Cloud VR

Secure, Fast and Distributed Virtual Reality Solutions

Leon Koster

A thesis presented for the degree of
Bachelor of Sciences

Academie Creative Technology (ACT)
Saxion University of Applied Sciences
Netherlands
May 2020

1 Acknowledgements

I would like to acknowledge my university and teachers, Matthijs van Veen, Yiwei Jiang and Hester van der Ent, for their help with the creation of this report.

2 Abstract

Recent years have seen the maturing of the cloud streaming and Virtual Reality (VR) space. Now that these technologies are mature enough by themselves, first endevours in cloud VR streaming have been made, but the combination of these technologies is still in its infancy.

Motivated by this Saxion and the Industrial Reality Hub (IRH) want to investigate the feasibility of this technology for use-cases inside of the IRH. To achieve this goal they created a multi-phase project, of which this report is the first phase, aiming to explore the current state of technology and research and using these results to support the later phases of the project. To this end the research question is as follows: "*Which Technology stack(s) for building a cloud Virtual Reality (VR) streaming application satisfies the clients demands (low latency, security, QoE) best ?* "

Contents

1 Acknowledgements **1**

2 Abstract **1**

3 List of Figures and Tables **3**

4 List of Abbreviations **4**

5 Introduction **5**

6 Preliminary Problem Statement **5**

7 Problem Analysis **6**
7.1 Latency 6
7.2 Security 7
7.3 Architecture for a cloud VR system 7

8 Theoretical Framework **8**
8.1 Cloud Streaming/Cloud Computing 8
8.1.1 Definition 8
8.1.2 Existing Solutions and Technology 8
8.1.3 System Architecture types (for a cloud VR system) 9
8.1.4 Latency 10
8.2 Constraints of Virtual Reality 10
8.3 Security (for streaming data) 10
8.3.1 Encryption 11
8.3.2 Identification and Access Management (IAM) 11
8.3.3 Latency Implications 11
8.4 Components of a cloud VR pipeline 12
8.4.1 Available components 13

9 Literature Review **15**
9.1 Cloud Streaming and Latency Reduction 15
9.2 Security 16

10 Final Problem Statement **17**

11 Research Questions **18**

12 Methodology **19**
12.1 Proposed Technology Stacks 19
12.1.1 NVIDIA CloudXR (+ NVIDIA Quadro on Azure) 19
12.1.2 WebRTC Prototype 1 20
12.1.3 WebRTC Prototype 2 21

13 Experiments **22**

14 Results **23**

15 Discussion **24**

16 Conclusion **25**

17 Appendices **28**
17.1 Cloud Computing SWOT Analysis 28
17.1.1 Cloud service provider 28
17.1.2 In-House Server 29
17.2 Cloud service providers 29

3 List of Figures and Tables

List of Figures

1 Cloud Server, Remote Edge, Local Edge visualized (Hou et al., 2017) 9
2 Example System Architecture 10
3 Overview of components in a typical cloud VR pipeline 12
4 Overview of NVIDIA CloudXR Prototype (Nvidia, 2020a) 19
5 Overview of WebRTC Prototype 1 20
6 Overview of WebRTC Prototype 2 21

List of Tables

1 Available Components 13
2 SWOT Cloud Service Providers 28
3 SWOT In-House Server 29
4 Cloud Service Providers by Revenue 29

4 List of Abbreviations

AES Advanced Encryption Standard. 11, 13, 14

FoV Field-of-View. 9, 15

GPU Graphics Processing Unit. 7, 19

HMD Head Mounted Display. 8, 9, 10, 12, 13, 15, 20, 21

IAM Identification and Access Management. 11

IRH Industrial Reality Hub. 1, 5

ms Milliseconds. 6, 8, 10, 15

MTP Motion-to-Photon. 8, 10, 15, 16, 18

QoE Quality of Experience. 1, 10, 15, 17, 18

RGB Red Green Blue. 6

TLS Transport Layer Security. 11, 14

VR Virtual Reality. 1, 5, 6, 7, 8, 9, 10, 12, 15, 16, 17, 18, 19, 20

5 Introduction

Recent developments in the field of Virtual Reality (VR) offer all kinds of opportunities in the field of training and entertainment. For training purposes, the audiovisual entry into a virtual world is where the biggest value is. The capabilities of artificial environments allow users to manage scenarios and experiences that cannot be simulated in the real world. VR also allows users to access the virtual training at any time and less physical facilities are required for exercises. Examples VR experiences include training maintenance at high altitudes (such as windmills), working under heavy loads and weather conditions in construction (Strukton) or maintenance on naval ships (Thales). These companies (and more) form the Industrial Reality Hub (IRH), which is one of the stakeholders of this project.The IRH is an industry consortium of 17 partners in AR/VR - The European digital innovation hub for industrial applied Augmented and Virtual Reality, stimulating cooperation and innovation between companies, government and knowledge institutes, resulting in world class business, knowledge and facilities. The hub is the AR/VR Fieldlab in the Dutch Smart Industry program and is recognized by the European commission as Digital Innovation Hub for industrial AR/VR (IRH, n.d.).

6 Preliminary Problem Statement

One of the essentials for a good Virtual Reality (VR) experience is a powerful computer system to render semi-realistic worlds. However, there are two problems here. First, this type of system is not available in every location. Certainly if realistic images have to be rendered in the simulation, it requires specialized and expensive machines that are difficult to move.

The second problem is that for rendering the VR training scenario, all kinds of data about the scenarios need to be available on the system. This can pose a problem when it concerns sensitive information, for example about all kinds of information defence systems or business sensitive information.

The hypothesis is that both of these problems can be resolved by a cloud rendering solution. By separating the rendering and displaying locations, VR systems become much more versatile. For example with only a lightweight client necessary these experiences could be offered as a Pay-What-You-Use service, which would make them more accessible to a broader audience. Furthermore security would be improved since sensitive data never leaves the secure server location.

The aim of this report is to investigate the feasibility of a streaming based VR approach, with emphasis on Latency reduction and Security. Qualitative research methods will be used to gain in-depth insights about existing solutions and the current state of research into this topic. The data will be contextualized via a literature review of recent research papers and capabilities of existing solutions when applied to the research problem.

7 Problem Analysis

Together with the companies from the Industrial Reality Hub mentioned in the Introduction, Saxion wants to investigate how virtual reality can be rendered in the cloud in a safe and efficient manner. This involves looking at state-of- the art technology in the field of virtual reality, cloud computing, rendering and machine learning for one complete CloudVR pipeline. There are a multiple research directions in the overarching project (Multi-User Experiences, GPU Scaling, etc), however this report will focus on the following:

7.1 Latency

Current market players such as Google Stadia (Google, 2019), GeForce Now (Nvidia, 2020b) and Xbox xCloud (XBox, 2019) already offer cloud gaming services that stream games over the internet. Powerful servers are used for rendering games that are then streamed to users in real time. A bottleneck with this technology is the latency (delay). This is because user input is first sent to a server, which renders these new images, after which they are sent back to the users, all without disturbing them. The mentioned platforms all use network optimization. Low latency is very important for VR, where head movements should be converted to images in under 20 Milliseconds (ms), to prevent motion sickness (Abrash, 2012). The research for techniques for reducing latency is one of the spearheads of the CloudVR project. The following research directions are relevant here:

Network optimization As with the platforms described above, network optimization is one of the techniques which needs to be investigated. The question is to what extent an optimized network can reduce latency and how it relates to the quality of the network connection.

Two-step rendering One of the options to bypass latency is to render in two steps. The delay is not so much reduced, but avoided. The server renders next to RGB also positions and BRDF variables for each pixel. Afterwards on the user's (less powerful) hardware adjustments are made so that the image corresponds to the current position of the user. By sending additional data, the user's local client can extrapolate the correct information and construct a frame that represents the correct head position in the last frame, meanwhile it is waiting for the correct next frame from the server.

Behavioral prediction Another possibility to reduce latency is by predicting user input through machine learning. This will mainly revolve around it analyzing head movements to find out what behavior can be expected. With this information we can render any part of the virtual world before it is viewed by users. If this information is then forwarded from the cloud to the location of the VR experience, what information is displayed can be selected on the spot.

7.2 Security

If an application contains sensitive data, it is advisable to keep the data in a secure location. Previously this was impossible with VR applications, due to their high demand for computing power which meant that VR applications could only be run on a powerful, local computer. This in turn means that the sensitive data (e.g. A 3D model created from CAD drawings) is available directly on the machine and thus could be extracted from the GPU for example. In a cloud VR setup the sensitive data would remain on the (secure) server and only the results will be streamed to the local (unsecured) device, preventing unauthorized access since the data is never streamed directly to the local device, only the visual results. Researching how to maximise security for the clients data is the secondary major focus of this report.

7.3 Architecture for a cloud VR system

One of the questions to be answered is what the CloudVR architecture should look like in terms of hardware and software. The major questions in this topic are whether to use an existing cloud computing service provider or an in-house server and which technologies for Latency reduction and Security fit best in the chosen architecture set up.

8 Theoretical Framework

In order to thoroughly understand the aim and subject of the report, it is important to explore different existing solutions and literature. Therefore, the subjects that will be discussed in the following theoretical framework are Cloud Streaming/Cloud Computing, Virtual Reality and Security in a streaming context. Within this theoretical framework definitions of the subjects will be given, as well as current insights into these subjects. The topics reflect knowledge needed to understand the problem space. Together all of the topics make up the 360 scan. Then this knowledge will be applied to the research problem by creating an overview of the individual parts of a cloud VR system and of the available components to create one.

8.1 Cloud Streaming/Cloud Computing

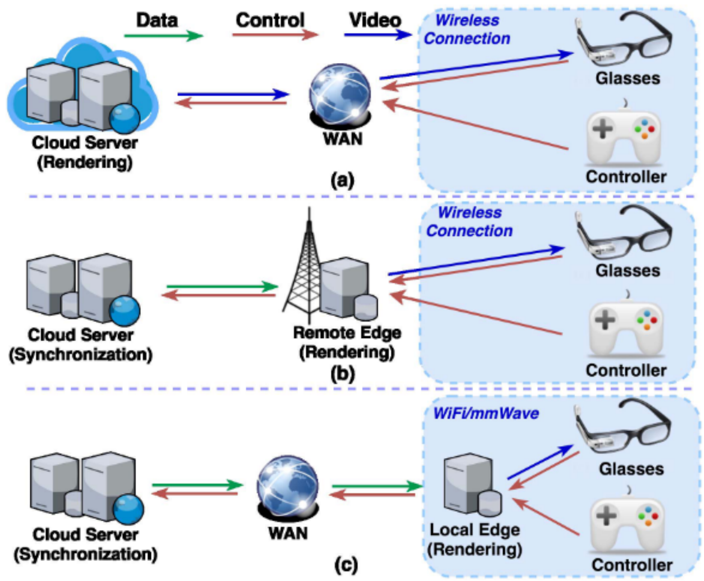
8.1.1 Definition

According to Armbrust et al. (2010) Cloud computing is defined as follows: ”Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the data centres that provide those services.” (Armbrust et al., 2010) We can then further define Cloud streaming as the applications that are delivered over the internet as a service.

8.1.2 Existing Solutions and Technology

Several commercial gaming cloud streaming services already exist, such as Google Stadia (Google, 2019), XBox XCloud (XBox, 2019) and Nvidia GeForceNow (Nvidia, 2020b). These applications deliver conventional games from a powerful computer in a server to the client device at home. Despite initial setbacks, cloud streaming is now a mainstream technology. The start of 2020 also saw the first experimental cloud VR streaming development kits, such as Nvidia's CloudXR (Nvidia, 2020a), and closed beta's for commercial cloud VR streaming services (Shadow, 2020) (Available on Windows, macOS, Ubuntu, Android and iOS) . Additionally the first commercial retail product with cloud VR has been released (Zerolight, 2020), however it runs on custom made HMD's and not on consumer platforms . There is also a variety of Infrastructure-as-a-service (IaaS) platforms, such as Amazon's AWS (Amazon, n.d.), Microsoft's Azure (Microsoft, n.d.) and Google's Cloud Platform (Google, n.d.-a), that provide generic computing power and storage in a cloud computing/streaming context. These services generally cannot achieve the latency requirements of cloud VR streaming (Shi & Hsu, 2015) as it requires an extraordinarily low latency of $>$20ms from Motion-to-Photon (MTP), where most (game) streaming applications have a higher tolerance for latency. Some companies actively develop technology to minimize latency exactly for purposes like this (e.g. enabling compute power as physically close to the end user as possible (Amazon, 2020), but generally as time and technology progress the capabilities of cloud streaming services will grow alongside.

Figure 1: Cloud Server, Remote Edge, Local Edge visualized (Hou et al., 2017)



8.1.3 System Architecture types (for a cloud VR system)

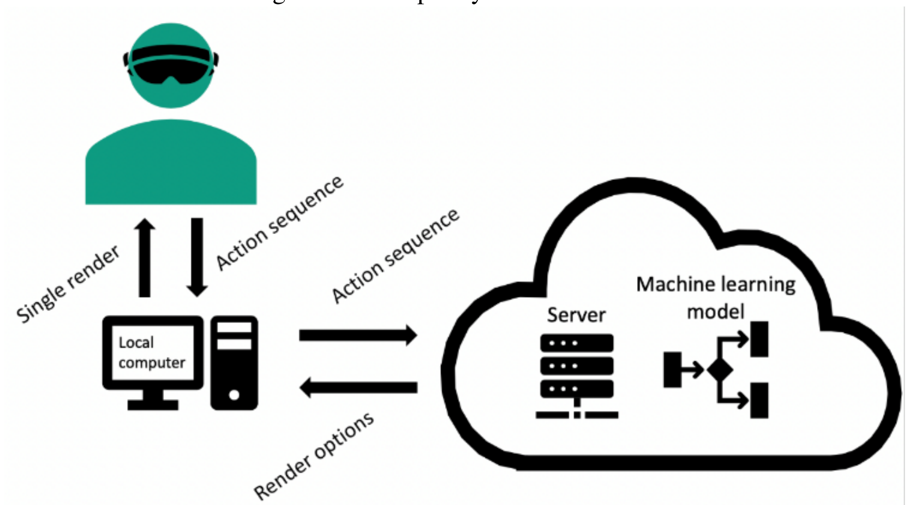
One of the main considerations when designing a cloud VR streaming application is the decision to either use a Cloud, Remote Edge or Local Edge computing device for the rendering of the frames (See Figure 1 and Hou et al., 2017):

- A cloud server renders the Field-of-View (FoV) (current view) remotely and streams the corresponding video to the user's Head Mounted Display (HMD).
- A Remote Edge sever receives information about the context from a cloud server, renders the appropriate frame and streams the video to the user's HMD. The main advantage here is that edge servers are located closer to the end user (thus improving response time and saving bandwidth)
- A Local Edge server receives compressed models as well as textures, renders it locally and streams the video to the user's HMD.

For the purposes of this research I will ignore Local Edge system architectures. The reason is that one of the major motivations for this report was the desire to keep data as safe as possible, which in this case means keeping in the cloud. A Local Edge system, by design, requests and receives business data to render the frame for the user locally. For this reason a Local Edge approach would be the wrong direction to research in. An example architecture of a cloud VR solution that keeps the business data in the cloud can be seen in Figure 2.

Furthermore one also has to consider if the application will be hosted on a cloud service provider or on an in-house server. Both ways have advantages and disadvan-

Figure 2: Example System Architecture



tages, which are elaborated upon in the Appendices, and the decision should be made based on the unique circumstances of each customer.

8.1.4 Latency

The most important metric for a system architecture is the latency between the user input, such as movement of the HMD, and the updated frame appearing on the users display. Recent measurements of cloud gaming services measure this latency at between 135 and 240ms (Chen et al., 2019). This is acceptable for most games, except maybe high intensity reaction games. VR unfortunately has severely stricter latency requirements, which are elaborated upon in Constraints of Virtual Reality

8.2 Constraints of Virtual Reality

As mentioned before, when developing a VR application, there are a few physical constraints that developers need to be aware of. The most important threshold to know is the 20ms MTP delay. Upon input from the HMD, the developer has to display a new rendered image within an average of 20ms to avoid motion sickness for users. The more this threshold can be undercut, the better the chances to have an acceptable gameplay experience without motion sickness. Interaction input, such as the input from the controllers, can safely be processed at delays of $>$100ms without any negative repercussions in terms of Quality of Experience (QoE).

8.3 Security (for streaming data)

From a technical standpoint there are 2 major categories of security implementations: Encryption and Access management. There are additional measures that companies can take such as having consistent security protocols and educating employees, but for this report the focus is on technical solutions:

8.3.1 Encryption

Encryption is the practice of scrambling data so that unauthorized users cannot use the data. Only an authorized party in possession of the decryption key can un-scramble the data and subsequently use it. One such encryption technologies is the Advanced Encryption Standard (AES), which comes with three different key sizes: 128, 192 and 256 bits. In 2016 it was estimated that it would take 500,000,000,000 years to decrypt just one AES-128 key. To encrypt the data in delivery, a technology such as the Transport Layer Security (TLS) can be used, which encrypts the data based on a shared secret that was negotiated at the start of the session, thus making the data only usable for the server and client who have the decryption key.

8.3.2 Identification and Access Management (IAM)

An IAM solution tracks users and what they are allowed to do. There are multiple existing solutions for tracking the users privileges, but for this report only cloud based services are relevant since the premise of this research is the ability to have a cloud solution. All major cloud providers have IAM solutions in their ecosystem and in case of a in-house server a independent IAM service provider can satisfy that requirement.

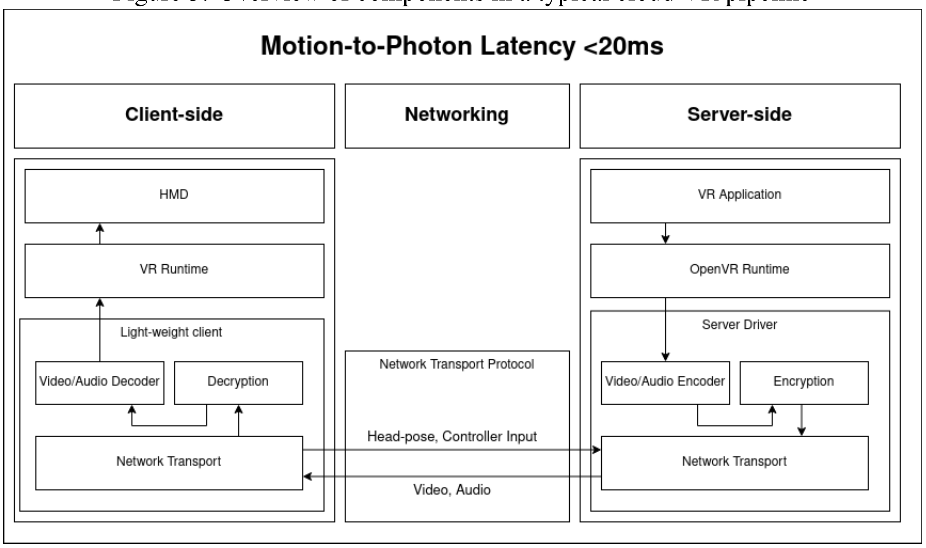
8.3.3 Latency Implications

As discovered by existing research, encryption does not greatly influence performance of the video stream in terms of latency (Kaknjo et al., 2019).

8.4 Components of a cloud VR pipeline

In this section all the individual components of a cloud VR pipeline are being presented. Furthermore an overview of pre-made components will be presented and which part of the pipeline they address.

Figure 3: Overview of components in a typical cloud VR pipeline



Server-side rendering The actual VR application will be running on the cloud server and use the servers hardware to render the game. The application will be running on the OpenVR SDK, which allows access to VR hardware from multiple vendors without requiring that applications have specific knowledge of the hardware they are targeting (Valve, 2016).

Server-side encoding The rendered frames will be encoded with a video compression codec before they are sent to the networking layer.

Server-side encrypting Before transmitting the data (encoded frames) over the network it will be encrypted to maximise security.

Networking Through the network connection both the output (rendered, encoded and encrypted frames) and the input (HMD position and controller input) will be exchanged between the server and the client.

Client-side decrypting Once received from the networking layer, the frames are decrypted to prepare for decoding.

Client-side decoding Once decrypted to usable packages, the data will be decoded and then sent to the VR Runtime

Client-side rendering / displaying The decoded frames will be warped to fit the lenses of the HMD and then finally be displayed to the user.

8.4.1 Available components

Table 1: Available Components

Name	**Description**	**Solves**
NVIDIA CloudXR SDK	”NVIDIA CloudXR™, a groundbreaking technology built on NVIDIA RTX™, delivers VR and AR across 5G and Wi-Fi networks. With NVIDIA GPU virtualization software, CloudXR is fully scalable for data center and edge networks” (Nvidia, 2020a).	Server-side encoding, Server-side encrypting, Networking, Client-side decrypting, Client-side decoding, Client-side rendering / displaying
Seurat	Seurat is a system for image-based scene simplification for VR. It converts complex 3D scenes with millions of triangles, including complex lighting and shading effects, into just tens of thousands of triangles that can be rendered very efficiently on 6DOF devices with little loss in visual quality (Google, 2018).	Server-side rendering
H.264	H.264 is a video compression standard based on block-oriented, motion-compensated integer-DCT coding.[1] It is by far the most commonly used format for the recording, compression, and distribution of video content. It supports resolutions up to and including 8K UHD.	Server-side encoding, Client-side decoding
VP8	VP8 is an open and royalty free video compression format.	Server-side encoding, Client-side decoding
Advanced Encryption Standard (AES)	AES is a specification for the encryption of electronic data.	Server-side encrypting, Client-side decrypting

Transport Layer Security (TLS)	TLS is a cryptographic protocol designed to provide communications security over a computer network by utilizing the AES technology.	Server-side encrypting, Client-side decrypting, Networking
WebRTC	With WebRTC, you can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice- and video-communication solutions. The technology is available on all modern browsers as well as on native clients for all major platforms (Google, n.d.-b).	Networking
WebXR	The WebXR Device API provides the interfaces necessary to enable developers to build compelling, comfortable, and safe immersive applications on the web across a wide variety of hardware form factors.	Server-side rendering, Client-side rendering / displaying

9 Literature Review

9.1 Cloud Streaming and Latency Reduction

Within the last decade the cloud computing space has expanded rapidly and with it the possibilities. Today, even individuals can set up an experimental cloud (Virtual Reality (VR)) gaming streaming solution from pre-made components (TayoEXE, 2019) (Riboulot, 2020). For less experimentally inclined customers, there are complete services, such as the one from cloud computing company Shadow (Shadow, 2015) who recently announced a closed beta for their dedicated VR streaming service (Shadow, 2020). Other major players in the cloud gaming scene are Google's Stadia (Google, 2019), Microsoft's XBox XCloud (XBox, 2019) and Nvidia's GeForceNow (Nvidia, 2020b), all of which were launched recently (>1 year old (Stadia, GeForceNow)) or have not even been released to the public (XCloud) . Early releases, especially Stadia, were quickly overwhelmed on launch and faced public scrutiny for failing to living up to their promises of turning any device into a gaming computer. Since then those services made improvements to their Quality of Experience (QoE) and transitioned into a mainstream technology service.

To facilitate the needed QoE, cutting edge technology is used to enable the necessary performance. Modern video compression codecs, like the AV1 codec introduced in 2018 (AllianceForOpenMedia, n.d.), are getting better at compressing high-resolution video streams and together with an application like WebRTC (Google, n.d.-b) which offers latency optimizations via peer-to-peer networking and more, they lay the foundation for modern cloud streaming applications. Technologies like Google's Seurat Image-Based Scene Simplification System (Google, 2018) and the Shading atlas streaming technique developed by the Graz University of Technology (Müller et al., 2018) offer even further optimizations in areas other than networking and transmitting data.

Research Papers like the ones from Liu et al., 2018 or from Shi et al., 2018 demonstrate the viability and technical feasibility of cloud VR streaming . They developed solutions to achieve and undercut the 20 Milliseconds (ms) Motion-to-Photon (MTP) barrier while streaming VR content. 20ms is the agreed upon threshold between receiving user head movement to displaying the frame on the Head Mounted Display (HMD), to avoid inducing motion sickness (Abrash, 2012). One such solution is a low latency control loop that streams VR scenes containing only the user's Field-of-View (FoV) and a latency adaptive margin area around the FoV. The additional margin allows the clients to render locally at a high refresh rate and compensate for the head movements before the next frame arrives, all of which contributes to the QoE (Shi et al., 2018). The technique known as 'Adaptive FoV' was explored by a multitude of research papers. In essence the optimization is to send only what the user sees (their FoV) and an adaptive area around it, to facilitate for local head movement before the next frame arrives. The idea of rendering only what the user has to see to keep up the immersion is well established within the game development community. View Frustum culling and Occlusion culling (Wikipedia, 2020) are widely used in games to increase performance, whereas Adaptive FoV aims to decrease latency by reducing the payload of network transmissions. Yet another angle of attack leverage's the power of parallel rendering, encoding, transmission and decoding, together with a Remote VSync

Driven Rendering approach to minimize MTP latency (Liu et al., 2018). The prototype for that experiment was based on commodity hardware, which further demonstrates the feasibility of cloud VR streaming.

9.2 Security

10 Final Problem Statement

As explored in the Theoretical Framework there are many existing products/technologies for all the individual parts of a cloud VR system. With this knowledge, the product will consist out of several pre-made components that are going to be combined to create a working prototype. Since there are already products on the market that prove the feasibility of creating a working solution by utilizing these technology components (Zerolight, 2020), there is no need to develop novel technology. Keeping this in mind, the problem statement of this report shifts towards finding the correct combination of available technology to satisfy the latency and security demands of the clients. The main problem is that the clients do not know which technology stacks work in practice. To identify if a tech stack satisfies the requirements, a prototype of the proposed stack will be created and subsequently compared to a ”traditional (local)” solution to measure the effectiveness of the prototype in terms of QoE and latency.

11 Research Questions

Main Question:

Which Technology stack(s) for building a cloud Virtual Reality (VR) streaming application satisfies the clients demands (low latency, security, QoE) best ?

Sub Question 1: Which technology stacks are available ?

Sub Question 2: What is the best way to measure MTP latency and how does it compare to a traditional VR setup?

Sub Question 3: How does security compare to a traditional VR setup?

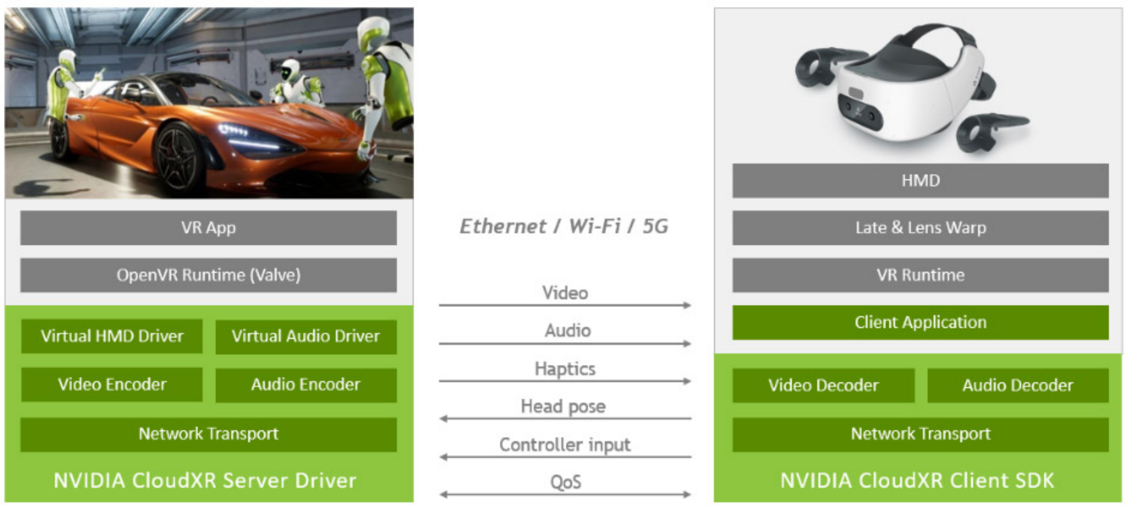
12 Methodology

12.1 Proposed Technology Stacks

In this section different technology stacks will be presented, all of which meet the requirements as detailed in the previous section.

12.1.1 NVIDIA CloudXR (+ NVIDIA Quadro on Azure)

Figure 4: Overview of NVIDIA CloudXR Prototype (Nvidia, 2020a)



The CloudXR SDK from NVIDIA offers a complete solution to stream VR/AR experiences from server to client. As the only complete package in this list it is a good starting point to create a cloud VR streaming prototype. After becoming familiar with the SDK the next step would be deploying it on a Server. Since Azure is the chosen cloud provider of Thales, one of the major stakeholders, it would be the first choice.

Pros:

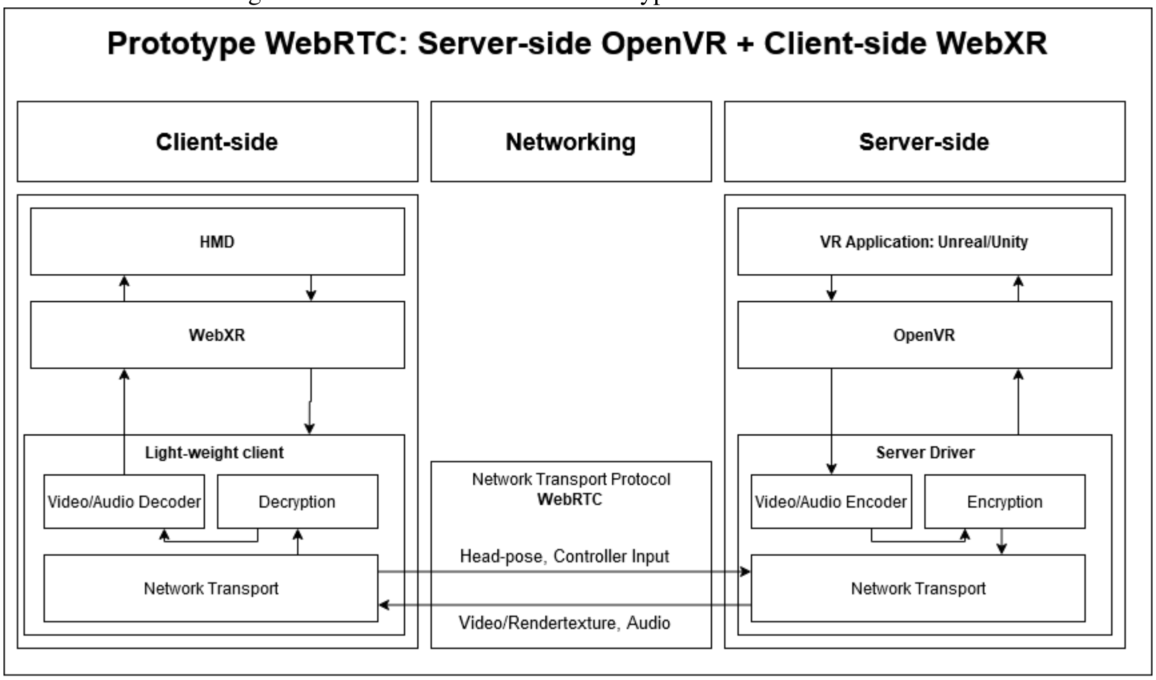
+ Only complete solution
+ Increased prototyping speed
+ Custom made for streaming VR content
+ Works with cutting-edge GPU's

Cons:

- Forced to use NVIDIA products
- Not guaranteed to get access to solution (Have to apply to NVIDIA)
- Limited control about the solution
- Limited documentation about the solution, since it is brand new

12.1.2 WebRTC Prototype 1

Figure 5: Overview of WebRTC Prototype 1



WebRTC is one of the premier web technologies to enable real time communications. Since it allows for streaming video and generic data it is a good candidate to create a cloud VR streaming prototype, because it can transfer both the video and input data. As it has a focus on real time communication it is optimized to reduce latency by default, however it is unclear if this is enough optimization by itself to support streaming VR content. To enable the application to run ”normally” OpenVR will be used to provide a virtual interface of the physical hardware on the removed remote server. By receiving pose updates from the Client and using those updates to create a virtual HMD the application can be developed like a local VR application.

Pros:

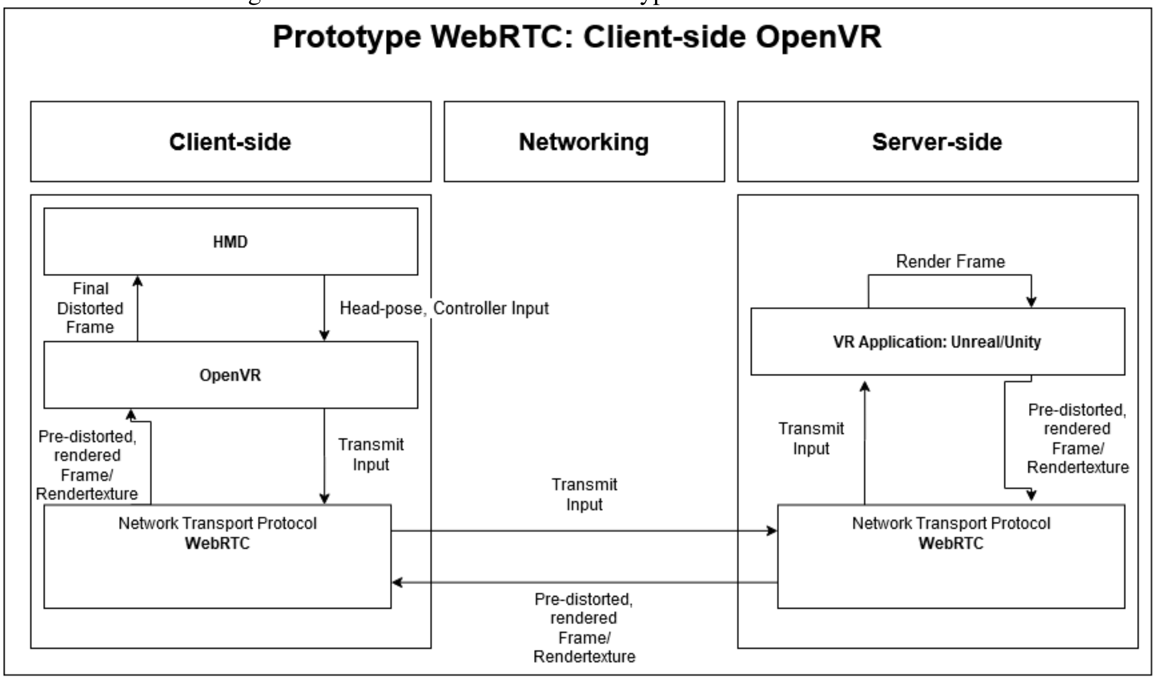
+ Works for the majority of HMD’s + platforms
+ Mostly Open source
+ WebRTC is well documented and supported

Cons:

- Lower performance Web Technologies (but there are native clients for all major platforms available)
- Higher complexity due to more components
- OpenVR + WebXR are sparsely documented

12.1.3 WebRTC Prototype 2

Figure 6: Overview of WebRTC Prototype 2



This protoype candidate is similar in design to the previous one, in the sense that it uses WebRTC for networking and OpenVR for interacting with a HMD. The key difference is that the application on the server does not know it is rendering specifically for VR. All the distortion operations, to generate an image for the HMD from the recieved frame, will be done on the client-side with OpenVR.

Pros:

+ Less complex than previous solution
+ Utilizes OpenVR to gain access to HMD's: The type/manufacturer does not matter

Cons:

- Limited usability → Not much more than a POC
- Less sophisticated than previous solution

13 Experiments

14 Results

15 Discussion

16 Conclusion

References

Abrash, M. (2012). *Latency – the sine qua non of ar and vr*. http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/ accessed: 14.05.2020

AllianceForOpenMedia. (n.d.). *Av1 features*. https://aomedia.org/av1-features/ accessed: 12.05.2020

Amazon. (2020). *Aws wavelength - amazon web services*. https://aws.amazon.com/de/wavelength/ accessed: 25.05.2020

Amazon. (n.d.). *Amazon web services (aws) - cloud computing*. https://aws.amazon.com/ accessed: 12.05.2020

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view on cloud computing. *Communications of the ACM*, *53*(4), 50–58. https://doi.org/https://dl.acm.org/doi/fullHtml/10.1145/1721654.1721672

Chen, S.-W. (.-T., Chang, Y.-C., Tseng, P.-H., Huang, C.-Y., & Lei, C.-L. (2019). Cloud gaming latency analysis: Onlive and streammygame delay measurement. https://www.iis.sinica.edu.tw/~swc/onlive/onlive.html

Costello, K., & Goasduff, L. (2019). *Gartner says worldwide iaas public cloud services market grew 31.3 percent in 2018*. https://www.gartner.com/en/newsroom/press-releases/2019-07-29-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31point3-percent-in-2018 accessed: 02.06.2020

Google. (2018). *Seurat: A scene simplification technology designed to process very complex 3d scenes into a representation that renders efficiently on mobile 6dof vr systems.* https://github.com/googlevr/seurat accessed: 14.05.2020

Google. (2019). *Stadia - one place for all the ways we play*. https://stadia.google.com/ accessed: 11.05.2020

Google. (n.d.-a). *Google cloud computing services*. https://cloud.google.com/ accessed: 12.05.2020

Google. (n.d.-b). *Webrtc*. https://webrtc.org/ accessed: 12.05.2020

Hou, X., Lu, Y., & Dey, S. (2017). Wireless ar/vr with edge/cloud computing. *26th International Conference on Computer Communication and Networks (ICCCN)*. https://doi.org/10.1109/ICCCN.2017.8038375

IRH. (n.d.). *Industrial reality hub*. https://industrialrealityhub.com/en accessed: 27.05.2020

Kaknjo, A., Rao, M., Omerdic, E., Newe, T., & Toal, D. (2019). Real-time secure/unsecure video latency measurement/analysis with fpga-based bump-in-the-wire security. *Sensors*, *19*(13), 2984. https://doi.org/10.3390/s19132984

Liu, L., Zhong, R., Zhang, W., Liu, Y., Zhang, J., Zhang, L., & Gruteser, M. (2018). Cutting the cord: Designing a high-quality untethered vrsystem with low latency remote rendering. *Proceedings of MobiSys ACM '18*. https://doi.org/https://doi.org/10.1145/3210240.3210313

Microsoft. (n.d.). *Cloud computing services — microsoft azure*. https://azure.microsoft.com/en-us/ accessed: 12.05.2020

Müller, J., Neff, T., Voglreiter, P., Mlakar, M., Steinberger, M., Schmalstieg, D., & Dokter, M. (2018). Shading atlas streaming. *ACM Transactions on Graphics*, (199). https://doi.org/https://doi.org/10.1145/3272127.3275087

Nvidia. (2020a). *Nvidia cloudxr sdk*. https://developer.nvidia.com/nvidia-cloudxr accessed: 13.05.2020

Nvidia. (2020b). *Your games. your devices. play anywhere*. https://www.nvidia.com/en-us/geforce-now/ accessed: 11.05.2020

Riboulot, A. (2020). *Cloud gaming: How-to*. https://kta.io/posts/cloud_desktop accessed: 18.05.2020

Shadow. (2015). *Transform your device into an gaming pc*. https://shadow.tech/usen accessed: 08.05.2020

Shadow. (2020). *Vr explorer log: Shadow's entrance into vr begins!* https://community.shadow.tech/usen/blog/news/vr-explorer-log accessed: 14.05.2020

Shi, S., Gupta, V., Hwang, M., & Jana, R. (2018). Mobile vr on edge cloud: A latency-driven design. *MMSys '19*. https://doi.org/https://doi.org/10.1145/3304109.3306217

Shi, S., & Hsu, C.-H. (2015). A survey of interactive remote rendering systems. *ACM Computing Surveys*, (47). https://doi.org/https://dl.acm.org/doi/10.1145/2719921

TayoEXE. (2019). *Test: No vr-ready pc required, cloud pc vr streaming via shadow and virtual desktop quest*. https://www.reddit.com/r/OculusQuest/comments/c5xnux/test_no_vrready_pc_required_cloud_pc_vr_streaming/ accessed: 08.05.2020

Valve. (2016). *Open vr sdk*. https://github.com/ValveSoftware/openvr accessed: 05.06.2020

Wikipedia. (2020). *Hidden-surface determination*. https://en.wikipedia.org/wiki/Hidden-surface_determination#Viewing-frustum_culling accessed: 20.05.2020

XBox. (2019). *Project xcloud*. https://www.xbox.com/en-US/xbox-game-streaming/project-xcloud accessed: 11.05.2020

Zerolight. (2020). *World's first boundless xr over 5g retail experience*. https://zerolight.com/news/press-releases/worlds-first-boundless-xr-over-5g-retail-experience accessed: 29.05.2020

17 Appendices

17.1 Cloud Computing SWOT Analysis

17.1.1 Cloud service provider

Table 2: SWOT Cloud Service Providers

Strengths	**Weaknesses**
Scalability: Using a cloud service enables effortless scaling	*Service Outages*: They do not happen often, but when they happen they are out of the customers control
Lower Costs: Paying only for what the customer uses and not having to worry about maintenance drives down costs	*Longer Upload/Download times*: Compared to an in-house server it will take longer to move large files, as the internet speed is the limiting factor
Lower Capital Expense: Since the customer is not the one buying the hardware	
Global Connectivity: Cloud service providers have a global network of servers and thus the customer can offer his clients a fast connection to a local server	
Security: Cloud service providers have invested heavily into security, since their reputation would be at stake if a breach happened. The customer will always have cutting edge security from a technical standpoint.	
Integration: Since the service is offered as a platform, the customer has access to other services within the providers ecosystem. Big providers like AWS or Microsoft Azure offer an ever expanding selection of services apart from pure server hosting	
Opportunities	**Threats**
Technological Advancements: A cloud service provider will always seek to have the best technology to offset themselves from competition, which benefits the customer	*Termination of Service*: It seems very unlikely, but in theory the service provider could go out of business/terminate the service and thus disrupt the business

17.1.2 In-House Server

In-House servers for rendering purposes can be acquired from graphic card manufacturers such as NVIDIA: https://www.nvidia.com/en-us/design-visualization/quadro-servers/rtx/

Table 3: SWOT In-House Server

Strengths	**Weaknesses**
Total Control: If the customer owns and operates the server, they have complete control over it. They can adjust the server to specifically fit their requirements and thus optimizing performance	*Increased complexity/costs*: Operating a server infrastructure requires experts to administrate and maintain
Faster development/response time: An in-house server is local by nature and thus modifying/fixing things on the server is faster compared to external servers	*Higher Capital Expense*: Since the customer has to buy all the necessary hardware for an in-house server, the upfront investment is higher
Well understood It is easier for developers to become familiar with and develop in-depth knowledge in a server infrastructure they can easily interact with	*Physical Requirements*: Building an in-house server infrastructure requires not only sufficient physical space, but also auxiliary systems such as cooling, (emergency-) electricity and cabling
Opportunities	**Threats**
TBD:	*Obsolescence*: As the owner, the customer would be responsible to upgrade the system if they want/need new features. This of course comes with more costs

17.2 Cloud service providers

The cloud service providers are ranked based on annual revenue from 2018 in Millions of US Dollars (Costello & Goasduff, 2019).

Table 4: Cloud Service Providers by Revenue

Rank	**Company**	**Product**	**Revenue 2018**
1	Amazon	AWS	15,495
2	Microsoft	Azure	5,038
3	Alibaba	Alibaba Cloud	2,499

4	Google	Google Cloud Platform	1,314