TR501196

Road to the Digital Shop Floor. Streamlined Decision with VRED Core

Lionel Graf
Autodesk – Implementation Consultant Automotive

[Co-Speaker/Panelist Name]
[Co-Speaker/Panelist Company]

Learning Objectives

- Learn about how to take advantage of VRED Core streaming capabilities.
- Learn about the business ROI of a streaming collaboration platform.
- Learn how to implement on-demand visualization as a service.
- Learn how to visualize design data in a device-agnostic environment.

Description

Digitization of creative design workflows has unleashed creators' capabilities and allowed them to experience the designed product earlier. But it has some downsides. Back in the day, one could go down to the shop floor and walk around clay models having an instant update on the project progress and design evolution. Today, it can be challenging to find the right visualization file aggregating the latest modelling data, unless you know who is working on the visualization data and you get over their shoulder to have a quick look on the latest evolution. What if the shop floor was digital too? What if anyone could go to a web site and walk through the latest design updates. Find out how you can serve visualization streams using a cluster of VRED Core instances managed through a simple web server.

Speaker(s)

Lionel Graf is an implementation consultant with the Automotive Consulting Team at Autodesk, Inc. He has been working for over 14 years in the rail industry as a creative design production manager. He specializes in creative design visualization and communication, with a deep knowledge of real-time technologies and processes to achieve high-end visual quality for aesthetic design communication and real-time design reviews using VRED software, ALIAS software, MAYA software, 3ds Max software, and creative field standard software for image creation.

The solution described here has been developed by Autodesk Consulting. It uses Autodesk VRED streaming capabilities to address access and sharing challenges Customers are facing when dealing with real-time 3D content and digital mock-up.
We developed a customized Web Server to manage users, data and VRED rendering instances.
Autodesk Sales representative can redirect to the right Consulting expert if more information is needed.

Why did we do it?

The need for democratized access to visualization data came from various discussion with Customers who struggle with accessing to the right data, with the right maturity, at the right moment.
Today, design studios creation pipeline is highly digitized, which produces an exceptionally large amount of data, while being managed with traditional method, using naming conventions and shared file storage. This requires a lot of effort to maintain and to make sure that growing teams follow the rules, often perceived as a constraint and an additional effort by creative people.

How do we work today?

Our working environment and tools have drastically changed in the last decade. The digital environment we are working in today give us the opportunity to access to a huge amount of information, share our work and experiences, react, and collaborate instantly on what our social network wants to share with us.
It is quite common today to work together and collaborate on a common document, even at the same time.

Digital design opportunities and struggles.

The tools we use are changing rapidly, so are the workflows.
It opens an entire world of opportunities, giving us the ability to create faster, to integrate more complexity earlier in the design process, and to be closer to reality, way before anything exists.
Most of the digital creation tools are today capable of reaching a level of realism that makes images and movies relevant enough to take informed decisions. It makes it simple to place the product we are designing in its context of use, explore variations, even make it interactive and have a life-like experience.
Technologies like Virtual Reality have reached a level of maturity which makes it affordable and easy to use, allowing us to reduce the need for costly physical mock-up in the early design phases.
But it requires specific knowledge. Creating such outputs usually relies on a small group of experts who need to handle a growing demand for communication materials to illustrate decision-making gate reviews.
Also, it usually requires dedicated hardware to create and consume this type of output and experience. It is not rare to see dedicated hardware setup for presentation, whether it is because of the hardware investment needed, or the expertise needed to manage it.

So, what happen when a decision maker wants to review the latest design in the next hour?

Best case scenario, we need to find who is working on it - to make sure we will get the right data – and ask him to urgently prepare himself to present his own work. We need to find the right hardware, it may be a presentation room where we can find the right software package which has been used for creation, or in the worst case, its own workstation.

Or we need to push it to a later date, the time to prepare everything…

What if we could make it easier, and give the ability to anyone, and especially decision-makers with no specific knowledge on design tools, to quickly access the most relevant data and have a friction-less experience of the work in progress, without compromising on the quality?

Technological opportunity.

Autodesk VRED is a visualization software that allow to bring your complex data to life. It is used in the automotive industry to

- create high-quality renderings and interactive experiences,
- visualize, review, and validate with ease and accuracy,
- experience and collaborate in a real-time 3d environment on any device, including VR (Virtual Reality).

Autodesk VRED is also proposing a web streaming feature which allows any VRED session to be streamed over the web and thus be displayed on any device, including portable devices, phones, tablets and more.

Coupled with a custom web server, Autodesk Consulting has built a solution which securely allow registered user to access shared data, managed and exposed through a customizable web front-end.

Value driven.

The solution, allowing anyone to access through a web environment to high-end visualization data, is driving the following value:

- Save operational time.

Reducing the need for expert support to find, prepare, and manage ad-hoc presentation

- Ease access to the right data

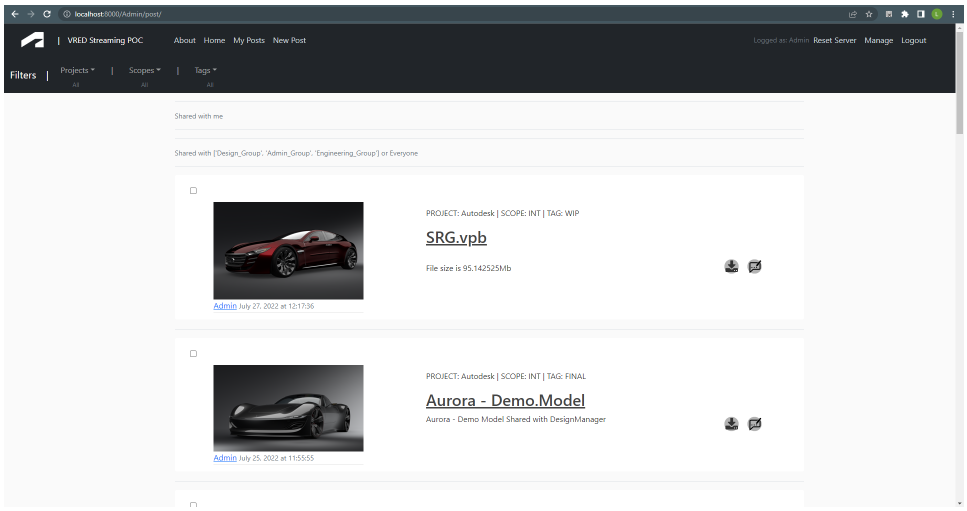
Remove the technological barrier by allowing any device to consume visualization data

- Shorten Design cycles

Influence design changes earlier, detect earlier wrong design direction, shorten time to find the right data

How does it look like?

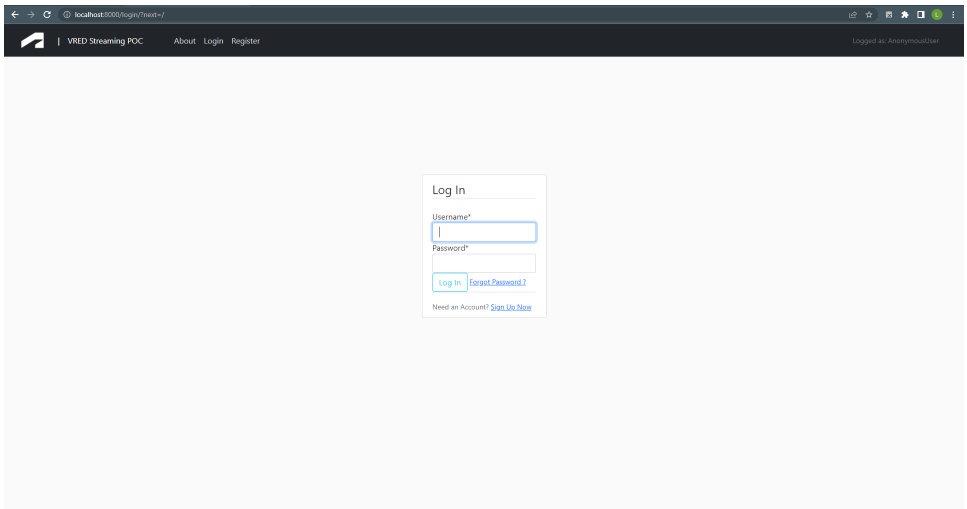
The goal was to ease access to available data. To do so, we designed a simple web front-end which looks like a blog, where authorized users can see a list of available data to review.
Each “Post” is illustrated with a thumbnail image of the content uploaded, and user can add information to describe the content, the context and choose who he wants to share the data with.



The web application Home page.

Login

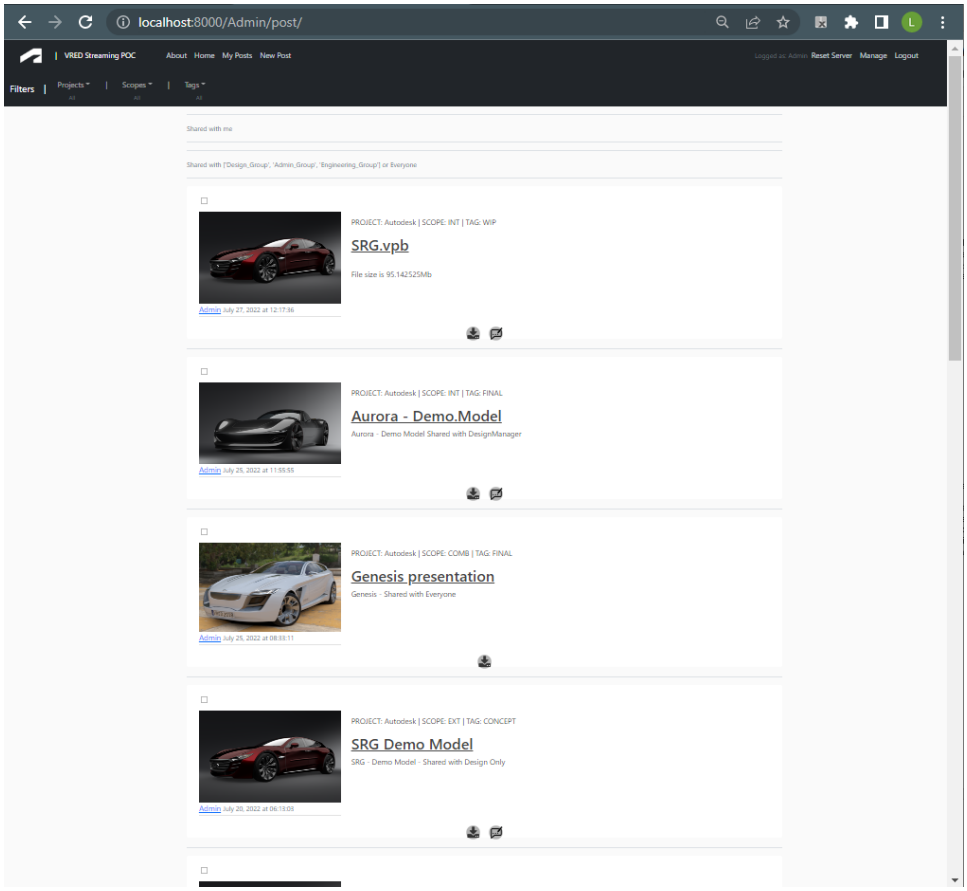
The system manages its own list of users.
Each user can be part of a group and have specific rights to create, see and change content.
When user access the system URL, he needs to login, using his personal credentials.



The login page.

Browse

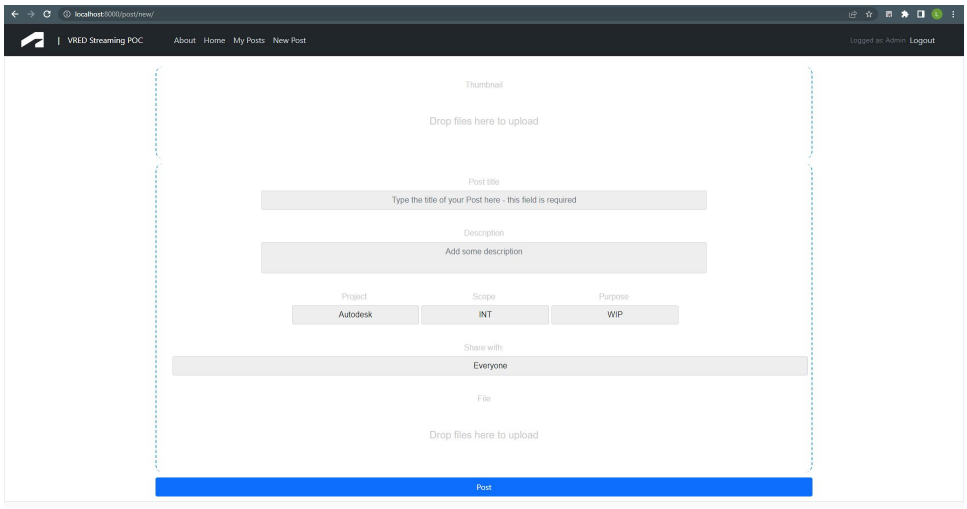
Once logged in, users can see the list of data available to them. Posts are organized by date, in descending order. Posts can also be filtered by metadata, the categories being fully customizable through the management interface.
As an example, user can here specify the project, the scope and the phase related. Each of them can be used and combined to filter the visible data.



The Posts list with filters.

Post / Share

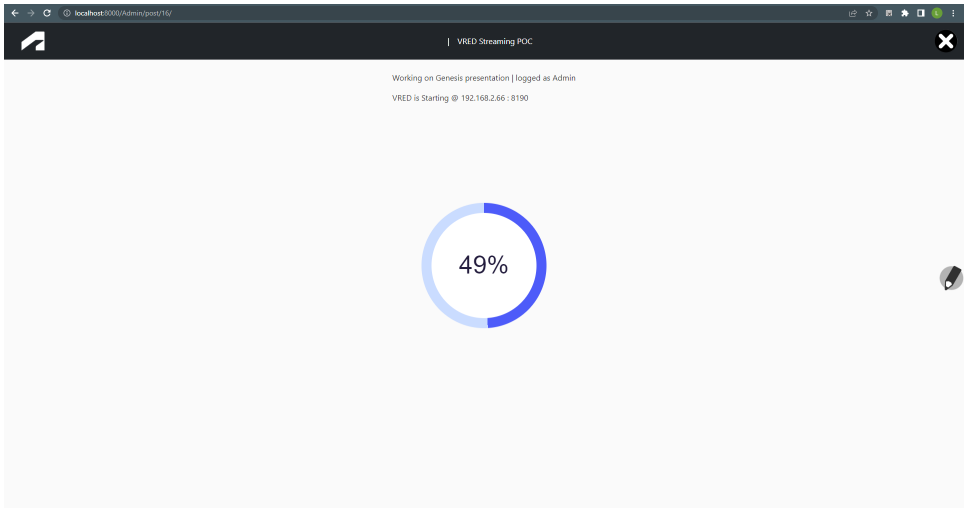
The philosophy here is that sharing is intentional. If a user wants to share data with a user or a group, he will need to create a new Post, and upload data to the system.
The post creation form requires the user to fill in information which will be used to track data type and content and will also be used to set the uploaded file name in the system.
Users will also decide here who will be able to see the uploaded data:
Data can be shared with one single person, or a group, or everyone. If data is shared with a specific person, it will only be accessible by this user, and the author only will be able to change it.



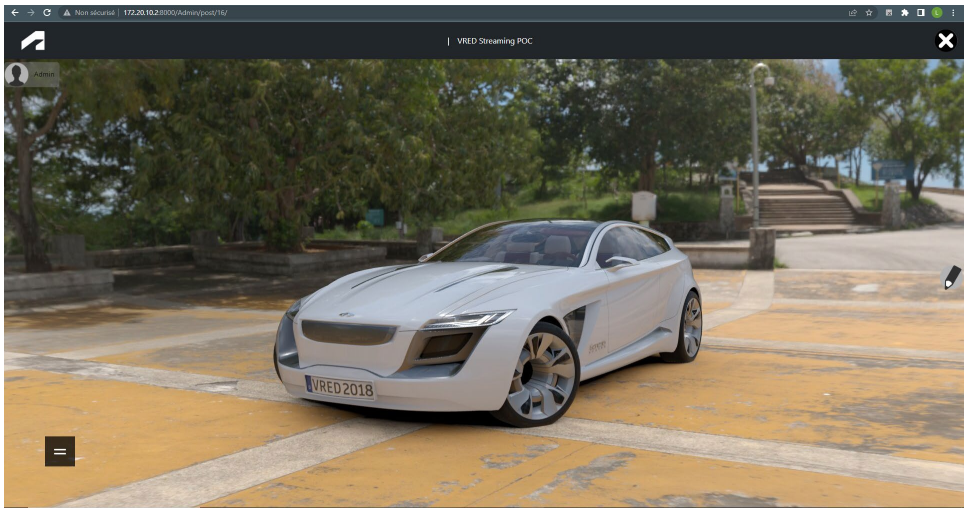
The Post creation form.

Review

Consulting a Post content is as simple as a click.
The system will take care of finding an available machine, start VRED, create a unique streaming pipeline, open the file, and display it in user's web browser.
From there, the user can have the same experience as if he were running VRED on his local computer, with the same visual quality and interactivity, without having to worry about where the data is stored or if his hardware can handle such a file.



The Post loading page.



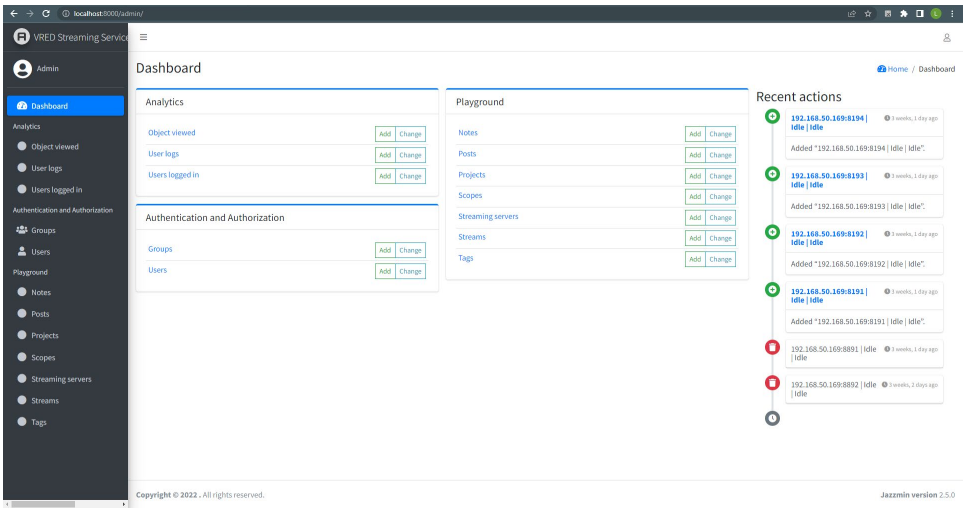
The Post VRED streaming page.

Additionally, the user will be able to take notes by drawing on top of the 3D viewport. On submit, the system will capture and store the notes information in the file: the note image, the viewpoint, the user who took that note.

Taking a note.

System Management

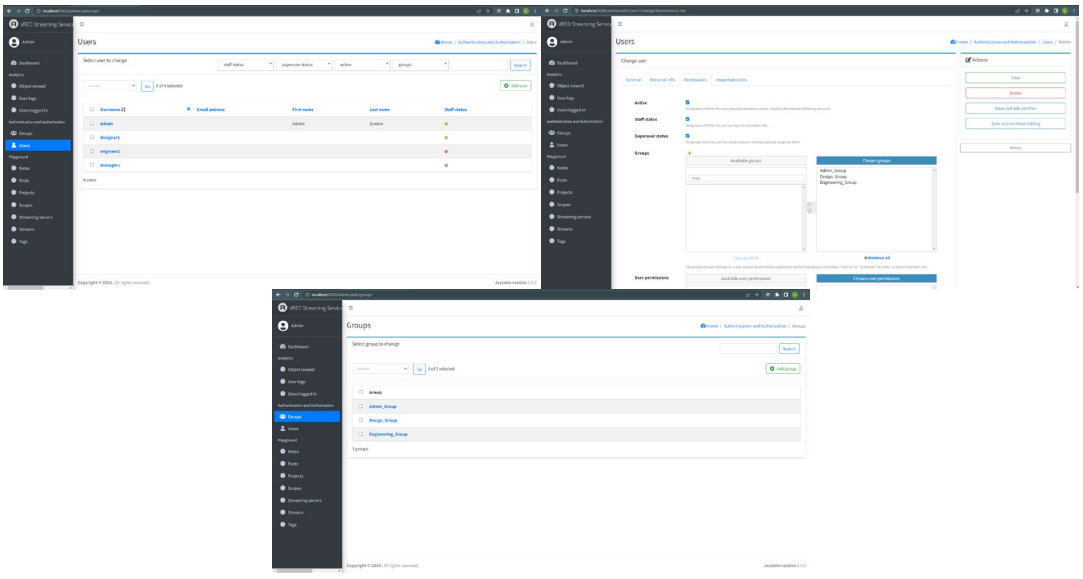
The system integrates an administration console through which power users can manage users list, groups, projects and posts metadata, and declare the rendering instances to be part of the streaming cluster.



The administration console.

Users & Groups management

The system manages its own list of users and groups.
Administrator can create new users in the management console, or let users create a new account through the login page.

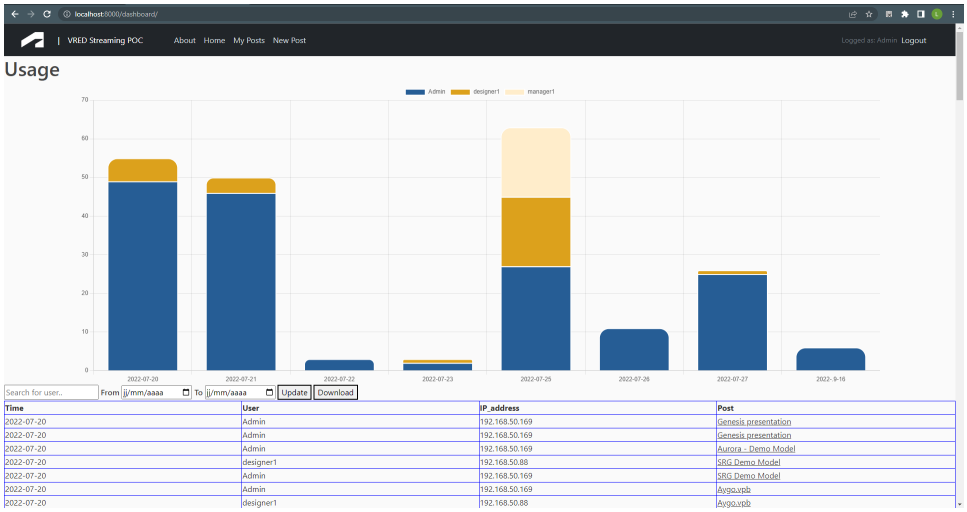


The user & groups management console.

Analytics

As we continue to develop new functionalities based on customer requests, we can also build custom features like, for example, tracking system usage for management purposes.

This allows us to know who consulted what and take decisions on system scale.



The dashboard page.

Data

We made the choice to have a dedicated data repository, so that uploaded files can be effectively managed and changed as needed, without interfering with work in progress data generated daily in the studio.
When a user chooses to share some data, the source file will be uploaded to the web server, renamed, and stored in a specified location (managed through a configuration file).
This way, we can store notes taken and other information as people collaborate on it, and let the user decide if this should be the starting point of the next working activities by downloading it.

Use-cases

The proposed solution applies to many use-cases, from getting a quick overview of work in progress to automated workflows.

Daily check on work in progress.

This is the most obvious situation where a visualization self-service application can be useful to decision makers and team leads who need to check on work being done by the team, without necessarily having to disturb the ongoing work.
Content creators can feed the system with relevant data, choosing what can be shared, with the right level of maturity, and give access to something managers can react to.
Thus, decision makers can detect earlier a wrong design direction, or be more efficient in influencing the creative direction, not having to wait for the next formal presentation or gate review to give feedback.
Thinking about team collaboration, and especially in the context of spread team, whether it is because of multi-site collaborative work, or because we are working from home, having a place where one can freely access to the latest version of a model and visualize it in its best quality, without compromising on the interactivity could help saving time in organizing online meetings to review the ongoing work and improve collaboration efficiency.

Collaborative Design Reviews.

One of the amazing capabilities of Autodesk VRED is to allow users to make collaborative design reviews, gathering people from distant sites around the same digital mock-up, and being able to see each other, share viewpoints, from a desktop workstation or in VR.
One just need to have the same model running on different machines that can connect to each other and exchange users position information and 3D scene state.

But anyone who has done this before knows that it requires experience and expertise to setup the system, be aware of some IT requirements and constraints, prepare the meeting upfront, making sure that everyone has the same model, configure the scene and start collaboration on each side. It may sometimes even require you to book a room and have the support of the local expert to set up everything.
What if we could automate everything and save the setup time and effort, and allow anyone to simply jump in a collaborative review in a click?
Well, this is what this system can do, using VRED API to automatically create a collaboration session whenever a user is viewing a model, or join it if someone is already viewing the same data.

Self-service Rendering.

Creating renderings usually requires a deep knowledge of visualization software. Even though one has access to the right software package, setting up the scene for rendering is not trivial, so we usually turn to the experts and request some images...
Such a system could automate all the heavy lifting stuff, data preparation, lighting setup, apply templates for environment, and let anyone just choose the viewpoint and hit “Render”.

Compare multiple Designs.

What if a decision maker could select multiple data sets uploaded to the system and bring them together in a single visualization context, put them side-by-side or on top of each other, and compare them?
In this context, taking informed decisions is made easier, removing the need to put someone in charge to reach out to multiple teams and gather information, aggregate them in a single scene and organize the meeting to review them together.

Automate data aggregation.

Being able to aggregate information could also be particularly useful in the context of split modeling work when model development is spread across multiple people.
Think about the simple example of a team working on the exterior of a car, while another one oversees the interior. The system could make it easy to review them together in the same environment.

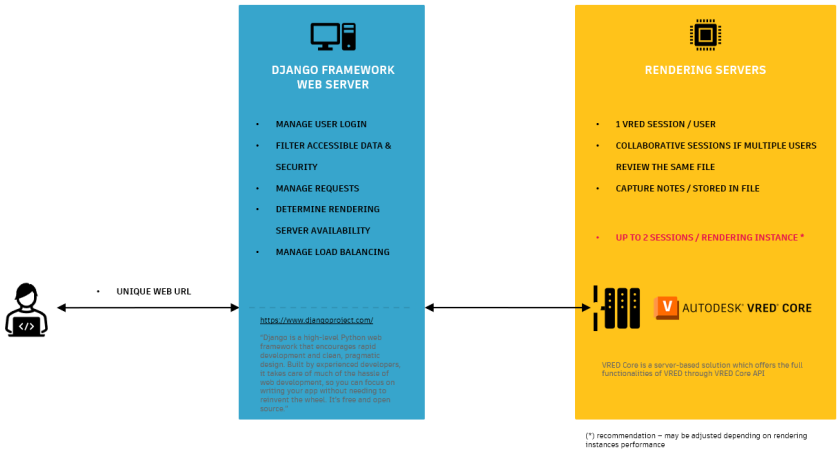
Sharing data outside of the organization.

Another situation where streaming is valuable is when confidentiality rules are preventing us from sharing data files.
Streaming content is there an ideal way to make sure that there is no loss of information, as the actual data never leaves the company-controlled repository, and tavoid misinterpretation thanks to real-time 3D interaction.

Additionally, depending on the audience, we could develop custom features like, for example, being able to take measures or section the model if we target engineering people, or leverage model information and display parts code, author, maturity, if we are reviewing engineering assemblies.
Possibilities are almost endless thanks to VRED's Python API which allows us to customize the user experience.

Technological Components

The solution is based on VRED's streaming capabilities, managed by a customized web server developed on Django framework.
The web server part is managing system access, users, posts, data, and the list of available streaming machines. These will use VRED to stream the 3D content to the client web browser.



The system components.

The web server.

Relying on Django framework, the web server is serving user frontend, and manages user requests to consult the data base and review content.

Django.

https://www.djangoproject.com/ says:
"Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the

hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source."
It fits very well with VRED's python API and makes custom development easier.

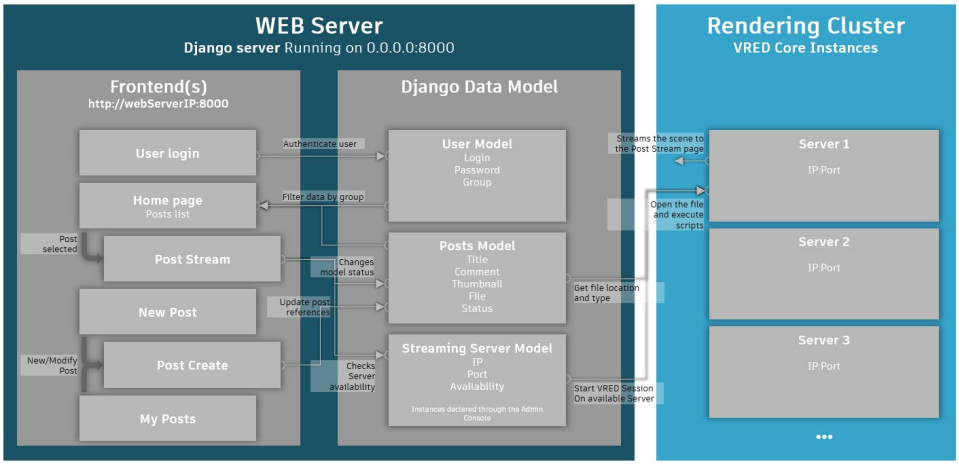
Data model.

The application has been built like a Blog.
The Django application relies on 3 main pillars:

1. The user/group model.
   The application is managing its own list of users.
   Each user belongs to a "group", which will help to determine what data will be available to him.
   In the future, User management will be further developed to allow Active Directory connection.

2. The Post model
   The Post is the data container.
   It holds information about the model, the title, description, image thumbnail and the path to the model and contains metadata, which are asked to be filled to the user on Post creation.
   Metadata are configurable. As an example, Post model currently considers 3 parameters: The project, the scope (Is the model related to Exterior, Interior, CMF…) and tags (WIP, CONCEPT, FINAL, ENG…).
   The Post is also holding information hidden to the user like who is currently viewing it, which is used for collaboration, and private collaboration session key.

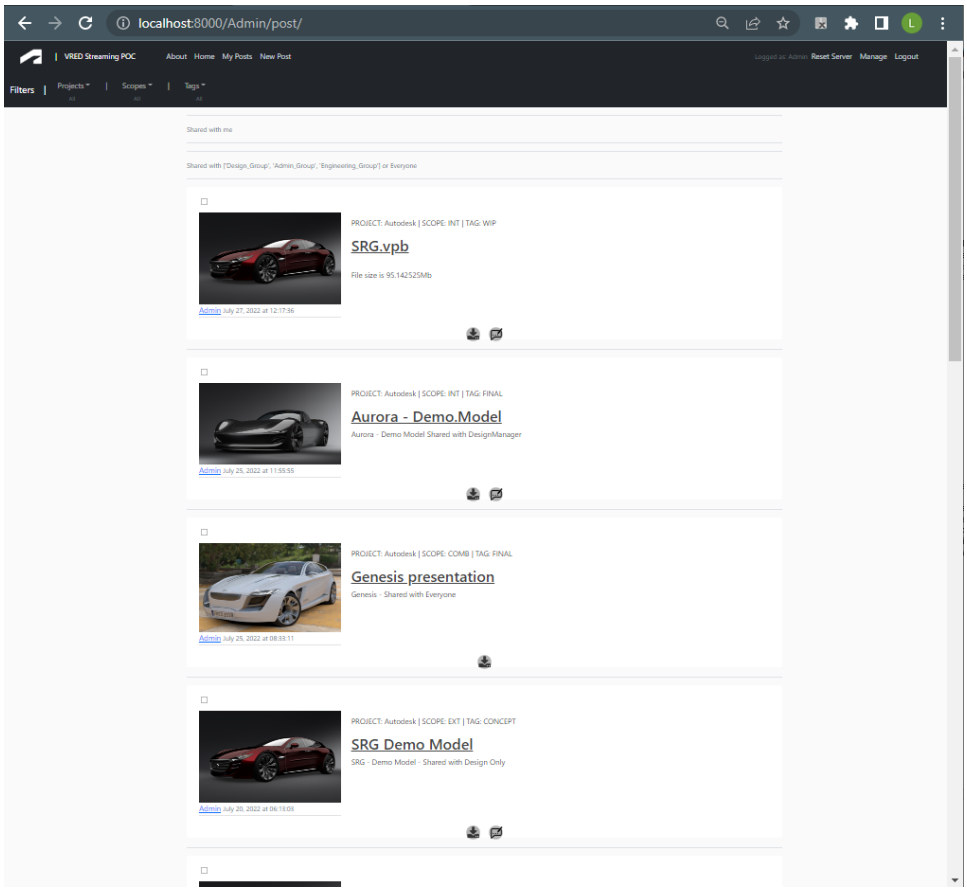
3. The Streaming Server model.
   A streaming server is a couple IP+port which will be streaming the VRED content.
   The system will use these information to start a remote VRED session and communicate with VRED during the review session.
   The streaming server model is also capturing the state of the given server which helps to determine server availability and manage load balancing.



The system architecture.

Front End.

The Web Server manages the web pages exposed to the user during navigation. These are fully customizable and can be tailored to the company.
The application Front End is using HTML, CSS, Javascript and Bootstrap framework.



The HTML Front-End.

The Rendering Server.

Rendering server can be hosted anywhere as long as it can communicate over Http requests with the Django web server.

When user requests to review a model by clicking on a post link, the Django Web server starts a VRED session on the machine with the specified IP, using the specified port for the VRED Web server. This way, the communication pipeline is setup between the Django Web server and the VRED session and Http requests can be exchanged between them, allowing to trigger custom actions, like for example in the case of note taking, or to display who is currently reviewing the same model.
At start, the Django Web Server sends also a Python script which holds the custom functions that VRED will use during review session. For example, collaboration session creation and update is automated through this script which makes sure a collaboration session is created at session start or check if another session exist already and joins it.
The Streaming Server model can support an unlimited number of VRED servers. Django web server is taking car of the load balancing, taking the next available server in the list at a session start, and making sure the server is released when a session ends.

What's next?

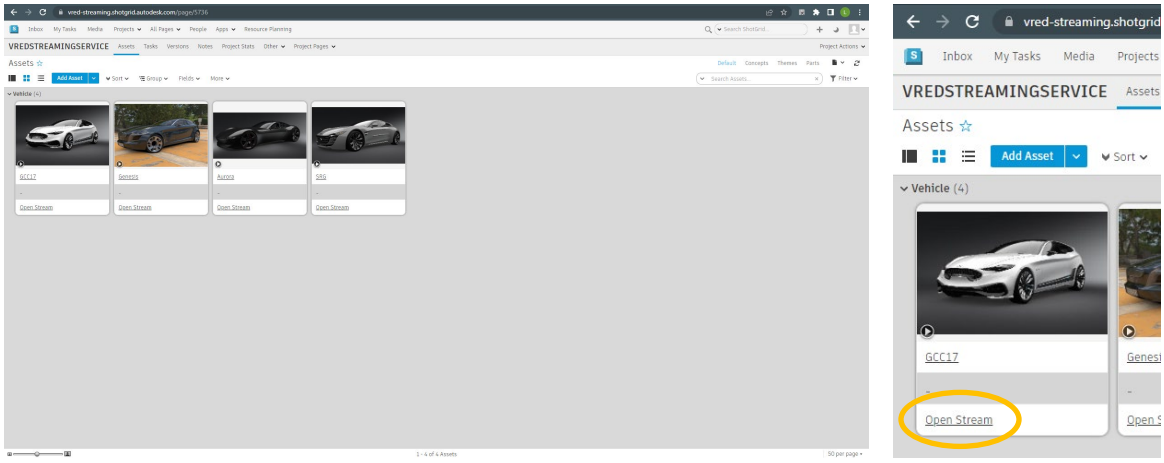
The application showcased here is a prototype that has been developed based on few customer feedbacks. Each creative studio may have his own pipeline and specific needs.
We believe that the power of this type of application is in its versatility and ability to be tailored to specific needs, thanks to the flexibility offered by VRED's Python API.
So far, we focused the first set of features on real-time 3D reviews and collaboration workflows, but it can also serve other purposes through automation.
It can also be integrated into any existing enterprise application.

Posts Model vs. Live data.

The chosen approach has been to make data sharing intentional.
Users have access to what someone else chose to share with them. This way it is easier to manage confidentiality, data relevancy, by making people responsible of sharing the data they own.
Another approach may be to give people access to live data.
What if the system could identify information added to a repository, like the network drive where the daily work is stored, and, based on some rules, grab, convert and prepare data for review.
It would be much easier to use, but would require people to follow more rules like naming convention for example in order to be sure the right data is pulled.

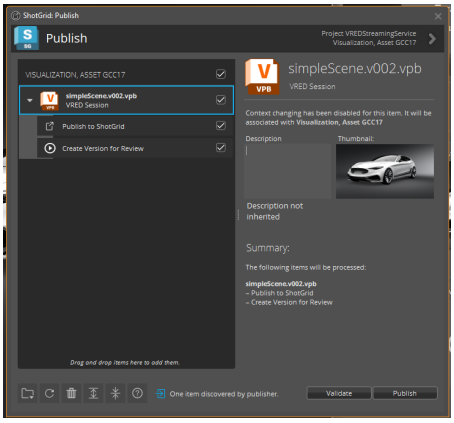
Autodesk Shotgrid integration.

A good approach to manage data is to use Autodesk Shotgrid.
As the number of user and the volume of data grows, it is a good idea to leverage the power of Shotgrid to make data management and collaboration easier.
We could for example integrate such an application with Shotgrid and create Shotgris assets whenever a new Post is created, use Shotgrid's collaboration feature to store notes, comments, and notify users when a note is created, and automate the creation of the link to view the file with VRED streaming.



Example of Shotgrid project with dynamic web link to .

Shotgrid can also be integrated with authoring application, which could make data sharing even easier: what about sharing a VRED real-time review scene from within VRED, without having to open any other application?
Shotgrid Desktop can do that.



By leveraging Shotgrid, we can easily think about extending the support to more data formats, images, movies…

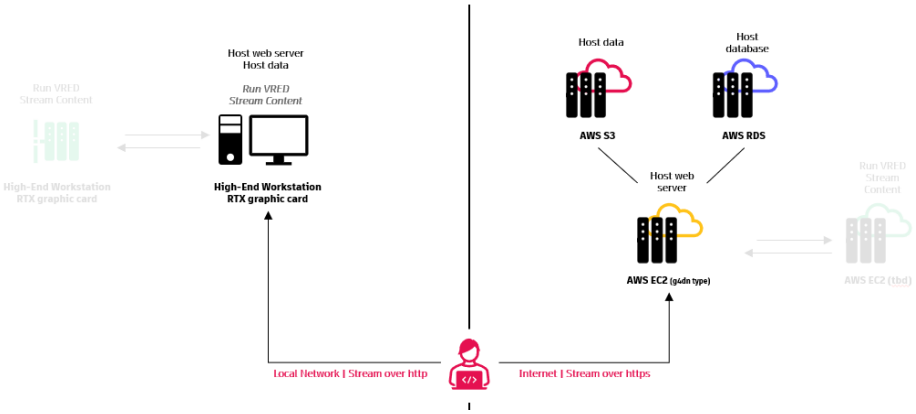
Scaling the service.

Decision making process has been impacted by the work dynamic shift created by the COVID pandemic. Design reviews we were used to make in-person at the Design Studio where stakeholder would congregate around large format screens and VR headsets tethered to powerful workstation has become something difficult to organize.
Within the last years, accessibility to VR and AR devices has been on the rise, leading to new capabilities in the Design Studio.
This paired with the need to facilitate reviews among stakeholders who are located remotely has created the perfect accelerant for virtual design reviews.

Hosting a service, maintaining it and making it scale as the demand grows is a challenging task. Cloud services providers are developing solutions to facilite deployment of the relevant technological components to support high-end visualization and streaming.

AWS for example has been working on developing an AWS Quickstart to easily deploy Autodesk VRED and NVIDIA CloudXR on AWS EC2 G5 instances, and help implement and extend Autodesk VRED workloads on AWS Cloud (https://aws-quickstart.github.io/quickstart-nvidia-cloudxr/).

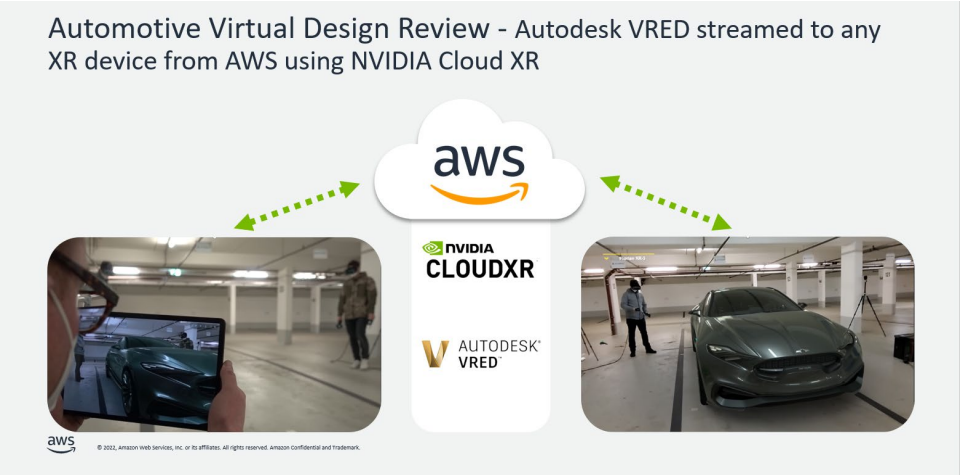
Originally designed to be deployed on-premise for confidentiality reasons, the application can also be tailored to work on a cloud architecture.



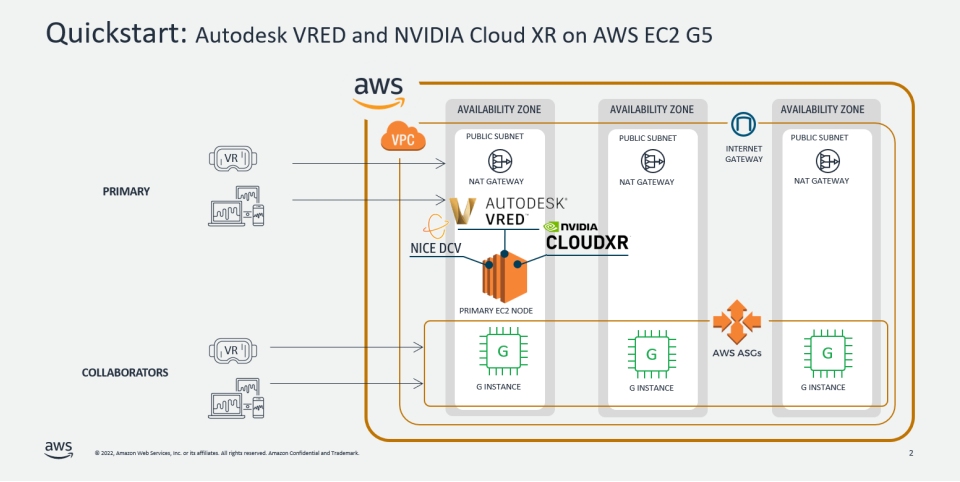
Powered by the new Amazon EC2 G5 instances, delivering up to 3x higher graphic performance an up to 40% better price perfomance than G4dn instances, the application can be hosted on the cloud and scale as the demand evolves.
It would remove hardware maintenance overhead and make following hardware evolution much easier.
Together with AWS VRED+CloudXR Quickstart, we can even think about offering VR and AR reviews in collaboration without any setup or configuration.

AWS Quickstart – Autodesk VRED and NVIDIA Cloud XR on Amazon EC2 G5.

Solution - In order to meet the desires of BMW, VW, and other automotive OEMs, three components are required: 1/ Design software such as Autodesk VRED that has the capability to render a 3D vehicle in context, at scale, and to multiple stakeholder points of view from an on-premise server or cloud server (VRED Core). 2/ A streaming protocol such as Nvidia CloudXR that compresses server-side rendering and decompresses client-side images at low latency, while transposing 6 DoF (6 degrees of freedom) data from devices, back to the server to render the next frame from the adjusted point-of-view in near real-time. 3/ Cloud or on-prem infrastructure supporting scalable, on-demand real-time graphics workloads with low-latency edge delivery capability. Automotive designers prepare the vehicles for evaluation in Autodesk VRED on desktop workstations (possibly a Nimble Studio opportunity in the future), then publish the content to the VRED Core (on a Windows G4dn instance residing on AWS or on Outpost/Local Zone). Nvidia CloudXR, on the same instance compresses real-time rendered images from VRED and streams them to Nvidia CloudXR clients (VR headsets or tablets) operated by design review stakeholders from independent locations and perspectives, simultaneously. (See Appendix A for a graphic representation and Appendix B for CloudXR architecture). This project involves establishing an AWS Marketplace Quickstart (Cloud Formation Template and documentation) to simplify the provisioning of this solution for all automotive OEM design studios and other Autodesk VRED customers.

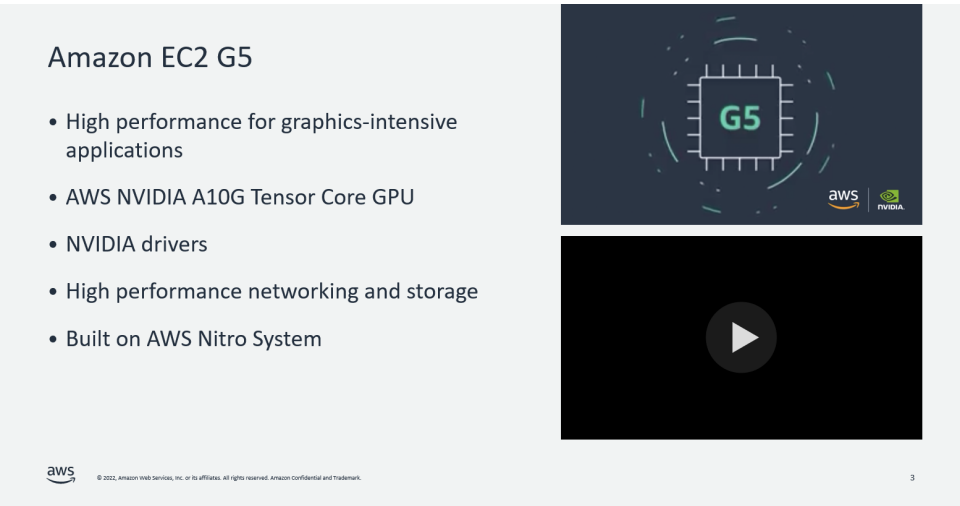


Opportunity - This use-case is prevalent across most automotive (consumer and commercial), industrial equipment and aerospace organizations. Each design program is a candidate for this workload. We predict that each brand/customer would leverage this solution on a regular basis, supporting multiple, parallel design programs for new and updated products. For example, automotive design studios may have 11 distinct product groups on various cycles of incremental or major design revisions, resulting in a cross section of weekly stakeholder reviews. Review sessions are typically 1-3-hour blocks with 5-8 participants. Projected costs would be ($1.5k - $2.5k+ ARR per customer based on ~$3/hr for G5 x 6 participants x 2 hours x 52 sessions per year).



https://aws-quickstart.github.io/quickstart-nvidia-cloudxr/
This Quick Start deploys VRED and CloudXR on the AWS Cloud. If you are unfamiliar with AWS Quick Starts, refer to the AWS Quick Start General Information Guide.
Autodesk VRED 3D visualization software **helps designers and engineers create product presentations, design reviews, and virtual prototypes** using interactive GPU ray tracing and both analytic and cloud-rendering modes.

This Quick Start is for IT infrastructure architects, administrators, and DevOps professionals who are planning to implement or extend their Autodesk VRED workloads on the AWS Cloud. There is no cost to use this Quick Start, but you will be billed for any AWS services or resources that this Quick Start deploys.



https://www.youtube.com/watch?v=zz38lS3gh_E

High performance for graphics-intensive applications
G5 instances deliver up to 3x higher graphics performance and up to 40% better price performance than G4dn instances. They have more ray tracing cores than any other GPU-based EC2 instance, feature 24 GB of memory per GPU, and support NVIDIA RTX technology. This makes them ideal for rendering realistic scenes faster, running powerful virtual workstations, and supporting graphics heavy applications at higher fidelity.

AWS NVIDIA A10G Tensor Core GPU
G5 instances are the first in the cloud to feature NVIDIA A10G Tensor Core GPUs that deliver high performance for graphics-intensive and machine learning applications. Each instance features up to 8 A10G Tensor Core GPUs that come with 80 ray tracing cores and 24 GB of memory per GPU. They also offer 320 third-generation NVIDIA Tensor Cores delivering up to 250 TOPS resulting in high performance for ML workloads.

NVIDIA drivers
G5 instances offer NVIDIA RTX Enterprise and gaming drivers to customers at no additional cost. NVIDIA RTX Enterprise drivers can be used to provide high quality virtual workstations for a wide range of graphics-intensive workloads. NVIDIA gaming drivers provide unparalleled graphics and compute support for game development. G5 instances also support CUDA, cuDNN, NVENC, TensorRT, cuBLAS, OpenCL, DirectX 11/12, Vulkan 1.1, and OpenGL 4.5 libraries.

High performance networking and storage

G5 instances come with up to 100 Gbps of networking throughput enabling them to support the low latency needs of machine learning inference and graphics-intensive applications. 24 GB of memory per GPU along with support for up to 7.6 TB of local NVMe SSD storage enable local storage of large models and datasets for high performance machine learning training and inference. G5 instances can also store large video files locally resulting in increased graphics performance and the ability to render larger and more complex video files.

Built on AWS Nitro System
G5 instances are built on the AWS Nitro System, which is a rich collection of building blocks that offloads many of the traditional virtualization functions to dedicated hardware and software to deliver high performance, high availability, and high security while also reducing virtualization overhead.

URHO KONTTORI - FOUNDER AND CHIEF TECHNOLOGY OFFICER, VARJO
"FOR HIGH-END VR/XR APPLICATIONS, AMAZON EC2 G5 INSTANCES ARE A GAME-CHANGER. WE'RE ABLE TO RUN PROFESSIONAL APPLICATIONS IN VARJO'S SIGNATURE HUMAN-EYE RESOLUTION WITH THREE TIMES THE FRAME RATE COMPARED TO G4DN INSTANCES USED BEFORE, PROVIDING OUR CUSTOMERS WITH NEVER-BEFORE-SEEN EXPERIENCE QUALITY WHEN STREAMING FROM SERVER."

Automation opportunities.

Leveraging VRED's Python API, we can do almost anything and extend the usage to automation…

Automate data preparation tasks.

VRED supports a broad range of data format, from exchange format (FBX, STEP, IGS, SAT, Alembic) to native CAD application files and assemblies (Alias, Inventor, Autocad, CATIA V4, CATIA V5, ProEngineer, Rhino) and content creation tools (Maya, 3DS Max, Cinema 4D).
Data structure can differ from one file format to another, and preparation tasks needed may be different.
Those tasks can be automated via python scripts which can be triggered on data import (scene organization, geometry optimization, material replacement, template application)

Streamline Visualization workflows.

Combining live data pulling and automated data preparation, this type of application could also be used to handle more complex tasks.
When dealing with large data assemblies, the system could automate data conversion and preparation tasks in the background, making sure that the right, optimized data is available when needed.
We could also think about automating data aggregation and update whenever a new version of part has been made available and streamline visualization workflows.

Off-line rendering service.

So far, the application use-case as been focused on real-time visualization, allowing user to quickly review and experience a dataset. What if we could use the available data and rendering power to reach the next level in visual quality and dedicate some rendering power to off-line rendering tasks, leveraging physically-based raytracing capabilities of VRED’s rendering engine and render realistic imagery and movies ?