# 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.

I would like to thank Jeroen van der Wel from Thales for his help throughout the prototyping phase.

I would also like to thank Gregory Jones and NVIDIA for granting me access to their solution and measurement set up, as well as for helping me with setting up the prototype.

Finally I would like to thank Vincent van Wingerden from Azure for helping me with setting up the prototype.

# 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

# 3 List of Figures and Tables

## List of Figures



## List of Tables

# 4 List of Abbreviations

**AES** Advanced Encryption Standard. 13, 16

**CPU** Central Processing Unit. 26

**E2E** End-to-End. 3, 4, 27, 28, 30

**FoV** Field-of-View. 11, 17, 18

**FPS** Frames per second. 26

**GPU** Graphics Processing Unit. 9, 22, 26

**HMD** Head Mounted Display. 10, 11, 12, 13, 15, 17, 23, 26

**IAM** Identification and Access Management. 2, 13, 30

**IRH** Industrial Reality Hub. 1, 6

**ms** Milliseconds. 8, 10, 12, 13, 17, 25, 28, 29, 30

**MTP** Motion-to-Photon. 10, 11, 13, 17, 18, 20, 25, 30

**OS** Operating System. 26

**QoE** Quality of Experience. 1, 13, 17, 20

**RGB** Red Green Blue. 8

**RTT** Round Trip Time. 27, 28, 30

**TLS** Transport Layer Security. 13, 16

**us** Microsecond. 29

**VM** Virtual Machine. 26

**VR** Virtual Reality. 1, 2, 4, 6, 9, 10, 11, 12, 13, 14, 17, 18, 19, 20, 21, 22, 23, 25, 28, 30

# 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. 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 cur-

rent 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, 2020c) 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.

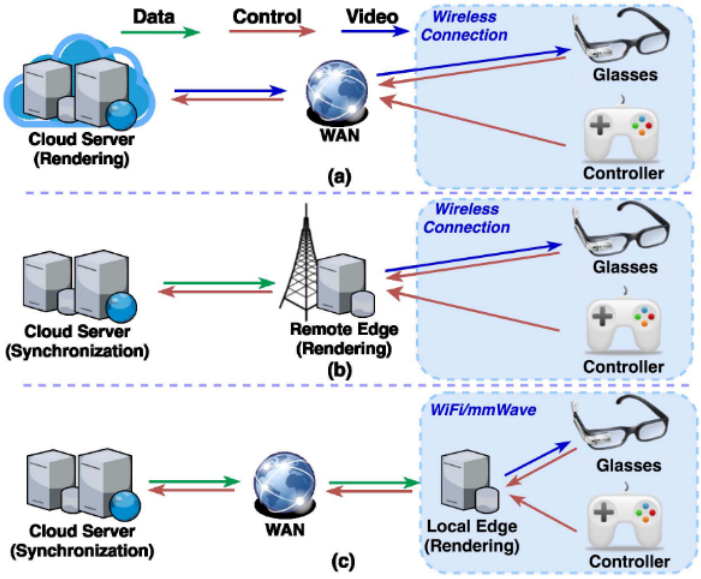
### 8.1.2 Existing Solutions and Technology

Several commercial gaming cloud streaming services already exist, such as Google Stadia (Google, 2019), XBox XCloud (XBox, 2019) and Nvidia GeForceNow (Nvidia, 2020c). 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, 2020b), 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.

### 8.1.3 System Architecture types (for a cloud VR system)

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

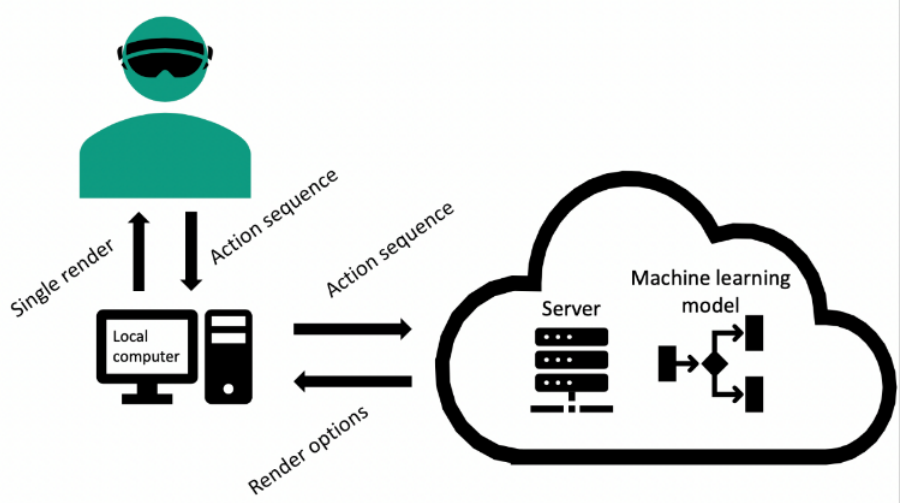



One of the main considerations when designing a cloud VR streaming application is the decision to either use a Cloud, Remote Edge or Local Edge computing device for the rendering of the frames (See Figure 1 and Hou et al., 2017):

- 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.

Figure 2: Example System Architecture




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 disadvantages, 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.

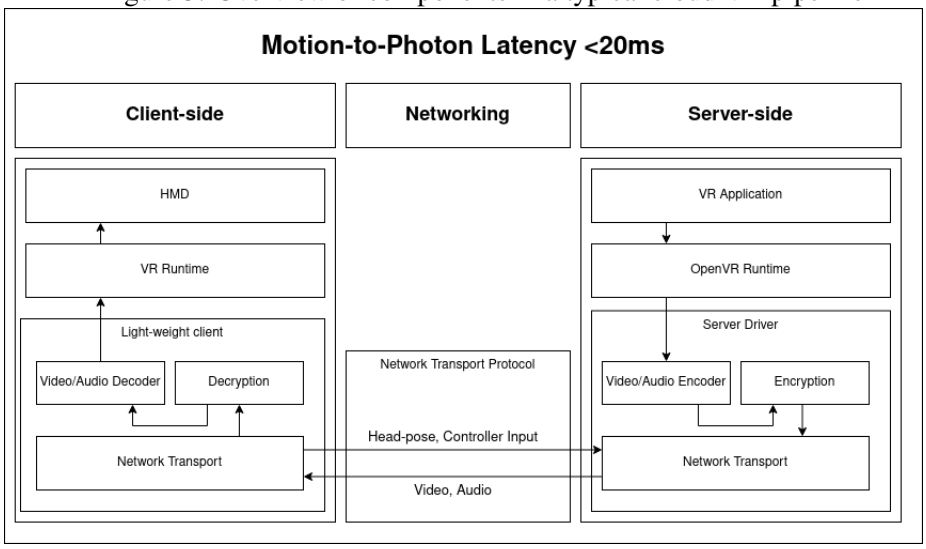
### 8.3.3 Latency Implications

As discovered by existing research, encryption does not greatly influence performance of the video stream in terms of latency (Kaknjo et al., 2019). The research also concludes that the benefits (privacy and data protection) outweigh the costs (latency) of the encryption itself and that other factors (such as hardware used) have a potentially greater influence on the latency.

## 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, 2020b). | 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, 2020c), 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.

# 10 Final Problem Statement

As explored in the Theoretical Framework there are many existing products/technologies for 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. There is already a product on the market that proves the feasibility of creating a product with cloud VR (Zerolight, 2020), but it uses a proprietary solution (Nvidia, 2020b). At the time of this research there are no complete solutions that are open sourced, but the vast majority of technologies used in the proprietary solution are open source themselves. 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.

# 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:** What is the best way to measure MTP latency and how does it compare to a traditional local VR setup?

**Sub Question 2:** Does the technology stack satisfy security requirements?

**Sub Question 3:** To what degree is the user experience compromised/impacted?

**Sub Question 4:** How difficult is it to develop applications for the technology stack?

# 12 Methodology

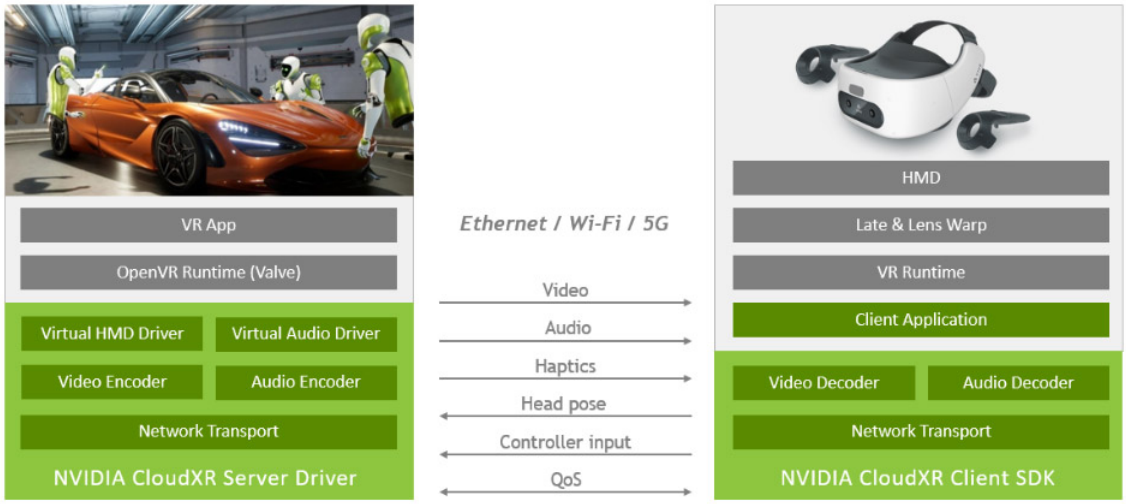
To solve the problem detailed in the Final Problem Statement, first an overview of potential technology stacks will be given. In the following exploratory research phase the first step is to test each available prototype proposal in terms of feasibility. After scoping out each stack briefly, a preliminary conclusion will be drawn to determine which proposal seems the most promising. The winning candidate will be the focus for creating a working prototype for the remainder of the experimental phase.

## 12.1 Proposed Technology Stacks

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

### 12.1.1 NVIDIA CloudXR (+ NVIDIA Quadro on Azure)

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




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:**

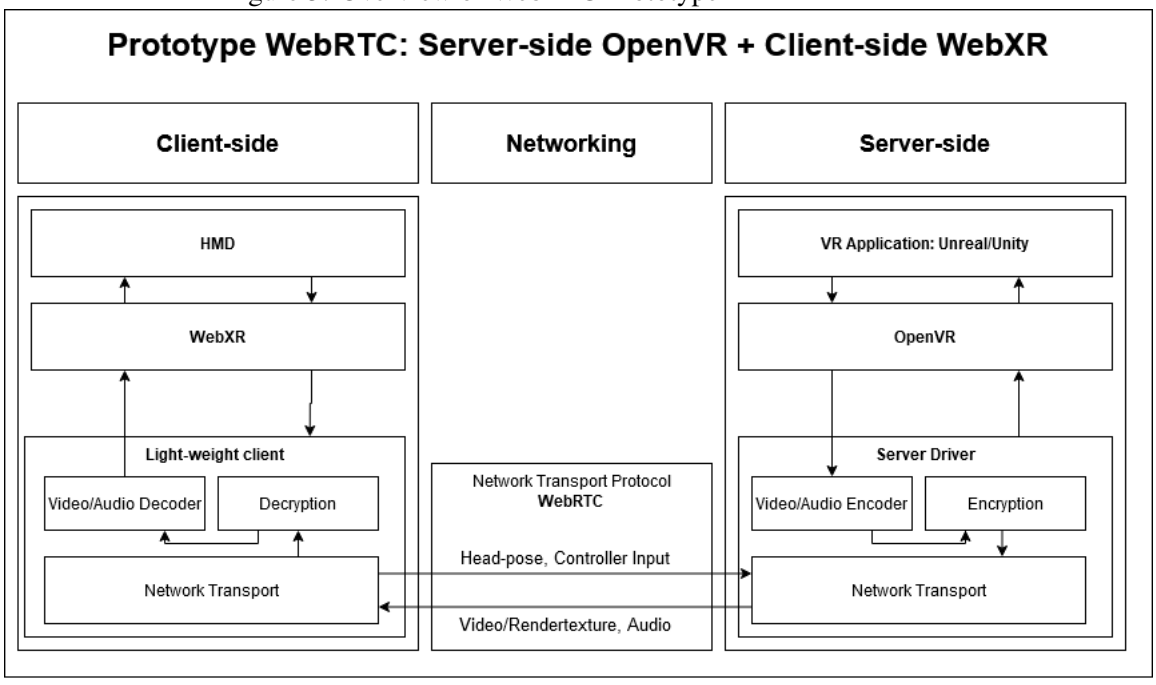
+ Only complete solution
+ Increased prototyping speed
+ Custom made for streaming VR content
+ Works with cutting-edge GPU's

**Cons:**

- Forced to use new generation NVIDIA products (Pascal Architecture)
- 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 and 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

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.

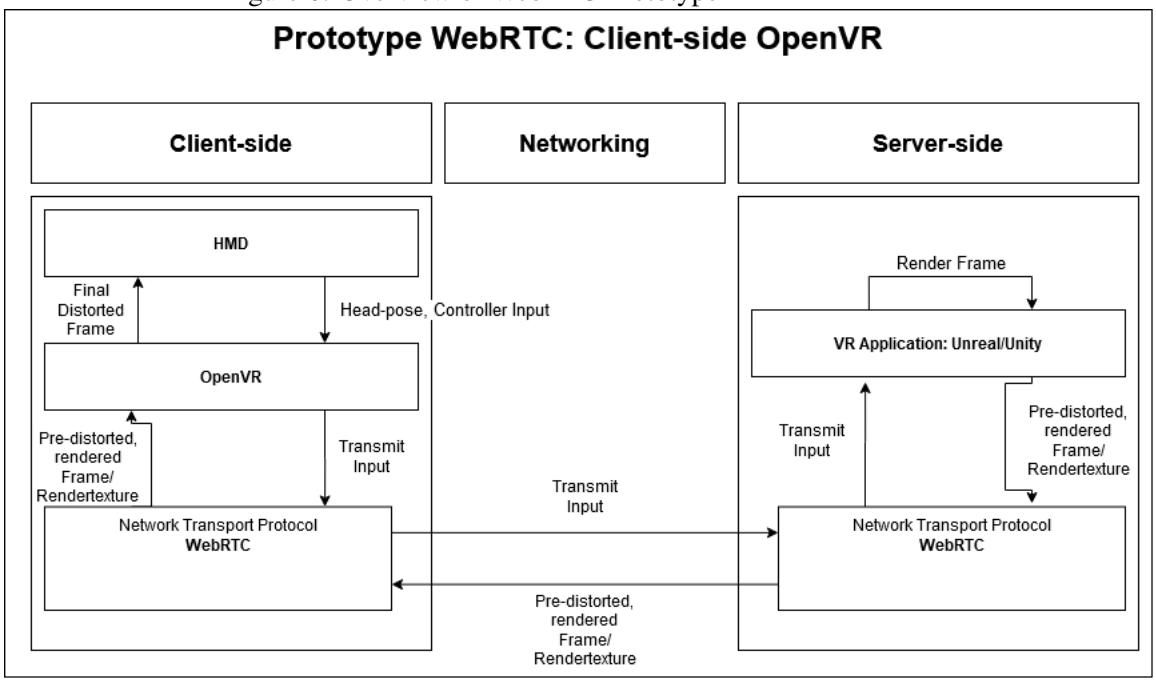
**Pros:**

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

**Cons:**

- Limited usability $\rightarrow$ Not much more than a POC
- Less sophisticated than previous solution

Figure 6: Overview of WebRTC Prototype 2



# 13 Experiments

## 13.1 Exploratory Research Phase

After about a month of researching for existing solutions, applying for NVIDIA CloudXR (Nvidia, 2020b) and testing the difficulty of different prototype architectures it became clear that making a prototype from scratch is not a feasible task for the duration and team size of the project. This conclusion is based on two major observations:

**NVIDIA CloudXR SDK is the only working solution** During the research a couple of existing products were found, that demonstrated the feasibility of key areas of interest for this project: Low Latency and High Security (Würster, 2020). All of the products found were launched shortly before the start or during the graduation, so it is safe to say that the technology and market is currently emerging. Another overlap of these products was that they all used NVIDIA's CloudXR SDK to enable them to stream VR content. In the conference talks, where these products were presented, CloudXR was always mentioned as the last piece of the puzzle to enable the creation of ambitions cloud VR projects.

**Creating a custom Prototype is impossible** When experimenting with the first and second prototype design, both in Unreal Engine 4 and Unity 3D, it was possible to quickly create a WebRTC connection from a 'server' machine to a 'client' machine in the same network. However before I ever got to implementing the whole application loop it was already evident that the performance would be a critical factor. As explored in the Theoretical Framework the Motion-to-Photon (MTP) latency for a single frame is not allowed exceed 20ms, where as the base latency of the connection in the prototypes would be atleast around 100-200ms. This measurement was taken in a local network instead of a connection to a cloud server and without streaming the significantly bigger VR resolution of 2160x1200, all of which things that would have impacted the latency negatively even more. Considering this it would be impossible to create and optimize a prototype to undercut the MTP barrier within the remaining time frame of the project. Additionally there was a serious lack of documentation and examples for SteamVR/OpenVR, the most critical component in the proposed designs. Lastly it also became clear that alternative solutions, like ReactVR (Facebook, 2019) and A-Frame (Mozilla, 2020), are not suitable for the use-cases of the stakeholders as they do not allow for the complexity needed as presented in the Introduction.

## 13.2 Building the prototype

At the end of the exploratory research phase I finally got into contact with an NVIDIA employee who was able to grant me access to the CloudXR SDK. I was able to successfully set up the pipeline in the local network of the XR lab within a couple of days. Some initial user testing from myself and volunteering teachers revealed that the solution was potent, as most testers experienced no or only minor degradations in their user experience

I then worked together with Thales and Microsoft Azure to gain access to a Virtual Machine (VM) with the necessary hardware inside of Azure. After two days, due to a driver and licensing issue, I was able to set up the prototype on the aforementioned VM. However the user experience was horrible, due to poor latency and FPS. The root of this issue was a mismatch of (v)GPU driver versions and Operating System (OS)'s. After installing compatible versions of vGPU drivers and CloudXR, as well as an virtual audio driver, the prototype finally worked as intended.

Table 2: Server and Client Specifications

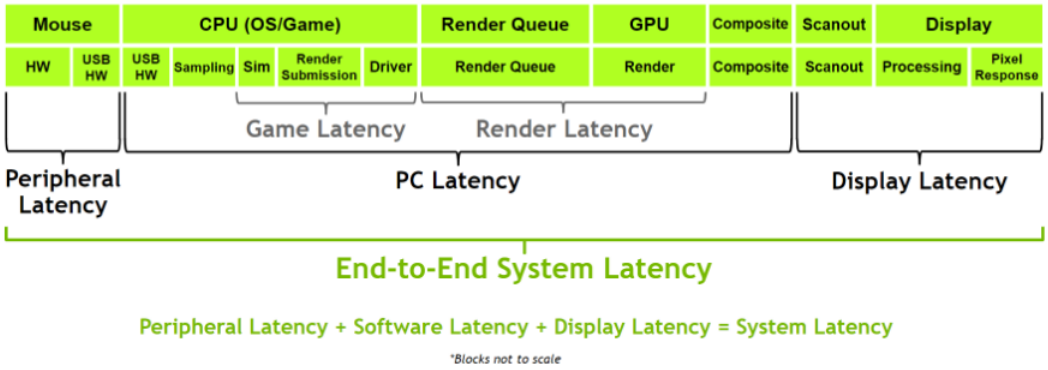
| Server Specifications | | Client Specifications | |
|---|---|---|---|
| **OS** | Windows Server 2019 | **OS** | Windows 10 Pro |
| **GPU** | P100 | **GPU** | GeForce RTX 2080 Ti |
| **Driver** | 451.48 | **Driver** | 432.00 |
| **CPU** | Intel Xeon E5-2690 v9 | **CPU** | Intel I7-7820X |
| **Memory** | 112 GiB | **Memory** | 64 GiB |
| **Location** | Europe West | **HMD** | Vive Pro |
| **Azure VM Series** | NC6s v2 | | |

## 13.3 Testing the prototype

Before starting to build the prototype, I had a meeting with Thales to discuss how to test it once it is running. Together we agreed on conducting 2 types of test with 2 different tools to measure the performance of the prototype.

### 13.3.1 Test 1: End-to-End (E2E) Latency

Figure 7: End-to-End (E2E) Latency Visualized (Nvidia, 2020a)




The first test will be to measure the overall E2E latency of the system.

It is also expected that the results of this measurement are quite high. This is due to the fact that this test's measurements include the hardware and software latency of two computers, as well as the network latency between those.

### 13.3.2 Test 2: Network Latency

The second test will measure the network latency, by testing the Round Trip Time (RTT). This will be done with the open-source tool Wireshark (Combs, 2020). By comparing the resulting measurements with the first test, the impact of network latency on the whole system will be revealed.

# 14 Results

## 14.1 Test 1: End-to-End (E2E) Latency

Table 3: End-to-End (E2E) Latency (ms)

| End-to-End (E2E) Latency (ms) | | |
|---|---|---|
| Local without VR | Local VR | Cloud VR |
| 75-80 | 90-100 | 145-150 |
| Effective performance difference | | |
| - | +25% | +87.5% |

Initial measurements show that the E2E latency of the Cloud VR prototype is about **87.5%** higher than a normal application on a local machine and about **50%** higher than a VR application on a local machine.

## 14.2 Test 2: Network Latency

Testing with Wireshark (Combs, 2020) revealed that the Round Trip Time (RTT) for a single packet is **5-6 ms**. Wireshark measures RTT for a TCP connection as the latency between a data packet and the subsequent acknowledgment packet. This is supported by data collected from the CXR (Nvidia, 2020b) software, which measured similar results.

Comparing this to the recorded E2E latency and 'Frame Time in CXR' latency it is revealed that the network latency has an impact of $\sim$**3.3%** on the E2E latency and an impact of $\sim$**26%** on the latency of a single frame.

### 14.3 Cloud XR Software statistics

Table 4: Cloud XR software statistics

| Frames Per Second | Frame Time in CXR (ms) | Client Queue Time (ms) | Average Video Rate (kbps) |
|---|---|---|---|
| 90 | 21-22 | 5-9 | 20-30 |

| Est. Avail. Bandwidth (kbps) | Bandwidth Utilization (%) | Jitter (us) | Round Trip Delay (ms) |
|---|---|---|---|
| 300-500 | 3-4 | 17 | 5-6 |

The following measurements were recorded from the Cloud XR software during a playtest with an example scene from Thales. The most interesting statistics is 'Frame Time in CXR' which is the time a single frame spends in the Cloud XR software. This measurement is taken from the moment the server receives the eye texture from the client, to the moment the eye texture leaves the Cloud XR library on the client. Additionally the 'Client Queue Time', which is the time decoded frames spend for de-jittering to account for network jitter, and the round trip time are included in this measurement.

# 15 Discussion

The circumstances that led to the inception of this project were that the stakeholders wanted to explore how to stream VR content from the cloud in a fast and secure manner and how such a solution compares against a local VR set up. The problem was that the clients did not know which technology stacks work in practice. The data from the finished prototype suggests that these requirements have been met.

When looking at the results of the End-to-End (E2E) latency tests, there are clear performance differences depending on the complexity of tested application and as suspected higher complexity correlates with higher latency. As observed the latency of the cloud prototype is about 50% higher than a local VR application. This difference in performance was expected and as such is not surprising.

On the other hand a surprising fact was the low network delay and utilization. While setting up the prototype the requirements for bandwidth were 50-60 Mbps, however only $\sim$4% are utilized at runtime. It seems like the excessive bandwidth capacity is primarily used to facilitate the low Round Trip Time (RTT) of $\sim$6ms. The low RTT is especially important for the latency of individual frames, as it makes up $\sim$One-Fourth of the overall individual frame latency.

The individual frame latency itself is $\sim$22ms, which is only marginally higher than the 20ms MTP threshold that got explained in the Theoretical Framework. This reflects the overall performance of the prototype best, as most users experienced a small but constant delay in the prototype when compared to a local VR set up.

The prototype has been deployed in the dutch secure defense cloud powered by Microsoft Azure. As such most security concerns are taken care of by the cloud platform itself. However there is still ample room to make the system even more secure, for example with a Identification and Access Management (IAM) solution.

Lastly it should be noted that these test results do not reflect the maximum performance possible, as due to time constrains the tests were conducted with non-optimized applications and hardware. Future researchers have the opportunity to optimize most parts of the system such as the cloud server and the test application. The expectation is that the rations between the results stay the same, even as the system itself is sped up.

# 16 Conclusion

This project aimed to identify "*Which Technology stack(s) for building a cloud Virtual Reality (VR) streaming application satisfies the clients demands (low latency, security, QoE) best ?* ". Based on the experiments done it can be concluded that NVIDIA's Cloud XR software (Nvidia, 2020b) is the only viable technology stack that can meet those requirements. It possesses only marginally higher individual frame latency than the much discussed 20ms MTP threshold and runs in a secure cloud. The high performance means that user experience is barely impacted. Furthermore the software works with the OpenVR SDK, which is by far the most popular choice for VR applications, so there are no added difficulties to developing applications on this technology stack.

To better understand the limitations of the system, practitioners should consider to test the application with scenes of increasing polygon complexity to determine the maximum amount of polygons a cloud VR scene can have before performance degrades.

# 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

Combs, G. (2020). *Wireshark*. https://wireshark.org accessed: 12.09.2020

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

Facebook. (2019). *Reactvr*. https://facebook.github.io/react-360/ accessed: 06.07.2020

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

Mozilla. (2020). *A-frame*. https://aframe.io/ accessed: 06.07.2020

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). *End-to-end system latency*. NVIDIA%20LDAT%20User%20Guide accessed: 07.10.2020

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

Nvidia. (2020c). *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

Würster, J. (2020). *Gtc 2020: From shop floor to engineering workspaces: Rich immersive experiences across the enterprise, nvidia cloudxr and esi im-*

*mersive cloud tech.* https://developer.nvidia.com/gtc/2020/video/s21432 accessed: 06.07.2020

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 5: 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 6: 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 7: 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 |