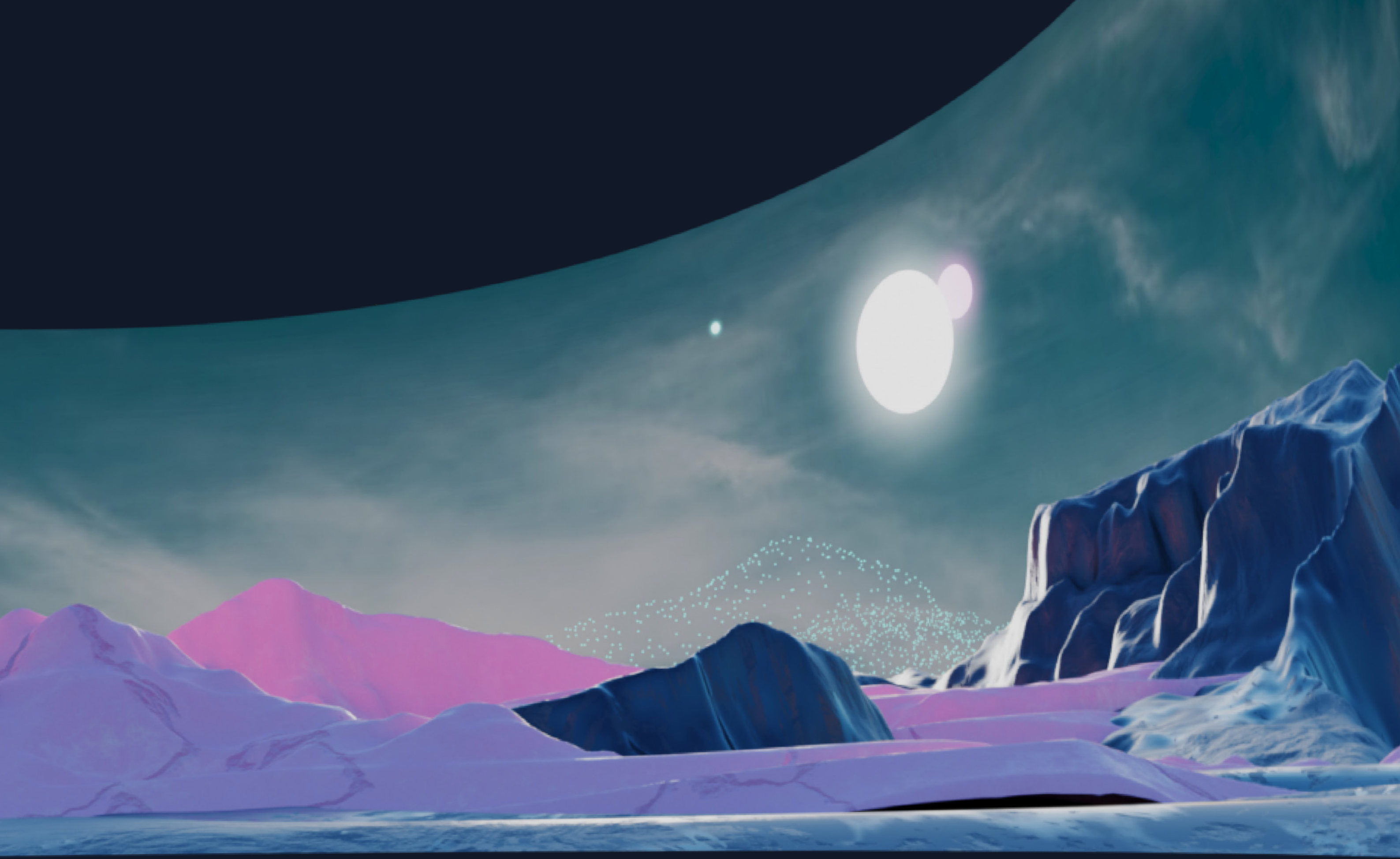
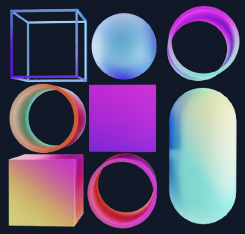
DISGUISE

The Key to Superior Content Rendering

How the Disguise architecture with RenderStream™ works together to maintain ultimate frame accuracy and minimal latency for any content imaginable.



Foreword



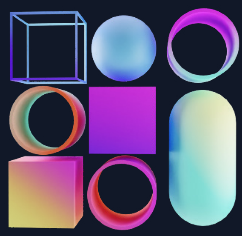
Over the past few years, the overall speed of GPUs has increased - allowing for low latency and high-quality content rendering. At the same time, game engines have ushered in a new era of graphics, allowing film and episodic TV productions to use real-time content rendered on LED volumes instead of green screens. This has unlocked greater collaboration opportunities on sets and reduced the need for on-location filming as well as the building of multiple hard sets.

Correct image generation where multiple systems and render nodes are involved has always been a major challenge. With productions requiring larger canvases and increased visual fidelity, we needed to find a way to seamlessly scale content. First, we provided a whole server dedicated to the job, allowing for maximised efficiency. Then we thought...what if we could do even better, and dedicate MORE than one server to the job? Thus RenderStream™ and the current Disguise system architecture was born.

Here, we explore the Disguise architecture and RenderStream™ and how these technologies minimise latency and allow for increased flexibility for in-camera VFX workflows (ICVFX), infinitely scalable content, and superior system efficiency - all while removing as many limits on performance as possible and opening up new creative possibilities for all.

Tom Whittock
Software Architect
Disguise

Architecture overview



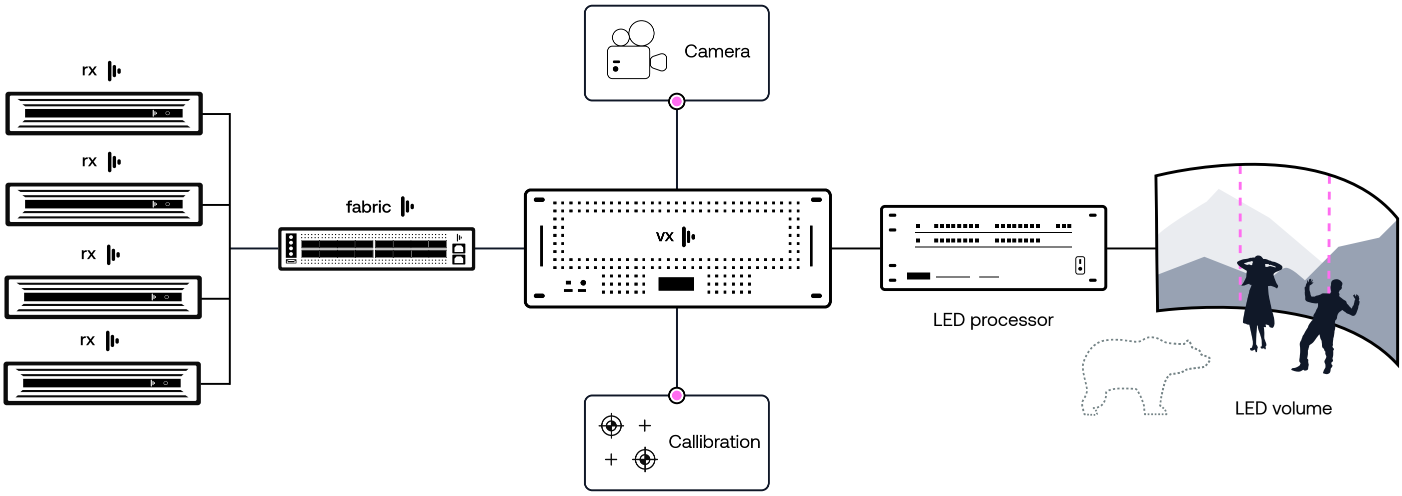
Disguise's architecture

Rooted in a 20+ year history of powering the content behind the world's most spectacular live events, our software and hardware platform, paired with its recently launched creative services, enable creatives and technologists to make their wildest vision a reality.

Our experience in image and projection mapping, large format video playback and working with associated technologies such as timecode, genlock and output to LED displays, provides us with the knowledge and expertise to make architectural choices that create the ultimate setup for accurate and seamless content rendering on an agnostic platform.

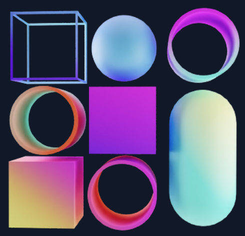
The Disguise architecture consists of media servers acting as compositors (which are connected to display devices - eg. Disguise VX and GX server ranges), and decoupled rendering nodes (eg. the Disguise RX range).

Communication between the rendering node(s) and the compositor is via a unique, open source infrastructure called RenderStream™. The rendering nodes can work as individual render instances or be combined together through RenderStream™ to enable a capability known as cluster rendering for in-camera visual effects (ICVFX) or extended reality (xR).



Render nodes can be used to scale content endlessly - rendering them across different surfaces including projection canvases and LED screens.

Architecture overview



What is RenderStream™?

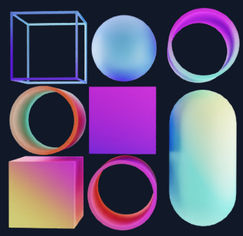
RenderStream™ is a bi-directional protocol for the transport of rendering information. This includes metadata associated with render requests (camera, scene parameters, texture information) and of course render buffers. This is used for communicating render requests to a render node(s) hosted render engine and receiving the resulting buffer information back from those render nodes. RenderStream™ can be implemented in any computer graphics/rendering application with the use of the RenderStream SDK. Example implementations are available for Unreal Engine and Unity on Disguise's Developer portal along with the full documentation.

Get the SDK

What is cluster rendering?

Cluster rendering is a robust system enabling you to render pieces of a single scene across multiple computers and then stitch those pieces back together to create a cohesive, high-resolution image or video. This is imperative when creating the most demanding photorealistic content for the virtual sets used in ICVFX. Typically, a single GPU for one render node is not enough to satisfy rendering needs, so cluster rendering allows you to add many render nodes and scale productions as needed. You can cluster by splitting up the render canvas, choosing different render nodes for different objects or, only with Disguise, cluster via different plates allowing you to render frontplate and backplate visuals on different nodes. With Disguise RenderStream™ you can scale up frustums, and so, for example, you can choose to allocate most of your rendering power to the inner frustum which will be captured in-camera.

Benefits of RenderStream™

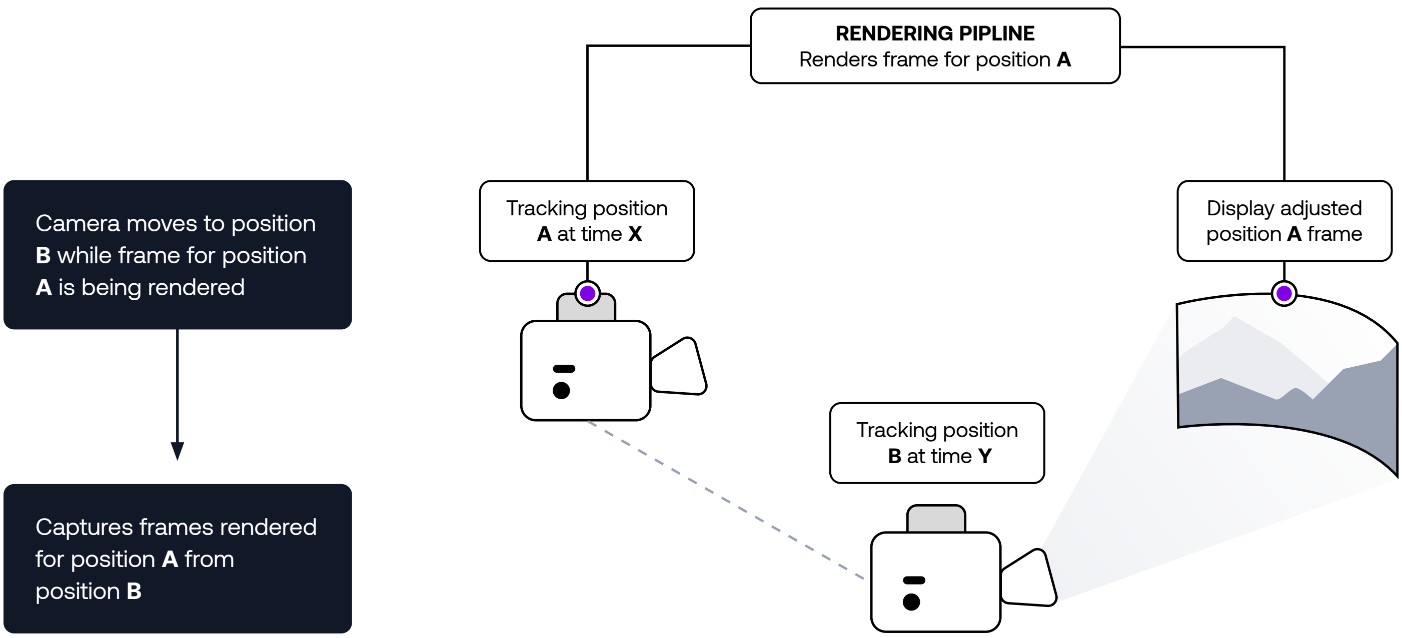


1. Managing actual latency

In order for real-time content to be used in film and episodic production, it needs to be photorealistic with low latency so cinematic quality can be preserved. Latency is the delay from when a signal is sent to when it is received/displayed at its destination. This delay is a function of the combined system latency of the camera tracking system, network latency, the rendering pipeline, the LED processing pipeline and the camera capture routine.

Whilst Disguise has strategies to manage these latencies, this document is focused on the render pipeline portion of this round-trip latency. While interacting between the real and virtual world, these latency gaps and synchronicity gaps become more noticeable when something in the physical world changes (for example, a camera moves) and there is a delay in reaction from the virtual world.

RenderStream™ manages latency by allowing each render node to receive the render request individually, meaning there is no requirement for the render nodes to render the frame at precisely the same time as each other across the cluster because the compositor is able to stitch the frames back together. This makes a RenderStream™ cluster much more resilient to network latency variations, as well as allowing each render node to only focus on rendering with far reduced synchronisation overhead. Simultaneously, this avoids the tendency for video artefacts or tearing, which is what happens when the tracking data is not received at precisely the same time by all render nodes in alternative implementations.



Benefits of RenderStream™

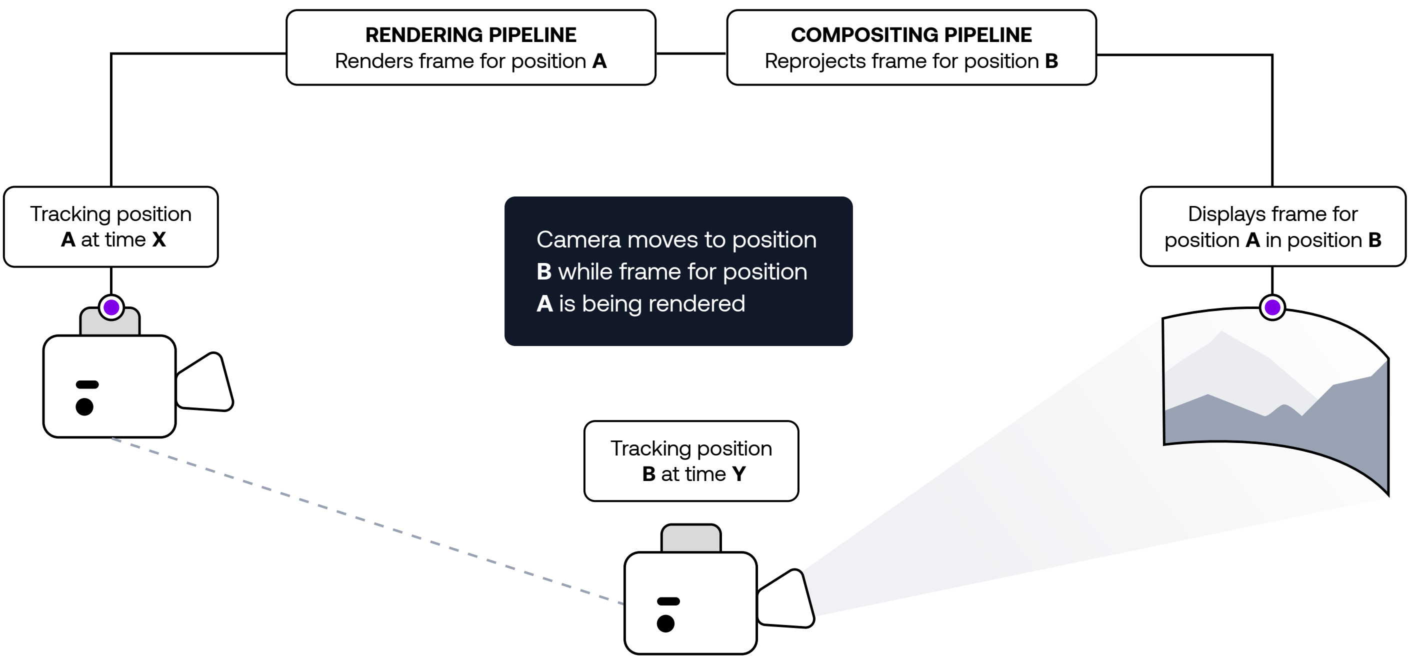
2. Managing perceived latency

Most virtual production solutions have mechanisms for overcoming actual latency, however, Disguise is the only solution on the market to overcome perceived latency through temporospatial compensation. With signal pathways within extended reality (xR) and virtual production still being largely referenced/synchronised on frame intervals or boundaries, most are often working with information that is temporarily out of step, meaning that, by the time the camera tracking data is received and processed by the render node, the camera has since moved on to another position in space.

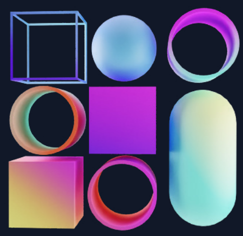
There is a circular dependency between all these processes and they are all working on their current best guesses based on recent information. This is often thought of as an impossible-to-solve problem.

Through RenderStream™, the Disguise architecture affords a complete view of the system state, therefore, allowing for minor adjustments to be made within the image processing pipeline - making full use of all the information to build a better result.

For example, the tracking data we receive is where the camera was at time point X, not when the tracking data is processed at time point Y. To avoid additional latency, we must request the frame render for this camera position as soon as possible. Additionally, as we have access to data in both time and space domains, we can build in some predictive capability, and use the more up-to-date information we receive after the event. We can therefore fix the image so it appears in the right time and space - allowing for a more accurate match to the camera movement itself.



Benefits of RenderStream™

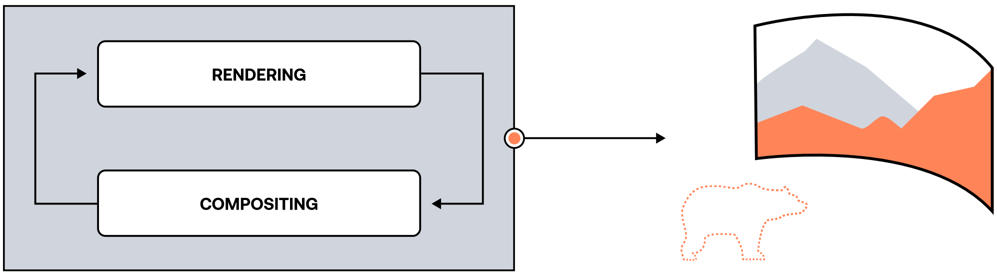


3. Near-linear scalability of content

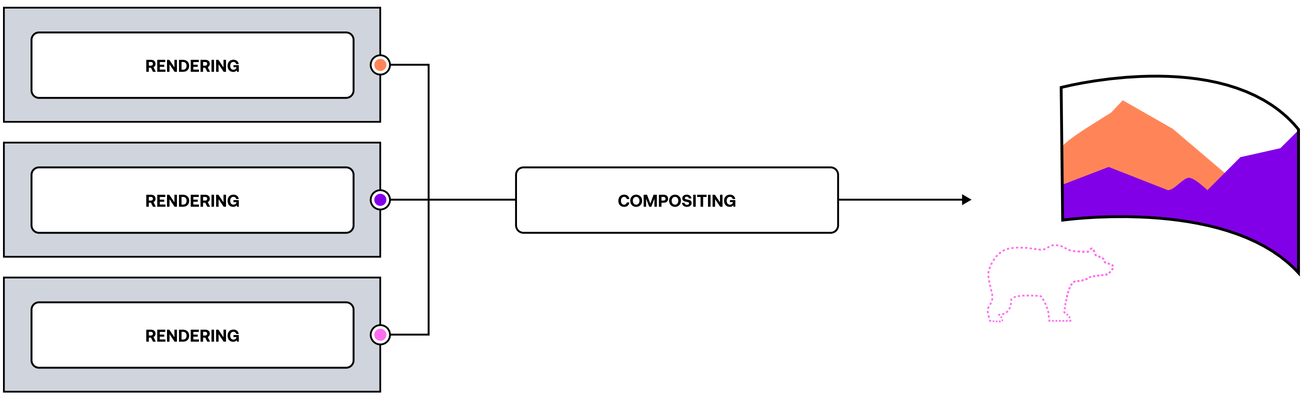
Scalability is the most immediately obvious benefit of the Disguise architecture. One of the key aims of RenderStream™ is to enable the scaling of render power beyond the power of a single GPU for a given amount of outputs. That means any content render is now within reach, no matter the scale - crucial for enabling cinematic content for films and episodics. Without RenderStream™, there is ultimately a ceiling beyond which further scalability is not possible without radically changing the deployment architecture to accommodate additional LED processors and servers.

Rather than this “big metal” approach, RenderStream™ separates the rendering portion of the frame generation from the physical LED outputs and uses IP streams to interconnect the systems, which uniquely enables near-linear scaling of content, meaning, essentially, that as your render farm gets larger, you can render increasingly complex scenes whilst maintaining a stable frame rate. The approach of splitting 3D rendering and compositing to separate servers results in much greater performance because it reduces resource contention and removes context-switching overhead.

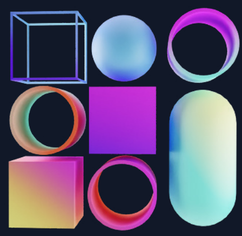
Option 1: Without RenderStream™



Option 2: With RenderStream™



Benefits of RenderStream™



Context switching overhead occurs due to competing types of tasks being run as quickly as possible. An example of this is when turning the 3D model into a 2D image to be displayed on a surface. This places a lot of demand on the processing power of the server. On the other hand, the different images are simultaneously composited together to make the final image rendered to the screen. This is less demanding but needs to be done as quickly as possible.

If you split those two different types of tasks to different servers then the servers can be specified for what is most performant for that type of job. It also means that rendering is not being slowed down by the demands of compositing and vice versa.

For explanation, one comparison is multitasking in the human brain. If you struggle to read and listen at the same time, it is because the written text is processed by the same part of the brain as the spoken text. The brain is therefore not optimised for either task and will do both poorly. If you, however, listen while doing something manually, stimulating the motor cortex, such as housework, the result will be more successful as parts of your brain controlling muscle movement are optimised for that while the language processing brain functions can listen successfully.

Splitting rendering and compositing allows for increased scalability. You can, for example, scale the rendering part if you have a very demanding set of Unreal Engine scenes, without having to scale the compositing part.

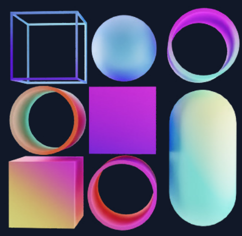
This is to be expected, because these two tasks have opposing demands, with 3D rendering being computationally heavy, and composition being computationally lighter but more latency sensitive. In addition, the approach has the added practical benefit of not needing many permutations of server models.

With this, productions can bring the most demanding and render-intensive content to principal photography and confidently shoot the scene they require. Some Disguise-powered stages such as CJ ENM in Seoul, South Korea, are running dozens or more rendering nodes simultaneously to power the most pixel-rich content to be rendered in real-time.

This technique has also proven especially useful for large-scale immersive experiences such as the Illuminarium venue in Atlanta and Vegas, where 20 Disguise VX 4 servers and 45 RX II render nodes drive immersive projection-mapped real-time content on high-resolution canvases.

Learn more

Benefits of RenderStream™



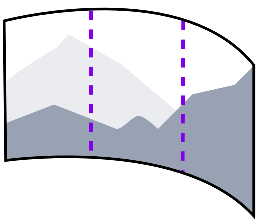
4. Flexibility for any type of shot

Flexibility to assign computing resources to the most demanding aspects of a scene is a major benefit of the Disguise system architecture. Each render instance can be divided and segmented according to whatever allocation rules we define.

Additionally, the typical RenderStream™ integration with content engines uses a virtual camera in the 3D scene. The mappings to the LED volume are handled in Disguise's Designer software, which enables the pre-visualisation and output of content to the LED.

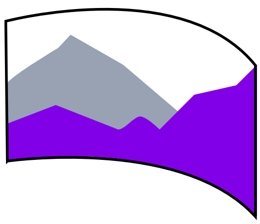
An example of this is Designer's mesh mapping feature, commonly used in VP, which is aware of the surface it is mapping to and allows the creation of a perspective render from an eye point.

This means that any camera-related settings which affect the scene composition in the content engine, such as aperture or field of view, can be applied regardless of whether the view is for the inner or outer frustum. This flexibility is crucial for using virtual sets for the creation of cinematic content.



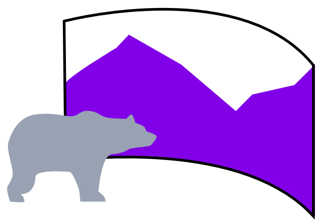
Cluster by splitting up the canvas

Split across the canvas, choosing how many render nodes to distribute each slice of content to.



Cluster by ICVFX object

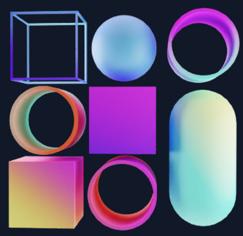
Where depth-based compositing is possible - eg. where more rendering power is required for a particular object that requires to be more in focus.



Cluster by plates (xR only)

According to background and foreground elements.

Benefits of RenderStream™



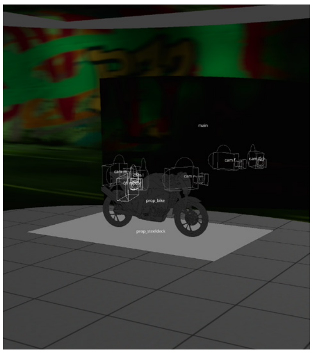
5. Superior system efficiency

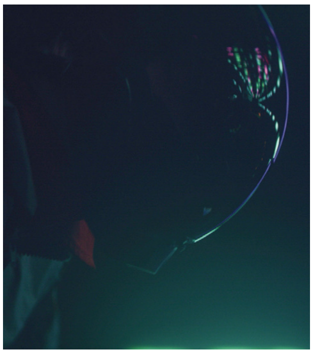
Certain shots require superior system efficiency in order to achieve the desired look. An example is a shot requiring both high-fidelity reflections and in-camera VFX. This is where you have a moving object with a reflective surface (eg. car window, or motorcycle helmet), but a background that requires VFX instead of standard plate shots captured in the real world.

In order for realistic VFX to be achieved, costly and lengthy post-production work would be required to not only add VFX to the background but also to have it appear as a reflection on the reflective surfaces (the helmet or car window).

With LED, the VFX appears not only in the background but is also reflected on surfaces like helmets and car windows. Additionally, with Disguise RenderStream™ you can focus rendering power on the desired part of the scene in order to get the part of the shot you want in the highest fidelity possible. There are two examples of this:

Shot 1

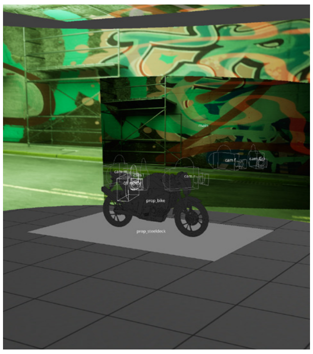




A close-up of the person with a background with VFX in a far depth of field. You can't really see the background as it is soft, but the person's face and potentially the car bonnet or motorcycle helmet are highly reflective and visible. The key element of that shot composition is the reflection render being as high-resolution as possible so that there is a really good reflection onto the glass of the vehicle or helmet, resulting in high quality and fidelity in-camera.

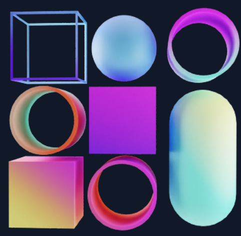
The reflections onto the vehicle from the outer frustum need to be really crisp and at a high frame rate, while the inner frustum is far away and out of focus.

Shot 2



The next shot commonly taken is to go the other way and do a wider shot of the car or motorcycle where the background (which is largely VFX) is in focus to provide the context of the shot. The person in the car or the motorcycle may also be in focus or may not be, depending on what you are trying to achieve. In that second shot, the most important thing is the hero frustum. The equation is reversed: the inner frustum, which needs to be the highest quality, makes the background look good as the car or motorcycle continues. The lighting and reflections from the outer frustum need only be sufficient for the car or motorcycle to be the right colour.

Benefits of RenderStream™



Efficiency is linked to flexibility in virtual production. Achieving the same level of high-quality rendering for any part of the LED volume would require all parts to be supported by multi-GPU render nodes. This approach would be expensive and inefficient, as one node would need to perform two render passes as well as compositing. For example, one GPU could handle the hero frustum, and another could handle the outer frustum. Re-cabling your infrastructure for different shots is, of course, impractical. With RenderStream™ it is simple to reallocate render power shot-by-shot, so that the power of the render nodes can be allocated to the right job at the right time. The Disguise approach enables a non-linear relationship between the amount of render nodes and the amount of pixels to be rendered for each fragment of the stage, while not overburdening nodes with multiple roles they can't handle.

6. Built-in redundancy

With a longstanding heritage in live events, known as one of the most unforgiving environments, RenderStream™ supports redundancy, with the Disguise Understudy concept which is, essentially, a hot backup.

If an individual content engine instance fails, failover is triggered either manually through the Disguise Designer UI or via an API with the help of RenderStream™. This results in the Understudy machine taking over, and the switch handled gracefully by the compositor that is dedicated to the main stream.

7. Using your content engine of choice

Finally, content engine agnosticism is a key Disguise philosophy enabled by the RenderStream™ approach. RenderStream™ is an SDK with four content engine implementations to date: Unreal, Notch, Unity and TouchDesigner and even more to be announced in 2023.

Importantly, when one content engine is swapped for another in the Disguise ecosystem, all the configuration around tracking and genlock remains the same, enabling creatives to swap between the most appropriate engine for the job.

8. Empowering limitless creativity

Even with the latest real-time game engine advancements, what you can put onto the LED volume largely depends on the sheer amount of GPU power you have at your disposal.

Improving the flexibility and scalability of your rendering platform allows for creative exploration, quick iterative changes and less of a need to fine-tune your real-time performance optimisation. With Disguise RenderStream™ enabling control of GPU power, the Disguise operator is empowered, via the RenderStream™ plugin in the content engine, to deliver the creative aesthetic at the required quality level without needing in-depth developer-level knowledge.

Case study: La Planète Rouge

La Planète Rouge's expansive volume in Marseille is powered by nine Disguise RX render nodes.

"The power of RenderStream™ and mesh mapping allows us to achieve very high levels of 3D rendering that can be easily configured to suit our needs," says La Planète Rouge's CEO and Producer, Lionel Payet Pigeon. They can dedicate one or more servers solely to the inner frustum and achieve exceptional levels of rendering and sharpness in-camera.

According to Lionel, "this flexibility also allows us to use several servers to render the outer frustum: no aliasing and no compromises. The 3D is uniform, even on the part of the wall that the camera does not see, but whose precision has a huge influence on the overall rendering, with the ability to render around 60 million pixels in real-time."

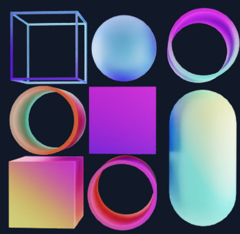
La Planète Rouge regularly uses RenderStream™ on commercials or fictional car scenes. When dealing with wide shots involving a car, the accuracy of the rendering across the wall becomes a key issue.

On-set for a recent Mercedes commercial, Disguise RenderStream™ enabled La Planète Rouge to cover the full 200° of their volume with a consistent level of accuracy in the 3D environment - improving the final output significantly. La Planète Rouge has recently used Disguise in the virtual production scenes done for Warsha, the winner of the Best Short Film Jury Award: International Fiction at the 2022 Sundance Film Festival.

It's impossible to achieve this level of rendering with any other system. We would have to make huge rendering concessions for a large volume like ours. Having started this way, we won't go back... this flexibility is essential to give productions and creative teams the best."

Lionel Payet Pigeon
CEO, Producer
La Planète Rouge

MELS recommends RenderStream™ for system efficiency



The fact that, with Disguise, we can use one computer (RX) to render one frustum is incredible. Before, when we were using the traditional method of ICVFX, it was one computer per part of the screen. Now we can use one computer per task and apply it to the entire screen. It makes it logical and easy to use. You can use the full power of each computer for different tasks, and you don't have to waste horsepower for part of the screen that you don't see or use."

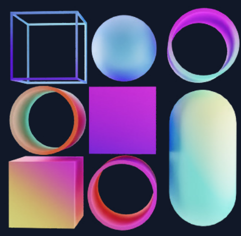
Simon Girard
VP Operator
MELS

XR Studios recommends RenderStream™ for engine agnosticism

RenderStream™ allows us to use multiple engines in our space, including Unreal, Notch VFX and Unity. This has been particularly useful for the bigger projects utilising lots of different engines, such as the Camila Cabello "Familia" TikTok Concert, where Silent Partners Studio used both Unreal Engine and Notch VFX for the six-song performance with unique environments and transitions."

J.T. Rooney
President
XR Studios

MEPTIK recommends RenderStream™ for cluster rendering



“Without RenderStream™ we would have had to rely entirely on the tracking system’s lens calibration to be 100% perfect, which is not best practice. With RenderStream™’s cluster rendering capability, we were able to split the rendering workload across multiple render nodes and pass the video frames through a network instead of needing more video outputs and LED processors.”

David Vargas
Senior xR specialist
MEPTIK

disguise

Create spectacular productions without limitations

Apply to join our Virtual Production Accelerator Course

Apply now

Access our Designer software

Access now

Learn more about Disguise’s Virtual Production solution

Learn more