Editing in the Cloud

Karsten Schragmann
Product Manager, Arvato Systems

Abstract

Changing working styles and the globalization of the media industry see a growing need for "remote" working and collaboration; an area where "the cloud" offers promising opportunities.

However, cloud-based editing presents a number of challenges to overcome while still maintaining an on-premises feel and presenting the full client experience to editors. In this paper, we will look at how streaming technology and formats can be implemented together with an Adobe Premiere Pro client to enable users to work anywhere without detriment to the user experience.

Based on tests carried out together with end users, this paper will consider some of the technical innovations required to maintain that user experience. It will examine the use of streaming servers and SMPTE RDD25-based HD Proxies as a general approach to optimize the 'work anywhere' editing possibilities for use cases addressing producers, journalists and editors working remotely.

vidispine
An Arvato Systems Brand

Introduction

Remote editing is complex and depends heavily on internet bandwidth. Editing projects consist of hundreds of source files, which can prevent fast delivery because content needs to be distributed across different sites.

Traditionally, there have been two approaches.

First, "proxy" or "low-res" editing: Editors use specialized editing clients, that utilize lower resolution, and therefore bandwidth. After the edit, projects are either sent to "craft" edit clients, such as Adobe Premiere Pro, where they link back to the high resolution material, or a new clip is created, usually by a server-side render engine based on the high res material. This approach has many merits, especially when used for journalistic or highlight editing which require only simple edits and/or voice over. But it is less suitable for other workflows as it offers fairly limited editing functions. Editing clients built for proxy editing also do not offer the same look, feel and functions of craft editing clients.

A second approach, especially as high-bandwidth connections have become more widely available, has been to connect directly to the high res storage with a craft edit client over a Virtual Private Network (VPN). However even with high speed broadband connections, this approach does not give users a local-like experience since the used protocols like Simple Management Protocol (SMB), Apple Filing Protocol (AFP) and Network File System (NFS) are not really designed to operate via high-ping networks. This results in a high potential for packet loss, which dramatically degrades their performance[1]. Even audio elements take valuable time for the client to analyze and generate wave forms, often resulting in users disabling useful functionality just to make editing practical.

The following document examines an approach called Cloud-based editing. Cloud technology can work as foundation for collaborative and remote editing. But can cloud technology itself create an editing experience, that comes close to established craft editing based on local content?

This paper will discuss a cloud or hybrid solution that is based on the use of Adobe Premiere Pro combined with an optimized streaming server and SMPTE RDD254-derived HD Proxies. HD Proxies, in this case are high definition H.264 compressed proxies, tailored for limited loss of quality. It considers the technical innovations required to make cloud-based editing possible.

High data rates

HD and UHD video formats results in very high data rates, which need to be transferred via the network. Common formats range between 50 and 800 Mbit/s, already creating challenges for on-premises installations that typically consist of a 10Gbit network infrastructure and storage tailored specifically to video production.

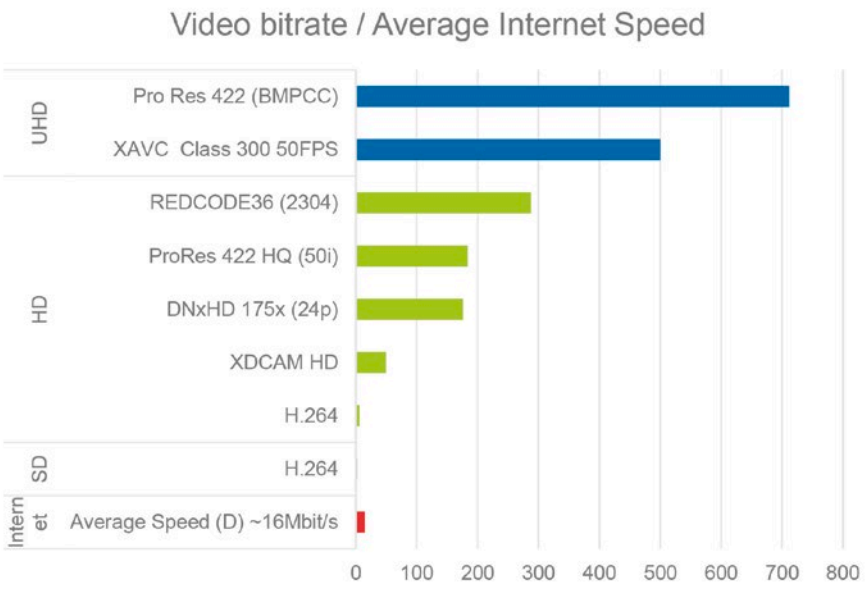

Figure 1: Video bitrate vs Average Internet Speed

This comparison contrasts the average speed of the internet in Germany[2] to different production formats. As long the average internet speed will not increase dramatically, an editing workflow based on a format with less bitrate is required. This necessities the use of a proxy file (proxy video as low bandwidth representation of a source clip).

Latency and perceived response
The bigger the distance between client

Cloud 'Craft' Editing: Challenges

There are a number of challenges that need to be overcome in order to create a similar editing experience for editors

[1] Microsoft Technet
[2] https://de.statista.com

and server, the larger the latency of the data that needs to transferred. This influences the perceived response of the editing client. A latency that is longer than the time of displayed frame, results in a bad perception by the user. Depending on the framerate the maximum acceptable latency can be as low as 15 – 20 milliseconds.

Tools

Depending on the size and structure of a production company, the number of different editors working on different platforms or operating systems varies. Moreover, variations in hardware and software tend to grow. Thus a solution is required, that is able to flexibly deal with the different variants. This could be achieved by using established and powerful craft editing software.

Security

How can media be protected from unwanted access? In an on-premises production environment this can be achieved by blocking external access to the network. However, it is the nature of the cloud to create a maximum availability, so a cloud editing solution requires protection via encryption or other security measures.

Costs

The costs of the technology that is necessary for an on-premises solution needs to be compared to a remote cloud solution. This results in a complex comparison, influenced by a variety of different cost drivers such as scaling, utilization and requirements. There is a choice between investment in hardware and continuous maintenance and support (CAPEX), or subscription to a SaaS cloud model with “pay as you use” subscription (OPEX). This paper does not address the general on-prem vs cloud comparison, but explains a technical streaming based solution, with a video bandwidth which is used by other distribution platforms like Netflix, which stream 125 million hours of content per day[3].

Methods for remote access

The following chapter compares three different approaches to achieve remote editing. They are based on different technologies that currently exist as remote editing solutions on the market.

Remote Display Control

A desktop environment can be hosted on a central system, including all available applications. This environment can be run remotely on another client. The client doesn’t necessarily require the full performance to run the hosted applications, but only needs to be able to display the content delivered. Historically the main use of remote desktop software was remote administration, but with the advent of cloud computing the functionality became more relevant to provide sophisticated applications such as graphical or video editing. The editor accesses the system via a thin client based on protocols including Remote Desktop Protocol (RDP) and Virtual Network Computing (VNC), or proprietary protocols like PCoverIP. Proprietary protocols provide optimized delivery of content, like graphics or video and include encryption.

An editing environment could be hosted on-premises at a production house, including servers, storages and editing client ,or it can be completely hosted in the cloud. The range of editing possibilities can range from simple editing to large enterprise-scale solutions.

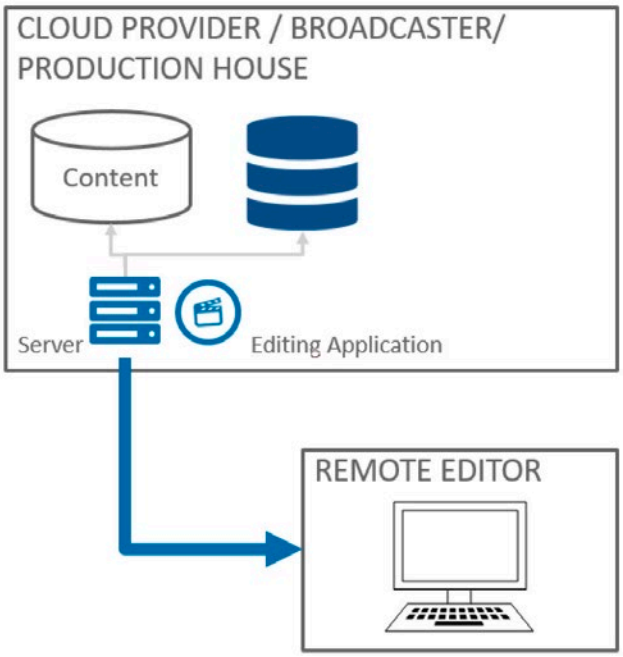

Figure 2: Remote Display Control

[3] https://medium.com/netflix-techblog/protecting-netflix-viewing-privacy-at-scale-39c675d88f45

Advantages

- The full feature set of an editing solution can be made accessible for remote users
- Proprietary solutions are optimized for transferring video content smoothly and provide encryption
- The editing solution can range from small simple systems to enterprise scale
- Proprietary solutions allow displaying the application within a web browser.

Disadvantages

- RDP and VNC are protocols designed for remote system access and administration. They additionally compress the streamed content and do not provide higher frames per second. Thus proprietary protocols are required in order to meet the requirements of perceived response
- Latency is vital. Distance to central system, hosted on-premises, can easily be too great
- Each editing instance requires a couple of centrally hosted server and connected client. The connection is 1:1. Scaling up the number of available editing clients requires the start of additional server instances
- Scaling up additional instances, might be limited to the licensing concept of application vendor
- Ingesting files cannot be performed via the editing client directly. The displayed application is passive. Other ways of adding material to collaborative editing are required
- Limited depiction of content due to compression and limitation of framerate
- Solutions require high internet bandwidth to display video content smoothly.

Remote Editing – Virtual Private Network (VPN)

This remote editing approach utilizes functionalities that are already proven for editing solutions. The editing application runs on a dedicated editing client computer and the video content is centrally hosted on storages which are based on common file systems and servers. The client is not directly integrated to the same Local Area Network (LAN) as the content. Instead, it is connected via a Wide Area Network (WAN) connection, utilizing a Virtual Private Network (VPN). Files are accessed via common protocols like Simple Management Protocol (SMB), Apple Filing Protocol (AFP) and Network File System (NFS). The editing solution can be used with hi-res video files or can work on proxy files, depending on the featureset of the application and configuration.

Advantages

- Allows simple scaling of editing clients, as long as the bandwidth of the VPN is sufficient
- An editing application running on the client provides the full experience of the application, including controls, responsiveness, file upload etc
- Established collaboration workflows can be utilized
- The content is protected via the encryption of the VPN tunnel.

Disadvantages

- Possible bandwidth limitations due to VPN. Since the connection is routed at least over one VPN Server, the latency will be high
- If proxy is used, the compression and codec influences the quality of the content. Some features like effects or focus approval might not be available
- File access protocols are not designed for networks with high latency. SMB, for example, is block-based and each time a block is transferred, there is

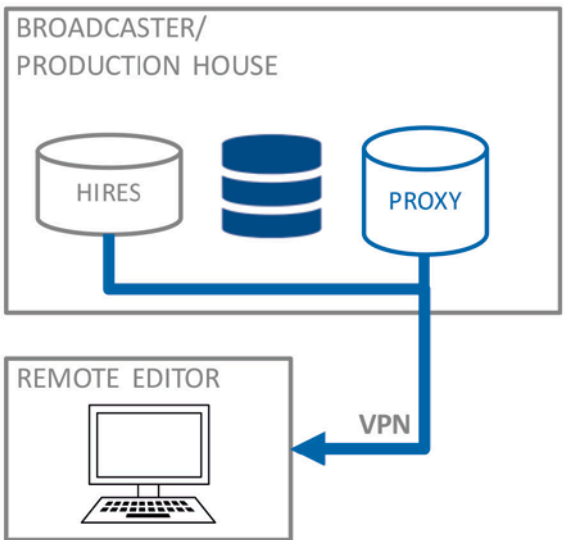

Figure 3: Virtual Private Network

communication to the server. This results in a total latency that is at least equal to the latency between the client and the server[4].

[4] https://blogs.technet.microsoft.com

> **The use of proxy files reduces the required bandwidth, thus the costs of cloud file transfers are reduced.**

Cloud Editing – Proxy Streaming

This approach is cloud-based and is running the editing application on the client. The look and feel of the application is the same as with an existing, proven editing application. The video content is stored on a public or private cloud storage, and, instead of access via direct files the proxy video is delivered to the client via a streaming server. Protocols used for Streaming are Hypertext Transfer Protocol (HTTP) or Transmission Control Protocol (TCP). A streaming server allows — depending on the codec and bandwidth used — the delivery of dozens or hundreds of parallel streams. The streaming server needs to be known to the editing client in order to request a dedicated video and the client needs to be aware of a reference that can be called on the video. Thus a management layer like a Media Asset Management is required.

Advantages

- The scaling of additional editing clients is done mainly on client level. Additional Streaming Server instances can be implemented using cloud auto scaling functionalities
- With the editing application running on the local client, the full functionality and experience of the client can be used
- The local client enables local ingest via editing client functionalities
- The use of proxy files reduces the required bandwidth, thus the costs of cloud file transfers are reduced
- Seamless transfer of clients working remotely and "in house". The client installation keeps its settings and additional plugins and can be used in both environments.

Disadvantages

- Due to the usage of proxy the quality of the content is not on-par to the original hi-res video
- Bandwidth of the connection can be limited depending on the used access (LTE, Fiber etc.)
- An additional management layer is required to orchestrate the streaming server (though this can provide additional functionalities enhancing the collaborative editing functionalities).

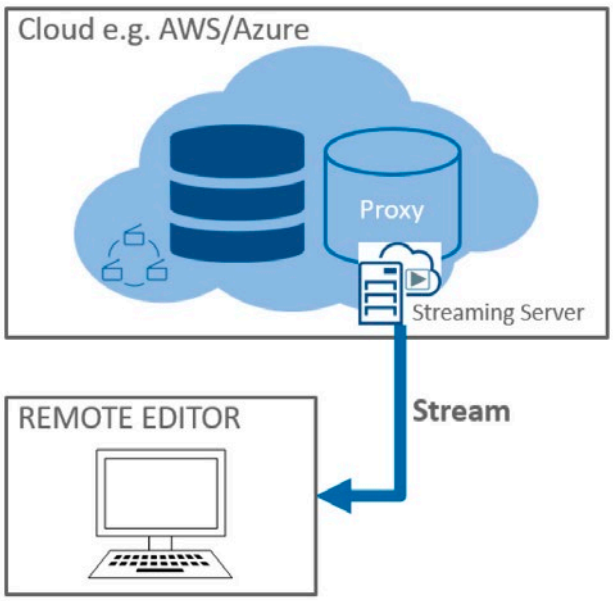

Figure 4: Cloud Editing

Cloud Editing with live encoding

An existing solution enables editing clients to handle video from a streaming source. It is based on a Streaming Engine which transcodes the source video based on the available bandwidth on the fly to the client. This means that each stream requires a dedicated compute resource in order to deliver video to the client.

- High costs driven by hardware requirements for live transcoding
- Additionally the streaming servers continuously require direct access to the source files, which limits the flexibility of the outlet of the video stream
- In most cases the streaming servers would be hosted on premise along with the production system. This limits the suitable location of reachable editing clients, because with growing distance the latency of the network prevents smooth operations.

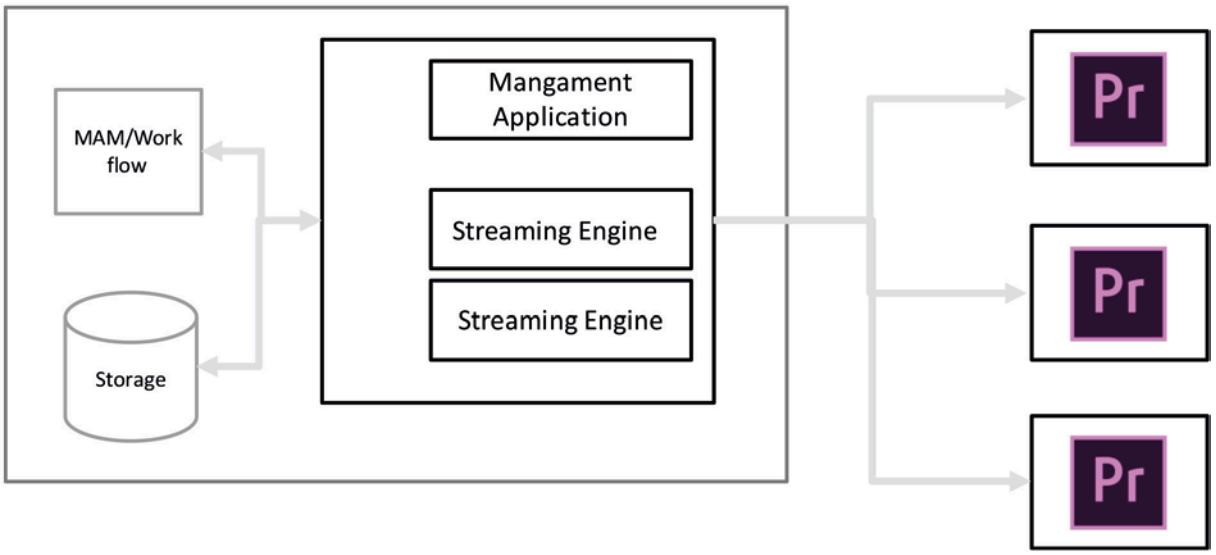

Figure 5: Cloud Editing with live encoding

Advantages

- Streaming provides the best possible quality depending on the available bandwidth
- With the editing application running on the local client, the full functionality and experience of the client can be used
- The local client enables local ingest via editing client functionalities.

Disadvantages

- A relatively high numbers of streaming servers is required in order to serve a typical number of editing clients

Factors for success

Based on the previous examples and requirements, we can list the factors necessary for a successful solution.

Bandwidth: Since the bandwidth is limited, the production formats cannot be used directly for editing. Thus a proxy format seems to be the appropriate approach. Hence the quality needs to be as close to the original file as possible.

Availability: A cloud solution provides the possibility to scale streaming servers based on the number of connected clients. Moreover a public cloud solution hosted on the major platforms like Microsoft Azure or Amazon Web Services allows to create streaming instances close to the clients location, resulting in lower latency.

Tooling: Integrating a streaming solution into a commonly used editing application, like Adobe Premiere Pro, allows editors to work with well-known software and does not require to adjust to new clients or workflows.

Security: Security is always an important aspect when dealing with IT systems. But when it comes to the cloud, the setup gets more vulnerable, since servers and storages are technically reachable via public internet. Even though this is a large field with several aspects, big cloud providers offer solutions to protect

data and put a lot of effort into securing connections. By providing detailed guidelines and restrictions[5] on how to secure systems, which are hosted in the cloud, the providers guide the process to create security, yet still ensure the businesses are responsible for applying necessary measures.

Additionally, there is a requirement to secure the transported data to provide privacy and data integrity. Encryption algorithms which are used in the Transport Layer Security (TLS) formerly known as Secure Sockets Layer (SSL), prevent third parties from reading and modifying any information transferred. Encryption needs to be applied to secure sensitive information or content, that is regulated by copyright[6].

Cloud Editing based on a Media or Production Asset Management System (MAM/PAM)

Another approach to realize remote editing is based on a similar technology as the described cloud editing with live encoding. It uses streaming servers and compressed video for playback in the editing client, but on top of that resources of a Media Asset Management System are utilized. A Media Asset Management System can provide components which can build a base for remote editing in the cloud. Associated with that are pre-generated proxy video, metadata enrichment and management of editing projects.

Technical Overview

An example of remote editing in the cloud, that is built on top of a MAM is illustrated below. This shows an overview of a hybrid craft editing installation, extended with remote editing.

The main site contains a "classic" setup of on-premise based craft editing. Media assets are centrally stored on hi-res storage. These files are accessed by local Adobe Premiere installations. The craft editor imports the video assets to its bin, edits a sequence, and renders it via the local Adobe Media Encoder, in order to create a new asset. All the benefits and limitations of an on-premises solution remain.

On top of that, there is a cloud solution extending the range of editing functionalities to remote locations. A project and media management solution is hosted in the cloud, which enables the local and remote editor to search and browse for centrally managed assets. The managed assets can be stored in the cloud or in the on-premises storage. The proxy, which has been created off the hi-res source files is also located in the cloud, which also applies to the streaming server. The streaming server accesses the proxy and streams it to the connected remote clients.

Whereas the architecture might vary depending on the system scaling, a solution hosting hi-res and Renderer in a cloud environment would be viable.

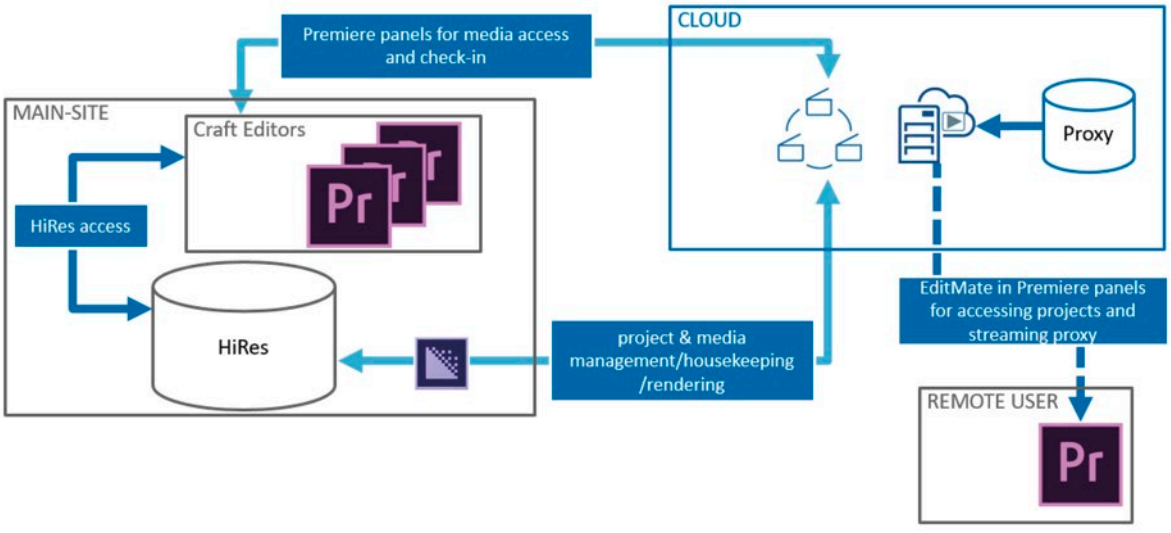

Figure 6: Cloud Editing and Hybrid Installation

[5] AWS_Security_Best_Practices
[6] https://medium.com/netflix-techblog/protecting-netflix-viewing-privacy-at-scale-39c675d88f45s

Figure 7: H.264 10MBit/s

Figure 8: XDCamHD 50MBit/s

Proxy Format

The selection of the proxy format used has significant influence on the perception of the edit experience since resolution and compression influence the availability to determine and evaluate quality or sharpness. Improved quality results necessitate higher bitrate requirements and processing power or processing time to create the proxy file.

A proven proxy format standard is SMPTE RDD25. It is an AVC "Long GOP" proxy with AAC audio, originally conceived to standardize low resolution proxies for use with low-res editors. With SD resolution (640x360) and a bitrate up to 2MBits/s and 4 Stereo Audio Channels it allows fast en- and decoding, but lacks the ability to provide a hi-resolution experience to a user. Thus, the existing format has been extended by resolution, bandwidth and audio channels. Based on the hi-res source and using the Main instead of the Base profile, a proxy can be created that is still possible to encode faster than real-time (depending on the source file up to 70fps). The H.264 with 6-10Mbit/s and 1920x1080 and 8 Stereo Audio tracks is getting close to the source video, but meets within its parameters the requirements of limited bandwidth. The proxy can be created in a mp4 container, which extends the interoperability, or in a Material Exchange Format (MXF) container which allows editing while the proxy is generated. The following illustration shows a comparison of details between an original XDCamHD hi-res and the H.264 10Mbit/s Proxy. Fine details of the structure on the roof and windows show compression artifacts, but the overall perceived quality of the proxy comes close to the original – See 'Analysis of Look and Feel'.

Extracted from source:

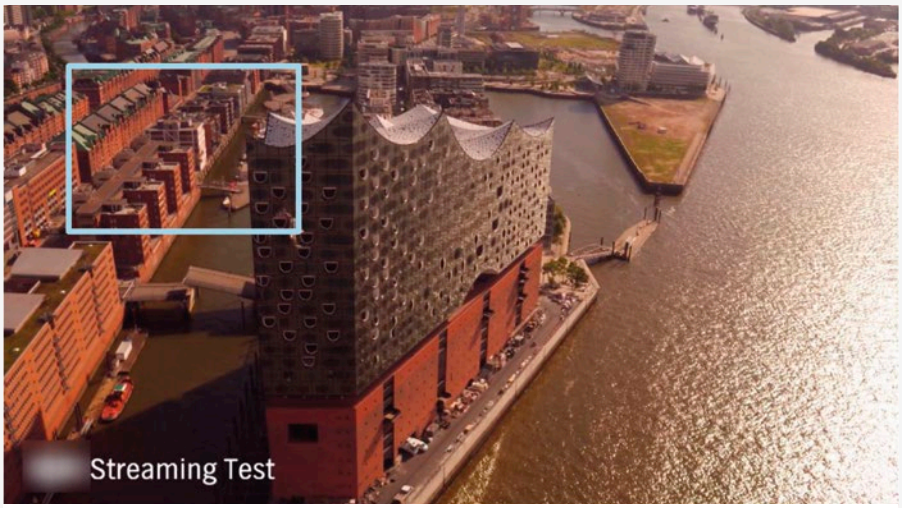

Figure 9: Source Video

With evolving compression technology, new codecs and container formats will extend the possibilities of proxy editing by allowing better quality with similar a framerate or vice versa. High Efficiency Video Coding (HEVC) also known as H.265 is a compression standard which offers about two times data compression ratio in comparison to AVC like H.264[7]. Yet it is not as widely distributed. However, it is becoming more and more popular as former limitations such as the high demand of processing power become less critical. Newer generations of CPUs include dedicated HEVC-decoding and allow more efficient playback. Hence it is a good candidate to replace H.264 for this use case. However, it still requires a certain amount of licensing costs in order to use it. Open and royalty-free video codecs such as AO- Media Video 1 (AV 1) and VP9 offer an alternative. With VP9, Google created a competing codec which is becoming more widespread, not only used on the world largest video distribution platform Youtube. With AV 1 and the recent start of its beta test on Youtube[8] the battle between new codecs and technologies has progressed into the next round[9].

Peak files

A peak file is a wave form representing audio level in a graph based on the time axis. Adobe Premiere creates this wave form on every object within the editing sequence. It analyzes the file and renders the view. Depending on the length of the imported content ,the process takes from seconds up to several minutes. During the rendering process, the editing client is slowed down and the editing usability is heavily compromised. For an editor, this is time lost. A MAM as basis for the proxy editing can create these peak files already during ingest. Adobe Premiere can import these pre-rendered wave forms in turn, which reduces the analysis of files on each client. Pre-rendered peak files, which are available immediately after import, speed up the workflow.

Latency and perceived response

The streaming protocol is a fundamental element for cloud editing; the subjectively perceived experience of editing stands and falls with performance of playback and responsiveness which is mainly driven by the performance of the streaming. A frequently used streaming protocol is MPEG DASH, which is used by the big online streaming platforms Youtube and Netflix in a HTML5 context. The requirement of these platforms is mainly to provide smooth linear forward playback. Although seeking is possible, scrubbing suffers from large segments that need to be transferred from server to client. In order to simulate a fast scrubbing functionality, both Youtube and Netflix use a “trick mode” that is based on thumbnails, where the user navigates via thumbnails to the designated position. A protocol is needed that meets the required perception and feel of an editing client directly accessing the high-resolution files.

Adobe Premiere allows the integration of custom made proprietary importer plugins which handle the playback of the content. The content can be stored locally on a network storage or provided via a streaming server. A proprietary protocol has been introduced by Arvato, that utilizes a TCP connection. (This paper describes the overall functionality and improvements, but will not explain the exact details of the implementation due to the need to protect intellectual property.)

The implemented Premiere Importer functionality has been adjusted in order to transport only the exact required individual frames as they are requested by the client application. This allows fast scrubbing as well as fast forward and playback. A video may be divided into many files (called chunks or segments), each containing only a few seconds of video at one extreme, or stored in a single unchunked file at the other[10].

With larger chunks it might happen that a single frame is requested, but two complete chunks are transported and decoded, because the frame is within a Group Of Pictures (GOP) which is separated over two chunks.

Producers — especially in sports and news use cases — frequently need to scrub through large amounts of video to find the elements they need for their project. When only those frames are downloaded which are needed for decoding, it will reduce latency in streaming, leading to a better user experience. Improvements in latency optimize the workflow for editors and producers. Subjective tests with operators and editors show, that a latency of >30ms results in a poor perceived experience of editing. So different measures need to be applied to keep the latency low.

An improvement is to support asynchronous send and receive of packages, as asynchronous frame requests improve response times. In an asynchronous frame request scenario, the client can send multiple requests at the same time while in parallel receiving all return information. The latency depends on the quality of network and especially on the distance between streaming server and client. Therefore, the outlet of a cloud environment needs to be as close to the client as possible. The large hyper scalers, AWS and MS Azure, with their distribution of data centers across the globe provide scenarios where this requirement can be met.

Initially, streaming servers have mainly served the purpose of channeling the access to the source files by providing video upon requests and preventing direct access to it. This avoids inadvertently move, copy, deletion or other unwanted action. With introduction of low band-width and long network latency due to Cloud, WLAN or VPN, an additional functionality has became more relevant: smart random access to the file.

[7] https://www.bbc.co.uk/rd/blog/2016-01-h-dot-265-slash-hevc-vs-h-dot-264-slash-avc-50-percent-bit- rate-savings-verified
[8] https://www.techpowerup.com/247584/youtube-begins-beta-testing-av1-codec-on-beta-web-browsers
[9] https://www.ibc.org/delivery/codec-wars-the-battle-between-hevc-and-av1/2710.article
[10] https://cs.uwaterloo.ca/~bernard/nossd47.pdf

MXF as a quasi standard container for video in broadcast context enables editing on growing files. This comes along with a random index pack and an index table, which needs to be opened and searched for. This results in several small read operations. SMB (Simple Management Protocol) via an on-prem network connection is a suitable protocol for dealing with this. But SMB is not really designed to operate via high-ping networks, resulting in high potential for packet loss, which results in degraded performance.

Common storages types that are used by AWS and Microsoft for media files are object storages like S3 and Blob. Object storages manage data as objects, opposed to other storage like file systems (manage data as file hierarchy) and block storages (manage data as blocks within sectors and tracks). Object storage provides some advantages compared to file storage, like better performance on big content and throughput. Data can be stored across multiple regions, scaling infinitely to petabytes and beyond. The objects can be enriched with metadata[11]. But when it comes to cloud editing, some obstacles need to be overcome.

If an editor imports a MXF file directly into Premiere from Blob storage, the performance of the importer would be suboptimal, because the MXF parser needs fast random access to be able to work smoothly. However, this is possible with Object Storages – even though Azure Blob Storage offers options for fast random access, this cannot be used for growing file support. The applied streaming technology solves this requirement to the storage. The client does not need fast random access to the server; frames are only transferred to the exact byte. The requirement for random access is server side in this case. But how can we connect cloud storage to a streaming server with sufficient random access performance? At the time being, this is done by locating the streaming server and storage in the same availability zone.

Analysis of Look and Feel

Evaluation of Performance

We have examined the solution as described above in action at a German public broadcaster, comparing different client computers and format scenarios. Experienced editors and engineers performed a subjective rating of the look and feel of the editing performance. The proxy files, streaming server and client were located on-premises. The editing clients were connected with a 1 Gbit/s network connection to the streaming servers.

The goal of this analysis was to figure out the possibility of combining four different editing sites into a single installation. The editing sites are allocated at different sites, hundreds of kilometers apart.

The existing hi-res craft editing installation is based on Microsoft

	Performance Premiere Importer											
	Format / Codec	1x PLAY	2x PLAY	4x PLAY	8x PLAY	16x PLAY	32x PLAY	rewind 1x Play	rewind 4x Play	rewind 8x Play	rewind 16x Play	Scrubbing in s	(x) Picture in Picture
Mac HiRes Hamburg	XDCAM HD 422 Macbook intern HDD	smooth	smooth	smooth	smooth	smooth	smooth	mostly smooth					16x smooth
	XDCAM HD 422 MediaGrid	smooth	smooth	smooth	smooth	smooth	some drops						10x smooth
Standard-Client HiRes Hamburg	XDCAM HD 422 NDR Std Client (USB)	smooth	smooth	drops occasionally	some drops	some drops	some drops	smooth	smooth	smooth	smooth	0	3x drops frequent
	XDCAM HD 422 NDR Std Client (SSD)	smooth	smooth	drops occasionally	drops occasionally	smooth	smooth						4x drops frequent
Workstation Z4 HiRes	XDCAM HD 422 Z4 intern SSD	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	0	8x smooth
	XDCAM HD 422 Z4 MediaGrid	smooth	smooth	smooth	smooth	drops	drops	smooth	smooth	drops	drops	1	8x smooth
Mac HiRes MediaGrid Hannover	XDCAM HD 422 -> without SSD Cache	smooth	smooth	fast stills	some drops	fast stills	fast stills	smooth	some drops	some drops		2	
	XDCAM HD 422 -> SSD	smooth	smooth	smooth	smooth (progr. Look)	smooth	smooth	smooth	smooth	smooth		0	
Workstation HiRes MediaGrid Hannover	XDCAM HD 422 -> without SSD Cache	mostly smooth	mostly smooth	drops	drops	fast stills	fast stills	smooth	drops	drops	drops	2	7x smooth
	XDCAM HD 422 -> SSD Cace	smooth	smooth	mostly smooth	drops	drops	drops	smooth	smooth	drops	drops	2	7x smooth
Proxy Workstation (Hamburg)	**progressive** 6 Mbit/s	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	0	4x smooth
	progressive 10 Mbit/s	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	0	4x smooth
Proxy Std. Client (Hamburg)	**progressive** 6 Mbit/s	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	0	1x some drops
	progressive 10 Mbit/s	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	smooth	0	1x some drops
Proxy Std. Client (Hannover)	**progressive** 6 Mbit/s	smooth	smooth	smooth	smooth	some drops	smooth	smooth	smooth	smooth	smooth	0	1x some drops
	progressive 10 Mbit/s	smooth	smooth	drops	some drops	smooth	smooth	smooth	mostly smooth	smooth	mostly smooth	0	1x some drops

Figure 10: Performance Evaluation

[11] https://cloudian.com/blog/object-storage-vs-file-storage/

Windows 10 and Apple MacOS clients connected to a central Harmonic MediaGrid Server storage. Each client has an available bandwidth of 1Gbit/s.

These test cases were as follows:

- Playback with different speed (1x,2x,4x,8x,16x)
- Rewind Playback with different speed (1x,4x,8x,16x)
- Scrubbing and navigating directly to different positions on the timeline
- Playback of picture in picture

The blue marked fields represent the hi-res editing setup in the city of Hamburg. Three different server types were each tested with internal HDD and central access to the MediaGrid.

The red marked fields represent another remote access via the city of Hannover. The clients there were MAC and a HP Z4 Workstation.

The grey field represent the results of HD-proxy editing. The proxy had been created with 6 and 10 Mbit/s
Two cases compared in detail underline the performance and possibilities of the technology used.

The streaming server is located in Hamburg, Germany and the clients are either in Hamburg or in Hannover. Hamburg and Hannover are connected via a 10Gbit/s connection, which is used for all network traffic between these two entities.

Workstation Hi-res and Workstation HD Proxy Streaming

A typical workstation for applications that require medium performance was used as a craft editing client at the German broadcaster that ran the tests and evaluation.

Figure 11: Workstation - with SSD and MediaGrid access test series

The reference test, which is in all test scenarios evaluated as smooth, has been performed in an ideal scenario, where the hi-res source files were locally available on that client using a SSD Hard Drive.

The typical scenario for production at this broadcaster is a setup of the second test on the Z4, which is used for daily news, documentary and feature production. The files are stored centrally as house format XDCamHD on the Harmonic MediaGrid Production Storage. This results in access and transfer via the on premise network.

Compared to the first series of tests, a slight decline of performance is noted, since fast forward and backward play results in notably drops. Hence this is recognized it does not influence its daily usage.

Figure 12: Workstation - HD Proxy test series

The same client was used, within the same location, to evaluate the performance with streamed HD Proxy. The result was in all cases smooth and closer to the reference as the production setup.

Standard Client Hi-res and Standard Client HD Proxy

A standard client at the broadcaster is a typical office PC which does not necessarily provide the hardware for craft video editing. Therefore the results for the test series show declined performance. Since these clients don't have access to the production storage, the test series was performed with the local Hard Drive and a mobile USB Hard Drive.

Figure 13: Standard client hi-res test series

The test series with HD proxy shows a better performance, than the direct hi-res access:

Figure 14: HD Proxy test series

The performance within the entity in Hamburg shows a smooth playback both for 6Mbit/s and 10Mbit/s HD proxy. Whereas the decoding of picture in picture results in some drops. Also the playback in the remote entity Hannover shows acceptable performance, with some additional drops.

With HD proxy, it seems to be possible to use existing office clients for editing when needed.
The overall results show that depending on the client type used and the connection to the storage, the performance varies even in a hi-res editing case. Overall hi-res and HDProxy editing shows a similar performance, which indicates that HDProxy Editing is a viable supplement for existing hi-res editing solutions. Except for picture in picture mode with multiple streams, the playback was described as smoothly with some dropped playback in fast forward.

Evaluation of Quality

The German Broadcaster performed additional tests to evaluate the quality of the utilized proxy. Experienced editors rated the subjective quality of the video, categorized typical use cases and evaluated the feasibility of those.

A test series was conducted with 2 (SD) ,6 (HD) & 10 (HD) Mbit/s H.264 HD proxy. Again a reference HP Z4 workstation and an office client served as basis. A red x marks cases where the quality of the proxy was not sufficient. A light green arrow marks cases which can be performed with proxy editing, but would require a final approval, ideally on the used hi-res or rendered clip. The dark green fields have been approved as cases which can be realized with proxy editing only. It is obvious that several cases cannot not be done with 2 Mbits/s SD proxy, since the resolution does not allow for the evaluation of sharpness, focal length or other technical parameters. Based on the parameters, a proxy will not match the quality of the original hi-res file, but utilizing a H.264 video codec with full HD resolution and a bit rate between 6 and 10 Mbit/s, results can be achieved that are acceptable for editors without significantly detriment of the user experience.

A HD Proxy shows good results on 'standard' edits like, titling, rough editing, simple effects, graphical templates and multi camera productions. Whereas elaborated editing, which requires e.g. technical evaluation and sophisticated graphics, touches the limits of proxy editing. For those cases, a viable setup could be a combination of HD proxy editing, with proxy hosted in the cloud and hi-res editing on-premises. Several edits can already be done and finished from remote, but when required, those edits can be completed on premise or exchanged with editors, which have access to the hi-res files.

	Requirement	Quality Evaluation			Comment
		2 Mbit/s	6 Mbit/s	10 Mbit/s	
Multicam	Multicam Edit 2 sources parallel	×	✓	✓	simple studio production
	Multicam Edit 6 sources parallel	×	✓⇗	✓⇗	e.g. concert with more than 5 cameras
	Multicam Edit more > 6 sources	×	✓⇗	✓⇗	e.g. complex concert production with more than 6 streams
Titling	ProRes titles	✓	✓	✓	Titles are imported to project as HiRes Graphics
	Titling with Easy Insert	✓	✓	✓	Titles are imported to project as HiRes Graphics
	Motion Graphics Templates (mogrt)	✓	✓	✓	
	Delivery Master: FCC + Mix+ Inserts	✓⇗	✓⇗	✓⇗	
	Set and edit subtitles	✓	✓	✓	
Rough Edit Work-flows	Browse material	✓	✓	✓	
	Collect content for project	✓	✓	✓	
	Simple edit with rough cuts	✓	✓	✓	
	Nonlinear Studio edit	×	✓	✓	Image sharpness needs to be evaluated
	Rough-cut for Program	✓	✓	✓	
	Trimming	✓	✓	✓	
	Assemble archive content	✓	✓	✓	
	Voice Over	✓	✓	✓	
Approval	editorial acceptance, internal control	✓	✓	✓	
	Technical quality control	×	×	×	
Material	Live Server Cut (Feed Ingest) Growing Files	✓	✓	✓	proxy is generated on growing file
	Take over VJ projects ("refine")	×	×	×	depending on project size, the proxy transcoding required
	Evaluation of pixelated material (artifacts) (LiveU)	×⇗	×⇗	×⇗	Artifacts from the source cannot be distinguished from the artifacts of the stream.
	Working with drone material (origin: 4K)	✓⇗	✓⇗	✓⇗	Downscaling causes moiré patterns and line flicker
	Edit with archive material	✓⇗	✓	✓	e.g. differentiate SD and HD Source
	Aspect Ratio Recognizable (4:3, 16:9, LB, PB, Zoom)	✓	✓	✓	for archive content of false scaled content
	Native cut with S-log, C-log, 10 Bit Material	×	×	×	working with lookup tables not possible. Proxy does not provide 10bit dynamic range

Effects	Cut with simple clip- or transition effects	✓⇗	✓	✓	wipe, white/black flash, transition etc.
	Cut with elaborate effects (multi-layering)	×	✓⇗	✓⇗	effects on multiple layers, key framing etc.
	Cut with effects like C. pinning or spilt screens	×	✓	✓	C. pinning or spilt screens
	Cut with After Effects elements	×	×	×	After Effects Projects with Dynamic Link relation
	Animate photos, Ken Burns effect etc	×	✓⇗	✓⇗	Key frame editing, Bezier curves. Etc.
	simple pixelations	✓	✓	✓	
	Pixelations and (auto) tracking	×	✓⇗	✓⇗	
	Simple keys	×	×	✓⇗	Assessing clean key is difficult, because artifacts might be result of compression or key.
	Elaborate Keys	×	×	×	Exact settings of the keys with spill and color suppression etc.
	Working with masks	×	✓	✓	
	Edit with graphical templates	✓	✓	✓	
	Edit with with trailer packaging	×	×	×	After Effects Integration
	Slow-motion or time-lapse	✓⇗	✓⇗	✓⇗	
	Slow-motion curves and Speed ramps	✓⇗	✓⇗	✓⇗	
Special Cases	Create web videos (div Aspect Ratio, etc.) without After Effects	×	✓	✓	
	360° Videos	×	×	×	including stitching -requires dedicated plugins and software
	Simple color correction on the editing client	×	✓	✓	standard pc lacks calibrated monitors
	Elaborate Color Correction, Color Matching	×	×	×	Would theoretically be possible in Premiere (HiRes) - but is not performed
	Highlight Edit incl. Playback from Timeline during Live Events	×	✓⇗	✓	sports - halftime edit etc.
	Edit with short time to air	✓	✓	✓	proxy is generated on growing file
	Projects with various distribution targets	✓	✓	✓	

Figure 15: Quality Evaluation

An additional test series was performed during the Arvato User Group Meeting in 2018. Different test clips were shown to an audience of ~120 experienced broadcast experts and engineers. The test clips showing a Siemens star and bars could not clearly be determined as proxy or original XDCAM HD, whereas a sequence of a drone flight was more obvious. The sequence of the drone flight did show more artifacts on details and areas of low structure. Nevertheless, engineers evaluated the test installation as suitable for producers and craft editing.

A poll on the question "For which user groups could HD remote editing be a replacement or extension to the current craft editing?" provided the following results:

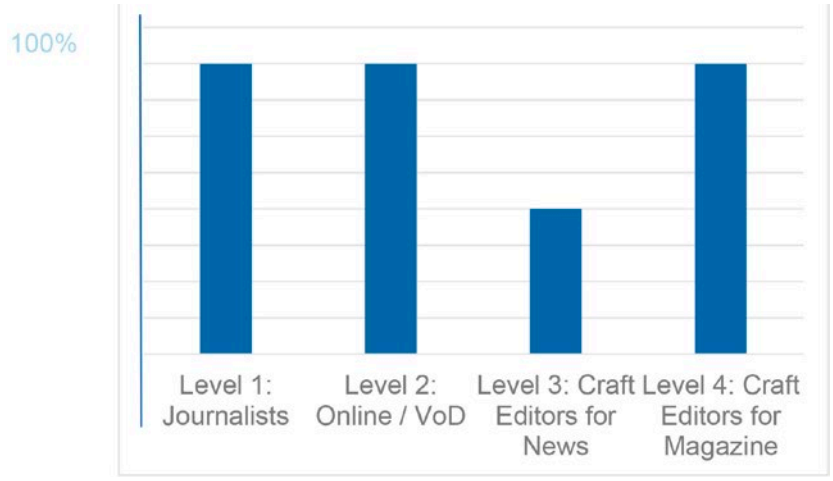

Figure 16: HD remote editing User Group evaluation

Conclusion

For a runner – Usain Bolt, for example – to improve his speed, there is no single element that he works on. He and his team work on improving multiple elements of his performance: if his stride can be a centimeter longer, his starting time a split-second faster, then he will achieve a still-faster 100m. In the same way there is no single change that will improve and accelerate cloud-based editing workflows to the extent that cloud-based editing becomes the norm.

What is needed is teams of experts with in-depth understanding of the various elements to work on each area – server-storage connectivity, the use of TCP, reduction in the number of frames transferred, improved streaming protocols and better handling of audio files, as well as smart local caching.

Cloud editing is possible. With these improvements, it can become a reality.