Hi everyone - I am Neil Trevett and I work on developer ecosystems at NVIDIA and I’m President of the Khronos Group. This presentation will give you the latest updates on Khronos open standards used for building Virtual and Augmented Reality - or XR experiences - and how they work together with the larger constellation of standards throughout the industry that will enable us to eventually build the dream of the metaverse.
So let start by introducing Khronos if you haven’t come across the group before. We are an open standards organization, focused on creating interoperability standards to enable applications to access the power of silicon acceleration for domains such as 3D graphics, virtual and augmented reality, parallel computation and increasingly vision and inferencing acceleration. We are a non-profit organization and all the standards we create are open and royalty-free for the industry to use. We have almost 160 members now - everyone from NVIDIA, Valve and Google down to small startup companies - spread across the world. We are open to any company who wishes to join to have a voice and vote in how any of the Khronos standards evolve.
Here are the most active standards currently being developed by Khronos today. There are four main groupings - 3D acceleration APIs including Vulkan, OpenGL and WebGL, 3D Asset Formats including glTF, the OpenXR API standard for portable augmented and virtual reality and lastly APIs and languages for parallel computation. I am going to focus today on the highlighted standards that are most relevant to this VR and AR audience - starting with Vulkan...
Vulkan is a new generation 3D API that provides low-level, explicit control over GPU hardware for the highest performance and lowest latency. Vulkan is unique in being the only new generation GPU API that is an open standard - and so is not restricted to ship only one vendor’s platforms. Vulkan is now being shipped by every GPU vendor, is being adopted by the leading gaming and application engines and is available on almost every platform including desktops, mobile, consoles, embedded systems as well as the cloud.

Although native Vulkan drivers are not possible on macOS and iOS - there are layering solutions that enable Vulkan applications to run on Apple platforms.

So - if you want to develop the highest performance, lowest latency XR applications - AND you want to deploy your application portably across almost any system - then Vulkan is the 3D API for you.
Vulkan 1.2 recently launched in January 2020 and integrated a number of proven extensions including timeline semaphores to enable easier application multi-threading. The key roadmap discussions post-1.2 are on the right - and include Ray Tracing, Variable Rate Shading and hardware accelerated video encode / decode. For the XR community, Variable Rate Shading will be particularly valuable as it enables a concentration of rendering power where the user is looking in eye tracked XR systems - to increase performance and reduce power consumption. All this new functionality will ship first as extensions - and those that get a positive reception from the developer community are strong candidates for integration into the next-generation core specification.
In fact the first item on the roadmap - Vulkan Ray Tracing - has already just shipped in March 2020 as a set of extensions that seamlessly integrate ray tracing into the existing Vulkan framework. This significant milestone is the industry’s first open, cross-vendor, cross-platform standard for ray tracing acceleration - and can be widely deployed either using existing GPU compute or dedicated ray tracing cores.

Many developers when first encountering ray tracing regard it as a monolithic all-or-nothing technique - but in fact ray tracing is a very flexible technology that can be used to incrementally add visual realism to traditionally rendered scenes. Ray tracing at its essence is calculating how light rays intersect with objects, materials and light sources. And you can use that basic idea to fire rays to implement a variety of visual techniques - a selection of which are shown here.
Vulkan Ray Tracing provides the choice of two methods of firing rays into a scene, to calculate ray intersections and undertake flexible, programmable calculations at those intersections.

The first method uses a completely new Ray Tracing Pipeline which enables the application to fire a grid of rays into the scene - and select a shader to run at each intersection. A lot of the housekeeping around ray management is handled automatically by the machinery defined by the Ray Tracing Pipeline.

The second method, Ray Queries, enables programmers to take more manual control over how they cast rays, by enabling any standard vertex or fragment shader to fire single rays into the scene and handle any returned intersection data however they wish.
Next I want to talk about OpenXR, which is Khronos’ open standard for cross-vendor, portable augmented and virtual reality. OpenXR defines a common API that all XR hardware runtimes can expose to application and engine developers. This enables engines and applications to be written once - and run on any XR hardware that supports the OpenXR API. As well as providing portability, OpenXR has enabled the industry to come together to learn from first generation APIs and to collaborate to create a new generation XR API with cutting-edge capabilities. OpenXR enjoys the support of a who’s who of the leading AR and VR companies throughout the industry.
OpenXR is early in its rollout - but an increasing number of implementations are beginning to appear. The version 1.0 specification was publicly released at SIGGRAPH 2019 and was supported from day one on Microsoft HoloLens 2 and the Monado open source XR engine from COLLABORA that runs on any OpenHMD-compatible hardware. Since then, OpenXR has shipped on Microsoft’s Mixed Reality headsets and the Oculus Rift and Quest HMDs and is available as a plug-in to the Unreal engine from Epic - PLUS many more implementations are in development. In parallel, the OpenXR Working Group is finalizing the APIs conformance test suite that will ensure consistent portability across multiple OpenXR implementations as they roll out across the industry.
I think the reason that OpenXR is gaining such wide industry adoption is that it’s a win-win for all involved. Software developers, or ISVs, win because they can write an application once and ship on multiple hardware platforms without re-porting - both saving time and money AND reaching a much larger audience. XR hardware vendors win if they expose the OpenXR APIs on their platform as they have access to a much larger library of OpenXR-compatible applications. And most importantly - end-users win as they will know that the XR applications that they want to run will be compatible with the system they have purchased - growing confidence - and the XR market for everyone
OpenXR contains everything an application needs to drive the XR devices in a system including device discovery, event processing, sensor tracking and pose calculation on the input side, and frame display timing and composition, plus haptics control on the output side. The only thing that OpenXR doesn’t have is 3D rendering - and so an OpenXR application will use a rendering API, such as Vulkan, to generate imagery. OpenXR can be used with any 3D API, but a new generation API such as Vulkan is particularly well suited to create applications with high rendering performance and the low latency that are so vital for a compelling XR experience.
So far, we have talked about Vulkan and OpenXR which are both native APIs. However, the metaverse of openly connected and searchable virtual spaces and assets is likely to evolve out of the World Wide Web - and so it is critical that the industry also bring XR capability to the browser. The good news is that Khronos and the W3C World Wide Web Consortium are cooperating to bring 3D and XR capability into the Web stack.

All 3D in the browser today uses WebGL, a Khronos standard that defines a JavaScript binding over native 3D APIs and which is used by popular 3D web engines such as three.js and Babylon. Now we also have the OpenXR native API exposing XR hardware capabilities - and Khronos is working closely with the WebXR working group at W3C to bring that capability into the browser. So just as native XR apps can use Vulkan and OpenXR together - XR apps in the Web can use WebGL and WebXR together - with the web APIs reaching down into the Khronos native APIs for acceleration and hardware portability.
As well as providing application portability between different HMD devices - OpenXR will also provide portability as AR is deployed on new platform architectures - such as Edge Servers. One architecture being developed uses 5G to connect a lightweight AR device to send it’s real-time sensor data to an Edge server - such as an NVIDIA EGX platform - that processes the sensor data, maybe using inferencing, and generates graphical augmentations - that can be much more sophisticated than could be generated on a battery powered device - and that are sent back to the device for display. The OpenXR API can hide these round trips over 5G from the application - enabling the same AR application to run natively on an AR device - or an AR device connected to an edge server.
Every XR application needs to access and display 3D scenes and objects - and so I would like now to talk about glTF - the Khronos 3D asset format that is focused on being ubiquitous, compact and easy to process - and so we like to call it the ‘JPEG of 3D’ as it is the last mile JPEG-equivalent that lets you easily transport and deploy and use your 3D assets wherever you need.

glTF uses JSON to describe the 3D scene or asset hierarchy and uses binary payloads to describe data-heavy geometry, animations and skins. Textures are included typically as JPEG or PNG files. The latest version of glTF - glTF 2.0 - has PBR - or Physically-based Rendered materials that use textures to map how metallic or rough materials are, or how specular and glossy. This results in very realistic looking objects that can still be rendered very efficiently - even on mobile devices.
Support for glTF is now widespread throughout the industry as companies realize the efficiency and cost savings of having tools, apps and engines that can all communicate assets easily to each other. The companies shown here have all implemented glTF support - both in tools that generate glTF assets and engines and applications that ingest those glTF models. If your company supports glTF - and your logo isn’t here please let us know - we would love to add it!
glTF is evolving to meet industry needs - and is very close to finalizing glTF Universal Textures that use Basis Universal royalty-free encoding and transcoding from Binomial - with a KTX2 wrapper to provide consistent cross-vendor access to texture assets such as mip map levels. This enables smaller than JPEG textures that are directly transcodable to pretty much any GPU accelerated texture format on the fly - so textures never need be stored as fully decompressed images on the GPU.

Next up on the glTF roadmap is second generation physically-based rendering - or PBR - that takes us beyond glTF’s current Metallic/Roughness Specular/Glossy model to include properties such as clearcoat and subsurface scattering. This will start shipping soon as a series of extensions starting with clear coat. A shout here out to both Dassault and the NVIDIA MDL teams that initially guided the group to a sweet spot between visual realism and widespread deployability - and now we have wide industry engagement from Adobe, Autodesk and many others to agree on a ubiquitous PBR format that will work for everyone. Please give us your input - a lot of discussion is on the glTF GitHub if you are not a Khronos member.

And now, due to glTF’s success - there are lots of incoming feature requests - and we would love to get your input and requirements there too. The glTF working group now has the interesting task of finding the right balance between adding more functionality - but more importantly keeping glTF focused on its primary mission as a ubiquitous 3D delivery format.
As a final topic today, I would like to talk about the most recent initiative at Khronos that will positively impact the XR industry - the recently formed 3D Commerce Working Group. The basic technology to use AR in online retail has been around for many years - for example the IKEA AR furniture shopping app shipped back in 2013 - and such pilot projects showed that AR was indeed an effective technology for driving customer engagement and increased sales. So - why isn’t 3D Commerce more ubiquitous today?
Many in the 3D Commerce industry have come to realize that, in addition to having the right acceleration and portability APIs, the main barrier to rolling out 3D Commerce out at industrial scale is the process friction around 100s of companies trying to cooperate over the design, manufacture and presentation of millions of products across multiple platforms. If you talk to online retailers trying to use 3D models - there are so many horror stories of process incompatibilities - from the complex down to seemingly trivial issues such as that no-one can agree what direction is UP for 3D models. Which does sound trivial until you think what that’s means for a retail web site trying to import 1000s of 3D models a day.
The desire to solve these 3D Commerce process issues prompted a number of retail companies such as IKEA, Amazon and Target to approach Khronos as a safe space to cooperate - and where there is already considerable 3D expertise - to create standards and guidelines to enable 3D Commerce at industrial scale. We now have a strong quorum of participants including a unique mix of retail, technology and platform companies working together to truly enable the widespread use of 3D in retail.
The 3D Commerce Working Group is working to ship initial guidelines and standards this year solving the first round of urgent process issues around creating and configuring assets - and ensuring that they can be faithfully displayed with sufficient visual realism - regardless of the end user’s device. This work leverages glTF and is inspiring glTF’s evolution - including glTF’s new generation PBR model to accurately display a wide of products - and glTF’s handing of advanced metadata.

The streamlining of the creation and deployment of 3D assets at scale will not only enable 3D Commerce - but will have a positive impact on deploying all types of virtual and augmented reality applications and solutions.
I hope you have found this update interesting and useful! If any of these Khronos standards are relevant to your business - any company is welcome to join Khronos to have a voice and a vote in their evolution. Annual memberships start at $3,500 for smaller companies - and if that is too much - you can apply to join one the Advisory Panels that provide requirements and specification feedback to the working groups themselves. I have provided some links here to more information - and if you have any questions or we can assist in any way - please do not hesitate to reach out to me here or on email or Twitter. Thank you!