Research Paper

Department of Information Systems and Operations Management

LV.-Nr. 0100- Seminar aus BIS (Schiseminar)
im WS 2020/21

Critical Evaluation and Comparison between Proprietary and Open-Source Cloud Systems

Michael Widy (h0965602)

LV-Leitung: ao.Univ.Prof. Mag.Dr.rer.soc.oec. Rony G. Flatscher

Welthandelsplatz 1, A-1020, Wien
Telefon: +43/1/313 36/4622 • Telefax: +43/1/313 36/717
Web: www.wu.ac.at/retail

Declaration of Authorship

I assure,

- to have individually written, to not have used any other sources or tools than referenced and to not have used any other unauthorized tools for the writing of this report.
- to never have submitted this report topic to an advisor in this, nor in any foreign country.
- that this report matches the report reviewed by the advisor.

Date: December 17th, 2020

Michael Widy (h0965602)

Table of Contents

1 Introduction

The following research paper aims to contrast two major cloud systems, namely open-source and proprietary systems. In the first chapter, I will give an overview of cloud systems and provide the theoretical background to them. This is achieved by first explaining the different types of cloud computing, also known as deployment models. After that, the different types of cloud services, also known as service models, will be explained. In chapter 3, a quick introduction to the problem "Open-Source Software (OSS) vs. Proprietary Software" will be given, in which I will argue that open-source systems play a major role in serving individual needs of companies. After that a critical comparison between open-source software and proprietary software will be outlined. In this chapter the main focus will be on the general advantages and disadvantages of open-source and proprietary software which are vital to consider if it comes to choosing the best solution for one's needs. In chapter 5, the positive and negative aspects of both software types will be applied to cloud systems and individual factors that influence one's choice to find the best solution for one's purpose will be outlined. In the final chapter, I will critically evaluate and compare which cloud systems meet requirements of a cloud system in the education sector.

2 Overview and Theoretical Background

Before we are able to discuss and review different software options in cloud systems, we first have to identify and explain some of the underlying concepts of cloud computing. First, we will try to differentiate between different kinds of cloud computing. After that, different kinds of cloud services will be explained. Finally, the difference between open-source vs proprietary software in cloud services will be laid out.

2.1 Different Types of Cloud Computing (Deployment Models)

In the last decade there was a shift into cloud computing. But how each company is handling their needs for digitalization is different. When a company decides to make the shift into cloud computing, it has to consider that each solution also has its advantages and disadvantages. Therefore, in the following section the different kinds of cloud computing systems will be explained, followed by some of their benefits as well as negative aspects which need to be kept in mind when opting for one over the other.

2.1.1 Public Cloud

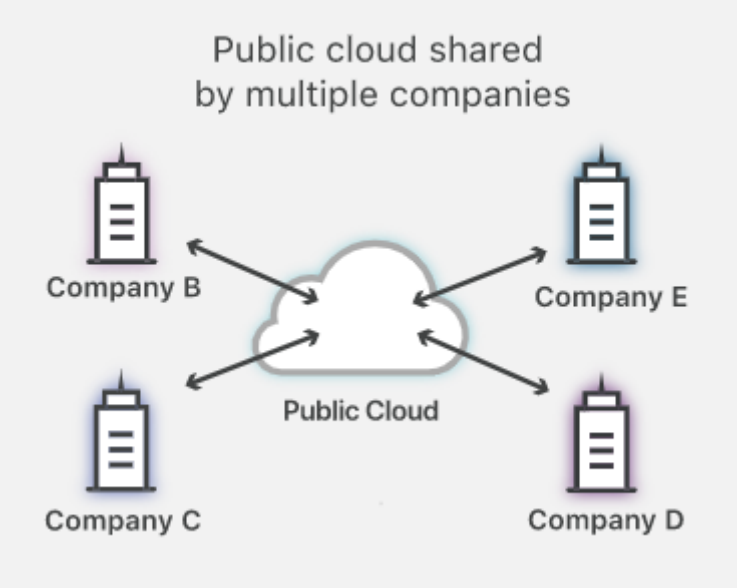

Figure 1: Public Cloud (Cloudflare, 2020)

Public cloud describes the concept of a service provider offering resources to the customers over the internet. These resources might include applications,

infrastructure or storage, which are only accessible through the internet. Some services are offered free of charge (e.g., Gmail), but could be also offered as a pay per use option. Normally, a customer won't know where the data is actually geographically hosted, nor has the customer direct control over the underlying infrastructure. (Balasubramanian & Aramudhan, 2012)

One of the advantages for customers lies in the availability of data on short notice. That means that if a customer needs resources for a short duration, it is possible to deploy additional applications quickly and solve temporary needs without purchasing additional equipment. (Singh & Jangwal, 2012)

2.1.2 Private Cloud

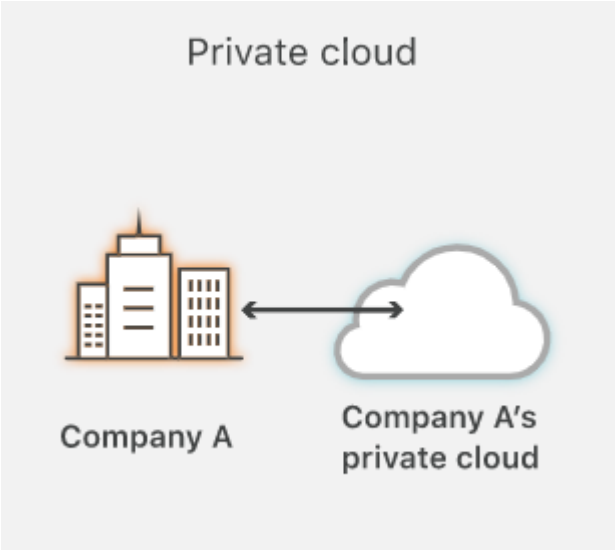

Figure 2: Private Cloud (Cloudflare, 2020)

A private cloud, which is often also referred to as corporate cloud or internal cloud, normally is hosted locally by the corporations themselves. This also means that the corporation typically owns their own infrastructure, or at least has physical control over it on their premises. The company´s IT department can then build, maintain and manage their own cloud, thus resulting in operating their cloud as they see fit and being able to adapt their resources as new requirements arise. (Singh & Jangwal, 2012) Private clouds are usually only accessed by members of their own organization, though access can be granted to third parties. So, the main difference between private and public clouds is, that private clouds do not offer

cloud services for the general public. Instead, their purpose is to primarily serve the needs of their own organization. (Goyal, 2014)

There are also many advantages in managing one's own cloud. For example, being able to provide more security due to the fact of having control over the cloud infrastructure. Moreover, cost savings can be achieved in the long run. Due to several cloud services breaches in the last couple years, companies have decided to turn to private cloud services to gain more data security and privacy, since data is arguably one of the most important resources. (Goyal, 2014)

2.1.3 Community Cloud

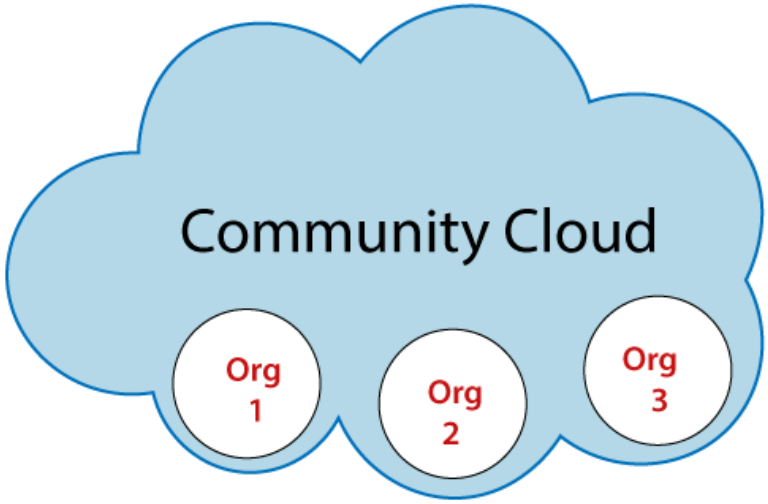

Figure 3: Community Cloud (JavaTpoint, 2018)

Community clouds are a more special kind of cloud computing, as their focus is often more on the consumer side. Community clouds are similar to private clouds, but instead of serving one organization, a community cloud is a private cloud which is shared and used by two or more organizations. (Goyal, 2014)

There are some advantages, like shared costs when setting up a community cloud or shared information, which can be used by the sharing parties. But there is also a flipside to each of these advantages. To name one of it, shared resources could lead to bottlenecks as there is only a fixed amount of bandwidth or storage available on the community cloud. (Goyal, 2014)

2.1.4 Hybrid Cloud

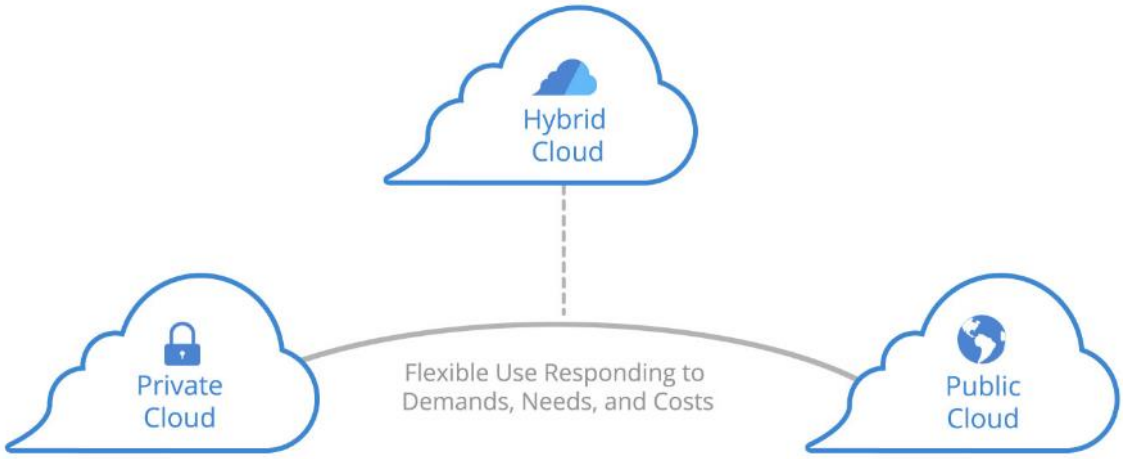

Figure 4: Hybrid Cloud (Bauer, 2018)

Hybrid Clouds are a combination of different kinds of clouds. Hybrid Cloud refers to two or more clouds, like a private and a community cloud combined. While they remain to be two unique entities, this form of cloud system enables data and application portability. (Tariq Ellahi, 2011) This means that they often complement each other if an organization is not ready, or doesn't want to, move their whole data and applications to a public cloud. Not only does a public cloud carry the inherent risk of data insecurity and privacy leaks, but also poses the risk of vendor lock. (Palwe, Kulkarni, & Dongare, 2012).

https://www.businesspost.ie/connected/covid-19-makes-the-case-for-cloud-and-hybrid-it-solutions-21b80b9b

2.2 Different Types of Cloud Services (Service Models)

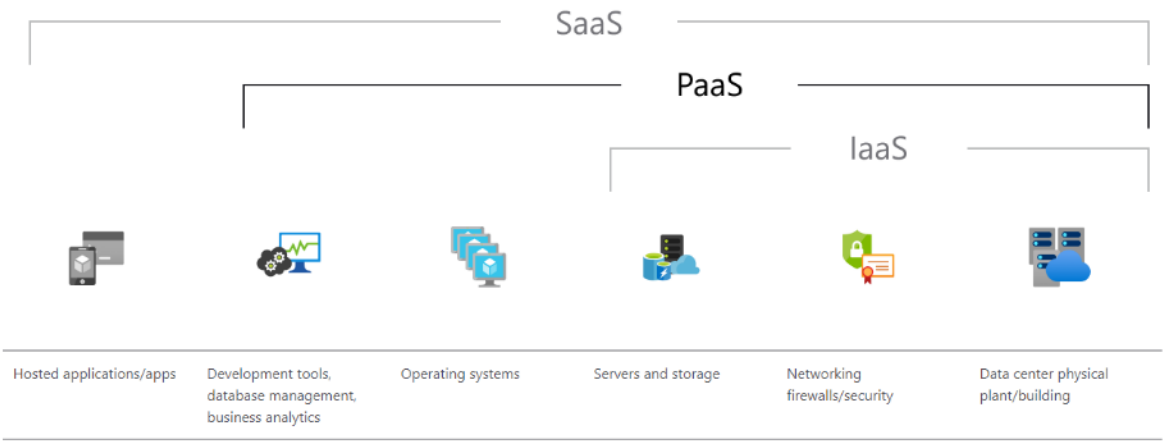

Figure 5: Microsoft Azure – What is paas (Microsoft Azure, 2020)

There are different service models of cloud computing, hereafter also referred to as kinds of cloud services.

These kinds of cloud services are providing customers with the opportunity to deploy applications more quickly, thus providing agility and making it possible to respond to changing demands in their needs quicker. As customers are only paying for the resources that they actually need, this also potentially reduces overall costs. (Sowmya, Deepika, & Naren, 2014)

2.2.1 Infrastructure as a Service (IaaS)

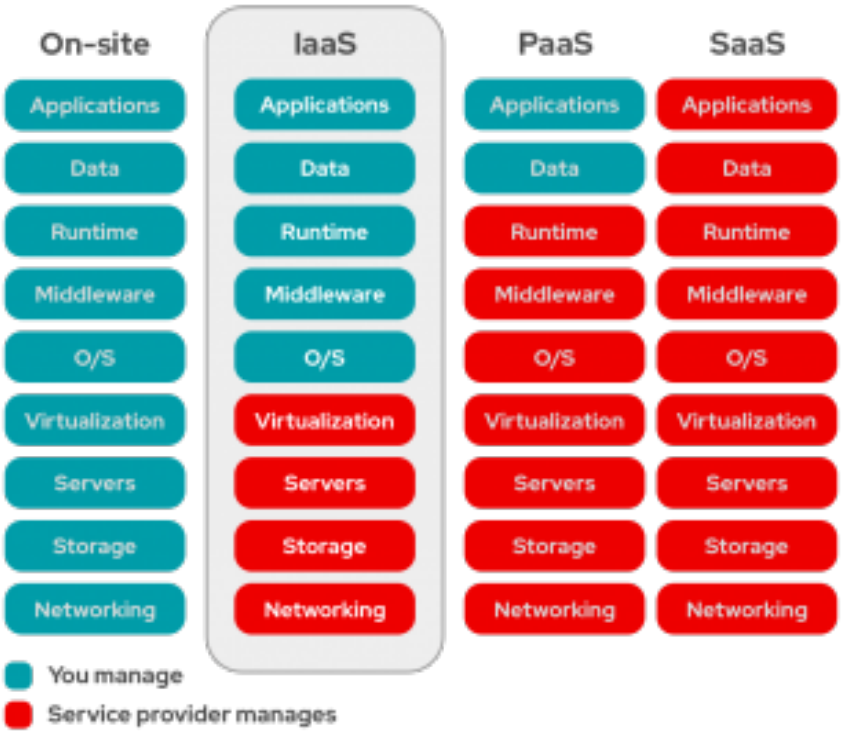

Figure 6: IaaS (Red Hat, 2020)

Infrastructure as a service refers to the idea that fundamental computing resources are being provided by a third party. These resources could, for example, mean storage, processing power and/or virtual machines. The customer can control and use these resources as they see fit, but potential security issues arise because physical control lays in the hands of the infrastructure provider. (Singh & Jangwal, 2012)

Infrastructure as a service therefore mitigates the need for local hardware and because of that is often called Hardware as a Service (HaaS). Some examples for enterprise solutions are Amazon Web Service (AWS), Windows Azure and Rackspace. (Sowmya, Deepika, & Naren, 2014)

2.2.2 Platform as a Service (PaaS)

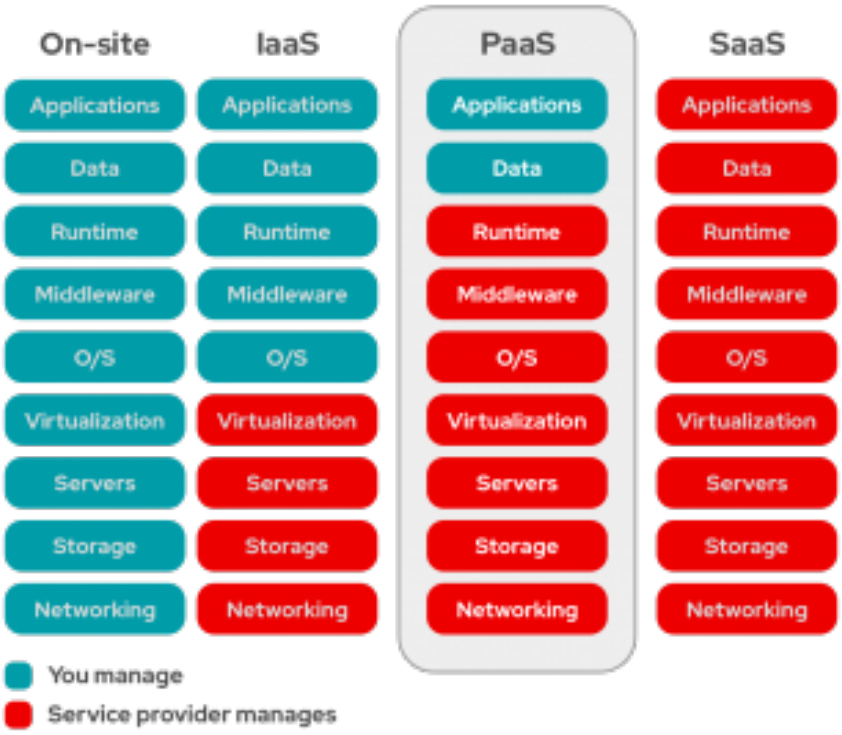

Figure 7: PaaS (Red Hat, 2020)

Even though there it is difficult to sharply draw the line between infrastructure as a service and platform as a service, PaaS enables customers to create and control applications without having to worry about the underlying resources like hardware. (Sahandi, Alkhalil, & Opara-Martins, 2013) This means that resources like databases, web servers and even the operating system automatically grows with the needs of the customer. This reduces the cost of the deployment of applications and the costs of managing the hardware, where these applications run on. (Sowmya, Deepika, & Naren, 2014)

2.2.3 Software as a Service (SaaS)

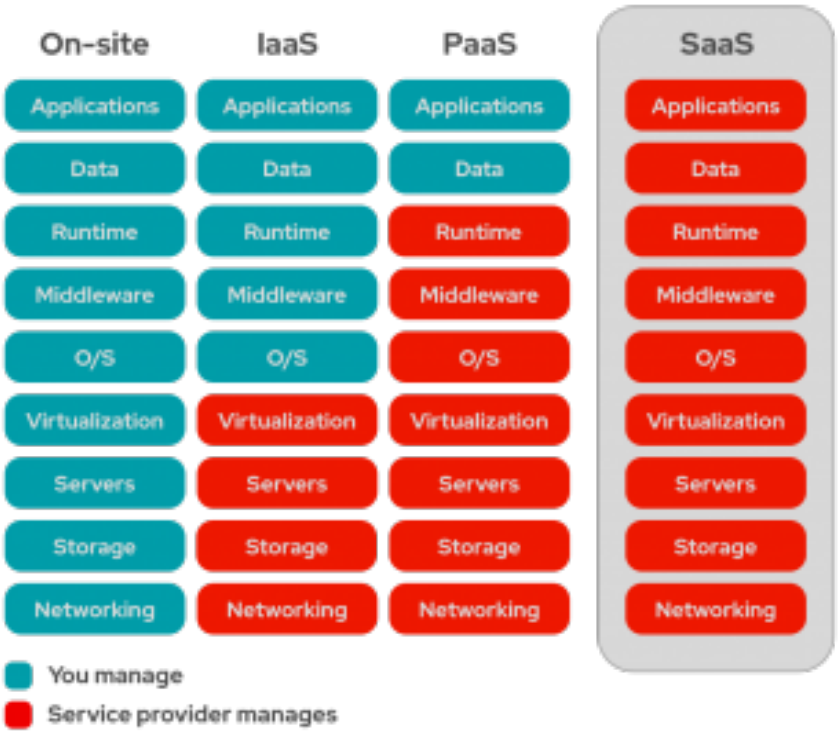

Figure 8: SaaS (Red Hat, 2020)

Software as a Service provides applications to the customers but the customers do not have control over the network infrastructure, hardware or operating system. This means that the software as a service provider manages the entire underlying application, which is being delivered to the customers. These providers are also responsible for keeping their applications secure. (Singh & Jangwal, 2012)

3 Open-Source Software (OSS) vs. Proprietary Software

Figure 9: Open-Source Vs Proprietary Software (eduonix, 2018)

In this section I want to lay out the definitions of open-source and proprietary software. Next, I will briefly explain, why open-source software plays a major role in promoting development and technological advancements in the open-source software sector. After that the advantages and disadvantages of OSS and proprietary software will be analyzed to better understand how these fundamental concepts can be applied cloud systems.

3.1 Open-Source Definition

The fact that the underlying source code of open-source programs is accessible to the general public, does not serve as a unique identifier of such systems. While there is no clear-cut definition for open-source programs, there are several initiatives that aim to establish criteria which define open-source systems.

One of the initiatives mentioned above is the Open Source Initiative 2007. This initiative has outlined the following criteria, which open-source programs should meet. (Open Source Initiative, 2007)

The Open Source Initiative has proposed the following 10 criteria: (Open Source Initiative, 2007)

1. Free Redistribution
2. Source Code
3. Derived Works
4. Integrity of The Author's Source Code
5. No Discrimination Against Persons or Groups
6. No Discrimination Against Fields of Endeavor
7. Distribution of License
8. License Must Not Be Specific to a Product
9. License Must Not Restrict Other Software
10. License Must Be Technology-Neutral

Moreover, there are also other projects, like the GNU project, that have defined a set of criteria to define Open-Source programs. While some of the criteria are somewhat similar, some items are different. The GNU project defines "Free Software" as four essential items that promote freedoms for the users: (GNU, The GNU Operating System and the Free Software Movement, 2020)

- The freedom to run the program as you wish, for any purpose (freedom 0).
- The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
- The freedom to redistribute copies so you can help others (freedom 2).

- The freedom to distribute copies of your modified versions to others (freedom 3) By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.

As stated by Stallman, free software and open-source almost refer to the same range of software programs, however, they are different when looking at their core values. Whereas free software refers to the freedom of the individual user, open-source does not. That is why GNU does not agree with the term "open source" and therefore won't use it. (GNU, GNU Project, 2020)

For the sake of completeness, it should be mentioned that there are also other terms used for open-source concepts. For example, "FLOSS" stands for "Free/Libre and open-source Software" or "FOSS" for "Free and Open-Source Software". These terms aim to forge links between free software and open-source software. (Stallman, 2016)

Since both, free and open-source systems stand in opposition to proprietary systems, it will be sufficient to differentiate between the two juxtaposing principals in this research paper, hence a detailed comparison between all three concepts is not within the scope of this paper.

Moreover, viewed from a political perspective, the two concepts mentioned above stand in strict opposition to one another and are subject to strong idealistic viewpoints by the respective supporters of each system. (GNU, GNU Project, 2020)

3.2 Proprietary Software Definition

Proprietary software describes the idea of imposing restrictions on the software. These restrictions can be imposed by the company or individuals, who developed the software. The term "proprietary" derives from the Latin word "proprietas" and means property, suggesting that a program developed by a certain entity is one's property, hence this firm or person has the right to decide how it is used. (The Linux Information Project, 2005)

One of the major restrictions is that the source code is kept secret. But there are often also other restrictions in end user license agreements (EULAs), which are obligatory for end users to comply with. These often include that it is forbidden to make unauthorized copies, reverse engineering the software or to use the software on more devices than the license allows you to. (The Linux Information Project, 2005)

It is worth mentioning, that the source code of some proprietary software programs is released to the public domain. This does not mean that they are automatically free or open-source software. This type of release often happens after a proprietary software program ends its life cycle and is abandoned, therefore it is sometimes named abandonware. (Wikipedia, 2020)

3.3 Why OSS matters

Over the last couple years open-source software (OSS) has seen a striking success. (Zhaoli & Sang-Yong, 2005) But OSS is more than just the code being available. OSS also means that the code "must be available for redistribution without restriction and without charge". But it's not only the licenses which makes OSS special. Most of the advances in computing nowadays are a product of the "hacker culture" and this culture is produces innovative and high-quality software. (O'Reilly, 1999) Certainly, a successful and one of the most famous OSS projects is Linux. Also, Apache is a well-known software. (Zhaoli & Sang-Yong, 2005)

A repeatedly upcoming question is why people contribute to something that is free. The answer to this question is versatile. Often, there are intrinsic motivations, like intellectual stimulation. However, there are also extrinsic motivations, like career concerns. (Lerner & Tirole, 2002)

Another vital question to answer is, how different entities benefit from open-source software. First, not all businesses or also individual users have unlimited resources for their technical needs. So, one solution is to save money on software and hence be able to cut costs which results in maximizing in one's profits. However, another, but far more important point is, that open-source software increases the landscape of competition. When offering a software for free, you indirectly increase usage of the software. When other people start using the product a community emerges. Through this community wonderful things can happen. For example, the open-source software developer starts to improve the functionality and usability of the program. (Pankaja & Mukund Raj, 2013) If competing proprietary software exists in the same space, they certainly start to recognize what is happening. Therefore, open-source software increases innovation through competition, so that proprietary software vendors cannot stand still. As a result, even if the competition comes from commercial open-source software, consumer surplus and social welfare increases. (Xing, 2014)

4 Critical Comparison of Open-Source Software (OSS) and Proprietary Software

In this section, I will include a critical comparison of open-source and proprietary software based on various distinguishable factors such as cost, service and support, innovation, security, usability, vendor lock-in, standards, availability, transparency and reliability.

4.1 Advantages and Disadvantages of Open-Source Software

4.1.1 Cost

As OSS is freely usable, there are no costs when using OSS like activation or a monthly subscription fee. This means that companies can save costs instead of running a cost-intensive system. These systems can be used as long as there are needed and updated and are not subject to an expiration date (Pankaja & Mukund Raj, 2013)

4.1.2 Service and Support

While proprietary software providers offer ongoing support to its paying customers, service and support in OSS is highly reliable on the online network community. If users seek help, they often find helpful support in the online community, which delivers support on blogs and forums. Moreover, turning to the online community to ask for help if working with an active OSS project, one can often expect a loyal and engaged community who is ready to help if needed. However, it should be also mentioned, that asking questions and receiving answers also requires a basic understanding and skills of the software, which you are using. Trouble shooting could sometimes be even faster than if proprietary software is used. (Pankaja & Mukund Raj, 2013)

4.1.3 Innovation

OSS offers freedom and flexibility when it comes to adapting the software to their needs. Providing users with that flexibility and freedom enables and fosters innovation. However, one potential problem is that not all users share their

adaption with the community. There is a debate about the fact that customized changes might hinder future support of the software because this could result in not being able to deploy future updates. This could leave the user with an older version. It is also worth mentioning that especially young firms in the open-source space are struggling to attract sufficient funding. (Pankaja & Mukund Raj, 2013)

4.1.4 Security

OSS is generally perceived as secure, though we have to differentiate between operating system software and other OSS. Linux, an open-source operating system one of the most secure operating software. For other OSS, it depends mostly on the size of the community and if big companies are using different OSS in their company or an OSS is their product. Another development which increases security to open-source systems is the fact that with the increasing importance of open-source, OSS are increasingly reviewed and corrected by peers (Pankaja & Mukund Raj, 2013)

4.1.5 Usability

In the past, OSS had been criticized for not being as user-friendly as proprietary software, due to the fact that they weren't reviewed by usability experts. This might have resulted in usability problems for users. Moreover, the lack of requirements for documentation or user-manuals of such systems resulted in reduced possibilities to create new tools. However, the growing support by the open-source community maximizes opportunities for creation and development (Pankaja & Mukund Raj, 2013)

4.1.6 Vendor Lock-In

A vendor lock-in means, that once a software is used it is difficult or costly to switch to another software provider. Due to the fact that open-source projects often use open standards, possible vendor lock-ins are unlikely.

4.1.7 Availability

OSS is freely available over the network. (Pankaja & Mukund Raj, 2013)

4.1.8 Transparency

OSS code is normally available freely on the internet and can be accessed by anyone who downloaded it. This means that every user can read, modify and build their own version from this code. So, it is also possible to review the code and understand what it is doing exactly. Moreover, a new modified version of the code can be distributed to other users. (Pankaja & Mukund Raj, 2013)

4.1.9 Reliability

OSS is available on different sites throughout the web, this also means that there are probably many different modified versions of a software out there. (Pankaja & Mukund Raj, 2013) This is why some users could experience a bad overall performance or crashes on their hardware because a modified version might cater an individual's need, however, might not be perfectly transferable to every other system, which might result in degradation of the original.

4.2 Advantages and Disadvantages of Proprietary Software

4.2.1 Cost

Proprietary software costs vary heavily, ranging from very little to a few hundred thousand dollars. The cost heavily depends on the complexity of the required software. The cost of proprietary software is determined by three different cost factors. First, there is a base fee for the software. Secondly, there is the cost of integration and services. Finally, there is usually an annual licensing and/or support fee. However, for these costs proprietary software often offers a more customized product, higher functionality, higher levels of security, greater ability to scale and probably on-going support and training and often requires lower technical skills from users using it. (Pankaja & Mukund Raj, 2013)

4.2.2 Service and Support

One of the biggest advantages of proprietary software is service. The support through a proprietary software vendor is a key selling point for organizations, where its users do not possess a great technical expertise. This means, if a user manual is not enough, the proprietary software vendor is likely to assist and offers individual solutions. This results in a reduction of risk for the organizations. They can be fairly certain, that if they need help, they can turn to the proprietary software vendor, who has knowledge of its products and services. Due to the fact that many organizations choose proprietary software providers over open-source because of its service premise, many proprietary software providers compete on service. This leads to increasing customer service levels in the proprietary software provider space. (Pankaja & Mukund Raj, 2013)

4.2.3 Innovation

When using proprietary software, the users are often not allowed to view the code, let alone change it. This fact could be viewed as a disadvantage. However, this also ensures that the software is secure and reliable. The software is also often customizable for specific use cases, which offers flexibility for the end users. Proprietary software vendors also invest in their R&D to make sure they can provide their clients with new products. Proprietary software vendors also have online communities, where individuals can share ideas and best practices. This also

creates values and fosters innovation, when the proprietary software vendor then adapts its product to changing needs. So, an advantage for the user is, that they don't have to invest in R&D themselves because upgrades are typically part of the service package. Proprietary software vendors also benefit from their R&D, because they have to ensure that they do not become redundant, so with continuous investment in innovation, they can stay in the game. (Pankaja & Mukund Raj, 2013)

4.2.4 Security

The security of proprietary software is always a debate. Open-source operating systems are perceived more secure than many proprietary options, but on the other hand total solutions from proprietary vendors are also viewed more secure due to the fact, that they are developed in an isolated environment by a closed team. Moreover, the code is heavily audited to avoid back doors and to minimize the risks of any bugs or other issues in the code. (Pankaja & Mukund Raj, 2013)

4.2.5 Usability

One advantage of proprietary software over OSS is that proprietary software vendors often employ usability experts, who are targeting the appearance of the software to the desirable audience. This is why proprietary software often appears more tailored. So, usability in proprietary software is normally ranked quite high. The proprietary software vendor also offers detailed guides and user manuals. Due to this fact users have a much faster learning curve. This could also be supported by seminars and targeted training courses to help the users to use the software to its maximum capability. Even though many people are thinking of proprietary software as a very closed solutions, the proprietary software of today often offers many ways of third-party enhancements. (Pankaja & Mukund Raj, 2013)

4.2.6 Vendor Lock-In

Vendor lock-ins are often imposed intentionally on users of proprietary software by not publishing their standards, so that other programs cannot use their files. This can often cause problems when users want to switch to another software.

4.2.7 Availability

Proprietary Software is available from respected companies, which own the right to offer these products. A possibility to use the software for free is sometimes offered via trail versions. (Pankaja & Mukund Raj, 2013)

4.2.8 Transparency

Proprietary software only offers user interfaces to work with. This means that the user cannot take a look at the internal structure and does not know other details of the internal processing. (Pankaja & Mukund Raj, 2013)

4.2.9 Reliability

Proprietary software is very reliable due to the fact, that it's developed by a closed team and only finished products find their way to the customers. (Pankaja & Mukund Raj, 2013)

5 Advantages and Disadvantages of OSS vs Proprietary Software with regard to Cloud Systems

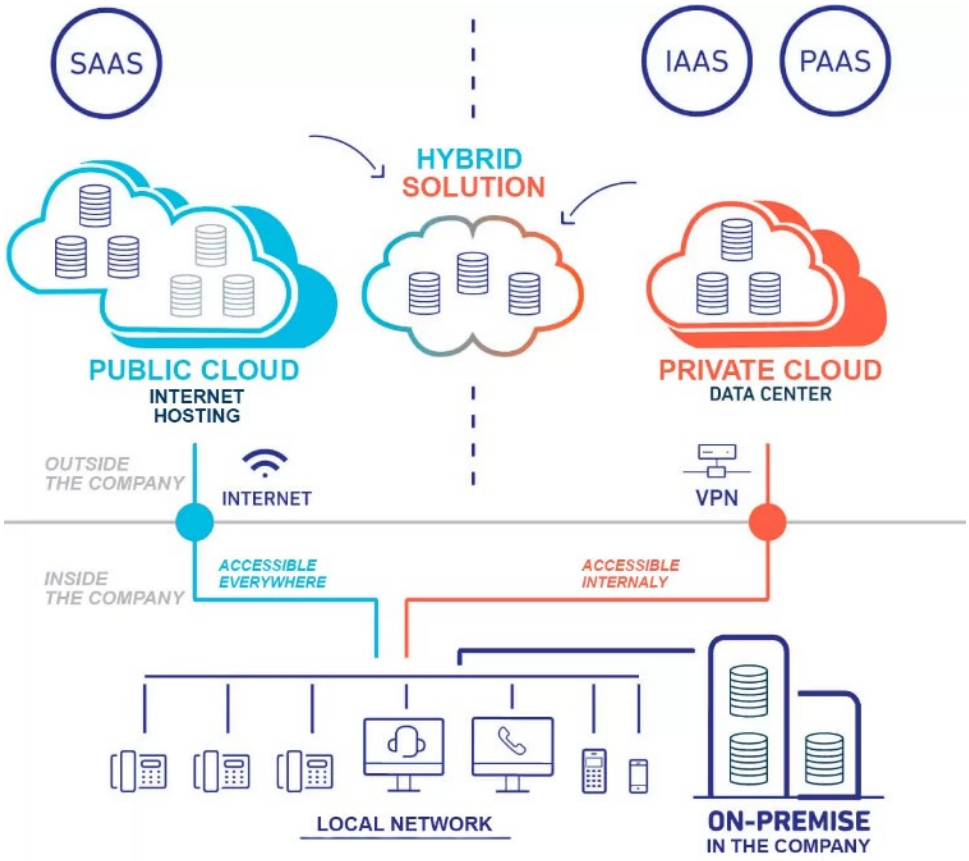

Figure 10: Complex Hybrid Cloud (Jembere, 2019)

In the previous section I have covered the advantages and disadvantages of open-source vs. proprietary software systems. In this chapter these findings will be applied to outline the positive and negative aspects of proprietary and open-source cloud systems, which are actually the focus of this paper. However, an intensive analysis of cloud systems reveals the complexity of such systems and hence makes it generally more difficult to make a clear-cut definition just between the two concepts, private cloud and public cloud.

Figure 10 illustrates the complexity of such could systems. Numerous configurations are possible. For example, a private cloud is possible which is hosted with a proprietary software. But also a public cloud with open-source software could be an option. This is why the lines clearly get blurry, when choosing one over the other and the choice heavily depends on different aspects of the

implementation. So, this chapter will serve as guide on what to take into account when making the decision of choosing open-source or proprietary software in cloud systems.

5.1.1 Cost

For cost comparisons between proprietary and open-source cloud systems the same concepts apply as comparing proprietary and open-source software. However, because running a cloud system is more complex than a single software, it is also possible that, even though open-source cloud systems are free, buying additional support is necessary.

5.1.2 Service and Support

For service and support the same advantages and disadvantages of open-source software and proprietary software applies to cloud systems.

5.1.3 Innovation

Regarding innovation advantages and disadvantages of open-source software and proprietary software applies to cloud systems, although it greatly depends on different factors. Even on proprietary IaaS platforms developers can build innovative new products, which can spark new innovations.

5.1.4 Security

For security principles of open-source software and proprietary software can be applied to cloud systems.

5.1.5 Usability

For usability the advantages and disadvantages outlined in the previous chapter can be applied to cloud systems. However, virtualization through containers and working with them, was a big step in the last few years, especially for the open-source community. Nowadays it is fairly simple to deploy various containers to get your own cloud system running without much knowledge. So, this is certainly a win for the open-source community.

However, this positive developments in usability using containers comes at a cost. The advantage of simple deployment solutions can result in vendor lock-in.

5.1.6 Vendor Lock-In

Normally, vendor lock-in only has to be taken into consideration, when choosing between open-source and proprietary software. But vendor lock-in is especially important when adopting various cloud systems. Although it is important when choosing between deployment models, the main focus lies on which cloud service to choose. This is because of the level of control you hand over to the vendor, which in return makes it more difficult to leave the vendor at some point.

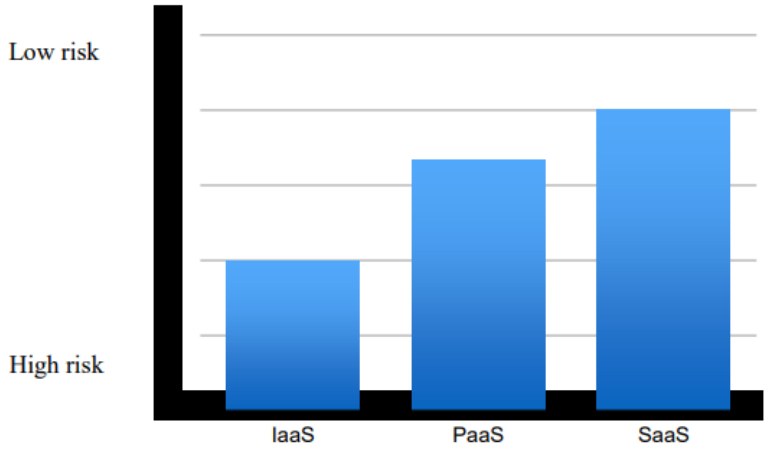

Figure 11: Vendor lock-in - Cloud Services (Kamel, 2015)

IaaS poses the lowest risk because when you use the infrastructure for example only for cloud storage, it should be easy to switch to another vendor. It is also easier to use other products on top of their service, if you only use cloud storage. (Kamel, 2015)

PaaS poses a higher risk of vendor lock-in because at this stage the vendor often forces you to use their APIs to access data on the back-end. (Kamel, 2015)

At the SaaS level, all key components are controlled by the vendor. (Kamel, 2015)

If we take into account what happens when a proprietary software vendor offers SaaS, we can assume that the vendor lock-in is pretty strong. So, when opting for proprietary software one has to take into account that switching costs increase

going from IaaS to SaaS. But the same can probably said about open-source when using a cloud service, although switching cost should be lower.

5.1.7 Availability

For Availability the same applies as to proprietary and open-source software.

5.1.8 Transparency

Regarding transparency is greatly depends on the used cloud service model. The more control over the hardware, operating system, data management tools is given up, the less transparency is received. For the software behind the implementation, when choosing to run it on-premise, the same applies as to proprietary and open-source software.

5.1.9 Reliability

For reliability the same as for open-source and proprietary software applies.

6 Critical Evaluation and Comparison between Proprietary and Open-Source Cloud Systems (in the Education Sector)

For a better practical perspective, we take a closer look on how open-source and proprietary software are used for cloud systems in the education sector. Therefore, we divide the basic needs of students into three subsections: Data; because all students and teachers eventually have to store their documents. Content management & learning; to provide documents, homework assignments or quizzes to the students and Live communication; because in COVID-19 times like these, they often have to communicate this way.

In these subsections we will compare an open-source to a proprietary solution using the same principals as in the previous chapters.

6.1 Data – Nextcloud vs Office 365

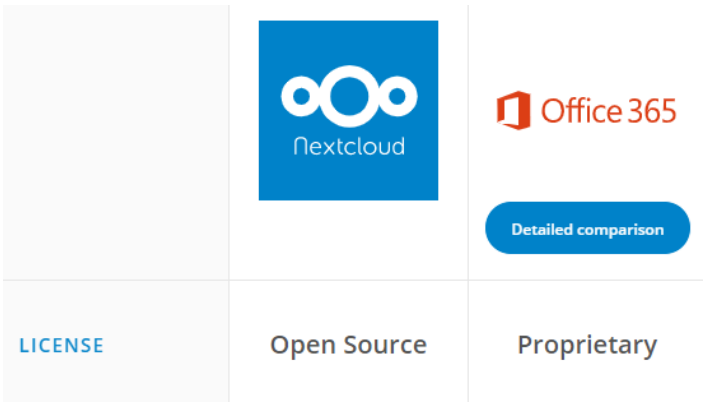

Figure 12: Nextcloud vs O 365 - License (Nextcloud, 2020)

Nextcloud (nextcloud.com) is open-source, whereas Office 365 (microsoft.com) is a proprietary solution.

6.1.1 Cost

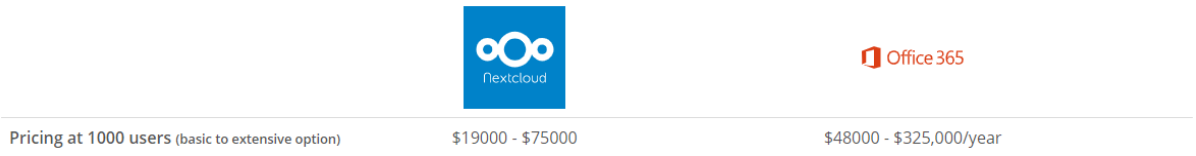

Figure 13: Nextcloud vs O 365 - Cost (Nextcloud, 2020)

In the figure above we can see the pricing for 1000 users, which ranges $19,000 - $75,000 for Nextcloud applications and $48,000 - $325,000 a year for Office 365 solutions. Although Nextcloud is free to use, there are other costs involved like hardware or if hosting has to be done off-premises. As we can see, Office 365 is substantially more expensive than Nextcloud.

For all of the following figures shown, the right column represents the property of Nextcloud, whereas the left column shows Office 365.

6.1.2 Service and Support

Both offer service and support for Enterprise solutions. Nextcloud offers help for home users through a forum. Although, the forum is hosted by Nextcloud, it is stated that it is about users helping each other.

6.1.3 Innovation

Looking at the changelog, it is safe to say that both options greatly innovate.

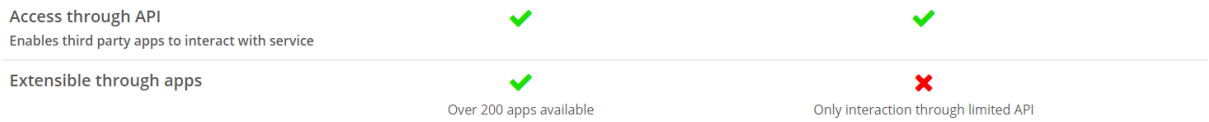

Figure 14: Nextcloud Vs O 365 - Innovation (Nextcloud, 2020)

It is also possible to interact with both software solutions through APIs, although Office 365 does not offer the possibility to extend the software through apps, whereas there are over 200 apps available for Nextcloud that allow extension.

6.1.4 Security

Figure 15: Nextcloud vs O 365 - Security 1 (Nextcloud, 2020)

With Nextcloud being open-source it is possible to self-host your own instance of Nextcloud. Office 365 does not offer this solution, so for security concerns it is a double-sided sword as the figure below will show.

Privacy, control and security

Full Audit trail	Unlimited	Limited to 90 days for some data
Protection of metadata	✔	✖
GDPR compliance	100% certified	Risky due to Cloud Act
Encryption	Always on own infrastructure	Server-side in MS data center

Figure 16: Nextcloud vs O 365 - Security 2 (Nextcloud, 2020)

6.1.5 Usability

Other features

Social network features	✔	✔
Online Office	✔	✔
Groupware	✔ mail server not included	✔
Task/project management	✔ Kanban, task list. No GANT chart view currently	✔
Workspaces (a collaborative space associated with each folder for context, todo lists, notes etc)	✔	✖
Mobile Device Management (Block devices, remote wipe, notification to devices)	✔	✖
Strictly enforce data and meta data locality	✔	✖
Limited Guest accounts	✔	✖
Secure Mailbox (Outlook add-in sending email body & attachments only through secure server)	✔	✖
Protect shares with password, expiration date	✔	✔
Video Verification (enforce video call with recipient to verify identity before granting access to share)	✔	✖
Ransomware Recovery (automated recovery of data after ransomware attack based on file analysis)	✔	✖ Only time-based, after detection of ransomware based on file names. Risks overwriting useful changes and missing ransomware without predictable names.

Figure 17: Nextcloud vs O 365 - Features (Nextcloud, 2020)

Both options offer many user-manuals and documentation online. But as the figure above shows, Nextcloud offers more features to improve usability for certain scenarios, like “Limited Guest accounts” or “Video Verification”.

6.1.6 Vendor Lock-In

Figure 18: Nextcloud vs O 365 - Vendor Lock-In (Nextcloud, 2020)

Office 365 offers, as the figure shows, no vendor lock-in prevention.

6.1.7 Availability

For availability general rules apply as discussed in previous chapters.

6.1.8 Transparency

As for transparency for Nextcloud it greatly depends on the implementation. When self-hosted, you have direct control and you can check for yourself if the version of the software is the same one as currently developed on GitHub. As O 365 is always off-side hosted, this possibility does not exist and therefore the general rules for proprietary software applies.

6.1.9 Reliability

Regarding reliability the same can be said like in the general chapter about open-source software and proprietary software.

6.2 Content Management & Learning – Moodle vs MS Teams

In the following chapter I would like to discuss Moodle and MS Teams. Moodle (moodle.org) is an open-source solution, whereas MS Teams (microsoft.com) is a proprietary solution.

MS Teams does not work on-premises. (Buckley, 2020) As it is part of Office 365 (Teams, 2020), everything said about Office 365 in the comparison above, also is valid in this comparison.

6.2.1 Cost

As Moodle is similar to Nextcloud resource-wise, the only difference is that Moodle focuses on learning solutions. The costs are relatively equal. What is more, Moodle and Nextcloud announced further integration steps between their software on October 3, 2000 (Press release - Nextcloud, 2020), making them an ideal team against MT Teams/Office 360 in the future.

6.2.2 Other comparisons points

As Moodle and Nextcloud show similar characteristics in service and support, innovation, security, usability, vendor lock-in, availability, transparency and reliability no further comparison is needed.

6.3 Communication – Jitsi vs MS Teams

Now I would like to compare Jitsi to MS Teams for the purpose of communication. Jitsi (jitsi.org) is open source, whereas MS Teams (microsoft.com) is a proprietary solution.

As MS Teams does not work on-premises, this solution has to be paid for. Jitsi on the other hand can be self-hosted, as the code is freely available on GitHub.

6.3.1 Cost

For MS Teams the principles explain in the previous chapter apply.

For Jitsi it depends massively on usage. As Nextcloud and Moodle are very modest in their requirements, the same cannot be said about Jitsi. Jitsi is very CPU and bandwidth demanding as benchmarks from the jitsi community are showing (Easy Jitsi, 2020). To get a good idea about what different hardware is able to archive, there is also a good thread on the community jitsi.org forum. (community.jitsi.org, 2020)

So, the costs for 1000 students for running jitsi are difficult to calculate and open to debate, but they should be comparable to MS Teams. Further analysis on these cost factors could be interesting.

6.3.2 Other comparisons points

As for all the other comparison points discussing the general principles for open-source software, open-source software with regard to cloud systems and the previous explanation of MS Teams, are also applicable to this comparison.

6.4 Possible further research

In this research paper cloud systems in the education space were not the main topic, but merely a way to grasp and apply the concepts of the debate between proprietary and open-source cloud systems in a practical way.

Therefore, further research on this topic could be important for future decisions being made in the education sector.

7 Conclusion

To critically evaluate and compare between proprietary and open-source cloud systems it is vital to take a look at the problem at a more abstract level. So, in order to get a better understanding, we have to take into account all different types off deployment models, service models and what type of software (open-source or proprietary) we use along this chain.

As there is clearly no easy solution to deploy the perfect cloud system, there are different paths to choose from. The most important aspect is to understand why a path is the right one, and what possible advantages or disadvantages this path brings.

8 References

Balasubramanian, R., & Aramudhan, M. (2012, 10). Security Issues: Public vs Private vs Hybrid Cloud Computing. *International Journal of Computer Applications*.

Bauer, R. (2018, April 19). *Backblaze*. Retrieved November 2, 2020, from Confused About the Hybrid Cloud? You're Not Alone: https://www.backblaze.com/blog/confused-about-the-hybrid-cloud-youre-not-alone/

Buckley, C. (2020, March 10). *buckleyPLANET*. Retrieved November 20, 2020, from https://www.buckleyplanet.com/2020/03/microsoft-teams-in-a-hybrid-environment.html

Cloudflare. (2020). Retrieved November 1, 2020, from What Is a Public Cloud? | Public vs. Private Cloud: https://www.cloudflare.com/learning/cloud/what-is-a-public-cloud/

community.jitsi.org. (2020). Retrieved Nobember 25, 2020, from Recommended Server Specs for 2020?: https://community.jitsi.org/t/recommended-server-specs-for-2020/32041

Easy Jitsi. (2020). Retrieved November 30, 2020, from https://docs.easyjitsi.com/docs/benchmarks/

eduonix. (2018, September 3). Retrieved from Open Source Vs Proprietary Software: https://blog.eduonix.com/marketing/open-source-vs-proprietary-software/

GNU. (2020, 10 6). *GNU Project*. Retrieved from Why Open Source misses the point of Free Software: https://www.gnu.org/philosophy/open-source-misses-the-point.en.html

GNU. (2020, 11 10). *The GNU Operating System and the Free Software Movement*. Retrieved from https://www.gnu.org/philosophy/free-sw.en.html

Goyal, S. (2014, 3). Public vs Private vs Hybrid vs Community - Cloud Computing: A Critical Review. *I.J. Computer Network and Information Security*, pp. 20-29.

JavaTpoint. (2018). Retrieved September 27, 2020, from Community Cloud: https://www.javatpoint.com/community-cloud

Jembere, D. A. (2019, September 8). *EzTalents*. Retrieved September 29, 2020, from Hybrid Cloud and Fog Computing for SMEs in Ethiopia: https://eztalents.com/2019/09/hybrid-cloud-and-fog-computing-for-smes-in-ethiopia/

Kamel, M. A. (2015). Vendor Lock-in in the transistion to a Cloud Computing platform. Sweden: KTH ROYAL INSTITUTE OF TECHNOLOGY.

Lerner, J., & Tirole, J. (2002, 6). Some Simple Economics of Open Source. *The Journal of Industrial Economics*, pp. 197-234.

Microsoft Azure. (2020, September 28). Retrieved from What is PaaS?: https://azure.microsoft.com/en-us/overview/what-is-paas/

Nextcloud. (2020). Retrieved November 10, 2020, from https://nextcloud.com/compare/

Open Source Initiative. (2007, 3 22). Retrieved from The Open Source Definition: https://opensource.org/osd

O'Reilly, T. (1999, 4). Lessons from Open-Source Software Development. *Communications of the ACM*, pp. 32-37.

Palwe, R., Kulkarni, G., & Dongare, A. (2012, 06). A New Approach To Hybrid Cloud. *International Journal of Computer Science and Engineering Research and Development (IJCSERD)*, pp. 1-6.

Pankaja, N., & Mukund Raj, P. (2013, 7). Proprietary software versus Open Source Software for Education. *American Journal of Engineering Research (AJER)*, pp. 124-130.

Press release - Nextcloud. (2020, October 3). Retrieved from Moodle and Nextcloud announce further integration steps at Nextcloud Conference: https://nextcloud.com/press/pr20201003-4/

Red Hat. (2020). Retrieved November 6, 2020, from Types of cloud computing: https://www.redhat.com/en/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud

Sahandi, R., Alkhalil, A., & Opara-Martins, J. (2013, 3). CLOUD COMPUTING FROM SMES PERSPECTIVE: A SURVEY-BASED INVESTIGATION. *Journal of Information Technology Management*.

Singh, S., & Jangwal, T. (2012, 04). Cost breakdown of Public Cloud Computing and. *International Journal of Computer Science & Information Technology*.

Sowmya, S., Deepika, P., & Naren, J. (2014, 6). Layers of Cloud – IaaS, PaaS and SaaS: A Survey. *International Journal of Computer Science and Information Technologies*.

Stallman, R. (2016, 11 18). *GNU Project*. Retrieved from FLOSS and FOSS: https://www.gnu.org/philosophy/floss-and-foss.en.html

Tariq Ellahi, B. H. (2011). The Enterprise Cloud Computing Paradigm. In B. G. Buyya, *CLOUD COMPUTING - Principles and Paradigms* (pp. 97-120). Wiley.

Teams, M. (2020). *Microsoft Teams*. Retrieved November 20, 2020, from https://support.microsoft.com/en-us/office/how-do-i-get-access-to-microsoft-teams-fc7f1634-abd3-4f26-a597-9df16e4ca65b#:~:text=Microsoft%20Teams%20is%20available%20in,E3%20prior%20to%20its%20retirement.

The Linux Information Project. (2005, July 3). Retrieved from Proprietary Software Definition: http://www.linfo.org/proprietary.html

Wikipedia. (2020). Retrieved November 5, 2020, from List of proprietary source-available software: https://en.wikipedia.org/wiki/List_of_proprietary_source-available_software

Xing, M. (2014, July). The Impact of Commercial Open Source Software on Proprietary Software Producers and Social Welfare. *Journal of Industrial Engineering and Management*.

Zhaoli, M., & Sang-Yong, T. (2005). *Open Source vs. Proprietary Software: Competition and Compatibility.* University Libary of Munich.

9 Table of Figures