

MAREK TRAWNY: I'm going to show you some nice things about our VRED server. First of all, I would like you guys to not take any pictures. I'm going to show a version which is in development and that reflects the current state. And I might select a safe harbor statement. I might talk about things which could change in the future without further notice. But anyways, what I'm saying is reflecting our current thinking and also our roadmap.

So first of all, I expect that some of you guys might have not heard details about VRED server. This is why I'm going to give you a quick overview about the technology. Basically, we see some very big challenges for OEMs in visualization. So VRED server is just the step towards resolving these challenges.

Challenge number one what we see is redundancy. So usually, if it comes, for example, to online configurators, marketing campaigns, you often do redundant work. So for example, you might have a web special being produced by one agency. You might print campaigns produced by another agency. And usually these agencies don't talk to each other.

You hand out some files, and for example the shaders set up for these different kind of projects are being made by the agency number one, agency number two. So this is really redundant work, since preparing shaders, for example, twice doesn't make any sense.

The next challenge we see is time. So usually, they're getting more and more models, more and more options of a car out every year. But at the same time, usually your team doesn't get bigger. So you face a real challenge in getting all your assets produced in time.

Next thing is consistency. This goes back to the first topic, redundancy. If you have shaders, for example, prepared by one agency for one campaign and by another agency for another campaign, are you always sure that the look and the feel of the car is really equal?

If it's that way, I guess you have done tons of review cycles, with each agency to make sure that the look and the feel off the car is really matching throughout different campaigns. Next challenge we see is complexity. So configuration options get more and more. Most of today's car configurators are still based on layout rendering technology, so everything's pre-rendered, different options.

And you layer everything one over another, and keeping track of all those different layers,

putting these over each other. And also, if there then is [INAUDIBLE] work being done at the model, like a facelift for example, you need to figure out which layers do I might need to re-render. And keeping track of all that is really a complex task.

And the next challenge we see is globalization. So with a global approach the OEMs have today of course, you want to make sure that your brand, your integrity, is being spread globally in the way you want it to be. And this is really big challenge-- working with agencies across the world, working for different campaigns across the world while keeping your brand integrity is key. I think this is also a big challenge for you guys.

In the end, it all sums up to cost. So if you would have infinite money, you could resolve all these challenges easily. But the main thing is, resolving these challenges and at the same time reducing costs. And this is where we are developing or placing VRED technology for these kind of use cases to make sure you keep track or you can address all your challenges while cutting down costs.

We do that with providing VRED technology as a base. So VRED server in that cases is a data center based system, which is leveraging the render engines, which you might already know from VRED professional. All the content you have is based on VRED data sets.

That means you create a single file for every single model. We call that a 150% model, where all the configuration options reside in that single file. That means you can address all work markets, left hand drive, right hand drive, all configurations, all colors, all geometry variations in that one single file, which then can be distributed to this kind of system. Then of course, you need the configuration options. And there we can connect to existing configuration logic systems to make sure that whatever is displayed on the website reflects actually a valid configuration of a car.

Next good thing about our technology is that it's nearly linear scalable. That means the more users you get on such kind of system, you can just add more computation nodes. So for example, if you want to have 100 users online, you can go with a certain number of nodes. So that really depends on quality, on resolution, things like that.

And if you say, oh, for a web special, no, I want to be able to double the amount of users which can access the system simultaneously, you just add double the nodes and you can be sure that you can manage double the users at the same time.

And another big benefit of our technology is that the front, and so the web pages the customer sees, they are running on any device. Because the technology is based on HTML5, JavaScript, CSS that can be displayed by every browser, every current browser. So you don't need to install any plug-ins or Shockwave or Flash what we know from the past.

So this is really running on any device. It can run on an iPad. It can run on the Android tablet. It can run on your Windows computer-- whatever. Then for the technology, we have different content generation types. So VRED server in the background running is able to generate output in different ways.

The first and easiest way is the pre-rendering. That means you can just throw a data set at the system. And from there, you can basically render millions of images, fully automated. So there would, for example, be you prepare your file. You prepare turntable animation.

Pass that file to the system. Go to a pre-rendering mode. And that will automatically create all variations of the car. That means all combinations of colors, rims, whatever. Put that to an image cache on a server. And it will be instantly available.

But of course, there are a lot of configurations available. And sometimes it would just be too much to pre-render everything. This is where rendering on demand steps in. So that would mean, for example, a situation where you have 80%, the most common configurations of a car already pre-rendered.

But if a user on a website interacts with the system and wants to have a special configuration, the system would recognize that, render that image on demand, display it for the end user, and then put that back to the cache. So if that special configuration gets requested again, it will already be there and not being re-computed.

And the last and most sophisticated content generation we have is streaming. That really means that we basically capture VRED pro viewport in the background on a server and we can stream that to any place in the world, which in that case means you have full interactive control of the scene.

You can move free in 3D space. Of course we can work on boundaries, saying, like, maybe you don't want to fly through the car, or things like that. So we can set boundaries in which you can move. But generally, this means you can really move in 3D space in a 3D environment on the website.

And getting that done-- so there had been workflows in the past where the product development was not really connected to marketing applications, for example. And this of course also gets back to redundant work. So things might have been already produced when the product was developed.

But these data is usually not being leveraged throughout the process in the way it could. And this is where our solution kicks in. So you can really reuse your data from design, take it over to engineering, take it over to marketing.

And the next slide is showing that in a little more detailed way. So basically, we can have a fully automated visualization backbone based on VRED technology throughout the creation process of the car. And as you have your car production-ready, you can reuse that data, put it on a server, and use that for your marketing content generation.

Now, talking about the different use cases we can address with our technology, the first one of course is online configurator. So that can be a set with pre-rendered images up to streaming. We can address point of sale installations that can be either run by a local workstation at the point of sale, driven by that workstation, for example, driving a disk play. But we can also drive that with VRED server, of course, with a stream being generated online on a server form and directly streamed to a dealer.

And last use case and what we have here is online collaboration and CGI tools. So this is really going to address needs in the design department where you maybe want to-- you're visualizing the car for a professional and you want to have a colleague check that out who is just across the world in a different department. You can upload it to our system. It will automatically generate a page, which looks like this.

And then you will have the option to connect different people to a 3D stream and at that time talk across countries, talk across continents with your colleagues based on a real time presentation. Of course, you can add annotations and things like that to leverage the idea behind a true collaboration.

And basically, the workflow in that case is of course everything starts with CAP data. You build a VRED virtual car model out of that. You can upload that to a VRED server data center, and then for example stream to online configuration websites. You can stream to point of sale installations, and you can also leverage the same kind of technology for your internal

collaboration and CGI creation tools. And this also leads to the next generation showroom, what we have in mind.

So basically, this is really a joint experience with different output channels so. You could have immersive presentations with your HMD. You can have big screen experiences controlled by an iPad. And you can have virtual reality and augmented reality experiences connected to one single showroom.

But how is all that working in detail? I want to talk a little about the structure of VRED server and how that's working. So our new generation of VRED server consists of two parts. One is what we call VRED server 2.0. So that's a Linux-based load balancer, which connects to all the computation nodes.

And for the management, we have a web-based platform which you can access to keep control of the whole installation. Developing that, we had some key features in mind which we want to make sure that this is developed in the best way. The one thing is cluster scaling. So having a cluster with maybe two or 10 nodes, that's an easy task.

But what do you do if you have thousands of nodes? You want to keep track of everything, you want to make sure everything runs, and you want to make sure that the scalability is really almost linear, as I said at the beginning. So we really put a close eye on the scaling and found a way in development where we can make sure that we have almost linear scalability throughout the whole system.

Then you want to make sure that when you put the system on maximum load, that everything's running fine, of course. So having thousands of nodes running in the system, all being accessed, all running at 100% CPU utilization. You want to make sure nothing crashes. And of course, this was key for us to make sure the software is rock-stable. Then you want a fault tolerance.

So for example, if you're running an online configurator based on the technology and one machine goes down, you want to make sure that the system is still working. So that's why we integrated redundancy to make sure that the availability of the system is at least 99.9%.

So basically, the different components of the system can be run in multiple instances, meaning that if one component, for example, goes down because of a hardware failure in the data center, there can be another component running the same instance, having access to the

same database, and making sure it can directly take over.

In that case, of course, what could happen is that the system in terms of response time gets a little slower. Because instead of maybe having two computers for the load balancing before, there might be only one left. But I think key in that situation is that the system remains working.

So this was the key. These were the key features we've been developing against. Then for the management console, our first goal was to make a clean and easy UI for your internal administration, people making sure that they can access the system, work with the system easily. Then we wanted to make sure to have all the management features in there, saying you want to create your clusters. You want to administer your data sets.

You want to administer users, and things like that. So that was also key to make it easy. Then it's all of course about security, so making sure data sets being uploaded, for example, to cloud instance need to be encrypted. They will be decrypted on the fly, on a node, with no keys being stored at any hard drive, for example.

So there can be security service in place of passing these keys to the system. So even if there would [INAUDIBLE] be hacking like your online cloud and getting the data set, everything's highly encrypted and those guys would need months or years of computation time to crack these files, making sure that your data's not being stolen.

And of course you want to have monitoring, since you always want to know what's going on the system if there any failures, how many images are you rendering, what about the utilization of the overall system. So this is being in our monitoring section. From example structure point of view, everything starts with a virtual garage, where all your VRED BPB files resist in.

From there, you put it on VRED server. VRED server would then connect to the render notes you have on your system and distribute the files. So in that case, they would really mean, for example, you have 100 nodes and you have 10 different car models available in your line. You can set the system up to fire up 10 instances of each car model, which is then already running in the cloud service.

If a user accesses the system, there's really no loading time. It would just be like forwarded to a node which is already holding the scene, making sure you can switch between different car files, for example instantly, And that we can make sure that there's no loading time being

experienced by a user.

Then we connect the management console and the deliverables which are generated by the system, which set that can be streams or images are passed through the management console by reverse proxying to the various front ends, making sure that, for example, no front end is-- please, no pictures-- that the front ends directly access the render nodes, so everything's highly secure through that reverse proxying.

And of course to the management console, we can connect third party components. In the first case here, I'm talking about configuration logic system, so most of the OEMs have logic systems already in place which define how a car can be ordered or not. So we can interface to these kind of systems, making sure that everything which is displayed on the website reflects a valid configuration of a car.

From a component standpoint, we're having the nodes as the base computation units. On top of that, we have VRED server. On top of that, we have the management system. And to a management system, we can have the different front ends connected. And with our new version, we completely redeveloped the front end and the management system. And we reworked VRED server and the nodes to make sure that we are on the latest state of technology possible.

The product behind that is the server and the nodes are simple install packages. The management system has a default configuration which is deployed. And for the front end, we have different kind of templates, which would reflect the default use cases. From a platform standpoint, the nodes can be either run on Linux or on Windows.

VRED server is a executable, which is running on Linux. The management system also has to run on Linux. And the front ends is set to make it available without plug-ins. It's based on HTML, CSS, JavaScript. But you could, if you would want to have a Flash application, addressing all system, that would also be possible.

And what's really key to the solution is VRED server basically is kind of an SDK with templating. And that always means this product always goes hand in hand with consulting. So it's really not an install-- I could hand over to you guys and you just install it and can work with it. So every OEM has different perspective addition systems.

Processes are different. So this is always connected to a consulting engagement. And in that

case, consulting would install and configure the server and the nodes for you. It would deploy and configure the management system, and consulting is there to build the dedicated front ends for you for the single use cases.

Talking a little about the Enterprise [INAUDIBLE] Workflow, because this is what we are addressing with VRED technology. Usually, everything is based on the source's [? CAT ?] data. So this is [? CAT ?] data. This is soft parts. This is material databases, for example. We're working on automated processes, how to create virtual garages out of these single sources as automated as possible.

From there on, you could put that data directly, for example, in an internal VRED server data center running internally at your premises. Of course, you would connect your product logic. And then you could run internal data approval. That's kind of these online collaboration tools, what I mentioned.

So after all the parts had been collected automatically, from the sources, you put that on a server and you can connect internally and have review sessions based on that and seeing, is the data correct? Is all the geometry there? Is the configuration logics working?

And of course, you can also connect to the internal VRED server data center to create online imagery for PR area, for example, or around collaboration for marketing campaigns in real time. Then what you would do is you could basically create assets. So these assets can be images, animations. And in terms of collaboration, this can also be knowledge which you just collect on the system and make it re-accessible at any point in time.

Then, from the approved data, you will end up at bunch of car files basically which is internally approved. So you can directly take this data, for example, hand it over to an agency. And that would mean that this file really contains all variants of the product, all the variance you want to give to the agency. You have your materials from internal material database in that, making sure that there's no redundancy anymore. So you can give that file to one agency.

You can give it to another agency. And if both agencies are working with VRED technology, you can make sure that the renderings being produced out of that content look equal, so there's no matching of materials anymore. You don't need to check if all the variants are there if geometries are correct. So this has already been done in the first step of the internal data approval. And after getting it out, you can really make sure that everything is consistent.

So I said this would result in a database for external partners. And the next big thing is of course that you then take the approved data, put it on your VRED server data center on the cloud or on premises. Again, connect your product logics and then can drive online configuration application areas on your website.

You can stream that to dealership point of sales. And of course, this is two systems. So one would be only accessible internally for your use cases, and the other one would be accessible from external that can be external trusted data center. This can be even a cloud installation.

Now I want to talk about some customers already using our technology. So we're having great success with Skoda. So they're using our technology for the web configurator as of today. They have around 10 different car models running on their website. And all the content is automatically produced by VRED server. So they'd have a DKM department, which means digital control model.

They built all the files for CEO, for C level reviews at a power wall. And after these files are approved, you can directly take the files from development, put that online, just maybe add some nice imagery for the backgrounds, and reuse that data for your online configurator.

And the good thing is that we're really reducing complexity here, since we're not rendering in layers. There will be a single image for every configuration of the car with everything in there like camera angle, which rim, so this basically generating like a key.

And with that key, you can directly-- that's usually the file name of the PNG image, for example, being created by the server. And with that name of the file, you can directly transfer that to a real configuration. So if you know the file name, you can say it's that car in that configuration, that angle, et cetera. So that makes it really easy to keep track of the complexity.

And of course, talking about updates, like facelifts of a car, usually if you're going layered rendering, I said you're introducing some new rims for example, you have to make sure, OK, I need to re-run some rims and they have to fit in there. But in that case, you just re-update the 3D model in VRED and [INAUDIBLE], load that back up to the server, and you will have instantly made sure that the facelift of the model is being uploaded in the correct state.

In that case, for example, Skoda is using a combination of pre-rendering and rendering on the mount. So they compute around 10 million images in advance on a big server farm. That takes

usually around two days. And then they have all the most common configurations of the car already available as pre-rendered images.

But as soon as a user interacts with the website and selects special options, for example one example here is special forklights you can order for the Skoda cars. These configurations are not being pre-rendered. So the system recognizes, OK, this image is not already there, sends a command to VRED server. VRED server on the background renders that image on the mount for the user. That gets passed to the website.

The complete process from the click on the website until the user sees the new configuration is about two seconds. And of course then the image for that special configuration is being stored in the cache, so if the next user goes online, make sure that the image is not being re-rendered again for the same configuration. And that whole environment at Skoda is being maintained by only four people.

So we have roughly 10 people in the DKM department preparing all the data. They prepare that data anyways for the internal reviews for the digital control model. And then the complete environment is just being maintained by four people. Everything's highly automated. And the same data is also being used at a point of sale.

So basically what they do is the same data they're uploading VRED server for online configurator, they're taking the data, giving that to local workstations at the dealership. And there's really redundant work anymore. So really, that streamlined process DKM data, use it for online configurator, and use it for a dealership point of sale.

In this last slide of my PowerPoint, I want to show a video of a customer statement running an online collaboration use case with us. And I hope that's working. I will try to run the video on that computer now. I have difficulties with my internet connection-- apologies. [INAUDIBLE].

So this was a lot of talking now. What I want to do now is show you the system life. So what I've prepared is just a simple demo front end for an online configuration system. So in that case, you just access a website.

And you have, in this example, two different car models to choose from. So I'm selecting the Polo. Now I'm in the pre-rendering and rendering on demand mode of the file.

So in that case, I want to for example change the color, change it to red, change it to black. And as you see, that means this configuration has not been rendered already. So we can set

an internal time frame of how long an image should render.

And if the image is not already existing, we say, OK, if you request an image which is not there, the node should render five seconds. And it will produce the best quality within that five seconds as possible and then pass that back to the front and make it instantly available. So as you see, I could also stripped that down to, for example, two seconds.

But then the quality would be a little worse, since of course there's ray tracing going on in the background. And the more time you spend on the ray tracing, the better the image will look in the end. Of course, we cannot only change colors. We have also different environments in here.

And as a simple example, if I now click on a configuration which has been rendered already, that's there instantly, and there's no more loading time for the user. Then, of course, what we can do is we can switch, for example, different lines of the car. So they're different trim levels, as you see now. We have a sunroof there. And the head line's changed.

And what we can also do is, as I was mentioning, we're talking about 150% models which hold all configuration options of a car. You see now I have a two-door version. And of course in the same file, I have the four-door version of that car available. And as you see, just a click, five seconds rendering and you see the corresponding change of the model.

Just clicking on different rims. Of course, this is what you know from other configurators. And you can have-- that one we already have-- we have different viewports as well, so that's pretty easy. And what we can do, of course, instantly switch to a different car. So that's also pre-rendered.

Now changing the color. I clicked on these colors before I started the presentation. So this is why they're there. I had not clicked on this rim, so again we're having now five seconds of computation happening on the background. And then we see the model.

Let me go back to the Polo. This is what we know from today's car configurators already from a user's perspective. What's key here is the process efficiency of how we get to this. But I guess what you have not seen yet at a dealer configurator, at an online car configurator is streaming.

So I just clicked a simple button and now I have node running in the background. And I can

really move through the system through a 3D data set live on a website. And of course, I have the same configuration options here.

And as you see, of course there is no the kind of five second rendering time involved anymore, since this is a real-time streaming which is happening in the background. So we really address a render node which is running in the background, directly send configuration commands to that node, and we'll instantly switch the configuration.

You might think, OK, that looks a little blurry-- not so perfect. So we have options, for example, to change the resolution. Now I changed the resolution from 800 by 450 to 720p. And as you see, it instantly gets better. And we're still streaming from online cloud.

Now switching even to full HD. And as you see in full HD, what we have here is you see some artifacts while I'm moving. This is an adaptive new streaming technology which was self-developed by us. So depending on the network bandwidth, it tries to find out the best possible interactive.

Or if you move in interactive space, it finds out how big the bandwidth is and adapts that to make sure that you have the best possible quality while streaming through environment. And as soon I leave the mouse, you see the streaming already then recognizes, OK, I'm not moving anymore and then refining the image from a streaming perspective.

But going back, for example, to that low resolution, what we can do is we can kick--

AUDIENCE: [INAUDIBLE]

MAREK TRAWNY: Yeah, that's running on one node. Actually what we have here is I have four sessions available. That means four people can simultaneously interact with one computation node. So later on, as soon as-- I want to run my presentation and then I have some question and answer in the end. And then you guys could take out your mobile. I'll get your IP address and you can enter that and check it out on your mobile.

So what we can also do is a technology which we implemented in VRED Professional. So basically everything we have in VRED Professional can be exposed through an API to VRED server. So basically, you could also run animations, things like that. So that's pretty easy.

But what I can now activate, for example, is real time [INAUDIBLE]. So as you see, the edges on the car, even though we are in a pretty low resolution of 800 by 450, if I disable it, you see

some more edges and can activate real-time [INAUDIBLE], which will then of course make the navigation a little slower. But as you see, the quality gets better in that case.

And it can also switch on higher resolution, of course, then go back to real-time [INAUDIBLE], enable it again. And that already looks pretty good, I guess. And as I said, you can instantly switch between different scenes. I can just now click here.

Currently, I'm running the Polo and I want to see the new Golf. Instantly there, instantly being assigned to different instance, running on a cloud. And now I can tumble the Polo. And of course here, I'm having the same options. I can go up with a resolution.

And I can even turn on the [INAUDIBLE] of the refinement, which you know from VRED Professional. So now moving through the scene, just leaving the mouse, it will first settle down on the streaming. And then it will start refining the image from a sampling perspective and ray tracing.

And of course, we have the different viewpoints here. In that case, I was flying through it since. But as a simple example, there are also animations in there. So we can set up camera angles in the way that we can have, like, fly animations. We could also integrate animations off the car itself, so making a door open, accessing the trunk, seeing different options of the trunk, with maybe different lighting configurations also in the interior.

And talking about interior, of course we also have interior viewpoints in here. So we can directly fly to the interior of the car, also explore that in real time. So I selected the viewpoint. Now I can move in real time, switch to a different viewpoint, fly there, explore the car in full 3D space, as mentioned.

And of course, I can switch back to the other car I had been accessing before. And as you see instantly, I'm on the other car again. So that was the front end presentation. Are there any questions up to here? Then I would continue to show you our management console.

OK, as said, this is a complex system. And you want to make sure that you can administer it as easy as possible. So if you access the system, in that case you come to our dashboard. On the dashboard you see different car files being currently available. So you see I have a Volkswagen Golf. I have a Volkswagen Polo running here.

I have, in that case, five instances of each car available for streaming sessions. And that little icon here means that currently, one of each instances is being used by online user. So as I

was here in the front end clicking both cars, it shows me, OK, there is a guide connected to both data sets.

Then we have some internal, which reflects the internal system, what's going on. So it says here the files are both started from a node perspective and the files are also running, means success. I could now just click Stop here. And if I click Stop, reload the front end, I would only have one car available.

And what you see here is an [INAUDIBLE], which just has the name the group render jobs. As said, we can leverage the system for different use cases. So if you have that system running, you can also just use your VRED Professional files, throw it at the system, and it will automatically create render jobs. So the system's not only there for running online configurators, point of sale installations, live collaboration. If the system is just not in use for some reason, you can just collect a bunch of nodes and have them connected to run your everyday render jobs.

From a setup perspective, you usually start with adding your scenes. So in that case, I would just edit the Polo. And see, in the Polo, I have the drawing on a certain node, a certain IP. So there, I could just switch. If you upload the scenes, you will have these scenes available here as a simple drop-down.

You can select the scene. You can give it a title. You could even give it a description. Here, I can upload a thumbnail, which is being displayed as access the configurator. You saw those images down at the bottom where it could switch between the car models. So this image will be leveraged there. Then I have different states.

So if it's published, the car is being available on the website. But we could also set it to a draft state. Then it would only be internally available, making sure that you before you put a car on your online configurator, of course you will make sure everything's working.

So you would set that to a draft mode. Then you could access it internally but not externally. To make sure that the car is really available from the outside, it has to be published and enabled. If that's not the case, you will not see it.

Then what we have is the different configurations of the car. Here, you can see the different options we have. For example, in the car paint variance, you see the variance we have here. In that case, for example, let's take the blue one. I will disabled the blue one, update the

variant.

Now I'm in the Polo, so if everything works out, I need to reload the web page, go to the Polo, in that case jump to streaming directly. Go here, and you see I just deactivated the black color, which had been there before. So going in there, small example, just enable it again, update it.

Going here, making the reload, taking the Polo, go to streaming, select the variant. And blue's back there again. So it's really easy to work with the system, making sure that all the variance are being shown in the way you want. So you can really switch on, switch off variants.

And currently, we're working also on an internal algorithm, which will parse all the data files internally and read out all the configurations and display that automatically. So you can even have-- you can of course render thumbnails, which you think look very good and represent your brand integrity in a perfect way. But what you can also do is have that automated. So basically, the system could generate these preview images automatically.

And as soon as I upload the VPB file to the system, it will be parsed in the background and all the different configuration options will already be there. And you can just go through the system and enable the configuration options you want to have available on the system.

And the same of course here is true for the environments. So we have different environments here. There are only, like, three are currently available. Just as the color, I could activate the other ones and that would work as well.

I have the different wheels here and of course the different viewpoints. So in that case, you can-- it wasn't mentioned before. In those different variants, so we have variants for animations, for geometries and materials, and for viewpoints. And you can create different groups here.

So I could create, for example, variant groups for rims, for colors. And then if the system is parsing the files, you can also rearrange that everything's in that group. And the groups are basically that what you see here on the front end. So you just name it. You put the variants in there. And then it will automatically reflect that.

From a setup perspective, what you need to do is after you have uploaded and created your scene with all the variants, you need to work on your nodes. In that case, I define 20 internal nodes. So you have a pretty good overview here. These are all the nodes currently running.

So I guess I have 20 nodes available.

And as you see here, the different nodes, they just have a name. You can say a node can stream. It can render snapshots. That means images. You can say a node can do both. A node can only do one of it. And here, I can just edit the node and then it can say, is that node enabled, can it stream, can it snapshot. So that's pretty easy.

And here also, that's just the range. So if you want to set everything up, you don't want to enter, I don't know, 1,000 IP addresses, for example. So you can just do ranges, for example. You just enter a range of IP addresses here. Then click what they are able to do. Click the Add Nodes and it will automatically internally create the configuration for that and update the system.

Then of course we have image repository. So there's internally that cache, which holds all the images. And for example, we have a media section over here. Go to Images. You see all the images which I just clicked, so everything which was-- you see all the preview images which are being shown for the variants. But you can also access all the single renderings which had been created by the system.

And what you can do of course, you can use scripts. So as you know, VRED has a Python interface. And you can automate things. You can make sure certain complex tasks are being managed by a script.

So you can just write a Python script, upload that here, assign that to a data set, which would mean in that case if the data set is being fired up in the background on a node, it would automatically execute a certain script, which for example, creates some variants automatically or does some reordering in the scene graph, whatever you might want to do. If you can script that with VRED API, you can just upload it here. It will automatically be executed. That can be [INAUDIBLE] to that extent to do automatic data preparation if everything's set up in the right way.

From the repositories, I'd like to talk a little about the infrastructure. So the base of everything are our API applications. So this is a unique ID and a key. And if you don't have that being passed to the front end, you will not be able to access it. So an API application basically reflects allowing a certain front end to connect to the back end.

And here in that client, of course, you have a certain VRED server which is there. You have

scenes where it's allowed to connect. You have the different environments. I will talk about the environments a little later and from the API applications. For example, this is one of these access tokens we can create. And sometimes you would want, for example, a web special for a car should only be online for a month.

So you can create a new token. Say that token's only valid for a month. Then the system will be running for a month and automatically after a month the token will not be valid anymore. And it will instantly be shut down. But you can of course extend the availability or just delete these kind of tokens.

Then in the environments, of course, you need to have the VRED server. So you basically just enter IP and a port where it's running. In that case, we have it running on AWS here in the US. It's just that IP address.

And here in the node groups, you can see, for example, as you have seen on the dashboard, I have the Golf, the Polo, and the render jobs. And if I now click for example that one here, it says, OK, you're not allowed to edit it since it's running. But if I go to the render nodes here, to the render jobs, I could just open it.

And now the gray ones are being used by the configurator environment now. So what I could do now is just click these notes here, check them. And then if I update it, these nodes will be automatically assigned to that group and making it available for my render jobs.

Then from an environment perspective, what we have is you do not want to have a single environment for everything. So if you imagine you have a big data center with thousands of nodes, you might want to say, OK, I want to have 500 nodes available for my online configurator.

I want 200 nodes for internal collaboration, and I want 100 nodes available for my internal staging, for example. So you can create different environments. And these environments are really encapsulated, which means from one environment you cannot get to another. They are all in different security areas.

So what you could simply do is you create in development environment, upload all your data sets there, check if everything's working. And if everything's working from an internal perspective, you can then just take the scenes you created you worked on internally and put that over to a production environment that is available from the outside.

Last thing I was just briefly mentioning it here, unfortunately I cannot live demo that to you now. We have that render job as a new feature in here. I can just create a render job. I would now take VPB file from my desktop, drag and drop it here. It will automatically upload the file. I will have it available.

And as you might know in VRED Professional, we have a sequencer module where you can define certain render tasks, for example, switch to a certain view, render something, switch to another view, do this, do that. And these are the sequences we can create.

And then it's pretty easy. You can just say, OK, I want to have sequence 1, 2, 3 created. And then you just would activate these sequences, create render job, and then you have a simple list where all the render jobs are in and everything will be just like one after another. It will run through your render jobs.

For example, in areas where a separate service not only for online configurators-- if you would have this kind of installation for online collaboration internally, of course people are only working eight hours per day. And as soon as they go home, you could switch the system from the online collaboration mode to render job mode and make sure that overnight it renders your images whatever you need to for your everyday work.

Of course, we also have some settings in here. So for the different node groups, we can say, should it produce JPEG images? Should it produce PNG images? What's the default resolution? What's the default render quality?

And a set here now in that example down here, the rendering is set to five seconds. So if I select a variant which has not been rendered already, it will use five seconds of computation power to render that image. I could put that down to 2. I put that up to 10.

Of course, 10 would result in a better quality. 2 would result in a lower quality. Then the system has internal mail functionality. So if something's going wrong, for example, you could shoot an email to the administrator. Or you can shoot an email if a render job is done to a certain user, for example.

And then there are dozens of parameters where you can set up the IP addresses and everything. So I don't want to get into detail, since this gets pretty boring. But basically, you have that settings area where you can manage all of the settings you want to use for that configurator.

Then for the administration, of course you can create your different users, roles, and grants. So you might have a super user, which is allowed to just do everything with the system. But of course, you might be in a situation where there's a certain back end user and he's just allowed to upload a VPB file and create a render job, but he's not allowed to change the node configuration, for example. So with grants and roles, we can define all that.

So you can first define different grants-- what they can do and what they can't. You can then sum that up to different roles. And then you can assign the roles to single back end users which you can create that you can really make sure that a single user who accesses the system has only that subset of features available which you want him to use.

And another thing is auditing. So what we have here, the system logs in the background everything what we do. So if, for example, user goes online and shuts down all the render nodes and your production environment is not working anymore, you can see which user logged on and who messed it up, basically. So that is, I guess, pretty important as soon as you get to production live environments.

And last but not least, and this is in development, so we're really working now heavy on all the monitoring features. But as you see, since the system is running, it has for example already done 55,000 different render jobs. No failed-- there's nothing what the system's doing now. Apologies, by the way-- this is German. We need to make this multi-language.

You see what's in the queue. You see the system load of itself. You can choose the pulling interval, things like that. So that's the monitoring section and that's really we're putting now detailed effort in that to make that bigger and make that better.

And as my system had a little issue with playing the video from the PowerPoint ahead before, now trying to do that here from that computer.

[BEGIN VIDEO PLAYBACK]

-Armstrong White is a company that specializes in 3D imagery, primarily for advertising agencies. Being from Birmingham, Michigan, we started off heavily in the automotive industry and now we're a global company. At Armstrong White, our clients don't necessarily think in 3D space. They maybe don't understand the technology the way our artists do who are working in it every day.

So we continually develop ways to communicate with them better. We invested not only in the VRED, but we worked closely with the consulting team and also invested in their programmers developing a user interface that's really simple to use. So it basically takes that complex product VRED and it simplifies it so anyone can use it for storyboarding and visualization.

So if you're an executive and you're traveling, you can log on from your cellphone or your iPad and quickly make decisions. You have product people now weighing in making quick decisions, account execs getting things to their client quicker. It's very beneficial and very fast.

In working with an advertising agency even a year ago, it could take a few days to get a render done. Using this AW viewer, the VRED technology, you're doing that in real time, instantaneously saving time and money. The collaboration that we've had with Autodesk Consulting, it's been very healthy.

Our programmers internally and Autodesk programmers communicate and we collaborate on what the next step is to this AW viewer. What used to take maybe a month to do a 360, an online configurator, now we're doing those sometimes in three days, with all the approvals. So it has greatly sped up time to market, which allows us to do more throughput.

[END VIDEO PLAYBACK]

So as said, the VRED server is a kind of software development kit what we're having. And it's not an out of the box solution. What you just saw now was a front end, which connects to our VRED server. So I showed you basically a front end of online configurator.

That was a basic collaboration, front end. And as said, these engagement always go hand in hand together with Autodesk Consulting. And this is why I asked my colleague Merton to come join me in the presentation.

He has worked together with Armstrong White to make this possible. And of course, this is not only available for Armstrong White. So this available for all our customers. And now I'd like to hand over to Merton from our consulting group showing the online collaboration front end for VRED server live running on his iPad.

MALE SPEAKER: Thank you, Marek. Can you hear me OK? So I just want to give you a-- first this is a good segue into what we're doing. So we worked very close, very intense with Armstrong White on this project. And since that project, we took that workflow into design, design review.

And that's what I want to show you. So you see an application here now. It's running on my iPad that's in this configuration used for design review-- mobile, web-based design review. So I want to walk you through. This is running in Amazon right now, so I'm connected through the AU internet.

I can, just with this regular orbit pinch to zoom navigation, manipulate the car, view on my iPad. It refines as I let go, so it's full ray tracing, full GI in the cloud. Single node is used here. And why this is so great for design, [INAUDIBLE] it's very similar. You saw all the benefits. You saw the quality of VRED server, the streaming.

So I'm not going to go too much into this. But in terms of workflow, so this is what our team does. We built the workflow on top of VRED server. Because when you get VRED server out of the box, you'll need to make it do something. So this is one of the example workflows that I'm showing here.

The nice thing is you're having a VRED scene file that you prepared for visualization and you're dropping it on this framework. You're making it accessible to anyone who has a browser or an iPad just by going to this URL. So I'm not having-- and it looks like an app, but this is actually Safari that's running here. Safari on the tablet-- it dynamically extracts all the variants, which is huge for design review, because it has to go fast.

So what you saw before in the VRED server [INAUDIBLE] interface for setting everything up manually, is something we're circumventing here. You don't need to do that. It's just grabbing the scene file. It's looking at the options, so it found paint options. You see calipers, license plate, rims.

And if I go back to the paint, they come directly from the data set. They extract the dynamic [INAUDIBLE] start, flip my model around here. Same with the environment, so there's a section, a tab for the different environments it found in the scene file. They're automatically showing up. So I can put that car on the beach, for example.

There's a section for camera views, predefined camera views from the data set. So again, all data-driven and I can directly drive into certain views, like the light close up. All these views, I can still manipulate so they're always live. So very nice, easy to use interface for design review.

One feature that's really important, it goes into synthetic photography. So that's already

something we started implementing for Armstrong White. But especially for design review, when you want to set up your step by step, kind of like a slide deck, what you want to talk about and walk through, you can define snapshots and angles with configurations.

And now I click on that camera icon, and it adds to my snapshot list. So now when I go to the very last icon here, you see all the different snapshots I took before. So you can set this up for design review before the review. And then you just step through it step by step. So if I go on the first image, I have an image. I have a snapshot of the first one.

However, it's interactive. So if somebody asks me to zoom into the light, I can just zoom in or orbit around it any time. So that's a nice, powerful workflow feature. And I just wanted to show you that workflow and see what this is on a user side, live demo. Yeah, that's pretty much it on that project.

AUDIENCE: [INAUDIBLE].

MAREK TRAWNY: It's a combination of things. So what I did here, first of all, the resolution of the iPad is fairly small. I'm not running full HD. I can slide this over here. You see I actually have a resolution configuration. I'm running at 70% right now, just because of internet performance primarily, but also ray tracing. And it ray traces only 70% of what you see and then blows it up on the client side. So there's a bunch of things going on here. Obviously it's a tradeoff.

I forgot to mention communication. Since collaboration is a big thing, I have an option to invite other people to this streaming session. And with the same session on the server, the same node, I can have multiple people pointing their browsers, their iPads to that same session. And they see the same stream that I have. Any more questions?

AUDIENCE: [INAUDIBLE]?

MAREK TRAWNY: It's depending on your connectivity. And so we've so far for the use cases, we've kind of limited to it four to five right now, just to not overload the streams. But that works fine.

AUDIENCE: [INAUDIBLE]?

MALE SPEAKER: Running--

AUDIENCE: [INAUDIBLE].

MALE SPEAKER: So currently, it's running two scenes, five session each. And the Amazon instance is about \$2 an hour. This is highly adaptable. So it really depends on what resolution do you want, what render qualities, et cetera.

So we can, in a customer project, we would sit down with a customer, drag all the sliders from left to right and really find out what the customer wants from a quality point of view. And then we can say, if you want that quality, it costs that amount of money, roughly. But we can also go from a different perspective, that the customer says, I want to have roughly these costs per hour. What can I get for it?

And then we can do different things, like you could have a low resolution but high rendering quality. You could have high resolution but lower rendering quality. So this is really also part of the project with a customer that we sit together and find out what the actual needs of the customers are and--

MAREK TRAWNY: Yeah?

AUDIENCE: [INAUDIBLE].

MAREK TRAWNY: Data preparation question.

AUDIENCE: [INAUDIBLE].

MALE SPEAKER: It depends on the complexity of the car model. So if you say, I want the full 150% model with all world model options, left hand, right hand drive, everything can take, depending on the skills of the operator, maybe two weeks for really full-blown model, then also with adding materials. But as soon as you have your material library ready, and you just need to assign it, you can strip that down.

And for example, if we're talking for a web special where you say, I have that car. I maybe have two trim levels, five rims, 10 colors. I know people who can do that in a couple of days-- one two, three, four days, depending on the skills.

MAREK TRAWNY: And again, by coming to design review, that's obviously it needs to be really fast, faster than for sales and marketing. So I'll be talking at 1:00 about some workflows that make it on a button click, some tools that we developed, some workflow tools based on the shader assignment automatically, variant creation automatically to make this really within a couple of

clicks. But that's when you come from design tools. When you come from heavy CAD data, it's going to be a little more complicated. Any more questions?

AUDIENCE: [INAUDIBLE].

MALE SPEAKER: Yes, they have to be dedicated to VRED server. But there are technologies of other companies like Grip Engine, for example. And with that kind of technology, you could basically from a lower level standpoint define when, which nodes are available for which use cases. And as soon as the grid engine says, now we have VRED server running here, could then automatically create IP commands, sending it over and saying, no, I have this IP available.

And together with some simple development work from our side, we could connect to that. And as soon as their IPs added or removed, we could put that into the system and then make sure that you have more or less computationals available. That's a little tricky, but we could make that work.

AUDIENCE: [INAUDIBLE].

MAREK TRAWNY: This is now everything what you saw is the running on the CPU. So that's ray tracing. But if you have nodes, render nodes which have a graphics port in there, we're currently working on making also OpenGL streams from the graphics ports available.

AUDIENCE: [INAUDIBLE]?

MAREK TRAWNY: Excuse me?

AUDIENCE: [INAUDIBLE]?

MAREK TRAWNY: I'm really not allowed to give you time frames. I can just say we're working on that, and it will be more short-term than long-term. It's just a matter of-- if you want to run a project with us and say, that's my main thing-- now I need that, we might be able to make it work faster.

AUDIENCE: [INAUDIBLE]?

MAREK TRAWNY: Good question. I guess my holiday next year is something I call mid-term.

AUDIENCE: [INAUDIBLE].?

MAREK TRAWNY: Any more questions? And of course, we're around here. So if anybody wants to have a more detailed conversation, wherever you see me or if you want to have a business card to get in contact with me at any point in time later on, feel free to approach me. I'd be glad if I can help.

MALE SPEAKER: And same for me, of course.

MAREK TRAWNY: Thank you.