3.4 the integration of data [14]. There is no explicit mention of a report or record of the integration process. This step creates the data set as its output. The specific mention of reports in other phases and steps but omission here can lead a reader to believe it is not required. So for example if a given table or column is used or if a particularly unique aggregation is used to create the final dataset. This will not be reported. Likewise, reordering or reformatting in step 3.5 also has no mention of reports or records of operations, hence this will also be undocumented. Likewise the dataset description produced in the end only described the fields of the created data, rather than the exact process of producing the final data or the location of the data. Since the location of the dataset is not outlined in the topics to be covered and there are several undocumented steps it will become hard to reproduce ML done by Crisp-DM. Since there are unknown aggregation and reformatting steps the boundary erosion and creation of unknown dependencies in the system is a risk. Similarly due to these unknown and undocumented operations you can easily compromise the ability to compose several models due to unknown reformatting. Because the outline of this process is not cyclical in nature there is no risk of direct feedback loops where ML models influence their own training data.

Serverless Frameworks

The increasing complexity of ML workflows requires developers and ML users to reach out to new solutions to tackle their daily challenges [15]. One possible solution to the difficulty of managing an ML workflow is serverless ML frameworks[16][17]. Serverless computing is also known as function as a service (FaaS) since instead of using servers, it is all based on functions on demand and it is completely stateless[18]. It was popularized by Amazon when the AWS Lambda was introduced [19]. There are numerous reasons why one would switch to serverless computing such as cost reduction, deployment process, easily scale up and down. Therefore, most of the biggest cloud suppliers have built serverless infrastructures to their cloud solutions - Google Cloud Functions [20], Microsoft Azure Functions [21] and IBM OpenWhisk [22]. In recent years, serverless computing has shifted the cloud platforms market and its market share is increasing at fast rates. This section reviews such frameworks that are based on serverless architectures and discusses how these match the criteria introduced in the previous sections.

Cirrus (based on AWS Lambda)

Cirrus is a machine learning framework created by UC Berkeley based on the AWS Lambda function. It offers an end-to-end solution for ML workflows by using a serverless infrastructure [23]. Cirrus comes with a lightweight worker runtime that matches the lambda granularity in order to provide developers to easily configure it to their requirements. The framework is built with the idea to minimize user effort and aims to reduce the amount of memory and storage that is required. Moreover, it combines the best of serverless and machine learning frameworks thanks to different contributions. The framework uses iterative distributed stochastic gradient descent to train the model and it constantly generates new model gradients that are fed into the training, until the model converges. Cirrus follows the following 4 design principles: Adaptive, fine-grained resource allocation, Stateless server-side backend, End-to-end serverless API and High scalability. The overview of the architecture of the framework is presented in figure 3.

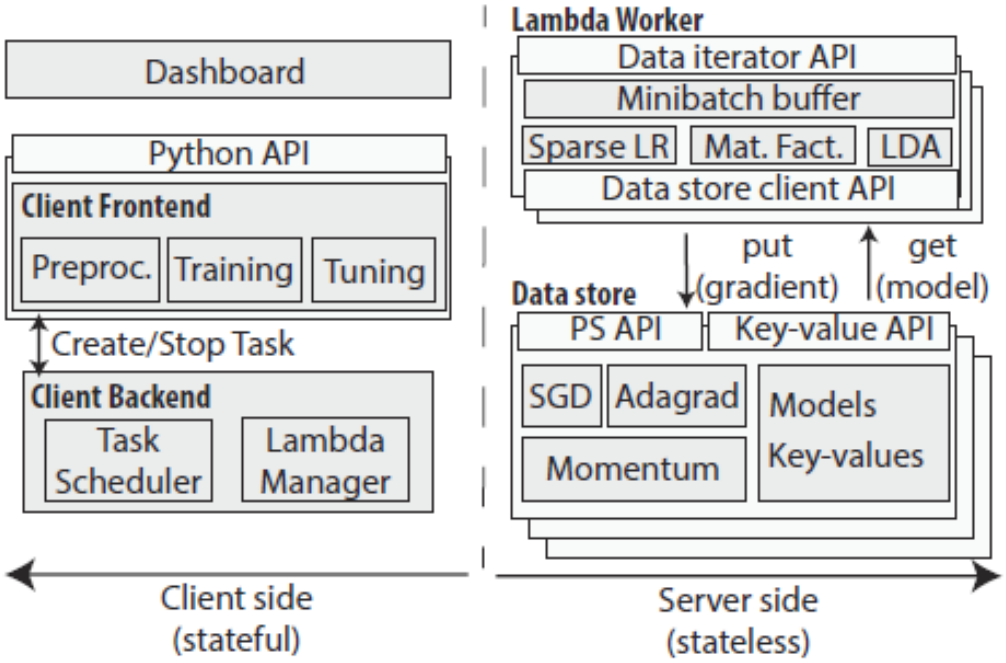


Figure 3: Cirrus overview

Process

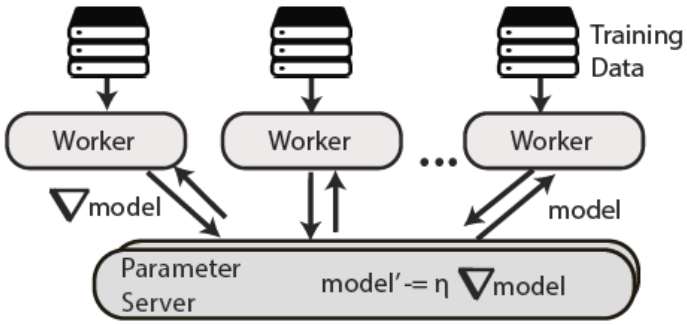


Figure 4: Cirus training

The Cirrus workflow consists of 3 stages:

- Data Loading and Preprocessing - It is assumed that the training data is stored in cloud storage such as S3 [24]. This data is converted to binary format data eliminating the need to deserialize at a later stage. Moreover, the framework divides the input into equal size parts and uploads it into an S3 bucket. In order to improve the training, Cirrus normalizes the dataset by dedicating one worker per input partition. In the end, the workers transform the partitions given the final per-column statistics. The amount of reads and writes that are generated is not supported by S3, therefore, Cirrus has its own low-latency distributed data store.
- Model training - To train the model the Cirrus uses a distributed SDG algorithm that workers run on lambda functions. This is shown on Figure 4. Each gradient computation takes two

variables as input - a minibatch and the most recent version of the model. The minibatches are retrieved from S3 using the Cirrus runtime iteration and the model is fetched from the data store. Each of the iterations produces a new gradient, which is stored in the data store to update the model.

- Hyperparameter optimization - Cirrus provides a hyperparameter search dashboard that monitors the model's loss convergence. In this way, the user has control over the training experiment and can adjust it if needed. This leads to limiting the use of serverless functions and cost reduction.

Criteria

The Cirrus framework looks into a lot of challenges in the ML workflows, such as memory constraints, high variance start time, lack of low-latency storage for models, etc. Unfortunately, not a lot of documentation is available on how to use the framework and therefore it is not really easy to work with it. The framework provides its own data management tool, which brings a lot of value compared to some other frameworks. This framework distributes models quite well and its latency is really low, in spite of this there are some fundamental limitations of the framework, the lack of RAM and total run time limits the size and number of models evaluated. Generally speaking chaining many network hops hampers performance, hence there is no way of composing large numbers of models in a single program with Cirrus. There is no information on the boundary erosion of Cirrus, however since it uses a general-purpose storage it is similar to SageMaker, both in the sense that there is a nice mapping between datasets and models and it is very easy to set up unintended dependencies.

AWS SageMaker

SageMaker is a section of services from Amazon [25] that comes with a series of white papers [26] and prescriptive documentation [11], describing the entire ML/DL process. As well as providing peripheral services, like tutorials, certification, hosting data storage, backups version control and more. Given the recent development of this service and the constant change at AWS we have considered the current documentation of June 2021 as the primary source of truth. We have chosen SageMaker with AWS documentation, since the documentation itself states that it is describing a typical ML workflow and has a dedicated section on workflows supported, as well as opinionated statements on several ML topics. [11].

Process

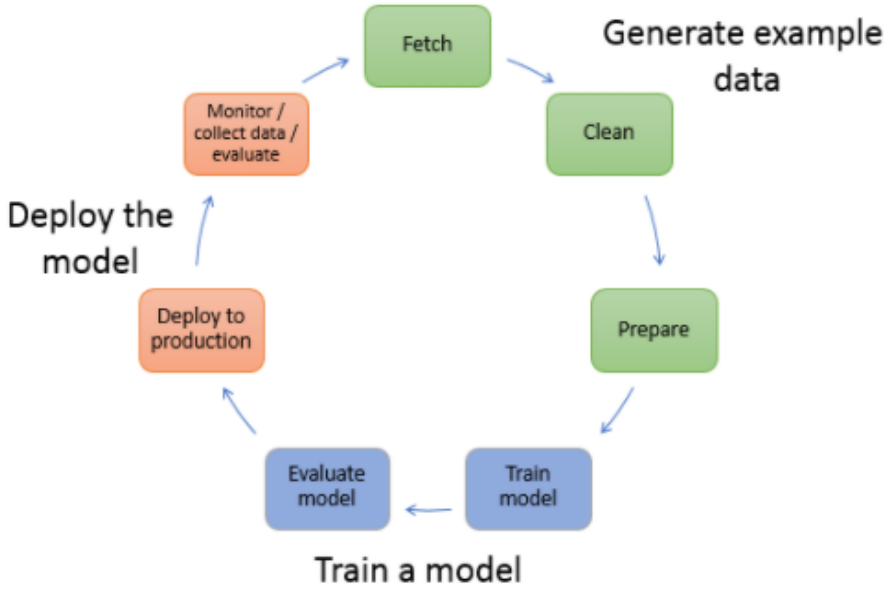


Figure 5: The SageMaker process

The workflow described by the AWS SageMaker documentation directly is not all that is supported by AWS [], however it is the one we consider in this paper. The workflow described is extremely iterative, it differs from many others in that it encourages to retrain on slight increases of data in short cycles. The documentation encourages the use of several ways to track your data sources, store data and persist the entire process of creating the dataset [27]

Criteria

SageMaker has many sections on pitfalls and issues that might occur in typical ML/DL workflows. indicating the required steps for regulatory compliance, such as bias monitor pre- and post- training and during production use. [] However there are some issues with the documentation outlined above. The constant use of easily accessible data sources to share predictions with other teams and services is encouraged [10]. However there is no mention of the dangers and pitfalls of this method eroding boundaries and creating undocumented dependencies on a system level. The documentation also encourages an extremely iterative process where models are trained on a short time scale with incremental data. There is no mention of possible feedback loops hampering the system. There is little talk of composing many models together in the AWS documentation, however the encouraging deployment strategies don't allow for latency-sensitive use cases. You can't deploy on many servers and serve predictions in real-time. It is possible to extract and host the models yourself, but this is somewhat cumbersome and no better than not using SageMaker. In spite of some issues SageMaker/AWS does provide well-integrated data management and versioning tooling. The documentation also highly encourages the proper use of mappings between data sets, models and the origin of datasets.

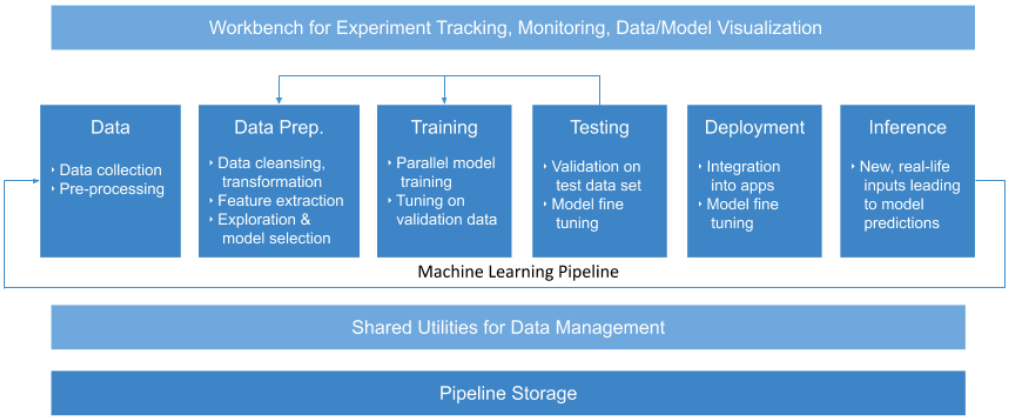


Figure 6: The agile stacks ML ops process

KubeFlow

KubeFlow in conjunction with MLOps KubeFlow is an open-source Cloud-Native platform for Machine Learning and Data Science. KubeFlow runs on top of Kubernetes, is fully integrated with it, and makes the most of the cluster's capabilities, such as auto-scalability and flexible resource control. KubeFlow includes many components that solve different Data Science and Machine Learning tasks. All of these components are Open Source, professionals can use them separately[28]. That is, working with KubeFlow, you can gain experience with these components and transfer them later to other tasks. Another integration of KubeFlow is that you can quickly launch JupyterHub on it and set up an individual environment for data scientists. This is very close to a normal working environment of a data scientist and hence requires much less time to learn.

Process

In the traditional approach, JupyterHub is often installed on a single powerful server, which can have 60-100 cores and several hundred gigabytes of RAM. But over time, as the Data Science department grows, resources for all begin to be insufficient. [29] In addition, the common environment leads to conflicts when some data scientists need to update versions of libraries or install new ones, while others do not.

The specific workflow to be used is not described on the KubeFlow site itself, however it is directly based on MLOps and it recommends agilestacks. And directly suggests to use agilestacks's method. Which notably has a highly iterative workflow in which many components are automated. Such as container management, autoscaling. The use of Kubernetes also allows the recreation of many past models.

Create a Jupyter notebook (in AWS or Azure), which allows us to use the free virtual machine Linux for fast recording and testing the model. You press two buttons and your environment is ready and can be used from any browser. There is no need to make sure you have the correct python version installed or are in the correct Conda environment. Start with a small amount of data, study it, work it out as a team. You can see some of my public ones here. The second step is to create a compute target (Azure VM or AWS EC) that you can train your models on. You can create an Azure VM, then in a notebook, give it an IP address, and boom, learning there. [30] Alternatively, you can spin up the workspace in the Azure ML service and now you have several different compute goals. The last step is to do an A / B testing to find the model that works and build a Dockerfile and build a container. From there, you present it as an HTTP endpoint that others are free to use, or store it behind some form of security / paid access using a tool like Azure Key Vault. It is important to deploy new experimental machine learning models quickly in production, otherwise the data will become obsolete and there will be problems with the reproducibility of experiments.

Criteria

Using KubeFlow inside cloud Kubernetes, the following problem is resolved: any Data Scientist can quickly deploy an experimental environment with the required amount of resources in a few clicks. And when the experiment is over, the resources are freed up and returned to the cloud. KubeFlow also allows to deploy fully isolated individual JupyterHub instances from separate Docker images. Every data scientist can customize the environment to suit his needs. It is important to note that some of the KubeFlow components are still in beta. Nonetheless, it is possible to start using KubeFlow progressively, because it is one of the few Production-Ready platforms that solve MLOps and machine learning problems. [29] To get started, KubeFlow can be used as a flexible version of JupyterHub, and then gradually get acquainted with the rest of the possibilities.

Because KubeFlow is based on MLOps principles it has a highly iterative nature, in fact many aspects are completely automated. Hence the threat of feedback loops is fairly high and in none of the tutorials or statements by KubeFlow is there a warning for such problems. The composability is also not great, but likely better than most other distributed systems, since Kubernetes is highly likely to schedule at least some pods on the machine. Thereby removing many network hops of a large model.[30] KubeFlow does recommend the usage of open exchange of model inputs and outputs, with very little dependency tracking, hence eroding boundaries. It does track its own data and has great version control for its models and their training data.

Discussion

Server vs Serverless

According to Wu et al. [31] serverless frameworks are a good option for model serving. The paper compares different frameworks based on server and serverless architectures. The authors provide an evaluation of these tools based on costs and performance. In this evaluation the paper shows that the AWS Lambda is outperforming the Google CF in the serverless frameworks comparison. The same is valid when comparing AWS Lambda to AWS SageMaker, the latter one performs better only in the beginning because of the cold start of the lambda functions. However, with the increase of the request rate the performance of the SageMaker drastically decreases, wherewith really high traffic the SageMaker is not able to keep up due to a high latency introduced by scaling operations. With the AWS Lambda, this is not the case, because of their serverless architecture the latency with a high request traffic is stable and it demonstrates a good elasticity. Wang et al [17] also prove that their serverless solution - Siren outperforms the traditional machine learning benchmark on a server with 44% at the same costs.

Unaddressed problems

The problems introduced by composing many models and the erosion of boundaries. are not addressed in the documentation or technical details of the frameworks under consideration. A cursory glance at GCP and Azure also shows that these are lacking in both categories. This might be because these are not always relevant to data scientists specifically. But even going through architecture whitepapers. We can't find any examples of warnings of these problems or suggestions on how to deal with them.

Missing criteria

The evaluation was based on a series of papers, which identified many common place issues with ML systems. We created our selection based on our opinion of what is due to choices of systems and workflows and what is not. We did this prior to reading the documentation and suggestions by the evaluated frameworks, hence some of their docs raise good points. Such as SageMaker's bias detection. KubeFlow's integration with JupyterHub and many more We chose not to include these in our evaluation. A new study could be done which includes a much broader set of possible problems, based on a consensus between many industry leaders, which would likely give better insight into which framework or workflow is lacking in the most important criteria.

Moreover given the recent development of ML and the even more recent developments of services, some of whom are launched even this year. We fully expect the areas of ML that create technical debt to surface in the coming decades. As long term problems are only revealed in the long term.

Overview

In this section, we present an overview of the frameworks that we have instigated. This overview can be found at Table 1.

Table 1: Overview of frameworks and their performance based on our criteria

Criteria	Baseline	Serverless	AWS SageMaker	KubeFlow MLOps
Integrated data management	none	depends on cloud provider	yes	yes
Composability	yes	high latency and likely requires extracting models	high latency and likely requires extracting models	not great, might work in cases of few nodes
Boundary Erosion	not mentioned	not mentioned	not mentioned	not mentioned
Hidden feedback loops	not iterative	likely	likely	likely

Based on Table 1 one can deduct that the frameworks that are based on Kubernetes(KubeFlow MLOps) are performing the best based on the criteria that are set in the paper. Furthermore, the KubeFlow framework brings the flexibility that every data scientist can deploy models within a few clicks.

Conclusions

From documentation and other prescriptive sources we have seen that some of the problems outlined in previous papers [5] [6] have been successfully addressed and communicated to end-users. Despite this, there are several cases where process documentation presents itself as a sequence of steps to be followed but does not adequately explain common pitfalls, such as the possibility of feedback loops. Moreover, many issues are at such a high level, the level of systems design and architecture that they are mostly relevant to professionals who are not data scientists. Due to the ever-changing nature of ML its job descriptions, workflows and its algorithms, we believe that many issues that arise in ML systems can only be solved by educating data science professionals and their managers on the pitfalls of their methods. This is particularly relevant given the fact that many of the pitfalls are intrinsically related to the use case. If we had to make a choice we would go with KubeFlow or some other open-source framework, simply because it allows disciplined practitioners to more easily create their own extension, which allows you to program in solutions to your specific problems.

References

[1] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, volume 1. Springer-Verlag London, UK, 2000.

[2] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. The emerging role of data scientists on software development teams. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, pages 96–107. IEEE, 2016.

[3] Meenu Mary John, Helena Holmström Olsson, and Jan Bosch. Developing ml/dl models: A design framework. In *Proceedings of the International Conference on Software and System Processes*, pages 1–10, 2020.

[4] Michael Smit, Barry Gergel, and H James Hoover. Code convention adherence in evolving software. In *2011 27th IEEE International Conference on Software Maintenance (ICSM)*, pages 504–507. IEEE, 2011.

[5] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. *Advances in neural information processing systems*, 28:2503–2511, 2015.

[6] Saleema Amershi, Andrew Begel, Christian Bird, Robert DeLine, Harald Gall, Ece Kamar, Nachiappan Nagappan, Besmira Nushi, and Thomas Zimmermann. Software engineering for machine learning: A case study. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300. IEEE, 2019.

[7] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. Data scientists in software teams: State of the art and challenges. *IEEE Transactions on Software Engineering*, 44(11):1024–1038, 2017.

[8] sagemaker team. *sagemaker SDK*.

[9] various aws teams. *aws infrastructure*.

[10] sagemaker team. *sage maker deployment*.

[11] sagemaker team. *sage maker workflow*.

[12] AWS. Aws certified machinelearning specialty. online.

[13] Brian Foote and Joseph Yoder. Big ball of mud. *Pattern languages of program design*, 4:654–692, 1997.

[14] Pete Chapman Ncr, Julian Clinton, Randy Kerber Ncr, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. Crisp-dm 1.0. 1999.

[15] Joao Carreira, Pedro Fonseca, Alexey Tumanov, Andrew Zhang, and Randy Katz. A case for serverless machine learning. In *Workshop on Systems for ML and Open Source Software at NeurIPS*, volume 2018, 2018.

[16] Jiawei Jiang, Shaoduo Gan, Yue Liu, Fanlin Wang, Gustavo Alonso, Ana Klimovic, Ankit Singla, Wentao Wu, and Ce Zhang. Towards demystifying serverless machine learning training. *arXiv preprint arXiv:2105.07806*, 2021.

[17] Hao Wang, Di Niu, and Baochun Li. Distributed machine learning with a serverless architecture. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 1288–1296. IEEE, 2019.

[18] Ioana Baldini, Paul Castro, Kerry Chang, Perry Cheng, Stephen Fink, Vatche Ishakian, Nick Mitchell, Vinod Muthusamy, Rodric Rabbah, Aleksander Slominski, et al. Serverless computing: Current trends and open problems. In *Research Advances in Cloud Computing*, pages 1–20. Springer, 2017.

[19] Tim Wagner. Getting started with aws lambda. re: invent 2014, 2014.

[20] google cloud platform. *cloud functions | google cloud*.

[21] microsoft azure. *Azure functions | serverless compute*.

[22] apache. openwhisk. `https://github.com/apache/openwhisk`, 2021.

[23] Joao Carreira, Pedro Fonseca, Alexey Tumanov, Andrew Zhang, and Randy Katz. Cirrus: A serverless framework for end-to-end ml workflows. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 13–24, 2019.

[24] AWS. *Cloud object storage*.

[25] AWS. *Amazon Sagemaker Documentation*.

[26] Christian Williams Bardia Nikpourian Ryan King Sireesha Muppala, Shelbee Eigenbrode. Machine learning lens.

[27] sagemaker team. *sagemaker feature store*.

[28] Rudolf Ferenc, Tamás Viszkok, Tamás Aladics, Judit Jász, and Péter Hegedűs. Deep-water framework: The swiss army knife of humans working with machine learning models. *SoftwareX*, 12:100551, 2020.

[29] Robert Philipp, Andreas Mladenow, Christine Strauss, and Alexander Völz. Machine learning as a service: Challenges in research and applications. In *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services*, pages 396–406, 2020.

[30] Osama Harfoushi, Dana Hasan, and Ruba Obiedat. Sentiment analysis algorithms through azure machine learning: Analysis and comparison. *Modern Applied Science*, 12(7):49–58, 2018.

[31] Yuncheng Wu, Tien Tuan Anh Dinh, Guoyu Hu, Meihui Zhang, Yeow Meng Chee, and Beng Chin Ooi. Serverless model serving for data science. *arXiv preprint arXiv:2103.02958*, 2021.

Machine Learning and Deep Learning using Edge computing/IoT in intelligent transportation systems - Literature study (Group 5)

Giancarlo Ianni
Universiteit van Amsterdam
1012 WX Amsterdam, Netherlands
gianco.ianni.bermudes@student.uva.nl

Daniel ten Wolde
Universiteit van Amsterdam
1012 WX Amsterdam, Netherlands
daniel.ten.wolde@student.uva.nl

Salmane Dazine
Universiteit van Amsterdam
1012 WX Amsterdam, Netherlands
salmane.dazine@student.uva.nl

Jim Stam
Universiteit van Amsterdam
1012 WX Amsterdam, Netherlands
jim.stam@student.uva.nl

Abstract

The paper tries to give an answer to how machine learning (ML) and deep learning (DL) are used for Internet of Things (IoT) and edge computing in intelligent transportation systems (ITS). Using a Scopus query to find relevant studies, each study has been reviewed. The studies have been categorized into three sections, Autonomous vehicles, Smart traffic and road infrastructure, and Communication systems. For Autonomous vehicles, the focus of the research is on the ML and DL models, with a limited focus on how edge computing or IoT could be utilized for this. The same can be said for smart traffic and road infrastructure. Research into communication systems is more focused on edge-computing. The paper concludes by providing suggestions for future research in the area of model deployment, and distributed storage of data.

1 Introduction

During this literature review, the aim is to provide an overview of the research performed on the use of edge- and IoT-devices with regards to autonomous vehicles. Not only will autonomous vehicles be discussed, but also various transportation and communication systems will be reviewed. Topics such as smart traffic lights systems show great potential for the future to reduce wait times and improve traffic flow. An active research field attempts to find solutions to how these smart systems should optimally operate, current research heavily focuses on the use of ML and DL for most of the topics contained in this literature review. A problem arises however with the amount of data that is needed to be processed to train these ML and DL models. Trying to train the models locally puts a lot of pressure on the, often limited, hardware power available in these systems. On the other hand, sending a large amount of data over a network requires a well-optimized communication protocol, and a low level of latency to provide reliable autonomous transportation.

This paper aims to answer the following question: "What is the state of the art in applying Edge computing and IoT, along with ML/DL in smart transportation systems? In what systems are they used and how?" The research in this area is a combination of many different types of studies, ranging from theory to quantitative.

The topics discussed in various papers that cover smart transportation systems have been split into three categories. The first category is systems related to the autonomous vehicles themselves, sub-

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

jects such as lane-keeping assistance or speed bump detection. The second category is focused on systems that support autonomous vehicles, such as Priority vehicle detection for intelligent traffic lights or intelligent road inspection. The last subject focuses on the communication systems used, for example decreasing the latency for ITS and providing a secure connection.

1.1 Scope

(TITLE((""machine learning*"" OR ""deep learning*"") AND ((""intelligent*"") OR (""autonomous*"")) AND ((""vehicle*"") OR (""*car"") OR (""transportation*""))) AND ALL(""distributed""*) AND (LIMIT-TO (PUBYEAR 2021) OR LIMIT-TO (PUBYEAR 2020) OR LIMIT-TO (PUBYEAR 2019) OR LIMIT-TO (PUBYEAR 2018) OR LIMIT-TO (PUBYEAR 2017)))

(TITLE((""*machine learning*"") OR (""*deep learning*"") AND ((""*iot*"") OR (""*edge*"") OR (""*distributed*"")) AND ((""*intelligent*"") OR (""*autonomous*"")) AND ((""*vehicle*"") OR (""*car*"") OR (""*transportation*""))) AND (LIMIT-TO (PUBYEAR 2021) OR LIMIT-TO (PUBYEAR 2020) OR LIMIT-TO (PUBYEAR 2019) OR LIMIT-TO (PUBYEAR 2018) OR LIMIT-TO (PUBYEAR 2017)))

The scope is defined by the papers found through two Scopus queries. The papers from these queries were then filtered based on some criteria.

- Published after 2016
- Published before mid 2020 - paper has at least 2 citations
- Subject is ground based vehicles

1.2 Machine Learning and Deep Learning

According to Zantalis et al. [1], ML is a concept that is tightly related to artificial intelligence. Through it, systems learn to perform tasks after training them using various algorithms.

There are two types of ML algorithms: supervised and unsupervised. Supervised ML algorithms can use labeled examples to apply what they've learned in the past to fresh data and predict future events. Unsupervised ML techniques, on the other hand, are utilized when the data being trained is neither classed nor labeled. Because they employ both labeled and unlabeled data for training, semi-supervised ML algorithms fall midway between supervised and unsupervised learning. They typically use a small quantity of labeled data and a significant amount of unlabeled data. Reinforcement learning algorithms are a type of ML algorithm that interacts with its surroundings by making actions and detecting faults or rewards.

When it comes to DL, Wu [2] defines it as the next generation of ML algorithms. It employs numerous layers to extract higher-level features (or knowledge) from raw data. The ability of a DL algorithm to handle both supervised and unsupervised learning tasks is one of its most powerful features. Computer vision systems, speech recognition systems, natural language processing systems, audio recognition systems, and bio-informatics systems often make use of DL techniques.

1.3 Edge-computing

As stated by Shaw [3], in the context of Edge-Computing, the term "Edge" refers to a geographical dispersion. Instead of relying on the cloud at one of a dozen data centers to conduct all the work, edge computing is the computation that is done at or near the source of the data.

Rather than relying on a central site that may be far away, it puts computing and data storage closer to the devices where information is gathered. This is done to ensure that real-time data does not suffer from latency difficulties that can degrade the performance of an application. Furthermore, organizations can save money by having the processing done locally, which reduces the amount of data that needs to be processed in a centralized location.

1.4 Internet of Things

Looking at the word "Internet" which simply means a network of interconnected computers, one can deduct that IoT is simply a network of things or physical objects. More accurately, it consists of a network of connected devices that collect, analyze and exchange data with each other. These smart devices are made able to assist someone with a specific activity or learn from a process by merging these linked devices with automated systems. In practice, this can include everything from smart windows to light switches and beyond.

In the context of transportation, the physical devices could be smart vehicles that gather, analyze and exchange data with other vehicles, creating an intelligent system of smart vehicles, as will be discussed later following the work of Zantalis et al. [1].

1.5 Intelligent Transportation Systems

Smart transportation is a term that covers several definitions, a number of them being: route optimization navigation, parking, lights, road anomalies, infrastructure, and accident prevention/detection. Zantalis et al. [1] provide a review of the different applications of ML and IoT in smart transportation systems. It is also tightly related to big data since IoT devices collect and communicate large amounts of data. They have emerged in many fields, notably, the smart transportation field.

With data growing, ML and IoT further enhance the intelligence of transportation systems. To do that, different ML algorithms can be used: Ensemble, Bayesian, Markov, Decision Trees, Clustering, Artificial Neural Networks, DL, Instance-Based, Regression Analysis, Non-probabilistic Linear Classification. Furthermore, the authors state that the significant progress that has already been made in the field of smart transportation with the help of IoT and ML has become apparent, with even more progress projected in the coming years. As the number of IoT devices grows, the diversity and volume of data grow as well, allowing ML to build a plethora of useful applications.

2 Autonomous vehicles

As described by Qayyum et al. [4], the areas of an autonomous vehicle that can benefit from ML can be described by the following categories.

- Perception
- Prediction
- Planning
- Decision Making & Control

To accomplish this, the vehicles rely on different sensors and models. In this section, various papers containing research about autonomous vehicles, and the application of ML/DL, are presented.

2.1 A Survey of Deep Learning Applications to Autonomous Vehicle Control

Kuutti et al. [5] provides an overview of the current state of ML used in the design of autonomous vehicles. It starts with describing the necessity of autonomous vehicles. The main reason being that 90% of all accidents are caused by human error. Furthermore, autonomous vehicles use less fuel, increase productivity and improve traffic flow. Modern cars already contain systems built using ML to aid the driver and improve safety, think of Lane Keeping Assistance and Adaptive Cruise Control. Work on autonomous vehicles has already started in the 1980s, both by academia and car manufacturers. Even non-traditional car manufacturers have researched this topic, such as Google's Waymo [1] driverless car.

The early autonomous vehicle systems heavily relied on sensory data to provide an accurate reading of the vehicle's surroundings. These systems were often hand-tuned, making it a very time-intensive task and difficult to generalize to all different scenarios that may be encountered on the road. The

[1] `https://waymo.com/`

more modern approaches make use of DL, thanks to its adaptability to new situations and thus being better suited for real-world situations. Due to Convolutional Neural Networks (CNNs) performing well with raw camera input, the vehicles of today require fewer sensors and are therefore cheaper.

Vehicle control is divided into two categories, lateral control and longitudinal control. Lateral control is influenced by movement of the steering wheel. This controller is responsible for keeping the vehicle in the right position of the lane or changing lanes. It is also responsible for making collision avoidance maneuvers if that is required. Capturing the surroundings using onboard cameras and feeding this to a neural network is the most popular method nowadays. On the other hand, longitudinal control makes more use of sensory data such as RADAR or LIDAR. This is mainly due to the distance from which the data needs to be read. Vehicles can safely drive next to each other two meters apart in most cases. Driving behind each other requires a greater distance to be safe due to the time needed to brake.

One of the first systems designed for lateral control was Autonomous Land Vehicle in a Neural Network (ALVINN). It was better at sticking to the middle of a lane than humans were. The first mention of reinforcement learning was proposed based on ALVINN and showed a lot of potential. The neural networks in these studies were significantly smaller than they are nowadays, thanks to the increase in computing power. The steering movement was typically determined at every frame. This caused a lot of noise in the output as the situation on the road can change rapidly. Therefore a study proposed the use of a Long-Short Term Memory (LSTM) Neural Network, to also base the steering movement on the result of previous frames. This further improved the smoothness and accuracy. Current research is mostly focused on larger networks and more complex environments.

As for longitudinal control systems, the effectiveness of traditional ML models is limited due to the nature of the problem. On the other hand, work has been performed using neural networks. One described approach used reinforcement learning, with the reward function based on the distance between two vehicles. However, this reward function is likely too simple to work in all possible scenarios. Other research has focused on improving the reward function, such as keeping a two-second gap to the lead vehicle. This reward function is also used in modern cars known as Cooperative Adaptive Cruise Control. Current research is focused on improving the smoothness of driving and further improving the reward function.

After describing the researches that have been conducted in both categories, the paper also states some of the key challenges for the upcoming research. The first major barrier that needs to be overcome is the high amount of computation required. Especially reinforcement, a popular method nowadays, requires both a lot of data and training time to properly train. The paper, unfortunately, fails to mention how Cloud-Based systems can help to tackle this issue, but does mention other solutions which could help. The first one is using reinforcement learning in combination with supervised learning, this could significantly reduce the training time. However, the huge amount of data required to properly build a reliable system is then still an open problem.

Further increasing the amount of data to improve the model is also no simple task since there is the risk of overfitting on the training data. Typically the data set is very unbalanced, with a large portion of the data being typical driving conditions. However, the more rare driving scenarios are critical for the safety and reliability of the system.

The paper suggests the use of high-performance computing hardware on-board. However, this will require a large amount of energy, driving up the cost. An alternative could be to only record rare driving scenarios and sending these to a central location. In this central location, all the data can be processed and be used for further improving the model. The paper never mentions the use of Cloud-Based systems, however, they could prove to be a vital part to, partially, solving this problem.

Other challenges that are mentioned in the paper are selecting the correct architecture of a neural network and defining the correct goal for a neural network. The current method for finding the correct architecture involves a lot of trial and error, making the process slow and tedious. Defining the right goal is often a difficult task given the complexity that is autonomous driving. The reward should be defined in such a way that the model adheres to a safe and efficient driving style, and does not exploit the reward function in an unexpected fashion. Lastly, some of the most important challenges involve safety, something which will be further touched upon later in this paper. The ability for a model to generalize well to any situation that can be found on the road is critical, therefore overfitting on the training data should be avoided. There also exists the challenge of

validating the results obtained. The most popular method of testing is using a simulation, however, it is always possible that the simulation is different and can contain errors. Real-world field tests are required to additionally verify the results obtained from the simulation.

2.2 Intelligent Vehicle Automatic Lane Changing Lateral Control Method based on Deep Learning

Jianglin [6] claims that the steering wheel control accuracy of current automatic lane changing vehicles is insufficient, and proposes a new method based on the use of DL. This paper contains uncited claims and suggests research has been conducted in areas without providing references, therefore we suggest to take claims made by this paper with a grain of salt. Jianglin suggests that there are two parts to autonomous vehicles, being able to follow another car at a proper distance, and being able to change the direction of the car using the steering wheel. This last element is being referred to as lateral movement.

The method used by Jianglin makes use of multiple millimeter-wave radars to detect obstacles, which is able to detect objects from 100 meters away. Other sensors have been considered, however due to its low cost and strong stability it is considered the best option for this study.

As for the lateral controller, Jianglin makes use of fuzzy logic in combination with a sliding mode variable to let the controller make decisions. A fuzzy controller, as it is called, creates control rules through a large amount of data and fuzzy logic. "Fuzzy logic attempts to solve problems with an open, imprecise spectrum of data that makes it possible to obtain an array of accurate conclusions."[2] This is likely a smart decision, as with steering movement there can be multiple conclusions with a correct outcome, as opposed to there only being one solution.

An important part to automating lateral movement is the ability to recognize the lane that is currently being driven in. Jianglin grayscales the image after which the positions of the highest gray values (the white lines) are taken.

The experiments are conducted in Simulink / CarSim to further verify the lateral motion algorithms. Two different network types are tested with different sets of parameters, namely CNN and LSTM. The results are presented in an unclear manner, making it difficult to judge which network type is the best performer. The paper concludes that: "It can be seen that the intelligent vehicle automatic lane changing lateral control method based on deep learning designed in this paper has higher control accuracy than traditional methods." From the single plot provided in the paper it is difficult to validate this conclusion as not enough information is provided. The lack of a discussion section does not help the author improve the strength of his claim. It is possible that the models used in the paper have the potential to outperform traditional methods, however this paper does not prove such claims conclusively.

2.3 Speed bump detection

Apart from methods to connect vehicles, recent research has also focused on methods to apply ML and DL on vehicles themselves. Dewangan and Sahu [7] propose a speed bump detection model using a Raspberry Pi, which acts as the IoT device. The authors create a test setup of a road and several models of speed bumps. Images of speed bumps are preprocessed and augmented for training with a neutral network. The model is trained on a cluster. A novel approach is taken to detect the distance between the speed bump and the vehicle. The camera module of the Raspberry Pi is used for the detection of the speed bumps. It draws a bounding box around detections. The change in the number of pixels between the corners of the boxes is used to determine the distance between the vehicle and the object. This is less computationally expensive than previously proposed methods. When the vehicle gets within a certain distance of the speed bump, it is automatically slowed down. In the paper, this is a small prototype vehicle. The proposed method has a higher accuracy score than state-of-the-art methods. The paper uses a simulated setup to test their model. Due to the difficulty in getting a real-life autonomous vehicle, this is a recurring theme. However, it could be questioned whether the model would produce similar results when on a real-life vehicle. Especially since a

[2] `https://www.investopedia.com/terms/f/fuzzy-logic.asp#:~:text=Fuzzy%20Logic%20is%20an%20approach,an%20array%20of%20accurate%20conclusions.`

Raspberry Pi is not the most powerful computational device. Furthermore, the question is whether training on a subset of speed bumps is enough to generalize to all speed bumps across the world.

2.4 Testing autonomous vehicles

Tuncali et al. [8] propose a requirements-driven testing method for autonomous cars. Due to the complexity of these vehicles, and the potential risks that come with them, a proper testing method is of critical importance for the future of autonomous vehicles. Previous research has been conducted on this topic, but so far, no universally agreed-upon method has been figured out. The team provides a framework that provides simulations on which adversarial testing for autonomous vehicles can be performed. The team describes five requirements using predicate logic which should at all times be adhered to, this should ensure safe use of autonomous vehicles. These requirements are:

- The vehicle should not collide with an object.
- Sensor s should detect visible objects within t_1 time units.
- Localization errors should not be too large for too long.
- A sensor-related fault should not lead to a system-level fault.
- The vehicle should not do excessive braking unnecessarily or too often.

Some scenarios are discussed in which the system is tested, though the team does mention that not every possible scenario can be tested, due to the sheer number of variables present in such a system.

2.5 Conclusion

In this section, papers related to various areas of autonomous vehicles have been discussed. Kuutti et al. [5] provided an excellent overview to describe the state of DL applications for the control of autonomous vehicles. Current research makes extensive use of DL, which provides good results and can generalize well to a wide variety of environments. The DL approach does not come without drawbacks, a major one being the computation required. The paper discussed an on-board solution, though that would increase the cost and energy consumption. An alternative could be sending the data to a central location.

An important aspect of autonomous vehicles is the safety aspect, perhaps even more so than manually controlled vehicles. Properly testing a given model is challenging due to the sheer number of possible scenarios that could occur. Therefore Tuncali et al. [8] propose a requirements-driven testing method. By defining five requirements using predicate logic, it could become more manageable to test future systems.

The majority of research makes use of simulations to test their implementation since this is cheaper and requires less setup time. For example, Jianglin [6] makes use of a simulation program called Simulink in combination with CarSim to test his implementation of an improved lateral controller. This controller can change lanes more smoothly than the traditional methods could. An alternative to simulation is building a small-scale implementation like Dewangan and Sahu [7] have done to build a speed bump detection model. Their model, using a Raspberry Pi, can detect speed bumps with higher accuracy than the state-of-the-art methods, though this was only on a small-scale prototype.

3 Smart traffic and road infrastructure

For autonomous vehicles to function correctly, one must realise that the road infrastructure must also grow along with the advances in self-driving. This is another area where a great deal of research is being performed.

In this section different areas pertaining to smart traffic and road infrastructure will be discussed.

3.1 Machine learning-based traffic prediction models for ITS: A Review

As a complement to autonomous vehicles in an ITS, traffic flow prediction becomes important in order to have a full end-to-end reliable and robust ITS. And, as part of the non-parametric methods of prediction, this review describes several ML methods for this end.

Boukerche et al. [9] categorize these methods in several sub-classes, such as a regression model or kernel-based model and explain the benefits and weaknesses in each when applied for the specific scenario they were created for.

3.1.1 Machine learning-based models

As part of this set of models, the following are explained:

- Regression model: These are considered as typical parametric prediction methods and are used for traffic prediction tasks because they are easily implemented and suited for traffic prediction tasks on a simple traffic network. They study the relationship between the dependent and independent variables through the use of a curve to fit the data set.
- Example-based models: These solve the prediction task by comparing the similarity between the input sequence and the historical data samples, it then uses the found samples to make the final prediction, for example, the k-Nearest Neighbors (KNN) model.
- Kernel-based models: These use a kernel function to map the input data into high-order vector space where the prediction tasks are easy to solve, for example, Support Vector Machine (SVM) and Radial basis Function (RBF) model.

3.1.2 Neural network-based models

These models belong in the category of non-parametric prediction methods which means that the parameters of the model do not need to be pre-set. Instead, they are obtained by learning historical data.

- Feed Forward Neural Network (FFNN): In this approach, the connection between nodes does not form a cycle. It can be used for traffic speed prediction by utilizing the spatial-correlated detector record.
- Recurrent Neural Networks (RNN): In this type of network, neurons can accept the output from the neurons in the previous layer in addition to the neurons in the same layer, which gives the network a so-called "short-term memory".
- Convolutional Neural Networks (CNN): This model, when used for traffic prediction has excellent features describing the spatial interactions among road segments within a big traffic network. It consists of at least one convolutional layer and a fully connected layer after the convolutional layer, and may also have at least one pooling layer.

Although research on ITS has been shifted from statistical traditional models and has been moving more and more towards ML techniques. Not all of these existing techniques will be applicable in ITS. However, DL techniques have proved their efficiency in predicting traffic flow thanks to their ability to handle nonlinear data. They have been originally designed due to a growing need for real-time traffic prediction. Unlike Deep Believe Network and AutoEncoder models, DL methods such as LSTM, aggregations of LSTM, and CNN models have been given more attention and have been widely explored. Compared to a single model, the hybrid structure is more accurate. However, it is difficult to conclude that any method is better than other methods in any particular situation, because different traffic data is designed for different proposals, and the prediction results obtained are based on the collected space-temporal traffic data sets.

3.2 Priority Vehicles Detection Based on Deep Learning for Intelligent Traffic Lights

In the study proposed by Barbosa et al. [10], some of the problems that traffic congestion brings with it are considered, such as the economic, through delays in the delivery of goods and fuel consumption. Health problems are considered as well. The authors state that physical problems can occur when drivers sit in the same position inside a car for a long time, along with breathing in the gasses emitted by vehicles. The approach of vehicle detection is a helpful way to realize the benefits of an ITS. With an ITS, the traffic should decrease due to the more accurate routes for a specific destination. As a result, the amount of time spent in the vehicle is decreased, decreasing also the possibility of dealing with traffic stress. Furthermore, by decreasing the time spent in a vehicle, people can have a more healthy life. This research paper proposes a vehicle detection system consisting of:

- A prediction model named PVIDNet (Priority Vehicle Image Detection Network)
- A design strategy to decrease the model execution time
- A traffic control algorithm
- A vehicles database

The proposed system is able to recognize cars and categorize them under predefined categories: Ambulances, Fire Trucks, Police cars, Buses, and regular cars. Once the vehicles are categorized by PVIDNet, this output becomes the input for the traffic control algorithm which, through the use of a priority table, assigns a priority to each vehicle, and this priority determines how to manage the green and red lights in streets, working with connected traffic lights.

As for weaknesses in this research it can be noted that, in the database used for the prediction model, the vehicle's perspectives considered were frontal and lateral, leaving behind the backside of the vehicles which could make it difficult for the model to identify a vehicle from the back.

It can also be mentioned that this work is tested through the use of SUMO, an Open Source road traffic simulator, in a simple four-leg road intersection. This does not consider the many different types of intersections that exist in real traffic, and it only works for that specific type. Working with intelligent traffic lights with computation units may be required for more complex traffic scenarios.

3.3 Deep Learning Based Motion Planning For Autonomous Vehicle

In the study of autonomous vehicles, it is essential to consider motion planning. Zhengwei et al. [11] propose a DL approach to it by making use of the already discussed LSTM. The proposed model consists of three parts: the Convolutional Long-short Term Memory (Conv-LSTM) used to extract hidden features through sequential image data, a 3D Convolutional Neural Network (3D-CNN) used to extract the spatio-temporal information from the multi-frame feature information, and then, the two neural networks are used to construct a control model for autonomous vehicle steering angle, demonstrating like this that the proposed method can generate accurate and reliable visual motion planning results. The network processing takes as input multi-frame picture segments into the system, which later passes the spatio-temporal LSTM network.

The proposed design is tested with the Keras software library to test the network, and as a dataset, it uses 80GB raw image data and vehicle actual state data. This data is recollected from sensors respectively. The test phase is made through simulations instead of actual vehicles, but considering the dataset, it is safe to assume that is close to reality. This data is mostly generated by cameras and sensors in the vehicle, and the prediction is made at an edge level which shows that the solution can work as a part of an intelligent transportation system. Possibly alongside traffic flow prediction tools, like the ones described in the previous section.

3.4 Intelligent Road Inspection with Advanced ML; Hybrid Prediction Models for Smart Mobility and Transportation Maintenance Systems

Karballaeezadeh et al. [12] proposes a variety of ML methods to assess the Pavement Condition Index (PCI). This PCI method is traditionally performed by humans, using a variety of tests. However, this method is deemed dangerous by engineers since it requires them to be on-site to perform these tests. Therefore the team proposes an automated variation of the test, to reduce the risk and the potential measuring error that is always present with humans performing tests. The related work introduces different ways to measure the PCI, based on the pavement quality, age, or deflection. Pavement sections were first analyzed using the traditional PCI method, which would form the input of the ML models. The first neural network discussed is Multilayer perceptron, which is a somewhat vague term for the general feedforward neural network. The paper lacks detail in this section on how the neural network was precisely constructed. It does well to mention the optimization algorithms used to reduce the error function of the network. The second neural network mentioned is the Radial Basis Function Neural Network. This section is very detailed, precisely describing the structure of the neural network. The optimization functions used are explained in quite some detail, making the explanation of this neural network substantially better than the last one. Finally, the paper describes a method, Committee Machine Intelligent System (CMIS), which combines the result of different models into one. This can help to create even more accurate results than any of the models separately were able to achieve. The results are presented per neural network and include both the train

and test data. For clarity's sake, it would have probably been better to only publish the test data, as that is the most relevant metric for real-world scenarios. The paper contains some spelling errors which could have easily been avoided. The paper lacks a discussion on the achieved results, something that would have provided users a deeper understanding of the results, possibly helping them in future research.

3.5 Intelligent Transportation and Control Systems Using Data Mining and Machine Learning Techniques: A Comprehensive Study

The study performed by Alsrehin et al. provides an overview of works done on the use of machine learning techniques to improve traffic management. Due to an ever-increasing number of vehicles, and the fact that expanding existing road systems is difficult, roads are getting more congested. This not only causes commuters to be stuck in traffic longer, potentially missing important meetings, it also causes issues such as pollution and noise. Therefore it is in the interest of many to create a more accurate traffic management system. Some of the solutions proposed to make use of IoT, such as adaptive traffic signals. Solving the congestion problem completely is unfeasible, however, the goal should be to minimize it. The paper has been separated on a per-issue basis, starting with a description of how to develop a smart traffic management system. The next sections include how to predict traffic parameters, detecting traffic-related objects, traffic planning, identifying traffic patterns, and smart traffic lights.

The development of smart traffic management systems contains several steps, which appear to be fairly similar to other ML-related systems. First is the collection of data, think of images and videos. This raw data then needs to be processed, after which an analysis can be performed. The next step mentioned is storage, something which we believe to be relevant for all stages. Cloud storage could provide a cost-effective storage method for large amounts of data. The last steps are communication, maintenance, and archiving. These steps are fairly standard when it comes to the development of any system. Communication does seem like a useful step since other researchers can also make use of your data to develop their systems further.

To predict traffic parameters an often-used method was to use past traffic data and try to predict the traffic flow for a short period. This can for example be done using an Artificial Neural Network, the more sophisticated methods make use of DL. These predictions are often done using two different approaches. Model-driven requires the researchers to build a simulation in which the traffic behavior is simulated. The data-driven uses real data collected at an earlier stage. The paper, unfortunately, does not mention which is these methods are favorable in which situation.

The paper follows with a section about the detection of traffic-related objects. One of the systems mentioned uses a Pixel Differential Feature to recognize pedestrians, however, a huge amount of research has been done on this topic. The paper does mention some ways to improve currently existing models, such as using more data and factors. This could potentially be interesting however, one should also be careful not to overfit on the training data.

Improving trip planning is another way to optimize road usage and minimize road congestion. Several studies are mentioned using a variety of ML algorithms for routing a trip. Each method comes with its strength and weaknesses and the paper mentions that more models should be explored to get a better overview. Also, more detailed information could help to further improve the models, however, as long as vehicles are driving, there is a risk of human error in the system.

Identifying traffic patterns, be it on a macro level in the number of vehicles at a certain point in time or at a micro-level how vehicles move and behave in traffic, can help to reduce the traffic condition. Several concepts are discussed, however, the works discussed all suggest that future work should focus on improving the simulation systems for traffic patterns. This should be done before more in-depth research can be performed in this area.

Reducing the amount of time spent waiting for a traffic light can be an important factor in reducing the mood for some drivers. Some of the algorithms discussed make use of image data to detect the number of vehicles that are waiting in line. The system then determines the priority one traffic light should receive over another to reduce the overall waiting time. Performing this in real-time is a topic that should be researched further.

To conclude, this paper provides a very in-depth overview of all systems related to smart traffic management. It acts as a good starting point for future work, as it highlights on a per-issue basis what the future work should include.

3.6 An Intelligent Machine Learning-Based Real-Time Public Transport System

Skhosana et al. [13] presents a system aimed at commuters to provide more accurate information on bus arrival times. It provides real-time information on bus location and commuter behavior. It does this with a system designed on Firebase, a Backend-as-a-Service (BaaS), which allows the data to be shared across multiple applications, as well as provide some computation infrastructures. The system design consisted of multiple parts. First off, the team used the FireBase Realtime Database [3], which has the ability to, as previously mentioned, synchronize data across multiple applications. The database itself is a non-structured query language, which has extra flexibility over a more standard structured query language, which is more static in its nature. The team has developed a mobile application for the commuter and bus driver. After the bus driver signs up in the app, he is required to fill in a bus code. When this code is filled in, the bus becomes available on the map for commuters and the driver can start his route. The bus manager then also has an active overview of the buses currently in use. For commuters, the app consists of a map, showing the location of all buses in real-time. In addition, the bus line schedules are listed in the app. These timetables are uploaded by the bus manager to the Cloud Storage for Firebase. When the user looks up the schedule for a specific bus line, the PDF file is fetched from the storage and downloaded on the user's local machine, minimizing the initial data stored on local machines. For the bus manager, a specific online dashboard is created, showing information on all buses currently in transit, all buses in the fleet, and information on the bus drivers.

In addition to these systems, data is collected by the mobile application to track the daily ridership. For bus drivers, it is tracked how accurate the predicted arrival time was to the actual arriving time. And for commuters, it is tracked when busy traveling periods are. The team used a data set from the Chicago Data Portal, which tracked the total daily ridership on a per-route basis since 2001, to train an initial adaptive ML model. This model proofed to be fairly good at being able to predict busy periods. This is convenient for a bus manager since he will no longer have to stick to a static schedule, but rather use a dynamic one.

The paper mentions a possible limitation to the system being the amount of battery used by the mobile application. Since it is constantly tracking the GPS location with fairly great accuracy, it is draining the battery significantly. The GPS draining the battery is a known topic and is actively being researched, such as by Uber [14].

3.7 Dynamic toll pricing

Shukla et al. [15] propose a system called DwaRa. The system uses blockchain to connect intelligent toll gates and smart vehicles and allow government authorities to dynamically set prices and offer easy payment. The proposed solution is to counter existing problems with electronic toll collection, where sensors are used, which are vulnerable to jamming and other security attacks. The proposed method uses Road-Side Units (RSU's), and sensors to collect information about traffic flow. Using Zigbee and GPRS protocols, used also for smart home technology and telecommunications respectively, the data is transferred to the analytics layer. This information is compared to the current weather and compared to historical weather and traffic flow data using a Spatially Induced Long-Short-Term Memory (SI-LSTM) algorithm. The output of the model is a dynamic price which is presented as a Smart Contract, using blockchain technology, to the vehicle owner and the governing authority. If enough funds are available in the wallet of the vehicle owner, the toll is paid automatically. Blockchain technology allows for this distributed payment system. The authors are very detailed in their description of the proposed algorithms, however, a substantial amount of abbreviations are introduced in the paper which can make it difficult to understand the full depth of it. The simulations to evaluate the results however are very extensive. Even though the methods cannot be tested in the real world, the efficiency and scalability are considered by the authors. Other studies in this review do not always consider the scalability of their solution, but especially with

[3] `https://firebase.google.com/`

smart transportation this is important as the systems will continue to grow. If the authors had also considered how to deploy their ML model, the study would have been more well-rounded.

3.8 Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges

Miglani and Kumar [16] provide an overview of the different challenges related to traffic flow prediction. Traffic flow is the optimization of the movement of traffic across roads, resulting in less congestion. They describe a technology called Cognitive Internet of Things (CIoT), a technique to move the computing part from dedicated computing clusters to the IoT devices, and state that this has become increasingly popular. As previously mentioned in this review, the authors state that ITS can help combat traffic flow problems, and lower the number of emissions due to traffic. The survey attempts to set itself apart by focusing on the DL techniques in traffic flow prediction. The paper first gives an overview of autonomous vehicles and traffic flow prediction, then provides an overview of deep learning. The final section is about methods to predict traffic, ending with open research issues and a conclusion.

At the beginning of the paper, the authors collect several figures which provide an overview of system architectures used by an ITS. This is novel, as research papers reviewed so far have often focused solely on a small part of the system. It is helpful to have the architectures shown in one place. They also describe the sensors available on an autonomous vehicle, such as RADAR, infrared and acoustic sensors.

The authors then continue to provide an overview of different deep learning techniques and architectures. Then, studies are reviewed which describe how these algorithms can be used in traffic flow prediction.

Several of the described technologies have already been discussed in this review or will be discussed. These are the edge-computing base-monitoring stations, the vehicle to vehicle communication, and the different layers in which predictions happen. Also, several of the DL algorithms have also been described.

The survey is very comprehensive. The authors keep the article readable by providing all information in text, but also in table and Figure format. However, the amount of papers discussed is very large which might cause the reader to lose track. Also, the authors introduce the term CIoT but do not further elaborate or refer to studies about this subject.

During the discussion of open research challenges, the authors highlight the need for more research into running DL-based traffic flow predictions in the cloud. They state that this can provide easy and cheap predictions. It is highlighted that much of the research is done into improving DL models, but not into deployment.

3.9 Conclusion

This section has covered a wide variety of topics relating to applying ML and DL in different IoT systems in smart traffic and road infrastructure.

One of the main issues with the research area is that there is a wide variety of topics and systems. However for intelligent transportation to realize its full potential, all the systems need to work together. Each paper proposes a variety of ways for data collection, for example through sensors or RSUs. Technologies like Zigbee, also used for smart home systems, or GPRS protocols, used for telephone communications, are used. Most papers briefly discuss the implementation details of their project, but mainly seem to focus on the technical aspects of the models. A popular model which many papers use is LSTM. It is widely used because it is good at taking time-sensitive data into account. The proposed DraWa system describes a blockchain solution that is a distributed way of storing the payments made at electronic toll gates. The paper about real-time public transport systems also proposes a system that uses Firebase as a backend. This backend can share data across multiple applications, allowing for a distributed way of storing the data. Other papers that focus on proposing models do not consider the aspect of deploying their models in a live environment.

The main drawback of the studies is the lack of attention for the deployment of the proposed models. An observed open research challenge is distributing the data across all of the smart connected

devices, how to store all this collected data, and making it possible to create live updates for models based on newly acquired data.

4 Communication systems

In this section different approaches on how the communication is managed in ITS are described. In some researches, in order to have more computation, the communication is offloaded and ML/DL techniques are applied to predict transmission channels. In others, it is visible how all the components of the ITS should be interconnected, and finally, the security of the communication between the components of an ITS is covered.

4.1 Deep Learning-Based Channel Prediction for Edge Computing Networks Toward Intelligent Connected Vehicle

The development in Intelligent Connected Vehicles (ICVs) is directly proportional to the needs of computation required by some of the many applications that have emerged with it. And, to support this intensive computation required in several communication systems (Vehicle to Everything, V2X), the edge computation networks framework is proposed. This approach trades off the wireless transmission to be able to exploit the computation feature of the edge nodes. Because of this, considering the importance of low latency in ICVs, Guangqun et al. [17] propose a model based on LSTM, considering that it is good at analyzing the spatio-temporal correlation in the transmission channel parameters to predict channel parameters in the wireless transmission.

There already exist some models for channel prediction like auto regressive integrated moving average (ARIMA), or support vector machine for regression (SVR) but these have weaknesses including not being able to capture of rapidly changing data sequences for ARIMA, or high algorithmic complexity and the selection of the kernel function parameters for SVR, being a ML method. The proposed model, one based in the DL approach (LSTM), is used to predict the future channel parameters, based on the past and current channels information. Then the model, under Rayleigh Fading is validated through simulations to confirm that it surpasses the commonly used models as ARIMA and SVR in its predictions. The paper makes visible the possibility of increasing the computation on the edge to depend less on the cloud and to achieve lower latency, which is an essential factor to consider for autonomous vehicles in general.

4.2 Deep Learning for reliable mobile edge analytics in ITSs

According to Ferdowsi et al. [18], ITSs will be an important component in the smart cities of the future, and, to achieve the true potential of ITSs, it is of great importance to rely on ultra low latency and reliable data analytics solutions that combine, in real time, a heterogeneous mix of data stemming from the ITS network and its environment. Then, considering that the communication and computing latency are high in cloud-centric techniques, it is more suitable for ITSs to work with edge-centric solutions, created specifically for each ITS.

The study proposes an edge analytics architecture for ITSs where the computation is performed at the vehicles and devices part of the system, to gain lower latency and reliability.

To achieve mobile edge analytics in an ITS, the vehicles and transportation system must be equipped with smart sensors that collect and process a heterogeneous set of data on each vehicle, its passengers, and its environment. And this information has to be collected at ultralow latency and in real time, considering that the ITS must support autonomous driving vehicles. To achieve the required high levels of computation by the system, gathering massive amounts of information from each of the many sensors composing the architecture, DL techniques need to be applied to allow the computing to be made at the edge. This method would offload the computing from the cloud and only send results and important information decreasing the load on the transmission, which otherwise would need to transfer too many information resulting in delay and high latency problems.

Although the proposed architecture is promising regarding computation with the help of sensors and DL techniques, it seems we are still are far from such a system, where every device in it counts with a micro processor to fulfill computing tasks, considering that currently, there is a very wide variety of cellphones used, for example. This would represent a problem when trying to use cellphones as

computing devices in an intelligent vehicle. The proposed architecture is not tested nor has it been implemented, which difficult to see its real reliability.

4.3 Vehicle self-diagnosis

DL is used in autonomous vehicles for the self-diagnosis of failing components. Several modules are employed to predict the chance of components in the vehicle failing. Finally, DL is applied to these predictions to calculate the risk of the entire vehicle failing.

Jeong et al. [19] propose a method for self-diagnoses called the "Integrated Self-diagnoses System (ISS)". It consists of three modules. The first module collects and classifies in-vehicle data into subjects, and sorts them for efficient sending of messages. The second module uses the sensor data to diagnose the vehicle components and then diagnoses the entire vehicle based on that output, using the individual components as training data. The training is done using edge computing devices. When training is complete this data is transferred to the main cloud server. The novel part of the paper is that the result of the training and learning is transferred to other vehicles. This is done when new information is learned. It is transferred through a notification service. When the model, called Data Processing Module (ODLM), is updated and different from previous models it is sent to other vehicles from the main cloud server. It is then used to improve the accuracy of future training. The vehicles also communicate with each other, informing vehicles when another vehicle is classified as "dangerous". The performance of the ODLM is compared with LSTM and Deep Belief Networks (DBN). The paper concludes that the ODLM has a 0.05s better latency than DBN and 0.1 seconds better than LSTM. It also concludes that using Edge Computing reduces the overhead of the system.

The conclusions are based on tests, however these tests were not performed on actual vehicles. The system was tested on PCs to simulate the communication. It could be argued that to obtain more reliable results the system should be tested in practice. Furthermore, the paper does not clearly describe how their edge computing system was implemented making it difficult to reproduce the results.

Nine months later, Jeong at al [20] proposed another solution. First, a new model is introduced to improve the speed of the messaging within the vehicle, allowing faster transfer of the sensory data. The module translates the messages between the hardware and software, along with transforming the messages to the same protocol. The main changes are in that the ODLM is replaced by an In-Vehicle Diagnosis Module (In-VDM). It uses two sub-modules to diagnose the vehicle. Again, one module is responsible for diagnosing the separate parts, while the other is responsible for diagnosing the entire vehicle. The first module uses a random-forest algorithm and the second a lightweight neural network. The authors state that using this implementation, all predictions are done locally and only the results are transferred to the cloud. This means the implementation is not reliant on cloud communication. A point of criticism is that while the paper is similar to the previous work, it does not directly compare itself to other methods. Both papers describe a messaging protocol for the transformation of different message formats, but the results of these tests can also not be directly compared due to subtle differences in the graphs used. Furthermore, the abstract mentions the transfer of data between vehicles using Bluetooth but this is not mentioned further in the paper itself.

There is an overlap between the two papers. Both attempt to self-diagnose a vehicle. One relies more heavily on the cloud and edge computing, while the other does all the calculations locally and only transfers the results to the cloud. Both papers do not test their technologies on an actual vehicle though, leading to results that could be called into question. Since both papers are by the same main author, a more direct comparison was expected. However, it seems the authors have refrained from doing this. A more clear description of the cloud and edge computing implementation would also have been desirable, as this could lead to more reproducible results.

The main takeaway is that DL is being used for the self-diagnosis of components of vehicles. Studies seem to attempt to minimize the reliance on the cloud and a server since self-driving vehicles must react quickly to situations, thus low latency is preferred. This can also be garnered from the fact that both papers devote large sections to the messaging protocols used, as this is an integral part of improving latency. This is especially a challenge since messages from components are in different formats, and must be translated before being sent to the cloud.

4.4 Securing Connected & Autonomous Vehicles

With research being done in several areas to apply new communication systems, IoT systems on autonomous vehicles must communicate with the internet and make decisions based on ML/DL algorithms. In this section the security of these systems is covered.

Qayyum et al. [4] provide a comprehensive overview of the security challenges in the application of ML/DL in autonomous vehicles. An overview is provided of the attack surfaces which exist. This is done by examining a large body of papers from different research areas, and providing a novel insight and interpretation. The paper starts with an overview of what an autonomous vehicle is, the different levels of automation and the history of the technology. The different security challenges are then described:

- Application Layer Attacks: Affects the functionality of vehicle applications
- Network Layer Attacks: Distributed attacks to bring down vehicle applications
- System Level Attacks: Attacks on the hardware or software
- Privacy Breaches: Attacks to gather confidential information

Other attacks include sensor attacks, attacks on the perception system, on intrusion detection and certificate revocation. The paper refers to other papers where these attacks have been successfully described.

The paper also describes the ML pipeline in autonomous vehicles, which was discussed early in this study. Another major topic described by the authors as a focus are the adversarial ML threat for autonomous vehicles. An adversarial attack is described as applying small changes to the input of a ML/DL model to compromise the integrity of the output. A real world example would be changing the appearance of a traffic sign to confuse the sensors. Furthermore, several papers are discussed which discuss the possibilities of countering these attacks. In this case, another model could be trained in order to recognize if a traffic sign has been tampered with.

The authors highlight the fact that efficient distributed storage of all this data is an open research question. Can ML/DL models be applied on a global level, when data needs to be accessible close to the source (the vehicle), and can be distributed over different locations.

This is something also highlighted in this study, as there seems to be a smaller amount of research regarding how to distribute the data efficiently over a large network of autonomous vehicles and other IoT devices.

The paper is a large and exhaustive study of the current state-of-the-art to secure connected, autonomous vehicles. It evaluated a large amount of works and provides an overview, while also providing some novel commentary on the state-of-the-art. The authors do not do any experiments of their own, but do provide the reader with all the information required to get an overview of this field. It complements the other papers reviewed in this study, as all the described systems would cease to function without connecting novel ideas with security aspects.

4.5 Conclusion

It is visible that a well interconnected system is an essential part of reliable ITSs because the data can be used over and over in different components of the system, whether to be used in vehicle diagnosis or in traffic prediction. It can also be concluded that the tendency is for most of the computing being done at an edge level, transmitting only results to the cloud, moving away from cloud-centric architectures, to edge-centric architectures. It seems that there are still some aspects to improve in the security of these systems, considering the importance of the data transmission, and how it could potentially compromise the whole system if the proper security measures are not implemented.

5 Discussion

In all areas of research, there seems to be a lack of experiments performed in real-life smart transportation systems or using actual autonomous vehicles. The cloud systems and compute nodes are

often mocked using a network of computers or software programs. This makes the research interesting, but less relevant. Since ITS require a network of different devices working together to achieve their full potential, it is a shame that research is mostly focused on proposing novel models which only serve a small area of research. There are few actual plans for implementation. This can also be seen in the survey papers, which have the largest sections dedicated to ML/DL models, while the smaller sections are about deployment and architecture.

The industry and academic research may be disjoint in this area. As Qayyum et al. [4] state, Google had passed the 10 million miles target of their self-driving car. Miglani and Kumar [16] also provide this as a motivation for their survey paper, highlighting the fact that leading technology companies spend a lot of effort and money designing these systems. The papers discussed in this review however, are focused on making small changes to models in simulations. It can be speculated that Google is already performing tests with creating a network with self-driving cars, using IoT devices, but that research is lagging behind. A study involving actual IoT devices and networked self-driving cars would prove beneficial, but this is an expensive undertaking.

Another observation is that the training of models is often done on a dedicated machine. More research could be done in the area of providing live updates to the models as they exist on the IoT devices, or even training them on these devices.

It seems there is a trend toward using DL more often than ML. Most of the research papers in the state-of-the-art propose DL models. They are more performant on the difficult task that is intelligent transportation.

6 Conclusion

This literature study describes the state-of-the-art in applying Edge computing and IoT, along with ML/DL in smart transportation systems. Three areas of research that use these technologies have been defined.

The first area is autonomous vehicles. IoT is applied with ML/DL to test autonomous vehicles, detect objects, and keep the vehicle correctly situated on the road. Research is focused mainly on doing the computation locally and no tests on real-life vehicles are done.

The second area is smart traffic and road infrastructure. This area is primarily focused on proposing different ML models, deployed on IoT devices, but the deployment of the models is discussed less than model creation. However, some backend systems are proposed.

The final area is communication systems. This area proposes the most solutions for using edge computing, to move away from the cloud-centric approach. It also proposes ways to secure these transmissions, which are required to make ITS possible. Furthermore, it discusses systems which can transfer updates of ML models to autonomous vehicles. However, it can be concluded that to have a well-connected system there are still some points that need to be upgraded, the uniformity of the devices, for instance. Some papers state that all devices in an ITS need to have a sensor to collect data, but there is no real specification as to what type of sensor could work through all the systems.

As stated in section 5 future research should focus on investigating the deployment of the proposed ML models. It has become clear that to enable ITS, large amounts of data need to be collected and processed. Research can be done to investigate whether it is feasible to have IoT devices collecting and using large amounts of data globally. From current research, it seems that using edge computing can help in solving this problem.

References

[1] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetsos, and Dionisis Kandris. A review of machine learning and iot in smart transportation. *Future Internet*, 11(4):94, 2019.

[2] Jun Wu. Ai, machine learning, deep learning explained simply. `https://towardsdatascience.com/ai-machine-learning-deep-learning-explained-simply-7b553da5b960`, July 2019. Accessed: 28-05-2021.

[3] Keith Shaw. What is edge computing and why it matters. https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html, November 2019. Accessed: 28-05-2021.

[4] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward. *IEEE Communications Surveys & Tutorials*, 22(2):998–1026, 2020.

[5] Sampo Kuutti, Richard Bowden, Yaochu Jin, Phil Barber, and Saber Fallah. A survey of deep learning applications to autonomous vehicle control. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):712–733, 2021.

[6] Lu Jianglin. Intelligent vehicle automatic lane changing lateral control method based on deep learning. In *2020 IEEE International Conference on Industrial Application of Artificial Intelligence (IAAI)*, pages 278–283, 2020.

[7] Deepak Kumar Dewangan and Satya Prakash Sahu. Deep learning-based speed bump detection model for intelligent vehicle system using raspberry pi. *IEEE Sensors Journal*, 21(3):3570–3578, 2020.

[8] Cumhur Erkan Tuncali, Georgios Fainekos, Danil Prokhorov, Hisahiro Ito, and James Kapinski. Requirements-driven test generation for autonomous vehicles with machine learning components. *IEEE Transactions on Intelligent Vehicles*, 5(2):265–280, 2020.

[9] Azzedine Boukerche and Jiahao Wang. Machine learning-based traffic prediction models for intelligent transportation systems. *Computer Networks*, 181:107530, 2020.

[10] Rodrigo Carvalho Barbosa, Muhammad Shoaib Ayub, Renata Lopes Rosa, Demóstenes Zegarra Rodríguez, and Lunchakorn Wuttisittikulkij. Lightweight pvidnet: A priority vehicles detection network model based on deep learning for intelligent traffic lights. *Sensors*, 20(21), 2020.

[11] Zhengwei Bai, Baigen Cai, Wei ShangGuan, and Linguo Chai. Deep learning based motion planning for autonomous vehicle using spatiotemporal lstm network. In *2018 Chinese Automation Congress (CAC)*, pages 1610–1614, 2018.

[12] Nader Karballaeezadeh, Farah Zaremotekhases, Shahaboddin Shamshirband, Amir Mosavi, Narjes Nabipour, Peter Csiba, and Annamária R. Várkonyi-Kóczy. Intelligent road inspection with advanced machine learning; hybrid prediction models for smart mobility and transportation maintenance systems. *Energies*, 13(7), 2020.

[13] Menzi Skhosana, Absalom E. Ezugwu, Nadim Rana, and Shafi'i M. Abdulhamid. An intelligent machine learning-based real-time public transport system. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca, editors, *Computational Science and Its Applications – ICCSA 2020*, pages 649–665, Cham, 2020. Springer International Publishing.

[14] Yohan Hartanto and Brian Attwell. Activity/service as a dependency: Rethinking android architecture for the uber driver app, May 2019. Last Accessed: 27-05-2021.

[15] Arpit Shukla, Pronaya Bhattacharya, Sudeep Tanwar, Neeraj Kumar, and Mohsen Guizani. Dwara: A deep learning-based dynamic toll pricing scheme for intelligent transportation systems. *IEEE Transactions on Vehicular Technology*, 69(11):12510–12520, 2020.

[16] Arzoo Miglani and Neeraj Kumar. Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications*, 20:100184, 2019.

[17] Guangqun Liu, Yan Xu, Zongjiang He, Yanyi Rao, Junjuan Xia, and Liseng Fan. Deep learning-based channel prediction for edge computing networks toward intelligent connected vehicles. *IEEE Access*, 7:114487–114495, 2019.

[18] Aidin Ferdowsi, Ursula Challita, and Walid Saad. Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview. *IEEE Vehicular Technology Magazine*, 14(1):62–70, 2019.

[19] YiNa Jeong, SuRak Son, EunHee Jeong, and ByungKwan Lee. An integrated self-diagnosis system for an autonomous vehicle based on an iot gateway and deep learning. *Applied Sciences*, 8(7):1164, 2018.

[20] YiNa Jeong, SuRak Son, and ByungKwan Lee. The lightweight autonomous vehicle self-diagnosis (lavs) using machine learning based on sensors and multi-protocol iot gateway. *Sensors*, 19(11):2534, 2019.

Cloud Security Practices: Literature Review

Chaoran Li
Department of Computer Science
University of Amsterdam
13511440
chaoran.li@student.uva.nl

Chih-Chieh Lin
Department of Computer Science
University of Amsterdam
13501313
chih-chieh.lin@student.uva.nl

Haochen Wang
Department of Computer Science
University of Amsterdam
13500198
haochen.wang@student.uva.nl

Kaixi Ma
Department of Computer Science
University of Amsterdam
12536016
kaixi.ma@student.uva.nl

Abstract

In this article, we discuss three important components of maintaining cloud security; and further, illustrate a practice example from AWS. Plenty of software applications are created using DevOps, while the security aspect shall be taken into consideration as well. This catalyzes the concept of DevSecOps or SecOps. In addition to the process of DevSecOps, cloud security audits are also essential in order to assess cloud services and trace significant events. Moreover, after completion of audits, organizations could be certified by Accredited Registrars. Hence, the international standard created by ISO is necessary for companies to follow. Finally, AWS GovCloud is examined to shed light on whether the security audits and ISO standard are included, and other features in detail in such a practical environment.

Keywords: Security, DevSecOps, Cloud Security, Compliance

1 Introduction

The original idea of cloud computing emerged around the 1950s with mainframe computing. However, the costs for individuals and organizations are unpractical. Then, in the 1970s, the concept of virtual machines was created. With virtualization, different virtual computers can run on the same physical hardware. In the 1990s, telecommunication companies began to provide virtualized private network connections, even though the connections were point-to-point data transmission.[20] With the increase of information system outsourcing services, the development of cloud computing has grown rapidly in recent decades.

There are three models in cloud computing: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). IaaS is at the bottom layer of cloud services and mainly provides some basic resources. Users need to control the bottom layer and implement the logic of using the infrastructure. PaaS offers a software deployment platform that abstracts hardware and software details, which can be scaled well. Developers only need to pay attention to the logic of their own business, not the bottom layer. SaaS means that software development, management, and deployment are all handled by a third party. It is unnecessary for users to care about technical issues and can use the service out of the box.

Cloud computing provides tremendous benefits to business operations. Customer service providers can share software and hardware resources to computer terminals and other devices so that it is unnecessary for business organizations to design and deploy the environment by themselves. Cloud computing service can also allow enterprise organizations to access applications and related data

without the restriction of location and physical environment, which not only saves costs for the deployment but also present a convenient method for business collaboration. If more computing resources are needed, customers can apply and pay on demand without worrying about insufficient cloud computing resources.

Since cloud computing can provide numerous benefits and much convenience, there is no doubt that it is rapidly emerging and widely accepted worldwide. However, users may not know where the data is stored and worry about data privacy issues. As part of computer security, cloud security should also be considered. To manage the cloud security issues, Ramgovind et al. mentioned that privacy must be designed within the cloud from the beginning, and the service providers need to provide adequate security measures in the daily operations.[24]

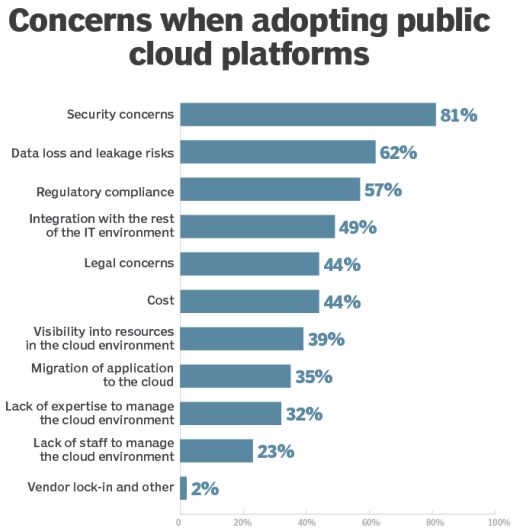


Figure 1: Concerns when adopting public cloud platforms - Security at the top (2019)[13]

In this paper, four practices of cloud security are illustrated and structured as follows: First, we will introduce SecOps. Second, we will describe auditing in cloud computing. Third, ISO compliance, cloud computing-related standards, will be elaborated. Finally, we will discuss in detail the practical design of AWS GovCloud for the United States.

2 SecOps

2.1 Definition

Before illustrating the definition of SecOps, we could first obtain the rules of creating this kind of terminology. For instance, the "DevOps" represents " Developments + Operations". Similarly, the "SecOps" could be seen as " Security + Operations". The corresponded definitions obtained by these simple combination rules are shown in Table 1. Moreover, the exact definition of "DevOps" is illustrated in Table 1 pursuant to the work by Bass, Weber, and Zhu [1].

However, the term SecOps, may not be easily defined in such a way, due to its variety of definitions we found. In this section, we would first list two different definitions of SecOps, followed by a table describing the details of definitions. Afterward, several points of view in related literature would be presented; and finally, the exact definition would be offered.

- 2 different definitions of SecOps
    1. SecOps is short for "security operations", which is the one (SecOps) we described in Table 1.

2. SecOps is interchangeable with DevSecOps and SecDevOps, which could be also seen as the integration of security practices in the DevOps processes. This is correponded to the "DevSecOps" in Table 1.

Table 1: Definitions of SecOps, DevSecOps and DevOps

	Definition	
SecOps	"Security + Operations"	A methodology that IT managers implement to enhance the collaboration between IT security and IT operations teams, helping to achieve the objectives of application and network security without compromising on application performance.
DevSecOps	"Developments + Security + Operations"	DevSecOps is the integration of both DevOps and SecOps, building security into applications during the development processes.
DevOps	"Developments + Operations"	A set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality. [1]

As two different definitions are listed above, there still continue to be a variety of SecOps definitions. Nevertheless, people have discussed "Security DevOps" more, instead of the discussion of "Security Operations". The fact could be demonstrated by the work of Jaatun et al [9]. They presented people's concerns about the security of DevOps, and further provided their metrics for the enhancement of DevOps security in cloud systems.

Furthermore, Gartner, a global research and advisory firm providing information, advice, and tools for leaders in IT, provided their recommendation and analysis of "DevSecOps"[16]. In Figure 2, we could see that they depicted the DevSecOps graphically.

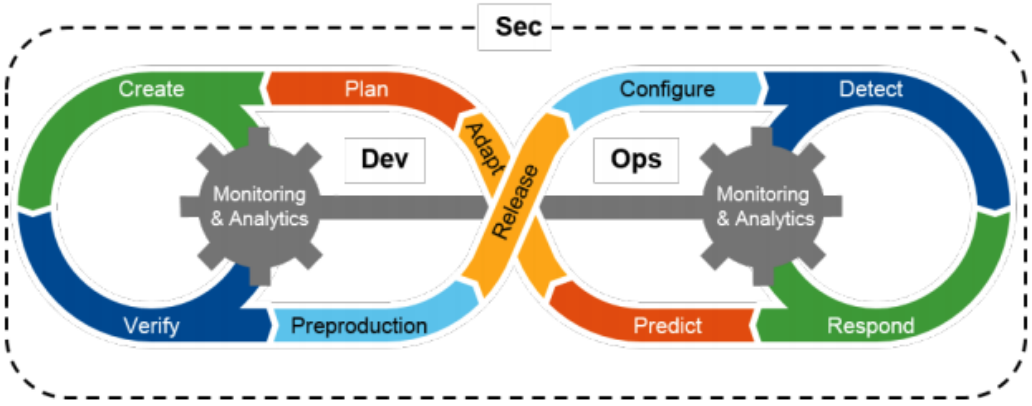


Figure 2: DevSecOps by Gartner (2016)[16]

Moreover, a multivocal literature review of DevSecOps by Myrbakken and Colomo-Palacios[19] interpreted the DevSecOps as *"a concept attempting to create and include modern security practices that can be incorporated in the fast and agile world of DevOps."* Indeed, they regard the DevSecOps as an extension of DevOps. It promotes such extension to the goal of DevOps (enhancing collaboration between developers and operators) by including security experts from the start. Also, Vaishnavi et al.[18] claimed that researchers from industry and academia agree that SecDevOps and DevSecOps imply integration of security practices in the DevOps processes. [2][23]

To conclude this part, the second definition in the list above is the one we are going to delve more into in the following sections since lots of literature and companies had paid their attention to such topics for a period. Therefore, all the SecOps in the following sections shall be seen as DevSecOps defined in the Table 1.

2.2 How SecOps Work

As the DevSecOps graphically depicted by Gartner in Figure 2, it presents the iteration from the developments to the operations with the core of continuous monitoring and analytics. Their objective is to automatically incorporate security controls without reducing DevOps agility, but achieve legal compliance (see section 4) and manage potential risks. However, it was still the superficial concept of DevSecOps. In this section, we would delve into how SecOps (or DevSecOps) works by illustrating its characteristics, workflow, and implementation in Cloud.

2.2.1 DevSecOps Characteristics

Table 2: DevSecOps characteristics [19]

Culture	It is focused on ensuring every member in the organization is aware of and responsible for security. For instance, engineers might report code injection attempts or sales may notice the suspicious emails. Moreover, a set of metrics would be created which each member agrees on and could support and implement.
Automation	The aim of automation is to make sure the security controls are 100% automatic, where the controls could be managed and deployed without manual configuration.
Measurement	In DevSecOps measurements, they not only involve business metrics (such as revenue, performance from DevOps), but also track threats and vulnerabilities throughout the development procedure.
Sharing	The security team shares the methodology, techniques, tools, knowledge to developers and operators. To be more specific, three teams (Developments, Operations, Security) share those to one another so that the security processes would be enhanced.
Shift security to the left	Unlike the traditional software development procedure which place the security at the end of process, DevSecOps move the security to the left which incorporates the security in each part of development process.

Five important characteristics - culture, automation, measurement, sharing, and shift security to the left - are illustrated in Table 2. Those features summarized by Myrbakken and Colomo-Palacios[19] are derived by the principles[1] for DevOps [7] with adding the security from the beginning of the software development process. Apparently, the core feature of DevSecOps is to merge the security controls into the DevOps procedure. Moreover, how such merging works would be illustrated in the next section.

2.2.2 DevSecOps workflow

DevSecOps workflow could be extended to 7 steps cycle from Figure 2, which includes a plan, code, build, test, release, deploy, operate. Dave in [28], emphasized that automating cloud security and management is a key DevSecOps characteristic. Figure 3 presents Dave's point of view. It is

[1]Principles of Culture, Automation, Measurement, and Sharing

important that automatic security controls are embedded in the original DevOps workflow. We further list the main tasks and sub-tasks of automating security mentioned by Dave as follows:

- Embed code analysis, testing in code quality assurance(QA)
- Add operations-centric controls:
  - Logging
  - Event monitoring
  - Configuration, patch, user, privilege management
  - Vulnerability assessment

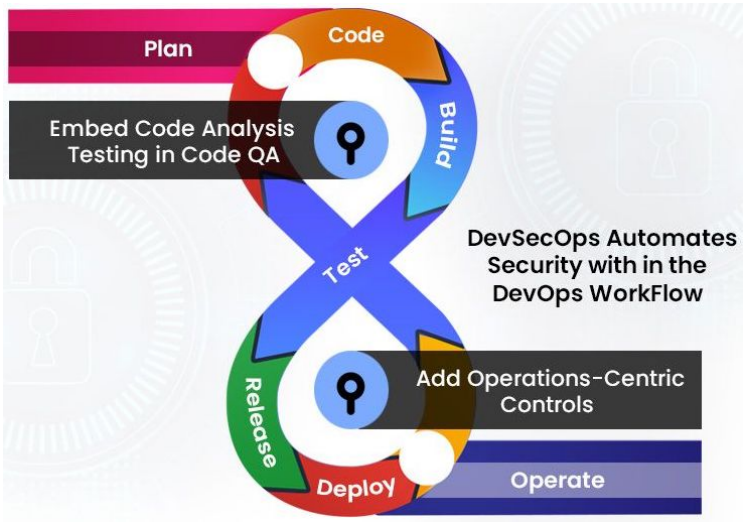


Figure 3: DevSecOps workflow[28][38]

2.2.3 Implement DevSecOps in Cloud

We have realized the characteristics and workflow of DevSecOps from previous sections. However, the implementation of DevSecOps in the cloud which fulfills the characteristics and follows the workflow is also important. We now provide 6 step process of implementation of DevSecOps in cloud[38]:

1. **Code Analysis:** The key to implementing suitable code analysis that matches the DevSecOps objectives is to realize the functionality of each approach and apply them in proper conditions. Nowadays, these functionalities could be delivered by following approaches:
   - **SAST:** Static Application Security Testing is a methodology that incrementally scans source code for vulnerabilities rather than compiles or execute it.
   - **SCA:** Software Composition Analysis detects open source security, license, and operation risks
   - **IAST:** Interactive Application Security Testing secures the runtime applications in real-time(e.g., the app is run by an automated test, interacting with the application functionality).
2. **Automated Testing:** As Dave [28] emphasized, automation is a key to DevSecOps. Automated testing could save time by simplifying the test process with minimum testing scripts and related tools.
3. **Change Management:** Incorporating the developers in the security process makes effective change management. The developers would be aware of associated tools and might find the possible vulnerabilities.
4. **Compliance Monitoring:** Compliance plays an important role in every organization in the IT industries. Here, having the legal compliance and keeping monitoring ease the burden of audit, and also maintain transparency.

5. **Threat Investigation:** Gain knowledge of possible security threats, and then establish methods to detect and respond to those threats.
6. **Personnel Training:** It is also an essential part of every organization. The personnel could gain knowledge of security via courses, training, lectures, or hand-on workshops. Indeed, improving the personnel's skills and background knowledge of security fulfills the "culture" characteristic in section 2.2.1 and would make an organization more successful in managing security issues.

3 Cloud Security Audits

3.1 Definition

There is no doubt that cloud computing is developing rapidly, business can utilize hardware and software at the same time through the Internet without having to deploy software in their physical computers. It can support convenience, reduce cost management, and also significantly improve business efficiency. There are three models in cloud services: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). The cloud auditor is a party that can independently assess cloud services [15], trace and log significant events during the system operation. These events can be used for analysis, verification, and validation of security measures.

Both traditional IT audits and cloud audits have the same concerns, such as protecting information assets, maintaining data integrity, and operating effectively. However, compared with traditional IT auditing, cloud auditing has more challenges. For example, auditors need to obtain sufficient knowledge of cloud computing, familiar with cloud computing terminology[26].

3.2 Challenges

Information security in the cloud presents a series of challenges and we need cloud computing audits to protect information security. In this paper, [26], Ryoo et al. illustrated four major challenges in cloud security audits and emerging approaches.

1. **Transparency** The first main challenge of cloud computing is transparency. A good cloud security audit needs to check whether data is transparent to cloud customers because data and security are managed by a third party, and users do not know who processes the data and where data are stored. The lack of transparency can fail to gain the trust of cloud service customers and increase the risk of information security threats. Transparency is not only essential for customers to protect their data, but also for cloud service providers to establish a safer and more reliable cloud computing environment. The higher the customer's trust in the cloud, the more cloud service providers will pay more attention to cloud security issues, thus pointing out a better direction for the cloud environment.
2. **Encryption** When talking about data security, we cannot ignore encryption technology, because customers care about their personal privacy and are unwilling to disclose the data to anyone. When considering encryption in cloud computing, we need to consider whether to encrypt all data or encrypt it before sending it to the cloud. Both methods have their advantages and disadvantages. Encryption on the user side can provide convenience, but it also brings the risk of abuse of authority by the system administrator. If the entire data is encrypted, the security of the data can be effectively improved, but the decryption process will consume a lot of computing resources. In order to address this situation, we can apply a third party, and also use the homomorphic encryption method to alleviate the computational resource consumption [14].
3. **Colocation** The most prominent benefit of cloud computing is flexibility and mobility that numerous organizations can share the infrastructure. With the development and widespread application of cloud computing, there are multiple hypervisors generated, including VMWare, Oracle Virtual Box, Microsoft Virtual Server, etc. Therefore, it is very crucial for cloud service providers to control the access permission of administrators and protect user's data. In the traditional IT environment, there are a series of standards that can provide auditing. However, in the cloud setting, there is still no clear cloud security standard. Therefore, establishing standards and increasing oversight is essential. In section 4, we have more specific research related to cloud computing standards and ISO compliance.

4. **Scale, Scope, and Complexity** In cloud computing, there are a large number of systems that need to be audited. The increase of IT elements brings scale challenges, and the emergence of new technologies also presents scope challenges. In addition, because the data centers where storing organization's data and information are located in different countries, they are subject to different laws and regulations. Cloud auditors need accounting and cloud computing knowledge, as well as local legal background.

3.3 Applications

3.3.1 TrustCloud Framework

Researchers have designed and implemented some cloud audit frameworks or infrastructures to solve real-world cloud audit problems. Ryan et al. mentioned in the paper [12] that virtualization is another challenge considering cloud computing. It is essential for virtualized layers to identify events on both virtual servers and physical servers, and accountability is required during the process. Accountability in cloud computing is used to protect sensitive data, gain consumer's trust and handle it responsibly.[22]

Instead of focusing on the preventive controls of the trust components, they pay more attention to the detective controls to increase accountability in this paper. In order to address the cloud accountability from several aspects, they present five abstraction layers in the TrustCloud framework, including system layer, data layer, workflow layer, polities, as well as law and regulations. This model can simplify issues in cloud security and make accountability more achievable. File-centric logging, data-centric logging, and auditing data in the software services can be accomplished in the cloud, which can not only prevent external risks but also internal risks from customer service providers.

3.3.2 An Efficient and Secure Dynamic Auditing Protocol

To provide a more effective cloud audit process, an audit service is required to check the data integrity in the cloud. However, some data checking methods can only verify static data and cannot be applied to cloud auditing services since the data is dynamic and updated in real-time in the cloud.

The paper [41] by Yang et al. introduced the research about auditing framework for data storage and data privacy. Then, they also present the auditing protocol to support the dynamic changes in the cloud, which can effectively solve the security issues, reduce computing costs for auditors and also provide efficiency. The cloud auditing protocol should have the following three characteristics: confidentiality of the owner's data, the dynamic operations of data updates in the cloud, as well as batch auditing to support multiple clouds and owners.

The system model of storage auditing protocol involves three parts: owners, cloud servers, and cloud audits. In order to prevent cloud auditors from decrypting the owner's data directly, which could lead to potential privacy risks, they implement a method that allows cloud auditors to check the correctness of authentication. For security dynamic auditing, there are two types of attack to consider: replay attack and forge attack. In response to these problems, the author introduces an index to record the abstract information of the data and modifies the tag generation algorithm to make the server unable to forge data tags. In addition, cloud auditors can combine numerous auditing requests together and utilize batch auditing techniques for multiple owners to improve the performance and efficiency of the system.

3.3.3 Privacy-Preserving Public Auditing

Another article[39] was written by Cong et al. also looks at the cloud auditing system for data storage and a public audit method for data privacy. They use HLA-based technology to verify the data and support the public audits. Compared with the MAC-based solution, the HLA technique can be aggregated and provide the authentications of linear combinations of a single data block. In order to implement the public audits system, they apply integrated homomorphic linear authentication with random masks, so that the third-party auditors cannot view or obtain customer data and ensure data privacy. The audit system also supports batch auditing and dynamic data updates, which significantly improve efficiency for auditors.

4 ISO Compliance

Compared to get a certification of the ISO, more and more companies decide to be ISO Compliance due to time-consuming and high cost. ISO Compliance means a company adhere to ISO standards but does not get the certification. It will also help the company build a security system.

4.1 Introduction

When it comes to the ISO Compliance, we need to introduce two international organizations related to this topic. The first organization is the International Organization for Standardization (ISO), which focuses on publishing new standards in many fields. The International Electrotechnical Commission (IEC) also works on proposing new standards, but it only focuses on the fields of electrical engineering and electronic engineering. The famous standards of building an information security management system (ISMS) is the family of 27000 standards. These standards are published jointly by the ISO and IEC. Nowadays, they have become the most powerful guidelines of information security.

4.2 ISO Standard

In this section, we will show an overview of the related standards in Table- 3 and describe them in detail later.

Table 3: Standards of ISO/IEC

Standard	Content
ISO/IEC 27001	The information security management system aims to provide a method for all types of organizations in establishing, implementing, operating, supervision, reviewing, maintaining and improving the information security management system[31].
ISO/IEC 27002	It is a instructional standard which is used as a reference to help company to select control measures when implementing an information security management system, or as a guidance for companies to select information security control measures[32].
ISO/IEC 27017	This standard is used to provide enhanced control for cloud service customers and cloud service providers to build a security cloud environment. It also clarifies the roles and responsibilities of both the users and the providers to ensure the data safety[33].
ISO/IEC 27018	PII (Personally Identifiable Information) is the most important part in this standard. It provides us a guidance of how can we protect the PII based on the ISO/IEC 27002[34].
ISO/IEC 27032	This standard focus on the Cybersecurity on the Internet. It shows us some issues in this field and proposes some instructions on how to avoid the Cybersecurity problems[35].

As we can see in the table above, these standards work together to help companies build an information security management system.

4.2.1 ISO Standard 27001 27002

Standard 27001 has the title of "Information technology—Security techniques—Information security management systems—Requirements". This standard can help companies of all sizes (from a small company to a worldwide multinational) and all fields (such as government, university, healthcare) implement a system that meets the requirements of being an information security management

system[31]. The requirements in this standard include all steps a company will experience in its life spans, such as the implementing stage, operating stages, and development stage. Including 27001, the ISO 27 K family of standards refer directly to the "Plan-Do-Check-Act" (PDCA cycle) cycle—well known from Deming's classic quality management[6], which can give us a brief understanding of the steps we need to take to build an information security management system by a model.

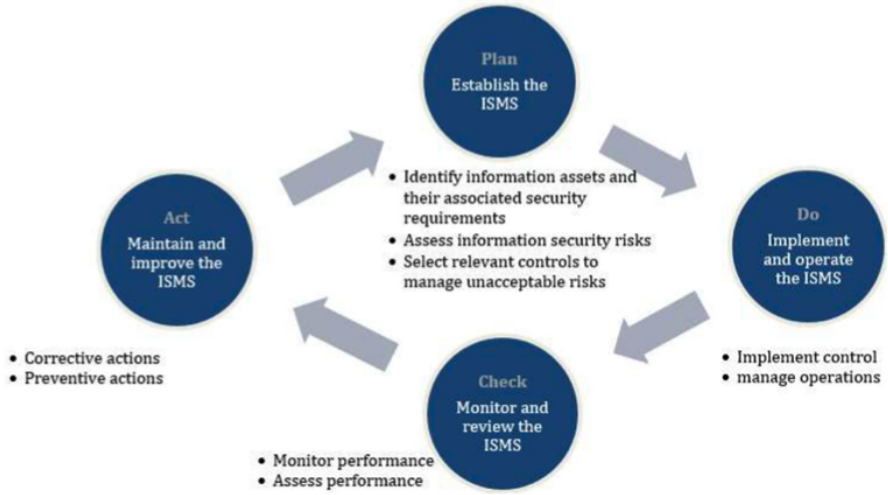


Figure 4: PDCA cycle in ISO 27001[36]

Figure 4 comes from the article written by Najat Tissir[36]. This cycle can work with ISO standards to help company build the system. According to this figure, when we want to build an ISMS, we need to understand the information assets and their security requirements and find out suitable controls to manage risks. Some documents used to guide the developers are necessary for this step. Implementation is the next step in this figure, controls and measures written in documents are implemented, ISMS is managed appropriately. We also need to formulate some plans to encounter risks in this step. After the implementation step, it is important to monitor and review your ISMS system. Take the documents we use before, check if the system is running to conform to the regulation in the first step. It will help organizations find out potential security risks and know more about the system. With the help of our monitoring and review, we can check whether the system has some risks and shortcomings. In the final step, we correct some wrong operations and take some precautions to improve our system. The cycle of steps will help a company continuously improve its ISMS.

The requirements in ISO 27001 are described in detail in another standard: ISO 27002. It describes the necessary steps to identify and assess security risks to determine the requirements for protecting information and information systems. The continuous development of ISO 27002 is based on the content of ISO 27001, which explains the 39 control objectives listed in ISO 27001 in more detail[32]. A total of 134 measures have been allocated to these goals, which are reasonable and described in detail in 27002[6].

4.2.2 ISO Standard 27017 & 27018

These two standards focus on the field of cloud. The ISO standard 27017 is an expansion of the 27001. It not only provides the control sets from ISO 27001 but also proposes 7 new controls about the cloud. It is necessary to clarify the responsibility between both cloud service providers (CSP) and customers[33]. Following are the terms of security responsibilities from 27017 with cloud service providers, some with the customer, and some are shared with both[40].

- Who is responsible for the relationship between the cloud service provider and the cloud customer
  - This relationship need to be clearly laid out, recorded and communicated.
- When the contract is terminated, how to deal with the assets

  - These assets belongs to the customers should be returned or removed.
- How to protect and separate the customers' virtual environment.
- Virtual machine configuration
  - Customers and providers should ensure the virtual machine's configuration and strength to meet the security requirements.
- Cloud environment-related management operations and processes
  - List the responsibilities with the customers on defining, monitoring and documenting the cloud related administrative operations.
- How can customers monitor the cloud activities
- Connection between cloud network environment and virtual environment
  - Addresses building standards and configurations must be consistent. So the virtual network environment is in line with information security policies around the physical network[40].

Another standard pay attention to the PII (Personally Identifiable Information). It gives us some definitions related to PII and guidance on how to protect the PII. It tells us some definitions about the PII Principal, PII Controller, PII Processor which can help companies to clarify the responsibilities in the cloud[34]. What should data processors do to protect the PII is described as follows[40].

- Notice customers if their PII is being used
  - Such as the PII is processing by others or accessed by unauthorized activity.
- Help customers to manage their PII
  - The PII controller should have rights to access, modify and delete their PII.
- Virtual machine configuration
  - Customers and providers should ensure the virtual machine's configuration and strength to meet the security requirements.

Companies can use these standards in conjunction to provide enhanced control for cloud services.

4.2.3 ISO Standard 27032

This standard mainly addresses the problems in the field of cybersecurity. It focuses on attacks caused by malicious and potentially unwanted software, social engineering attacks, information sharing in cybersecurity[36]. What's more, it provides some advice and guidance to deal with the cybersecurity risks and proposes to build a formalized framework to share cybersecurity problems and experiences in dealing with them[35].

4.3 Application

For those large companies, get the ISO certification is worth doing because you can drive the trust of others in your business, provide a competitive advantage to your company, protect brand reputation, facilitate the development of enterprises, and help your company become a multinational early.

However, for those small companies, it is a time-consuming and unaffordable cost. So there are many types of research focus on using these standards to check security by companies. Article from NA Kamaruddin proposed a pre-assessment model for cloud providers to determine if their system is safe[10]. It can assess the cloud security readiness level to judge if the measures used here are safe enough. And it also can help the organization increase the awareness of cloud security.

These standards can increase the development of cloud services. Here we would like to introduce an example about PII data in cloud services from Kemp. Microsoft uses this standard immediately after the publication of the 27018 to protect the services in its cloud. And this standard can be used across the service delivery models and the layers in the main model [11].

Finally, we can also use these standards to investigate the data quality in cloud services[25] and propose solutions related to cloud security in smart business[8].

4.4 Problems in Cloud

There are some areas that ISO does not include in their standard, it may cause some risks in security. It does not contain the protection of the Audit part, which may help attackers delete logs after access the cloud services. The missing part of high-risk environments in these standards can lead to advanced persistent threats and side-channel attacks[4].

5 AWS GovCloud

With cloud computing developing fast, the market size of cloud computing grows over a 30% compound annual growth rate reaching $383.4 billion in 2020. This figure is expected to be updated to $832.1 billion by 2025.[30] Governments around the world have passed various legislation related to the safeguard of sensitive or private data. In this section, we will look insight at one security practice of the famous cloud service provider, AWS GovCloud, especially for the United State.

5.1 Definition

AWS GovCloud literately contains the connotation of two parts, AWS and GovCloud.

AWS stands for Amazon Web Services. Amazon has a long history using a decentralized IT infrastructure. With the impact and promise of cloud computing, Amazon has spent more than thirty years and over six billion dollars building and managing the large-scale, reliable, and efficient IT infrastructure that powered one of the world's largest online cloud service providers. AWS was launched in 2006 to enable other organizations could benefit from Amazon's experience and investment in running a large-scale distributed, transactional IT infrastructure. Today, it serves hundreds of thousands of customers worldwide.[37]

Users purchase AWS to request compute power, storage, and other services in minutes and have the flexibility to choose the development platform or programming model that makes the most sense for the problems to solve. Some well-known cloud services provided by AWS are EC2 and S3 These services have a different purpose and use different databases such as RDS, DynamoDB, and Elastic Cache.[21]

GovCloud means AWS gives state and federal government customers and their partners the flexibility to architect secure cloud solutions. AWS GovCloud has commercial cloud capability across all classification levels making it possible to execute missions.

- Unclassified: non-sensitive data and workloads
- Sensitive: classified sensitive non-government workloads
- Secret: classified sensitive by non-intelligence government organizations
- Top Secret: separated from the public Internet, hosted on-premise at the CIA

AWS GovCloud (US) provides an environment where those customers can run ITAR-compliant applications and provides special endpoints that utilize only FIPS 140-4 encryption.[21] As mentioned previously, AWS GovCloud follows and complies with the government regulations and compliance regimes, which is discussed detailedly in the next Subsection.

5.2 Regulation and Compliance

For all AWS data centers worldwide, they include the ISO 27001 certification[31]. AWS holds the FISMA moderate certification also it operates over the FIPS 140-2 validated hardware.[29] Adhering to the FISMA guidelines AWS rotates keys. Amazon provides reporting processed for security vulnerabilities and penetration testing.[21]

For United States Government, AWS GovCloud compliance features include data safety and access control, with granular control of individual data at the API level. AWS GovCloud (US) complies with the FedRAMP High baseline, Cloud Computing Security Requirements Guide, and other compliance regimes.

These and other security-related regulations bring it into full compliance with a broad range of United State government security and restricted access regulations.[17]

- Federal Risk and Authorization Management Program (FedRAMP)
- Department of Defense Security Requirements Guide (SRG) through level 5
- Department of Justice Criminal Justice Information Service Security Policy
- Defense Federal Acquisition Regulation Supplement (DFARS)
- U.S. International Traffic in Arms Regulations (ITAR)

The USA Air Force's Next Generation GPS runs in AWS GovCloud, and so does the General Services administration's website, which is the central cloud platform used by the federal government. Plus, the Justice Department uses AWS GovCloud both for internet operations and public-facing services.[17] Therefore, AWS GovCloud (US) must follow the regulations and compliance regimes of the American state and federal government to control access to sensitive data and keep authentication of secret workloads.

5.3 Security Features

The same principles of security apply as for other non-cloud systems, but the main differences are the lack of control of the cloud services and the secrecy of how these systems are managed by AWS GovCloud.[5] It takes some measures to protect the security of data and workloads. We discuss three of the most common practice AWS GovCloud (US) has taken.

Besides government strict requirements of regulations and compliance, users must pass the AWS GovCloud screening process. All customers who use AWS GovCloud (US) must either be Government organizations or other approved private entities in Government-related industries such as we mentioned before.[17] Each customer is vetted to ensure they are a United States entity and cannot be prohibited or restricted by the United States government from exporting or providing services. AWS GovCloud no matter US-East or West Region is only operated by employees who are United States citizens on United States soil. It is only accessible to United States entities and root account holders who pass a screening process.

The second feature is the independence of resources. Network, Data, and Virtual Machines in GovCloud are isolated from all other AWS Cloud Regions. AWS GovCloud features a separate identity and access management stack with unique credentials, which only work with the GovCloud region, and comes with a dedicated management console, as well as endpoints that are specific to the GovCloud region.[3] Separate physical resources make sure of enclosure of a working environment. The probability of being attacked significantly decreases.

The third measure is a specific department, AWS Security Hub. It is the department responsible for cloud security. AWS Security Hub gives a comprehensive view of the security posture of the services. These security controls detect when accounts and deployed resources do not align with security best practices defined by AWS security experts. There is a range of powerful security tools, from firewalls and endpoint protection to vulnerability and compliance scanners. Important and common security measures include DDoS protection, brute-force detection, secure HTTPS access using SSL, built-in firewall, multi-factor authentication, private subnet etc.[21]

5.4 Services

Look back to the cloud service itself, AWS GovCloud not only provides services the same as ordinary region version but also has additional unique services mainly focused on security.[27]

- Safeguard sensitive data — shield sensitive unclassified data with server-side encryption in Amazon S3. Store and handle security keys yourself with AWS Key Management Service (AWS KMS).
- Improve cloud visibility — audit access and use of sensitive data with your keys in AWS CloudTrail, operated by US citizens.
- Strengthen identity management — restrict access to sensitive data by time and location, and specify which API calls users can make. GovCloud offers powerful access control features.
- Shield accounts and workloads — apply continuous security monitoring for AWS accounts and workloads using Amazon GuardDuty. Monitor workloads for malicious or unauthorized behavior that may indicate an account compromise.

We also review two commonly used services, EC2 and S3, which we are familiar with. Compared with a regular type of AWS, we can discover what special limitations are set to raise the security level.

Elastic Compute Cloud (EC2) provides resizable computing capacity that users build and host software systems. In AWS GovCloud, we must launch all EC2 instances in a virtual private cloud. EC2 Serial Console is currently disabled in AWS GovCloud (US). The image copy and snapshot copy do not support the origin of another AWS Region. And we can only use the API-only method to conduct the CPU optimization in AWS GovCloud (US).

Simple Storage Service (S3) is storage for the internet, which provides several interfaces for access. AWS GovCloud is a closed storage environment, as we mentioned in **5.3** independency. We cannot copy the data of an S3 bucket in the AWS GovCloud Regions to or from another AWS Region. It is required to use the Amazon Resource Names identifier. The name of the bucket must be unique in AWS GovCloud and not be shared across other standard AWS Regions. S3 transfer acceleration is also disabled.

There are many restrictions and limitations in usages of AWS GovCloud (US), which can be found in the manual documentation. For more intuitively acknowledged, we just show some of them to learn the advanced security protections applying to AWS GovCloud.

6 Discussion

It could be emphasized that the core of DevSecOps is to shift security to the left. Unlike the traditional DevOps which implements the security testing after the step of releasing, the DevSecOps keep security controls through all the process of software development. That is, the security scanning is implemented from the start of the development procedure. An apparent benefit is that the vulnerabilities could be found earlier. Therefore, the security issue could be solved and be managed as soon as possible; and it would decrease the probabilities of bug found by customers. That is why the "security shift to the left" should be considered to merge into the DevOps procedure.

Cloud auditing plays an important role in cloud computing. Based on the literature review, we have discussed several major challenges in cloud auditing and related strategies. We have also studied some technologies of audit frameworks or infrastructure, and the benefits they bring. As cloud computing is still undergoing continuous development and progress, it is very necessary to design a credible auditing mechanism for cloud computing.

When we look at the standards related to information security management systems and cloud security, plenty of organizations have published multiple standards to verify the security of a cloud. However, there is no single standard that can cover all the potential risks. As the BSI C5, it omits six controls compared to other standards. And according to Di Giulio et al. 's attack model, the most frequent risks caused by omissions in C5 are at the cloud level and are internal threats[4]. So we think it is important to combine them to get a higher safety level.

The practice of cloud security for both public and private cloud providers is still facing challenges in implementing the cloud computing model. That is one of the main reasons that AWS GovCloud has only open its business to the United States. The legislation protecting sensitive data and workloads is another obstacle. Laws, regulations, and compliance vary around the world. Therefore, it is impossible for a cloud service provider to deploy the private cloud services globally, like government-oriented business. One current solution is that government cooperates with local cloud service providers to establish private clouds to maintain the security of cloud services.

7 Conclusion

In this review, we first look at the history of cloud computing and security problems related to the cloud nowadays. Afterward, the definition of SecOps is clarified by comparing the concepts of SecOps, DevSecOps, and DevOps; and also by the information gathered from lots of literature. The workflow and characteristics of DevSecOps in this review are used to introduce the implementation of DevSecOps in the cloud. We find that several features are of great importance in implementation. In the Audits section, we introduce the definition of cloud audits and the challenges. The audits are wildly used in the cloud to check if the system is safe enough. We also introduce some standards

about ISMS and cloud security that come from ISO to use as guidance to build a safe system. These standards can not only be used as certification references but also increase the development of cloud services. After that, we describe an example of AWS and GovCloud. It is a real-world example that pays more attention to the security problems in the cloud because of its usage. The security of the cloud attracts an increasing number of attentions , but there is still some problem in this field.

References

[1] Len Bass, Ingo Weber, and Liming Zhu. *DevOps: A software architect's perspective*. Addison-Wesley Professional, 2015.

[2] S Cash et al. "Managed infrastructure with IBM cloud OpenStack services". In: *IBM Journal of Research and Development* 60.2-3 (2016), pp. 6–1.

[3] CloudBasic. *AWS GOVCLOUD (US)*. [EB/OL]. `https://cloudbasic.net/aws/rds/alwayson/govcloud/` Accessed May 26, 2020.

[4] Carlo Di Giulio et al. "Cloud standards in comparison: Are new security frameworks improving cloud security?" In: *2017 IEEE 10th International Conference on Cloud Computing (CLOUD)*. IEEE. 2017, pp. 50–57.

[5] Oscar Diez and Andres SIlva. "Govcloud: Using cloud computing in public organizations". In: *IEEE technology and society magazine* 32.1 (2013), pp. 66–72.

[6] Georg Disterer. "ISO/IEC 27000, 27001 and 27002 for information security management". In: (2013).

[7] Jez Humble and Joanne Molesky. "Why enterprises must adopt devops to enable continuous delivery". In: *Cutter IT Journal* 24.8 (2011), p. 6.

[8] Igor Ivkic et al. "On the cost of cyber security in smart business". In: *2017 12th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE. 2017, pp. 255–260.

[9] Martin Gilje Jaatun, Daniela S Cruzes, and Jesus Luna. "Devops for better software security in the cloud invited paper". In: *Proceedings of the 12th International Conference on Availability, Reliability and Security*. 2017, pp. 1–6.

[10] Nur Ahada Kamaruddin et al. "CLOUD SECURITY PRE-ASSESSMENT MODEL FOR CLOUD SERVICE PROVIDER BASED ON ISO/IEC 27017: 2015 ADDITIONAL CONTROL". In: *Revolution* 2.5 (), pp. 01–17.

[11] Richard Kemp. "ISO 27018 and personal information in the cloud: First year scorecard". In: *Computer Law & Security Review* 31.4 (2015), pp. 553–555.

[12] Ryan K.L. Ko et al. "TrustCloud: A Framework for Accountability and Trust in Cloud Computing". In: *2011 IEEE World Congress on Services*. 2011, pp. 584–588. DOI: `10.1109/SERVICES.2011.91`.

[13] George Lawton. *Use modern cloud security best practices*. Aug. 2019. URL: `https://searchcloudcomputing.techtarget.com/tip/Use-modern-cloud-security-best-practices`.

[14] Jian Li et al. "A simple fully homomorphic encryption scheme available in cloud computing". In: *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*. Vol. 1. IEEE. 2012, pp. 214–217.

[15] Fang Liu et al. "NIST cloud computing reference architecture". In: *NIST special publication* 500.2011 (2011), pp. 1–28.

[16] N MacDonald and I Head. "Devsecops: How to seamlessly integrate security into devops". In: *Gartner, Tech. Rep.* (2016).

[17] Joseph Mahakian et al. *AWS GovCloud Resource and Cost Analysis*. Tech. rep. The MITRE Corporation, 2020.

[18] Vaishnavi Mohan and L Othmane. "SecDevOps: is it a marketing buzzword". In: *Department of Computer Science, Technische Universität Darmstadt, Darmstadt* (2016).

[19] Håvard Myrbakken and Ricardo Colomo-Palacios. "DevSecOps: a multivocal literature review". In: *International Conference on Software Process Improvement and Capability Determination*. Springer. 2017, pp. 17–29.

[20] Maximilliano Destefani Neto. *A brief history of cloud computing*. Mar. 2014. URL: https://www.ibm.com/blogs/cloud-computing/2014/03/18/a-brief-history-of-cloud-computing-3/.

[21] Deepak Panth, Dhananjay Mehta, and Rituparna Shelgaonkar. "A survey on security mechanisms of leading cloud service providers". In: *International Journal of Computer Applications* 98.1 (2014), pp. 34–37.

[22] Nick Papanikolaou, Siani Pearson, and Nick Wainwright1. "Accountability in Cloud computing". In: *BUILDING International Cooperation for Trustworthy ICT* (2011), pp. 1–3. URL: www.bic-trust.eu/files/2013/01/Papanikolaou_AccountabilityInCloudComputing_June2012.pdf.

[23] Akond Ashfaque Ur Rahman and Laurie Williams. "Software security in devops: synthesizing practitioners' perceptions and practices". In: *2016 IEEE/ACM International Workshop on Continuous Software Evolution and Delivery (CSED)*. IEEE. 2016, pp. 70–76.

[24] S Ramgovind, M M Eloff, and E Smith. "The management of security in Cloud computing". In: *2010 Information Security for South Africa*. 2010, pp. 1–7. DOI: 10.1109/ISSA.2010.5588290.

[25] Jonathan Roy, Hebatalla Terfas, and Witold Suryn. "On the use of ISO/IEC standards to address data quality aspects in Big Data Analytics cloud services". In: *International Conference on Business Information Systems*. Springer. 2017, pp. 149–164.

[26] Jungwoo Ryoo et al. "Cloud Security Auditing: Challenges and Emerging Approaches". In: *IEEE Security Privacy* 12.6 (2014), pp. 68–74. DOI: 10.1109/MSP.2013.132.

[27] Amazon Web Services. *AWS GovCloud (US)*. [EB/OL]. https://aws.amazon.com/govcloud-us/ Accessed May 26, 2020.

[28] Dave Shackleford. "The devsecops approach to securing your code and your cloud". In: *SANS Institute InfoSec Reading Room A DevSecOps Playbook* (2017).

[29] Preston Smith, Baijian Yang, and Carolyn Ellis. "Trusted CI webinar: REED+ Purdue's Evolution From a CUI Environment to an Ecosystem to a Community". In: (2021).

[30] Aishwarya Soni and Muzammil Hasan. "Pricing schemes in cloud computing: a review". In: *International Journal of Advanced Computer Research* 7.29 (2017), p. 60.

[31] International Organization for Standardization. *ISO/IEC 27001:2013*. 2013.

[32] International Organization for Standardization. *ISO/IEC 27002*. 2005.

[33] International Organization for Standardization. *ISO/IEC 27017*. 2015.

[34] International Organization for Standardization. *ISO/IEC 27018*. 2019.

[35] International Organization for Standardization. *ISO/IEC 27032*. 2012.

[36] Najat Tissir, Said El Kafhali, and Noureddine Aboutabit. "Cybersecurity management in cloud computing: semantic literature review and conceptual framework proposal". In: *Journal of Reliable Intelligent Environments* (2020), pp. 1–16.

[37] Jinesh Varia, Sajee Mathew, et al. "Overview of amazon web services". In: *Amazon Web Services* 105 (2014).

[38] Veritis. *DevSecOps Solution to Cloud Security Challenge*. https://www.veritis.com/blog/devsecops-solution-to-cloud-security-challenge/.

[39] Cong Wang et al. "Privacy-Preserving Public Auditing for Secure Cloud Storage". In: *IEEE Transactions on Computers* 62.2 (2013), pp. 362–375. DOI: 10.1109/TC.2011.245.

[40] Tim Weil. "Taking compliance to the cloud—Using ISO standards (tools and techniques)". In: *IT Professional* 20.6 (2018), pp. 20–30.

[41] Kan Yang and Xiaohua Jia. "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing". In: *IEEE Transactions on Parallel and Distributed Systems* 24.9 (2013), pp. 1717–1726. DOI: 10.1109/TPDS.2012.278.

Mobile Cloud Computing: Service Level Techniques for Connectivity Constraints

Group 7

June 4, 2021

Abstract

With the rapid development of wireless network infrastructures, the number of mobile devices growth rapidly during the past decade. Accessing remote large scale cloud computing data centers for an external support with data storage and processing has become a common demands nowadays. The mobile cloud computing, which utilized mobile web and cloud computing, demonstrates a number of great properties that could orchestrate the mobile devices by different types of applications. However, the portable devices with wireless network inherently suffer from the fundamental constraints of network connections. In the service level, two main actors take different strategies to handle the network constraints. The service provider, which is the first actor, makes a choice between those proactive and reactive methods with the trade-off. Besides, the bandwidth optimization is also an option. As for the service consumers, the caching and prefetching techniques optimized the efficiency of data processing. So does the application offloading, which provides another alternative for dealing with network instability.

1 Introduction

Nowadays, the number of mobile devices is rapidly growing with the commercialization of 5G techniques. By 2020, the number of mobile devices has reached 14 billions. By 2024, this number is expected to be 17.72 billion.[1] As a result, the mobility devices have deeply engaged in the users' daily life, which gradually accumulated demands for services for mobile applications. Under those demands of portable and mobility, the mobile cloud computing, which is considered as a combination of mobile web and cloud computing,[2] provides a solution to resolve the demands of accessing the large scale resources from the remote cloud infrastructures in an efficient and convenient way. However, despite of the great advantages of mobile cloud computing, mobility of is considered to be more hazardous compared to the traditional wired cloud computing, especially in the network connection aspect. So in this review, after observing the different actors

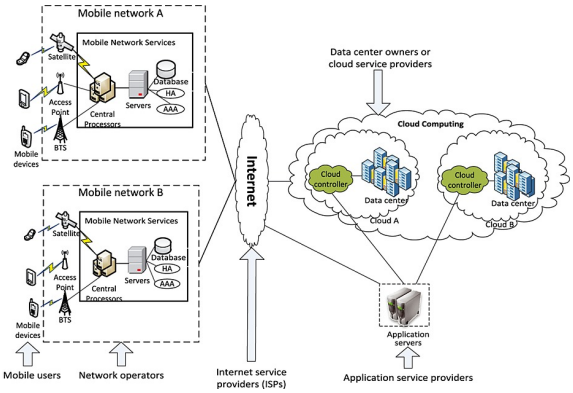


Figure 1: Architecture of Mobile Cloud Computing

from the mobile cloud computing architectures, the methods and strategies of handling network connectivity constraints would be discussed both from the service providers and applications.

The structure of the review would go as follows. The rest part of the introduction includes the detailed background information of the mobile cloud computing, which will also include some of the motivations in our reviews. Then the section 2 includes the strategies that were taken by the services providers. The methods included were divided by the exact constraints the works intend to handle with. Similarly in Section 3, this part introduces the methods from the application sides. And finally, Section 4 tries to sum up with all the ideas and also included some future works or some assumptions from the most recent works.

1.1 Overview of Mobile Cloud Computing

There are various versions of definitions of mobile cloud computing. According to the definitions from mobile cloud computing forum and some related works, one of the most widely used definition could be 'refers to an infrastructure where both the data storage and data processing happen out side of the mobile device. Mobile cloud applications move the computing power and data storage away from mobile phones and into the cloud, bringing applications and MC to not just smartphone users but a much broader range of mobile subscribers.'[3, 4] At the same time, some other versions of definitions were also mentioned, which explained mobile cloud computing in a rather simple way by cloud computing with mobile web services.[4] In fact, those two definitions of mobile cloud computing explain the same things in the brief and detailed ways, which were consistent.

To clarify the quoted part as well as how it work for combining mobile web services with cloud computing, the figure 1 from [4] illustrates a very basic version of architecture of mobile cloud computing, which comply with with both of the definitions above. To look into the left part which could be considered as

the mobile web services part, users with different mobile devices could access to the mobile networks based on their locality. With different mediums in-between, the devices connected to the lowest hierarchical levels of the mobile network services, which is then forwarding to upper levels through the Internet. Then as for the right side, the service provider would be able to deploy the elastic cloud computing over the large scale data centers.

In this way, a wide range of mobile devices could be able to access the remote computing powers and data storage and executing workloads outside the mobile devices. This is also the way to orchestrate mobile web services with cloud computing.

1.2 Motivation of Mobile Cloud Computing

Despite of the design that supports a wide range of devices with different network middleware, there are various advantages of mobile cloud computing, which lead to a boost of mobile devices applications. Firstly, those beneficial properties were addressed from different perspective as follows.

Energy Efficient: With the lithium battery contained, the mobile devices are struggled with the concern of battery life, which can be resolved by moving the data processing part out of the mobile devices. Moreover, some preliminary works also put their efforts in optimize the power management of mobile systems as well as some low power designs.[4, 5] Those researches tried to design models for mobile devices with the power consumption of CPU, memory, GPS readings and other components of the devices. In this way, it would be possible to develop the power-aware applications, which could optimize the power usage.

Improving Resources Available: With mobile cloud computing, one of the biggest motivation is to access a larger scale of cloud computing resources, including data storage and data processing. However, managing such a great number of devices that keep moving unpredictable could be a complicated process. For instance, the cloudlet offloading techniques provide the idea of using a cluster of computers as an access point, which also lead to the optimized placement and allocations of cloudlet especially under the senerio of wireless metropolitan area networks(WMAN).[6]

Dynamic Provisioning: Dynamic provisioning, which refers to accessing the resources dynamically without needs of resource reservations before, is not always guaranteed by the definitions. However, for the quality of service concern, most of the common-used programming models and systems have their own scheme for this property. To take an example, the AlfredO framework, which is a conceptual extension of the OSGi model, computes the optimal deployment of the services and provides elastic resources management.[7] Similarly, supporting for seamless VM migrations in Cloudlet and public cloud [8] would be another demand for high performance when meet with the limitation of resources.

The advantages of mobile cloud computing mostly refers to a better performance and higher availability of the remote cloud services with the portable devices, which has a limitation of batteries and computing or storage resources. And the vision of researches is mainly focus on orchestrate the mobile devices

with the cloud servers in smooth manner with higher performance.

Derived from the great properties of mobile cloud computing and the great number of demands under wide range of use cases, a great number of applications could be developed with the back-end of mobile cloud services. By summarizing all the implementations of application designs, three groups of applications has became the main part of the mobile cloud computing applications. The first type of the applications are accessing the cloud resources, which means to use the mobile web services as the middleware to extend the limitations of mobile devices. For instance, the Amazon Simple Storage Service (Amazon S3)[9] provides an online storage services which is available by the mobile devices, which could be considered as a simple way of extending the storage of devices. The second type of applications would be sensor related work, with which the cloud computing could collect and work with the data from the sensors of mobile devices. It provides a possibility for the remote cloud servers work with the local data. For instance, most of the navigation system on the vehicle would be able to collect the traffic information to the cloud and provides the feedback to other users, which is one of the use case of croudsensing. Thirdly, the mobile cloud computing also provides the possibilities for communications between devices, which is the demand of the social networks. The Facebook, Twitter and Instagram all work for their services with lower latency and higher reliability.

In a nutshell, mobile cloud computing has great properties that built a bridge between the mobile devices and remote large scale cloud server. With the easily accessible resources and sensors, new variations of applications could be developed based on mobile cloud computing.

1.3 Connectivity Constraints in Mobile Cloud Computing

Apart from the positive aspect of mobile cloud computing, there are also a large number of challenges. However, in this review, only the constraints of connectivity would be discussed. Mobility refers to changing from wired connection to wireless connection, which reduces the reliability to the whole networked system. 'Mobility is inherently hazardous.' The research in 1996 pointed out that mobility of portable devices would lead to the higher risk of physical damage and variable in connections.[10] Those challenges brought about with mobility would always exist as an issue in the mobile cloud computing. Although according to the summarize of frameworks in [11], one of the mono-objective optimization-based framework named CloneCloud proposed a methods for partitioning the mobile applications to optimize the overall costs. However, the CloneCloud made a somehow radical prediction of strong wireless network connection to optimize the model accurately, although the assumption has not realized until 10 years later. As for other frameworks, the Hyrax uses the fault tolerance mechanisms of Hadoop. Computational Offloading Dynamic Middleware(CODM) provides management of resources and quality level and make adaptions based on the connection status. As for Virtualized Mobile Replicas Offloading Architecture (VMROA), it uses loosely synchronization for less bandwidth demand.[11] To sum up, nearly all of the mobile cloud based frameworks take the unreliable

connection as one of the a core demand to handle in the design.

Moreover, those constraints of connectivity is not the same among all the senerios of applications. In this way, the following three types of constraints of wireless network connections would cover most of the conditions with the frameworks.

Low bandwidth: Wireless network is only the last 1km of the Internet connection, which is the diffused extension of the cable. All the connected devices shared the total bandwidth of one cable. The bandwidth for each mobile device would be relatively lower than the devices with wired connections. Moreover, due to the millimeter wave of 5G only have an effective range of 500 meters from the tower, the LTE and low bandwidth 5G would still go with 5G in the future.[12] It means that the bandwidth would be restricted by the distance to the nearest antenna.

Mobility of devices: The mobility of devices could also be a problem. Whenever the devices migrate out of the range of the connection, it would try to establish another connection, which would lead to the variation of network environment. With newly established connections, sometime the VM image should be migrate seamlessly for better Quality of Service. This process would raise some new issues about resources management.

Network fluctuations: The information of wireless network is carried by the wave through air. Comparing with the data traffic throw the cables, it is much more likely to get much latency, noise or even get blocked during the reflection and refraction during the transmitting period. Similar to the conditions in devices mobility, since the effective range of the signals is restricted by the power of antenna, devices might switching from one tower to another, which could lead to non-negligible connection lost.

As the matter of the fact, different actors in the mobile cloud computing systems take different methods to get rid of those connectivity constraints. In the hardware level, the mobile devices manufacturers would try to work with larger batteries and lower power usage. In the network connection level, the network providers try to optimized allocate more antennas that covers as much part of the areas as possible, which also serve with higher level of bandwidth. Moreover, in the service level, the service providers and service consumers would take different strategies for different constraints which might become the bottleneck of the system within different use cases. The following part of the review will mainly focus on these two actors' methods.

2 Strategies in service providers

There are some strategies aimed at solving the above connectivity constraints on the side of cloud service providers. The two major problems we focused on in this section are user mobility and limited bandwidth in MCC.

2.1 Mobility

As stated before, mobility, as a nature of mobile devices, causes a big challenge in MCC. In this part, we reviewed several current strategies focused on reducing communication interruptions while changing the Point of Attachment (PoA) of mobile nodes, which is considered ad handover management[13]. The handover techniques are categorized as proactive and reactive in [14], and the overview of methods introduced in this paper is shown in Table 1.

Table 1: Strategies for mobility

Methods	Handover Procedure	OSI Operating layer
PMF in [15]	Proactive	Clouds
UMAP in [16]	Proactive	Clouds
TMSS in [17]	Proactive	Clouds
D-PMIPv6 in [18]	Reactive	Network
Bicasting in [19]	Reactive	Network

2.1.1 Proactive

In proactive methods, the cloud or network predicts the new location of mobile terminals to establish a data forwarding path before initiating the handover process in order to provide a smooth communication environment between nodes.[14]. These methods predict potential handover actions in the clouds with information about mobile devices and the network. The prediction methods evolve from simple predicting, such as straight lines model in [16], to complicated algorithms based on the essential patterns of user mobility, such as the TMSS (Tail Matching Sub-Sequence) in [17]. All these methods are proposed for cloudlet systems in MCC, whose architecture is shown in figure 2.

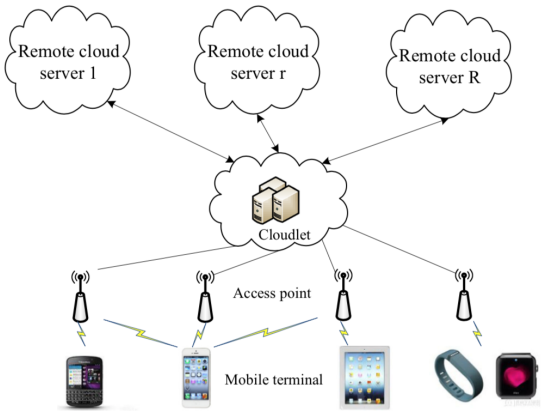


Figure 2: Architecture of Cloudlet Systems

Researchers in [16] recognized the importance of mobility in seamless connectivity and proposed an offloading algorithm, which makes offloading decisions based on the predictive locations of mobile devices for an intermittently connected cloudlet system. The prediction method in [16] is based on the assumption that the geographical distributions of both cloudlets and mobile users follow an independent Homogeneous Poisson Point Process (HPPP) [20], [21]. Based on the HPPP assumption, they use the distribution of mobile devices and distribution density of cloudlets to calculate the probability that users can access the cloudlets successfully and obtain an optimal handover policy to reduce offloading failures. However, in the implementation stage, users are assumed to move in simple straight lines, significantly reducing the accuracy of predictions.

[15] improved the straight mobility pattern in [16] with the random user mobility model to reduce service interruptions in resource allocation during offloading. However, this method still fails to consider the essential pattern of user mobility features in the real world, such as periodicity. Therefore, the Tail Matching Sub-Sequence (TMSS) algorithm is proposed in [17] to represent the inherent patterns of user mobility and make predictions on the terminal's access. It first obtains all candidate Target Access Points (TAPs) using sequence matching in the access history of mobile devices and removes all duplicate TAP. Then it uses the TMSS algorithm to calculate the access probability of the remaining TAPs and the largest one is considered to be the next access point. The effectiveness of this method was proved by their experiments with the Dartmouth dataset.

Proactive or predictive methods ensure less latency and smooth migration between different access points in frequent movements of mobile devices. However, it also requires additional computation power and high signaling costs between networks and mobile terminals.

2.1.2 Reactive

By contrast, reactive methods only trigger the handover process and establish a new connection with the candidate network after the mobile node is attached to a new network. These methods are usually implemented in the network layer, such as the [18] and [19]. However, the late release of mobile terminals brings more handover delays and causes communication failures and packet loss. Because the reactive approach does not efficiently solve the communication problems caused by the delays, it has become a less useful method.

The D-PMIPv6 proposed in [18] is inspired by Proxy Mobile IPv6 (PMIPv6) and aimed at improving the performance of PMIPv6 with a new distributed mobility management framework. The framework splits the function of Local Mobility Anchor (LMA) into two parts: the control plane (CLMA) and the data plane (DLMA). The responsibility of CLMA is to manage signaling about binding registration, such as Proxy Binding Update, and the DLMA is responsible for transmitting data traffic between different MAGs with a localized routing mechanism proposed in that paper.

Researchers in [19] proposed a reactive data bicasting handover procedure

to reduce Service Interruption Time (SIT) and the packet loss during frequent movements between LTE macrocell and femtocell. The method defines mobile management entities (MME) to control the handover. The MME triggers handover when it receives HO required messages and sends a data bicasting request to the serving gateway (S-GW), which will then bicast the request to both the source and the target. With the bicasting method, the number of lost data packets was small, however, for strict lossless handover requirements, it also provides a buffering mechanism. It uses the two Key Performance Indicators (KPIs) to estimate the needed buffer size to avoid packet loss as much as possible during the handover process.

Reactive methods are less efficient in communication latency and packet loss but have low signaling expenses. On the contrary, proactive methods minimize communication failures with a higher cost of signaling. It's a trade-off between fewer communication failures and higher signaling and computational power costs when choosing from these two kinds of methods.

2.2 Low bandwidth

The low bandwidth is another nature of mobile devices, which will cause congestion in the communication with remote clouds and lead to delays in resource transfer. From [22], there are few methods targeted at solving the limited bandwidth problem by the time the paper was written, which is 2018. They proposed two directions to overcome the bandwidth overhead, which are bandwidth optimization and load scheduling. Bandwidth optimization methods aim to optimize the efficient usage of bandwidth when allocating the bandwidth resource. The load scheduling methods allow mobile users to share their wireless network bandwidth to maximize the utilization of bandwidth. However, most current approaches focus on optimizing bandwidth resources, thus we will only talk about the optimization models in this paper.

2.2.1 Bandwidth optimization

Bandwidth optimization methods are about allocating bandwidth resources more efficiently, which formulate the wireless bandwidth allocation problem as an optimization problem, such as the triple-stage Stackel-berg game in [23]. The optimization model needs some parameters, such as the cost of service, to obtain optimal allocation solutions.

In [23], a triple-stage Stackel-berg game model was proposed to address the problem of wireless bandwidth allocation in a cloudlet-based MCC. The proposed triple-stage Stackelberg game has three optimizing goals. First, it aims to maximize the utility of mobile devices when the allocated bandwidth is less than the total bandwidth of the network. Second, it needs to maximize the revenue of the cloudlet by providing virtual machines to the mobile terminals as many as possible. Finally, it has to find the optimal VM price of the cloudlet to maximize each cloud server's profit. It uses a backward iterative algorithm to

achieve the equilibrium of the Stackelberg game and proves its effectiveness with some simulation experiments.

Researchers in [24] proposed a cheating-resilient bandwidth distribution (CRAB) algorithm, inspired by the descending price auction theory. In this algorithm, the interface gateways between mobile terminals and cloud service providers are regarded as buyers, and the cloud service providers act as sellers. The process of finding the optimal bandwidth allocation performs as a trade-off between the number of bandwidth requests and the unit price of bandwidth, whose optimization goal is to maximize the revenue of cloud service providers.

Those bandwidth optimization methods have a good performance in allocating bandwidth resources efficiently to maximize the usage of bandwidth. However, research works about limited bandwidth are still not abundant, and more works about load scheduling should be attempted and proposed.

3 Strategies in service consumer

Mobility limitations affect not only service providers, but also at the service consumer. Because mobile devices need to connect to the network via wireless and cellular data networks, this exposes them to network latency problems caused by low bandwidth and unstable network connections. Therefore, in this section, we will focus on some measures to address the mobility limitations at the service consumer aspect caused by these two factors.

3.1 Caching and prefetching

Mobile devices have become the main gadget for people to access network, as well as data[25], which means optimize the data access and network stability are essential.

3.1.1 CAFE schema

In [25], it introduces a caching and prefetching schema based on a Cloudlet model(CAFE schema). This schema is designed to reduce latency and improve bandwidth utilization by increasing the processing efficiency and more efficient type of data transfer between the service consumer and service provider. Prefetching is one of the strategies used to reduce the network latency and caching can mitigate data access latency and improve data hit ratio in mobile environments[26].

In the CAFE schema, prefetching defines the predicted access patterns mainly based on the user's access traces and then requests the cloud to fetch the data that the user may request in the future in advance based on the prediction algorithm. CAFE solutions utilize correlation rules for prediction because they can identify large-scale data. This is because correlated prefetching between data items improves the hit rate of caching and by identifying data that has a high probability of being accessed. The data is cached as soon as it is fetched,

which reduces the delay in network connection time. This is because, in the state of early prefetching, the mobile device is guaranteed to get the cached data in advance even if the network connection is unstable.

In the CAFE schema, mobile devices request data from the service and download it via Cloudlet. To determine which mobile device is expected to request data, the architecture uses a spatial index structure. The spatial indexing structure uses a tree structure to store multi-dimensional data to improve search and update efficiency. As a result, the structure can be updated quickly if a mobile device enters or leaves coverage, which ensures that if Cloudlet finds several mobile devices that hold the same data, it can find the closest one to communicate with. The mobile device first checks if the requested data is in its memory, and if the information is not there, the mobile device sends the request to Cloudlet. Then it searches for the information in its corresponding specific data set, which saves communication time. This approach improves bandwidth utilization and to some extent solves the problem of network latency caused by low bandwidth.

3.1.2 Cooperative caching framework

Another approach is a cooperative cache based data access frame work for mobile cloud computing. The proposed approach uses the cloudlet architecture presented by M.Satyanarayanan[27] [28].The purpose of adopting cloudlet is to reduce the distance between cloud services and mobile devices by using cloudlet as an intermediate layer between the cloud and mobile devices, which can be used to reduce network latency for mobile users, as end-to-end user latency increases when the distance increases. With cooperative distributed caching, data traffic and latency will be greatly reduced.

In [28], it introduces that cooperative cache consists of multiple distributed caches, which allows the system to handle concurrent client requests as well as shared content and retrieve objects from different cache sites simultaneously. All this is done to reduce the system response time, which caused by network latency and low bandwidth problems. The cooperative cache caches the combined state of the different virtual machines from which the user can get services. When the object is not in the cache, a request is sent to the base tier to get the corresponding startup VM, the VM can separate the ephemeral customer software environment from the permanent host software environment. If the corresponding base tier VM does not exist, the remote cloud is contacted to obtain the service.

Cooperative caching can only improve hit rates and reduce response times if the three conditions of uniform cache distribution, extensive cache sharing, and low discovery overhead are met.

The two methods mentioned above address network latency by improving bandwidth utilization through increased data transfer in mobile devices and service providers, thus reducing the latency between network communications. Also caching and prefetching can keep mobile devices open for a short time when the network is disconnected, but they cannot completely solve the problem

of network disconnection and low bandwidth faced by MCC at the root. To some extent, they are simply optimized for spotty network connections and poor bandwidth utilization.

3.2 Application offloading

Offloading a task from a mobile to the cloud reduces the load of the local CPU and hence the energy consumption for mobile computing[29]. In MCC, application offloading is implemented as a significant software level solution for sharing the application processing load of smartphones[30], which can not only solves the battery optimization problem of mobile devices, but also solves the problem of network latency while solving the problems encountered by application offloading.

The novel technique of live prefetching, which seamlessly integrates the task-level computation prediction and prefetching within the cloud-computing process of a large program with numerous tasks[31]. In [31], the researcher proposed that living prefetching is mainly done by implementing prefetching based on task-level computational prediction and its simultaneous operation with cloud computing. It designs optimal and suboptimal prefetching policies, selects prefetched tasks given randomly ordered tasks, and controls the size of prefetched data in slow and fast decay derives a simple threshold-based policy structure, thus enabling low-complexity real-time manipulation with minimal mobile energy and network communication consumption.This technique avoids unnecessary prefetching of those tasks in the unlikely execution of the offload procedure by calculating the predictions in real time. As a result, this technique reduces the data rate and mobile transmission energy consumption for wireless transmission. This will also reduce the latency of the network connection. Since mobile devices are connected to service providers through the network, the reduced data rate of wireless transmission will reduce congestion in the network connection.

Although this technology was originally designed with the goal of reducing the energy consumption of mobile devices during data transmission and increasing the lifetime of mobile devices. However, because the analysis of its implementation process can be concluded that it solves the problem of low bandwidth at the same time, because once the transmission it solves the problem of network congestion and improves the efficiency of data communication, it is able to improve the bandwidth utilization to reduce the cost of bandwidth application and moderate the network instability of mobile devices and cloud connections.

4 Conclusions

Under the background of portable devices and wireless network, mobile cloud computing is rapidly evolving to support a large range of devices with hierarchical networks. With the mobile web techniques and cloud based services, several types of mobile cloud applications were developed due to the great advantages.

However, every coin has two sides, the fundamental constraints of mobile cloud computing lead to great efforts from service providers and service consumers.

In order to help solve the connectivity constraints of mobile cloud computing, we also researched some strategies adopted by cloud service providers to improve the quality of service. These strategies are aimed at two major challenges in MCC, including user mobility and limited bandwidth. Methods about user mobility are classified as proactive and reactive, and there is a trade-off to choose from these two kinds of methods. Some bandwidth optimization methods are presented in this paper to maximize the utility of bandwidth resources.

To help address the connectivity limitations of mobile cloud computing, we also examine some of the strategies that service consumers have adopted to improve the quality of service. These strategies aim to address two major challenges of mobile cloud computing, including user mobility and limited bandwidth. The report presents three approaches to address network latency and low bandwidth utilization by improving data transfer rates through caching and prefetching.

5 Study design

5.1 Research topic and Introduction

Research Topic: We firstly decided a wide topic of **Mobile Cloud Computing**. Later on, we look for the surveys to seek for a narrow down topic. And we noticed that in the **service level**, a great number of methods were discussed in the fields of both **application** and **service provider**. We also discussed a few ideas about **avalibility connectivity** before we made the decision.
Background and preliminary work: This is not a big survey of mobile cloud computing. So we did not spent too much time in the introductory part. A few researches about **mobile devices** and **cloud computing** is addressed in this part.
Overview of mobile cloud computing: In this part we intend to clarify the **definitions of mobile cloud computing**. We took two definitions and made a comparison. We also want to show a brief **architecture of mobile cloud computing**, which could demonstrate the two main actors.
Motivation of mobile cloud computing: Two parts were contained in this part. The first part is the advantages of **mobile cloud computing**. To enrich the claims, we found some research topic related to these advantages. Some design for **energy effective** and **resources management**. And in the second part, some **applications** were found. And we classified them into three types.
Connectivity Constraints: This part also contains two part. The first part is to discuss that **connectivity constraint** is a **fundamental problem**. We also noticed that **CloneCloud** assumed a **strong network connection**. We argued about the claim is radical and most of the **frameworks** proposed solutions for unstable networks. The second part refers to the three perspective of network connection constraint. And we also discussed different layers of actors' work. Here we took **service provider** and **service consumer** from the **service level**

as the main actors to focus on.

5.2 Strategies in service providers

Mobility: We used the combination of **Mobile Cloud** or **Mobile Cloud Computing** with **mobility**, **seamless connectivity**, **connection**, **communication**, **reliability** or **QoS** to find papers about user mobility in MCC. If the paper is focused on solving the problems caused by user mobility, such as communication latency and packet loss, in the layer of clouds or networks, we will include it in our research.
Low bandwidth: The combination of **Mobile Cloud** or **Mobile Cloud Computing** with **bandwidth**, **traffic overhead**, **bandwidth optimisation**, **load scheduling**, **access**, **limited bandwidth** or **QoS** is used to find papers about low bandwidth in MCC. If the paper is focused on utilizing bandwidth more efficiently, such as bandwidth optimization and load scheduling, in the layer of clouds or networks, it will be included in our research.

5.3 Strategies in service consumer

Caching and prefetching: We used the combination of **Mobile Cloud** or **Mobile Cloud Computing** with **connection**, **low bandwidth**,**network delay**, **caching** or **prefetching** to find papers which focus on solving mobility problems in MCC. If the paper is relevant to addressing network latency and disconnection, low bandwidth at the mobile device or service consumer level, or mitigating network latency and improving bandwidth utilization by improving the ability to access data between mobile devices and the cloud, we will include it in our research.
Application offloading: We used the combination of **Mobile Cloud** or **Mobile Cloud Computing** with **offloading**, **delay constraints** or **offloading optimization** to find papers targeted at solving mobility constraints by application offloading. If this paper has anything to do with mitigating network latency by offloading reduced bandwidth costs, we will include it in our research.

References

[1] S. O'Dea. *Forecast number of mobile devices worldwide from 2020 to 2024 (in billions)*, 2020. https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/.

[2] Leslie Liu, Randy Moulic, and Dennis Shea. Cloud service portal for mobile device management. In *2010 IEEE 7th International Conference on E-Business Engineering*, pages 474–478. IEEE, 2010.

[3] *Mobile Cloud Computing Forum*, 2010. http://www.mobilecloudcomputingforum.com/.

[4] Hoang T Dinh, Chonho Lee, Dusit Niyato, and Ping Wang. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18):1587–1611, 2013.

[5] Yong Cui, Xiao Ma, Hongyi Wang, Ivan Stojmenovic, and Jiangchuan Liu. A survey of energy efficient wireless transmission and modeling in mobile cloud computing. *Mobile Networks and Applications*, 18(1):148–155, 2013.

[6] Mike Jia, Jiannong Cao, and Weifa Liang. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Transactions on Cloud Computing*, 5(4):725–737, 2015.

[7] Dejan Kovachev, Yiwei Cao, and Ralf Klamma. Mobile cloud computing: a comparison of application models. *arXiv preprint arXiv:1107.4940*, 2011.

[8] Paramvir Bahl, Richard Y Han, Li Erran Li, and Mahadev Satyanarayanan. Advancing the state of mobile cloud computing. In *Proceedings of the third ACM workshop on Mobile cloud computing and services*, pages 21–28, 2012.

[9] Amazon Web Services. *Amazon Simple Storage Service User Guide*, 2021. https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html.

[10] Mahadev Satyanarayanan. Fundamental challenges in mobile computing. In *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing*, pages 1–7, 1996.

[11] Ejaz Ahmed, Abdullah Gani, Mehdi Sookhak, Siti Hafizah Ab Hamid, and Feng Xia. Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges. *Journal of Network and Computer Applications*, 52:52–68, 2015.

[12] VIAVI Solutions Inc. *What is 5G Technology?*, 2021. https://www.viavisolutions.com/en-us/5g-technology.

[13] Deguang Le, Xiaoming Fu, Dieter Hogrefe, et al. A review of mobility support paradigms for the internet. *IEEE Commun. Surv. Tutorials*, 8(1-4):38–51, 2006.

[14] Abdullah Gani, Golam Mokatder Nayeem, Muhammad Shiraz, Mehdi Sookhak, Md Whaiduzzaman, and Suleman Khan. A review on interworking and mobility techniques for seamless connectivity in mobile cloud computing. *Journal of Network and Computer Applications*, 43:84–102, 2014.

[15] Mahinur AKter and Fatema Tuz Zohra. *QoS and Mobility Aware Optimal Resource Allocation for Dynamic Application Offloading in Mobile Cloud Computing*. PhD thesis, East West University, 2016.

[16] Yang Zhang, Dusit Niyato, and Ping Wang. Offloading in mobile cloudlet systems with intermittent connectivity. *IEEE Transactions on Mobile Computing*, 14(12):2516–2529, 2015.

[17] Yan Shi, Shanzhi Chen, and Xiang Xu. Maga: A mobility-aware computation offloading decision for distributed mobile cloud computing. *IEEE Internet of Things Journal*, 5(1):164–174, 2017.

[18] Li Yi, Huachun Zhou, Daochao Huang, and Hongke Zhang. D-pmipv6: A distributed mobility management scheme supported by data and control plane separation. *Mathematical and Computer Modelling*, 58(5):1415–1426, 2013.

[19] Tao Guo, Atta ul Quddus, and Rahim Tafazolli. Seamless handover for lte macro-femto networks based on reactive data bicasting. *IEEE Communications Letters*, 16(11):1788–1791, 2012.

[20] Dmitri Moltchanov. Distance distributions in random networks. *Ad Hoc Networks*, 10(6):1146–1166, 2012.

[21] François Baccelli, Bartlomiej Blaszczyszyn, and Paul Muhlethaler. An aloha protocol for multihop mobile wireless networks. *IEEE Transactions on Information Theory*, 52(2):421–436, 2006.

[22] Talal H. Noor, Sherali Zeadally, Abdullah Alfazi, and Quan Z. Sheng. Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, 115:70–85, 2018.

[23] Sachula Meng, Ying Wang, Zhongyu Miao, and Kai Sun. Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment. *Peer-to-Peer Networking and Applications*, 11(3):462–472, 2018.

[24] Snigdha Das, Manas Khatua, and Sudip Misra. Cheating-resilient bandwidth distribution in mobile cloud computing. *IEEE Transactions on Cloud Computing*, 7(2):469–482, 2016.

[25] Hou Zhijun, Robson E. De Grande, and Azzedine Boukerche. Towards efficient data access in mobile cloud computing using pre-fetching and caching. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, 2017.

[26] Yih-Chun Hu and David B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, page 231–242, New York, NY, USA, 2000. Association for Computing Machinery.

[27] Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4):14–23, 2009.

[28] Preetha Theresa Joy and K. Poulose Jacob. Cooperative caching framework for mobile cloud computing. *CoRR*, abs/1307.7563, 2013.

[29] Karthik Kumar and Yung-Hsiang Lu. Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51–56, 2010.

[30] M. Shiraz, E. Ahmed, A. Gani, and H. Qi. Investigation on runtime partitioning of elastic mobile applications for mobile cloud computing. *The Journal of Supercomputing*, 67(1):84–103, 2014.

[31] Seung-Woo Ko, Kaibin Huang, Seong-Lyun Kim, and Hyukjin Chae. Live prefetching for mobile computation offloading. *IEEE Transactions on Wireless Communications*, 16(5):3057–3071, 2017.

Table 1: Work distributions

Names	Tasks
Hongzhong Yu	Research on the introduction part, Write the corresponding section and the abstract
Li Zhong	Research the strategies on the side of service providers in MCC, Write the corresponding section
Shuhan Pi	Research the strategies on the side of service consumer in MCC, Write the corresponding section

Big data and cloud computing in smart cities

Mohamed Haddadi(10830936)
Yu Wang(12962988)
Mahmoud Abd Elrehim(12626600)
Quanyi Wu(2670800)

Abstract

The current society has digitized every aspect of human life due to technological advancement. Generally, ICT is transforming urban surrounding by providing the requirements for sustaining and ensuring the resilience of smart cities. In most cases, the tools for ICT, particularly for smart cities, revolve around differing application domains, including urban planning, transportation, resource management, public health, and education. As a result, there has been a fast growth of data volume, velocity, and variety. Analyzing these huge data sets from various sources is paramount for organizations plan for their future and anticipation of emerging trends and dynamic nature of their requirements to meet their objectives. This particular study aims to dissect the big data applications in the background of cloud computing in smart cities. Section 1 describes two terms, big data and cloud computing. Section 3 explains big data technology and challenges, and section 4 shows a wide range of big data applications in smart cities regarding domains of urban planning, transportation, resource management, and public health. Section 5 explains the reason why cloud computing is a good solution for big data challenges and section 6 illustrates big data and cloud systems within urban smart cities. Moreover, the benefits that smart cities gain from big data, the importance of smart city applications, and the challenges, opportunities and future concerns in the effective use of big data in smart cities are highlighted in section 7.

1 Introduction

As the twenty-first century progresses, urbanisation, combined with global population growth, has resulted in a massive increase in the population living in cities. With two-thirds of the world's population residing in cities in the following decades, the relevancy of Information Systems (IS) on urban life and global urbanisation topics is becoming more and more critical as stated in [2]. Making a city smart is emerging as a strategy for addressing the problems that come with rapid urbanisation and population expansion. In contrast to traditional cities, smart cities aim for distinct goals based on networked systems and distributed creative processes, which deduced from [26]. Furthermore, a smart city strategy includes utilising cutting-edge technologies, particularly Information and Communication Technologies (ICT), to improve city resilience and quality of life. These technologies open up a whole new world of possibilities for decision-making and innovation. According to [1], the amount of data generated will skyrocket in the coming years. By 2022, global mobile data traffic will be over 71 Exabyte per month, with 60% of that coming from IoT [1], which is a crucial component of any smart city. The generated data can provide information to improve smart city applications, paving the path for new technologies to help with this. On the other hand, smart cities are vulnerable to the hazards of not being able to meet the needs of their rapidly rising metropolitan populations. In addition, [3] has concluded that climate change, economic volatility, social mobility, and cybercrime are all global influences that play out at a municipal scale, posing new problems for smart cities. As the number of people living in smart cities grows, so does the scope of the threats. Due to the

complexity of governing city systems and the unpredictability connected with external threats, risks become more unpredictable. In this sense, a resilient smart city advocates a smart city vision in which efforts are made to improve the smart city's ability to respond to a variety of pressure factors such as climate, environment, energy, and economics. In [25] the ultimate goal is to establish smart city resilience by ensuring a greater quality of life and sustainable urban growth.

Given this scenario, this paper mainly focuses on two important concepts, Big Data and Cloud Computing. And we will elaborate on them in the context of smart cities. In our research, we use the combination of articles focusing on big data challenges from multiple aspects, technological frameworks, quantitative case studies, and applications in several domains. We use scientific articles as our main resources, together with some commercial report published by credible companies.

Big data (BD) refers to huge sets of data that are generated by various programs. In most cases, big data cannot be handled or perused on a regular computer. This data is obtained from digital sources such as the internet, sensors, social platforms, mobile phones, scanners or digitizers. The type of data itself could be in the form of videos, sounds, texts, geometries or integration of each one of them as illustrated in [4]. The continuous technological evolution and human comprehension of data have transformed data handling from traditional approaches to a data arena that comprises value, volume, velocity, and variety. the fact of the matter that, big data encompasses 5Vs. The first V refers to the volume of data. The proliferation of data has exceeded the capability of managing large sets of data. The second V refers to velocity, which translates to the fast generation and transmission of data with the help of the internet. The social media data collected from social sites and the micro sensor data being transmitted to macro level are typical examples of velocity. The third V stands for variety. It refers to different forms of data and the type of structure or model used to archive them. The fourth V is known as veracity. It refers to the difference in quality, precision and reliability of the data. The four Vs are integrated to make the fifth one, value, which concentrates on particular research and decision-support applications that boost the lives of human beings and their success. The dynamic nature of big data, specifically its adoption by the government and industry, broadens big data. The initial meaning of big data was defined based on its volume. However, it has been changed to be comprised of data itself and the technology that is proficient in generating, collecting, storing, managing, processing, analyzing, presenting, and utilizing data.

On the other hand, cloud computing (CC) refers to processing any material that includes big data analytics using the cloud. Cloud refers to a set of servers that are of high power and from manifold providers. Cloud has the ability to view or query huge data sets much faster than a standard computer. Therefore, big data means large sets of data being gathered, and cloud computing means the mechanisms that incorporate this data and conducts any operation needed on that specific data. Cloud computing make use of the Software as a Service (SaaS) structure that owns significant capability in allowing clients to process data conveniently and efficiently [17]. Although it can take special commands and parameters through a console, everything can be done using the user's interface, bring better convenience for all stakeholders. SaaS represents a series of product, such as database management systems, identity management systems and cloud reliant virtual tools. It is currently important to understand the huge amount of data obtained from our surroundings because we are now living in an era where information is crucial. This has ameliorated the significance of data in unlocking infinite possibilities with respect to gaining a competitive advantage.

2 Queries

2.1 Query list

This is the main query we use in our literature review, which is conducted in scopus:

(big data OR cloud) AND (smart city) AND (LIMIT-TO (SRCTYPE,"j") OR LIMIT-TO (SRCTYPE,"p")) AND (LIMIT-TO (SUBJAREA,"COMP") OR LIMIT-TO (SUBJAREA,"ENGI") OR LIMIT-TO (SUBJAREA,"MATH")) AND (LIMIT-TO (PUBYEAR,2021) OR LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO (PUBYEAR,2019) OR LIMIT-TO (PUBYEAR,2018) OR LIMIT-TO (PUBYEAR,2017) OR LIMIT-TO (PUBYEAR,2016) OR LIMIT-TO (PUBYEAR,2015)) AND (LIMIT-TO (LANGUAGE,"English"))

The following criteria have been taken into consideration:

- Important citations by sorting the results on Cited by (highest).
- Recent citations by sorting the results on Date (latest).
- Important details that have several valuable papers and find key papers based on their reference lists.

2.2 Referencing research design

In this literature review, references have been used to acknowledge the authors' contribution to this research topic. With this regards, evidence has been provided in support of the ideas, concepts and arguments elucidated in the review. Research articles have intellectual rights that require an individual to give credit to the work whenever those articles are used. These authors spent years researching for valuable insights and billions of ideas. Moreover, using proper citations enables us to navigate a specific field and map the relevant space for a specific discipline. In this particular paper, references have been selected based on the year of publication and how much they are aligned with this specific research topic.

3 Big data technology and challenges

Big data applications are perceived as improvement of parallel computing, albeit the scale is an exception. The scale is a requirement that comes from the target, including the relationship between dimensions of data and storage units, the extent of parallelism needed for the computation process, and the acquisition of the final results. For big data to be analyzed successfully, resource management is a crucial and fundamental factor that should be carried out with extreme precaution. Notably, these backgrounds utilize a substantial amount of hardware resources that are virtualized so as to optimize the costs as well as the outcomes. However, managing these resources is quite a challenge, especially due to the complexity of big data application systems. As a result, it is imperative to come up with computational architectures that perform better in order to boost the incumbent and future needs of the application.

Owing to the obstacles posed by traditional data, digital big data also has its own predicaments due to its 5V characteristics in various areas of the industry and the government [30].

3.1 Data storage

The storage of data poses challenges due to the volume, variety and velocity features of big data. It is difficult to store data on traditional methods or tools such as hard disk drives, which are subjects to failure, while the data protection techniques are ineffective and inefficient. In addition, the velocity of digital data needs the storage systems to scale up within a short period of time which is a mission impossible with traditional storage systems. Storage systems of cloud provide unlimited virtual storage characterized by high tolerance of vault and hence offer any possible solutions that are important in solving the predicaments generated by big data [19]. However, moving and hosting big data on the cloud is quite costly, attributed to the large scale of data.

3.2 Data transmission

This process takes various stages. Firstly, data is collected from sensors to storage. Secondly, data is integrated from different data centers. Thirdly, data management is carried out by moving the data that has been integrated to other platforms for processing. All these stages tend to be jeopardized by the volume of the data being transferred. For this reason, smart preprocessing mechanisms and algorithms useful in compressing data are required to alleviate the size of data systematically before transferring them. Additionally, moving big data from data centers to cloud systems while developing effective and efficient algorithms is paramount so as to optimize the speed of data transfer and alleviating costs are quite difficult.

3.3 Data management

Heterogeneous data is challenging to manage, conduct analysis and visualize using computers. The features of big data, particularly veracity and variety, reframe the meaning of data management

orientation by demanding novel technologies crucial for cleaning, storing and organizing unstructured data.

3.4 Data processing

The large scale of big data needs more intensive computing resources in order to process it. These resources include a high speed of the CPU, storage as well as network. Processing big data requires resources that are way more sophisticated than traditional computing techniques. However, cloud computing provides a solution through virtually unlimited power that is available for processing on demand. Nonetheless, this has a number of limitations, including the bandwidth of the network, which tends to be ineffective and inefficient.

3.5 Data analysis

Data analysis is the most significant phase in the stages of big data in mining and predicting information. The entire process challenges the complex nature of algorithms and their scalability. Sophisticated algorithms are proposed and implemented to analyze big data.

3.6 Data visualization

This process makes hidden patterns visible and solves correlations that are unknown in order to ameliorate decision-making. The nature of big data is heterogeneous, making visualization a crucial phase to making big data sensible. Obtaining real-time visualization is quite difficult for big data because of its semantics and its structure in general. There are various functionalities of visualization of big data. The first one is taking in highly interactive graphics data visualization excellent practices. The second one is including visual analytics that is intuitive and attainable. The third one is the availability of web-based interactive for sampling, previewing or filtering data before the visualization is done. The fourth one is the in-memory phase of processing, and the last one is disseminated answers and information through mobile gadgets and the internet. All these functionalities are challenging to design and establish with big data because it has many features.

4 Big data technology and challenges in smart cities

This section gives an overview of some of the important research articles about applying big data technology to smart cities and its challenges. These studies highlight how the Internet of Things has made virtually everything quite accessible and easy for the public at a large scale. However, when it has brought ease to the public's life, it has also created a subsequent problem of managing the Big data produced by it. Thus, the following studies cover all those possible aspects, including how the internet and smart devices are used to make human life easier and how to manage the big data created by them efficiently. This section also highlights the importance of utilizing Big data in smart cities and their corresponding benefits.

Figure 1 is an important infographic published by IBM in 2014, which is generally regarded as an excellent overview of smart cities in their early stage within commercial considerations. This figure clearly shows that a large number of domains would be involved in the digital transformation from offline to smart cities, including city planning and operations, transportation analytics, government management, and resource allocation. It also made a prediction of utilizing the cloud when cities start their future digital transformation, which is now being proved as a suitable solution in 2021.

[16] highlighted the big challenge that Big Data has regarding academia, industry, and governments around the globe. Systematically [16] discussed the grand challenges, namely, data complexity, computational complexity, and system complexity. It is concluded from the [16] that big data project practitioners need to work innovatively to manage big data with the help of cloud computing.

[18] conducted a thorough examination of technological foresight and social change. In addition, new ideas have been proposed for advancing new ideas for better monitoring the new smart cities initiative. Furthermore, by using data from sustainable and livable cities to address the Smart Cities initiative better.

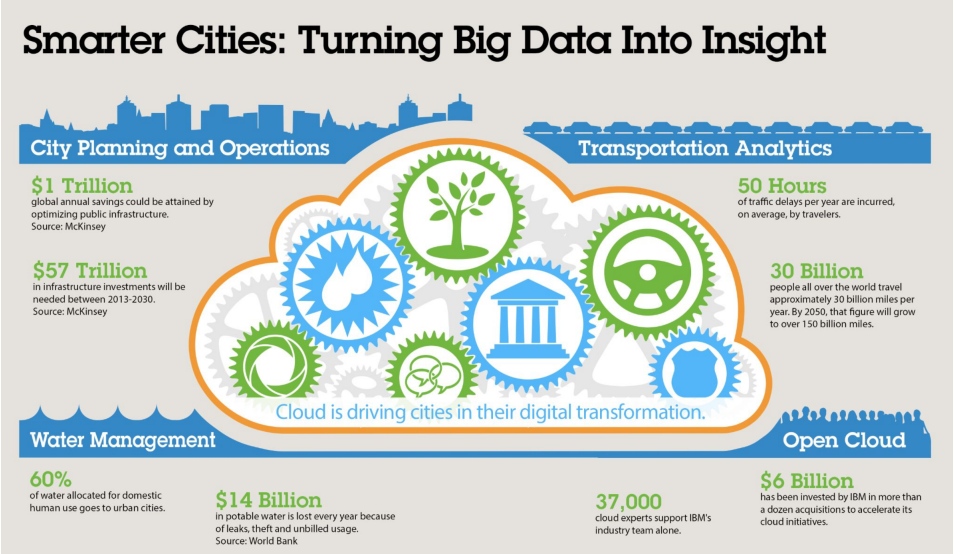


Figure 1: Smart cities: turning big data into insights

In a very important article, [30] expressed that cloud computing is of great help in managing big data in smart cities. It is exclaimed that looking into how Cloud Computing can be used to handle Big Data issues is essential to allow changes. Climate studies, geospatial knowledge mining, land cover simulation, and dust storm modelling were among the four geospatial scientific examples they presented and discussed. As a guide to leveraging Cloud Computing for Big Data solutions, the method is offered in a tabular format. The framework method enables the life cycle of Big Data processing, encompassing management, access, mining analytics, simulation, and forecasting, as demonstrated by the four examples. Cloud computing has evolved as a new paradigm for solving various processing needs with the following characteristics:

(a) On-demand services.

(b) Pooled resources.

(c) Elasticity.

(d) Broad band access.

(e) Measurable services.

The benefit of offering computational capability encourages a potential solution for transforming Big Data's four Vs into a fifth V, which is value.

[9] described that Big data is becoming a more significant "engine" for better understanding open collaboration's complicated "nervous system. It is believed that researchers in this field should provide new datasets, unique computational models, and analytical techniques to open collaboration among researchers. It stressed the importance of open digital collaboration and derived the data analytical challenges that need to be addressed to answer big data in smart cities.

As the improving population comes with accelerating urbanization worldwide, deploying a smart transportation management system and an intelligent resource allocation strategy is important to stabilize the functioning of road transportation. Modeling travel behavior and detecting traffic events with the help of big data can release people from the inconvenience and utilize energy more efficiently by assigning it a larger capacity on running.

A study [12] stated that due to a lack of capacity, rail transit struggles to meet passengers' daily travel needs there is a vast population and urbanization is speeding up. Examining urban travel patterns with Big data assists in infrastructure design and personnel allocation optimization. The research using smart card data in two cities with different geographical features to analyze the temporal-spatial characteristics of urban travel behaviors. Dynamics characterize the travel behaviors at different time and locations, which could show the rule of urban traffic operations in these two

cities. Finally, it concluded that travel habits are linked to urban layout and structure, allowing for a better understanding of urban distribution. Experiments provide policymakers with valuable insights about urban travel behaviour, allowing them to improve rail transit service planning and scheduling techniques.

[10] expressed that many smart city applications and expert systems have been developed as a result of advancements in the Internet of Things, which help citizens and authorities better understand the dynamics of cities and make better planning and use of city resources. Smart cities are made up of intricate systems that process and analyse large amounts of data from the cyber, physical, and social realms. In smart transportation modelling and management, traffic event detection is a critical and challenging task. It use semi-supervised deep learning with data from many modalities, such as physical sensor observations and social media data, to solve this problem. An experiment is performed on the San Francisco Bay Area dataset for four months, containing traffic sensor observations and social media data. The model can automatically extract and classify traffic events based on a large amount of data, which would finally lead to an improvement in traffic management plans.

A study from [5] showed us another great practice of detecting traffic-related event using big data and machine learning methods. This paper proposed an automatic labelling method to detect traffic events from Twitter data in the Arabic language. The method has been implemented in a software tool called Iktishaf+. It uses distributed machine learning algorithms like SVM, naive Bayes, and logistic regression-based classifiers over Apache Spark and MongoDB. The results of detecting and validating several real events in Saudi Arabia without prior knowledge, such as fire, rain, and accidents, have shown the effectiveness of Twitter as a media in detecting important events without prior knowledge. Along with techniques like this, we could have a better management system for smart city transportation resources and significantly reduce the time to solve accidental problems.

Another great practice of big data in smart cities is in the public health sector. [8] researched that overweight, obesity (OO), and type-2 diabetes (T2D) have a significant public health impact that is increasing globally. Rather than a simple energy input (food) and expenditure (exercise) imbalance, the cause of OO and T2D is complicated and multifaceted. However, previous research into food and physical activity (PA) neighbourhood environments has primarily focused on associating body mass index (BMI) with proximity to stores selling fresh fruits and vegetables, fast food restaurants and takeaways, or neighbourhood walkability factors and access to green spaces or public gym facilities, making largely naive, crude, and inconsistent assumptions. Different people and population groups respond differently to the same food and PA environments, mainly due to a variety of unique individual and population group factors and their complex interplays with each other, such as genetic/epigenetic, metabolic, dietary and lifestyle habits, health literacy profiles, screen viewing times, stress levels, sleep patterns, environmental air and noise pollution levels. Furthermore, because a single grocery shop or fast food restaurant can frequently offer or serve both healthy and unhealthy options/portions, a simplistic binary classification into 'good' or 'poor' business/outlet should be avoided. Moreover, though vital for overall health and disease prevention, regular physical activity is not highly effective for weight maintenance or reduction (particularly when primarily relied upon) and cannot offset a poor diet's effects. It expressed that scientific research should focus now on using systems thinking approach to investigate and consider all of these factors. This appeal is in the quest to design better targeted and more effective public health interventions for OO and T2D control and prevention, aided by recent advances in sensors, big data, and related technologies. However, this research might lead to some privacy-related issues when gathering data from grocery shopping and the customer's health conditions. If these issues could be solved by privacy-by-design, a more effective public health intervention system would definitely be of great help in people's health conditions, which is not only limited to OO and T2D.

Aside from public health, with the increasingly severe climate problem on earth, sustainability is now becoming the top issue in the field of urban planning and development. This priority has also been integrated with the concept of smart city and resulted in a new but essential concept in this big data era, the smart sustainable city.

[29] built up an assessment framework to evaluate smart sustainable city using big data and analytic network process. This paper is a great bridge to reduce the gap between sustainable cities and smart cities. It uses a static evaluation system and a dynamic model constructed by big data techniques to evaluate urban environmental planning and design strategies. Big data techniques are mainly used to simulate changes in urban environments under dynamic conditions, such as regional development

policies and space structures. The assessment framework with big data simulations can capture the effects of both economic development and environmental issues on urban planning. The case studies showed us the series of weights of indicators and performance scores for various cities like Taipei and Singapore. This paper is of great inspiration for researchers and the urban planning industry to incorporate big data techniques with urban planning evaluation methods. It could lead to a more scientific and quantified assessment method as well as better capability to capture changing situations, which is especially necessary for big cities with complex situations like New York or San Francisco.

As described by [31] and [32], the energy efficiency of the public sector is an important issue in the context of smart cities. Buildings are the biggest energy consumers, and public buildings have an even larger usage frequency, such as education, health, government, and military purpose public buildings. This paper integrates the big data collection platform with predictive machine learning models to predict energy consumption for each energy source in public buildings. After experiments, researchers found that the combination of Boruta variable selection and random forest algorithm showed the best results. Moreover, the most important variables in this problem are heating and occupational, constructional, cooling, electricity, and lighting. This integrated intelligent energy system can be used in public building information system and their IoT networks to better manage and improve the energy efficiency of the public sector. Also, this proper energy management system will be of great help in digital transformation, leading to the increased energy efficiency of public administration, higher quality of public services, and healthier environment of the smart cities.

Another practice of big data practice in smart cities is in the field of logistics. The growing importance of integrating digital technologies like big data into the logistics of port cities has caught much more attention during the past five years. A study from [11] gives us an excellent overview of the logistics field. The paper proposed a framework using many state-of-the-art technologies to collect, process, monitor, and analyze big data concerning port cities' economic, environmental, social, and technological sphere. These technologies include IoT sensors, cloud computing platforms, big data analytics, AI, GPS tracking systems, drones, real-time monitoring stations, and smart grids. Based on these technologies, mobile and fixed platforms are helping logistics operators to digitally and efficiently optimize the management of flows about water, waste, emissions, raw materials, people, and monetary investments. Moreover, the various factors, domain, and goals that characterize smart and sustainable logistical development are also described in details in the paper. However, the current ideal framework has not been implemented in an actual port city due to its complexity and high infrastructure requirement. Instead, this paper performed a case analysis of the best practices in several pioneering port cities such as Rotterdam, Hamburg, Singapore, Los Angeles, and Amsterdam. These pioneering cities have already (partly) launched the technology in their logistics with companies like Cisco, IBM, Huawei, SAP. As seen from these pioneering cities, advantages of such a combination of big data and logistics in port cities could be concluded as the following points:

(a) Having the potential to enhance the efficiency of the environmental, economic, technological, and social flows.

(b) Increasing the awareness and involvement of stakeholders about utilizing big data techniques.

(c) Activating the process of transition from traditional logistics process into sustainable digital solutions.

The integrated framework would benefit various stakeholders in the logistics pipeline of port cities, including policymakers, urban managers, port authorities, local administrators, shipping companies and couriers. Integrating big data into port logistics aims to develop increasingly digitalized logistic processes and promote smart and sustainable logistics development in port cities. Thus, we could see the significant value of this study in real life, especially for modern port cities with higher digital level and more significant cargo throughput.

Aside from the domains we mentioned above, [13] recently published a study regarding a very recent global issue, the Covid-19 pandemic, in the field of utilizing big data in smart cities. The study proposed a Social Network Software (SNS) big data analysis framework for Covid-19 outbreak prediction in smart city. As we can see in the current big data era, smart cities citizens rely more on SNS rather than traditional communication media to follow official news and latest updates regarding the outbreak, share their opinions, and express their feelings and symptoms. This situation makes twitter a useful data source for predicting the Covid-19 outbreak based on people's social media data. This paper performs NLP on 10000 English tweets collected within two months with the spatial location in the USA. Moreover, the results demonstrated an outbreak cluster predicted seven days

earlier than the number of confirmed cases significantly increased. This paper demonstrates the importance of adopting smart city applications under sudden situations and using big data techniques to help people deal with significantly unpredictable scenarios. However, as mentioned by [23], the connectivity is unequally distributed across cities and neighbourhoods, which is clearly revealed by the attack of Covid-19. These disparities have a severe effect on the smart city application, not only on the individual level but also on the community level, leading to bias, uncertainty, and even misleading decisions. Thus, if the advanced ICT infrastructure are being more and more deployed worldwide, the social media data will show its power in predicting global trend in many aspects, such as disease outbreak, congestion, water supply.

Above mentioned articles briefly state the influx of data produced by the Internet of Things and many other applications used in smart cities. They have also shown us the great practice of applying big data techniques to the smart city domain by mentioning how various methods have been used to protect and save the big data and use it for better effective functioning. Therefore, with the opinion proposed by [27] and all articles mentioned above, we can conclude four vital components in smart cities with greatest importance, which are people and environment, smart utilities, smart technology, and smart administration.

However, there are still various challenges with these big data application in smart cities, such as cost control, energy consumption efficiency, and infrastructure upgrade. The solutions would be discussed in the following section.

5 Solving the challenges of big data using cloud computing

There are several valuable technologies in tackling the challenges of big data. However, cloud computing has emerged as the most significant and elusive. Technically, the solutions provided by cloud computing are fine-grained and provide services to a large number of users that work independently from different locations with unshared data that consists of multiple interactions and not batch-oriented. Cloud computing and big data share the concerns of autonomic resource management and scaling [6].

Cloud computing is maturing with time. It has pushed many organizations to build effective and efficient cloud surrounding while cloud computing providers continue to grow and expand their territories through the provision of services. These challenges are fixed by two major factors, namely scalability and provision of resources on demand.

5.1 On-demand resource provision

Big data's major challenges arise from two of its features, namely volume and velocity. Empirical research has been conducted and is still being done to comprehend the big data applications and their dynamic patterns. This will enable the formation of a clear structure that is useful in predicting the system's behaviour for the evolution of usage patterns and alteration of working loads.

5.2 Scalability

Scalability serves as an upper hand in processing big data using cloud computing. The scalability of the cloud is contingent upon various factors such as the scope of clusters, whether virtual or physical, the ratio of data to compute and power consumption. Due to the improvement of different clouds with a particular having its own advantages and disadvantages, establishing a hybrid cloud consisting of multiple clouds that have been integrated together would be an ultimate novel trend while incorporating cloud solutions to utilize various clouds.

6 Big data and cloud systems within urban smart cities

The broadening of big data and the transformation of the internet play a huge role in ascertaining the feasibility of initiatives in smart cities. Big data provides the capability for cities to gain significant insights from extensive data obtained from various sources [15]. The internet makes it possible to integrate sensors, blue tooth and radio-frequency in real-time with the help of high-network services.

The joining of the internet and big data has brought about new trends and challenges that stand in the way of the future of small smart cities [7].

It is worth noting that these challenges emphasize the predicaments connected to both business and technology that make these smart cities realize the vision, principles, and needs for applications for smart cities. This is made much easier by actualizing the major features of a smart surrounding. Smart cities benefit from the information provided by big data collection, data integration, analysis and sharing that is made possible by cloud computing services [14]. However, this form of data needs the correct software tools in order to visualize in a smart city or modern environment.

Currently, most nations are considering adopting the concept of smart cities through the implementation of big data applications that boost the components of smart city to meet the required extent of sustainability and ameliorate living standards. Smart cities make good use of technologies to enhance the performance of education, health systems and transportation, just to mention a few [20]. As a result, the residents of smart cities are able to live comfortable lives and consequently alleviate costs and resource utilization.

Big data analytics is one of the major technologies that significantly impact the services of smart cities [28]. Without analysis of the huge amounts of data collected, big data serves no purpose. Therefore, data analysis that is conducted effectively and efficiently is a primary determiner of the prosperity of businesses and the entire domain of a smart city. Applications of smart city produce vast amounts of data. At the same time, the intelligent systems use these big data to avail the information that is vital for enhancing the applications of smart cities. Big data systems offer storage, processing and mining services to information in the smart city applications to ensure information is generated to support different domains in smart cities. Additionally, big data assist in decision making with respect to expansion in the services in smart cities [33].

The relationship beteween smart city and big data applications is shown in figure 2.

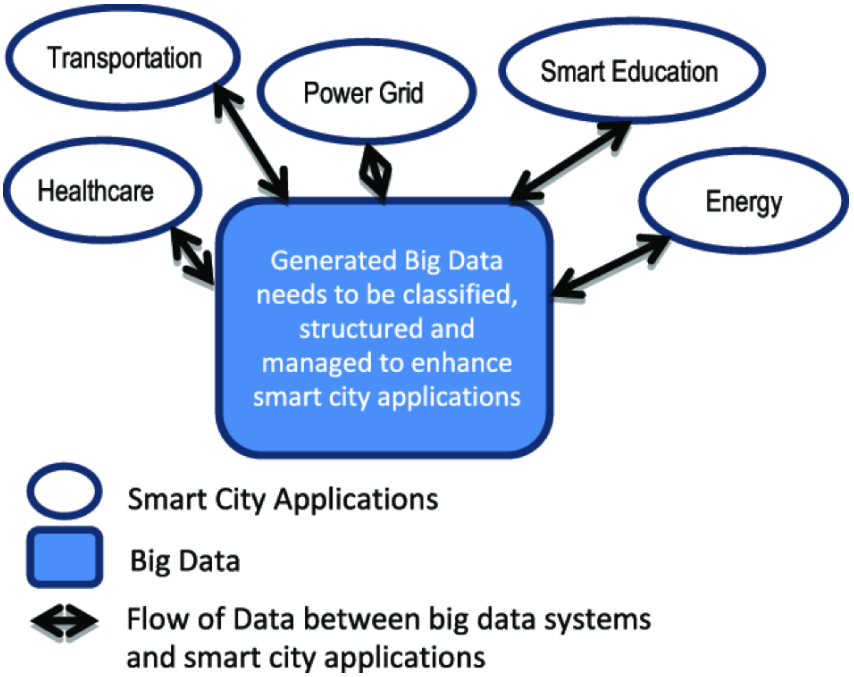


Figure 2: The relationship between smart city and big data.

Smart cities enjoy the benefits of technological advancement both economically and socially. This has shaped the competition between many cities in an attempt to transition into smart cities.

There are many benefits that smart cities enjoy from big data [22]. Firstly, systematic utilization of resources is guaranteed. Technological advancement tools such as Enterprise Resource Planning (ERP) and Geographic Information System (GIS) can ensure effective and efficient resource allocation and monitoring. Secondly, there is an increased level of openness and transparency because data has

to be shared with every critical individual. This facilitates effective management and autonomy of the smart city. Lastly, improved quality of life is an outcome of improved planning and performance of the smart cities domains. Additionally, the availability of sufficient information results in better decision making. The benefits elucidated above are generated by highly sophisticated applications, personnel and resources. Therefore, investments have to be made in the technological advancement and utilization of big data. However, policies need to be set straight so that data accuracy and precision is guaranteed in conjunction with data security and quality.

The outcome of smart city applications is shown in figure 3.

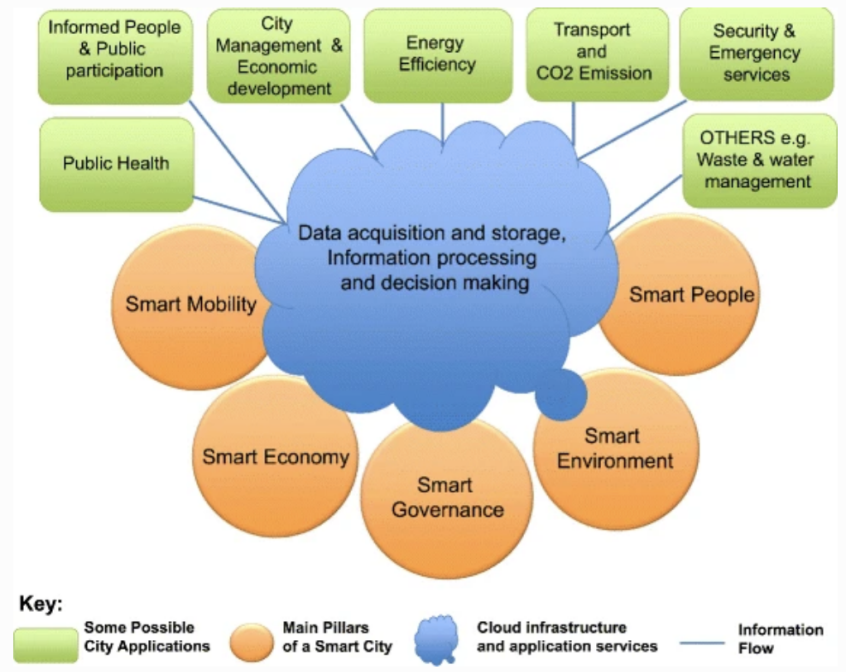


Figure 3: The outcome of smart city applications.

Big data application can be successfully deployed with the help of a well-developed ICT system. For instance, the ICT system can ease the movement of data or information from one location to another, thereby minimizing transport costs. Thus, a smart city is transformed into a smarter city when the appropriate ICT infrastructure is utilized. Therefore, a combination of cloud, big data and ICT assist in addressing several issues, for instance, the provision of tools for storage and analysis of data.

This is a pathway to effective communication and cooperation between different parties in a smart city. This can be made possible by developing big data groups that can work as one entity to enhance the collaborative and innovative solutions to predicaments.

There are two classifications of the applications of big data to smart cities, namely offline big data and real-time big data [20]. These applications are categorized based on the period of time at which data analysis is conducted. Real-time applications depend on instant input and quick analysis to decide within a short period of time that is also specific. For these particular applications, data should be made available in a timely manner so that reliable decisions can be made [21]. For this reason, applications require technological advancement. Offline big data applications, on the other hand, are utilized in planning smart cities, for example, education and traffic.

Although big data is beneficial in smart cities, designing and developing applications for such cities is quite challenging [24]. One major challenge associated with big data applications for smart cities is the multiple nature of the features and sources of data. Data can be available in differing formats, sizes, and characterized by different features, as mentioned earlier, thereby complicating the entire process. Another challenge revolves around sharing of data and information of every section of the smart city. This makes the information being shared public even when it intended for confidential purposes. Moreover, data quality might be jeopardized due to the lack of standard techniques to

collect and store data. Also, the cost of big data is high when all the users are factored in, for instance, using a system that reduces energy and makes tests in smart traffic lights.

In conclusion, smart city and big data are two significant concepts that improve every aspect of the lives of individuals residing in smart cities. Despite the differences between the two concepts, there are benefits associated with utilizing big data to operate the applications of smart cities. Working using big data is challenging because it requires applications that are more complex and extensive. For this reason, cloud computing is used to solve these challenges. In order to build and develop successful smart city applications, addressing predicaments or obstacles along the way is paramount. Figure 1 and figure 2 clearly depict the relationship between big data and smart cities and the results of utilizing smart city applications, respectively. Nonetheless, it is essential to highlight the two major factors that ensure the safety of applications in smart cities.

These two factors include privacy and security. Big data needs to be integrated and shared with all the relevant personnel. This requires privacy and security to avoid jeopardizing the value of the information at hand.

7 Discussion

Despite the fact that smart cities are becoming increasingly popular throughout the world, their definition remains a mystery. As far as the published literature is concerned, it indicates that certain well-known traits can be identified in a smart city, although they are still poorly defined. Nonetheless, there appears to be consensus on what a smart city would do for its residents and the environment.

In general, a smart city will enhance government performance, the city's economic status, the quality of life of its residents and contribute to the development of environmentally friendly and sustainable infrastructures. As a result, some common qualities, features, and components that may define the viewpoints of a smart city have been highlighted.

These include increased use of ICT and next-generation information technology, the use of ICT to integrate the physical and social components of the city, the implementation of advanced monitoring and control tools and applications to improve efficiency and quality, and the improvement of infrastructures to support a better quality of life and greater sustainability. These considerations apply to all smart city proposals, regardless of their size. Governments all across the globe are worried about the cost and advantages of using smart city technology. Many people are concerned about financial trends, available resources, and regulatory system capabilities, as they are difficult to address. On the other hand, new technology can assist in the mitigation of some of the problems and provide greater potential for success. Furthermore, there is much promise for leveraging big data to address many of the challenges that come up in smart cities by employing analytics to get deeper insights and make better decisions.

We looked at and contrasted several definitions of big data because it is seen as a key enabler for smart city applications. The different Vs of big data demonstrate the complexity and difficulty of collecting, managing, storing, and analysing large amounts of data. The sheer volume and diversity of big data, on the other hand, presents a fantastic chance to develop smart applications that respond efficiently to current data and provide accurate decision-making tools. Including big data applications to support smart cities is not without its hurdles; yet, effective implementations will catapult a city to new heights in terms of smartness. Multiple nations throughout the world, including South Korea, the United States, and the United Arab Emirates, are being urged to construct and promote smart cities as a result of this forward-thinking technology.

Understanding the features of smart cities and recognising the requirement for enhanced big data and ICT support makes the process of combining all of these technologies to begin developing smart city apps much easier. Policymakers may now look at how smart cities might be planned and built. These may be divided into three categories: "public infrastructure," "public platform for smart city construction," and "application system construction". Each of these areas has topics and difficulties that might be investigated more in the future. Many of the concerns and obstacles will be addressed, and answers will be found as plans grow and more research and development efforts are poured into smart city design. As a consequence, more cities will begin to become smarter, improving the overall quality of life.

Furthermore, comprehensive, trustworthy strategic plans for smart cities beyond fragmented initiatives or stand-alone projects are required. Such designs must consider the many smart city requirements (physical, social, and technical) and avoid treating each component as if it were a separate silo. The holistic approach will assist provide a clearer picture of what is required, resulting in more comprehensive, well-designed smart city solutions rather than islands of disparate components and applications that cannot recognize or interact with one another. As a result, efforts should be focused on developing a success roadmap that includes various stages. Some of those stages are being discussed below.

- Using the acquired data and smart apps, optimise smart city services and operations to improve services and identify infrastructure and environmental improvement needs.
- Construct smart-ready public infrastructures and platforms, including the ICT necessary to enable smart city applications. This will entail assessing and analysing present problems and making the required modifications and additions to achieve the desired outcome.
- To govern ICT and large data utilisation, establish rules, principles, resources, and expertise guidelines.
- Determine the goals and utilise them to identify the most significant smart city components and apps with the biggest impact for the least amount of money.
- To provide better and more efficient citizen experiences, integrate infrastructures, services, and big data smart city applications.
- Monitor existing developments and their impact, emergent difficulties, and new needs to identify new prospects for continued development.

Indeed, the use of ICT and information technology, particularly big data, will open up a plethora of possibilities for developing smart city applications that will effectively and efficiently meet the demands of the many entities that live in and utilize it. As a result, sufficient resources and funding must be allocated to support application development activities at all phases of smart city development. This investment is necessary to reap the full advantages of smart cities and achieve all features and capabilities that have been promised. It is advised that some of the following actions can be included in the process to assist the job and reduce project costs:

- Developing simulation technologies to aid in the prediction and visualisation of probable changes and the forecasting of prospective difficulties. This will assist in preventing or at least mitigating some of the dangers involved and minimising installation and testing expenses in many circumstances.
- Taking use of other smart city experiences in order to follow successful models and avoid risky approaches.
- Using experts and researchers to investigate existing market systems (smart systems/services, data systems) and new possibilities for more advanced systems that are compatible with the smart city's goals.

The relationship between big data and smart city applications is being investigated. This knowledge will aid in incorporating the appropriate data into the appropriate apps in order to make better judgments and improve various operations in the smart city.

As we get to the end of this topic, we can attest to the importance of big data in smart city applications. We have gone through various instances of how to use big data and the advantages of doing so. However, certain unresolved concerns must be addressed and overcome before big data can be used successfully for smart city applications. Several unresolved concerns arise from the many obstacles we described before, while others may be related to additional factors we overlooked. Nonetheless, many of these unresolved concerns are now being investigated and scrutinized by the business and scholarly sectors. However, there are no comprehensive answers available, and there is always the potential for development and innovation in this subject. Some of these unresolved concerns include, but are not limited to:

- Political concerns and consequences on any city have an impact on how well (or poorly) it performs, and this is true of smart cities as well. Various persons in varied power or political positions have different access to information, which must be considered and treated properly.

- Security and privacy concerns are other key factors to consider. Data will be shared across all organisations in the smart city after all systems are interconnected. As a result, infrastructure and platforms must be safeguarded, privacy must be maintained, and data must be adequately safeguarded.

- Is Social Media a significant data source in smart cities, and how will the communication between governments, residents, and businesses be? Should all public and private institutions have access to the same information and knowledge when everything is connected and integrated?

Another topic to investigate is the negative consequences of employing technology. We must analyse all the hazards and repercussions of their usage since we will have a communication infrastructure that spans private and public networks, many of which may be wireless. Furthermore, there will be no board for numerous devices owned and controlled by different persons for varied reasons and with such a wide range of ICT knowledge. It is unclear how this amount of technological connection will influence consumers or whether there will be any negative consequences. Many people worry about the negative consequences of having mobile phones around for long periods, so it is only natural to wonder about the implications of these technologies being integrated into the lives of smart city inhabitants.

The demand for highly educated, competent individuals to design, develop, install, and run smart city infrastructures, platforms, and apps are continuously increasing. To build this workforce, specialised education and training in these fields must be established and supplied.

For smart applications, it is necessary to establish standard metrics and control principles. In order to assure the correctness, efficacy, and quality of deployed smart city apps, a smart city must monitor and regulate projects and implementations using various tools and methodologies.

8 Conclusion

Since smart cities and big data are two modern and important concepts, many people have begun to combine them to create smart city applications. These applications will aid in achieving sustainability, improved resilience, effective governance, improved quality of life, and intelligent management of smart city resources. Our research looked at these applications and discovered certain similar characteristics for both big data and smart cities. Regardless of the different definitions, each notion has a set of features that distinguishes it. We were able to determine the broad benefits of using big data to create and support smart city applications based on these common traits.

Following that, we explored the different options accessible, which will lead to the development of smart apps that can use all available data to improve their operations and outcomes. We also reviewed the numerous hurdles in this field and outlined numerous concerns that might stymie build big data applications. We proposed a list of general requirements for big data smart city applications based on that discussion. Designing and implementing effective and efficient applications necessitates the fulfilment of certain requirements. Furthermore, these criteria attempt to address the obstacles by proposing several approaches to resolving some of the concerns and achieving better results.

Finally, we highlighted some of the major unresolved problems that need to be explored and addressed further in order to gain a more thorough understanding of smart cities and develop them in a well-thought-out form.

Building and deploying successful big data smart city applications will necessitate addressing challenges and open issues, including adhering to rigorous design and development models, having well-trained human resources, utilizing simulation models, and being well prepared and supported by governing entities. Making a city smart will be feasible with all success criteria in place and more profound knowledge of the ideas. Further upgrading it for smarter models and services will be a realistic and sustainable aim.

9 Work distribution

Name	Work (Sections)
Mohamed Haddadi	Section 7 and 8
Yu Wang	Section 2 and 4
Mahmoud Abd Elrehim	Section 5 and 6
Quanyi Wu	Section 1 and 3

Table 1: Work distribution

References

[1] Ericsson mobility report June 2017. https://www.ericsson.com/49de56/assets/local/mobility-report/documents/2017/ericsson-mobility-report-june-2017.pdf.

[2] Hashem T.-Chang V. Badrul-N. Adewole K.-Yaqoob I. Chiroma H. Abaker, I. The role of big data in smart city. *International Journal of Information Management*, 36(5):748–758, 2016.

[3] A. AlDairi and L. Tawalbeh. Cyber security attacks on smart cities and associated mobile technologies. *Procedia Computer Science*, 109:1086–1091, 2017.

[4] Dhunny Z. A. Allam, Z. On big data. *Artificial intelligence and smart cities.*, 89:80–91, 2019.

[5] Katib I. Albeshri A. Yigitcanlar-T. Mehmood R. Alomari, E. Iktishaf+: A big data tool with automatic labeling for road traffic social sensing and event detection using distributed machine learning. *Sensors*, 21(9):2993, 2021.

[6] Thakuriah P. V. McHugh-A. Sun Y.-McArthur D. Mason P. Walpole R. Anejionu, O. C. Spatial urban data system: A cloud-enabled big data infrastructure for social and economic urban analytics. *Future Generation Computer Systems*, 98:456–473, 2019.

[7] Seng K. P. Zungeru-A. M. Ijemaru-G. K. Ang, L. M. Big sensor data systems for smart cities. *IEEE Internet of Things Journal*, 4(5):1259–1271, 2017.

[8] Koh K. Boulos, M. N. K. Smart city lifestyle sensing, big data, geo-analytics and intelligence for smarter public health decision-making in overweight, obesity and type 2 diabetes prevention: the research we should be doing. *International Journal of Health Geographics*, 20(1), 2021.

[9] Bertino E. Matei S. Brunswicker, S. Big data for open digital innovation–a research roadmap. *Big Data Research*, 2(2):53–58, 2015.

[10] Wang W. Huang K. De-S. Coenen-F. Chen, Q. Multi-modal generative adversarial networks for traffic event detection in smart cities. *Expert Systems with Applications*, 177:114939, 2021.

[11] Szopik-Depczyńska K. Dembińska I.-Ioppolo G. D'Amico, G. Smart and sustainable logistics of port cities: A framework for comprehending enabling factors, domains and goals. *Sustainable Cities and Society*, 69:102801, 2021.

[12] Wang-J. Gao C. Li-X. Wang Z.- Li X. Deng, Y. Assessing temporal-spatial characteristics of urban travel behaviors from multiday smart-card data. *Big Data Research*, 576:126058, 2021.

[13] Singh S.K. Park J.H. EL Azzaoui, A. Sns big data analysis framework for covid-19 outbreak prediction in smart healthy city. *Sustainable Cities and Society*, 71:102993, 2021.

[14] Chang V. Anuar-N. B. Adewole-K. Yaqoob I. Gani A Chiroma H. Hashem, I. A. T. The role of big data in smart city. *International Journal of Information Management*, 36(5):748–758, 2016.

[15] Yaqoob I. Anuar-N. B. Mokhtar-S. Gani A. Khan S. U. Hashem, I. A. T. The rise of big data on cloud computing: Review and open research issues. *Information systems*, 47:98–115, 2015.

[16] Wah B. W. Cheng X.- Wang Y. Jin, X. Significance and challenges of big data research. *Big Data Research*, 2(2):59–64, 2015.

[17] Anjum A. Soomro K. Tahir-M. A. Khan, Z. Towards cloud based big data analytics for smart future cities. *Journal of Cloud Computing*, 4(1):1–11, 2015.

[18] Colomer-Llinàs J.- Meléndez-Frigola-J. Marsal-Llacuna, M. L. Lessons in urban monitoring taken from sustainable and livable cities to better address the smart cities initiative. *Technological Forecasting and Social Change*, 9:611–622, 2015.

[19] Nesmachnow S. Tchernykh-A. Avetisyan-A. Radchenko-G. Massobrio, R. Towards a cloud computing paradigm for big data analysis in smart cities. *Programming and Computer Software*, 44(3):181–189, 2018.

[20] Al-Fuqaha A. Mohammadi, M. Enabling cognitive smart cities using big data and machine learning: Approaches and challenges. *IEEE Communications Magazine*, 65(2):94–101, 2018.

[21] Skarmeta-A. Moreno, M. and A. Jara. How to intelligently make sense of real data of smart cities. *International Conference on Recent Advances in Internet of Things (RIoT)*, 2015.

[22] Terroso-Sáenz-F. González-Vidal A.-Valdés-Vela M. Skarmeta A. F. Zamora M. A. Chang V. Moreno, M. V. Applicability of big data techniques to smart cities deployments. *IEEE Transactions on Industrial Informatics*, 13(2):800–809, 2017.

[23] Tolbert C.J. Mossberger, K. Digital citizenship and digital communities: How technology matters for individuals and communities. *International Journal of E-Planning Research*, 10(3):19–34, 2021.

[24] Triyason T. Pal, D. and P. Padungweang. Big data in smart-cities: Current research and challenges. *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, 6(4), 2018.

[25] Galderisi A.-Vigo Majello M. C. Saretta-E. Papa, R. Cmart and resilient cities a systemic approach for developing cross-sectoral strategies in the face of climate change. *Tema-Journal of Land Use Mobility and Environment*, 8(1):19–49, 2015.

[26] Ricciardi F. Pierce, P. and A. Zardini. Smart cities as organizational fields: A framework for mapping sustainability-enabling configurations. *Sustainability*, 9(9):1506, 2017.

[27] S.R. Salkuti. Smart cities: Understanding policies, standards, applications and case studies. *International Journal of Electrical and Computer Engineering*, 11(4):3137–3144, 2021.

[28] Song H.-Jara A. J. Bie R. Sun, Y. Internet of things and big data analytics for smart and connected communities. *IEEE access*, 4:766–773, 2016.

[29] Peng-T.-C. Wey, W.-M. Study on building a smart sustainable city assessment framework using big data and analytic network process. *Journal of Urban Planning and Development*, 147(3), 2021.

[30] Huang Q.-Li-Z. Liu K. Hu F. Yang, C. Big data and cloud computing: innovation opportunities and challenges. *International Journal of Digital Earth*, 10(1):13–53, 2015.

[31] Mitrović-S.-Has A. Zekić-Sušac, M. Machine learning based system for managing energy efficiency of public sector as an approach towards smart cities. *International Journal of Information Management*, 58:102074, 2021.

[32] Mitrović-S.-Has A. Zekić-Sušac, M. Predicting energy cost of public buildings by artificial neural networks, cart, and random forest. *Neurocomputing*, 439:223–233, 2021.

[33] Luo J. Zhou, Q. The study on evaluation method of urban network security in the big data era. *Intelligent Automation Soft Computing*, 24(1):133–138, 2017.

Literature review: Open Cloud innitiatives

Aida Lavandera Gonzalez
Bas van der Borden
Paul Puttbach
Web Services and Cloud-Based Systems - Group 10
University of Amsterdam

June 4, 2021

Abstract

The concept of Open Science has evolved to meet the needs of researchers regarding data size and computing capacity. Open Science Cloud initiatives are arising all over the globe during the past two decades, and they aim to provide their users with an infrastructure and services for Open Science practices and research.

Currently several regions, nations or institutions are developing Open Science Cloud platforms to store, share and process data and tools for research. This paper will review the main initiatives in this field, taking as example the European project for Open Science Cloud.

The research objectives are to determine how the European Open Science Cloud compares to other Open Science Clouds in regards to their evolution, level of maturity, usage and standards. Is the European Open Science Cloud a pioneer and model for good practices and standards?

1 Introduction

The term "Big Data" has increasingly become more popular during the past two decades, the amount of data available keeps growing at an exponential rate to this day.[19] In order to manage the humongous amounts of data, distributed computing has seen a similar rise in its popularity. The demand for computing power and services fostered the widespread emergence of cloud computing infrastructures.

According to NIST (National Institute of Standards and Technology), "Cloud computing can be defined as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction"[23].

Cloud computing infrastructures in general have been around for a while, but the idea to use such a highly distributed accessible infrastructure for Open

Science is somewhat newer. In this paper the focus will be on the Open Science Cloud computing initiatives, which have been designed and launched during the past 6 years and make use of the Cloud Computing infrastructure to facilitate cooperative, open scientific progress.

1.1 What is an Open Science Cloud?

The concept of Open Science is not a new one. The Human Genome project for example is considered the first Open Science project and dates back as far as (1990)[13]. This international research project had the goal to discover the sequence of all the base pairs of the human DNA. Collaboration between different researchers and institutions made it possible for the sequencing to be completed faster. Another contributing factor was the general improvement and higher computation capacity of both hardware and software which enabled researchers to work with processing power like no one before them.

The idea behind Open Science is the nature of collaboration: the sharing of data, knowledge and techniques for the progression of research projects that span various fields. The concept of an Open Science Cloud bridges the gap between the goals of Open Science and the need for a solution to efficiently store, share and process data used in scientific research. Open Science Clouds aim to provide researches with infrastructures, services and standards to work with vast amounts of data (Petabyte scale) and tackle ever increasing computational challenges[17].

Open Science Clouds also strive to adhere to the FAIR principles[33] for data, the aim of which is to establish guiding principles supporting the reuse of scholarly data. The four FAIR principles for data are:

1. Findable: making data and tools easily findable by both humans and computers is achieved by including the right metadata. This medatadata being machine readable is key for the automatic discovery of both datasets and services, and therefore the re-usability of these.

2. Accessible: once the data is found it also need to be accessible to the user potentially allowing authorization and authentication. This is underlining the open aspect of Open Clouds

3. Interoperable: the data needs to be interoperable in regards to storage, applications, analyses and processes, as well as easily integratable with other data.

4. Reusable: the end goal is for the data to be re-usable, to ensure this data must be well described and maintained, so that it's easily replicated or integrated. Machine actionability of the data is of key importance here.

1.2 Brief history review

The creation of Open Science Cloud infrastructures was a response to the growing need of the scientific community to tackle the challenges of datasets increasing

their size from MB to GB to TB to even PB[18]. The main challenges were:

1. The storage and transportation of these huge datasets.
2. The need to balance data management and data analysis capabilities: most databases are optimized for data accessing via indexing, but not for numeric computation intense tasks.
3. Archiving, updating and migrating large datasets.
4. The lack of formal definition and standards around Cloud Computing.

As an answer to these challenges, the Open Science Grid (OSG) Consortium introduced the concept of the OSG as a national petascale distributed facility for science in 2007[28]. While the majority of the communities participating on the project were global, the resources were still not easily available to all researches, since the participating stakeholders still had the need for the purchase of hardware and building of computational facilities by and for each science community.[28]

Later in 2015, the European Commission presents the initiative of the European Open Science Cloud (EOSC). This initiative is presented as "a federated environment for scientific data sharing and re-use, based on existing and emerging elements in the Member States, with light-weight international guidance and governance, and a large degree of freedom regarding practical implementation."[9]

During the past recent years, similar initiatives have flourished in other regions of the world. Some such initiatives are the China Science and Technology Cloud (CST Cloud)[26] and the Australian Research Data Commons (ARDC)[8], both first launched in 2018, or the Malaysian Open Science Platform[25] or LA Referencia (Latin America)[30], which both launched in 2020.

As more and more similar initiatives are created globally, the Global Open Science Cloud Council has started working towards a global and interoperable cloud, the Global Open Science Cloud (GOSC)[14]. The aim of this last project is to bring together all these aforementioned (and many more) scattered projects under the same standards.

1.3 Aim of this research

In this review, the current status of most renowned Open Science Cloud platforms is presented. The central point of comparison is the European Open Science Cloud (EOSC) initiative enabled by the European commission. The reason behind this choice for the main initiative for comparison is that the EOSC is the state of the art Open Science infrastructure, the first of these initiatives to offer cross-border open access to both services and data, as well as adhering to the FAIR principles, whilst being the most technically mature platform.

The concept for Open Science is not new, however, real implementation of internet-based Open Science platforms for data sharing and processing has only been realized in the past decade. For this reason, there is not a lot of

literature available that provides a clear picture of the status, similarities and interconnection of Open Science Cloud projects.

The objectives for this research are to provide a clear introduction to the topic of Open Science Cloud platforms. Furthermore, the research focus will be on providing an overview on the current status of the European Open Science Cloud, their role as pioneers in realizing a fully functional open cloud for researchers and how other Open Cloud initiatives compared to their European counter parts.

Research question How does the European Open Science Cloud fare compared to other Open Science Clouds in regards to their evolution, level of maturity, usage and standards? Is the European Open Science Cloud a pioneer and a reference point for similar initiatives?

2 The European Open Science Cloud versus similar initiatives

The following section contains an overview of the current status of the European Open Science Cloud initiative, a description of its perceived role as pioneers as well as a summary on the other main initiatives globally that are similar.

Lastly, the future trends on globalizing open cloud initiatives and the short term goals for the Global Open Science Cloud commission are presented.

2.1 Current Status of the European Open cloud

The current status of the European Open Science Cloud (EOSC) can be viewed through different perspectives. Firstly a discussion of the current status through its usage. Secondly a review the financial side of EOSC. And finally the governance changes EOSC has undergone will be presented.

Starting with the usage of EOSC. It is found that EOSC is broadly supported throughout the EU, in which is necessary to mention that this is not only through the European Union but also from countries themselves. According to [3] EOSC is currently supported through 33 different countries, whilst [5] reports support from 37 EU members. With usage not only the support from the EU and its members is important but also the interaction with institutions and other providers. Currently projects and resources can be found through the EOSC Portal.

"The EOSC Portal is a gateway to information and resources in EOSC, providing updates on its governance and players, the projects contributing to its realisation, funding opportunities for EOSC stakeholders, relevant European and national policies, important documents, and recent developments" [2]

From this it is concluded that the EOSC Portal is the way to interact with EOSC and from it we found the following: The EOSC Portal divides all it's resources in either Scientific Domains or Categories. The Scientific Domains supported are: Medical Health Sciences, Engineering Technology, Natural

Sciences, Generic, Humanities, Agricultural Sciences, Social Sciences, Support Activities and finally Other.

Switching to the financial perspective on EOSC. [1] states that through the initial development 320 million euros was invested through project calls in Horizon 2020.

"Horizon 2020 is the biggest EU Research and Innovation programme ever with nearly €80 billion of funding available over 7 years (2014 to 2020) – in addition to the private investment that this money will attract. It promises more breakthroughs, discoveries and world-firsts by taking great ideas from the lab to the market."[4]

Continuing on EU countries and countries associated with Horizon 2020 agreed unanimously to run EOSC as a co-programmed European Partnership under Horizon Europe from 2021. Relating this back to our financial perspective we find that there is little change in the way EOSC is funded, Although the names of the funding programmes may change the underlying structures stay the same.

Finally from the governance perspective on EOSC. From the beginning EOSC's view on governance was always to be lightweight. This resulted in 4 governance recommendations from HLEG [24]:

1. Aim at the lightest possible, internationally effective governance.
2. Guidance only where guidance is due.
3. Define Rules of Engagement for service provision in the EOSC.
4. Federate the gems (and amplify good practice).

From these recommendations came the previous interim governance structure which consisted of:

- EOSC Governance Board
- EOSC Executive Board
- EOSC Stakeholders

From 2021 onwards a new governance model will be used by the EOSC. This new model will be tripartite including:

- The EU represented by the Commision
- The European research community represented by the EOSC Association
- EU countries and countries associated with Horizon Europe represented through a Steering Board

Concluding on the current status of the EOSC it is noted that the EOSC is broadly supported and used, with both a stable financial and governance perspective. Furthermore the EOSC provides a timeline on ongoing and future projects, as depicted in figure 1.

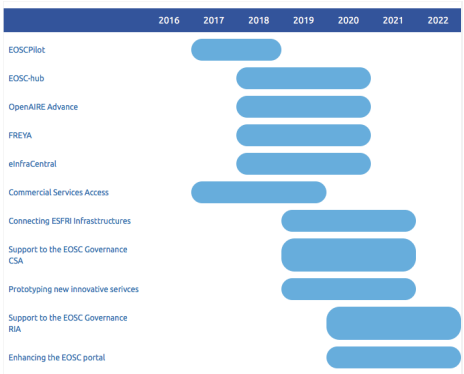


Figure 1: EOSC Timeline.
Taken from https://eosc-portal.eu/about/eosc on 4-6-2021

2.2 The role of Europe as pioneers in Open Science Cloud

To look at the role of Europe as pioneers in Open Science Clouds, it is necessary to look at how the EOSC came to be. The first step towards EOSC was made in May 2015, where in [6] one of the first policy references to open research data and cloud can be found. Furthermore, the announcement to launch a cloud for research data, the "research Open Science Cloud", was created. This resulted in [7] being published a year later in which the EOSC was first described. Finally in November 2018 the EOSC was officially launched. The journey towards the launch of the EOSC is described in [12].

Based the information available, it is noted that Europe was one of the first pioneers within the Open Science Cloud scene. Especially with regards to the EOSC not being restricted to one country, as can be seen with other Open Science Cloud initiatives. In the following sections other different initiatives are presented and more information about their development plans can be found.

Whilst looking to Europe as pioneers in setting standards, it is important to note that many of these standards are in regards to policy and governance. As can be seen with the EOSC, governance and policy were two main pillars on which the system is build. Firstly by guidelines and recommendations by the HLEG on governance with drafts on the EOSC. Secondly with the process in realizing the EOSC, described as a Greek tragedy by [12].

Lastly the EOSC can be considered pioneers with regards to interoperability and standardization. Here it is found that the work from Europe on the EOSC portal focuses on interoperability, and from the development timeline we find that this is a current focus point of the EOSC. With regards to standardization little information can be found. The information available for resource providers of the EOSC portal describes a common template for the provided service. Furthermore the standardization efforts made by the EU and EOSC are constricted with the FAIR principles.

2.3 A revision of the status of other Open Science initiatives

Soon after the launching of the EOSC, other countries and regions also introduced similar projects pursuing the same goals. In this section some of the main initiatives are reviewed and if applicable a comparison is made to the EOSC and the level of influence of the EOSC in the conception and structure of these other initiatives.

The following is not an extensive list, as many of these projects are in the early stages of their development and changes are occurring rapidly. It is expected that other projects are launched in different countries or regions in the coming months and years.

2.3.1 CST Cloud in China

The China Science and Technology Cloud (CST) is the initiative taken by the Chinese government to facilitate scientific research. The platform was created as part of the "13th Five-Year Plan" informatization program of Chinese Academy of Sciences[22] to be implemented between 2016 and 2020.

The first version of the CST cloud was officially launched in April 12, 2018[26], and a subsequent 2.0 version released on December 2019. The platform can be accessed nationally by researchers and scientists and it aims to be the central point for data retrieval, storage and sharing within the scientific community.

The CST platform offers sophisticated infrastructures for Authorization and Authentication, advanced networking and computing services as well as connection to the main national data centers.

Even within its relative short time of existence, the CST cloud has already been a useful tool within a variety of projects, ranging from small particles to Space Exploration projects or even to combat the COVID-19 outbreak originated in China by supporting the CAS Shanghai Institute of Materia Medica and the CAS Institute of Microbiology[27].

2.3.2 Australian Research Data Commons

The Australian Research Data Commons (ARDC) is also an initiative that aims to provide researchers with data and infrastructure that facilitate scientific research performed with high volumes of data.

Similar to the cases of Europe and China, this initiative is also part of a bigger scheme (in this case a 20-year vision) to facilitate scientific capacity and results[10]. The ARDC concept was firstly introduced in 2018 and the foundation infrastructure released in 2019.

In line with other initiatives discussed, the ARDC does not only provide the data storage and sharing capacity, or the distributed computing services; it also aims to unify these activities by setting standards among the scientific community and providing and single point of reference.

Even from its inception, it has considered collaboration and alignment with similar Open Cloud initiatives, including the EOSC[10].

2.3.3 NDRIO New Digital Research Infrastructure Organization

Amongst the plethora of Open Science Cloud initiatives the NDRIO is the Canadian variant. The NDRIO aims to facilitate Open Science data and data service exchange through the means of a "member based not for profit" [11], centralized, government regulated digital infrastructure that Canadian researchers may access and utilize. It is still in the beginning stages of the development and launched just recently in march 2020. A board and governing structure was build and recruitment is well on the way with a membership size of 136 organizations and institutions.

However, considering its relatively recent inception there are still quite a few steps to be taken to achieve the self describe criteria for success in [11]. These criteria are very similar to the necessary steps found by the "Commission high level expert Group" in "realising the European Open Science Cloud ([24]). E.g for the success of the European Open Cloud it was found that "Innovative, fit for purpose funding schemes are needed to support sustainable underpinning infrastructures and core resources." [24] whereas the NDRIO annual report defined further success of NDRIO as "Bringing access to stable federal and provincial funding through new models that are predictable and reflective of national services" [11]. So both organizations understand a need for the shift in the funding scheme to realize the vision of collaborative data reuse under the FAIR principles.

2.3.4 NFDI National Research Data Infrastructure

NFDI is another Open Science initiative out of Germany. It keeps its organization very loosely defined in the smaller sclare (infractructure between different science departments) to allow for easy integration in existing research facilities and to allow other existing open data infrastructures to integrate into NFDI seamlessly as an independent consortium. The smallest element of the structure of the organization is a consortium [16]. Typically these consortia are defined based on a specific scientific discipline.

The consortia are organized into consortia assembly as a means of interdisciplinary, inter organizational research. So each Consortium is part of the network of consortia assembly. The consortium has to govern its own operation, goals and methods but additionally conform to the standards of the consortia assembly to allow interoperability of research data and tools between multiple consortia. One such consortium is a group of 5 universities in the state Nord Reihn Westphalen in Germany. The consortium "is based on the ideas of the European Open Science Cloud (Mons Tochtermann, 2016) and represents a development for the participating universities towards the National Research Data Infrastructure planned for Germany" [29].

This shows how the NFDI in itself has many consortia that are based on the ideas pioneered by the EOSC and how the German NFDI is not separate from the EOSC but an integral part of it.

2.4 The future of Open Cloud

In a world more interconnected than ever before, Open Science will play a key role going forward. The importance of Open Science is already backed by many success stories from multiple different fields, from medical research[13] to open source software [15] or even open humanities [31].

Open Science creates a synergy, by making the data and the means available to more researchers all around the globe, the results are not only faster but potentially more impactful. This impactfulness comes with the cost of restructuring the current research paradigm in terms of funding, organization and annotating your data results with metadata and making it machine actionable. As stated in [21] "the FAIR principles are simple, but implementation is not" The general consensus is that the benefits far outweigh the costs which is why Open Science initiatives are more prevalent than ever before. When it comes to the infrastructures and organizations enabling Open Science, what is the next step?

During the 40th session of the UNESCO General Conference held in Paris in 2019, the state members agreed with the organization for the creation of Open Science Standards[32], for these to be adopted by said state members starting on 2021.

Also in the same year (2019) at the COData Beijing Conference the concept of the Global Open Science Cloud (GOSC)[14] was introduced by the International Science Council. The goals for this GOSC initiative is not only to facilitate further cooperation, but ultimately to enhance standardization and to ensure interoperability between all Open Science Clouds.

The evolution of Open Cloud is to grow beyond its current regional, institutional or national borders, to enable global cooperation for the scientific community as well as adhering to the FAIR principles.

3 Discussion and Conclusions

The following section contains a discussion on the findings and summarizes the findings of this research.

3.1 Discussion

The aim of the research was to achieve an overview on the status of the European Science Cloud initiative and its differences and similarities with other open cloud projects oriented to scientific research.

The main challenges faced were the short history of these initiatives and the lack of scientific literature surrounding these. Whilst the conceptual idea is not that new, the first fully operating platform has only been launched in recent 2016. Some of the Open Science Cloud initiatives discussed in this research have only been launched months ago. Due to the short live time of these projects, their usage and the documentation of their impact on scientific research is very limited.

Therefore the scientific literature and papers published that made use of or review most of these platforms is almost non-existent. The most reliable information regarding these Open Science platforms can only be found on the official web-pages or documentations of the bodies or commissions backing said platforms and some isolated articles. These sources in most cases do not provide a critical view on the initiatives and the lack of sources make the actual assessment of the platforms challenging.

Furthermore we would like to mention that with this paper we took the FAIR principles as firstly described in regards to software and data. Whilst recent developments with the FAIR principles look to tackle the implementation challenge, not only for software but including the infrastructure and services. [20]

It is important to note that due to these initiatives being overall in a very early stage of production, the changes, updates and releases of new versions are frequent as the field and discoveries are evolving. It is likely that this overview becomes quickly obsolete. The idea of EOSC as a pioneer however remains relevant.

3.2 Conclusion

Open Science initiatives that aim to provide technical resources to researchers are becoming a reality all around the world, they are generally referred to as "Open Science Clouds". Besides access to services surrounding data, such as storage, transportation and sharing, they generally also provide computing capabilities for high volumes of numeric data.

The common denominator of these initiatives is the focus on enabling scientific research through collaborative projects. They all ascribe to FAIR principles of data and aim to foster an open and collaborative environment for more impactful results.

The European Open Science Cloud impulsed by the European commission has been taken as the reference project, since it is the first one being in full production and already providing a cross-national collaborative environment for research. Similarly, other similar projects are also backed by their respective national or regional governments.

Relating back to our research question, we conclude that EOSC seems to be leading in regards to level of maturity and usage, although hard conclusions on this are difficult to make with the lack of documentation and other outside reviews. Furthermore we found that EOSC has propagated good practices and standards which do get followed by other Open Science initiatives, and which are still under further development.

References

[1] [n.d.]. About EOSC. https://eosc-portal.eu/about/eosc. Accessed: 2021-06-04.

[2] [n.d.]. About the EOSC Portal. https://eosc-portal.eu/about-eosc-portal. Accessed: 2021-06-04.

[3] [n.d.]. EU+ Countries. https://eosc-portal.eu/policy/EU-Countries. Accessed: 2021-06-04.

[4] [n.d.]. Horizon 2020. https://ec.europa.eu/programmes/horizon2020/en/what-horizon-2020. Accessed: 2021-06-04.

[5] [n.d.]. Open Science overview in Europe. https://www.openaire.eu/os-eu-countries. Accessed: 2021-06-04.

[6] 2015. A Digital Single Market Strategy for Europe. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52015DC0192. Accessed: 2021-06-04.

[7] 2016. European Cloud Initiative - Building a competitive data and knowledge economy in Europe. https://eur-lex.europa.eu/legal-content/en/TXT/?uri=CELEX%3A52016DC0178. Accessed: 2021-06-04.

[8] Australian Research Data Commons (ARDC). [n.d.]. Australian Research Data Commons (ARDC). https://ardc.edu.au/about_us/. Accessed: 2021-05-31.

[9] Paul Ayris, Jean-Yves Berthou, Rachel Bruce, Stefanie Lindstaedt, Anna Monreale, Barend Mons, Yasuhiro Murayama, Caj Södergård, Klaus Tochtermann, and Ross Wilkinson. 2016. Realising the european open science cloud. (2016).

[10] Michelle Barker, Ross Wilkinson, and Andrew Treloar. 2019. The Australian Research Data Commons. *Data Science Journal* 18, 1 (2019).

[11] NDRIO board of directors. 2020. NDRIO annual report 2019-2020. https://engagedri.ca/about/what-we-do. Accessed: 2021-06-01.

[12] Jean-Claude Burgelman. 2021. Politics and Open Science: How the European Open Science Cloud Became Reality (the Untold Story). *Data Intelligence* 3, 1 (2021), 5–19.

[13] Francis S Collins, Michael Morgan, and Aristides Patrinos. 2003. The Human Genome Project: lessons from large-scale biology. *Science* 300, 5617 (2003), 286–290.

[14] International Science Council. [n.d.]. Global Open Science Cloud. https://codata.org/initiatives/strategic-programme/global-open-science-cloud/. Accessed: 2021-05-23.

[15] Brian Fitzgerald. 2006. The transformation of open source software. *MIS quarterly* (2006), 587–598.

[16] DFG Deutsche Forschungsgemeinschaft. [n.d.]. NFDI National Research data Infrastructure. https://www.dfg.de/en/research_funding/programmes/nfdi/index.html. Accessed: 2021-05-23.

[17] Robert L. Grossman, Matthew Greenway, Allison P. Heath, Ray Powell, Rafael D. Suarez, Walt Wells, Kevin White, Malcolm Atkinson, Iraklis Klampanos, Heidi L. Alvarez, Christine Harvey, and Joe J. Mambretti. 2012. The Design of a Community Science Cloud: The Open Science Data Cloud Perspective. In *2012 SC Companion: High Performance Computing, Networking Storage and Analysis*. 1051–1057. https://doi.org/10.1109/SC.Companion.2012.127

[18] Robert L Grossman, Yunhong Gu, Joe Mambretti, Michal Sabala, Alex Szalay, and Kevin White. 2010. An overview of the open science data cloud. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. 377–384.

[19] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan. 2015. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems* 47 (2015), 98–115. https://doi.org/10.1016/j.is.2014.07.006

[20] Annika Jacobsen, Ricardo de Miranda Azevedo, Nick Juty, Dominique Batista, Simon Coles, Ronald Cornet, Mélanie Courtot, Mercè Crosas, Michel Dumontier, Chris T Evelo, et al. 2020. FAIR Principles: interpretations and implementation considerations.

[21] Varsha Khodiyar, Heidi Laine, David O'Brien, Raul Rodriguez-Esteban, Yasemin Türkyilmaz-van der Velden, Grace Baynes, Matthew Brack, Anne Cambon-Thomsen, David Carr, Elisa Carrus, and et al. 2021. Research Data: The Future of FAIR White paper. https://doi.org/10.6084/m9.figshare.14393552.v1

[22] Jun Li, Jingjing Li, and Shanshan Shi. 2020. China Science and Technology Cloud Situation and Prospects. In *China's e-Science Blue Book 2018*. Springer, 155–169.

[23] Peter Mell, Tim Grance, et al. 2011. The NIST definition of cloud computing. (2011).

[24] Barend Mons, Klaus Tochtermann, et al. 2016. Realising the European open science cloud. First report and recommendations of the commission high level expert group on the European open science cloud.

[25] MOSP. [n.d.]. Malaysian Open Science Platform. https://www.akademisains.gov.my/mosp/about/what-is-malaysia-open-science-platform/. Accessed: 2021-05-31.

[26] Chinese Academy of Sciences. [n.d.]. China Science and Technology Cloud. https://www.cstcloud.net/cstcloud.htm. Accessed: 2021-05-31.

[27] Chinese Academy of Sciences. [n.d.]. CSTCloud Use Cases. https://www.cstcloud.net/ucintro.htm. Accessed: 2021-06-01.

[28] Ruth Pordes, Don Petravick, Bill Kramer, Doug Olson, Miron Livny, Alain Roy, Paul Avery, Kent Blackburn, Torre Wenaus, Frank Würthwein, et al. 2007. The open science grid. In *Journal of Physics: Conference Series*, Vol. 78. IOP Publishing, 012057.

[29] Anne Thoring Raimund Vog, Dominik Rudolph. 2019. *Bringing Structure to Research Data Management Through a Pervasive, Scalable and Sustainable Research Data Infrastructure*. Springer, Cham.

[30] RedCLARA. [n.d.]. LA Referencia:Latin America Open Science. `http://www.lareferencia.info/en/`. Accessed: 2021-05-31.

[31] Peter Suber. 2005. Promoting open access in the humanities. *Syllecta Classica* 16, 1 (2005), 231–246.

[32] UNESCO. [n.d.]. UNESCO Recommendation on Open Science. `https://en.unesco.org/science-sustainable-future/open-science/recommendation`. Accessed: 2021-05-23.

[33] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 1 (2016), 1–9.

Web Services and Cloud-Based Systems Literature Study IDaaS (Identity as a Service)

William Ford
Department of Computer Science
Vrije Universiteit Amsterdam
w.a.ford@student.vu.nl
Student-ID: 2712009

Gyan de Haan
Department of Computer Science
Vrije Universiteit Amsterdam
g.de.haan@student.vu.nl
Student-ID: 2715795

Max Raams
Department of Computer Science
Vrije Universiteit Amsterdam
m.raams@student.vu.nl
Student-ID: 2711022

Tim Vorstenbosch
Department of Business Analytics
Vrije Universiteit Amsterdam
t.vorstenbosch@student.vu.nl
Student-ID: 2588989

Abstract

This literature review explores the subject of identity as a service (IDaaS), the proposed answer to the growing demand for improved security in the cloud. The main research question being: "how do IDaaS systems handle security currently and how resilient are these architectures for future business applications". The origin of IDaaS is discussed in the form of a brief summary of cloud computing. Various security properties are introduced and other services and platforms corresponding to cloud computing services are explored. Finally, the available literature is reviewed and the resiliency of IDaaS for future business applications is evaluated. Our conclusion is that the research into IDaaS is still relatively new, that various issues still remain unsolved and that it is recommended for companies to wait a few years with the implementation of IDaaS.

1 Introduction

Cloud computing is a relatively new yet efficient and effective technology that has taken over various traditional in-house systems [1]. Additionally, a huge demand for improved security and flexibility of current control mechanisms has emerged recently, specifically for the identification of cloud users [2]. The solution to this growing demand is a new resource called IDaaS (Identity as a Service), which is a system that is able to provide security measures for companies through external providers and enables various on-demand service providers for a company's consumers [3]. Internet giants such as Google and Facebook created the possibility for such a service, due to the large-scale private Wide Area Networks (WANs) distributed throughout the continents [4]. According to various sources, this new system has numerous advantages over the previous adaptations, such as an automated trust negotiation between cloud users to enforce their privacies [3], various algorithms to efficiently handle bandwidth reservations [4], and it can provide platform-specific security infrastructure at runtime for given business scenarios [2].

Despite the numerous advantages and conveniences of the new cloud resource, it is apparent that there are numerous gaps in this complex cloud-based environment [5]. Managing such complex systems in a secure way is one of the most challenging aspects of the new system [6]. Consequently, numerous papers and even books have been written to identify, improve or fix the abundance of gaps within the current system [5] [7] [8]. Although these papers provide solutions and alternative approaches to these problems, few papers have been written on studying the overall security of IDaaS. Therefore, this literature review will discuss how IDaaS systems currently deal with security and

privacy via web services and will analyze how resilient and secure these architectures are for future business applications. The review will start with investigating some general security properties of IDaaS, as well as doing some exploration of various services and platforms corresponding to cloud computing services. Next, all of these parts will be analyzed and discussed in the Discussion section. Additionally, the resiliency of IDaaS will be discussed based on the analyses of the aforementioned properties and services. The review will end with a small conclusions of our analysis.

2 Background

To get an idea of what Identity as a Service (IDaaS) represents, it is meaningful to review the history of Cloud Computing, since this is an integral part of the relatively new service concept. One of the earliest concepts of cloud computing came in the form of *time-sharing*, which became prominent through the usage of the Remote Job Entry procedure. This procedure was adopted by notable IT businesses such as IBM and DEC [9] and was first described by John Backus in 1954 [10]. Despite being the first person to describe such a procedure, he is more famously known for being leader of the team that implemented the language of Fortran. The concept of time sharing can be interpreted in two ways, namely multiprogramming and computer efficiency [11]. Multiprogramming is oriented towards hardware efficiency to fully utilize all external components of the computer, while the other variant focuses more on the usage of computers themselves [11].

The latter definition is viewed as a basis for various other ideas that appeared decades later, such as the Virtual Private Network (VPN) and consequently one of the first web services known as Amazon Web Services in July 2002 [12]. The same company proceeded to create more cloud-based systems in 2006, such as the Simple Storage Service (S3) and Elastic Compute Cloud (EC2), which helped to popularize the idea of cloud computing to the rest of the world [13] [14] and are categorized as a Infrastructure as a Service product [15]. By means of an increasing popularity in the paradigm of cloud computing, various other network services started to emerge [16].

An interpretation of such a service is the idea of offering a hosted set of software and hardware that the receiving party does not own themselves and for which the receiving party has to pay [16]. The first service that emerged from this increasing demand in cloud computing is the Infrastructure as a Service concept, which is generally considered the basis for all the other services. It offers a variety of functionalities, which includes hosting, hardware provisioning, storage and provisioning of networks [17]. Platform as a Service (PaaS) is the second service the emerged from the concept of IaaS and it is able to deploy a set of software applications with a corresponding computing platform [18]. In addition to these services, it is also possible to exclusively provide software applications to customers over a network, which is also referred to as Software as a Service (SaaS) [18].

Despite the increasing interest in these new services, various papers have been written on the potential benefits, risks and issues of these new products and technologies, which can concern both technical and social aspects [19] [20]. One aspect in particular that still causes major dilemmas is security. One could argue that this is the most relevant and crucial factor to consider when dealing with cloud computing concepts and techniques [21]. Handling security issues becomes much more complex and risky due to the outsourcing of services to third parties [17]. There are numerous factors that contribute to the idea of a secure system for both the provider and customer, which includes the "availability, integrity, authenticity, confidentiality and privacy" of the cloud systems [22], which makes this aspect of a system particularly difficult to manage.

Identity as a Service (IDaaS) is a variation of the aforementioned services. It tasks itself with providing the receiving party with an identity management system over the network. It most likely originates from the idea of identity management systems being implemented on the network. The goal of identity management systems is to successfully manage identities within specified domains while decreasing costs, downtime and repetitive tasks for the system [23]. This definition of IDaaS is not to be confused with other definitions, such as in the paper by Li et al., where this term is defined as a "Inter-datacenter network as a service" [4].

Although Identity as a Service is a relatively new concept, there is an abundance of traction towards such an Identity-Centric Internet paradigm. Such a paradigm is able to provide businesses with an innovative and long term solution for the issue of dealing with digital identities [24]. The increasing number of digital identities and lack of interoperability have impeded the task of successfully managing identity on a domain basis [25]. Additionally, service providers demand that such identity

mechanisms are extremely well protected while also being flexible [2]. Therefore, it is no coincidence that there is an increasing demand for such a service, yet managing such identification services remains an arduous task [6].

In comparison to the other services, the significance of security becomes even more prevalent for IDaaS specifically, since identification systems are crucial for the protection of confidential data. Therefore, various literature has been written on finding and improving this current architecture [5] [8] [6] [7]. However, few studies have analyzed the level of security for such given systems. Therefore, this paper makes an attempt to provide a study on the level of protection provided by recent IDaaS systems, which will be further discussed and explored in the following sections.

3 Meta Analysis

The literature study was performed by searching for documents in the SCOPUS database [26], with the following search query: TITLE-ABS-KEY ((idaas OR identity AND as AND a AND service) AND cloud AND (security OR privacy)) This resulted in a total of 729 documents. The meta analysis can of these documents can be viewed in Figure 1, 2, 3, 4, 5. These figures show the distribution of the document by area, by author, by year, by type en by source respectively. Based on this meta analysis it is clear that the research field is relatively new. The documents have been published since 2009, with a increased interest and number of publications over the years. The majority of the documents have been published in conference papers or articles, with the leading portion being in the computer science related sources. The distribution of document by author show there are "experts" in this field that have published more than 5 documents. This meta analysis does not show collaborations, although these have been observed among the top 10 authors by documents amount.

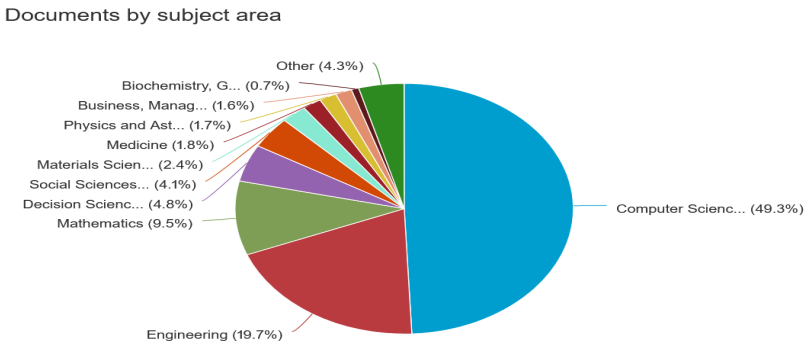


Figure 1: Plot of documents by area [27]

4 IDaaS Security Properties

As was mentioned in the Background section, security is a relatively broad term which is often divided into various distinct parts. For this paper, the security properties defined by J. J. Stapleton [28] will be used to divide the topic, namely: confidentiality, integrity, availability, authentication, and privacy. This division was chosen, since it comprehensively incorporates the most critical aspects for a secure identification management system by combining the properties of the well-known CIA and AAA models [29] [30]. Besides that, these properties are often regarded as the pillars of the principle of “Security without Obscurity” [28], meaning that knowing how the system works should not have an impact on its security [31]. The following subsections will each discuss one of the security properties. A definition will be given and an analysis will be performed on these aspects using various corresponding papers.

4.1 Confidentiality

One of the first important properties of a secure system is its confidentiality. An unauthorized party should never be able to access sensitive data that is stored within the system. Since cloud-based

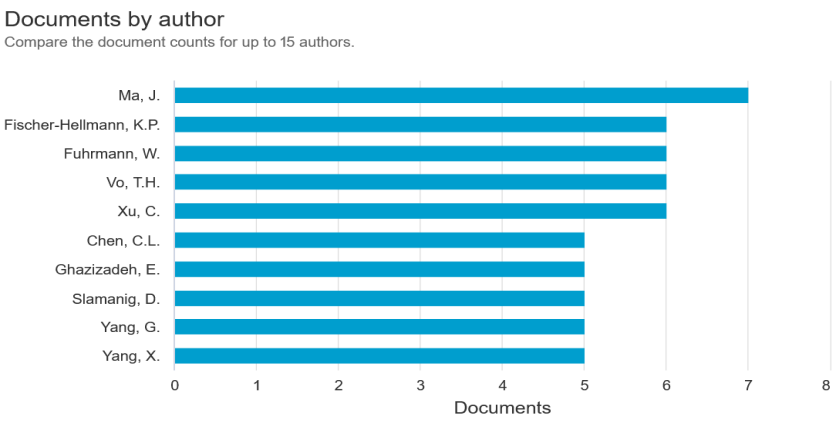


Figure 2: Plot of documents by author [27]

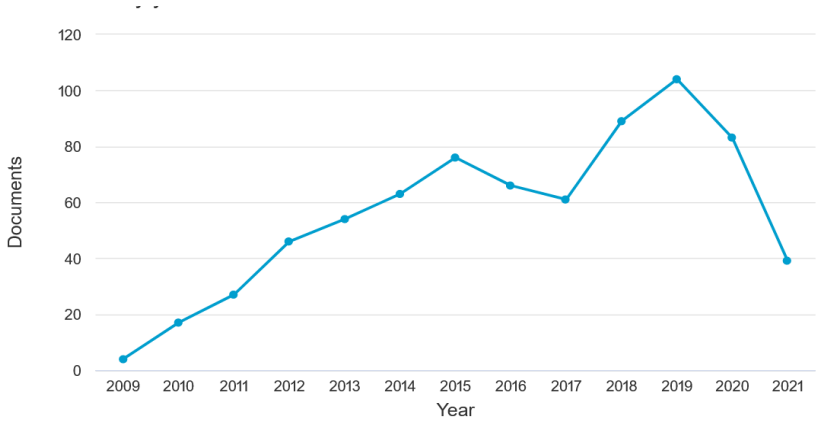


Figure 3: Plot of documents by year [27]

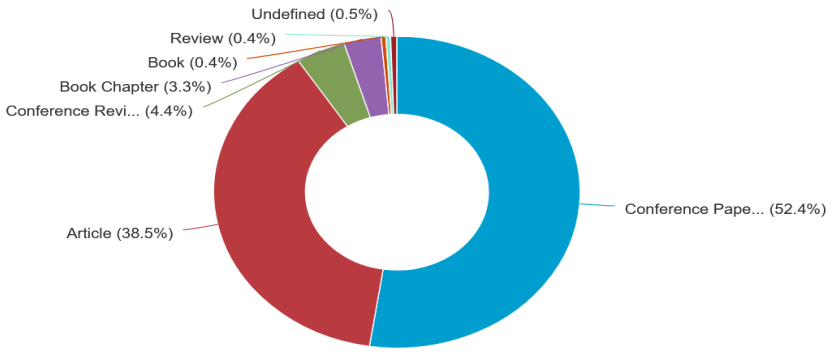


Figure 4: Plot of documents by type [27]

architectures are based on a business model with shared resources, it becomes essential to isolate data partitions on a virtual level. [32].

Since Identity as a Service operates for a large part on personal data, this principle is extremely important. The paper by Hoang Vo [2] defines this data as Personal Identifiable Information (PII) and proposes a purpose-based encryption to protect the disclosure of PII from intermediary entities and untrusted hosts. Purpose-based encryption is an approach to combine purpose-based access

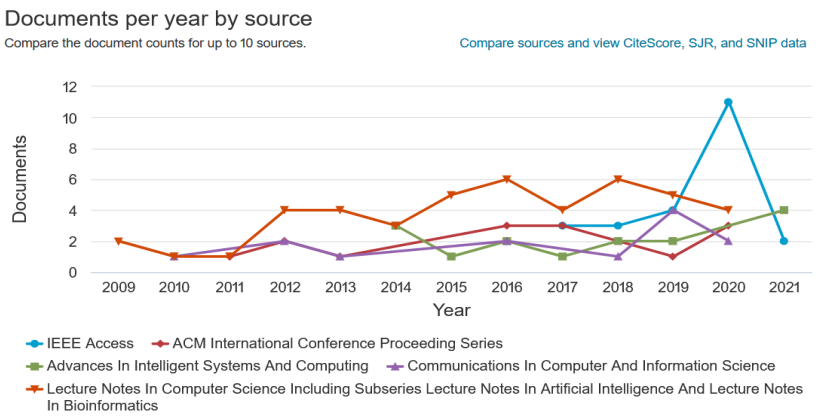


Figure 5: Plot of documents by source [27]

control and attribute-based encryption. It thereby protects the confidentiality of distributed data among multiple parties.

When certain Personal Identifiable Information needs to be encrypted it will follow a *disclosure policy*. This policy is based on three factors:

- Domains: which domains are allowed to decrypt the PII (e.g. Facebook, Salesforce)
- Time: within what time period can the PII be decrypted (e.g. 14 days)
- Purposes: what is the purpose of the service (e.g. marketing, purchase)

When this encryption is finished, the PII can safely be distributed. Only the services that are hosted within the specified domains gain the ability to decrypt it within the given period and only if they satisfy the intended purposes that were specified during the disclosure policy.

4.2 Integrity

As the use of time-sharing systems started to grow, it became clear that data needed to have some form of protection [10]. If a system is vulnerable to the loss of accuracy of data, unintended behavior or even system crashes can happen. As described by Browne in 1960 [33], there must be a protection system in order to ensure the *data integrity*; data needs to be protected from damage or inconsistency at any time.

Later on, this integrity requirement not only became important to prevent system faults, but also to build secure systems [34]. An unauthorized party should never be able to create false information, nor be able to modify or delete sensitive information. This applies for data that is stored, but also for data that is in transit. Especially for transit, precautions must be taken to ensure that data can never undetectably be changed.

To prevent this from happening, the Identity as a Service architecture makes sure all data is properly encrypted. Only Service Providers that satisfy the disclosure policy may receive the key to decrypt the data. When a malicious server tries to change data or attempts to bypass disclosure policies, the decryption will simply fail and the process will halt [2]. To prevent data from being modified while in transit, a transport layer security mechanism is used (e.g. SSL) [5].

4.3 Availability

Another important security property is the availability. Availability refers to the uptime of a system that provides a service during its usage. Since Identity as a Service is a cloud-based architecture, the long term benefits of achieving high availability can be achieved by establishing robust monitoring and management tools and practices [35].

One basic method to achieve high availability is to avoid a single point of failure as much as possible. The use of cloud servers will handle that. When a virtual machine instance crashes, it can instantly

be restarted or even recreated on another server. These cloud servers can also be geographically distributed, which also strengthens the availability of the service.

The use of cloud servers will also provide high scalability, since the number of cloud servers can be dynamically scaled horizontally. When the network traffic suddenly increases, several instances of virtual machines can be added to a cluster. This helps Identity as a Service in providing high availability to its end users.

4.4 Authentication

Authentication plays a central role in nearly all previously mentioned security properties. A service cannot guarantee confidentiality, nor integrity if it does not verify the identity of the entity on the other side of the connection (henceforth referred to as the "user"). Besides verifying identity, this security property also concerns determining which resources a user is permitted to access and what actions the user can perform after successful authentication. This principle is called *access control* [36].

Typically, the cloud provider implements user authentication in a PaaS environment and picks the authentication method they see fit. These methods range from basic (e.g. specifying a token [37]) to advanced (e.g. keystroke analysis [38] [39]). Rolling out a system with poor authentication methods or protocols might make the service vulnerable to data breaches, account hijacking, abusive use, and the like [40]. It is therefore important to consider the different types of authentication and their strengths, vulnerabilities, and other challenges.

- **Private access tokens** and **username-password combinations** are currently the most common methods of authentication for cloud-based services. Users are familiar with the flow of working with them, which makes them excel on the front of usability. However, they remain subject to numerous basic security threats. Weak passwords may be stolen in a data breach or brute forced in seconds. Strong passwords may be intercepted in data traffic, phished, or obtained otherwise and used fraudulently [41]. The implementation of password and token timeouts might provide a solution to this problem, requiring users to change their password frequently or automatically refreshing tokens regularly. However, employing this procedure runs the risk of a large decrease in usability and encouraging users to adopt sequences within their passwords (e.g. 123). These sequences may severely increase the system's vulnerability to password guessing attacks [42].
- **Federated authentication** (also known as **Single Sign On (SSO)**) uses a central authentication server to let a user authenticate themselves once, after which they can make use of several services. This approach reduces the number of passwords a user needs to remember, which allows the user to make them more complex and secure. Since this single password can now give access to multiple services, it is crucial that the authentication server is made as secure as possible [43].
- **Multi-factor authentication** can be defined as authentication using a combination of two or more of the following factors: the *knowledge factor* (exclusive knowledge of secret information, e.g. a password or PIN); the *possession factor* (physical entities only possessed by the authorized user, e.g. their phone or bank card); and the *inherence factor* (metrics intrinsically owned by only the authorized user, e.g. their fingerprint or face) [44]. Multi-factor authentication relies on the separation of these components. Violations of this separation are writing one's PIN on their bank card [45] or sending text message to a phone that is synchronized to a PC web browser. These violations may lead to synchronization vulnerabilities: usability features that deliberately blur the boundaries between factors, weakening the security guarantees of multi-factor authentication in the process [46].
- **Biometric authentication methods** (e.g. fingerprint scanning, iris scanning, voice printing, facial recognition, hand geometry scanning, and vein recognition [47]) may be applied to employ a higher level of data security in the cloud [41]. While passwords or PINs may be forgotten, shared, or leaked, and cards or keys may be lost, duplicated, or stolen, biometric data is highly difficult forge, share, or steal [45], which makes it a strong contender for increasing security in cloud-based authentication systems. One of the major critiques of the various methods of biometric authentication is the privacy risk associated with the storage

of biometric data. The data stored in these images is considered sensitive and should be protected with systems that are very secure in order to ensure end-user privacy [45].

- A number of studies suggest the application of **keystroke analysis**, the analysis of keystroke dynamics, to support the authentication process [38] [39]. Keystroke dynamics are the idiosyncratic patterns and rhythms a user exhibits while typing. They are part of a subset of biometrics termed the *behavioral biometrics*, together with other methods such as handwritten signatures and mouse use characteristics [48]. These behavioral biometrics are much more reliable than one would initially expect. In the 19th century, telegraph operators could recognize each other solely based on their typing style [39]. Keystroke analysis may be applied in various ways, but for IDaaS, a static keystroke analysis would fit best. It can authenticate a user by a typing pattern based on a predetermined text (e.g. their username and password). This way, a service can authenticate a user by not only what they typed, but also how they typed it. [38].

4.5 Privacy

The CIA security objectives discussed above (confidentiality, integrity and authentication) influence privacy. These objectives, directly or indirectly, preserve the privacy of the cloud service users [36]. Users store personal identifiable information in the cloud environment so that cloud services may access and use it on demand. From the user's perspective, they want to use the functionalities offered by cloud providers, but at the same time they also want to preserve their privacy [3]. The safeguarding of privacy cannot be guaranteed. Even if a cloud service specifies and discloses its privacy policies, it remains possible that they will not follow their policies and may (accidentally) transfer user data to another party [3]. One also has to consider outside attacks. It is not possible to guarantee 100% protection from malicious hosts, malware or other forms of external attacks [2].

In the best case scenario, an identity management system reveals the least amount of user information to third parties, which are necessary in the participation of the transaction [3]. Privacy protection for identity roaming is given as one of the requirements for a good performing Identity management system [3]. Noticing that absolute protection does not exist, an efficient solution has to be sought that is compliant with the law, in particular, the General Data Protection Regulation (GDPR) [2].

One such effort is described in [2] to be Purpose-based Encryption. Which is described as "The first approach to combine Purpose-based Access Control and Attribute-based Encryption(ABE) to protect the confidentiality of disseminated data with multi-authorities support." In this way, the personal identifiable information of the user is protected in the frontend service, as well as the backend service. The IDaaS can only disperse the encrypted data in certain domains, with services in those domains getting access for a limited amount of time for a the previously specified purpose [2].

Traditional access control models, such as role-based access control, focus on which user is performing what action on a data point. Access control is limited to roles that have the same meaning and permissions [2]. However, heterogeneous systems may have different semantic and permissions about certain roles. Purpose-based access control (PBAC) can be a solution for a large distributed and heterogeneous environment [2].

A second approach to enabling a better privacy security in IDaaS solution is the use of Predicate Encryption (PE) [49]. With PE, data is encrypted on the server. If the data needs to be decrypted, a secret key needs to be obtained. If a predicate check is evaluated to true, the server can decrypt the data. PE has two subclasses: PE with public index and PE with private index. In public index, the index is visible to everyone, including the server. In private index, the server can learn about the result of the predicate but nothing else about the index [49].

5 Other Cloud Computing Services and Mobile Cloud Computing

Although the main focus of this report is on the security and resiliency of IDaaS, this does not render analyses of other services and platforms useless for this evaluation. As was discussed in the Background section, services such as IaaS, PaaS and SaaS can be seen as the ancestors of IDaaS and thus can experience similar issues. Therefore, we decided to also investigate these other services to look for equivalent problems and solutions and consequently gain an even better insight into the security properties of IDaaS. Additionally, it has also come to our attention that there has been a

surge of interest of cloud computing being used for mobile platforms through various papers [50] [51] [36]. Therefore, we find it essential to also touch upon this type of new service and platform. As a result, the following sections will study the security aspects of these other services and platforms of cloud computing, so that their conclusions can be taken into account for the security and resiliency of IDaaS.

5.1 Cloud Computing and Other Services

In one of the earliest papers on cloud computing services, Jensen and Gruschka [52] already noted various technical security issues related to cloud computing services. It mentions an example of such a service, which is one of the most popular IaaS services at the time, namely Amaon's Elastic Compute Cloud (EC2) [13], which is also discussed in our Background section. The attacks mentioned in this paper includes Malware Injection and flooding attacks, integrity and binding issues, issues with the Same Origin Policy of scripting languages and attacks on browser-based cloud authentication systems. For each of these sections, the authors elaborate on the issue itself and afterwards discuss potential impact for each of these problems. Additionally, the authors comment that deep analyses are required to solve these technical issues of the cloud computing paradigms and that their research is ungoing.

The paper by Almorsy et al. [53] uses Jensen and Grushka as one of their references for their article, in which the authors discuss various security issues for various perspectives on the cloud computing paradigm and tries to provide solutions for each of the security holes present. Examples of such perspectives include the architecture perspective, cloud characteristics perspective and the delivery service model perspective. A few of the security issues present in cloud computing models are service portability, vendor lock-in, Service Level Agreement (SLA) management and multi-tenacity, which seem to impact "creditability" and "pervasiveness" of these models, which is mentioned in their Introduction section. These characteristics are elaborated in section 4 of their paper, where the writers comment on the number of properties these various characteristics ought to have, such as isolation and location transparency for a secure multi-tenancy cloud computing model. Furthermore, the authors mention how the diversity of numerous cloud services, such as IaaS, PaaS and SaaS, complicates the development of a standardized security model.

Additionally, it observes that the public clouds are the most vulnerable deployment model, which includes the various services IaaS, PaaS and SaaS as mentioned in the Background section. More importantly, they also note the deep dependency stack of cloud computing services, where the security of the higher layers (applications and services/APIs), which complicates the solutions to security issues. Furthermore, it is noted that every stakeholder has their own security management systems or processes, which leads to distinct tenants of security requirements and thus no standard security specification notations. Therefore, one of their solutions is to support abstraction of the problem, so that different perspectives can be linked. Moreover, given the previous prerequisites for a secure cloud computing model, support needs to be given for multi-tenancy and integration/coordination with other security controls.

This same article also manages to dive deeper into the various issues per model in section 7, where mainly various issues of IaaS, PaaS and SaaS are discussed. The main security issue for IaaS resides within its use of Virtual Machines (VMs). One of the issues with VMs is that they are still security risks if they are offline. In addition, there is an increase in the likeliness of exploiting vulnerabilities in the DNS servers, since different tenants are utilized within one server. Hypervisiors also seem to be a recurring issue, since all the VM operations are traced unencrypted. PaaS security issues can be traced back to the Service-oriented Arcitecture (SOA) model. Since these models have been around for a relatively long time in the cloud computing paradigm [54] [55], it is evident that this system is still victim to various DOS, Man-in-the-middle Replay, Code Injection and various other attacks [52] [56]. SaaS inherits all of the issues mentioned previously as in builds upon these models, although the article observes that web applications contain the most critical vulnerabilities as cited by OWASP [57]. Moreover, the importance of security for cloud management and access methods is perceived , since a vulnerability or breech can have devastating effects and various enablers of security cloud computing are mentioned as a result to help prevent such breeches in section 8.

Kumar et al. in 2017 [15] comment on the escalation of security issues through the accumulation of digital assets with a reference to Khalil et al. [58]. Additionally, it shows in the introduction section how vulnerable cloud security has become due to various "architectural foundation-elements", which

includes concepts like virtualization and the Service Level Agreement. Next, the authors discuss various security issues in more detail with the three services mentioned previously. One such notable issue is the security problems emerging from existing applications and runtime, which remains a challenging issue across the three service models. A few possible threads of this includes Command injection attacks, Cross-Site Scripting (XSS)/request forgery (XSRF) and Cookie poisoning among many others. A second important security thread originates from the data stack, since this information is not stored anymore in enterprise computing and thus various measures need to be created to prevent harm to this data. There are several components to this aspect, such as CIA triad (confidentiality, integrity, availability), triple A components (Authentication, Authorization and Accounting) and broken access control (lack of restriction on users). Other parts of the system where protection of the service is undermined includes the middleware stack, operating system stack, virtualization stack, server stack, storage stack and network stack. In the conclusion, the authors remark even newer developments of services, such as Container as a Service (CaaS), Software-defined networking and the Cloud of Things (CoT).

5.2 Mobile Cloud Computing

Mobile Cloud Computing (MCC) is a combination of mobile computing, cloud computing and wireless technology and is mainly used for data storage and processing for a wide area of mobile users [36], The main difference between Mobile Cloud Computing and normal cloud computing systems is the extensive use of mobile devices [6], which implies that both are relatively similar and are intertwined. Due to this similarity, these systems could possibly have also the same security issues and solutions. Although the concept of Mobile Cloud Computing (MCC) is not new, it has gained much traction in the last decade [51]. Due to this increase in recognition of Mobile Cloud Computing platforms, a literature review has been written on this topic by Khan et al. [51].

The goal of this review was to highlight current state of the art infrastructures for mobile cloud computing, create a taxonomy related to these infrastructures and identify potential problems with this new technology. Examples of these potential security issues includes "data replication, consistency, limited scalability, unreliability, unreliable availability of cloud resources, portability, trust, security, and privacy", as was mentioned by Zissis et al. [59]. Additionally, Khan et al. discusses the challenges of adopting cloud computing for mobile platforms by mentioning various papers that have performed surveys and concluded that numerous companies did not want to adopt cloud computing systems due to the security issues of this relatively new technology [60] [61]. More recent papers, such as by Naik and Jenkins [6], also discuss the complexity of the authentication and authorization tasks.

However, the literature review also mentions that much work has gone into the research of security related topics for such cloud computing environments, which is also noted by Mollah et al. [36]. The paper by Khan aims to evaluate various frameworks in section 3 using specific evaluation parameters from section 2. These definitions are similar to those mentioned in the IDaaS Security Properties section. Naik and Jenkins also added to these parameters by proposing and updating the criteria for an effective Identity and Access Management system [6]. The most crucial examples of these criteria are Extensive Authentication and Authorization support, Data Integrity/Confidentiality and support for consumers, enterprises and sign-on approaches. Despite the abundance of work, review by Khan et al. concludes that there is still much work to be done on this particular aspect of Mobile Cloud Computing.

An article by Gao et al. [50] in this same year likewise identifies various issues and challenges for Mobile Cloud Computing. Unlike the previous review, it also exemplifies many advantages of MCC for both users and clients, such as computation, storage and energy efficiency in the Introduction. Moreover, multiple motivations for choosing MCC are given in section 2.2 to support the positive aspects of MCC. On the other hand, it discusses various models related to MCC and divides them into three different generations and mentions numerous limitations for each of the generations. Furthermore, it mentions various issues with the concept of MCC, such as privacy and mobility and describes these in much detail. Thus, this paper also arrives at a similar conclusion to Khan et al.

The more recent survey by Mollah et al. [36] reached a similar conclusion in 2017, where various security and privacy issues are still present in MCC. It also mentions how new difficulties have emerged due to the usage of new technologies and due to the restrictions presented by mobile devices. Additionally, several other applications of cloud computing for mobile devices have emerged according to Mollah et al., such as cloud storage, data sharing, gaming and cloud assisted Internet of

Things, among others, in the Introduction section. The article also defines various possible services for mobile platforms, which falls under two categories: adaptation of existing services and new services. An example of an adaptation of existing services is Mobile Cloud Infrastructure as a Service (MIaaS), while a new example of a service is Mobile App as a Service (MAppaaS). They continue with mentioning various security challenges for MCC, which include data security, security for partitioning and offloading, security for virtualization, mobile cloud applications security challenges, challenges for mobile devices and privacy issues.

6 Discussion

In this section, all of the previously discussed papers will be analyzed. This analysis is first done for each individual paper and afterwards will be discussed in the last section. Each separate paper will be indicated by a **bold** styling at the start of the paragraph. For each individual paper, the following aspects will be discussed: positive aspects, negative aspects and relevancy.

6.1 Analysis IDaaS Security Papers

Zissis et al. [32] This article provides a research about the Confidentiality, Integrity and Availability (CIA) principles in cloud computing. It therefore first gives an extensive overview on how cloud computing starting to develop over the years and thereby explains a lot of key concepts. A lot of information is given before covering the actual CIA principles, and after that the paper suddenly becomes very technical.

S. Browne. [33] Despite the fact that this paper is written in one of the early stages of computer development, it covers some useful concepts. It reminds one to take privacy and integrity of data very serious and thereby gives various arguments. The article tries to convince the reader to strive for data protection and sometimes make trade offs. However, the article doesn't provide any solutions or technical examples.

Deswarte et al. [34] This article elaborates on how the CIA principles can be used to prevent intrusions in distributed systems. It first introduces the intrusion tolerance concept and why this is useful for distributed systems. Then it covers the technical implementation by giving an example of such an intrusion tolerant system. Since this article is from a long time ago it doesn't cover cloud-based architectures, but this article can still be a good foundation to set up such systems.

Ahuja et al. [35] This article defines availability and describes why this is important for cloud services. It therefore gives several examples and shows how this is implemented in common known service providers. The article concludes with some useful guidelines on how availability can be improved. The conclusion also says that particular areas still need to be further explored, but the article misses some elaboration on these areas.

Konoth, van der Veen [46] This master thesis by VU Amsterdam computer scientists explores a vulnerability in multi-factor authentication. This vulnerability involves the juxtaposition of multi-factor authentication relying on the separation of factors on one hand, and the desire of modern application vendors to offer synchronization between their web and mobile apps on the other. The paper identifies a new class of vulnerabilities: 2FA synchronization vulnerabilities. It then goes on to support its claims by demonstrating a real-life attack.

Masala et al. [41] and **Ratha et al. [45]** These articles provide a comprehensive overview of respectively how biometric authentication works in the context of cloud computing, and how to enhance security and privacy regulations in these systems. The former article first gives some formal definitions of what cloud platforms entail. This is followed by the "example cloud platform", which reads as a step-by-step guide on how to implement biometric authentication in cloud-based environments. The latter article goes deeper into the security and privacy concerns regarding biometric authentication, discusses many possible attacks, and shows how to secure the system against these attacks. Due to the surface-level nature of the analyses of biometric authentication systems this literature review, for both articles, the relevancy to this paper lies mostly in the introduction.

Dowland et al. [38] and **Zhong et al. [39]** These two articles from 2002 and 2012 respectively outline the possibilities and benefits when it comes to keystroke analysis as an authentication method. The former article mostly outlines the different types of keystroke analysis (e.g. static, periodic,

continuous) and some possible keystroke metrics which may be profiled (e.g. typing rate, latency). The latter article does a slight review of the relevant literature and proposes a new metric in an attempt to lower the overall error rate of the authentication method. Similar to the analyses of biometric authentication methods in this literature review, the relevancy of these articles lies mostly in the introduction, where basic definitions and fun facts are stated.

Tri Hoang Vo, K.P. Fischer-Hellmann [3] In this article the notation of IDaaS as a successor of more traditional Identity Management systems is addressed. Relevant scenarios are given that highlight the use of IDaaS and important components are mentioned of the IDaaS design. There is a small focus on privacy and the privacy laws, but there is no mention of the challenges that come with implementing this IDaaS design.

Tri Hoang Vo, K.P. Fischer-Hellmann [49] In this article from Tri Hoang Vo the notion of a Purpose based Access control in combination with predicate encryption is explained and implemented. The focus is rather on a technical implementation side of the concept mentioned above and less on the architectural justification. It would have been nice to see a more expanded section of the evaluation of the implementation and certain criteria for this evaluation.

Tri Hoang Vo et al. [2] This article is an extension on the paper that Tri Hoang Vo et al. published the year before [49]. In this article they strengthen the concept of a security infrastructure for the cloud environment. The focus of this article is on trust adaptation and purpose based Encryption. In contrast to the paper the year before it also covers the compliance with the General Data Protection Regulation (GDPR). This really gives an added value to this paper since GDPR becomes more and more important. The article gives an architectural insight and overview of the adaptations needed for better privacy in the cloud environment.

Conclusion

Many studies have been done regarding following the CIA principles, and through the years there has been a shift in the purpose of following these principles. In the early stages of computer development, the reviewed papers mentioned these principles as a way to prevent errors in systems. As computer development progressed, these principles became more relevant to prevent malicious intrusions within systems. As cloud computing started to rise in popularity, these principles became the foundation of how to build secure cloud-based systems that also take the privacy of its end-users into account.

Biometric and behavioral biometric authentication methods may need to become more prevalent to ensure sufficient security for cloud providers in the future. The approach of relying solely on user-password combinations or private access tokens is too vulnerability prone to meet the security requirements of IDaaS. Most modern cloud providers currently provide an option to their users to enable multi-factor authentication. However, many of these current cloud systems still work with two-factor authentication based on text messages to validate the possession factor, even though text messages are notoriously insecure.

The reviewed papers show that there is a large overlap between privacy and the CIA terms. This highlights the complexity of the concept of privacy and how full data protection cannot be achieved. The proposed design solutions in the papers are interesting, but they sometimes fail in mentioning the implementation challenges. The papers do highlight the importance of seeing privacy as one of the main concepts to take into account and expend on the dangers when this is not upheld.

6.2 Analysis MCC

Jensen and Gruschka [52] This is one of the earlier articles that manages to observe various issues with the cloud computing services. They mention how innovative systems, such as EC2 by Amazon, have numerous security and trust issues. One of the strongest aspects of the papers is that it is able to explain these issues on a technical level in such a way that non-experts are able to understand these technicalities relatively well. This is supported by the Foundation section, where all of the prerequisite definitions and terms are clarified and a foundation is created for the technical sessions. Additionally, despite already going relatively deep in detail for each of these technical issues, they acknowledge that these problem require a much deeper analysis and that this research is still ongoing. This is supported by the fact that their paper is still being mentioned by papers years later [53]. Therefore, they are still relevant in this research area and this also implies these issues are still not

fully resolved. However, this paper is not able to tackle all possible problems within these services and these other problems will be described by the next papers.

Almorsy et al. [53] Despite its short length, there is an abundance of information within the article by Almorsy et al., which is supported by the length of the Other Cloud Computing Services and Mobile Cloud Computing section. In contrast to Jensen and Gruschka, which targets the technical issues of these cloud computing services, this paper focuses less on the technical issues and more on the general overview, architecture and perspective on these services. This not only creates an interesting contrast, but also confirms the idea that there are still various security issues unresolved within the earliest cloud computing services. Specifically, the cloud security management system is still being investigated by the authors, which is related to the idea of IDaaS. This implies that the paper is relatively relevant for solving issues related to this new service. However, one of the weaker aspects of the paper is that it does not always support its arguments through literature, such as in Figure 3, where a dependency stack is shown without any reference to its origins.

Kumar et al. [15] Kumar et al. discusses the various security issues that have arisen from the new technologies within cloud computing, of which various issues are provided in the Other Cloud Computing Services and Mobile Cloud Computing. The paper not only depicts the multitude of issues present within the increasing number of services, but also describes them well and their components, as shown by the section on the data stack, and provides solutions for each encountered issue. Additionally, it acknowledges in the Introduction that security is becoming an escalating problem, as was also mentioned in section 5, which is mostly due to the emergence of even newer technologies, such as Container as a Service (CaaS) and Cloud of Things (CoT). This makes the paper extremely relevant by today's standards, since such applications are still extremely popular [62] [63] [64]. However, one aspect that the paper does not investigate are the open issues, which could help further identify the resilience of such service applications. Additionally, the contents of this paper are rather technical and thus some background knowledge is needed in order to fully comprehend their article.

Conclusion It is evident that, based on the previous three papers, various security issues within even the most basic cloud computing services still remain and that a thorough analysis is required to solve these issues. Additionally, all three papers seem to have similar ideas on the rise of new technologies within the sector of cloud computing, since all authors are skeptical about these technologies and acknowledge most of it is still a work in progress. Moreover, due to the rise of these new technologies, it becomes more difficult to create a standard security model for all of these services. It is also clear that not only technical issues can still persist in these systems, but also faults in the architecture, API or applications of these cloud computing services. Additionally, the authors consistently write on various recurring issues, such as issues with VMs for IaaS, XML attacks, Command injection attacks and numerous confidentiality/privacy violations. Therefore, these services are far from perfect and various security issues still remain within these models. Moreover, there does not seem to be any perfect solutions for solving all these issues currently, given that relatively recent papers still write on the same issues. It is thus alarming that more services are being developed in addition to IDaaS, while the main systems do not even appear to be that secure.

6.3 Analysis Mobile Cloud Computing

Khan et al. [51] This paper has provided the readers with an extensive literature review of specific aspects of Mobile Cloud Computing (MCC). Through this literature review it became clear how popular MCC had become in 2013 and the various security issues that have come along with this new development, which is consistent with the other papers in this section. The paper manages achieves its goal of identifying potential problems by clearly annotating what aspects will be covered and provides reasoning for this, which is shown in section 2 with various evaluation parameters. Additionally, each paper is described in detail, which ensures that their explanation is sufficient for understanding each of the reviewed papers. Furthermore, each paper is evaluated given the taxonomy in section 2 and consequently is able to be relatively critical of the papers. However, the review could have stated more clearly to what extend these security frameworks are "state of the art", as was mentioned in the Abstract of the paper. Even if the papers are relatively recent for their time, it is not as recent now or in a few years, which makes it hard to analyze to what extend their frameworks are still relevant. Additionally, the authors could have spend a bit more time on the discussion of each paper and more explicitly state which evaluation parameters are satisfied.

Naik and Jenkins [6] Naik and Jenkins have written a short paper in 2016 describing some evaluation criteria for Identity and Access Management (IAM) (IDaaS) on Mobile Cloud Computing concepts. The article succesfully manages to directly interest the reader on the paradigm of Mobile Cloud Computing in the Introduction section, as well as state why these evaluation criteria are extremeley. important. Although the quality of this short article is solid, a few more references for each of the criteria would have helped their case in deciding these metrics, which is exemplified by the small number of references in this paper. Additionally, the authors could have elaborated more on their future suggestions for their audience to improve upon the critical analysis of their own work and show the relevancy of its statements better.

Gao et al. [50] This article focuses on the advantages, motivations, disadvantages and issues related to MCC systems and concepts. One of its greatest strengths is the clear overview of all the positive and negative aspects of the MCC concept and argument each of these aspects well. Examples of this are shown in the Introduction with their itemization of MCC advantages and in the lists of motivations and benefits in section 2.2. Additionally, their discussions are described extensively and excellently, as is shown in section 4 and is able to improve upon Khan et al. in terms of general explanations and discussion on the relevant issues of the MCC concept. Furthermore, their conclusion is inline with most of the other works that have been presented. One weakness of the paper is that its figures and tables can be quite cluttered with information, for instance in tables 1 and 2, or have bad quality, such as in figures 4 and 6. Moreover, they could have stated more clearly what their conclusion is of their results, by either adding a small section at the end of chapter 4 or by extending the Conclusions section, since it is relatively difficult to extract this information from their current paper.

Mollah et al. [36] Mollah et al. have, like Khan et al., mostly written on the challenges and possible solutions of privacy and security for Mobile Cloud Computing. Another similarity between both papers is that they are extremely long and detailed in their explanations, which definitely benefits the quality of both papers. However, in contrast to Kahn et al, it adds the idea of newly developed techniques as new challenges for the security and privacy for the mobile platform. Additionally, numerous newer papers have been used for this review, which range mostly from 2013 at the earliest to 2016 at the latest, as is shown in tables 5 and 6. Thus, this makes the paper more relevant than the review by Kahn et al., although both have different analyses. Moreover, it has a much better structure than the previous papers, since it follows up on the criteria from section 3 in section 5. The weakest aspect of the paper is how it does not elaborate on certain aspects of privacy and security, such as the trade off between cost efficiency and security, as was done in Kahn et al. Additionally, adding a few more images for explaining certain concepts could help with making the paper a bit more accessible.

Conclusion: Therefore, Khan et al., Goa et al. and Mollah et al. observe how guaranteeing user privacy remains one of the major issues within these system. Furthermore, despite these frameworks addressing one or few aspects relatively well, most evaluation parameters or security requirements are not yet solved by one framework in particular, which currently leads to an agglomeration of multiple individual frameworks. The major issue with this idea is that such agglomeration of deploying and configuring security issues in a MCC environment is not cost efficient, as was noted by Mollah et al. What Mollah et al. does not mention is that, virtualization techniques for mobile users can help reduce costs, as was stated by Khan et al. and Goa et al. However, both articles, as well as Sgandurra and Lupu [65], note that such systems generate several security challenges themselves, which can quickly increase the number of security challenges currently presented in the literature. This was also the conclusion from the previous section on the services of IaaS, PaaS and SaaS.

6.4 Resiliency Future Business Applications

In order to determine whether IDaaS is sufficiently resilient for future business applications, the General Data Protection Regulation (GDPR) will be taken into account as a basis for our judgement. Furthermore, various issues and their solutions will be discussed to provide the reader an idea of how severe the problems are across the various topics, other services and platforms, which was discussed in sections 4 and 5. Moreover, the quality of the papers will also be taken into account when determining the severity of the problems, as well as the quality and applicability of the solutions for these problems.

Since cloud computing is still a relatively new concept [1], the number of papers written on IDaaS is still relatively small with most articles containing few citations compared to other services, such as IaaS or PaaS [2] [36] [52] [15]. This can also be argued, given that various old principles, such as

CIA, have started to become popular, due to the increasing interest in cloud computing [33], [34] [32]. Despite the age of these principles, it is evident that they are still crucial for defining the level of security in cloud computing systems. Due to these principles being crucial and due to IDaaS adhering, in theory, to these princples, it is able to conform to all the rules provided by the GDPR act [2]. This resiliency is mostly a consequence of much research in the field of cloud computing and identity management. This created the idea of some extremely useful algorithms and models, such as purpose-based encryption [2], isolation of data partitions [32], Secure Sockets Layer (SSL) [5], Multi factor authentication (MFA) [46] and Purbose-Based Access Control (PBAC) [2].

Despite the large number of protocols, policies and authentication methods for securing Identity in the cloud and despite adhering to the GDPR act, it is also indisputable that cloud computing, and thus IDaaS, is still in the early stages of development [52] [53] [5] [6]. Although a standard model exists for defining security in the cloud [2], due to the increase in various services as an adaptation of IaaS [16] [5], obtaining a standard model gets more difficult at each newly presented service [5] [60] [61]. Additionally, numerous it is also clear that numerous issues still persist in the services on which IDaaS is based, such as Command/Code Injection attacks, Deep dependency stack issues, Cross-site scripting and Cookie poisoning [15] [52] [53]. Furthermore, even if there are solutions to these issues, some of these solutions generate several other issues themselves, such as virtualization [65] [51] [50]. It was also observed that most of the solutions provided by these papers only manage to solve a particular part of the issues presented [51] [36] and that often an extremely deep analysis is required to get to the bottom of these issues [52] [53].

When discussing all of these issues related to the various cloud computing services, it should be mentioned that there is no 100% guarantee that the system is secure [2]. However, it can be argued that the CIA features become much more prominent when considering IDaaS even if future security issues are inevitable, which still is extremely complex and challenging as both social and technical aspects have to be taken into account [17] [20]. Furthermore, privacy and authentication still remain one of the most complex aspects to solve [41] [45] [2]. Moreover, is it also observed that, despite the advances in the various security methods, only recently have companies decided or thought about switching to Multi Factor Authentication [66]. This is most likely due to management not directly seeing the value/priority of these various authentication methods and the expensiveness of these security measures [67]. However, even if companies have started to apply the popular technique of MFA, it is not foolproof and various issues have been confirmed when using this technique [45] [46]. Therefore, it is recommended for companies to go a step further with these ideas and apply biometrics for these systems due to the increasing number of issues reported for the MFA technique [47] [45] [46].

Given all of these results, it remains doubtful whether the security for IDaaS is secure enough for future business applications. This is due to the large number of papers mentioning that such services are still in their early stages. Moreover, many issues are imported from older variants, which makes it difficult to provide a standard model for guaranteeing security for these types of services. Additionally, numerous companies, despite their usage of MFA, will need to keep updating their techniques for identifying users on the internet. This is crucial, since a violation of such princples will result in extremely severe problems for all victims involved in these hacks [49] [35] [33] [46] [34]. Therefore, it is recommended to give such systems a few more years to solve all of the recurring issues in these cloud computing services and allow research to investigate IDaaS on a much deeper level, so that it is resilient enough for future business applciations.

7 Conclusion

In this literature review, we have investigated and discussed IDaaS' security properties and its resiliency for future business applications. This was achieved by first providing some background information into the idea of Identity as a Service. Additionally, it was noted that security is a rather abstract concept and thus the investigation of IDaaS' security properties was divided into the section shown in the IDaaS Security Properties section. Afterwards, some literature research was carried out for similar cloud computing services and mobile platforms. In the Discussion section, all of the literature is discussed and the resiliency of future business applications for IDaaS is discussed. From this discussion, it is concluded that there are still numerous issues present in both the old and new cloud computing systems, which are not entirely resolved. Despite the fact that IDaaS adheres to the GDPR act, it is recommended for companies to wait with the implementation of IDaaS in business

applications to first remove all the vital issues within the current system, since little research has been done for this service in particular. Within a few years, this service ought to be evaluated again and companies should reconsider the idea of using an identity management system in the cloud.

References

[1] Isaac Odun-Ayo, Sanjay Misra, Nicholas A Omoregbe, Emmanuel Onibere, Yusuf Bulama, and Robertas Damasevicius. Cloud-based security driven human resource management system. In *ICADIWT*, pages 96–106, 2017.

[2] Tri Hoang Vo, Woldemar Fuhrmann, Klaus-Peter Fischer-Hellmann, and Steven Furnell. Identity-as-a-service: An adaptive security infrastructure and privacy-preserving user identity for the cloud environment. *Future Internet*, 11(5):116, 2019.

[3] Tri Hoang Vo. Identity-as-a-6ervice (idaas): a 0issing* ap for 0oving (nterprise $ pplications in inter-cloud. In *Proceedings of the Eleventh International Network Conference (INC 2016)*, page 121. Lulu. com, 2016.

[4] Wenxin Li, Deke Guo, Keqiu Li, Heng Qi, and Jianhui Zhang. idaas: Inter-datacenter network as a service. *IEEE Transactions on Parallel and Distributed Systems*, 29(7):1515–1529, 2015.

[5] Apurva Kumar. Model driven security analysis of idaas protocols. In *International Conference on Service-Oriented Computing*, pages 312–327. Springer, 2011.

[6] N. Naik and P. Jenkins. A secure mobile cloud identity: Criteria for effective identity and access management standards. In *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pages 89–90, 2016.

[7] Ch Rupa, Rizwan Patan, Fadi Al-Turjman, and Leonardo Mostarda. Enhancing the access privacy of idaas system using saml protocol in fog computing. *IEEE Access*, 8:168793–168801, 2020.

[8] Umme Habiba, Rahat Masood, Muhammad Awais Shibli, and Muaz A Niazi. Cloud identity management security issues & solutions: a taxonomy. *Complex Adaptive Systems Modeling*, 2(1):1–37, 2014.

[9] JE White. Rfc0105: Network specifications for remote job entry and remote job output retrieval at ucsb, 1971.

[10] John Backus. *Computer Advanced Coding Techniques (Archived)*. MIT, 1954. First known instance of time sharing.

[11] Fernando J Corbató, Marjorie Merwin-Daggett, and Robert C Daley. An experimental time-sharing system. In *Proceedings of the May 1-3, 1962, spring joint computer conference*, pages 335–344, 1962.

[12] Amazon Inc. Amazon.com launches web services; developers can now incorporate amazon.com content and features into their own web sites; extends "welcome mat" for developers, July 16, 2002. Archived, link: `https://archive.is/dx9Qj`, accessed 5-5-2021.

[13] AWS Amazon. Announcing amazon elastic compute cloud (amazon ec2)—beta, 2006. link: `https://aws.amazon.com/about-aws/whats-new/2006/08/24/announcing-amazon-elastic-compute-cloud-amazon-ec2---beta/`, accessed 3-6-2021.

[14] Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. Cloud computing: An overview. In *IEEE International Conference on Cloud Computing*, pages 626–631. Springer, 2009.

[15] P Ravi Kumar, P Herbert Raj, and P Jelciana. Exploring security issues and solutions in cloud computing services–a survey. *Cybernetics and Information Technologies*, 17(4):3–31, 2017.

[16] Sushil Bhardwaj, Leena Jain, and Sandeep Jain. Cloud computing: A study of infrastructure as a service (iaas). *International Journal of engineering and information Technology*, 2(1):60–63, 2010.

[17] Keiko Hashizume, David G Rosado, Eduardo Fernández-Medina, and Eduardo B Fernandez. An analysis of security issues for cloud computing. *Journal of internet services and applications*, 4(1):1–13, 2013.

[18] Sunilkumar S Manvi and Gopal Krishna Shyam. Resource management for infrastructure as a service (iaas) in cloud computing: A survey. *Journal of network and computer applications*, 41:424–440, 2014.

[19] Ali Khajeh-Hosseini, David Greenwood, and Ian Sommerville. Cloud migration: A case study of migrating an enterprise it system to iaas. In *2010 IEEE 3rd International Conference on cloud computing*, pages 450–457. IEEE, 2010.

[20] Maciej Malawski, Gideon Juve, Ewa Deelman, and Jarek Nabrzyski. Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in iaas clouds. *Future Generation Computer Systems*, 48:1–18, 2015.

[21] Luis M Vaquero, Luis Rodero-Merino, and Daniel Morán. Locking the sky: a survey on iaas cloud security. *Computing*, 91(1):93–118, 2011.

[22] B Hari Krishna, S Kiran, G Murali, and R Pradeep Kumar Reddy. Security issues in service model of cloud computing environment. *Procedia Computer Science*, 87:246–251, 2016.

[23] Craig Mathias Linda Rosencrance. identity management (id management), November 2020. link: `https://searchsecurity.techtarget.com/definition/identity-management-ID-management`, accessed: 10-5-2021.

[24] Mikael Ates, Serge Ravet, Abakar Mohamat Ahmat, and Jacques Fayolle. An identity-centric internet: identity in the cloud, identity as a service and other delights. In *2011 Sixth International Conference on Availability, Reliability and Security*, pages 555–560. IEEE, 2011.

[25] Jaweher Zouari and Mohamed Hamdi. Aidf: An identity as a service framework for the cloud. In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–5. IEEE, 2016.

[26] https://www.scopus.com/home.uri.

[27] https://www-scopus-com.proxy.uba.uva.nl/term/analyzer.uri?sid=4f43686394f1cdfc5e325fc26ef02f9b&origin=resultslist&src=s&s=TITLE ABS-KEY%28%28idaas+OR+identity+AND+as+AND+a+AND+service+%29+AND+cloud+AND+%28security+OR+privacy%29%29&sort=plf-f&sdt=b&sot=b&sl=96&count=729&analyzeResults=Analyze+results&txGid=2d42628e7294159eec6fb6104028ffe4.

[28] Jeffrey James Stapleton. Security without obscurity: a guide to, confidentiality, authentication, and integrity. Boca Raton, Florida, 2014. CRC Press.

[29] L. O. Nweke. Using the cia and aaa models to explain cybersecurity activities. volume 6. PM World Journal, 2017.

[30] David Price. The cia, aaa, and the ethical problems inherent in secret research. Lanham, Maryland, 2003. Altamira Press.

[31] & Breithaupt J. Merkow M. S. Information security: Principles and practices. Pearson Education, 2014.

[32] Dimitrios Zissis and Dimitrios Lekkas. Addressing cloud computing security issues. *Future Generation Computer Systems*, 28(3):583–592, 2012.

[33] Peter S. Browne. Data privacy and integrity. *Proceedings of the 1971 ACM SIGFIDET (now SIGMOD) Workshop on Data Description, Access and Control - SIGFIDET 71*, 1971.

[34] Y. Deswarte, L. Blain, and J.-C. Fabre. Intrusion tolerance in distributed computing systems. *Proceedings. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*.

[35] Sanjay P. Ahuja and Sindhu Mani. Availability of services in the era of cloud computing. *Network and Communication Technologies*, 1(1), 2012.

[36] Muhammad Baqer Mollah, Md Abul Kalam Azad, and Athanasios Vasilakos. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *Journal of Network and Computer Applications*, 84:38–54, 2017.

[37] Hyokyung Chang and Euiin Choi. User authentication in cloud computing. In *International Conference on Ubiquitous Computing and Multimedia Applications*, pages 338–342. Springer, 2011.

[38] PS Dowland, SM Furnell, and Maria Papadaki. Keystroke analysis as a method of advanced user authentication and response. In *Security in the Information Society*, pages 215–226. Springer, 2002.

[39] Yu Zhong, Yunbin Deng, and Anil K Jain. Keystroke dynamics for user authentication. In *2012 IEEE computer society conference on computer vision and pattern recognition workshops*, pages 117–123. IEEE, 2012.

[40] Naseer Amara, Huang Zhiqui, and Awais Ali. Cloud computing security threats and attacks with their mitigation techniques. In *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pages 244–251. IEEE, 2017.

[41] Giovanni L Masala, Pietro Ruiu, and Enrico Grosso. Biometric authentication and data security in cloud computing. In *Computer and Network Security Essentials*, pages 337–353. Springer, 2018.

[42] UK National Cyber Security Centre. Protect your email by using a strong and separate password, Dec 17, 2018. link: `https://www.ncsc.gov.uk/collection/top-tips-for-staying-secure-online/use-a-strong-and-separate-password-for-email`, accessed: 26-5-2021.

[43] Jisc community. Passwords: Threats and counter-measures. link: `https://community.jisc.ac.uk/library/janet-services-documentation/passwords-threats-and-counter-measures`, accessed: 26-5-2021.

[44] ProofID. What are knowledge factors, possession factors and inherence factors?, Apr 2020. link: `https://proofid.com/blog/knowledge-factors-possession-factors-inherence-factors/`, accessed: 26-5-2021.

[45] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3):614–634, 2001.

[46] Radhesh Krishnan Konoth, Victor van der Veen, and Herbert Bos. How anywhere computing just killed your phone-based two-factor authentication. In *International Conference on Financial Cryptography and Data Security*, pages 405–421. Springer, 2016.

[47] Aleksandr Ometov, Sergey Bezzateev, Niko Mäkitalo, Sergey Andreev, Tommi Mikkonen, and Yevgeni Koucheryavy. Multi-factor authentication: A survey. *Cryptography*, 2(1):1, 2018.

[48] Salman H Khan and M Ali Akbar. Multi-factor authentication on cloud. In *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7. IEEE, 2015.

[49] Tri Vo, Woldemar Fuhrmann, and Klaus-Peter Fischer-Hellmann. Privacy-preserving user identity in identity-as-a-service. 02 2018.

[50] Jerry Gao, Volker Gruhn, Jingsha He, George Roussos, Wei-Tek Tsai, et al. Mobile cloud computing research-issues, challenges and needs. In *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, pages 442–453. IEEE, 2013.

[51] Abdul Nasir Khan, ML Mat Kiah, Samee U Khan, and Sajjad A Madani. Towards secure mobile cloud computing: A survey. *Future generation computer systems*, 29(5):1278–1299, 2013.

[52] Meiko Jensen, Jörg Schwenk, Nils Gruschka, and Luigi Lo Iacono. On technical security issues in cloud computing. In *2009 IEEE international conference on cloud computing*, pages 109–116. Ieee, 2009.

[53] Mohamed Almorsy, John Grundy, and Ingo Müller. An analysis of the cloud computing security problem. *arXiv preprint arXiv:1609.01107*, 2016.

[54] Michael P Papazoglou and Dimitrios Georgakopoulos. Introduction: Service-oriented computing. *Communications of the ACM*, 46(10):24–28, 2003.

[55] Indranil R Bardhan, Haluk Demirkan, PK Kannan, Robert J Kauffman, and Ryan Sougstad. An interdisciplinary perspective on it services management and service science. *Journal of Management Information Systems*, 26(4):13–64, 2010.

[56] Wenjun Zhang. Integrated security framework for secure web services. In *2010 Third International Symposium on Intelligent Information Technology and Security Informatics*, pages 178–183. IEEE, 2010.

[57] OWASP. The ten most critical web application security vulnerabilities., 2010. link: `http://www.owasp.org/index.php/OWASP_Top_Ten_Project`, accessed: 01-06-2021.

[58] Issa M Khalil, Abdallah Khreishah, and Muhammad Azeem. Cloud computing security: A survey. *Computers*, 3(1):1–35, 2014.

[59] Dimitrios Zissis and Dimitrios Lekkas. Addressing cloud computing security issues. *Future Generation computer systems*, 28(3):583–592, 2012.

[60] Subashini Subashini and Veeraruna Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(1):1–11, 2011.

[61] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.

[62] Sari Sultan, Imtiaz Ahmad, and Tassos Dimitriou. Container security: Issues, challenges, and the road ahead. *IEEE Access*, 7:52976–52996, 2019.

[63] Ado Adamou Abba Ari, Olga Kengni Ngangmo, Chafiq Titouna, Ousmane Thiare, Alidou Mohamadou, Abdelhak Mourad Gueroui, et al. Enabling privacy and security in cloud of things: architecture, applications, security & privacy challenges. *Applied Computing and Informatics*, 2019.

[64] Bashar Alohali. Security in cloud of things (cot). In *Cloud Security: Concepts, Methodologies, Tools, and Applications*, pages 1188–1212. IGI Global, 2019.

[65] Daniele Sgandurra and Emil Lupu. Evolution of attacks, threat models, and solutions for virtualized systems. *ACM Computing Surveys (CSUR)*, 48(3):1–38, 2016.

[66] Rob Lemos. The state of mfa: 4 trends that portend the end of the solo password, Jan 12, 2021. link: `https://techbeacon.com/security/state-mfa-4-trends-portend-end-solo-password`, accessed: 03-06-2021.

[67] Kara Heinrichs. Top 7 reasons companies don't use two-factor authentication, Dec 14, 2012. link: `https://duo.com/blog/top-7-reasons-companies-don-t-use-two-factor-authentication`, accessed: 03-06-2021.

Serverless Computing and Function as a Service: A literature study

Abinash Satapathy
Department of Computational Science
University of Amsterdam
Science Park 904, 1012WX Amsterdam
abinash.satapathy@student.uva.nl

Ahmed Abdelghany
13526588
University of Amsterdam
Science Park 904, 1012WX Amsterdam
ahmed.ahmed.mohamed.abdelghany.hefny@student.uva.nl

Ivo de Geus
11251190
University of Amsterdam
Science Park 904, 1012WX Amsterdam
ivo.de.geus@student.uva.nl

Abstract

The domain of cloud computing has seen a major advancement with the introduction of serverless computing. In this mechanism, the code maintenance is the priority as opposed to the management of the server. Different companies like Microsoft, IBM, Amazon, etc. are offering serverless computing features with a "pay-as-you-go" model. This enables developers to solely focus on the code and wait for the task to provide output based on the functionality they have subscribed for from their provider. Various experts have provided different patterns for handling serverless operations. In this work, provide a literature review of the of various peer-reviewed journals and establish a performance analysis of serverless computing and its providers with a special mention of Function-as-a-Service (FaaS).

1 Introduction

This literature review treats the development of Serverless computing and Functions as a Service, of which a well-known example are the AWS Lambda-functions. The increase of these business cases has developed steadily with the recent shift of enterprise application architectures to more shared, independent microservices [18]. In this report, we discuss the different types of webservices, their development over time and the business cases where these can be applied.

In an effort to reduce cloud service costs, virtualisation has risen steadily, moving first from hardware to containers, and further onward to serverless computing. Offering of these services exists on a spectrum ranging from owning infrastructure directly, also known as bare metal, to the architecture known as Functions As A Service (FaaS). See an implementation of this spectrum in **??**. The amount of work required for clients, and implicitly therefore also control available to the developing client, changes as different tasks are done by the provider [4].

As described by [3], Amazon Web Services were the first to launch their Lambda service, which saw significant adoption from the end of 2014 to the end of 2016. Currently, a lot of Cloud Service Providers offer similar kind of products. In [4], we see competitors such as Google Cloud Functions, Azure Functions (Microsoft), IBM OpenWhisk and the open-source implementation OpenLambda. In [3], it is also argued that the term 'Serverless' is merely referring to a new generation of platform-as-a-service offering, which have existed beforehand, with the only difference being that capacity planning and task scheduling have now changed in ownership from the developer to the owning company. We will first describe roughly four steps which through which these business cases have evolved, after which we will discuss their consequences and implementations.

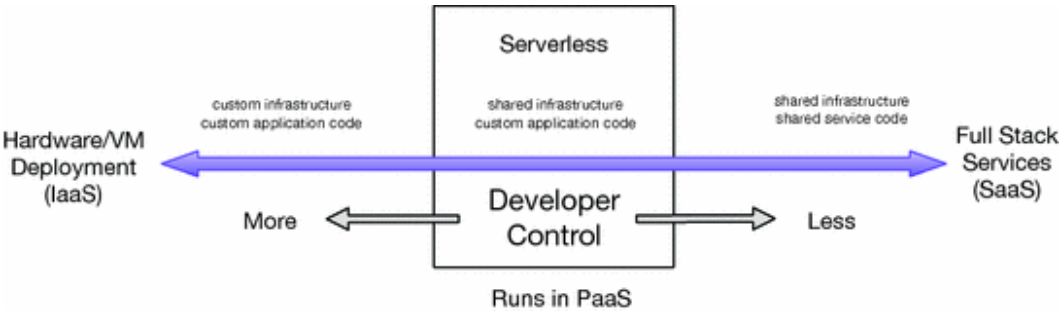


Figure 1: Developer control and serverless computing from [4].

The most straight-forward is using bare-metal. Infrastructure and software licences have to be bought, and all maintenance is responsibility of the company. Some simple limitations that are common with this type of infrastructure are resource limits, cold starts and inflexible price increases when the need for expansion arises [15].

A direct step up of hiring the bare metal is Infrastructure as a Service (IaaS), where the provider sets up the infrastructure and ensures its location and functioning, but control is mostly with developers over running operation and the code of the application in the cloud. In this case, the responsibility of provisioning virtual machines, hardware specifications, optimizing all features of deployment, and execution of an application is the developer's responsibility. The most traditional approach has a single machine with a process listening on a utp/tcp port, ready to add a request to the action queue [2]. This would be the typical example of a web server or a message queue.

Arguably the next step up is the containerization of software, where provisioning and spinning up of virtual machines or containers (such as Dockers) with software pre-configured is increasingly facilitated. In this spectrum, we can define Backend as a Service (BaaS) as an online facility that delivers a specific job over the cloud, such as notification or authentication. Also, we can identify the Function as a Service (FaaS) where developers can program, manage, and run functionalities of applications without handling the burden of infrastructure, software updates administration, and their requirements. Both FaaS and BaaS require no management of resources from the customer side. Whereas FaaS presented to execute users' functions, and BaaS offers a fully online service, a combination of FaaS and BaaS gives us the serverless service that includes the following features.

The previous items can be called a type of Software as a Service which have the potential to lower business costs for end-users by sharing resources and adding flexibility to the client [9]. In this way, we can consider serverless computing as a similar type of adding flexibility and economies of scale, only for developers. These economies of scale are shown by Cisco Public [7], where 83% of all public cloud servers are maintained by about 24 Cloud Service Providers (CSPs). Serverless computing is in this sense an innovative way of cloud computing wherein the developers are solely focused on execution of the code without the hassle of the maintenance of the servers in the back-end. As argued by [4], this offers cloud providers a large amount of extra control over the entire development stack and add flexibility of their scaling and management of their cloud resources.

In the case of Functions as a Service, code will be deployed in the structure of small, independent fragments in containers, which will execute a specific set of instructions when an event-trigger is fired. A developer has control over the code developed and when it is triggered and the CSP is tasked with facilitating the underlying requirements. Serverless computing want characteristic as a provider and storage as a provider on the cloud infrastructure, which is deployed with the required library

bundle. A big advantage of containers deployed as Functions as a Service are their independence which allow them to scale horizontally fairly well.

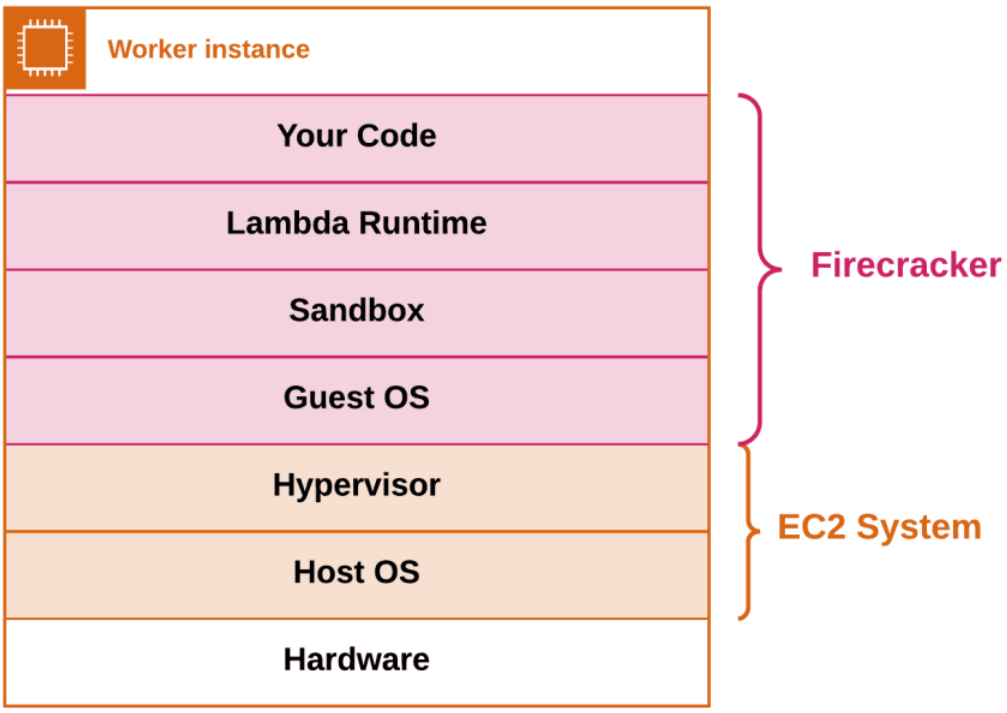


Figure 2: Implementation level of Lambda code on a system [8].

In an Application, microservice represents small performance as a service, and see a typical setup on Amazon Web Services in Figure 2. Application that are constructed on microservices will be orchestrators of more than one microservices. This will turn out to be a dynamic technique for routing, distributing and managing a couple of offerings to signify the enterprise state of affairs in a software in a unique order. More complexity will create challenges in checking out of a software as well. Managing model of every micro carrier and information trade between a couple of micro offerings will amplify greater complexity to the scenario. The notion of having smaller functions that each are dedicated to a specific task is in some way similar to the map-reduce technique, which can effectively be implemented using serverless computing [13].

1.1 Serverless Computing problems/constraints

In building functions to run in a serverless computing environment, several characteristics define its performance. A range of architecture and system limits which define performance, like restrained runtime resources, optimized code, wide variety of concurrent requests, reminiscence and CPU reachable for feature invocation. Each Cloud Service Provider (CSP) tends to provide a set of utilities and equipment for monitoring logs, records, management changes and invoked events. [12] These utilities send notifications in the case of a alert, such as authentication and authorization which might also be used to get records for computer gaining knowledge of on overall performance trying out of FaaS. Scaling of FaaS can be, in theory, from zero to infinity, and depends on the defined limits by the client and the offered infrastructure by the CSP. Several levels of code complexity can be implementated as an excecution on a FaaS platform. Depending on this, launch instances are distinctly variable and can take a few seconds to a few minutes to begin the service. After this, a process can take as long as it needs be, contstrained by the uptime of the CSP. [3] Performance here can be enhanced in low latency, high-throughput and optimised storage. As argued by [3], the easy separation of functions allows for an easy versioning implementations, where several versions of the same functions can easily co-exist, depending only on (for example) a header sent by the client on the required target version.

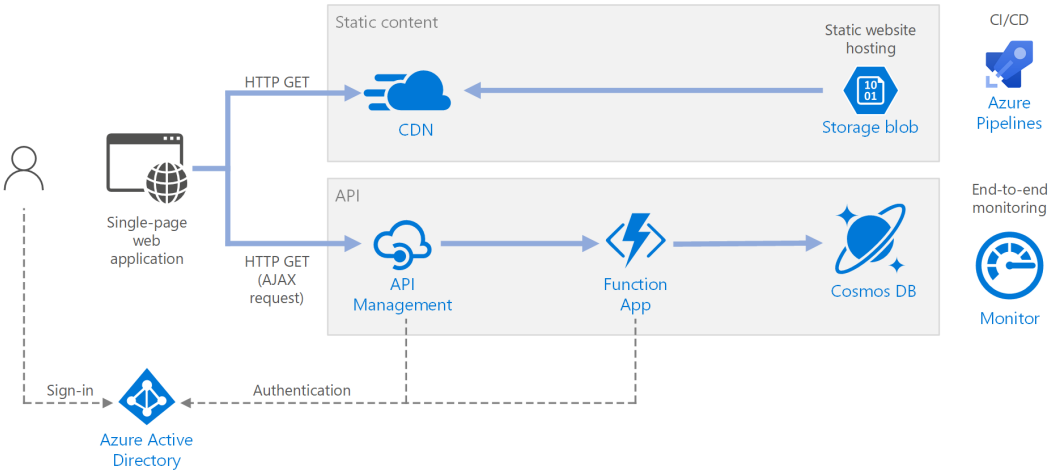


Figure 3: Azure setup of a serverless function endpoint [16]

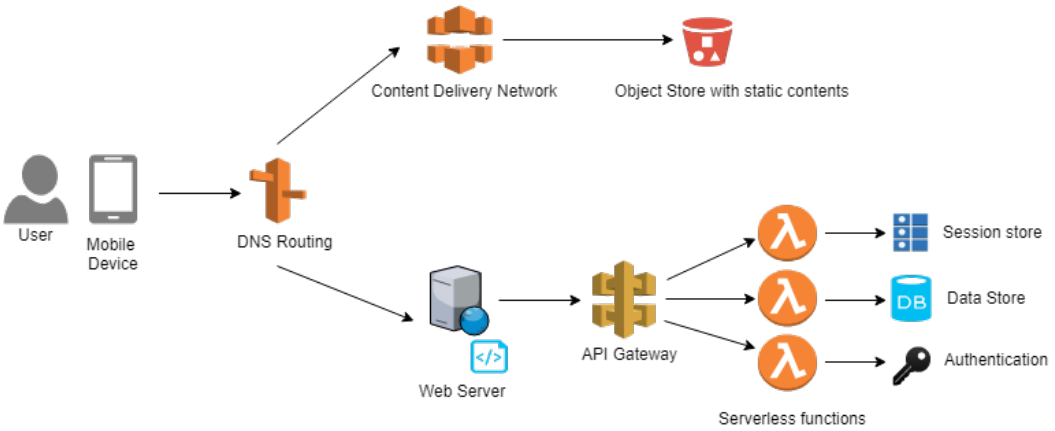


Figure 4: Serverless Architecture diagram as implemented by AWS [19]

There are 4 states of serverless infrastructure: provider, VM, container cold, and warm, demonstrating how microservice output varies up to 15x relying on these conditions [14]. Even on FaaS, higher configuration can enhance performance. One of the experiments suggests that proper configuration of FaaS will supply higher overall performance continually on community and computing platform.

When a CSP experiences increased load on one of the core resources, a situation of peak and spike demand can create a shortage of required sufficient resources. There are several ways to cope with this problem, of which one is agreeing on enforceable useful resource limits on a serverless environment, with provisioning and a dynamically allocated reserve. These agreements can encompass memory, execution time, bandwidth, and CPU usage. The scalability requirements are that they proactively additionally aggregate resource limits that can be utilized throughout a wide variety of features or throughout the complete platform. As per the evaluations on overall performance of serverless computing, there are quite some enhancement in serverless computing to optimize the cost of those constraints.

2 Features and characteristics

According to [1], there are some features of a serverless service, which includes the knowledge of the service provider of the code and run-time state after and while execution, outside libraries dependencies and data dependencies of data, which imply the responsibility of the service provider to execute the code when requested in a timely manner, with auto-scaling capabilities so resources are

allocated and released according to requests lead, and subsequently, billing will be according to the model of the pay-as-you-go. As argued by [3], typical prices for usage are counted by incorporating billable units per hundred milliseconds and adding the fail-over costs. Normally, the orchestrator of defined FaaS-functions will shut down functions after a specified time of inactivity, keeping a copy of the virtual machine to facilitate a warm start. According to [4] this can lead to both a risk-scenario where a sudden increase in computing requirements cause a spike in billing, as well to an optimum-scenario where in the case of zero demand the bill is reduced to zero as well. This in contrast to a Platform as a Service (PaaS)-architecture, where these costs are typically fixed.

Serverless service providers can be distinguished with several characteristics, Serverless clients should mind when they want to use serverless services [5]: Debugging and monitoring: essential debug procedures are supported by each provider with records in the execution logs. Serverless providers may deliver additional facilities to support developers better recognize function execution circumstances and find trace errors and bottlenecks.

Accounting Security: Serverless providers must confirm isolation functions execution between clients and secure a comprehensive accounting structure to allow users to calculate how much they will pay.

Deploying: Serverless providers make every effort on deployment procedures to make it straightforward as possible. What clients need to supply is a file with the source code. In addition to many choices like archive software with multiple files or create a docker image with binary code with options to version or group iterations of code.

Building: serverless providers offer ways to make new instances from one serverless function code, and many providers offer simple procedures for composing readymade functions for several applications and may even help customers to construct more complex serverless applications. Development template: most of the time, serverless providers run a single main function that requires a dictionary-like as a JSONobject as an input and generates a dictionary as a deliverable.

Languages of development: Serverless providers offer many languages to use for development like Python, Java, Go, C#, Swift, and Javascript. Nearly all of the providers offer many development languages. They also welcome packaged Docker images that support well-defined APIs with whatever development language. Limitations of Performance: serverless clients need to take care of many aspects of serverless code runtime resource requirements, like the CPU and memory resources allocated to run each function, concurrent requests that could be delivered. Some numbers are inherent from the platforms like the maximum allocated memory, and others could be expanded when users' demands increase, such as the threshold of concurrent delivered requests.

Price: clients pay only for the time and resources that they used when serverless functions are available. One of the serverless platform distinguishers is the capability to scale to zero instances when no requests hitting the service. Serverless service providers offer different cost structures of measured computing resources, like CPU and memory, in addition to performance features like off-peak discounts.

Suffice it to say, the main reasons for serverless computing to be in high demand is because of the reasons summarised as follows:

- Elasticity of the service: automatic scaling up/down
- Pay as you use concept
- No need of maintenance of the server
- Any language can be deployed to code
- Always available when needed

2.1 Service Oriented Architecture

The affect of service-oriented architecture on enterprise. The lookup made until now in the area of administration and service-oriented technological know-how led to a series of managerial guidelines with the intention of growing the price of business using service-oriented science. There have additionally been proposed elements of exchange in administration and adjustment of commercial enterprise features to serviceoriented technology. We are in modern times going through the want of growing an surroundings for the adjustment of corporation to service-oriented technology. The nature of business activities may additionally seriously decide the success of the services' paradigm. [17]

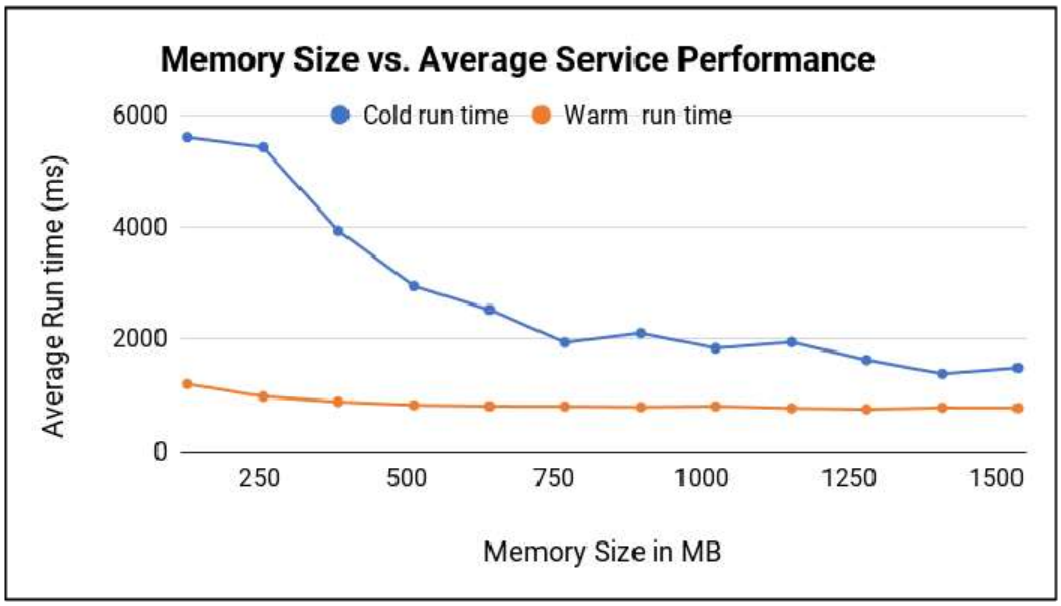


Figure 5: Memory vs Service performance from [14]

Agile structure for the service-oriented enterprise. At present, there are a limited wide variety of corporation architectures, architectures of structures of applications and overall performance evaluation environments for service-oriented environments. In addition, improvement methodologies, overall performance evaluation and optimization models for service-oriented enterprises/networks of organizations are uncommon and no longer very systematic. The aspect of novelty resides in the introduction of an architectural model for service-oriented companies that have to take away the gap between the stage of the business and the stage of the statistics system, and need to enable transformation of organization know-how into technical knowledge, wished by means of records structures in order to make certain interoperability between partners. The architectural structure must comprise operation and improvement agile structure for the integration of the enterprise inside the service-oriented environment. The stage of complexity resides in the collaborative context and in the effort to optimize the structure of service-oriented enterprises. [17]

2.2 Development Steps

Developing around a Function as a Service requires thinking in micro-services. Every function is spun-up "on-the-fly" as soon as required and deactivated typically after a predefined timeout. Typically, a specific function will be as atomic as possible, focussing on one specific task only, reinforced with unit tests and continuous integration.

Cloud Service Providers typically offer a wide possibility of shared code so there is little need for expansive provisioning instructions. Languages which can be used in programming Functions as a Service are extensive and can range from Python and Java to Swift, Node.js and Go. Depending on the implementation, retrofitting older code to a function is easy enough. An example is given by the frameworks Zappa [24] and Chalice [6], which can be retrofitted in existing python code by using a decorator. Developing a Function is typically done by adding a json-object with extra instructions on build environment, configuration parameters and dependencies.

When an programming language is not available, or when the complexity of a function rises further, it is sometimes possible to create a docker binary, which is previously compiled and can be started and stopped by the CSP. This has its downsides, as this complexity is not typically easy to optimize for the CSP. This form is closer to a Platform as a Service, as the developer still configures the docker itself.

Serverless clients need to take care of many aspects of serverless code runtime resource requirements, like the CPU and memory resources allocated to run each function, concurrent requests that could

be delivered. Some numbers are inherent from the platforms like the maximum allocated memory, and others could be expanded when users' demands increase, such as the threshold of concurrent delivered requests. Clients typically pay only for the time and resources that they used when serverless functions are available. One of the serverless platform distinguishers is the capability to scale to zero instances when no requests hitting the service. Serverless service providers offer different cost structures of measured computing resources, like CPU and memory, in addition to performance features like off-peak discounts.

3 Traditional Approaches

The traditional frameworks of serverless service started with Virtual Machines (VMs) or containers, to guarantee isolation of application and provisioning of resources. These frameworks are not effective for delivering low latencies, mainly when code is triggered for the first time and has a heavyweight to operate at Edge systems[10]

A serverless function in a VM environment is an instance of a group of memory (stack and heap), application code, language runtime, and library dependencies. the footprint of a serverless function is drastically lighter than a provisioned VM footprint, which contains virtual hardware resources, and guest OS. While provisioning of a VM could start in tens of seconds, serverless service providers use innovative techniques to optimize provisioning the serverless environment execution interval.

AWS serverless service, AWS Lambda has two serverless functions techniques of segregation and sandboxing [3]:

- Assigned AWS EC2 instances (traditional VMs) used to segregate various AWS accounts, meanwhile, different function calls are sandboxed by usual container methodologies.
- Firecracker microVMs are used to offer sandboxing environments for each function calls.

The other methodology is container-Based, which offers a traditional way of bundling the application's code of each user, its configurations, in addition to libraries dependencies into a single container, which shares an operating system with all other containers. Containers are resource-isolated processes with the underlying operating system with features like control groups (cgroups) and namespaces.

Several cloud service providers as IBM, Google implement the technology of containers for supporting the platform as a service (PaaS) and more recently the serverless services[23].

4 Challenges faced by Serverless computing

Here, we list out some basic challenges faced by serverless computing. Just like any technology has got some drawbacks or challenges, this new concept of serverless computing also falls victim to the same. Here are some of those:

- As serverless computing is to pay as you use mannequin by using a range of service providers, it is vital to optimize the code and configuration of service. If the code and configuration is now not optimized and enable positive connections to be stored alive all the time, it will value higher. To preserve the fees down the use of FaaS, one must be worried with the requests' execution time. In addition, test has proven that dependency on the use of exterior services, such as database, authentication services, amongst others, can appreciably intrude with the charges.[12]
- One of the overall performance evaluation experiments suggests that microservice purposes that are concurrently importing and downloading information executing on AWS Lambda are no longer going to scale properly.[5]
- Research published that the availability of greater assets had the high-quality have an impact on on ordinary performance. In the FaaS environment, there used to be a direct relationship between the quantity of memory and the processing assets assigned by way of the function.
- Slow network may cause user to get timed out quite often
- Quality of code is at once linked to the cost. The horrific code will price greater in pay per execution time in serverless computing.[10]

5 Serverless solutions limitations

Serverless solutions have been used with several applications comprising processing of event stream, ETL, and API serving, below we will review the limitations to use it in other fields:

Storage and serverless performance: Because of the serverless nature that it does not save the current state of the application, it is not the perfect solution for stateful applications which need state sharing. Cloud service providers offer object storage like Google Cloud Storage, Azure Blob Storage, and AWS S3 offer inexpensive long-term object storage but exhibit high access latencies, which need at least 10 milliseconds to write or read any object. [6] If clients choose to use fast IOPS services offered by cloud providers, prices grow dramatically [7].

A large gap in foreseen performance: As per the experiment [20] serverless functions many times got CPUs from different hardware generations which affected performance due to the variety in the hardware capabilities, as per the serverless principles serverless providers have the freedom to select the underlying infrastructure. As serverless provider operations prefer to maximize the use of their resources, the uncertainty of serverless performance becomes a feature of these services.

UC Berkeley has tested the AWS Lambda serverless service in 2018 [11] regarding the below aspects after they have noticed the advantages of serverless service over the other cloud services like autoscaling features which allow clients to pay only for resources that they used:

Equipment options are limited: AWS lambda clients are permitted only to configure few RAM size options and CPU hyper-thread, which are coming in limited prepackaged offers. Freedom to choose specialized equipment configurations is not offered to clients. Later in the same year, another team [10] was promising the feature of allowing clients to choose specific equipment to build their serverless service available in near future.

Sluggish Storage is the only option to interact: As a serverless service running, it does not have a network interface, while it can commence network connections outbound. Subsequently, when AWS lambda functions need to interact, they should use another service which will be Amazon S3 storage service which adds new cost, that was not existing if they can talk directly point to point and dramatically slower, which is the same case when clients need to save the previous output of lambda function to use it for iteration or further analysis, they have to access slow storage like S3 to maintain status across client interactions with a Lambda function.

Bandwidth limitations: regarding the network bandwidth allowed to use by serverless cloud services to connect to the internet or other cloud services, a study [22] a single serverless function from one of the three famous cloud providers, Google, AWS, and Azure can reach an average of half a gigabit per second network bandwidth which is slower than the bandwidth speed an SSD storage can deliver. Moreover, cloud providers tend to aggregate each client's serverless functions on a single virtual machine together, which ends up in a situation where all the client functions share the same above bandwidth, which gets worse every time the client adds a new function or scale up to serve new requests.

Lifespan restriction: Lambda serverless function is cashed in a virtual machine to enable the fast start of the function when triggered but after specific idle time around a quarter-hour, function instance is terminated, and state will be lost, with no way to recover it in the next trigger. Subsequently, the function should be developed in a style that does look for a recoverable state when it starts[9].

Provider boundaries: implementing serverless function applications with one service provider, makes the migration to another one is a challenging task. This is due to the different implementation methodologies like numerous API-Gateways used by each provider and required integration with the provider facilities[21].

6 IoT as an Application

The scientific and business societies are searching for an adequate infrastructure to serve the emerging Internet of things applications, this infrastructure should have execution in real-time, low latency communications, event-driven initialization, and efficient operation.

Event-based nature: With the ever-increasing growth of IoT applications, the community was required to devote extra effort to address their requirements. Requirements include low latency, real-time

execution, event-driven developments, and efficient deployments [12]. Serverless functions with event-driven features support the IoT nature of their limited operation and periodic processing rather than continuous operation, which will minimize the consumption of energy to cover only the event processing time needed by IoT applications.

Stateless actions: IoT applications run tasks like record reading or process input which mostly do not require any knowledge of the previous events, like in surveillance applications which process images with specific criteria regardless of any dependencies between them. So, the stateless nature of serverless function serves IoT applications demand and even optimized for IoT needs of stateless function [13].

Parallel processing: IoT can benefit from the serverless nature of parallelism especially when the event processing can be decomposed into separate tasks that need to be handled independently, as IoT devices are idle most of the time but when an event is triggered, serverless functions can initialize parallel instances to process the input signals independently as in security tracking systems that need to make different independent tasks in the same time like determine the moving object dimensions and its direction, how many people in the car[14].

Granular scalability: resource is always a challenge at the edge of the network, and traditional virtual machines are not the best solution because of their big memory requirements which do not have the options to scale due to the limited resources which serverless function can use in an economical pattern with their ability to create and terminate in a smaller time and storage [1].

7 Discussion

While serverless computing is showing potential to add flexibility to both small-scale and enterprise software development, their programming model and added challenges such as the required independence can add unnecessary complexity to a program [15].

Serverless computing has possible pitfalls, as showed by critical academic responses. [11], a paper from 2019, argues that FaaS can be seen more of a data-shipping architecture, than a specialised data-processing architecture. This is due to the limited lifetimes of virtual machines, possible I/O bottlenecks (as processes are distributed and require more internal data transfer) and a possible lack of specialised hardware. [3] argues that the very nature of Functions as a Service can cause for a potential high latency, as the requirement of scaled up instances is defined on the fly. While a guaranteed instantiated nodes according to a schedule can help with this, this takes away control from the CSP, returning the control to the developer. As argued by [14], the problem of increased load which cannot immediately be satisfied by existing running instances can be satisfied by always overprovisioning a certain factor.

8 Conclusions

In this article, we have looked into the creation and applications of serverless applications and Functions as a Service. There are many ways to approach it, from the original implementation as Amazon did it in 2014, to more diverse approaches. While serverless platforms are very attractive for business due to their flexibility in scale-out horizontally in a quick and easy way, their potential fallbacks can create problems for business-cricital tasks. The notion of applying seperate functions does not necessarily have to be applied in a serverless environment, but can help archieve stability and a shorter time to production due to a reduced software architecture.

In this paper, a number of challenges of serverless are also mentioned primarily based on existing studies. Although exsiting research suggests a variety of observations and experiences with serverless computing, nevertheless there is a lot of probability to increase the perception of serverless platform overall performance in the location of clocking and community latencies to test the exceptional measures, language runtime, task utilization, characteristic code size, device overall performance of serverless platforms, match types, CPU allocation scaling.

Previous sections of this paper have pointed in the direction of overall performance challenges in serverless computing, now there is want to measure it and enhance performance. It is feasible by using doing overall performance engineering with the proper approach and device sets.

References

[1] Paarijaat Aditya et al. "Will Serverless Computing Revolutionize NFV?" In: *Proceedings of the IEEE* 107.4 (2019). ISSN: 15582256. DOI: 10.1109/JPROC.2019.2898101.

[2] Gojko Adzic and Robert Chatley. "Serverless Computing: Economic and Architectural Impact". In: *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. ESEC/FSE 2017. Paderborn, Germany: Association for Computing Machinery, 2017, pp. 884–889. ISBN: 9781450351058. DOI: 10.1145/3106237.3117767. URL: https://doi.org/10.1145/3106237.3117767.

[3] Gojko Adzic and Robert Chatley. "Serverless computing: economic and architectural impact". In: (2017), pp. 884–889. DOI: 10.1145/3106237.3117767.

[4] Ioana Baldini et al. "Serverless Computing: Current Trends and Open Problems". In: *Research Advances in Cloud Computing*. Springer Singapore, 2017, pp. 1–20. DOI: 10.1007/978-981-10-5026-8_1. URL: https://doi.org/10.1007/978-981-10-5026-8_1.

[5] Ioana Baldini et al. "Serverless computing: Current trends and open problems". In: *Research Advances in Cloud Computing*. 2017. DOI: 10.1007/978-981-10-5026-8{_}1.

[6] *Chalice*. URL: https://github.com/aws/chalice.

[7] Cisco Public. "Cisco Global Cloud Index: Forecast and Methodology, 2015–2020". In: (2016), pp. 2015–2020. URL: http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf.

[8] Dev.TO. *Anatomy of AWS Lambda*. 2018.

[9] Abhijit Dubey and Dilip Wagle. "Delivering software as a service". In: *The McKinsey Quarterly* 6.2007 (2007), p. 2007.

[10] Phani Kishore Gadepalli et al. "Challenges and opportunities for efficient serverless computing at the edge". In: *Proceedings of the IEEE Symposium on Reliable Distributed Systems*. 2019. DOI: 10.1109/SRDS47363.2019.00036.

[11] Joseph M. Hellerstein et al. "Serverless computing: One step forward, two steps back". In: *CIDR 2019 - 9th Biennial Conference on Innovative Data Systems Research*. 2019.

[12] Deepak Khatri, Sunil Kumar Khatri, and Deepti Mishra. "Potential Bottleneck and Measuring Performance of Serverless Computing: A Literature Study". In: *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*. 2020. DOI: 10.1109/ICRITO48877.2020.9197837.

[13] Mariam Kiran et al. "Lambda architecture for cost-effective batch and speed big data processing". In: *2015 IEEE International Conference on Big Data (Big Data)*. 2015, pp. 2785–2792. DOI: 10.1109/BigData.2015.7364082.

[14] Wes Lloyd et al. "Serverless computing: An investigation of factors influencing microservice performance". In: *Proceedings - 2018 IEEE International Conference on Cloud Engineering, IC2E 2018*. 2018. DOI: 10.1109/IC2E.2018.00039.

[15] Theo Lynn et al. "A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms". In: *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*. 2017, pp. 162–169. DOI: 10.1109/CloudCom.2017.15.

[16] Microsoft. *Azure architecture*. URL: https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/serverless/web-app.

[17] Marinela Mircea. "Service-oriented enterprise: Taking the next step beyond agility in the digital economy". In: *Communications in Computer and Information Science*. Vol. 194 CCIS. 2011. DOI: 10.1007/978-3-642-22603-8{_}36.

[18] NGINX. Inc. *NGINX Announces Results of 2016 Future of Application Development and Delivery Survey*. 2016. URL: https://www.nginx.com/press/nginx-announces-results-of-2016-future-of-application-development-and-delivery-survey/.

[19] Sadequl Hussain. *Amazon Web Services*. URL: https://www.sumologic.com/blog/serverless-aws/.

[20] Vaishaal Shankar et al. "Serverless linear algebra". In: *SoCC 2020 - Proceedings of the 2020 ACM Symposium on Cloud Computing*. 2020. DOI: 10.1145/3419111.3421287.

[21] Davide Taibi, Josef Spillner, and Konrad Wawruch. *Serverless Computing-Where Are We Now, and Where Are We Heading?* 2021. DOI: 10.1109/MS.2020.3028708.

[22] Liang Wang et al. “Peeking behind the curtains of serverless platforms”. In: *Proceedings of the 2018 USENIX Annual Technical Conference, USENIX ATC 2018*. 2020.

[23] William Wong. “VM, containers, and serverless programming for embedded developers”. In: *Electronic Design* 65.10 (2017). ISSN: 00134872.

[24] *Zappa*.

NewSQL and Cloud-native Database systems

Aratz Manterola Lasa
Vrije Universiteit Amsterdam
Amsterdam, NL 1081 HV
a.m.lasa@student.vu.nl

Tom Ebergen
Vrije Universiteit Amsterdam
Amsterdam, NL 1081 HV
t.g.ebergen@student.vu.nl

Ean-Dan Tjon-Joek-Tjien
Vrije Universiteit Amsterdam
Amsterdam, NL 1081 HV
e.tjonjoektjien@student.vu.nl

Bojana Arsovska
Vrije Universiteit Amsterdam
Amsterdam, NL 1081 HV
b.s.arsovska@student.vu.nl

Abstract

SQL and NoSQL databases have been around for a long time, as they both offer solutions for data storage depending on the use-case/requirements. However, a growing number of applications require advantages offered by both SQL and NoSQL databases. The demand for a combination of these requirements has lead to the development of a new database type, NewSQL. Understanding how these systems are created could help us improve them. In this work we provide a literature study of NewSQL and cloud-native databases. The literature on the current technologies cover different aspects, such as performance, consistency, fault-tolerance, scalability etc. and the underlying mechanisms implemented within each aspect. We find that NewSQL databases deliver on the promise to offer scalable data storage with ACID properties.

1 Introduction

Databases have long been a required part of software and application development for storage needs. As applications have grown in terms of features offered and performance, so too has the need for more performant database systems. Two well known databases classes are primarily used depending on storage requirements, NoSQL databases and relational databases. Relational databases were some of the first databases, and impose ACID properties [1]. ACID properties ensure that transactions are **A**tomic, **C**onsistent, **I**solated, and **D**urabile. However, these rules make it difficult for relational databases to scale horizontally as storage needs increase. A table with terabytes of information will need to be sharded across servers, and running complex queries across shards can be time consuming [2]. NoSQL databases don't enforce ACID rules, and rather rely on a BASE model. The BASE model stands for **B**asically **a**vailable, **S**oft state, and **E**ventually consistent [3]. These properties make horizontal scaling for databases easy. However, a NoSQL database system can suffer from inconsistency, as transactions aren't always atomic or isolated to a single server [1].

Many enterprises and start ups have been migrating much of their infrastructure to the cloud [4, 5]. This migration has also motivated cloud providers to build many cloud-native services to simplify migrations and cloud management. Cloud-native services are designed specifically for programs that run in the cloud using cloud-provided infrastructure. The cloud migration and evolution of cloud-native services have played a large role in how databases are being developed. Originally, companies with social networking and other basic web based applications used NoSQL databases, simply for storing key-value pairs. NoSQL provided these companies with the fast performance their users desired. Companies with stricter requirements on consistency and data

Web Services and Cloud-Based Systems. Universiteit van Amsterdam. Group 15. June 4th, 2021.

integrity continued to use relational databases because data inconsistencies would have been too costly [2]. With the growth of the internet, these web-based data workloads have been labeled Online Transaction Processing (OLTP) workloads. OLTP workloads consist mostly of many users executing simple reads and writes to a few records at a time. The transactions do not do full table scans or complex distributed joins. Those types of transactions are considered more analytical and belong in a different workload class called online analytical processing (OLAP) workloads. OLTP workloads consist of random, short-lived, and repetitive transactions [6, 2].

With the growing popularity of cloud computing, and the rise of OLTP workloads, cloud hosting services like Amazon Web Services, Microsoft Azure, and Google Cloud Enterprise have started building services specifically to handle these short-lived, low impact, and repetitive transactions [2]. These are cloud-native services and are sometimes referred to as NewSQL because they can offer the scalability of NoSQL databases, with the consistency of relational databases. Google developed Cloud Spanner, Amazon developed Aurora, and Microsoft developed Socrates. Open-source options exist as well, VoltDB and Hybrid-Volt, Lightning HTP, and Cockroach DB. Many of these systems were built with the similar requirements in mind;

- Managing cross-replicated data with high availability and durability (Spanner, Aurora)
- Need for strong consistency in the presence of wide-area replication (Spanner, CockroachDB)
- support for a SQL-like query language (Spanner, Cockroach DB)
- Need support for cross row-transactions (Spanner)
- High-performance transactions (CockroachDB, Aurora, Socrates, CockroachDB)
- Satisfy the demand for an elastic, pay-as-you go model for OLTP databases (Spanner, Socrates, Amazon Aurora)

These requirements identify the common theme that whatever database is handling the data needs to satisfy requirements from both relational databases and NoSQL databases. The need for this type of database system comes primarily from Online Transaction Processing workloads. There is a need to keep data in a strict relational format to avoid consistency errors that can cascade into monetary losses. From this combination of requirements, the term NewSQL has become popular (a combination of SQL and NoSQL, and the fact that these databases are new). For this literature survey, we have asked the following question *"How does NewSQL deliver the promise of offering scalable data storage with ACID properties?"*. We will present a number of current database systems that qualify as NewSQL systems and analyze how they deliver on the promise of scalable data storage with ACID properties.

The rest of this literature survey will be organized as follows. Section 2 will discuss benefits and drawbacks of ACID and relational databases in more depth. Section 3 will discuss the current NewSQL technologies. Section 4 will compare and discuss the non-relational properties of the NewSQL databases like performance, fault-tolerance, consistency, scalability, and APIs. Section 5 will be a general discussion of each database. Lastly, Section 6 will address future research areas in NewSQL database systems.

1.1 Article selection strategy

For article selection, we used the following keywords in academic research websites Scopus and Google Scholar. We used articles from these search results and articles cited by the resulting papers.

newsql, oltp, oltp acid, oltp shared-nothing, shared-nothing, oltp consistency, stream acid, stream, shared-nothing, stream consistency, consistency, dbms acid, dbms shared-nothing, dbms consistency, newSQL API, Cloud-native, relational, databases.

In the end our Scopus query looked something like

```
(TITLE-ABS-KEY(newsql) OR TITLE-ABS-KEY(oltp) OR TITLE-ABS-KEY(oltp acid)
OR TITLE-ABS-KEY(oltp shared-nothing) OR TITLE-ABS-KEY(shared-nothing)
OR TITLE-ABS-KEY(oltp consistency) OR TITLE-ABS-KEY(stream acid)
```

```
OR TITLE-ABS-KEY(stream) OR TITLE-ABS-KEY(shared-nothing)
OR TITLE-ABS-KEY(stream consistency) OR TITLE-ABS-KEY(consistency)
OR TITLE-ABS-KEY(dbms acid) OR TITLE-ABS-KEY(dbms shared-nothing)
OR TITLE-ABS-KEY(dbms consistency) OR TITLE-ABS-KEY(newSQL API)
OR TITLE-ABS-KEY(Cloud-native) OR TITLE-ABS-KEY(relational)
OR TITLE-ABS-KEY(databases))
```

2 Relational and NoSQL databases

2.1 Databases History

The first relational Database Management Systems (DBMSs) came out in the early 1970s [1]. Over the years until the 2000s, relational DBMSs would grow in performance and capabilities until the arrival of the internet, when databases needed to scale-up, be on-line all the time, and start supporting concurrent users. A temporary fix for scaling up was the addition of middleware, which presents one logical database to an application, when in reality, the database is split over several physical nodes [2].

Overtime, middleware could no longer fulfill the requirements of web applications mentioned earlier. In addition, companies who relied on always on-line databases thought full-featured DBMSs like MySQL were overkill for simple data storage. Enforcing consistency and durability for each transaction wasn't the best way to store data that would simply be read again in a look-up query. In the late 2000s, NoSQL databases became popular as they could fulfill the requirements for growing online companies. As mentioned in the introduction, NoSQL databases ignore the ACID guarantees of a relational database, and instead provide eventual consistency. Programmers believed ignoring consistency guarantees would allow databases to properly scale and be highly available to support web-based applications. MongoDB is one of the most well-known NoSQL databases today, while MySQL and OracleDB are well-known relational databases [2].

2.2 Relational vs NoSQL

Relational databases enforce ACID guarantees. All transactions have the following four guarantees;

1. *Atomic*, transactions are performed as one single complete data operation.
2. *Consistent*, data is in a consistent state when a transaction starts and ends
3. *Isolated*, intermediate states of a transaction are invisible to other transactions
4. *Durable*, once a transaction is committed, the changes to the data persist.

In relational databases, data items are organized in formally described tables, so they are easy to access in multiple different ways. Since the data is stored in a predictable way, with unique instances of data in each row, the data is easy to summarize and report on. In addition, relational databases are easy to extend through schema changes. One problem with relational databases is that they are hard to scale as data needs increase. As mentioned in the previous section (2.1), better hardware will only improve performance so much before data needs to be sharded across nodes. Once data is partitioned across many nodes, joining tables and enforcing ACID properties starts taking more time with every transaction. Relational databases can only store data in tables as well, so any unstructured data can lead to a lot of complexity (i.e graphs or media such as photos and videos) [1].

NoSQL databases, on the other hand, are much more flexible when it comes to storing large amounts of data. NoSQL databases does not use structured query language (SQL) for data queries. Since NoSQL isn't relational, there aren't any foreign keys for running join operations. NoSQL doesn't guarantee ACID properties properties either, but rather the BASE properties mentioned earlier. The absence of these features give NoSQL databases the ability to easily scale across multiple nodes. The ability to easily scale horizontally using commodity software satisfies the needs of web-based companies that are continually collecting and processing data. NoSQL systems also offer high transaction throughput because ACID properties are ignored.

NoSQL systems have their disadvantages however. Since ACID properties are ignored, application developers can spend a significant portion of their development time writing code to handle

data inconsistencies [2]. For other companies, giving up strong consistency is not possible because data consistency and integrity is necessary to ensure the value of the underlying product (i.e financial systems that ensure bank account balances are always correct). Table 1 summarizes the differences between relational on NoSQL databases.

Table 1: Relational vs NoSQL

DBMS	Relational Databases	NoSQL
Structure	Data tables with fixed columns and rows	Unstructured data. Key-Value, Document store, Graphs, etc.
Scalability	difficult to scale without writing complex middleware or affecting performance	Easily scalable with commodity hardware
Query	SQL is the typical query language	Depends on the database in question, but no consistent query language
Transactions	Transactions follow the ACID model	Transactions follow the BASE model
Performance	Low data throughput	High data throughput

3 Current technologies

NewSQL databases are relational databases that aim to provide scalable performance of NoSQL databases, while also providing ACID guarantees for OLTP workloads. To achieve these characteristics different architectures and some smart mechanisms, such as sharding of data, have been implemented. In this section we will cover the current technologies within NewSQL and describe each. These implementations form the basis for our evaluation of the non-relational properties such as Performance, Scalability, and Consistency in Section 4.

Spanner is a globally-distributed database created by Google in 2016 which has high availability and is highly scalable. It replicates data across machines using Paxos leader election. The placement of machines is geographically diverse enough to withstand natural disasters. The distribution of machines provides availability and data locality. Data is quickly replicated and once it is written to, can be moved between servers and data centers in response to failures or request load [7]. Spanner is a shared nothing architecture, which means that nodes do not share memory or storage, which is the reason that Spanner is highly scalable [8]. Furthermore, Spanner has an API called TrueTime, which is a global clock that is highly accurate [7]. It allows for ordering of read and write transactions within the database. TrueTime provides non-locking read operations, based on the timestamp.

NuoDB is a distributed SQL database that provides the guarantees of ACID transactions. NuoDB provides high scalability, high availability, and no single points of failure [9]. The architecture consists of a transactional layer and a storage layer. The isolation of these two layers enables them to be scaled individually. Moreover, failure handling of transactions does not impact the storage and vise-versa.

CockroachDB is a distributed SQL database built on a transactional and a strongly-consistent key-value store. CockroachDB scales horizontally. It is inspired by Google Spanner and is open-sourced [10].

Socrates is a cloud-native SQL database, offered as a service by Microsoft. It separates computation and storage by using log replication to achieve availability, elasticity, and durability. These qualities are also isolated, just as in NuoDB [4].

Aurora is a database created for OLTP workloads with network traffic in mind. Since I/Os are spread across replicas, Amazon researchers treated the network as a bottleneck. Similar to Socrates, Aurora uses Log Replication to offer NewSQL qualities. Aurora provides both high

throughput and availability in a cloud environment. Just like Socrates, the separation of compute and storage enables Aurora to recover quickly from failures [5].

VoltDB is an in-memory relational database created, with Big Data workloads in mind. It is a shared nothing architecture, where each node shares a part of the data. VoltDB supports a lock-free concurrency protocol, which increases the performance, as reads do not conflict with writes. The database can be scaled in two dimensions by scaling up (vertical scaling) and scaling out (horizontal scaling) [3].

MemSQL/SingleStore DB is a distributed, in-memory database, that compiles SQL queries into C++ through code generation. It leverages lock-free data structures and Multi-Version Concurrency Control. This database also does not have a single point of failure, due to replicas created and partitioned across nodes [11].

TiDB is a database created for Hybrid Transactional and Analytical Processing (HTAP) workloads [12]. It uses a Raft Algorithm to store the rows and columns, which makes it efficient at reads and keeps the data consistent with the transactions.

CumuloNimbo is a relational database, which is based on HBase, an open-source distributed database running on top of Hadoop Distributed File System (HDFS) [13]. It is built on scalable and fault-tolerant components and uses a transaction manager. HBase organizes tables which contain rows referenced by keys and columns organised in column families. A set of rows referenced by key ranges are defined as Regions. The nodes in a particular Region are managed by a Region Server(RS). HDFS distributed file system is capable of storing large files. Composed of a master node (Name Node) and a worker node (Data Node). Name Node is in charge of keeping track of files stored in HDFS and Data Node stores the data of those files [14].

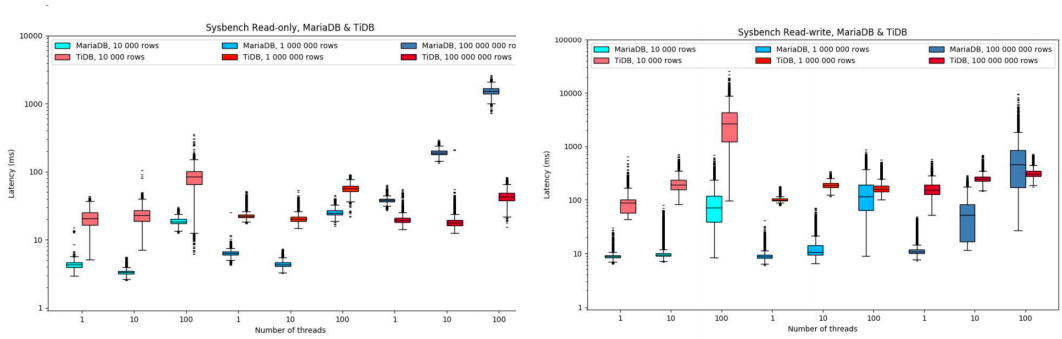
All of these NewSQL implementations have a few things in common. All NewSQL systems provide a SQL-like query interface, which is the primary mechanism for application interaction. Therefore, all NewSQL implementations support ACID properties for transactions. The distributed databases attempt to find a balance between replication of data and its storage overhead, while preventing data loss. Furthermore, due to replication of data, all of these implementations are distributed databases with no single point of failure. NewSQL has a non-locking concurrency control mechanism feature for real-time read transactions, which do not conflict with write transactions. The non-locking concurrency protocol can be achieved with one of the following mechanisms: Timestamp-based concurrency control, Optimistic concurrency control, or Multi-version concurrency control. On write transactions, locking protocols such as two-phase locking and Log Sequence Numbers are used to ensure consistency (these are detailed in Section 4.3). Lastly, NewSQL architectures provide higher per-node performance than available from traditional RDBMSs.

4 Non-relational properties

For more understanding of these relatively new technologies a comprehensive evaluation of the non-relational properties is needed. In this section we illustrate the similarities and differences between mechanisms that comprise NewSQL and cloud-native database systems. The performance provided by the database systems are first illustrated. Next, the fault-tolerance, consistency, scalability, and APIs of these systems are covered.

4.1 Performance

In this section two performance comparisons are introduced. The first comparison is between the NewSQL sharded database TiDB and the SQL database MariaDB. These databases were tested in three separate categories: load testing, complex queries, and performance in a realistic environment. The realistic environment test was measured as the average response time of queries executed on a server running a WordPress website. The database systems were evaluated in similar conditions since they were both running the same WordPress website and web stack, only the underlying database and connectors were changed. The complex query benchmark tested how well the databases handle singular large queries that may need to access several different tables to complete. The final web-stack



(a) MariaDB and TiDB read-only

(b) MariaDB and TiDB read and write

Figure 1: Average runtime of each workload on MariaDB and TiDB database in milliseconds. [16]

benchmark load tested the databases and query response times while supporting a real website with production workloads. More about the benchmarking setup could be found in [15].

Load testing
Sysbench is a benchmarking tool which enables parallel execution of queries. These queries were executed against databases of 10.000, 1.000.000, and 100.000.000 rows on varying number of threads.

Read
Figure 1a shows that MariaDB achieves lower latency than TiDB when tested on 10.000 and 1.000.000 row tables. However, 100.000.000 row tables tested on TiDB ended up outperforming MariaDB for all thread-configurations.

Read-write
In Figure 1b, MariaDB has lower latency than TiDB in most cases. Their performance gets much closer with 1.000.000 and 100.000.000 row databases running on 100 threads. MariaDB was also observed to be growing quicker than TiDB as the number of threads increased.

Complex queries
They use raw SQL queries that are run through a MySQL client. Reports of the execution time are used to measure the latency of TiDB and MariaDB. When testing complex queries, the results have shown that TiDB outperforms MariaDB with 3 to 4 times lower latency. Such results correspond to a query that returns 10 rows from the database and another one that enables a join between tables and selects values. In addition, queries that count the number of rows in a large table execute faster in TiDB in comparison to MariaDB.

In general, it cannot be clearly stated which database performed better. Concluding from the research paper [15], MariaDB achieves higher performance with small queries and databases. In most cases read and write workloads are executed with low latency in MariaDB. TiDB, however, provides the best performance for complex queries, large databases and high load. Nevertheless, read and write workloads had higher latency in most cases. Therefore, for specific workloads and queries NewSQL database like TiDB have the ability to achieve better performance than MariaDB.[15]

Performance Benchmarking of NewSQL Databases with Yahoo Cloud Serving Benchmark [16] also provides an analysis on NuoDB, MemSQL and VoltDB. These databases were evaluated on various workloads by implementing the Yahoo! Cloud Serving Benchmark (YCSB) benchmarking framework. The YCSB framework tests "key-value" and "cloud" serving stores. The YCSB benchamrking framework is used to compare different workloads which are suitable for analyzing the performances of NewSQL databases [16].

In line with Figure 2, loading the workload takes most time for NuoDB. In comparison to the other database, MemSQL is 14 times faster and VoltDB is 24 times faster. But the most extreme case was running workload E which makes NuoDB 8 times slower than the other two databases. NuoDB under-performed in high random read workloads with few updates. Comparing MemSQL with VoltDB, VoltDB slightly outperforms in workload A,B,C and D, where's in E they are equivalent.

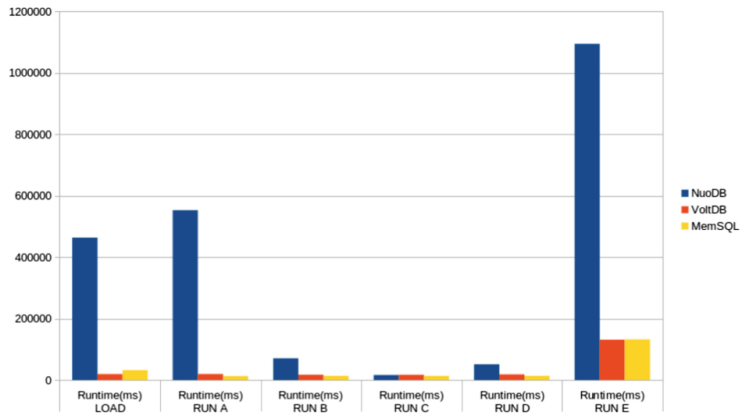


Figure 2: Average throughput of each workload on each NewSQL database in operations per second. A: mix of 50/50 reads and writes; B: 95/5 reads and writes respectively; C: Reads only; D: Inserting records and reading them immediately after afterwards; E: Reading within short ranges; F: Reading, updating and writing those changes back. [16]

Running the workload C, NuoDB closely beats VoltDB, by positioning itself in the range of VoltDB and MemSQL [16].

4.2 Fault-tolerance

As the number of machine grows, the probability of faults or byzantine failures increases in NewSQL databases. These databases are expected to provide high availability and fault tolerance as part of evolving customer demands explained in section 1.

Let's examine how VoltDB handles failures. Here, each transaction either rolls back or executes successfully and the data persists in the event of a failure. Transaction procedures are usually executed on partitions. Each node can execute transactions across multiple partitions. To achieve fault tolerance VoltDB ensures 'K-safety'. This is a mechanism that duplicates database partitions by creating multiple nodes based on the value of K. K is the number of replications. This makes the database fault-tolerant in case one or more nodes fail. Failures could also occur when one of the shards is down [17].

4.2.1 How do failures affect performances of a New SQL data store, CumuloNimbo?

The paper [14] analyses how failures affect the regular operation of CumuloNimbo. Their TPC-C benchmarking produced results by varying the number of nodes in different failure scenarios. Their first experiment was carried on 4/5 nodes for which three cases are taken into consideration: one region server failure, two region server failure and all region server failure.

Results of one Region Servers show that 44 seconds are required for the system to reconfigure itself with 4 nodes and 49 seconds with 5. This indicates the consequence if one Region Server fail, as all the regions in the server are distributed among other available RS. It is quite visible that the increase in latency is due to the new order transactions. According to [14] the average latency had an increase of 2% after the failure as a result of reconfiguration.

The second case is when two RS in different nodes fail. It takes 48 seconds with four nodes and 62 seconds with 5 nodes for a recovery. As expected, the reconfiguration time is slightly more than if the one RS fails. The reported average latency increase is 7% in four nodes.

If all region servers in one node fail, then the Data Node still remains. RS reconfiguration of four RSs takes 46 seconds and 52 for one more node. Such a failure results in a 6% latency increase. Initially, the replacement RSs do not contain a copy of the data and the Data Node of the failed ones. Thus the data needs to be requested. Then a copy of the corresponding blocks need to be

made in the newly reserved nodes. Some RSs can access a local replica without creating further delays [14].

4.3 Consistency

Consistency is of a huge importance when it comes to databases. It ensure that data it consistent after each transaction. Therefore it is important that NewSQL systems do not allow multiple transactions to operate on the same piece of data at the same time. Without consistency, many issues could emerge between users about the right version of the data written in the database. Fortunately, NewSQL has ACID properties, where C promises consistency [1]. There are two consistency mechanisms which are discussed in this section, namely two-phase locking and logs.

4.3.1 Spanner and CockroachDB

Spanner and CockroachDB ensure consistency by providing a two-phase commit (2PC) and strict two-phase locking (2PL). 2PL is concurrency control method which utilizes locks during transaction processing in order to preserve serializability and consistency. 2PC is an atomic protocol that instructs all the processes participating in the atomic transaction by deciding on whether to commit or abort the transaction. The coordinator asks the participants to vote on whether they can commit a transaction or not and makes a decision based on their vote [18].

Writes are serialize and ordered by the help of a timestamp protocol in Spanner and Cockroach. In Spanner, we have the TrueTime API, and Cockroach DB has a Hybrid Logical clock system. According to the Spanner paper, TrueTime returns a bounded time interval that is guaranteed to contain the time when the TrueTime API was invoked at the caller [7]. This time interval, along with the use of locks, can be used to provide serializability of writes in Spanner. In CockroachDB, each node maintains a hybrid logical clock which combines physical and logical time. The logical part of these clocks can provide causality tracking, while the physical time component can be strictly monotonic. Together these two properties can ensure two causally dependent transactions have timestamps that reflect their ordering.

In general, synchronized clocks can be used for avoiding communication between nodes in a distributed system. This happens on the basis of local computation. Each node can deduce the clock value of another node based on their past interaction. As stated in [19], *"TrueTime is a global synchronized clock with bounded non-zero error: it returns a time interval that is guaranteed to contain the clock's actual time for some time during the call's execution."* In Spanner and CockroachDB, a there is still a risk that the time intervals overlap. When this happens, it is impossible to know the ordering of transactions. Both Spanner and CockroachDB acknowledge this possibility when it comes to lease holders for leaders in Paxos and Raft. As leases are granted for a time interval based on the TrueTime/HLC systems, leases can potentially overlap. Spanner solves this by only allowing lease holders to abdicate a lease after the TrueTime API has determined that the lease period has ended. New leases can only be granted after the old lease period [7]. CockroachDB solves this by requiring each lease acquisition request to contain a copy of what is believed to the the current valid lease [10].

4.3.2 Amazon Aurora and Socrates

Amazon Aurora and Socrates take different approaches towards achieving consistency and concurrency. Instead of writing to replicated requests to either tables or a key-value store, Aurora and Socrates use a log to create a relational database. The log is stored and replicated in cheap storage services, and the database is created in a compute service which is updated whenever the log is updated. If a database fails, a new one can quickly be re-created by reading the log and applying the data updates [4, 5].

Since these databases use logs rather than timestamps, they don't have to deal with timestamp contention or locks. Instead each request is given a place in the log, naturally ordering all of the requests. This will monotonically advance points of consistency and durability as transactions are acknowledged. If a transaction doesn't immediately receive a place in the log, the transaction can just be submitted until it does get into the log. Logs are eventually sent to replicated storage nodes. If any storage node is missing a set of logs, it can just gossip with the other storage nodes to retrieve those

logs [4, 5].

Separating the database log from storage enables the separation of durability as and availability as well. Durability is implemented by the log, as the log can be used to recreate the database, and availability is implemented using storage, as fast storage systems are easy to replicate. Using fast and highly available storage is also a benefit to companies like Amazon and Aurora because those companies already have those services abstracted out as cloud-native services [4, 5].

4.4 Scalability

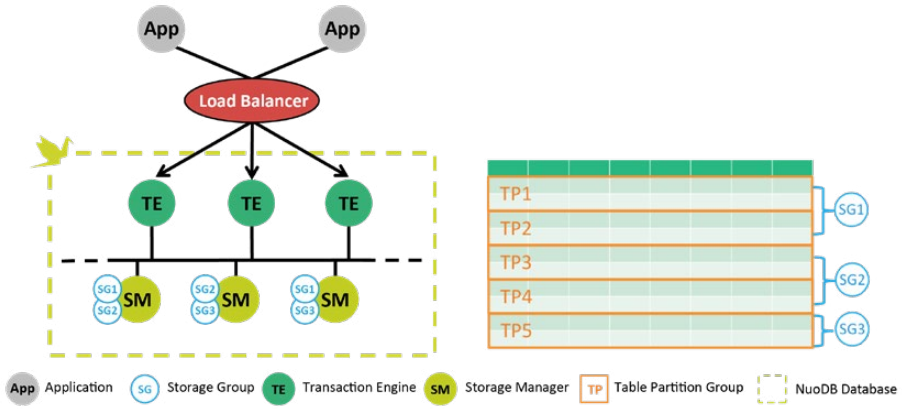


Figure 3: NuoDB architecture. Storage groups allows users to control the physical location of where the data is located and how many copies of that data are stored for redundancy, continuous availability, and separate processing purposes.

One of the main objectives of NewSQL systems is to offer scalability similar to that of NoSQL, and it seems that many systems manage to scale to very high throughput and requests [16, 10]. There are two ways to scale, vertically and horizontally. NewSQL prioritizes horizontal scaling, where they increase capacity by adding more nodes to the system. But this alone does not ensure good scalability of the system as a whole, and therefore, it is important to understand the internal mechanisms that NewSQL systems employ.

The most important mechanism that NewSQL uses to achieve great scalability is sharding. Most systems, including Spanner [7] and VoltDB [3], partition the data horizontally and distribute it across multiple nodes. Horizontal partitioning splits a relational table into groups of rows. Then, they add a layer on top of the indirection, which receives the client's requests, and decides where to apply the transaction, or from which nodes to get the data, transparently. The partitioning allows more concurrency in the requests, and to be able to manage higher volumes of data, by simply adding more nodes. They use two main techniques to assign partitions to nodes: range or hash. Through the range, there is more control because a central node decides how to distribute, while the hash distributes uniformly. The NuoDB case takes scalability one step further [9]. It uses a two-tier architecture as can be seen in Figure 3, which allows the transaction throughput and storage capacity to be scaled independently. There are some nodes (labelled TE) that are in charge of the transactional operations and keep all the data in memory, while there are other nodes (labelled SM) that are in charge of committing the transactions and permanently saving the data on disks. This allows the durability mechanisms not to interfere with transactions.

A second design decision which can offer great scalability is avoiding any major contentions regarding concurrency when the number of request are increased. To do this, NewSQL implements different concurrency control mechanisms. VoltDB runs a single thread on each partition, this way it can run everything serialized and doesn't need to use locks. However, most systems, including Spanner, CockroachDB, NuoDB or MemSQL implement Multi-version concurrency control (MCC or MVCC). In MVCC no locks are necessary, instead it treats all the data in a versioned way, that is, the data is not overwritten but new versions are generated. In this way there is no contention between a request that wants to read and another that wants to write.

Even so, this mechanism does not prevent write-write conflicts, and for this each system implements a different solution, some with locks, others with lock-free structures like in-memory queues.

Finally, it is also important to mention that several systems such as VoltDB or CockroachDB, use peer-to-peer architectures to distribute cluster metrics data or to replicate the data between nodes. This allows for greater scalability as there is no central node that receives all the load.

4.5 APIs

Application Programming Interfaces are interfaces for multiple software applications to connect with each other. As an example, you could think of a food ordering system using an API to process the payment of the client. In terms of NewSQL databases, the application uses the API provided by the NewSQL database to create, update, and delete data in the database. Next, the APIs that Google Spanner provides are listed.

Google Cloud Spanner has three available APIs for clients to connect to [20]. The first and best option according to Google is using gRPC, which is a remote procedure call system created by Google [21]. CockroachDB also offers gRPC as an option. It supports client libraries that are built in popular languages such as Java, Python, C++, C#, Go, etc. Spanner also has an RPC API if the client uses a language that is not supported by gRPC. Spanner also supports a standard Rest API. Google suggests users use the gRPC API versus Rest API as it has multiple advantages [20]. For each request you can cancel a request and you can set a deadline for a timeout, which is not possible using the Rest API.

The TrueTime service is an API that was implemented alongside Spanner which encapsulates a global clock [7]. The TrueTime API is used to determine a time interval for when a transaction has taken place or if it is yet to take place. This time interval is guaranteed to be bounded by a certain time uncertainty, varying from 1ms to 7ms. This API is very useful because Spanner uses Multi-version Concurrency Control. Due to the ordering of transactions, which is guaranteed with the timestamps, reads are consistent without blocking writes.

Most of the NewSQL databases in this report have Rest APIs for integration with client systems [20, 22, 23]. Some examples such as Spanner and CockroachDB have more options for integration by using gRPC or other Remote Procedure Calls. However, there is not a lot of literature on the APIs of NewSQL databases.

5 Discussion

The findings of this work are discussed in this section. We interpret the results and explain new understandings and significance about NewSQL and cloud-native databases. Design decisions taken with regards to consistency, fault-tolerance and scalability all affect performance. Therefore, we will discuss performance in these sections.

5.1 Consistency

Consistency is primarily achieved in two different ways. As mentioned in Section 4.3, NewSQL systems like Spanner and Cockroach DB implement a solution using timestamps and 2 phase locking for writes. Other systems like Amazon Aurora and Socrates use log replication. Log replication has it's benefits because write amplification is greatly reduced and consistency can be maintained at a high level throughout the whole NewSQL system. This also makes understanding the life-cycle of a transaction much easier. Since every transaction gets a log identification number, it's easy to reason about the order of all transactions. Log-based NewSQL databases also take advantage of compute and storage cloud-native services. Storage services simply store an ordered log while compute services can quickly combine logs to recreate a database. An issue with this separation, however, is that the architecture of these systems can be difficult to grasp, and it's not always clear what steps need to be taken before a log is finally committed to replicas, and if the replicas are database replicas or storage replicas.

In other NewSQL systems, consistency is guaranteed via a means of timestamps and repli-

cation is done via Paxos or Raft protocols. In earlier sections, we mentioned Spanner and CockroachDB do this. While these mechanisms work (to some degree), there seems to be a lot of communication overhead compared to a log replication system. Communication is created by client requests, Paxos/Raft leader election protocols, the 2 phase locking protocols, and the TrueTime API/Hybrid-Logical Clock. The Aurora and Socrates papers made less communication a priority in their systems and it shows. Timestamp mechanisms used in combination with Paxos and Raft also introduce split brain problems. When leader leases are dependent on certain timestamp intervals (which they are in Spanner and CockroachDB), it's possible leases, and therefore leaders, overlap. Such complexity is avoided in the Log replication NewSQL systems.

Next to the timestamp mechanisms are the two-phase lock and commit protocols which eventually write transactions to memory and assign timestamps to them. While two-phase commit protocols can be seen as noisy, and slow down performance of replicated databases, this may soon be an idea of the past. Barthels et. al show how to reduce communication overhead of Two-Phase Locking (2PL) and Two-Phase Commit (2PC) to make it suitable for scaling on thousands of cores and hundreds of machines. The core of their findings is that such results are achievable by using a Remote Direct Memory Access (RDMA). RDMA is a hardware mechanism which enables the network card to directly access parts or all of main memory [24]. This is a very interesting finding that could result in even more NewSQL systems, specifically at Google, as RDMA has become popular in recent years, and the Spanner paper was published almost 10 years ago. The main barrier, however, will be adding RDMA enabled hardware across all the physical machines of a cloud provider. There would need to be a significant increase in performance of the resulting NewSQL database to warrant such a large infrastructure change.

5.2 Fault Tolerance

Fault tolerance depends on the building elements of the database. For instance Spanner, although it is a NewSQL database, provides low fault tolerance. As mentioned in Section 4.3.1, Spanner prioritizes consistency. According to the CAP theorem, it must forsake network partitioning tolerance in order to achieve consistency and availability (CA) effectively.

Transitioning to a more fault tolerant database, VoltDB provides fault tolerance by replicating partitions once or multiple times. Although this will mechanism most certainly ensure better fault tolerance, it still has its downsides. This is due to the fact that it requires the database to utilize storage at least twice bigger than the original size. The storage size required for the replicas could be much larger, depending on the value of K for K-safety. Which is why this is not the most optimal solution for implementing fault tolerance.

From the gathered results, CumuloNimbo proved as one of the most fault tolerant databases. It benefits from its special fault-tolerant components, HBase and HDFS. Their mechanisms provide fast replacement of failed Region Servers (RSs) by storing Data Nodes for each RS which can be reattached to a replacement RS in case of failure. The impact of failures on the latency and performance depends the number of Region Servers that need to be replaced. As already seen in the Fault Tolerance section (4.2), the percentage of latency increases are quite low. It fits in a range of 2% to 7%. Luckily this ephemeral latency climb happens only once, which is after the failure has appeared. Afterwards the latency goes back to its initial value.

5.3 Scalability

Regarding scalability, NewSQL has not implemented any new innovative design to accomplish anything new. We consider that the same techniques found in NoSQL databases have been used, that is, horizontal partitioning. The only real difference from NoSQL is that NewSQL prioritizes consistency over performance. If previously SQL did not partition the table rows between different nodes, it was due to the lack of computing performance, not due to the lack of innovative designs. Thanks to improvements in networking speed, data storage and computational power, database engineers have been able to partition tables horizontally and implement a redirection layer on top, without sacrificing performance. Previously an intermediate layer to forward the queries to different nodes and coalesce query results across several partitions was too great penalty. That is why it should be noted that relational databases were first proposed in 1970. The same goes for the mechanisms they

use to resolve concurrency conflicts and avoid high contentions. Most NewSQL systems make use of MVCC or lock-free data structures, however, SQL databases (such as Microsoft SQL Server) have been using MVCC since 2005. That is, most of the concurrency control methods have already been implemented in SQL databases before. They have simply become more important now. Therefore, we consider that there is no genuine innovation by which NewSQL achieves great scalability.

The most innovative design, or at least the least standard, is the use of peer-to-peer architectures. This design allows them to scale without any reconfiguration and almost linearly, unlike more centralized architectures (e.g. master-slave). We have seen that several of the NewSQL systems such as NuoDB or CockroachDB have opted for this type of architecture instead of a centralized one. It is true that this has already been used in NoSQL databases like Cassandra, but even so, it is a very big contrast to traditional relational databases, which are strictly centralized. In addition, we consider that it is a feature that deserves to be mentioned, because it is not such a popular option within NoSQL either. Even so, it is important to explain that peer-to-peer is not the main piece of NewSQL, since this is mainly used to broadcast very light messages between all nodes, such as cluster metrics or information about storage capacity and network addresses. In addition, not all systems opt for it, large solutions such as Spanner have preferred more centralized designs.

In general, we do not consider that NewSQL has really and separated itself from relational databases. Rather, we understand NewSQL as the natural evolution of relational databases, due to improvements in technology that have increased computing and data transmission speeds. Even so, we believe that it is important to have created a new term "NewSQL" and to have made explicit the new solutions that offer ACID properties along with great scalability. We believe that this generates new illusions and attracts researchers to this topic. In addition, it also serves to make the industry see that traditional solutions are improving and that they may not need to migrate to a NoSQL solution to achieve high scalability and availability. Despite this, the NewSQL family of solutions is still very young, and a large part of the articles available are from the creators of the solutions themselves. Therefore, we have had a hard time finding neutral evaluations that study the performance, scalability, fault-tolerance and other properties of these systems under production workloads. Although the creators' articles detail the system with great precision, they almost never detail the weaknesses of their designs, and this does not offer a clear vision of the maturity of NewSQL. That is why we believe that third-party researchers are needed to analyze these new solutions in detail and to offer a more neutral perspective on the current state of NewSQL.

5.4 APIs

From the article search, not a lot of papers that cover APIs have been discovered. We hypothesize that this is because the Rest API vs RPC is a extensively studied design for APIs, outside of the domain of NewSQL databases. Google have provided multiple types of APIs for clients to connect with, which allows for flexibility of software design.

The TrueTime API does serve as an important breakthrough, as it removes clock uncertainty for transactions. Moreover it serves as an API that is performant to the extent not seen before.

6 Future research and trends

More and more companies use NewSQL solutions to implement OLTP. However, it is relatively young compared to relational databases and has not yet reached their maturity level. In addition, as new trends emerge and their use starts becoming popular, they will want to expand the functionality offered. Analyzing the last articles that come out on this topic, we have identified two main branches of future research or evolution that NewSQL solutions require: greater security, and more capabilities for analytical processing.

NewSQL databases were born from the need to have greater scalability while offering ACID properties. These needs mostly applied to Online Transactional Processing (OLTP). These systems were not designed for Online Analytical Processing (OLAP). For this, external tools or platforms are used that duplicate the data. However, today, there is a growing commercial interest in performing analytics and transactions on the same data, as it simplifies analytics in real-time. Today's big

solutions like Spanner or CockroachDB are optimized for OLTP workloads and do not offer dedicated out-of-the-box HTAP functionality. The difficulty of joining OLTP and OLAP in a single database lies in being able to optimize the internal representation for the two types of workload, and in not interfering with each other. That is why one of the two main branches of research and improvement that we have identified is on how to extend NewSQL databases to work with HTAP workloads. There are already projects that are generating solutions such TiDB [12] or F1 lightning [25], which is developed by Google to extend NewSQL databases like Spanner. Even so, there is still a lot of work to do, since these solutions are very young and there is not enough knowledge about them nor experience with them. So, there is a need to carry out experiments and to settle on optimal solutions. For example, it is not clear how well they overcome the trade-off between availability and freshness of data.

The second branch of research and improvement that we have identified is in security [26]. Today, databases have become the central elements of data management. That is why it is increasingly vital to ensure that systems offer Confidentiality, Integrity and Availability (CIA) properties. Over the years, relational databases have strengthened their security mechanisms. However, as we have mentioned, NewSQL is still a very young technology, and they do not offer the same fine-grained levels of security. In addition, these databases tend to be deployed in the cloud, which increases security and privacy risk. NewSQL systems that are cloud-native, such as Google Spanner or Cosmos DB implement reinforced security mechanisms. However, less popular solutions such as VoltDB or NuoDB do not offer access control mechanisms nor data encryption, to implement them it is necessary to make use of external tools. For example, in VoltDB they document that to encrypt the username and password it is recommended to use TLS or Kerberos connections.

This lack of security concern in NewSQL databases is because the top priority for NewSQL solutions is performance, and therefore they decide to sacrifice security. However, there are not yet enough studies conducted that analyze the extent to which security in these systems can reduce performance. Moreover, Spanner or CosmosDB offer high performance despite implementing fine-grained mechanisms. Therefore, we understand that in terms of safety there are three main points of work. The first is a deeper audit and analyzing more dimensions of security of the current systems and its consequent fix recommendations. The second is the study on the trade-off between security and performance, seeking improvements in mechanisms that do not sacrifice security. And finally, a study focused on vulnerabilities that may arise due to the specific nature of NewSQL databases.

7 Conclusion

In this report we have tried to answer *How does NewSQL deliver the promise of offering scalable data storage with ACID properties?*. First we have presented an introduction to NewSQL, giving the context in which it has arisen and we have broadly compared it with the two other alternatives that are NoSQL and relational databases. Later, we have introduced the main NewSQL solutions today like VoltDB, Spanner, CockroachDB, Aurora, and Socrates so that the reader can translate the theory into real products. Next, we have analyzed NewSQL in depth, dividing it into the different non-functional properties: performance, fault-tolerance, consistency, scalability and the design of the APIs. We consider these properties are the best taxonomy to understand NewSQL and how it manages to deliver what it promises.

NewSQL systems achieve a performance similar to NoSQL, sometimes even surpassing them. To offer strong consistency along with great scalability, they make use of log replication and distributed replication using consensus algorithms. Furthermore, there is a great opportunity for improvement by making use of new hardware technologies that reduce the round-trip-time needed to implement distributed locking. In general, NewSQL systems prioritize consistency over higher fault-tolerance, since they do not implement eventual-consistency. The vast majority offer fault-tolerance through data replication, but there is a trade-off through performance and fault-tolerance, especially when faults occur. Some solutions have chosen to outsource the fault-tolerance making use of HDFS and HBase that are in charge of replacing the failed nodes automatically. Regarding scalability, the systems implement horizontal partitioning and make use of lock-free concurrency control mechanisms such as MVCC. However, neither of these designs is genuinely innovative, as NoSQL systems already use it. However, a design choice of some NewSQL

that is more unconventional is the use of peer-to-peer architectures for transmission of thin data between nodes.

NewSQL is still a relatively young area, and there is work and research to be done. There are two main subtopics of future research. One is an investigation to improve the security mechanisms of NewSQL without sacrificing the performance required by databases. Another is to investigate how to support analytical queries together with transactional ones without sacrificing overall performance. This would help NewSQL systems support not only OLTP workloads, but also HTAP workloads.

References

[1] Nishtha Jatana, Sahil Puri, Mehak Ahuja, Ishita Kathuria, and D. Gosain. A survey and comparison of relational and non-relational database. *International journal of engineering research and technology*, 1, 2012.

[2] Andrew Pavlo and Matthew Aslett. What's really new with newsql? *ACM Sigmod Record*, 45 (2):45–55, 2016.

[3] Geomar A Schreiner, Denio Duarte, Guilherme Dal Bianco, and Ronaldo dos Santos Mello. A hybrid partitioning strategy for newsql databases: The voltdb case. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, pages 353–360, 2019.

[4] Panagiotis Antonopoulos, Alex Budovski, Cristian Diaconu, Alejandro Hernandez Saenz, Jack Hu, Hanuma Kodavalla, Donald Kossmann, Sandeep Lingam, Umar Farooq Minhas, Naveen Prakash, et al. Socrates: the new sql server in the cloud. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1743–1756, 2019.

[5] Alexandre Verbitski, Xiaofeng Bao, Anurag Gupta, Debanjan Saha, Murali Brahmadesam, Kamal Gupta, Raman Mittal, Sailesh Krishnamurthy, Sandor Maurice, and Tengiz Kharatishvili. Amazon aurora: Design considerations for high throughput cloud-native relational databases. In *SIGMOD '17: Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1041–1052, 05 2017. doi: 10.1145/3035918.3056101.

[6] Rick Cattell. Scalable sql and nosql data stores. 39(4), 2011. ISSN 0163-5808. doi: 10.1145/1978915.1978919. URL `https://doi.org/10.1145/1978915.1978919`.

[7] James C Corbett, Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, Jeffrey John Furman, Sanjay Ghemawat, Andrey Gubarev, Christopher Heiser, Peter Hochschild, et al. Spanner: Google's globally distributed database. *ACM Transactions on Computer Systems (TOCS)*, 31(3):1–22, 2013.

[8] Life of Cloud Spanner Reads Writes. Technical report, Google, 2021. URL `https://cloud.google.com/spanner/docs/whitepapers/life-of-reads-and-writes`.

[9] NuoDB Architecture. Technical report, NuoDB, 150 Cambridgepark Drive Cambridge, MA 02140 United States, 2020.

[10] Rebecca Taft, Irfan Sharif, Andrei Matei, Nathan VanBenschoten, Jordan Lewis, Tobias Grieger, Kai Niemi, Andy Woods, Anne Birzin, Raphael Poss, et al. Cockroachdb: The resilient geo-distributed sql database. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1493–1509, 2020.

[11] Single Store Design Principles. Technical report, SingleStore DB, 2021. URL `https://docs.singlestore.com/v7.3/key-concepts-and-features/distributed-architecture/design-principles/`.

[12] Dongxu Huang, Qi Liu, Qiu Cui, Zhuhe Fang, Xiaoyu Ma, Fei Xu, Li Shen, Liu Tang, Yuxing Zhou, Menglong Huang, et al. Tidb: a raft-based htap database. *Proceedings of the VLDB Endowment*, 13(12):3072–3084, 2020.

[13] B. Kemme I. Brondino J. Pereira R. Vilaca F. Cruz R. Oliveira R. Jiménez-Peris, M. Patiño-Martınez and M. Ahmad. Cumulonimbo: A cloud scalable multi-tier sql databas. In *Proceedings of the 2015 Computer Society Technical Committee on Data Engineering*, pages 73–83, 2015.

[14] Ainhoa Azqueta-Alzúaz, Marta Patiño Martinez, Valerio Vianello, and Ricardo Jimenez Péris. Fault-tolerance evaluation of a new sql database. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 81–86, 2018. doi: 10.1109/EDCC.2018.00023.

[15] Mathias Johansson and Jonatan Röör. *Performance comparison between NewSQL and SQL: Sharded TiDB vs MariaDB*. PhD thesis, 2020. URL `http://urn.kb.se/resolve?urn=urn:nbn:se:his:diva-18712`.

[16] Irina Astrova, Arne Koschel, Nils Wellermann, and Philip Klostermeyer. Performance benchmarking of newsql databases with yahoo cloud serving benchmark. In *Proceedings of the Future Technologies Conference*, pages 271–281. Springer, 2020.

[17] Akash Budholia. Newsql monitoring system. 2021.

[18] Two-phase commit protocol, Mar 2021. URL `https://en.wikipedia.org/wiki/Two-phase_commit_protocol`.

[19] Eric Brewer. Spanner, truetime and the cap theorem. Technical report, 2017. `https://storage.googleapis.com/pub-tools-public-publication-data/pdf/45855.pdf`.

[20] Overview of APIs and client libraries. Technical report, Google, 2021. URL `https://grpc.io/`.

[21] gRPC, 2021. URL `https://cloud.google.com/spanner/docs/whitepapers/life-of-reads-and-writes`.

[22] NuoAdmin REST API. Technical report, NuoDB, 2021. URL `https://doc.nuodb.com/nuodb/latest/api/rest`.

[23] 5.5. Using the VoltDB Deployment Manager REST API. Technical report, VoltDB, 2021. URL `https://docs.voltdb.com/v7docs/AdminGuide/DeployRest.php`.

[24] Claude Barthels, Ingo Müller, Konstantin Taranov, Gustavo Alonso, and Torsten Hoefler. Strong consistency is not hard to get: Two-phase locking and two-phase commit on thousands of cores. *Proceedings of the VLDB Endowment*, 12(13):2325–2338, 2019.

[25] Jiacheng Yang, Ian Rae, Jun Xu, Jeff Shute, Zhan Yuan, Kelvin Lau, Qiang Zeng, Xi Zhao, Jun Ma, Ziyang Chen, et al. F1 lightning: Htap as a service. *Proceedings of the VLDB Endowment*, 13(12):3313–3325, 2020.

[26] G Dumindu Samaraweera and J Morris Chang. Security and privacy implications on database systems in big data era: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 33 (1):239–258, 2019.

Comparing microservice architectures used in the Internet of Things for homes and cities

Tom Siebring (12844519)
Faculty of Science (FNWI)
University of Amsterdam
Science Park 904, 1098 XH
tom.siebring@student.uva.nl

Auke Schuringa (11023465)
Faculty of Science (FNWI)
University of Amsterdam
Science Park 904, 1098 XH
auke.schuringa@student.uva.nl

Jelle Stoffels (10803130)
Faculty of Science (FNWI)
University of Amsterdam
Science Park 904, 1098 XH
jelle.stoffels@student.uva.nl

Abstract

The Internet of Things is one of the final steps in the adoption of the internet. Any device that can perform measurements and communicate these with other devices can be considered to be part of it. As promising as it might sound, many difficulties exist as well. For example, the manner at which these devices communicate, or can see other devices. To assist in this purpose, design patterns from other fields might prove to be useful. One particular useful approach is the microservice architecture. Even though its origin and general use case is very different from that of IoT applications, the general setup of distributed devices/services that communicate internally is a very important similarity. In this paper we investigate i) how these concepts overlap and ii) how to apply these to two IoT applications: smart homes and cities.

1 Introduction

To compare microservice architectures used in the Internet of Things, first we discuss these concepts separately. Next, the two are compared, after which we specify the research goals. As the title suggests, we are particularly interested in the application of microservice architectures in the Internet of Things for homes and cities. Therefore, in the Literature section, we dive into these concepts separately. Subsequently, we discuss microservice architectures for smart homes and cities, and investigate their similarities and differences.

1.1 The Internet of Things

The Internet of Things (IoT) is a broad concept in which physical and virtual objects are interconnected. Usually, the Internet of Things considers devices connected to the internet directly, or indirectly, for example when connected to an internet device via for example Bluetooth. These devices can range from cars to weather stations, as long as they can communicate with other devices in some form. As the term Internet of Things is not defined specifically, one can also regard the network itself as the IoT.

Note that the hardware of these devices can differ significantly. For example, a self driving car is equipped with hardware than can handle computationally intensive tasks, whereas the humidity sensor

from a weather station is solely equipped with the ability to create data and send it to a receiver. Some are online continuously, whereas others are switching between states. Also mode of communication differs based on purpose, location, brand, permissions and many other factors.

Due to the heterogeneity of IoT devices, a service connecting these devices is of vital importance. These services are called middleware services, and their tasks include [10] (i) to provide modelling abstractions of real-world IoT devices and sensor systems, (ii) enabling search and discovery of these devices and their resources by applications and services, (iii) providing unified APIs and protocols for historical and (near) real-time sensor data access.

1.2 Microservice Architecture

The microservice architecture (MSA) is an approach to software development that has been taking over the monolithic approach in the past ten years.

1.2.1 Monolithic code

The monolithic approach for application has long been the default. There may be multiple definitions of the monolithic approach, with small differences. In this case it is taken [7] that it consists of a single repository encompassing multiple projects, the code is available to all engineers with access to the large repository, all commits are eventually done to the head branch, or master branch, of the repository. Furthermore, all dependencies are in the monlithic repository itself, and there is a standard set of tools with which the developers interact with the code in the repository.

It has many advantages [13], for example there is a 'universal truth' for the some of the files application rely on. All uses of a certain script or part of the application is available from within the same monolith, the same repository, of code. When a change it made to code or to an API, this makes it easy to reach all code where this API is used, so that these can be updates. Whenever some code is updated, all parts of the monolith will immediately use this new updated code, eliminating the need to explicitly make the a program use a new version of an external updated program. Furthermore, monolithic code encourages high quality code, and forces engineers to take note and learn from code and methods others have used.

The monolithic approach to programming and applications has not only upsides [13]. The complexity of the monolithic code base increases over time, making it more difficult to contribute code. This becomes more difficult because one is less able to to have a complete picture of the entire code base, which makes it more difficult to be aware of where updates need to be made to for example support changed output of an API, or replace a deprecated API with a new one. Other problems include that of tools and code health.

1.2.2 Move to microservice architectures

In the modern age, cloud computing is very popular and has become more and more important. With this, and the problems with cloud computing listed earlier, it is often advantageous to move to microservice architectures instead of a monolith application. For software this has special challenges [8]. Due to the nature of the IoT, a monolith is often not possible at all and a microservice architecture (MSA) is used.

In general, MSAs are a type of Service Oriented Architecture (SOA). Overall there a number of characteristics of MSAs [10]. The first is that the components work together as services, they can be upgraded, replaced, and managed individually. They can even be tested individually by mocking input and comparing the output against the expected output. The second is that they are often organised around businesses, which means that every component has some function related to some business. The third is that they handle input in a smart way, but give the output in a usually simple format so that other services can make use of this data to perform more tasks. MSAs usually are decentralized, which means there is not central of control, and most devices work together independently, which is also the case of the data handling. The data os the MSA does not live on a single device, but on different devices, and is passed to devices when needed for their tasks. Finally, MSAs makes it easy to evolve the system over time and with that makes it easy to improve the system and increase the capabilities and abilities.

When MSAs are correctly implemented, the network is scalable and resilient, which means the network is able to grow and add more devices easily, or even duplicate services in order to handle a higher load.

1.3 IoT vs MSA

Clearly, the MSA is a top down approach, where a potentially monolithic application is split up into many separate microservices. IoT on the other hand is bottom up. Namely, there are different types of devices with varying hardware, purpose, location and method of communication. In contrast to MSAs, which often have a general purpose within a shared environment, IoT lacks these properties. Nonetheless, both consist of distributed services operating independently, which allows for merging the lessons learned of these fields.

A paper investigating how patterns and best practices from the MSA approach can be applied to the IoT can be found here [3]. A paper discussing an agile methodology that i) contemplates the main characteristics of the IoT and ii) guides the development of appropriate software solutions based on microservices architectures to manage IoT environments, acknowledging the serious difficulties that microservices imply, can be found here [4]. Note that before microservices became the leader in IoT applications, Service Oriented Architectures were the default approach. A paper discussing a particular form of IoT network, namely one consisting of wireless networks, and the corresponding MSA and required middleware for this, can be found here [2].

1.4 Research Goals

After the introduction on the Internet of Things and microservice architectures the research can be defined. This paper looks further into the use of microservice architectures in the IoT. Internet of Things is used in a many areas nowadays, of which the setting at home (a more personal setting) and the larger setting of in a city or country (a public setting) are the two most important.

The papers looks into how the microservice architectures used for the IoT in homes an cities differences from each other. It also highelights the similarities between the two. Part of the research question are a number of sub questions. First of all a definition needs to be made of what exactly a ‘smart home’ and ‘smart city’ is, where ‘smart’ is used to note that the Internet of Things is being used here. A second subquestion is on what microservice types are necessary or used for the IoT. The third and final question is on how these difference types of types interact with each other. Having these questions answered, more can be said on the differences and similarities between ‘smart homes’ and ‘smart cities’.

The study is designed by first finding the topics in which the research will be done. For each topic literature is found. Good literature is found by taking candidates and readin the abstract, and the conclusion and deciding whether it fits well with the research question that is being looked into.

2 Literature

In order to answer these questions, a literature study has been conducted. The papers used give a clear understanding on how different microservice architectures suit both the similar and different needs of homes and cities. Different perspectives have been used in order to gain a broad understanding of the requirements that these IoT-environments have. Firstly, different use cases are presented that contextualize IoT in the modern world. Secondly, the relationship between IoT and different microservice architectures will be explained by investigating these architectures and identifying each their advantages and weak spots. Thirdly, a comparison is made between microservice architectures designed for homes and cities.

2.1 The IoT for homes

An important setting where the IoT is used is in homes, which can be seen as private settings. There are a number of important aspects about the use of IoT in these settings that make it different from other uses. The first is that devices are usually closer together, and thus communicate on a smaller range than in for example a city. This smaller range makes it possible to use technologies for communication that work on a shorter length, for example Bluetooth. The smaller distances also

bring about a shorter latency. The network is not as decentralized as other networks, as what the devices do it usually orchestrated by a central point, like a smart phone or a device like Alexa or Google Home, which are again controlled by humans.

An aspect of these network is that they are setup by individuals most of the time, which means they are usually not perfect, and have loss of information, bandwidth, or signal, which can make then operate under what they should be able to operate at. Networks at larger places, such as companies, which have more money available for the design of a network than individuals, often perform more optimal, which affects the use of the network.

Privacy is also an important aspect of the devices, but since this section discusses the network in the home itself, and not connections to the outside world, this is not discussed in this section. The section on the IoT for cities looks further into privacy.

2.1.1 Interoperability

Problems that network may face is interoperability of the devices. In a home setting, there is usually a variety of brands used, which sometimes have different ways of communication. This causes differences in how communication can be established, and the type of data that is transferred between devices. A way to solve this problem is by using standards that are industry wide accepted. This means that all devices that want to belong to a certain network should adopt the standard, and adjust the device to only communicate using certain protocols. This allows individuals to have different brands, but still have these different brands communicate with entry points through which the user communicates with the devices. Such an entry point is for example Alexa from Amazon, or a smart phone. However, there not always clear communication protocols, especially since the IoT is not a very clearly defined term.

Manufacturers of devices that are made to be attached to a network usually follow whatever communication protocols the most popular devices have. If for example Amazon Alexa is a popular device, it will be made sure that the device can at least communicate with tha device. Adding to the that, there are brands that have grown to such a size, that they can afford to not have good interoperability with devices from other brands, they aim for a vendor lock-in. One such example is Apple, which can mostly only be used with devices from that same manufacturer (Apple). This can make it difficult on purpose to communicate with devices from other brands. This creates an effect where a user with many devices from a certain brand is forced to buy more devices from this brand, and is thus discourages to try out other brands. This is an effect that only increases as the user buys more devices.

2.2 The IoT for cities

The IoT is can be used on a larger scale, for example at that of a city, which can be seen as a public setting. On this larger scale there are a number of technical challenges. The first challenge is the distances over which devices should work. While technologies like Bluetooth could work for private IoT networks, on a larger scale technologies like Wi-Fi, IEEE 802.11p, GSM, LTE, and 5G networks can be considered [12] [16], which all have different properties, for example the GSM network is slow, whereas the 5G network is fast. However, the GSM network works over a larger area, where 5G works on only shorter distances. Next to these factors, latency and the usage of power are also important factors. Especially when a network is fully implemented in a city, the power usage can become a problem (especially since the number of devices increases over time), and choices have to be made on which technologies to use where.

The IoT networks on a home-scale can also be connected with a larger network, where the smaller network can be seen as sepearate devices in this larger network. There are different aspects [15] to the network that is implemented, which are uses like social welfare and environmental improvements, but also technical components like distributed generation and communication, as was previously talked about. One may also think of example like healthcare (related to welfare), weather, transportation, and survaillance systems. These ways in which network can be used need special handling, and the involvement of inhabitants in order to ensure privacy, as will be talked about later.

2.2.1 Data processing

Other than in a smaller setting such a home, the IoT in a city produces a lot of data. This data needs to be processed in some way. Every event in the city which is recorded with a device in the network, or from which the effects are noticed by a device in the network creates an event that needs to be processed. Using this output, devices, or the supervisors of those devices, which may be people, can respond to situations. These events come in a variety of types, and require a variety of output types as well.

To process these different data types, Complex Event Processing (CEP) plays an important role. It considers each measurement of a sensor as an event and handles it accordingly. Many CEP implementations use orchestration to handle these events, requiring some central entity that does the orchestration. An advantage of this method is that a governing body, such as the city board, can keep strong control over the system. However, an immediate disadvantage is the reliance of the system on the orchestrator, for example on performance. A distributed CEP is proposed here [14], which uses a choreography-based microservice architecture, allowing for greater horizontal scalability.

2.2.2 Privacy

The subsection of data processing brings up the question of privacy. How can the data used responsibly, without violating the privacy of civilians, and while still being able to act on certain situation.

To prevent misuse of the data, the network should not only be governed by companies or governments, but also by people. While companies and governments can be heavily involved in them, the inhabitants of a city should be at the heart of the decision making process when it comes to these IoT networks. The implementation of IoT network in cities, to created so-called 'smart cities' is nowhere near completion, and government are currently mostly working on pilot projects to see how a 'smart city' can be shaped best. It is shown that that early smart cities not only improve technology usage and infrastructure with the use of the IoT, but also show that an enhanced involvement between citizens and governing bodies is occuring [5]. It is shown that there is an increase in innovation and active involvement of citizen in setting up projects. However, it is clear that not all inhabitants of a city equally participate in the new technology, which is due to a variety of reason including income and background. Intermediate organizations can help to give more citizens insight into the processes, and allows for more citizens to be taken into account when making decisions [5].

This does not always go well however, while official statements from governing bodies often remark that new technologies and new uses of technologies in networks are done for the people and in cooperations by the, the actions often favor the private industry, and with that the privatization of the public sector into the private sector [11]. Besides, many cities seem to be not completely sure how to fit in the IoT, since many project start as pilot projects and end as pilot projects. It is important that citizens have a say over the use of new technology. However, there is no evidence that moving control from the private sector to the public sector will prevent citizens from being more 'controlled' by new developments. What may work are urban labs, where citizens can come together to talk about and come up with solutions for problems, and how to make the environment and their city better inhabitable. This could create another democratic layer under the local governing bodies, but before the private sector [11].

2.3 Microservices

Any IoT environment uses three types of microservices, as any smart home or smart city depends on a network of devices that have a need to communicate with each other and cloud-based services [9].

The first type of microservices is **device microservice:** The applications that run on the devices locally. These microservices are responsible for local functions, such as reading local sensors, accepting user input and managing storage. It is desirable to have these microservices run locally, because it decentralizes load as the necessary computing power is spread over many small processors that individually compute their respective operations. A second advantage of device microservices is that in case of a disruption within the network, sensor- and user input logs can be stored locally, so that after connection to the network is restored, normal operations can continue without any missing data.

The second type of microservice is **gateway microservice:** gateways handle the communication between devices and cloud-based services. They ensure that the intended data is sent and received and secure data streams from reaching unintended destinations and reaching destinations in redundant or overwhelming amounts.

The third and last type of microservice is the **service microservice:** This term describes the cloud-based microservices that enable the processing and generating of relevant data. An example would be a weather-forecasting application that uses the temperature and humidity readings from many IoT sensors to calculate a forecast and presents in via a web service.

2.4 Microservice architectures

After establishing an understanding of the different types of microservices necessary for the IoT, this section will dive into the ways that these microservices interact with each other within various architectures that have been proposed by others in the past.

2.4.1 Microservice architecture for smart homes

A smart home is an entity that consists of a collection of connected devices and a gateway that serves as a hub [1]. The paper by Bao et al describes an architecture in which devices do not communicate directly, but via an IoT Message Bus. This message bus is more commonly known as a hub in which a collection of microservices handle communication as a gateway. The devices gather data by their sensors or user input. This data is then used by the microservices running on the hub.

An example would be a presence sensor that only has the capability to give a 'true' signal when the hardware sensor detects a presence in the room. When this signal is received at the hub, a microservice X may use this signal to distinguish if the presence is a security threat or not. It may call to a microservice Y that keeps track of a person's smartphone WiFi connection. When the person is connected to the home WiFi, microservice X can then set security threat as false and set home-owner presence as true for a specific room. These states can be used for many applications such as turning on and off lights automatically and thus reducing energy consumption.

The smart home microservice architecture described by Bao et al thus uses a intranet-type concept that enables smart features for a single home, without the need to communicate with third parties. It does, however, centralize the computing power to a single device that acts as a hub. This has the disadvantage of relying to a single device for all IoT functionalities. In our opinion, this disadvantage is overcome by the advantage of cost saving in individual devices, as it enables smart home features whilst utilizing 'dumb' devices that are single-purpose, such as the aforementioned presence sensor.

2.4.2 Microservice architecture for smart cities

As smart cities have different use cases and scales, a different architecture is needed to suit their requirements. Infrastructure is on of the greatest beneficiaries of IoT technology. Gathering data from an entire city's traffic, however, sums to a vast amount of data. This is a smart city's greatest hurdle: being able to gather, distribute and process vast amounts of data. De Iasio et al [6] describe an architecture that copes with this problem in the following way. In their proposed architecture, IoT devices, such as traffic monitoring sensors, route their data streams through an IoT Broker. This is a gateway microservice that manages data streams by aggregating matching data and dropping redundant data. This has the purpose of not overloading any microservice down the line. Various other microservices in turn distribute the data to the relevant servers on which applications can process the data.

The main advantage of this architecture is the fact that data streams arrive at the processing microservice in predictable and digestible batches. This ensures that the application server is never overloaded. This does mean on the other hand that entities that control these applications do not have control over the early stages of these data streams. A IoT Broker could withhold data that it flags as redundant, but could actually be useful for certain applications down the line.

An example use case would be the implementation of smart traffic lights. IoT sensors enable the count of vehicles throughout a city. When a large number of vehicles is being detected on a highway in the direction of the city centre, an application could make way for the upcoming traffic in advance

by giving priority to vehicles currently waiting at traffic lights on the through-ways of the city centre. Once the large stream of outer-city traffic arrives, these through-ways have cleared up.

Another use case could be forecasting energy demand, potentially on district level. We are transitioning from stable to more intermittent energy production, and from a fossil fuel to electricity based energy system. Therefore, making accurate energy demand predictions will become more valuable. By combining for example data on the weather, local humidity and temperature, traffic information and district characteristics, demand for electricity can be predicted more accurately. As electricity prices become more volatile, this could reduce the city's electricity bill.

2.5 Comparison

Both smart city and smart home technologies rely on distributed devices that need to be connected in some form. In the previous subsection we discussed properties for either one. Now, we look more specifically at what similarities and differences exist between the technologies and how that reflects to the requirements of their microservice architectures.

2.5.1 Similarities

Both smart homes and cities deal with different type of applications in their respective IoTs. Both try to combine these heterogeneous data streams at some central place, either the user interface or the city's data centre. Therefore, in both cases, reliable transmission of data is important. Also, features such as device discovery are important for the success of either application. However, the most important similarity between these two technologies is their end goal: predicting (human) behaviour. Many AI applications that are developed these days try to automate away human actions that could not be automated before. The reason this became possible is that there now exist tools, such as deep learning, that can do this in specific situations.

There are two important consequences. Firstly, the AI tools currently developed to better understand humans will most likely be applicable to other fields as well. Examples of this could be traffic predictions, or a climate control scheme suggestion from the smart home to the user. Secondly, many new devices are equipped with hardware that can perform simple AI calculations. The main reason is that these applications can henceforth learn user behaviour such as to increase the user experience. Of course there will also be an increasing number of devices, vehicles for example, that have powerful hardware to do more demanding AI calculations. This increasing availability of AI calculation power, which has an important use case in understanding human behaviour, might therefore contribute as well to the development of the smart home and city. Because, as mentioned, both technologies have the shared goal of predicting human behaviour.

2.5.2 Differences

Two important differences between smart homes and cities is the number of devices that need to be connected and their heterogeneity. Cities can easily contain millions of devices, whereas the smart home is limited to at most 100 in most cases. Furthermore, the range of device purposes in a city is much bigger, as any allowed device that measures data with potential economic, scientific or societal benefit can most likely be found in some city. Devices in smart homes are mainly oriented on home or building owners and their day to day activities. The variation is therefore much smaller and the user interface more important. In contrast, smart city applications are mainly oriented at local governments or businesses, which can handle much lower level interfaces, such as API endpoints.

Downtime is another topic for which the negative effects will have a much larger impact on smart cities than on smart homes, as cities are utilized much more hours per day than smart home applications. As a consequence, smart home architectures can be much more centralized, whereas smart city architectures should be more decentralized and scalable. Of course the extent to which highly depends on the type of application it is used for. Having public transportation information correct and ready during rush hours has higher priority than measuring correct humidity levels.

Lastly, as cities cover a much larger area than homes, and require significantly more computing resources, it would make sense for a city to have a cloud provider handle the data gathering and processing, potentially at multiple places in or around the city. In contrast, smart homes usually

internally have the required computing power to do all calculations locally. Therefore, data streams need not be connected directly to the internet, increasing privacy, security and user experience.

3 Discussion

There are a number of important discussion points that can be talked about regarding the finding of the report.

3.1 Definitions

The most important point of discussion is the definition of words used in the reports. The Internet of Things, as noted before, is not a clearly defined term, and people have different opinions on what it is exactly. However, often there is some consensus on what the Internet of Things does and how it works, but there not be consensus on when one talks about the IoT, and when one simple talks about a few connected devices.

For example, one may have two devices connected with each other and communicating, it is not entirely clear of this an Internet of Things, or if this is simply 'two connected devices'. Do we need ten devices, or twenty before we call this an IoT implementation? Or perhaps some data needs to be moved through the internet before something like this called the Internet of Things. Overall, this shows how different interpretations are very possible for the term.

In contrast, a microservice architecture is more clearly defined. This is because there have now been many implementation, it is very actively used, and there is a good consensus on what a MSA is.

3.2 Extremes

A point of discussion on the differences and similarities are the extreme cases. This is somewhat related to the previous subsection on definition. For the findings of the report to be correct, the report assumes that a house is small, and that a house does not contain many devices, and thus that the complexity of the network is not very large. However, it is entirely possible for a house to be large.

The same point can be raised on the amount of data. An important point raised on the differences for the MSAs for smart cities and for smart homes is that the amount of data generated in a city is much larger than the amount of data generated in a home. However, equally large amounts of data may be generated for a single home as for a small city. This depends on how densily a city if populated with devices in the network, and what the devices measure.

4 Conclusion

One of the most difference between microservice architectures for the Internet of Things in smart homes and smart cities is the processing of data. Home give less data, and can be allowed to communicate together on a single IoT Message Bus for communication, while the networks in cities require filtering of data to not overload systems.

The consequences of a malfunction is also very different between MSAs for cities and homes, this would have a larger impact on cities than on homes.

However, overall MSAs for IoT for homes and cities is very equal to each other. Each get input, each provide output, and the output eventually reaches the end user either directly, or through decisions that are with the data. Privacy is an aspect there that should be watched closely.

Acknowledgments

The authors of this paper would like to thank the coordinator dr. A.S.Z. Belloum and staff members Yuri Demchenko, dr. Z. Zhao, Saba Amiri, Reginald Cushing, and Onno Valkering of course Web Services and Cloud-Based Systems (course code 5284WSCB6Y) at the University of Amsterdam for their support, and information given during the course. The information proved to be a very valuable resource for the this report.

References

[1] Kaibin Bao, Ingo Mauser, Sebastian Kochanneck, Huiwen Xu, and Hartmut Schmeck. A microservice architecture for the intranet of things and energy in smart buildings. In *Proceedings of the 1st International Workshop on Mashups of Things and APIs*. ACM, December 2016.

[2] Ayoub Benayache, Azeddine Bilami, Sami Barkat, Pascal Lorenz, and Hafnaoui Taleb. MsM: A microservice middleware for smart WSN-based IoT application. *Journal of Network and Computer Applications*, 144:138–154, October 2019.

[3] Bjorn Butzin, Frank Golatowski, and Dirk Timmermann. Microservices approach for the internet of things. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, September 2016.

[4] Edwin Cabrera, Paola Cardenas, Priscila Cedillo, and Paola Pesantez-Cabrera. Towards a methodology for creating internet of things (IoT) applications based on microservices. In *2020 IEEE International Conference on Services Computing (SCC)*. IEEE, November 2020.

[5] Carlo Francesco Capra. The smart city and its citizens. *International Journal of E-Planning Research*, 5(1):20–38, January 2016.

[6] Antonio De Iasio, Angelo Futno, Lorenzo Goglia, and Eugenio Zimeo. A microservices platform for monitoring and analysis of IoT traffic data in smart cities. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, December 2019.

[7] Ciera Jaspan, Matthew Jorde, Andrea Knight, Caitlin Sadowski, Edward K. Smith, Collin Winter, and Emerson Murphy-Hill. Advantages and disadvantages of a monolithic repository. In *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*. ACM, May 2018.

[8] Miika Kalske, Niko Mäkitalo, and Tommi Mikkonen. Challenges when moving from monolith to microservice architecture. In *Current Trends in Web Engineering*, pages 32–47. Springer International Publishing, 2018.

[9] Kalliopi Kravari and Nick Bassiliades. StoRM: A social agent-based trust model for the internet of things adopting microservice architecture. *Simulation Modelling Practice and Theory*, 94:286–302, July 2019.

[10] Alexandr Krylovskiy, Marco Jahn, and Edoardo Patti. Designing a smart city internet of things platform with microservice architecture. In *2015 3rd International Conference on Future Internet of Things and Cloud*. IEEE, August 2015.

[11] François Mancebo. Smart city strategies: time to involve people. comparing amsterdam, barcelona and paris. *Journal of Urbanism: International Research on Placemaking and Urban Sustainability*, 13(2):133–152, August 2019.

[12] Yasir Mehmood, Farhan Ahmad, Ibrar Yaqoob, Asma Adnane, Muhammad Imran, and Sghaier Guizani. Internet-of-things-based smart cities: Recent advances and challenges. *IEEE Communications Magazine*, 55(9):16–24, 2017.

[13] Rachel Potvin and Josh Levenberg. Why google stores billions of lines of code in a single repository. *Communications of the ACM*, 59(7):78–87, June 2016.

[14] Fernando Freire Scattone and Kelly Rosa Braghetto. A microservices architecture for distributed complex event processing in smart cities. In *2018 IEEE 37th International Symposium on Reliable Distributed Systems Workshops (SRDSW)*. IEEE, October 2018.

[15] Saber Talari, Miadreza Shafie-khah, Pierluigi Siano, Vincenzo Loia, Aurelio Tommasetti, and João Catalão. A review of smart cities based on the internet of things concept. *Energies*, 10(4):421, March 2017.

[16] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things Journal*, 1(1):22–32, February 2014.

SaaS business model

Literature study

Group 18: Naim Civan, Karel Geraedts, Mihai Popescu, Teodora Serbanescu

Abstract

Cloud computing is a novel way of delivering and consuming information technology services. Cloud computing is expected to be the most essential technology in the future, given its benefits for businesses and the opportunity for change. Since the birth of the internet, there has been a surge in information technology innovation. The purpose of this study is to bring about a comprehensive and comparative analysis of Cloud and Software-as-a-Service (SaaS) compared to other previous systems. The method will be in a way of exhibiting a top-down approach, beginning from the generic cloud technology and its evolution to the specifications of Software-as-a-Service (SaaS). Hence, the main contribution of this paper will be a better understanding of how Cloud and its services evolve out of previous technologies and the critical assessment of SaaS. We then look at how different businesses look at different types of cloud. We analyse the impact of SaaS risks and benefits concluding that the technologies evolved very rapidly. Finally we discuss pricing and how different capabilities and the architecture impact the price of the service.

1. Introduction

Cloud computing is the result of the evolution and adoption of existing technologies and paradigms such as cluster computing, distributed computing, utility computing ,and grid computing in general. Therefore, the understanding of its development in the way of Cluster-Grid-Cloud plays a critical role. Over the past decade, there has been heightened interest in Cloud because of the rapid decrease in hardware cost, increase in computing power and storage capacity, growing data size in scientific domains as well as widespread adoption of Web 2.0 applications [1]. As worldwide adoption and usage of Cloud services increase, the question arises as to whether adequate overarching analysis exists to satisfy the rising demand. The background of this study is the concerns about SaaS since the responsibility of business models has switched from the user's part to the provider's part. As shown in the Figure 1:

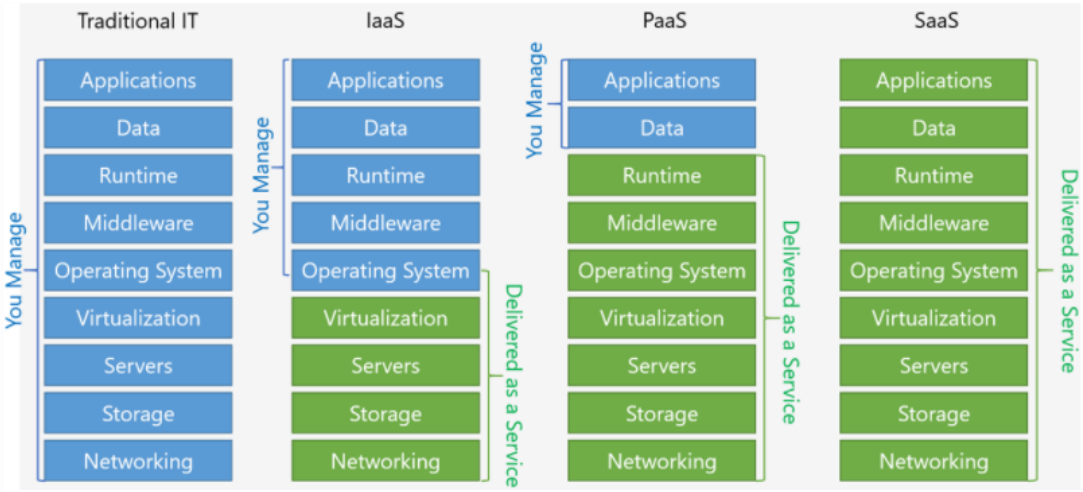


Figure 1: Shared responsibility in the cloud [2]

This paper focuses on Cloud service SaaS but begins with a comparative analysis of computing models' characteristics then drills down to SaaS benefits and drawbacks. If SaaS is the latest trend in this realm then there should be a comparison between business models in terms of their risks and benefits. Since 92 percent of organizations already are at least somewhat in the cloud as described in (Knorr, 2020), the point of discussion shifted to business models within cloud computing rather than traditional-to-cloud transition. Even if previous studies offered a set of assessments, there are still some confusing definitions that are making the comparison blurry. This is an important problem because companies make large investments in Cloud every year in order to leverage the new capabilities that Cloud provides. However, an uncontrollable way of transition or the lack of an auditable path would create strategic disadvantages in the future that is why the positive and negative sides should be assessed with a holistic approach. The rest of the paper is organized as follows. Issues and challenges related to cluster, grid and cloud computing models are explained in section 2.

2. Evolution of Cloud

The increased popularity of the Internet and the high availability of powerful computers and high-speed network technologies has improved the way computers are used. At first, supercomputers were irreplaceable for many years because of the effective computing power, then the way of computing was upgraded to a new level due to the arisen problems in science, engineering and business.

Thereafter, Cluster computing has shown up to avoid the problems faced during the supercomputers period by means of cheaper ways for gaining access to high computing power. As can be understood from the name, Cluster computing is a collection of parallel and or distributed computers which are interconnected among themselves using high-speed networks. For example, Hadoop [4] is an open-source framework for running data-intensive applications in a processing cluster built from commodity server hardware. The main working logic is that the user's requests are received and distributed amongst the computers to form a cluster then share the computational workload as a single virtual system. However, Cluster computing is later replaced with Grid computing for a variety of reasons. The most challenging part in Cluster is that the computers in the cluster should have the same

task, same hardware and same network that is not always possible for the problems happening today. The other differences are demonstrated in the figure below:

Figure 2: Cluster versus Grid [5].

Grid computing dominated the academia in the mid 1990s by enabling users to remotely utilize any idle computing power. Buyya et. al. [6] defined grid as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users quality of-service requirements. For instance, an example of Grid projects would be more meaningful. Globus [7] is an source grid software that addresses most challenging problems in distributed resource sharing.

As for the comparison of Cloud computing and Grid computing, the features of these computing models seems to be similar conceptually since both share the same vision of providing services to the users through sharing resources among a large pool of users and they are capable of multitasking based on a high-speed network technology. Also, Grid and Cloud computing shows up to be a promising model particularly centering on standardizing APIs, security, interoperability, unused commerce models, and dynamic pricing for complex configurations. Hence, this similarity creates confusion about how they differ among themselves. The Cloud is a result of a technology evolution since there is a huge shift from an infrastructure that delivers storage and compute resources to one that aims for more abstract

resources [1]. To gain a better understanding, the provided table summarizes the differences between computing models very well.

	Clusters	*Grids*	*Clouds*
SLA	Limited	Yes	Yes
Allocation	Centralized	Decentralized	Both
Resource Handling	Centralized	Distributed	Both
Loose coupling	No	Both	Yes
Protocols/API	MPI, Parallel Virtual	MPI,MPICH-G, GIS,GRAM	TCP/IP,SOAP, REST,AJAX
Reliability	No	Half	Full
Security	Yes	Half	No
User friendliness	No	half	Yes
Virtualization	Half	Half	Yes
Interoperability	Yes	Yes	Half
Standardized	Yes	Yes	No
Business Model	No	No	Yes
Task Size	Single large	Single large	Small & medium
SOA	No	Yes	Yes
Multitenancy	No	Yes	Yes
System Performance	Improves	Improves	Improves
Self service	No	Yes	Yes
Computation service	Computing	Max. Computing	On demand
Heterogeneity	No	Yes	Yes
Scalable	No	Half	Yes
Inexpensive	No	No	Yes
Data Locality Exploited	No	No	Yes
Application	HPC,HTC	HPC, HTC, Batch	SME interactive apps.
Switching cost	Low	Low	High
Value Added Services	No	Half	Yes

Table1: Comparison Of Cluster, Grid And Cloud Computing [8].

This modern paradigm has attempted to be defined many times through the last decade and it also caused a bunch of confusing and incomplete definitions. This definition challenge is perfectly handled by Foster et. al. [1] as “A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet”. This cloud model is composed of three service models, and four deployment models.

3. Characteristics of Cloud

The Cloud was developed to solve Internet-scale computing problems that is usually referred to as a large collection of storage resources, which can be accessed by means of the standard protocol through abstract interfaces. The cloud framework can be seen as containing both a physical layer and an abstract layer. The physical layer comprises the equipment assets that are fundamental to support the cloud administrations including server, capacity and network components. The abstract layer consists of the software deployed across the physical layer and sits above the physical layer [9].

There are four deployment models in Cloud services.

Private cloud [9] is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

Community cloud [9] is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

Public cloud [9] is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

Hybrid cloud [9] is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

Clouds in general provide services at three different levels (IaaS, PaaS, and Saas [10]) as follows:

Infrastructure as a Service (IaaS) [10] provisions hardware, software, and equipment (mostly at the unified resource layer, but can also include part of the fabric layer) to deliver software application environments with a resource usage-based pricing model. *Platform as a Service (PaaS)* [10] offers a high-level integrated environment to build, test, and deploy custom applications. *Software as a Service (SaaS)* [10] delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model. As mentioned in the introduction phase, this paper will shed light on SaaS specifications compared to other business models.

4. Cloud Computing Services

In recent years many companies are looking for cloud deployment solutions to increase performance, contingency, security, availability and more. As these companies look to the Cloud it is important that they know and understand the Cloud Computing marketplace. Broadly speaking this marketplace can be

separated into three different service levels [24]. These levels are infrastructure as a service (IaaS), platform as a service (PaaS) and software-sometimes also called application- as a service (SaaS). See Figure 2 for a graphical representation of this framework. In the 3 subsequent subsections we will explain in more detail what each of these different services are, how they are being used and their respective pros and cons.

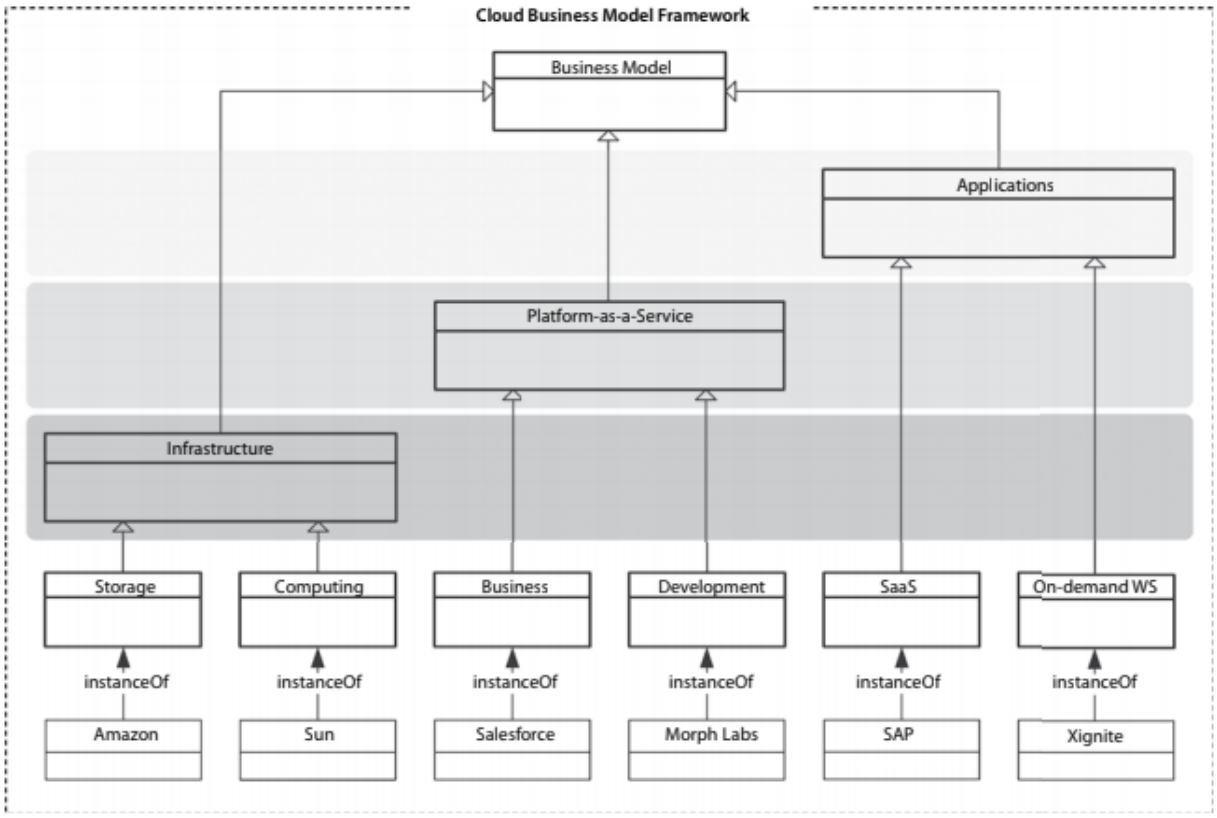


Figure 2: Cloud business model framework [24]. The Application layer is often called SaaS.

4.1. Infrastructure as a service (IaaS)

IaaS is the lowest level service. Providers of IaaS make it easy for a buyer of the service to access storage or compute power, I.e. They offer access to hardware. In this way it looks much like Grid Computing, but as mentioned above there are some differences. Mainly, the pricing is different with Cloud using pay-per-use instead of subscription based as Grids do. Examples of IaaS are Amazon's S3 Buckets and Amazon's EC2 (Elastic Cloud Compute). The first is a simple storage service with a user interface that can be accessed over the Web. This is often used to store large amounts of data for backup and analytics. The latter is a service through which a user can initiate a virtual computing environment called an 'instance'. Through these instances one can use Amazon's hardware: CPU, storage, memory and more.

The benefit of IaaS is that it enables cheap access to hardware without having to manage complex IT infrastructures. However, this is still quite a low-level service and for some companies, especially start-ups or non-technical companies, it might not be the right solution. Using EC2 instances requires the

user to manage much of the product levels themselves, which requires certain know-how within the company that can be expensive to get. For these companies a PaaS or IaaS service might be better.

4.2. Platform as a service (PaaS)

PaaS is a service that is built on top of IaaS (see Figure 2) and as [27] says, “It provides the central platform and marketplace where all other actors come together, trade their services, and interact with each other.” The platform service plays a key role in bringing software developers and consumers together. The consumer uses the platform to browse different software and the developer uses the platform to offer their service. The platform in exchange charges by asking the developer a small fee for being able to promote their service on the platform, by asking the customer a fee for being able to access the platform, or by charging the developer a fee for all newly sold.

Though there are many benefits to PaaS in terms of releasing responsibility to the infrastructure provider this also has its drawbacks. This consequently also makes the consumer reliant on the provider's infrastructure and changes/updates or breakdowns can endanger projects.

4.3. Software as a service (SaaS)

Vendors using SaaS offer software or applications to the users, like for example Dropbox. These services are built on top of a cloud infrastructure and accessed over the Web. A provider will typically use an available cloud infrastructure for the hardware, build software on top of it and sell the software on a Cloud platform. In this way they offer value to the customer by selling/renting software opposed to hardware. Their pricing models are generally subscription based and pay-per-use. A principal benefit compared to the traditional selling of software is that the user doesn’t need to concern themselves with scaling.

Especially start-ups and medium enterprises will benefit from using SaaS since the creation, deployment and scalability of software is made much easier compared to on-premises software. However, among the disadvantages of SaaS are the lack of control, security and data concerns, and performance.

5. Defining the Cloud Computing Business Model

Before discussing the suggested categorization of Cloud Computing as proposed by [25]. In [26] the authors argue that scholars do not yet agree on what a business model is and that the literature has developed largely in silos. Though, they find there is also consistency in definitions which they attribute within 4 different sections. We will use one of these sections to clarify what we mean by business model. ‘Business models seek to explain how value is created, not just how it is captured.’ For Cloud this means that not only pricing models are of interest but also privacy, security, scalability, development and more are part of the Cloud Computing Business Model. Now that this is clear, in the next section the categorization proposed in [28] is introduced.

5.1. A categorization of Cloud Computing Business Models

Companies can choose from a wide range of business models in Cloud Computing. Knowing which model to adopt can be a confusing task for a company. To help with this [25] come up with a Cloud Cube Model (CMM) from which 8 models are identified. 1) Service Provider, 2) Support and Services, 3) In-House Private Clouds, 4) All-In-One Enterprise Cloud, 5) One-Stop Resources and Services, 6) Government Funding, 7) Venture Capitals and 8) Entertainment and Social Networking. Each of these 8 types cover parts of the Cube Cloud Model.

The Cloud Cube model is a 4-dimensional model, see Figure (cube). These are Internal/External, Proprietary/Open, Perimeterised/De-perimiterised and Insourced/Outsourced. The 4th dimension is given by the colouring of the cubes. Internal/External refers to private or public clouds. Proprietary/Open refers to paid services versus open source (free) services. Perimeterised/De-perimiterised refers to IaaS/PaaS and SaaS. Insourced/Outsourced means the in-house development of clouds versus using other Cloud provider's services.

The aim of this categorization is to help companies to pick the most suitable option. However, we think it the categorization is not specific and detailed enough. It can give a company rough guidelines for which Cloud BM to choose but more practical cases are still so different that a more detailed study would be needed.

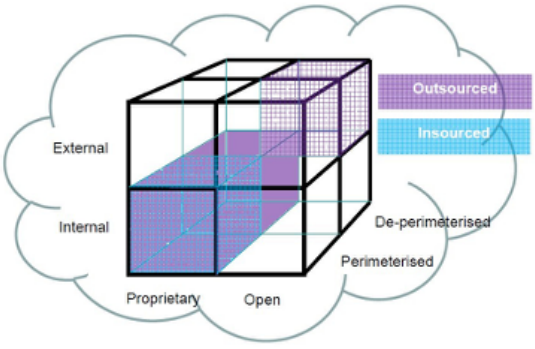


Figure 3: CCM for Support and Service Contracts

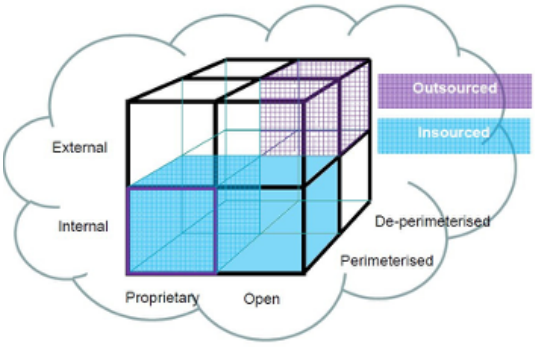


Figure 4: CCM for In-House Private Clouds

6. SaaS Analysis

6.1. Benefits

As previously said, SaaS software differs from traditional software because it is already installed and configured by the provider and there is no need to go through any installing issues. Since the environment where the software runs is usually shared the costs should be lower based on the hardware

costs alone, but also the maintenance cost is lower because the provider has the same environment under control. There are a lot of other advantages for SaaS, which include scalability, lower costs of integration. Compatibility is another important aspect, both for simple users which would want to have access at anytime on web/mobile and also for businesses which would not want to include another stack of applications in the workflow which would have to be configured and installed.

Various studies analyse the opportunities and risks of SaaS, showing that cost advantages are the strongest benefit, but flexibility and improved quality also have a high rating in the surveys[17][18].

6.2. Risks

An article[12] focused on identifying benefits and risks associated with utilizing cloud computing identified the following risks and split them into tangible and intangible risks which can be seen in the figure below. Of course, the risks depend on the users and use cases, for example governments could focus on service agreements, besides the technologies provided[14]. Confidentiality, integrity and data protection would mainly be an important factor to users such as banks.

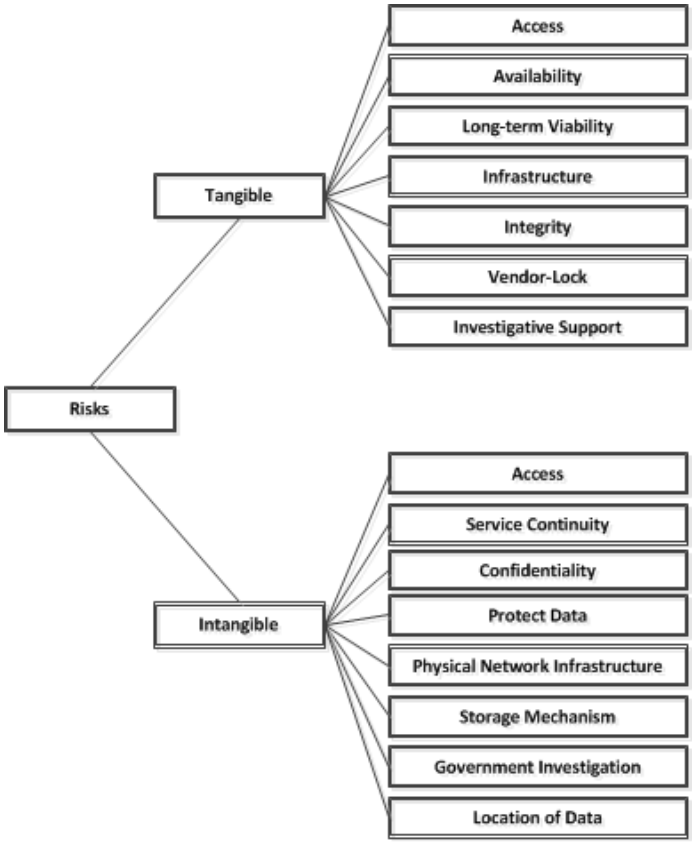


Fig.4 SaaS risks[12]

Security is sometimes referred to as a top selection criteria for SaaS[18], while other times it is viewed as one of the more concerning aspects of the model[17]. While some risks can not be accepted by everybody, for most of the medium sized companies studies show that SaaS security is better than on

premises applications. This is because big cloud providers have strong solutions and expertise for intrusion detection and other malicious activities and would prevent security related issues from happening. Policies like file encryption are usually enforced by providers. Also, there are solutions known as CASB (cloud access security brokers)[15] to improve the security of the service. These solutions could include

- Cloud governance and risk assessment
- Data loss prevention
- Control over native features of cloud services, like collaboration and sharing
- Threat prevention, often user and entity behavior analytics (UEBA)
- Configuration auditing
- Malware detection
- Data encryption and key management
- SSO and IAM integration
- Contextual access control

These solutions are not only for SaaS, but could also be deployed for PaaS or IaaS.

Audits of the networks for unauthorized access of the services or compromised accounts are important, but using external auditors happen only for medium size service providers because large companies usually do internal audits and reviews.

6.3. SaaS pricing

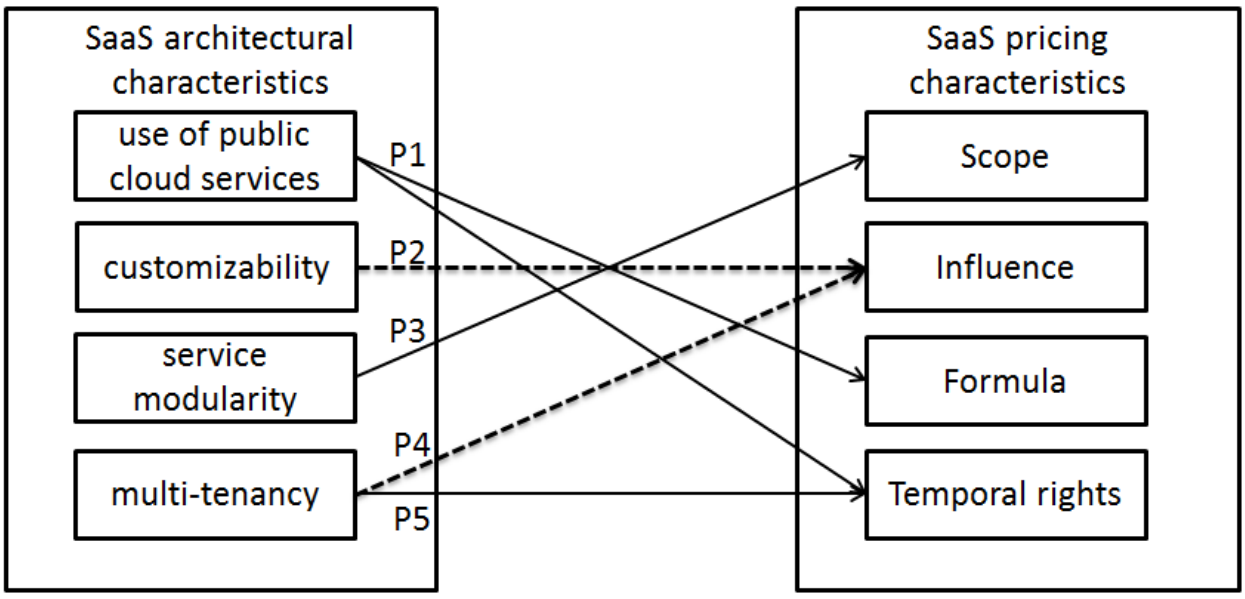


Fig.5 The impact of architecture on pricing models[11]

SaaS pricing is usually an important aspect to be considered, but also other characteristics. An interesting research question is “What is the impact of pricing on the SaaS architecture?” A multicase study[11]

analyses this specific question and presentes the relationship between architectural and pricing aspects which can be seen in the figure.

The classification of the SaaS pricing characteristics is described in a cloud services pricing model research[13] which describes the SBIFT (figure below) model for cloud pricing. Scope represents the granularity of the offering, base is the information the price is set on, formula is the connection between the price and the volume and temporal rights the kind of the service (eg: Pay-per-use or Subscription based, etc).

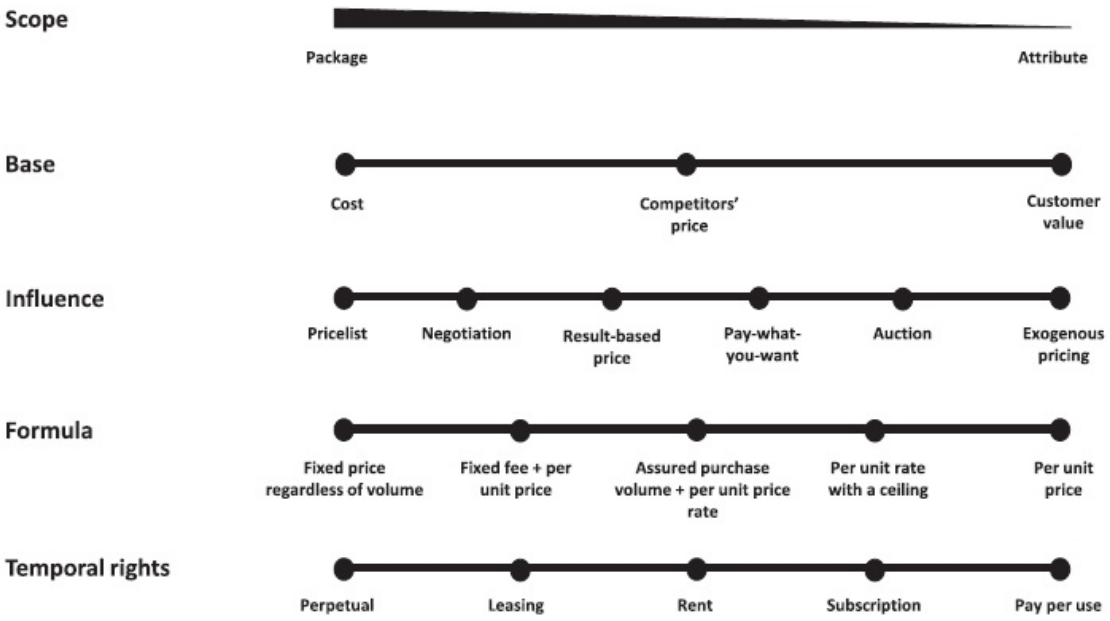


Fig.6 SBIFT model [13]

6.4. Discussion: Business Model in SaaS Firms

As previously mentioned, SaaS adoption criteria varies based on the application type[19]. For example, a survey conducted by Benlain et al.[19] suggests that complex, strategically important applications are less likely to adopt a SaaS approach, compared to simpler, less specific applications.

As it would be expected, it has been observed that there is a difference in preference between open source software and traditional software products preference[20].

It is clear that different customers have different needs and SaaS vendors may adapt to fulfill those requirements with different offerings. The book “Economics of Grids, Clouds, Systems, and Services”[20] opens a discussion on this topic and analyses the different models. Their conclusion and suggestion would be to replace the different discrete types with a continuous personalization of SaaS tailored to the needs of each environment.

Now there is the open ended question of which environment is more well suited to succeed with SaaS: start ups or larger companies? While it may be easier for a long-established software vendor to be able to provide reliable, high-quality services, a smaller company may be able to offer these services at a lower price. The best example of such a successful story is Salesforce.com.

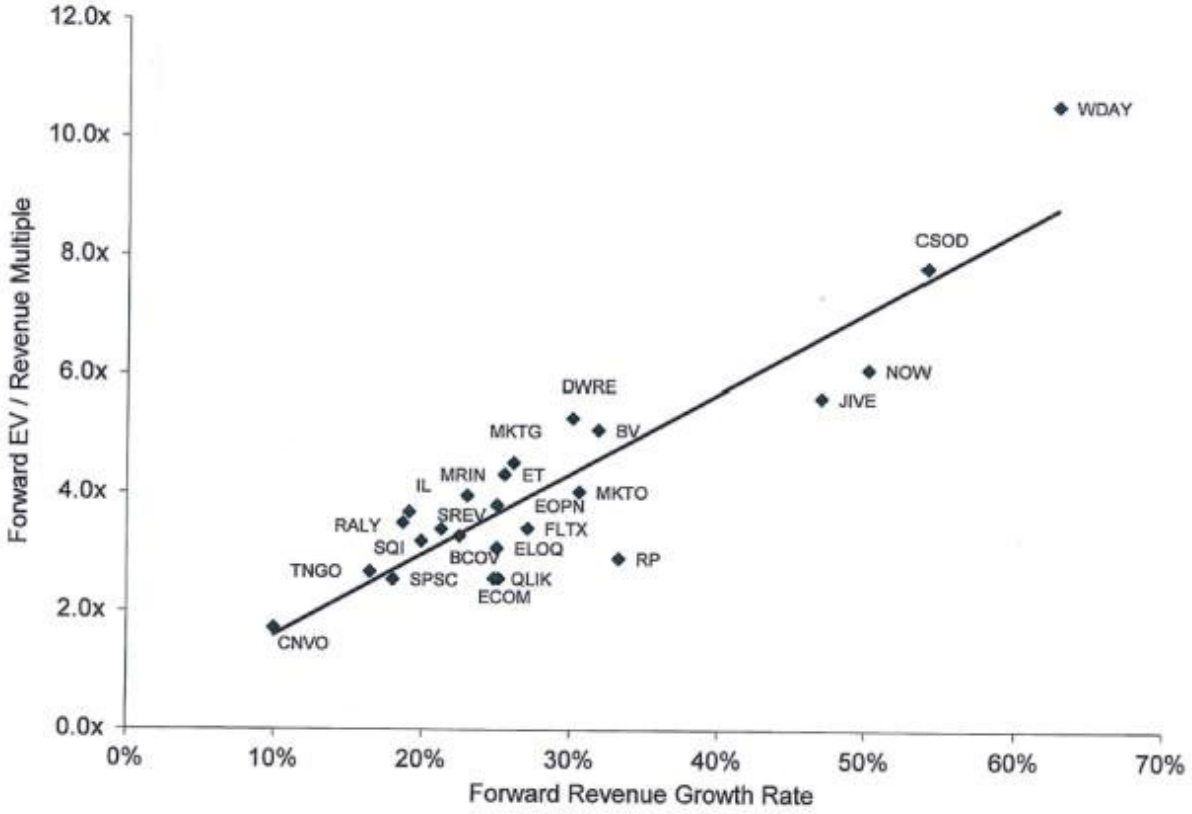


Fig.7 Cross-plot of SaaS company EV multiple relative to revenue growth[21]

But there is room for both small and large enterprises in this field. Now more than ever, given that everyone has to work remotely due to the COVID-19 pandemic, there is a high demand for popular service models like SaaS and PaaS. This comes with its risks, of course, considering the data privacy and security issues that come with this setup: personal devices, application issues[22]. But when these matters are treated with care, startups can benefit from the advantages of SaaS and use them for elegantly solving relevant issues. One good example is Docusign, which makes electronic signatures possible and safe[23].

Conclusion

Current SaaS market only accounts for approximately 20% of total enterprise software spending[16] so there is still a lot of room to grow. Major spending goes to a few well known providers which dominate the market and also offer other types of services such as IaaS and PaaS. Although, considering the

benefits and the risks, we can safely say that the SaaS market is an important aspect of the current technologies.

References

[1] I. Foster, Y. Zhao, I. Raicu and S. Lu, " 2008 Grid Computing Environments Workshop, 2008, pp. 1-10, doi: 10.1109/GCE.2008.4738445.

[2] Sharma, P. (2020, April 22). Paradigms for Hosting Applications – IaaS, PaaS and SaaS. Tech-Quantum. https://www.tech-quantum.com/paradigms-for-hosting-applications-iaas-paas-and-saas/

[3] Knorr, E. (2020, June 8). The 2020 IDG Cloud Computing Survey. InfoWorld. https://www.infoworld.com/article/3561269/the-2020-idg-cloud-computing-survey.html

[4] “Hadoop”, http://hadoop.apache.org.

[5] Lithmee. (2018, September 8). Differences Between Cluster and Grid Computing. https://pediaa.com/difference-between-cluster-and-grid-computing/

[6] M. Chetty and R. Buyya, “Weaving Computational Grids: How Analogous Are They with Electrical Grids?”, *Computing in Science and Engineering (CiSE),*4, pp. 61-71, 2002.

[7] Ian Foster and C. Kesselman, “Globus: A Metacomputing Infrastructure Toolkit”, *Intl. Jr. of Supercomputer Applications,* 11(2), 1997.

[8] N. Sadashiv and S. M. D. Kumar, "Cluster, grid and cloud computing: A detailed comparison," 2011 6th International Conference on Computer Science & Education (ICCSE), 2011, pp. 477-482, doi: 10.1109/ICCSE.2011.6028683.

[9] P. Mell and T. Grance, “The NIST Definition of Cloud Computing ”, *Special Publication 800-145,* September, 2011.

[10] “What is Cloud Computing?”, Whatis.com. http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci1287881,00.html, 2008.

[11] Laatikainen, G., & Ojala, A. (2014). SaaS architecture and pricing models. In E. Ferrari, R. Kaliappa, & P. Hung (Eds.), Proceedings of the 2014 IEEE international conference on services computing (SCC 2014) (pp. 597-604). IEEE. doi:10.1109/SCC.2014.84

[12] “Identifying Benefits and Risks Associated with Utilizing Cloud Computing” The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE’13], San Francisco State University, CA, U.S.A., March 2013. Doi: 10.7321/jscse.v3.n3.63

[13] G. Laatikainen, A. Ojala, and O. Mazhelis, “Cloud Services Pricing Models,” in Software Business. From Physical Products to Software Services and Solutions, Springer, 2013, pp. 117–129.

[14] Paquette, Scott & Jaeger, Paul & Wilson, Susan. (2010). Identifying the security risks associated with governmental use of cloud computing. Government Information Quarterly - GOVT INFORM QUART. 27. 245-253. 10.1016/j.giq.2010.01.002.

[15] SaaS security solutions, https://www.mcafee.com/enterprise/en-us/security-awareness/cloud/what-is-a-casb.html

[16] "Cloud Market Share – a Look at the Cloud Ecosystem in 2021" https://kinsta.com/blog/cloud-market-share/

[17] Benlian, A., Hess, T.: Opportunities and risks of software-as-a-service: Findings from a survey of it executives. Decision Support Systems 52, 232–246 (2011)

[18] Repschlaeger, J., Wind, S., Zarnekow, R., Turowski, K.: Selection Criteria for Software as a Service: An Explorative Analysis of Provider Requirements. In: AMCIS

[19] Benlian, A., Hess, T., Buxmann, P.: Drivers of SaaS-Adoption An Empirical Study of Different Application Types. Business & Information Systems Engineering 1, 357–369 (2009)

[20] Eetu Loma: "Economics of Grids, Clouds, Systems, and Services: 1. Examining Business Models of Software-as-a-Service Firms"

[21] Cohen, Benjamin; Neubert, Michael, "CORPORATE VALUATION OF SAAS COMPANIES: A CASE STUDY OF SALESFORCE.COM"

[22] Ziyad R. Alashhaba, Mohammed Anbara Manmeet, Mahinderjit Singhb, Yu-Beng Leauc, Zaher Ali Al-Saib, Sami Abu Alhayja'aa: "Impact of coronavirus pandemic crisis on technologies and cloud computing applications"

[23] Bojana Lobe, David L Morgan: "Qualitative Data Collection in an Era of Social Distancing"

[24] Weinhardt**:** "Cloud Computing – A Classification, Business Models, and Research Directions"

[25] Victor Chang: **"**A Categorization of Cloud Computing Business Models"

[26] Christoph Zott: **"**The Business Model: Recent Developments and Future Research"

[27] Frank Keuper: **"**Application Management" (https://link.springer.com/content/pdf/10.1007/978-3-8349-6492-2.pdf#page=40)

[28] Jing-Jang Hwang: "A Business Model for Cloud Computing Based on a Separate Encryption and Decryption Service"

UNIVERSITEIT VAN AMSTERDAM

VU

Pricing Models in Clouds and for Cloud-based Applications - Literature Review

Ilias Daia
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands

Lennart Kerkvliet
Department of Computer Science
Universiteit van Amsterdam
The Netherlands

Lloyd Nyarko
Department of Computer Science
Vrije Universiteit Amsterdam
The Netherlands

Mick Vermeulen
Department of Computer Science
Universiteit van Amsterdam
The Netherlands

Abstract

Cloud computing is a concept for providing on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that may be swiftly supplied and released with minimal administration effort or service provider contact. As a result, the ability to respond to shifting demand is improved. It also reduces the cost of infrastructure and the cost of coordinating IT resources. Investment and operations costs are lowered due to the Cloud's economies of scale. In this paper the three payment pricing models are discussed. Based on the research we conducted, we believe pricing methods can be classified into three categories. We will look at fixed price first, then dynamic pricing, and finally value-based pricing. We examine and explore these models in depth in this study.

Keywords: Cloud Computing, Saas, IaaS, PaaS, QoS, Payment Pricing Models

1 Introduction

Cloud computing technology has emerged as a potential concept for utilizing on-demand computational and software resources. It delegated to service providers the complicated and time-consuming resource and software management responsibilities that are traditionally handled by customers. As a result, it improves the ability to respond to changing demand. It also lowers infrastructure costs and the expense of coordinating IT resources. Because of the Cloud's economies of scale, investment and

operational costs are reduced. As a result, Cloud based systems have caused a lot of buzz in the IT business. Another important point is that it attracts not only to corporate clients, but also to small research organizations in the field of computational science and engineering.

Cloud-based systems are the most popular way of hosting an application, with many different types of services. There are many different kinds of services which are also built on each other. In this paper we are going to differentiate between three different types of services; IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service).

With IaaS, a third party will do the provisioning of hardware (servers, storage, and networks) as well as supporting software (operating systems virtualization technologies, file system) as a service. It's a step forward from traditional hosting in that it doesn't need a long-term commitment and allows customers to provide resources on demand. Unlike PaaS services, the IaaS provider undertakes very little administration other than keeping the data center running, and customers must deploy and operate software services as they would in their own data center. IaaS products include Amazon Web Services' Elastic Compute Cloud (EC2) and Secure Storage Service (S3). Previously, if you wanted to run a service, you had to purchase a server and physically set it up somewhere. Being able to employ this might save a lot of money, especially for fast-paced businesses [1].

PaaS goes a little further than IaaS. Platform As-a-Service (PaaS) is a web-based application development and deployment platform. It enables application development and deployment without the cost and complexity of purchasing and managing the underlying infrastructure, by providing all of the necessary facilities to support the entire life cycle of developing and delivering web applications and services that are entirely accessible via the Internet. The whole platform is available for the developer, which includes the operating system and parts of the software. This lets the developer focus on developing *only* the application. A database, middleware, and development tools are often included on this platform. This infrastructure software is frequently based on a virtualized and clustered grid computing architecture. Some PaaS services require you to use a specific programming language or API. Google AppEngine, for example, is a PaaS platform that allows developers to write in Python or Java. EngineYard is a Ruby on Rails application. Some PaaS providers, such as Salesforce.com's force.com and SAP's Coghead, have proprietary languages [2].

SaaS goes even further by providing the whole application. SaaS is a very broad term, it can be described as a software distribution paradigm in which a vendor or service provider hosts applications and makes them available to consumers across a network, usually being the Internet. As underlying technologies that support web services and SOA (service-oriented architecture) evolve and new development methodologies become popular. A pay-as-you-go subscription licensing model is frequently connected with SaaS. In addition, SaaS apps must be able to connect with other data and applications in a wide range of contexts and platforms. SaaS is intertwined with the other service delivery methods we've discussed. SaaS is most commonly used to give business software capabilities to enterprise clients at a cheap cost, allowing them to gain the same benefits as commercially licensed, internally run software without the complexities of installation, maintenance, licensing, high initial cost etc. Microsoft Office is one of the more famous SaaS products, but the SaaS model is used in many businesses and it can also be used for API's. In essence it is (online) software that is provided for a fee.

These three things often build on each other. SaaS is the appellation layer of *Cloud Computing* and underneath that is the platform (PaaS) and infrastructure (IaaS) layer. In all layers the customer has the advantage of not having to purchase the hardware and software, they just have to pay for using

it. The customer also doesn't have to accommodate for application management, like preventing unauthorized access.

Traditionally customers had to pay a fixed fee. They could select a plan and if that plan didn't provide their needs they could upgrade their plan to a higher tier. Most cloud services now provide a way of *dynamic billing*, where you only pay for what resources you have used [3]. Agreements on matters such as pricing are included in a service level agreement (SLA), which is is an essential component of cloud computing. It details the negotiation between provider and consumer. Some other points included in the SLA are the provision of services, the specific level of Quality of Service (QoS), and guarantees. The final agreements are documented in a contract with the concerned parties [4].

This paper is divided into four sections. Some paper summaries are discussed in the first half. Following that, modern price models for fixed pricing, dynamic pricing, and customer-value-based pricing are listed. The future of Cloud Pricing Models is discussed in the next section. Finally, the review's discussion and conclusion are presented.

2 Paper summaries

2.1 An Overview of Pricing Models for Using Cloud Services with analysis on Pay-Per-Use Model

This paper [5] outlines fixed, static and market dependent pricing models for cloud computing and shines light on some of the methods involved to achieve either of these. According to the author, fixed pricing is the most commonly used and easy to implement pricing model. With fixed pricing you pay a fixed fee for a predetermined unit, this unit could be per month, per computing node, per CPU usage, etc. Advantages of this model are mainly its ease of implementation, transparency and stability: no complex software is required to make these systems and they are easy to maintain. When using pay-per-use pricing this model can also prove highly cost-effective for the user. Disadvantages could be that the price might not reflect supply and demand in the market, the service provider or user may be overpaying with regards to market standards. It also makes it difficult for providers to change the price, limiting potential profits.

Dynamic models change based on supply and demand, if many resources are required at a certain moment in time, the price per unit can rise. The main advantages of this model is that the service provider has more control over the price and therefore can potentially generate more profits. For users this introduces price risk, though, the user has to take into account when they reserve resources to prevent unintentional overpaying. Dynamic systems are also more difficult to implement as they need more features: customers generally want to be able to set upper and lower bounds to reduce their price risk and algorithms are required to determine the current market price of the requested resources.

Market dependent pricing models are rarely used and rather difficult to implement. In this model the price is not only based on supply and demand like dynamic pricing but there is also a bid-ask marketplace introduced, here multiple bidders can buy resources based on real-time market conditions. This model can be lucrative but is very difficult to implement as it requires a functional marketplace and is therefore rarely used.

The paper also briefly highlights pricing indicators that can be used to determine which pricing models are most effective, these indicators are:

1. Initial investments
2. Lease period of the user
3. Quality of service (higher quality leads to higher prices)
4. Rate of deprecation (older hardware is cheaper to maintain and repair)
5. Cost of maintenance

Finally, the paper provides a summary of different pricing models that exist within the fixed and dynamic pricing space:

- Fixed pricing
  - Pay as you go: makes users pay for some predefined resource like a server, this model is not cost-effective for the user but easy to implement
  - Subscription: user pay a fixed fee each month. This pricing model easy to understand and transparent but also not cost-effective for the user as they may pay more than they need.
  - Pay for resource: users pay per the amount of resources they use (e.g. CPU hours or GBs RAM), this method can be very cost effective for the user
  - Hybrid: this is dynamic pricing in the sense that the price can change but this usually works with a job-queue where users specify a max price for some task to run when the price is right based on user limits. The limit is thus still fixed while the pricing is dynamic. Profits are minimal for the provider but the same can be said about the costs for the users
- Dynamic pricing
  - Resource pricing: a variant on the static resource pricing where market conditions determine resource prices. Can be more profitable for the provider but also introduces price risk for the user
  - Genetic pricing: Uses a genetic pricing model that can lead to high profits for the provider but works less well in high-demand and irrational market conditions
  - Value based / novel financial: pricing is provided by the added value the customers receive. Very fair to customers but also rather difficult to implement as value is not an objective measurement
  - Competition: works by supply and demand but between competitors, prices are always determined by competing prices. Very fair to customers but can induce lower profits

In conclusion the authors state that none of these models are perfect and that there should always be a healthy consideration of costs into any of them to make it possible for the provider to survive. On the other hand, revenue heavy models might seem like a good solution from the providers' point of view at first but might hurt the longevity of user-relations as they might not be as cost-effective for users over time.

2.2 Cloud Computing Pricing Models: A Survey

This paper [4] builds on many of the principles of the previous paper but adds descriptions of some more exotic pricing models. Next to agreeing on the same cost structure as the previous paper, this paper quantifies the quality of service costs into the following sub types: availability, privacy, security, scalability and integrity of the service provider. All of these should be considered when trying to price the cost of QoS correctly. They also noted the difference between fixed and dynamic pricing: generally fixed pricing is easier to understand and implement than dynamic pricing but can be more unfair to the customer as a user might be paying for resources they do not necessarily need, also fixed pricing is generally less profitable for the provider than dynamic pricing. Next to the pricing models discussed in the previous paper, this paper describes:

- Fixed pricing
  - The novel financial model: using financial option theory in combination with Moore's law this pricing model is capable of providing a fixed lower and upper limit to pricing which maximizes QoS, this pricing model is therefore beneficial to providers as the price can be varied and beneficial to users as a maximum QoS is achieved
  - Data center optimization: using this pricing model, requested jobs are scheduled using an algorithm based on whether or not they may be interrupted. Using this pricing model electricity costs can be minimized by efficiently deciding when interruptable jobs should run
- Dynamic pricing:
  - Genetic pricing: This easily implementable and highly flexible genome model can be used to increase profits 10-fold under the right conditions as this genetic model is able to efficiently predict the most profitable prices under market conditions
  - Federated cloud: in a federated cloud users and providers are one, both being able to buy and sell resources. This pricing model can lead to very efficient market prices but can be tough to implement due to hardware limitations
  - Advanced reservations: using an advanced reservation model the provider of resources can dynamically price into the future, allowing users to allocate resources ahead of time for the dynamic price that seems fair to them. This can lead to higher profits as customers might be more likely to schedule resources, lowering overall prices but increasing profits during peak hours

The authors conclude that although many pricing models exist and many of them prove to be profitable in simulations, only few of them are actually deployed in production environments. They also state that many of these models focus solely on revenue generation of the provider, these models do not take into account the QoS for the user and therefore it is not taken into account how long a user will stay a customer when using said model. Also, users that always need compute resources, not just when jobs need to be executed, generally do not benefit from dynamic pricing as it introduces price risk, for such users a static pricing model would always be a more cost effective solution.

2.3 Cloud Pricing Models: Taxonomy, Survey, and Interdisciplinary Challenges

This paper [6] approaches cloud pricing from an interdisciplinary angle, thereby drawing from four knowledge domains: microeconomics, value theory, operation research —which is a method that enables cloud decision-makers to make better decision for cloud prices— and cloud technologies. The authors observe that in earlier related works the emphasis in many cases was placed on either cost-based or market-based pricing. Therefore their study incorporates an increased focus on value-based pricing. The paper contains a historical review of cloud pricing, introduces a cloud pricing taxonomy, and provides a detailed survey. Due to page count constraints, we will focus primarily on the taxonomy in this summary.

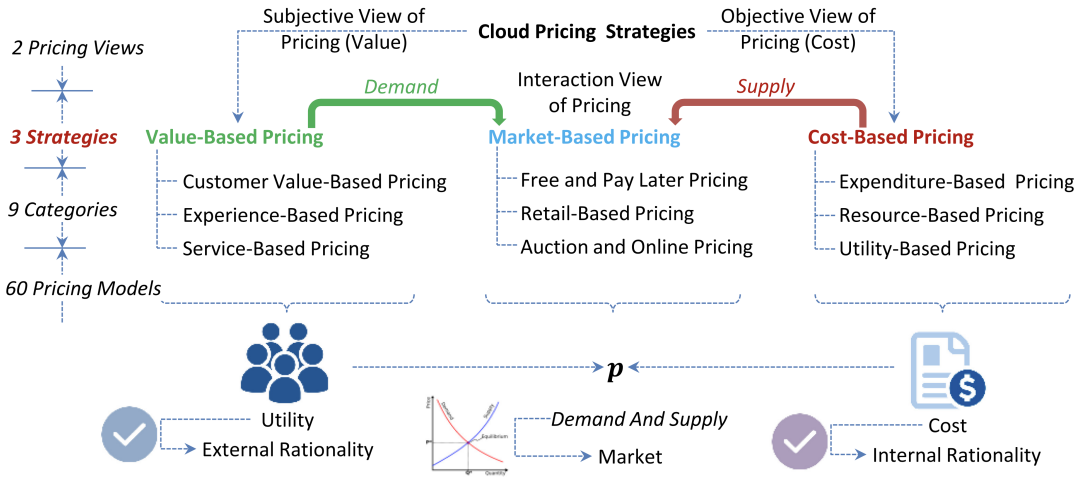


Figure 1: Hierarchical framework by Wu et al. [6]

The authors introduce a hierarchical pricing framework. The top level consists of two pricing views: a subjective view (value) and an objective view (cost). These diverging views influence the three pricing strategies that form the level directly below. Value-based pricing is in line with the subjective view and is demand driven. Cost-based pricing is in line with the objective view and is supply driven. The interaction (or balance) between the two views is market-based pricing, which is the third pricing strategy. The level beneath the pricing strategies captures nine categories, also referred to as tactics. The categories that are placed in the value-based bracket are: customer value, experience, and service-based pricing. Under the market-based pricing umbrella are: free upfront and pay later, retail-based, and auction and online pricing. Lastly, the categories rooted in cost-based objectivity are: expenditure, resource, and utility-based pricing. Based on these categories, the authors identify 60 pricing models, which are placed in a level further below in the hierarchy.

In their taxonomy of pricing models, the authors provide more details of the nine categories they introduce in their framework.

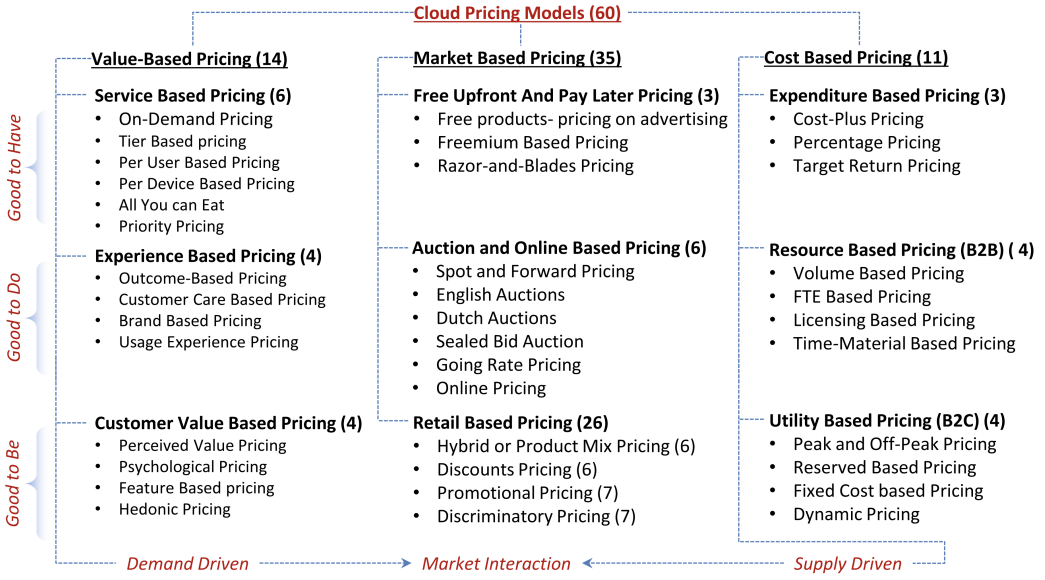


Figure 2: Taxonomy of cloud pricing models by Wu et al. [6]

The first category, *service-based pricing*, is described as an intangible part of value delivery. Examples of service-based can be found when looking at the banking, airline, insurance and other similar industries. Many CSPs of SaaS utilize service-based pricing models, for example Azure and Salesforce.com. The metric of the pricing value is indicated by tiers, levels, devices, users, and priority. The advantage of service-based pricing models is that their values are identifiable and predictable.

In this category the following pricing models are identified by the authors. These models are on-demand, tier-based, per-user-based, per device-based, all you can eat, and priority-based pricing.

The second category, *performance-based pricing*, is based on a set of concrete metrics, often set by service performance metrics, such as cloud service reliability or utilization rate of a limited resource. The example that is provided is that of online advertisement payment, since this is dependent on measurable data, such as the number of clicks or purchases. The basic idea behind pricing models in this category, is that CSP's services meet the customer's business objectives or values. Thus there is an alignment in values for both parties. The advantage here is that there is a possibility of a win-win pricing model that would be equitable. In a cloud setting, this can be observed through the many B2B cloud services that prioritize performance-based pricing, as CSPs offer guarantees of performance such as a specified Gigabit/s network throughput. The CSP would then carry the uncertainty risk. On the flipside, there are complications and limitations in quantifying performance metrics.In this category the following pricing models are identified by the authors. These models are outcome-based, customer care-based, brand-based, and usage of experience-based pricing.

In the third category, *customer value-based pricing*, price is derived a from a subjective view of a customer, with an emphasis on the customer's value delivery. The main advantage is that CSPs would be able to maximize their business profit. However the challenge lies in how to quantitatively define

the customer's subjective value. In this category the following pricing models are identified by the authors, namely perceived value, psychological, feature, and hedonic-based pricing.

Free upfront and pay later. Market competition has led to the adoption of free upfront and pay later pricing models by many CSPs. Pricing models in this category leverage free products with limited features in order to capture more customers and profit from premium customers.In this category the following pricing models are identified by the authors. These models are free products-pricing on advertising, freemium, and razor-and-blades pricing.

The authors go on to describe the *auction and online-based pricing* category. In auction-based pricing the price is determined by the auction mechanism. Examples in practice are AWS EC2 and S3 pricing, which are also subject to bidding rules. The advantages of auction-based pricing are its speed, the absence of backward or forward processing steps, transparent pricing, fairness, straightforward and direct process. The limitations lie in the relatively little time the bidder has to make a decision, and the possibility that the bidding price exceeds the actual value of the goods. Online pricing refers to the purchasing goods that strictly require processing via the internet. Although there are instances where online retailers offer the same good offline, they could do so at varying offline and online prices. Advantages to online-based pricing are the ability to instantly reach a large amount of customers, the lack of extra handling costs, the convenience on the customers' end. The disadvantages are high security risks, privacy concerns, lack of significant discount, and fraud. In the cloud context, a CSP can leverage online information to customize cloud services for a personalize price. In this category the following pricing models are identified by the authors. These models are spot and forward pricing, English auctions, dutch auctions sealed bid auctions, going rate pricing, online pricing.

The *retail-based pricing* category can be subdivided into four subcategories: product mixing pricing, discounts and allowances pricing, promotional pricing, discriminatory pricing.

Important for the *product mix pricing subcategory* are the relationships between various products and establishing the right mix that achieves maximum profit. The benefits of pricing models in this subcategory lie in the ability to boost sales, generate a surplus of revenue, and meet diverse demands or market segments. Conversely, the disadvantages of these model have to do with customers that may feel frustrated with trapping into a cost black hole. Others may refrain from purchasing at all. There is also the concern that the CSP might get a bad reputation due to backlash from premium customers. Hence, rational decision making that considers customer's demands from different market segments is required. In this subcategory the following pricing models are identified by the authors, namely product line, optional feature, captive product, two-part tariff, by-product, and product bundling.

Discounts and allowances pricing models can be adopted by firms to react to changing market conditions. Firms tend to give discounts (i.e., reduce pricing) for the following reasons: product promotion, off-season, cash payment, bulk purchase, display, bundle, wholesale, and two-part tariff. Whereas allowance, which is a type of discount, is directed at wholesale customers, commercial clients or SME. The advantages of this pricing subcategory are the reduction of stock inventory or to improve the capacity utilization rate, particularly for perishable assets, like cloud resources. The disadvantages of models in this category are a decline in profit margin and lack of brand identity. At the time, the three leading CSPs offered a price discount, such as spot, preemptible, and low priority. In this subcategory the following pricing models are identified by the authors. These models are early payment, off-season, bulk purchase, retail discount, cash discount, and trade-in allowance.

The next subcategory is *promotional pricing*; within a predetermined timeframe a discount is given. Similar to discounts pricing models, the advantage is an increase in sales and a decrease of stock inventory. Additionally promotions can aide in generating demand for slow-selling products and produce an immediate cash flow into the business. Likewise, the downside is a decline in profit margin. In this subcategory the following pricing models are identified by the authors. These models include loss leader, special event, cash rebate, low-interest financing, longer payment terms, warranties and service contracts, and psychological discounting.

The last subcategory of retail-based pricing is *discriminatory pricing*, where different prices are charged for the same services depending on the customer. Two degrees of price discrimination pricing are described. First-degree price discrimination pricing is typically contingent on a one-to-one negotiation. Typically, a lot of effort is required to capture the subjective maximum value of a customer. Hence it does not scale well to adopt first-degree price discrimination pricing for commodity products. Second-degree price discrimination pricing is dependent on sales volumes. For example, discount selling products or services in bulk, which is common in wholesale. Or the adoption of student discounts. The core idea behind this subcategory is customer segmentation. In the cloud industry, second-degree discriminatory pricing is rather ubiquitous. AWS S3, for instance, has a bulk-selling price. In this subcategory the following pricing models are identified by the authors. These models are customer segment, product form, image, location, geographical location, dynamic or surge-based, and loyalty programming pricing.

The seventh pricing category is *expenditure-based pricing*. In this category, a unit of "cost" is the core component that underlies each model. The upside to these models is the transparency they provide CSPs, which allows them to know their targeted returns. On the other hand, these modes do not take customer values and the market's supply and demand conditions in consideration. In this category the following pricing models are identified by the authors, namely cost-plus, percentage, and target return pricing.

Resources-based pricing, which is the next category, focuses on scalability and scarcity. Resource-based pricing is adopted across many industries, including the services industry and the IT industry, and is common in many cloud services.In this category the following pricing models are identified by the authors. These models are transaction-based, FTE-based, licensing-based, and time-material-based pricing.

Finally, the ninth and last pricing category is *utility-based pricing*, in which the usage of services is monitored, and the price is determined accordingly. The benefit of utility-based pricing is the ability it provides individuals to access to the cloud services without prerequisites. The downside of models in this category is that they might not go well with new or innovative cloud services features. In this category the following pricing models are identified by the authors. These models are Peak and Off-Peak and fixed cost-based pricing.

3 Contemporary Pricing Models

Fixed pricing			
Pricing Model	**Description**	**Value-based view vs. Cost-based view**	**Implementation**
Pay-as-you-go	Price is determined by the vendor.	Risk lies with the customer.	Implemented.
Subscription	User pays a fixed fee each month.	Customer carries more risk.	Implemented.
Pay for resource	Users pay per the amount of resources they use (e.g. CPU hours or GBs RAM).	This method can be very cost effective for the user.	Implemented.
Hybrid	Dynamic pricing in the sense that the price can change but this usually works with a job-queue where users specify a max price for some task to run when the price is right based on user limits.	Profits are minimal for the provider, but the same can be said about the costs for the users.	Implemented.
Novel financial [7]	Pricing model that includes financial option theory in combination with Moore's law and has a fixed lower and upper limit to pricing which maximizes QoS.	This pricing model is therefore beneficial to providers as the price can be varied and beneficial to users as a maximum QoS is achieved.	Theoretical study with simulation.
Data center optimization	Using this pricing model, requested jobs are scheduled using an algorithm based on whether or not they may be interrupted	Using this pricing model, vendors can minimize electricity costs by efficiently deciding when interruptible jobs should run.	Theoretical study with simulation.
Cost-plus	This model is designed to generate as much revenue as possible.	Risk lies with the customer.	Implemented.

Table 1: Fixed pricing models

Dynamic pricing			
Pricing Model	**Description**	**Value-based view vs. Cost-based view**	**Implementation**
Resource pricing	A variant on the static resource pricing where market conditions determine resource prices.	Can be more profitable for the provider but also introduces price risk for the user.	Implemented.
Genetic pricing	Uses a genetic pricing model that can lead to high profits for the provider but works less well in high-demand and irrational market condition.	Risk lies with the customer.	Implemented.
Value based / novel finance	Pricing is contingent on the added value the customers receive.	Very fair to customers but also rather difficult to implement as value is not an objective measurement.	Implemented.
Competition	Works by supply and demand but between competitors, prices are always determined by competing prices.	Fair to customers.	Implemented.
Federated cloud [8]	Users and providers are both able to buy and sell resources.	This pricing model can lead to very efficient market prices but can be tough to implement due to hardware limitations.	Theoretical with simulation.
Advanced reservations	The service provider can dynamically price into the future, allowing users to allocate resources ahead of time for the dynamic price that seems fair to them.	This can lead to higher profits as customers might be more likely to schedule resources, lowering overall prices but increasing profits during peak hours.	Implemented.
Auction	The price is determined by the auction mechanism.	Risk is fairly distributed between vendors and customers.	Implemented.
Online pricing	Pricing model where the purchasing of goods can only be processed online.	Overall, fair to both parties.	Implemented.
Product mix pricing	Both on-demand and spot instance pricing models are combined to accommodate both predictable and unpredictable workloads.	Primarily designed to increase revenue for the service provider.	Implemented.
Discount pricing	Pricing is (temporarily) reduced for reasons such as product promotion, off-season, cash payment, etc.	Fair to customer; vendor takes on more cost, but can improve their net present value.	Implemented.
Allowance pricing	Discount directed at wholesale customers, commercial clients or SME.	Fair to customer; vendor takes on more cost, but can improve their net present value.	Implemented.
Promotional pricing	A sales tactic where a discount is given within a specified period.	Vendors are rewarded as sales increase and stock level diminishes, however profit margin is negatively impacted. Fair to customers	Implemented.
Discriminatory pricing	Different prices are for the same services charged depending on the customer.	Customer value based perspective.	Implemented.

Table 2: Dynamic pricing models

Customer value-based pricing			
Pricing model	**Description**	**Value-based view vs Cost-based view**	**Implementation**
On-demand	Customer pays for compute capacity by the hour or the second depending on which instances they run. No longer-term commitments or upfront payments are needed.	Value-based.	Implemented [9].
Tier-based	Vendors offer two or more packages, or fixed sets of features, for a specific price. Each tier is customizable to an extent, to fit the customers needs.	Value-based.	Implemented.
Per-user-based	Pricing is based on the number of users a customer has. As the users increase, the pricing increases accordingly.	Value-based.	Implemented.
Per device-based	Pricing is based on the number of devices a customer has. As the users increase, the pricing increases accordingly.	Value-based.	Implemented.
All you can eat (or flat-rate)	Pricing is set for a specified period and covers unlimited access to resources, no user limitations, and no device limitations.	Value-based.	Implemented.
Priority-based pricing	Services are labeled and priced according to their priority	Value-based.	Theoretical.
Perceived value [10]	Pricing is determined by considering what product image a customer carries in his mind and how much he is willing to pay for it.	Value-based.	Theoretical with simulation.
Outcome-based	Pricing is set by considering customers' business outcomes.	Value-based.	Theoretical.
Customer care-based	Price is based on consumer requirements and needs.	Value-based.	Implemented.
Psychological	A pricing model that utilizes specific techniques to form a psychological or subconscious impact on consumers. It integrates sale tactics with price.	Value-based.	Implemented.
Per feature pricing	Somewhat similar to tiered pricing. Customers pay for different features within each tier. As the functionalities increase, the pricing increases accordingly.	Value-based.	Implemented.

Table 3: Value based pricing models

4 Future of Cloud Pricing Models

Pricing is moving further away from the physical box oriented model [6]. It seems to be going in a direction where the operation system does not matter (No OS/NoOPS). Cloud features are part of the CSP and the user only needs to monitor, which is how tools like Kubernetes and Docker Swarm work. This transformation will result in CSP's leveraging its strength to offer innovative pricing models.

In the same paper they argue four challenges for the future of cloud pricing, which are derived from the direction cloud computing is already going.

- Moving from pure cost-based to both value-based and cost-based pricing.
- Driving from statefulness to stateless resource pricing.
- Transferring from mutable to immutable pricing.
- Pricing cloud services for both intrinsic and extrinsic features by consideration of cloud infrastructure lifecycle.

These challenges combined with the fluctuation of demand (which is a given) give CSPs the opportunity to maximize their revenue and profit with the finite resources that they have. On the side of the customer this would mean more flexibility and easier scalability.

The emphasis on these targets also means that pricing will go even further away from the value of hardware because it will be as invisible as possible for the customer. This also makes it easier to pay for the value it has in that point of time (value-based), as opposed to paying the cost of the hardware (cost-based). The vendor will set a price for the resource pool and will want to maximize its use as much as possible.

5 Discussion

Based on the studies we examined, we surmise that pricing models can be sorted in three classes.

First, there is fixed pricing, which can also be seen as static pricing or cost-based pricing. Pricing models in this class are more objectivity inclined. The price is predetermined, thus there is transparency and stability between the CSP and the customer. The risk is predominantly carried by the customer, as they might pay more than the market value [11]. Fixed pricing systems are the easiest to implement and maintain. They are also the most pervasive in the cloud industry.

Second, dynamic pricing and the closely related market-based pricing. Conditions in the market with regards to supply and demand determine the price on a dynamic basis. Because of this, there is a lot more risk involved for both parties when compared to the other two classes. Under the right circumstances, a customer might be able to save on their expenses. In the same way, CSPs might attain more business profit if the the right conditions are met. This creates an added incentive for the CSP ensure the Quality of Service (QoS) for the user. Dynamic pricing systems are harder to implement and maintain than fixed pricing systems [12]. Market dependent pricing, involving bidding mechanisms are even more complex, and introduce more risk [13].

Third and last, is value-based pricing, which approaches pricing from a customer's value perspective. The customer's experience and satisfaction are central components of the pricing model. Like the first two studies we examined, many literary works related to cloud pricing models either do not analyze models of this type, or categorized these models as dynamic. In many cases however, there is an acknowledgement that many of the models that are used in the industry do not take a user-based perspective into consideration. Pricing models in this class are relatively risky for CSPs due to the subjective nature that is inherent in customer's value perception. They are not easy to quantify and plan for. Moreover the implementation and maintenance of such a system are considerably more complex [14]. Nonetheless, there are elements of service-based and performance-based pricing models presently commonplace in the cloud industry. For instance, service-level agreement guarantees regarding service reliability, and tier-based pricing. This is also illustrated by [6]. Customer-value based pricing, however, remains a challenge. Cong et al. [10] developed a user perceived value-based pricing mechanism for cloud markets, that focuses on dynamic profit maximization and achieves favorable results. However, their results are solely based on simulations, so there is no real-world evaluation of their model's performance.

In practice we see that CSPs adopt different pricing models depending on service types and service levels. Among SaaS offerings subscription-based pricing models are the most common, whereas consumption-based pricing is most prevalent for IaaS [15]. [12] analyzed and compare three pricing models, subscription-based pricing, pay-per-use pricing and two part-tariff pricing from the perspectives of CSP's profit, consumer surplus, and social welfare. Three conclusions are drawn. First, both service providers and consumers prefer pay-per-use pricing to subscription-based pricing. Next, firms prefer two-part tariffs when compared to pay-per-use pricing, from a revenue standpoint. Lastly, if there is no additional cost for measuring consumer usage, pay-per-use pricing is the most convenient pricing model, from both a consumer surplus and social welfare perspective.

Besides the risk of overpaying, there are other risks that consumers should be cautious of. Vendor lock-in —or cloud-lock in —is the situation where customers are dependent (i.e. locked-in) on a single CSP technology implementation and cannot easily change to a different vendor, or integrate services from different providers, without incurring substantial costs, legal constraints, or technical

incompatibilities [16]. Other points of consideration for consumers prior to choosing a CSP are customer support, availability of the service, SLA breach penalty, etc. [17].

6 Conclusion

Pricing Cloud computing services is a dynamic field that involves revenue creation and management with optimal resource allocation.

In this paper we researched pricing models in the cloud industry. This was done by reviewing three studies, analyzing contemporary models and exploring the future of cloud pricing models. Based on our findings we determine that certain themes are relevant with regards to cloud pricing models. Themes, such as the notion of consumer-directed fairness, which is prevalent in recent literature and is observable in industry; and the increased importance of what a customer perceives as valuable service and their willingness to pay. These factors are not easy to quantify and require deliberate effort, yet they have a tangible effect on pricing models we see today, in particular on the dynamic pricing and customer value-based models. Hence Quality of Service is an important point of consideration for both cloud service providers and customers. For vendors one of the main benefits of customer value-based models is its ability to maximize profit. The more established fixed pricing models on the other hand, are easy to plan for and more cost effective in setup and maintenance. The downside however, is that they might miss out on profit. Customers tend to find fixed pricing schemes less ambiguous in comparison to other types of models, although these kind of models are also less fair, as they do not consider market supply and demand conditions, or disregard the customers' value perspective.

A cloud service provider will have multiple pricing models. Pricing also varies between vendors for the same requirements. Thus it is incumbent upon customers to make decisions that are informed by their own situation and market dynamics prior to choosing a service provider.

Although there lie challenges ahead, as new services continue to be developed their pricing models are expected to evolve accordingly.

Bibliography

[1] S. Satyanarayana. Cloud computing: Saas. computer sciences and telecommunications, (4), 76-79. *International Journal of Engineering & Technology*, page 01, 01 2012.

[2] K. Jamsa. Cloud computing: Saas, paas, iaas, virtualization, business models, mobile, security and more. jones & bartlett publishers. page 12, 05 2017.

[3] Arenas-Marquez F. J. & Aguayo-Camacho M. Palos-Sanchez, P. R. Cloud computing (saas) adoption as a strategic technology: Results of an empirical study. mobile information systems,. page 12, 05 2017.

[4] May Al-Roomi, Shaikha Al-Ebrahim, Sabika Buqrais, and Imtiaz Ahmad. Cloud computing pricing models: a survey. *International Journal of Grid and Distributed Computing*, 6(5):93–106, 2013.

[5] Devesh Lowe and Bhavna Galhotra. An overview of pricing models for using cloud services with analysis on pay-per-use model. *International Journal of Engineering & Technology*, 7:248, 07 2018.

[6] Caesar Wu, Rajkumar Buyya, and Kotagiri Ramamohanarao. Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. *ACM Computing Surveys (CSUR)*, 52(6):1–36, 2019.

[7] Mario Macías and Jordi Guitart. A genetic model for pricing in cloud computing markets. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 113–118, 2011.

[8] Marian Mihailescu and Yong Meng Teo. Dynamic resource pricing on federated clouds. In *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pages 513–517. IEEE, 2010.

[9] AWS Economics 2 (EC2). `https://aws.amazon.com/ec2/pricing/on-demand/`. Date Accessed: 20-05-2021.

[10] Peijin Cong, Liying Li, Junlong Zhou, Kun Cao, Tongquan Wei, Mingsong Chen, and Shiyan Hu. Developing user perceived value based pricing models for cloud markets. *IEEE Transactions on Parallel and Distributed Systems*, 29(12):2742–2756, 2018.

[11] Sahar Arshad, Saeed Ullah, Shoab Ahmed Khan, M Daud Awan, and M Sikandar Hayat Khayal. A survey of cloud computing variable pricing models. In *2015 International conference on evaluation of novel approaches to software engineering (ENASE)*, pages 27–32. IEEE, 2015.

[12] Se-Hak Chun et al. Cloud services and pricing strategies for sustainable business models: analytical and numerical approaches. *Sustainability*, 12(1):1–1, 2019.

[13] Juong-Sik Lee. *Recurrent auctions in e-commerce*, volume 69. 2007.

[14] Aishwarya Soni and Muzammil Hasan. Pricing schemes in cloud computing: A review. *International Journal of Advanced Computer Research*, 7:60–70, 02 2017.

[15] Ethann Castell. The present and future of cloud pricing models, Jun 2013.

[16] Justice Opara-Martins, Reza Sahandi, and Feng Tian. Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective. *Journal of Cloud Computing*, 5(1):1–18, 2016.

[17] MK Mohan Murthy, HA Sanjay, and JP Ashwini. Pricing models and pricing schemes of iaas providers: a comparison study. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pages 143–147, 2012.

[18] Jay A Alexander and Michael C Mozer. Template-based algorithms for connectionist rule extraction. In *Advances in neural information processing systems*, pages 609–616. Citeseer, 1995.

[19] James M Bower and David Beeman. *The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System*. Springer Science & Business Media, 2012.

[20] Michael E Hasselmo, Eric Schnell, and Edi Barkai. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region ca3. *Journal of Neuroscience*, 15(7):5249–5262, 1995.