Final Implementation Plan

Cloud VR. Secure, Fast and Distributed Virtual Reality Solutions.

Leon Koster

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

Contents

Acronyms

AES Advanced Encryption Standard. 10, 12, 13

FoV Field-of-View. 8

GPU Graphics Processing Unit. 4

HMD Head Mounted Display. 8, 9, 11, 12

IAM Identification and Access Management. 1, 10

IRH Industrial Reality Hub. 3, 4, 17

ms Milliseconds. 9, 16

MTP Motion-to-Photon. 9, 16, 17

QoE Quality of Experience. 9

TLS Transport Layer Security. 10, 13

VR Virtual Reality. 1, 3, 4, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 17

1 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.).

2 Reason for Assignment

2.1 Graduation Assignment

The assignment for this graduation is to analyse the current state of cloud VR streaming. This includes existing commercial services, open source libraries/frameworks, relevant research papers and experimental technologies. The 4 research directions for this project are system architecture for a cloud VR system, Latency, Multi-User Experiences and GPU Scaling. This paper will focus on researching the system architecture and latency improvements. These findings will be presented in the form of a theoretical framework and a literature review.

Armed with the information from the initial part of the research, the student and the stakeholders will agree on a suitable research focus for the reminder of the projec, including a proof of concept focused on reducing latency.

2.2 Client outline

The stakeholders in this project are the student (Leon Koster), Saxion's XR Lab (Matthijs van Veen, Yiwei Jiang) and the Industrial Reality Hub (IRH).

3 Objectives of the client

Together with the companies from the Industrial Reality Hub mentioned in the Client outline, 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.

In the bigger picture this project is a multi phase project, with this project being the initial phase. It aims to give an overview of the current situation regarding cloud VR streaming and focus on system architecture and latency optimization. These informations will subsequently be utilized in the later stages of the over arching research project.

4 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.

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.

5 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. Lastly I will be proposing technology stacks, made up of said components, that can satisfy all necessary aspects of a cloud VR system as described in the overview.

5.1 Cloud Streaming/Cloud Computing

5.1.1 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.

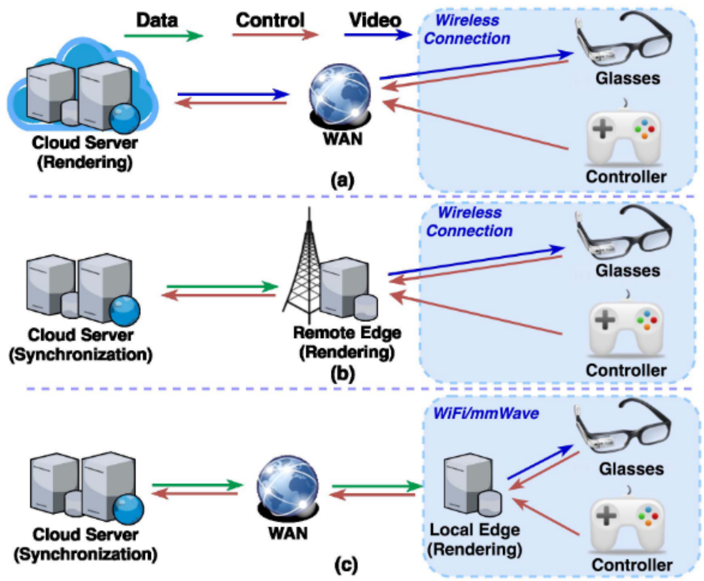
5.1.2 Existing Solutions and Technology

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). Additionally the first commercial retail products with cloud VR have been released (Zerolight, 2020). 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), but companies are actively working on finding a solution (Amazon, 2020). For more information about these applications and technologies, please refer to the **??**

5.1.3 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, Lu, and Dey, 2017):

Figure 1: Cloud Server, Remote Edge, Local Edge visualized (Hou, Lu, & Dey, 2017)

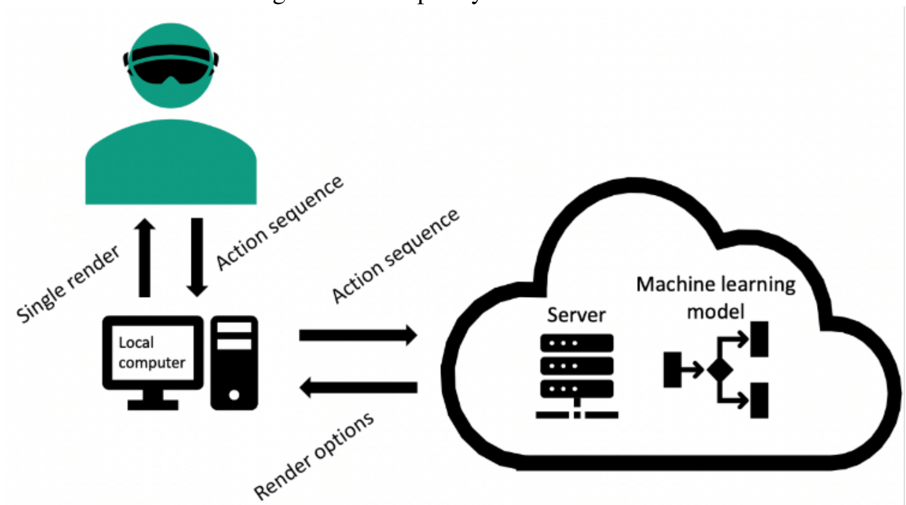


- A cloud server renders the Field-of-View (FoV) (current view) remotely and streams the corresponding video to the user's Head Mounted Display (HMD).
- A Remote Edge sever receives multiple views that are rendered remotely on cloud servers, stitches them together to a 360-degree video, and streams the video to the user's HMD
- 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 disadvantages, which are elaborated upon in the **??**, and the decision should be made based on the unique circumstances of each customer.

Figure 2: Example System Architecture



5.1.4 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, Chang, Tseng, Huang, & Lei, 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

5.2 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). For more information, please refer to the **??**.

5.3 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:

5.3.1 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.

5.3.2 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.

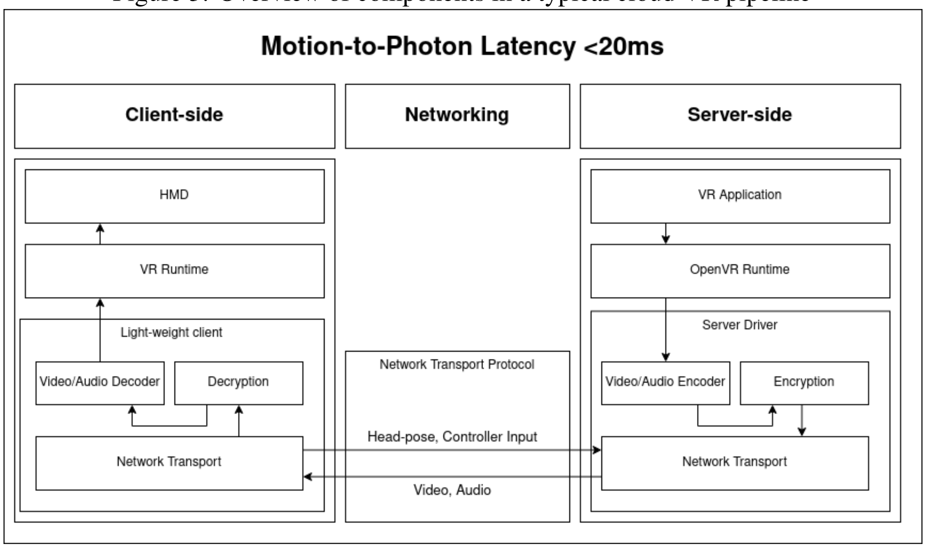
5.3.3 Latency Implications

As discovered by existing research, encryption does not greatly influence performance of the video stream in terms of latency (Kaknjo, Rao, Omerdic, Newe, & Toal, 2019).

5.4 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.

5.4.1 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

5.5 Proposed Technology Stacks

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

5.5.1 NVIDIA CloudXR (+ NVIDIA Quadro on Azure)

The CloudXR SDK from NVIDIA offers a complete solution package 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. Since Azure is the chosen cloud provider of Thales, one of the major stakeholders, the idea would be to deploy this tech stack there.

Pros:

+ Only complete solution
+ Rapid prototyping speed (presumably)
+ Custom made for streaming VR content
+ Created by one of the leading companies in the field of (remote) rendering

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

5.5.2 WebRTC + High Performance Video Codec

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.

Pros:

+ Open source
+ Mature technology
+ Well documented and supported
+ Platform independent
+ Developed by the same companies that offer cloud computing services (improved integration ?)

Cons:

- Generalized solution (it will take more work to optimize it for VR)
- Lower performance Web Technology (but there are native clients for all major platforms available)
- Higher complexity (might run out of time while trying to create the prototype)

6 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. Thus the focus of this report shifts to finding the technology stack that benefits the stakeholders the most.

7 Research Questions

Main Question:

Can a cloud Virtual Reality (VR) streaming application be built from available resources, that satisfies the latency and security requirements?

Sub Question 1: How (in)effective are existing solutions when applied to the cloud streaming VR context?

Sub Question 2: What are the most important considerations when designing an architecture for cloud VR streaming, in order to maximise security and minimize latency?

Sub Question 3: What methods are the most efficient way to reduce Motion-to-Photon (MTP) Latency to $\leq$20ms?

8 Approach

To answer the research questions, I will create a prototype cloud VR system. I aim to include as many pre-existing solutions as possible in order to accelerate the prototyping speed. Initially I want to design and implement a Minimal Viable Product with components I understand. If time allows I plan to create more prototypes with different technologies, in order to be able to compare the effectiveness of different solutions when applied to the research problem.

As I outlined earlier, the main focus for this report is on latency reduction and security. To compare solutions and understand their effectiveness I will measure Motion-to-Photon (MTP) latency as the primary benchmark. It is harder to measure security, but by understanding what the state-of-the-art security practices are and implementing them to the best of my ability, I aim to create a solution that keeps the clients data safe.

To obtain the needed knowledge to execute upon the presented approach, I plan to refer to Research Papers and Experts. There is a multitude of research papers for individual parts of the prototype (Networking, Latency Reduction, Encryption, Encoding, etc.) that contain concentrated knowledge. To assert which information are useful when applied to the research problem and to discover potential pitfalls, I plan on talking to experts that have real world expertise in developing similar products. I want to source experts from inside the Industrial Reality Hub (IRH) and outside of it in order to maximise my chances of picking the right information/technology.

9 Scope

In this project/report I will outline the available technology that is useful for a cloud VR system. I will furthermore work to create a Prototype solution that demonstrates the viability of cloud VR, with a focus on latency reduction and security.

I will not create a production ready product and I will not be responsible for setting up a networking system.

10 Graduation Schedule

I was asked to give an overview of the time line for this project. These times are from the graduation manual, translated to the time frame of my graduation project.

Start of Project The project started on the 4th May 2020.

Preliminary Implementation Plan To be handed in at the end of week 2: 17th May 2020 23:59

Final Implementation Plan To be handed in at the end of week 5: 7th June 2020 23:59

Graduation Report Draft To be handed in at the end of week 10: October 2020 23:59

Final Graduation Report To be handed in at the end of week 16: November 2020 23:59

References

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

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

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., … Zaharia, M. (2010). A view on cloud computing. *Communications of the ACM*, *53*(4), 50–58. doi: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. Retrieved from https://www.iis.sinica.edu.tw/~swc/onlive/onlive.html

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. Retrieved from https://github.com/googlevr/seurat. (accessed: 14.05.2020)

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

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

Google. (n.d.-b). Webrtc. Retrieved from 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)*. doi:10.1109/ICCCN.2017.8038375

IRH. (n.d.). Industrial reality hub. Retrieved from 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. doi:10.3390/s19132984

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

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

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

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

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

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

XBox. (2019). Project xcloud. Retrieved from 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. Retrieved from https://zerolight.com/news/press-releases/worlds-first-boundless-xr-over-5g-retail-experience. (accessed: 29.05.2020)