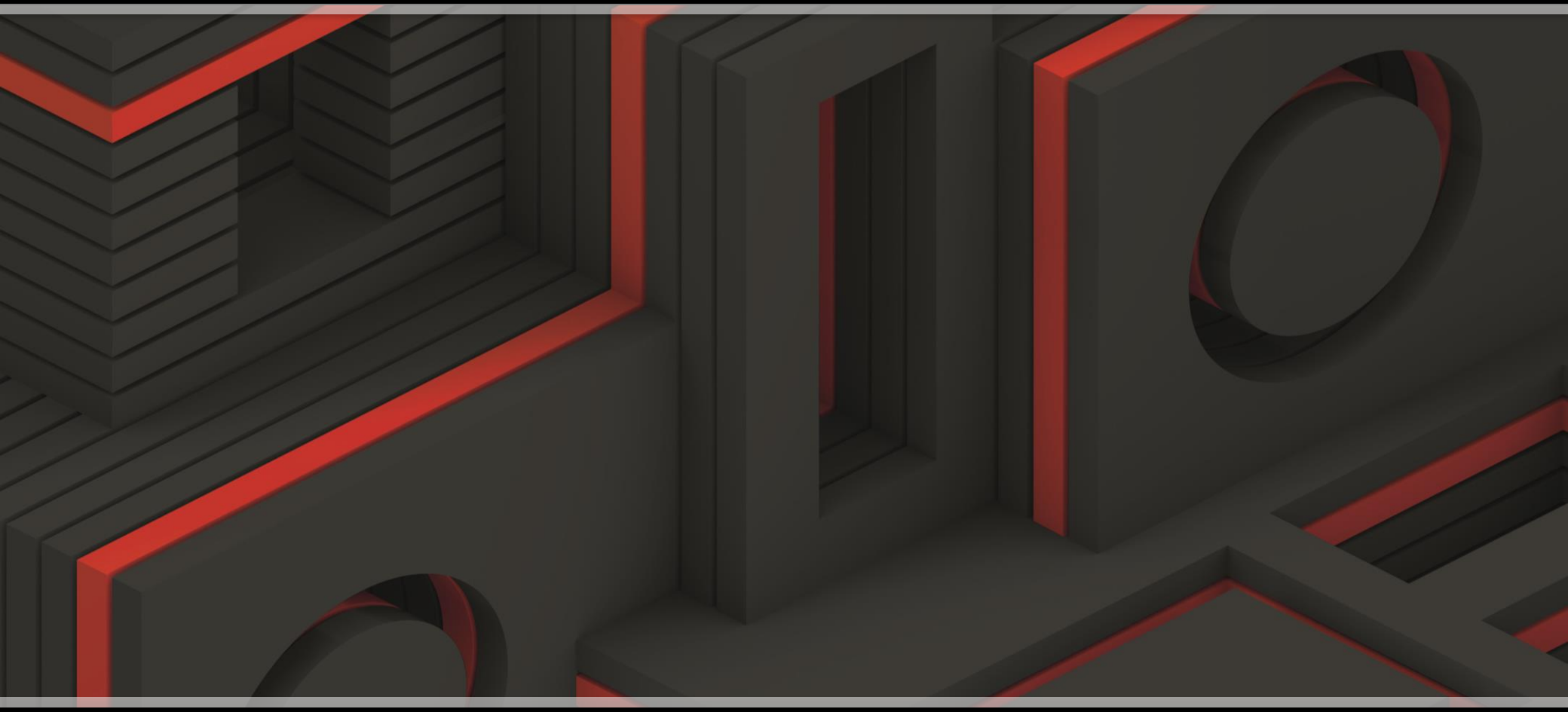
Whitepaper

VENTUZ



Ventuz X

Interaction data exchange using Ventuz Interactive Rectangle

Interoperability Workflow

Preamble - The need for interoperability

According to Wikipedia, "Interoperability is a characteristic of a product or system (…) to work with other products or systems, at present or in the future, in either implementation or access, without any restrictions".

From a day-to-day graphics workflow perspective, interoperability, or interop for short, is the need to combine various software and hardware solutions flawlessly in order to create a graphics project.

Therefore, interop is a key requirement in modern day graphics, with end-customers asking for more complex, visually appealing and multi-platform graphic presentations, that involve not only an increasing number of software applications working together, but also a heterogeneous mixture of hardware systems and devices .

All this means that an application cannot work as an island anymore, without communication to other devices and applications, or integrated just with solutions belonging to the same eco-system.

In order to achieve this goal the support for industry standards and protocols protocols and the integration of 3rd party HW solutions are key to bring all the pieces needed to build modern workflows for today and the future.

Talking about graphics software this means three with 3rd parties of all sort, from various HW vendors, to graphic creation software, new control devices and integrating with a wide range of data types.

Some software solutions, like Ventuz, include all the connections to be the central piece in the workflows mentioned above.

This openness includes sharing media assets, exchanging different types of data or outputing the final result to a wide range of displays and/or projection solutions.

This makes Ventuz the ideal solution to build a very robust, production-proven and open workflow that integrates with software and hardware solutions from many vendors.

"A graphics solution like Ventuz cannot work as an island anymore, with no communication with other devices or applications."

Ralf Stanke
CEO, Ventuz Technology

An example of large-scale corporate presentation made with Ventuz

Sharing rendered pictures in real time

Since we are talking about graphic workflows, visuals are the first assets that must be shared between applications. This includes not just pictures, but also other types of graphic content like video files or 3D models.

The graphic assets that can be transferred to Ventuz include various standard image files formats, even Photoshop PSDs with separate layers and their settings, SVG 2D vector graphics and 3D scenes or models, thru industry standard 3D exchange file formats, like FBX, OBJ, Collada (DAE), or even the more advanced GLTF format.

In addition to these files, Ventuz also supports various audio and video file types to transfer animated content using standard video codecs and containers, like H.264 or the more specialized HAP codec family, designed for mega pixel resolution projects, or Avid DNxHR for broadcast applications.

Ventuz is all about real-time, so graphic interop must go beyond offline transfer of media assets, in many cases rendered pictures must be transferred as they are rendered and composited within Ventuz graphics.

Usually Ventuz will be at the very last stage of the graphics workflow, rendering the final result and outputting to multiple displays or projectors – then Ventuz will receive the graphical output of other applications/systems.

Ventuz open nature offers many ways of integrating visuals in real-time, like:

- Using Newtek NDI video streaming technology and the free NDI Tools that allow streaming the system or application GUI.
- Using 3rd party HDMI Video Input boards from Deltacast, Blackmagic, AJA or Datapath.
  For a list of supported models check here.
- Using Spout, a real-time video sharing framework for Windows, that it is supported by many graphic applications in the market.
- Using Ventuz VIO, Ventuz Video Input Output API, allows Ventuz users to program applications that send video streams towards Ventuz. Apart from the pixel color and alpha data, Z depth buffer and other ancillary data can be transferred to Ventuz.
  You can get more info about VIO here.

Beyond images – Interaction data exchange

As mentioned on previous pages, these mixed workflows require not only transferring graphic and media assets, but also integration with a certain number of control devices and other graphic solutions.

This communication must be bi-directional in many cases, like data exchange with external databases that require not only retrieving data from them, but also pushing data that will be added to the database records.

This bi-directional data transfer is also crucial when connecting to external control devices or applications, that often require some sort of feedback or other types of data exchange.

On the other hand, real-life projects usually require displaying other media, such as external computer graphical interfaces (GUI), presenter screens and the like. Besides, in many projects the personal preferences and tech proficiency of the presenter have a huge impact on how these contents are prepared and displayed.

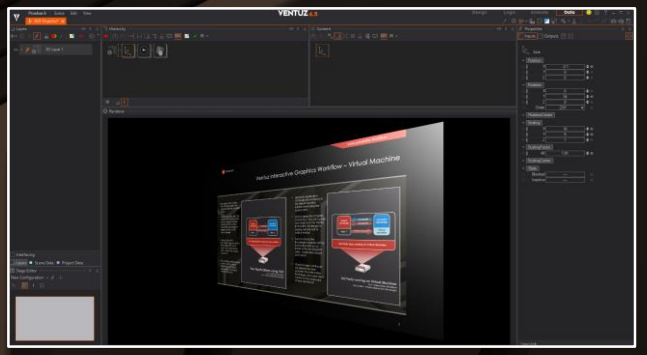
Large scale corporate presentations are the paradigm of these requirements, A-level staff present certain sensitive data that they want to keep secret until the very last minute, or they want to control how it is presented or triggered.

One of the most usual requirements is integrating keynote style presentations within flashy corporate graphic content, which include external data and complex content logic, while keeping a certain degree of manual control over the keynote presentation.

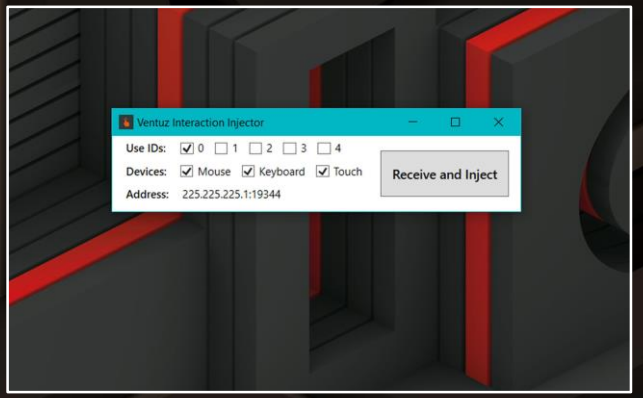
Transferring the keynote computer display is not an issue, with all the options mentioned before, using some sort of video streaming protocols like NDI. Another option would be using 3rd party devices to capture the DP or HDMI output of the computer running the keynote and mapping it as any video media inside your graphics.

In addition, Ventuz offers a very advanced solution called VIO that allows image and data transfer on the GPU level, this opens the door to the integration with 3rd party render engines.

In all these cases, the bit that was missing was the 3rd party content triggering, how to keep control of these keynotes from within the final graphics environment.



A remote keynote presentation inserted in a Ventuz scene



Vinjector application receives and injects the interaction data on remote system

The Ventuz solution - Interactive Rectangle node

The remote control of other systems has been a usual request from our users over the years and we have added new tools to improve these workflows on our latest version, Ventuz 6.5.

One of the main new features in Ventuz 6.5 is a new node designed to address these workflows requirements, the **Interactive Rectangle node** .

Interactive Rectangle renders a rectangle geometry inside of the Ventuz scene, and collects all the customer interaction - keyboard, mouse and touch - that occurs inside this rectangular area.

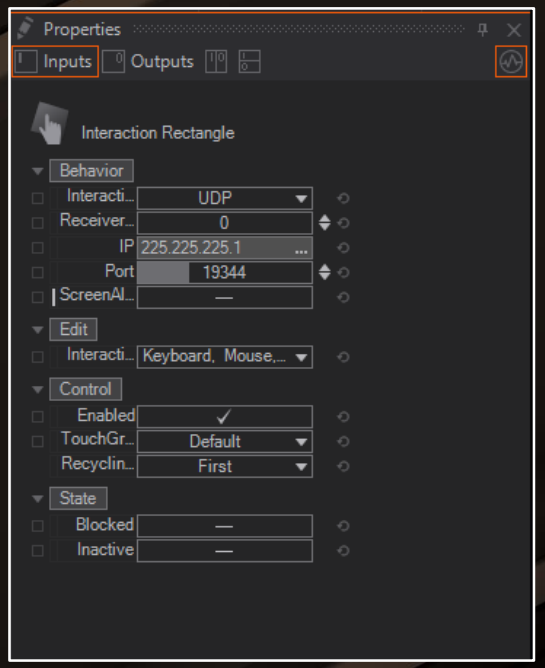
The interaction data is then streamed to the 3rd party application in real time to enable a remote control of the external application within the Ventuz scene space.

All this customer interaction data is sent back to the 3rd party application using an OSC-based protocol over network via UDP, in the case of a remote system or a virtual machine or using the VIO Ancillary Data (ANC) channel, in the case of two applications running in the same system.

If your project requires two application running in the same system, Ventuz VIO goes beyond sharing rendered pictures, VIO also includes the VIO Ancillary Data (ANC) channel to exchange other control and scene data. Interactive Rectangle supports this VIO ANC channel to send back the interaction to the external application.

Therefore a typical workflow example includes an Interactive Rectangle which is textured with the graphical interface of the 3rd party software. This allows the GUI of the external software to be displayed inside Ventuz graphics, while keeping all interaction capabilities – you could think of it like a remote desktop inserted in the Ventuz graphics environment.

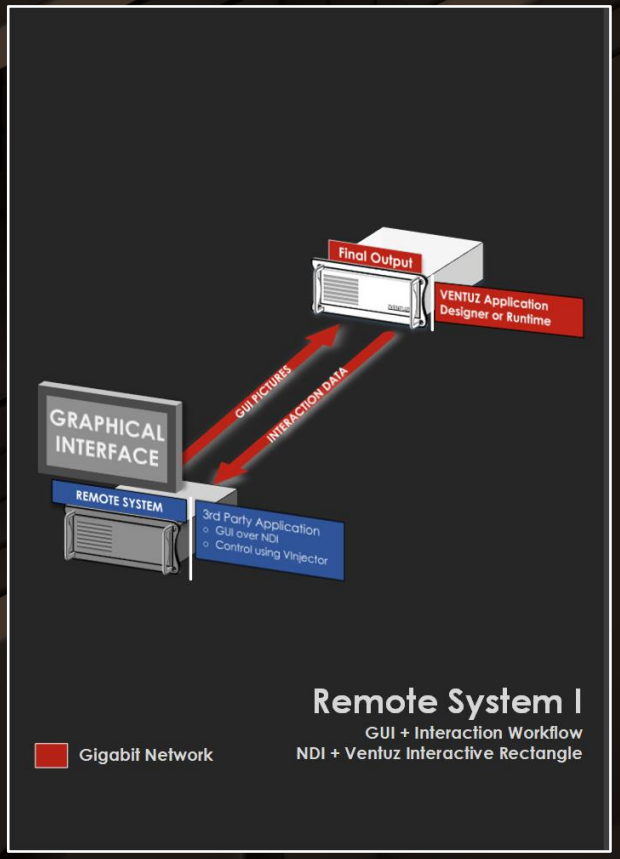
For 3rd party applications running on an external system or virtual machine, interaction data is sent over a standard network connection and the remote system must run a small software called Vinjector, that receives and injects the interaction data back into the remote system.



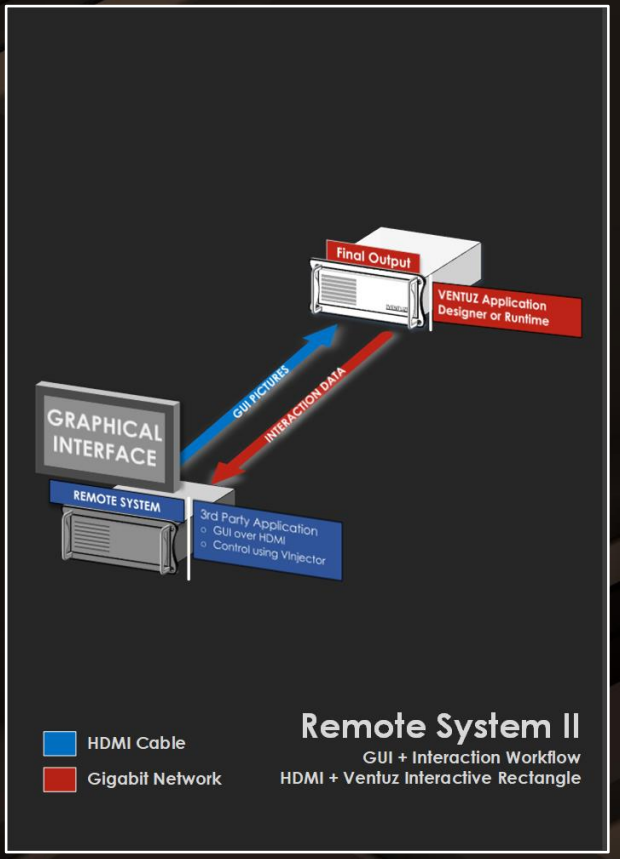
The input properties of the Interactive Rectangle, including the remote system IP address/port and interaction options available

Ventuz Interactive Graphics Workflow – Remote system

- The rendered pictures or the 3rd party application graphical interface are streamed over a standard network connection using Newtek NDI.
- NDI Stream is mapped to the Ventuz Interactive Rectangle to display remote GUI or render.
- Ventuz Interactive Rectangle capture all the interaction that occur inside of the rectangle area – Keyboard, mouse and touch
- Vinjector app receives the Interactive Rectangle data via network and takes control of the local keyboard, mouse and touch

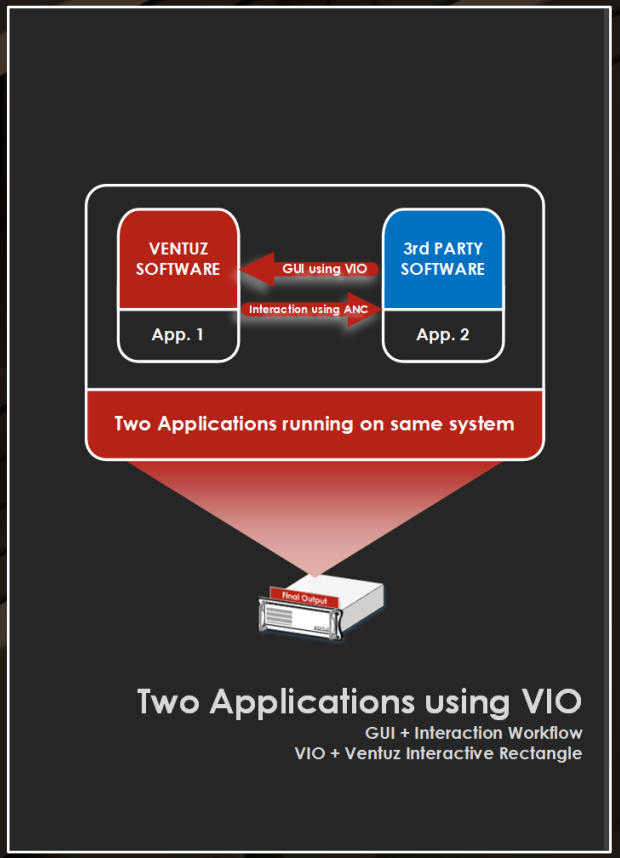


- The rendered pictures or the 3rd party application graphical interface are streamed over HDMI Cable.
- HDMI signal received on Ventuz system using 3rd part Video I/O board is mapped to the Interactive Rectangle to display remote GUI or render.
- Ventuz Interactive Rectangle capture all the interaction that occur inside of the rectangle area – Keyboard, mouse and touch
- Vinjector app receives the Interactive Rectangle data via network and takes control of the local keyboard, mouse and touch



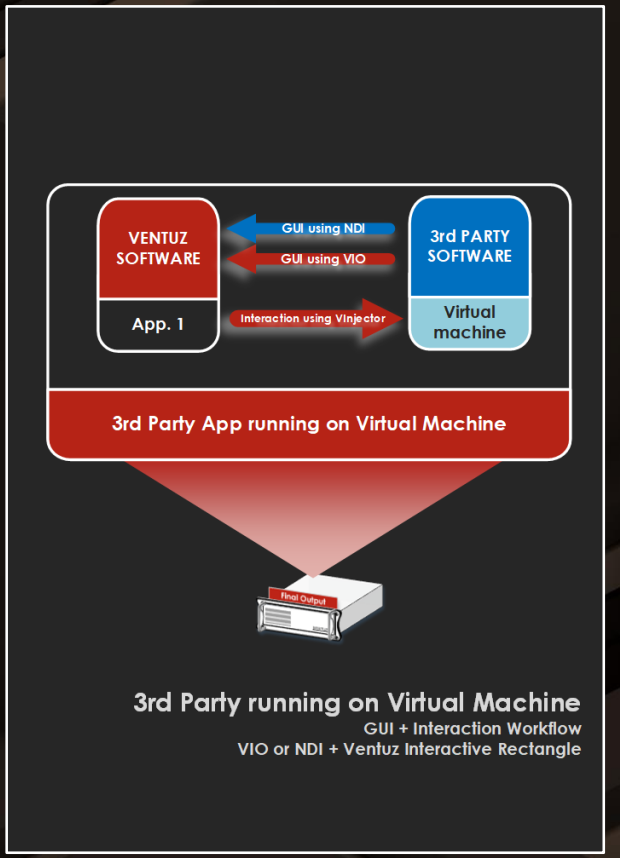
Ventuz Interactive Graphics Workflow – Virtual Machine

- The system GPU renders the 3rd party application graphical interface/render output.
- Ventuz VIO grabs the GPU render and transfers it into Ventuz, where it can be mapped to the Ventuz Interactive Rectangle to display remote GUI or output render.
- Ventuz Interactive Rectangle captures all the interaction that occur inside of the rectangular area – Keyboard, mouse and touch
- VIO Ancillary data channel is used to send control data back to 3rd party application - keyboard, mouse and touch



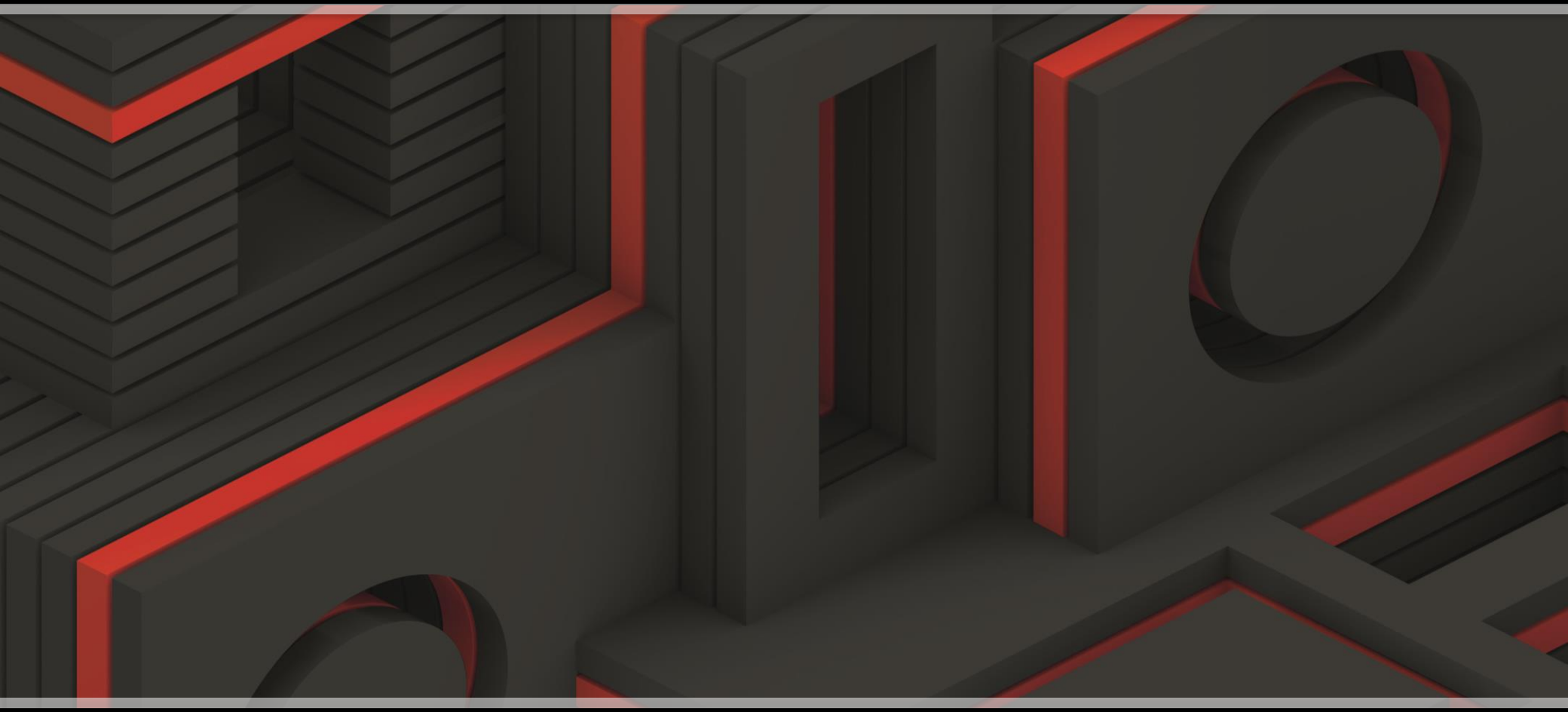
Two Applications using VIO
GUI + Interaction Workflow
VIO + Ventuz Interactive Rectangle

- 3rd party Application GUI/output is rendered in the virtual machine environment using the system GPU.
- Ventuz grabs the 3rd party GUI/Output thru VIO or NDI and maps in to the Ventuz Interactive Rectangle, to display remote GUI or output render.
- Ventuz Interactive Rectangle captures all the interaction that occur inside of the rectangular area – Keyboard, mouse and touch
- Vinjector app running on the virtual machine receives the Interactive Rectangle data and takes control of the keyboard, mouse and touch



3rd Party running on Virtual Machine
GUI + Interaction Workflow
VIO or NDI + Ventuz Interactive Rectangle

VENTUZ



Ventuz Technology AG
Luttherothstrasse 16a
20255 Hamburg – Germany
P. +49 40 413 66 09 70
info@ventuz.com
www.ventuz.com

Interaction data exchange using Ventuz Interactive Rectangle

By David Paniego