Accelerating Realism with the (NVIDIA Scene Graph)

Holger Kunz
Manager, Workstation Middleware Development
Phillip Miller
Director, Workstation Middleware Product Management

SIGGRAPH 2009 presentation

NVIDIA® application acceleration engines ("AXE")

NVIDIA® SceniX™ **was NVSG**

The NVIDIA Scene Graph (with **Cg 2.2**), powering the world's most demanding real-time applications

NVIDIA® CompleX™ **was NVScale**

Making massive data sets interactive, greatly rising the ceiling for frame buffer memory

NVIDIA® OptiX™ **was NVIRT**

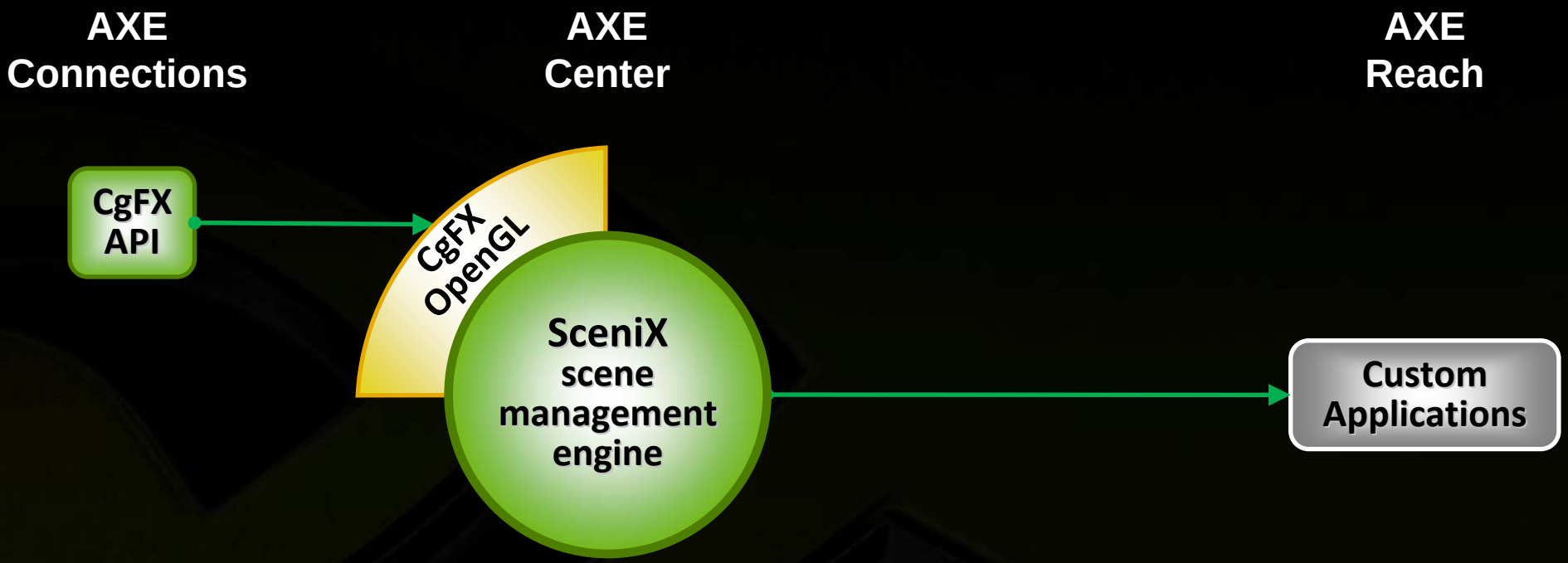
Interactive GPU ray tracing,
taking interactive realism to a new level

NVIDIA® CgFX™

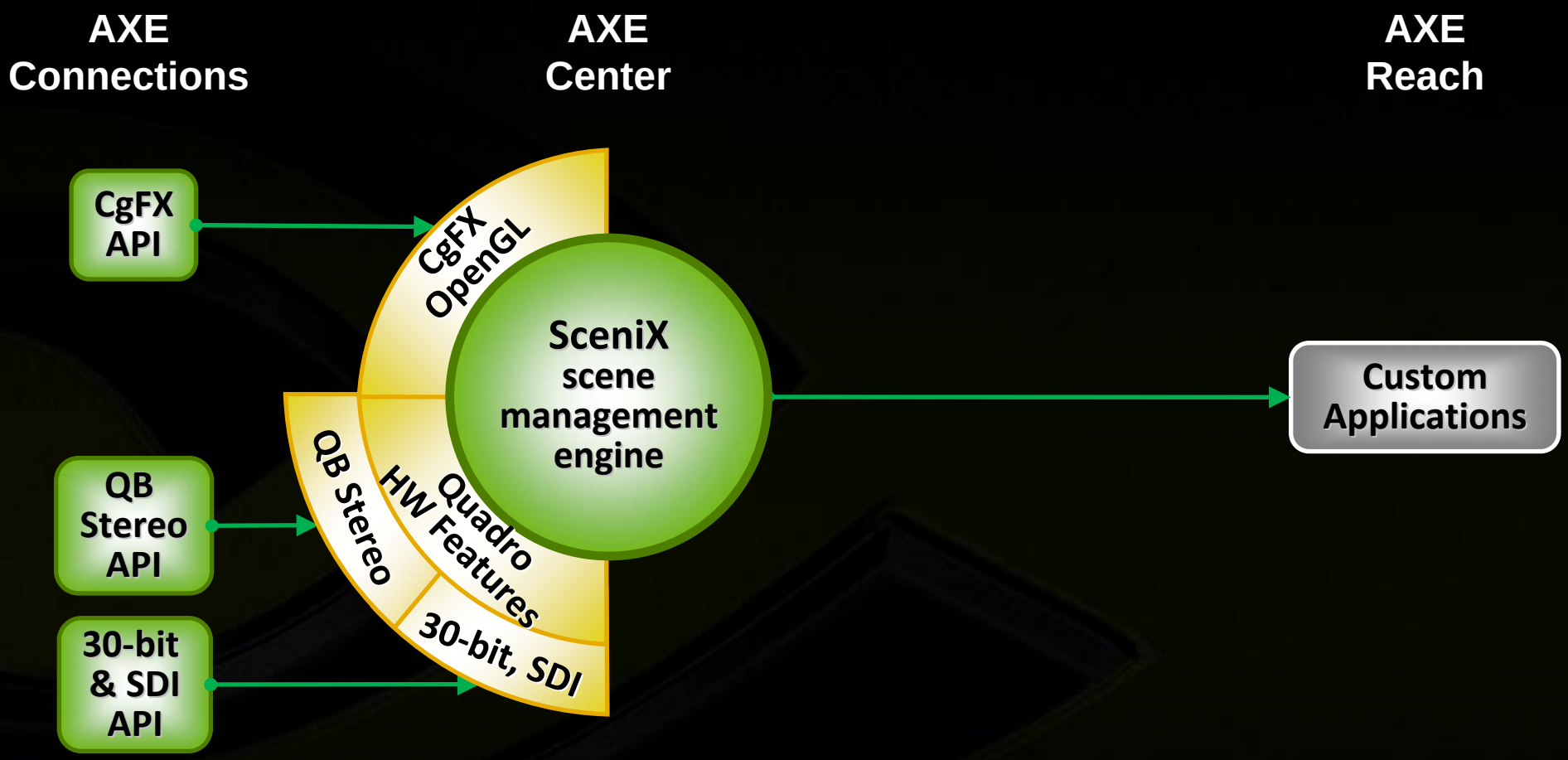
Meta language and runtime approach, taking programmable shading and material development to the next level.

customer examples courtesy of Autodesk Showcase, RTT, Mercedes Benz

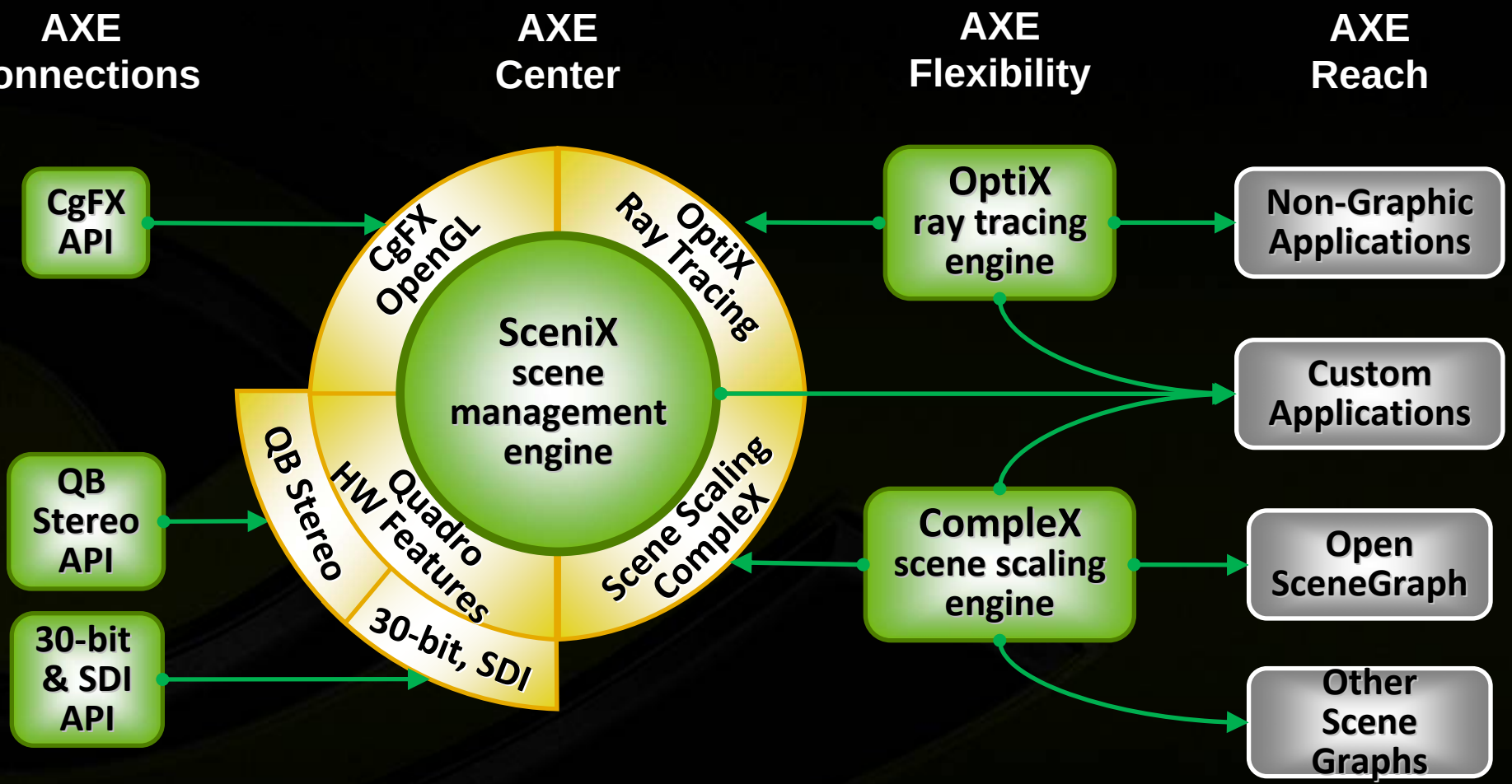
AXE – Engine Relationships: Initial



AXE – Engine Relationships: 2008



AXE – Engine Relationships: Today

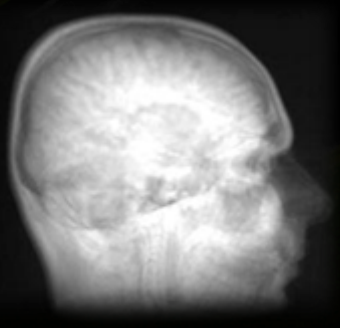


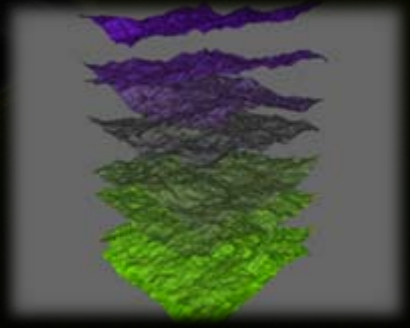
SceniX (NVIDIA Scene Graph)

the fastest route to a high performance 3D app

Created by NVIDIA for software developers, to speed the creation of high performance 3D applications that exploit the latest GPU and OpenGL capabilities, by providing:

- A solid, free-to-use, cross-platform foundation for creating 3D App's
- The latest CgFX support for maximum rendering options and quality
- Constant speed improvements in scene management & rendering
- **Freely available** for building your interactive applications





SceniX 5 - Built-in support for latest NVIDIA features and products

Cg 2.2

Superior rendering and performance

10-bit color, SDI i/o

Support for demanding color and video markets

3D Vision

High end 3D stereo support for immersive and desktop systems

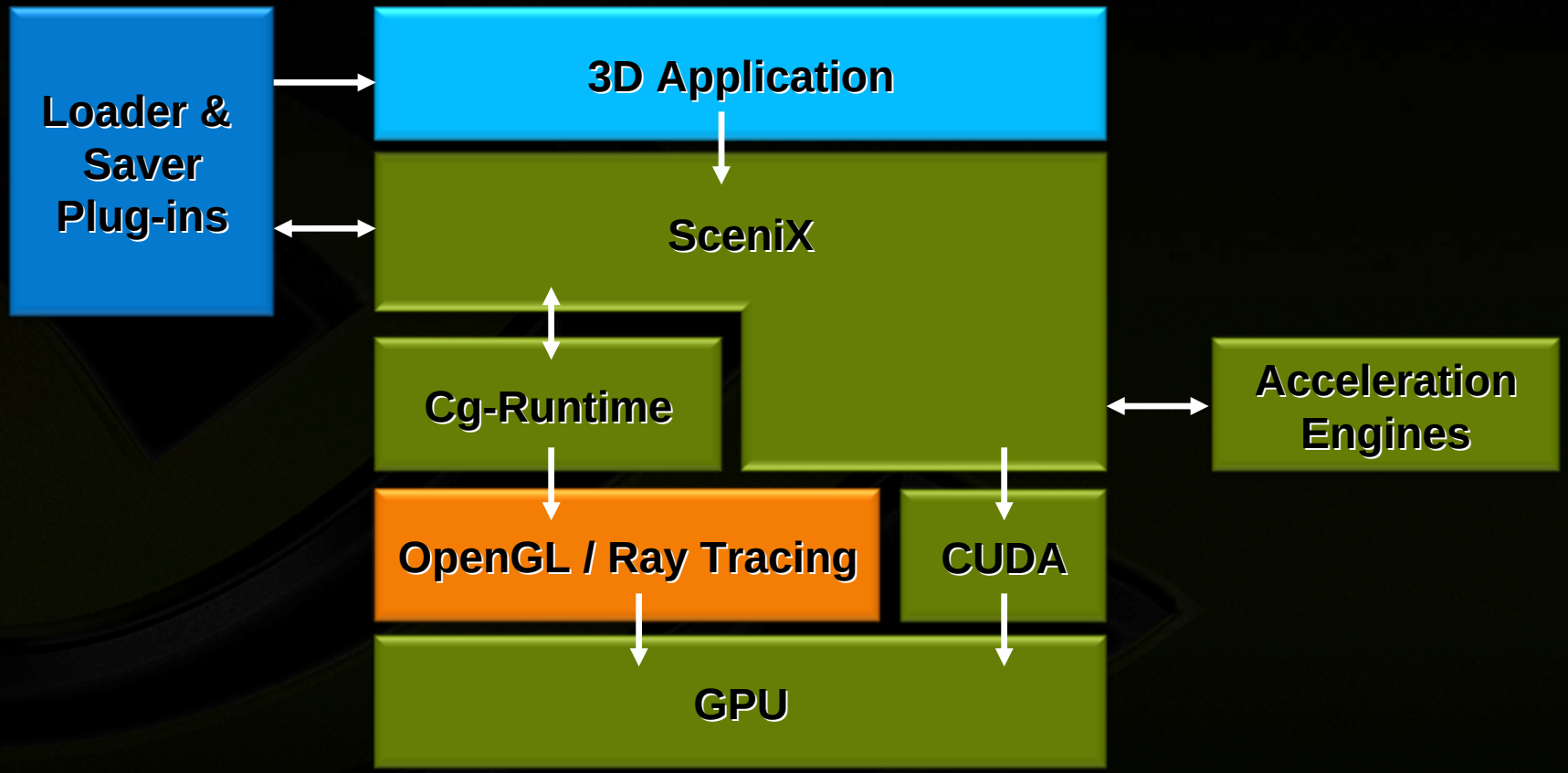
MetaSL

Support for sharing mental mill® shaders between renderers & applications

CompleX and (soon) OptiX

Built-in support for massive scenes via GPU distribution and **interactive** GPU ray tracing (later this year)

SceniX Software Stack



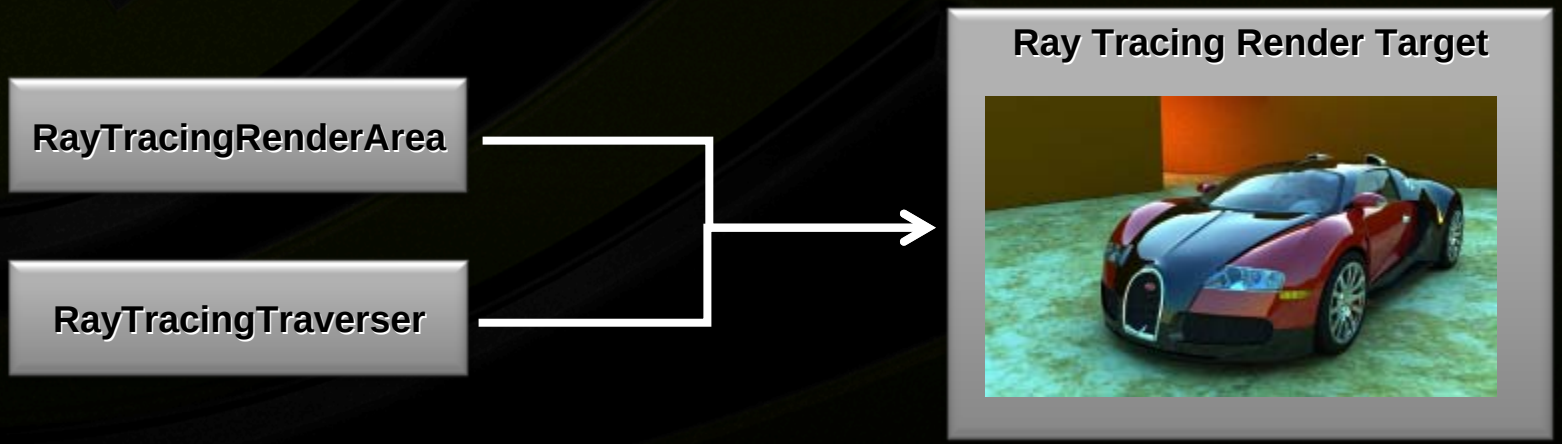
Support of Different Render Targets

SceniX allows different render targets to achieve different effects, like:

- Framebuffer (direct output)
- FBO (e.g. Re-use the FBO for reflections or shadowmaps
- 2D Overlay (e.g. For menus and GUI elements)
- Ray Tracing (high quality shadow, reflections and refractions)
- Broadcast Graphics Hardware (GVO / SDI)
- Transform Feedback (e.g. HW accelerated skinning)

Render Target for Ray Tracing

- Ray tracing render area needs to provide the output buffer for the ray tracing engine and a mechanism to present the final image on the screen
- The ray tracing traverser is responsible for sending the information contained in the scene graph to the ray tracing engine

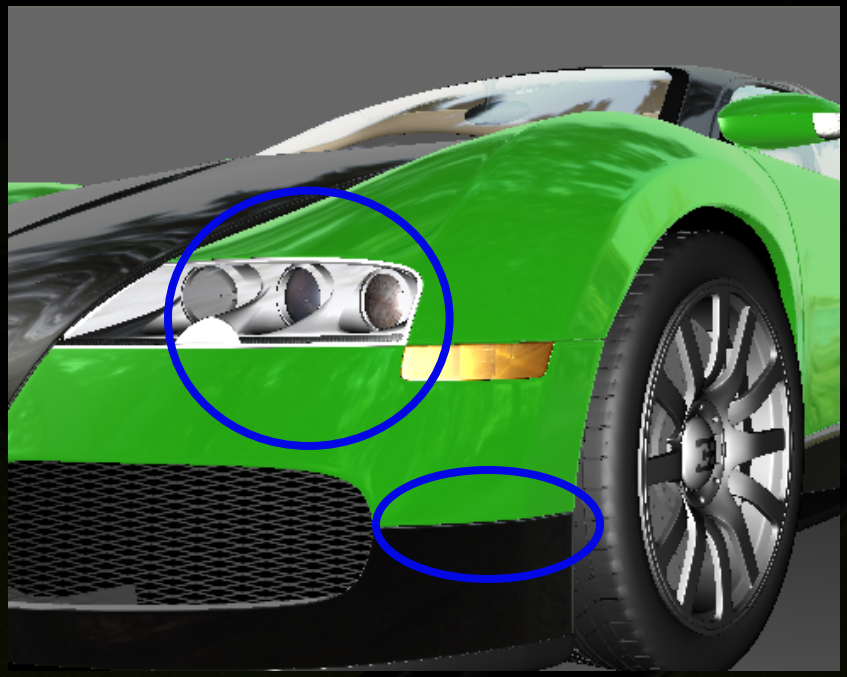


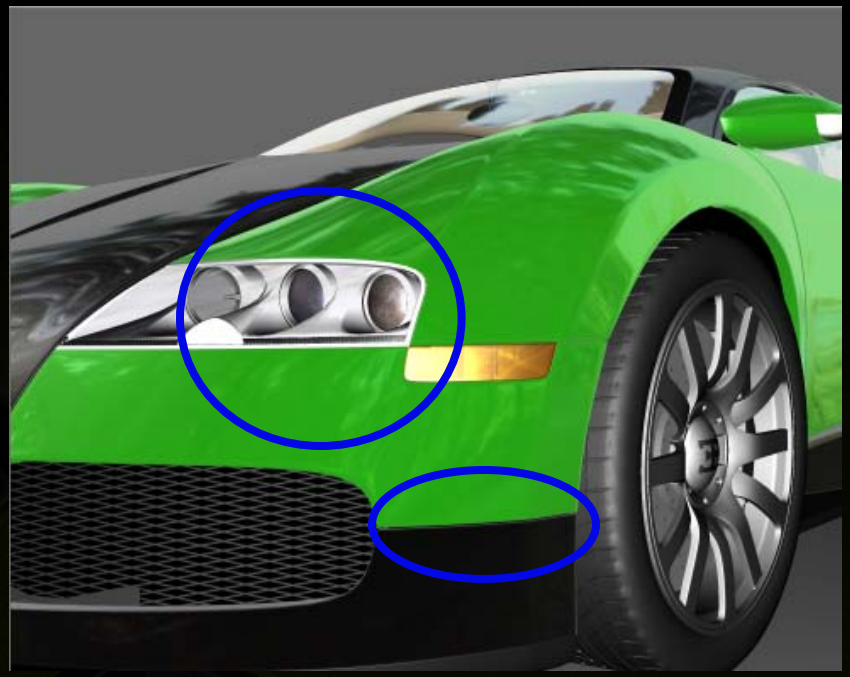
Multipass Rendering

SceniX supports multipass rendering for a wide range of multipass algorithms such as:

- Depth of Field
- Order Independent Transparency
- FSAA
- Stereo (implementation dependent)
- And many more

Maximum FSAA Quality





No FSAA

16x MPAA

MPAA: Combine HW FSAA with Software controlled FSAA to get benefits from both worlds:
maintain speed and quality

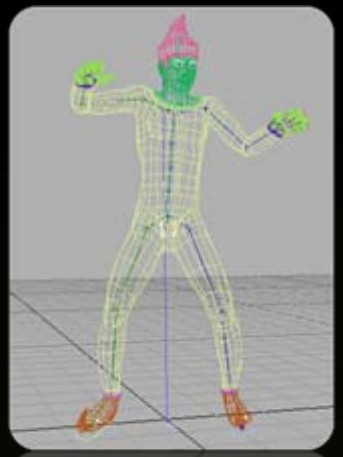
Stereo Support

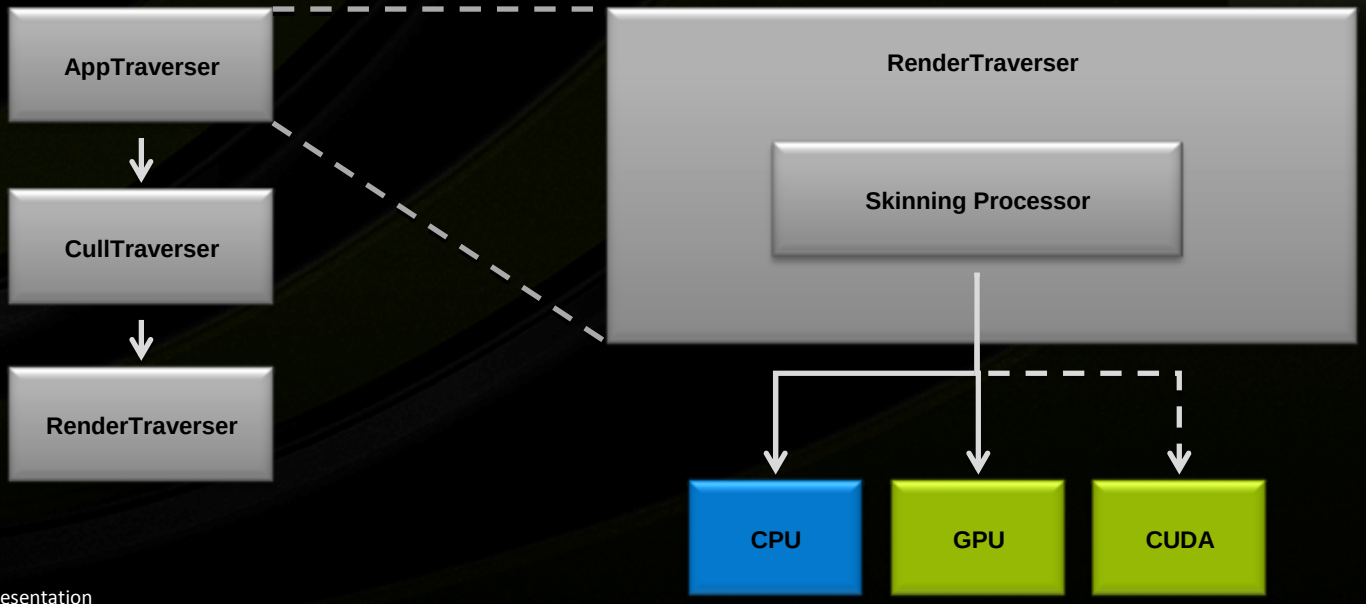
- **SceniX offers ways to support different kind of stereo formats**
  - Active Stereo – Quad-buffered stereo
  - Passive Stereo – Render left and right eye and compose final image
- **Support for different Stereo solutions**
  - Interlaced stereo
  - Red/Cyan
  - Custom

Skinning Support

Usage of skinning processors

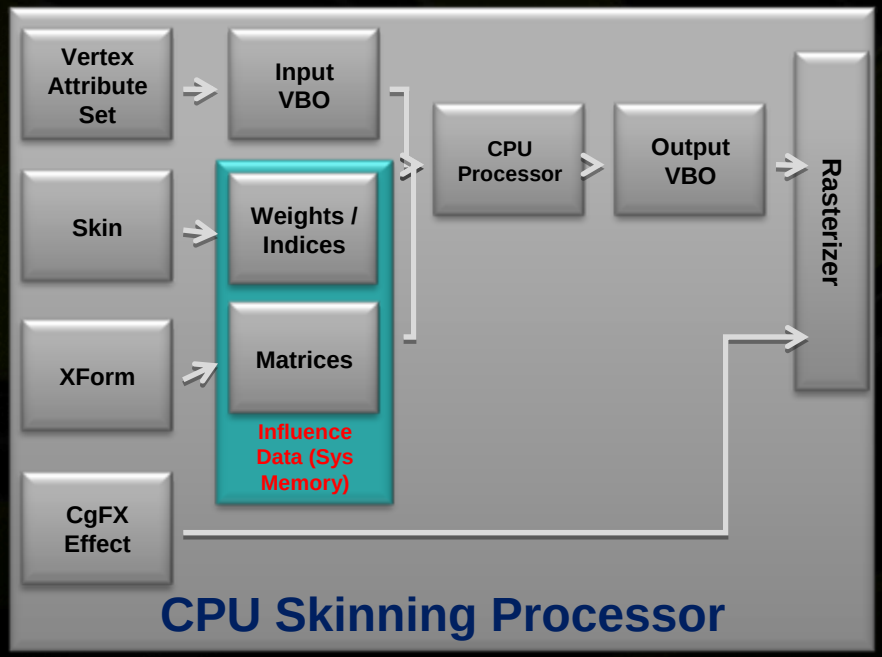
- CPU Skinning Processor
- GPU skinning Processor
  - Hardware accelerated skinning
  - Minimal requirement: Shader Model 4
- Future: CUDA skinning

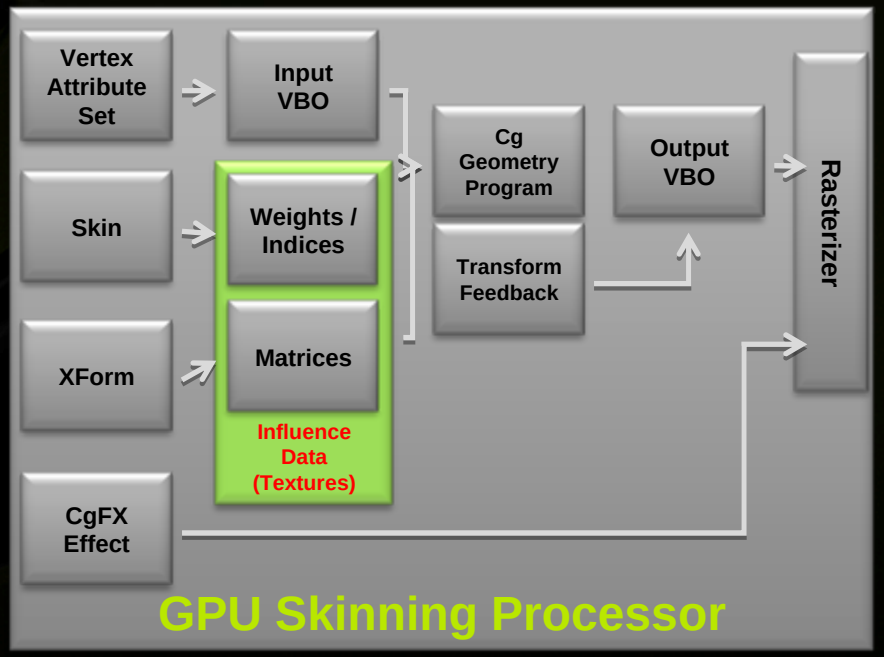




Skinning Processors

- CPU Skinning – data needs to be moved from system memory to video memory on a frame by frame basis
- GPU Skinning – data can reside and manipulated on the graphics board



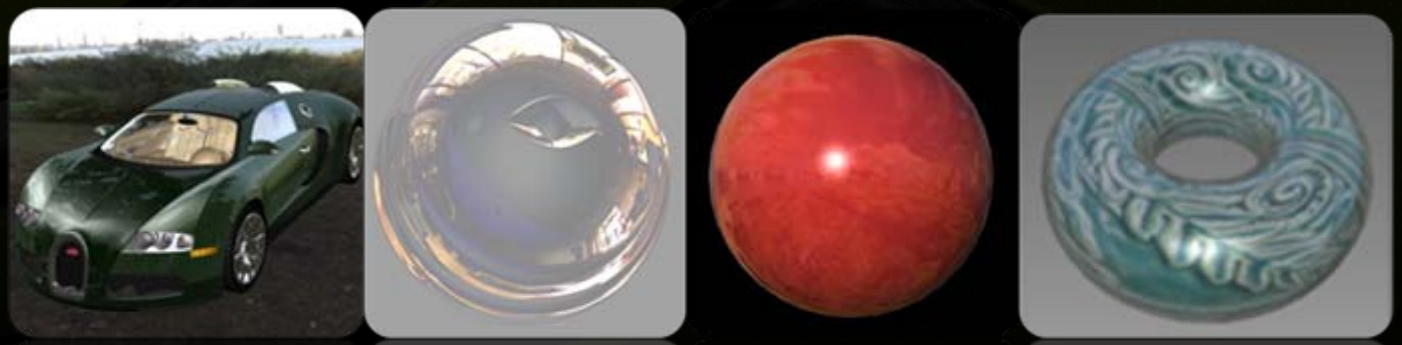


Shading – Cg/CgFX

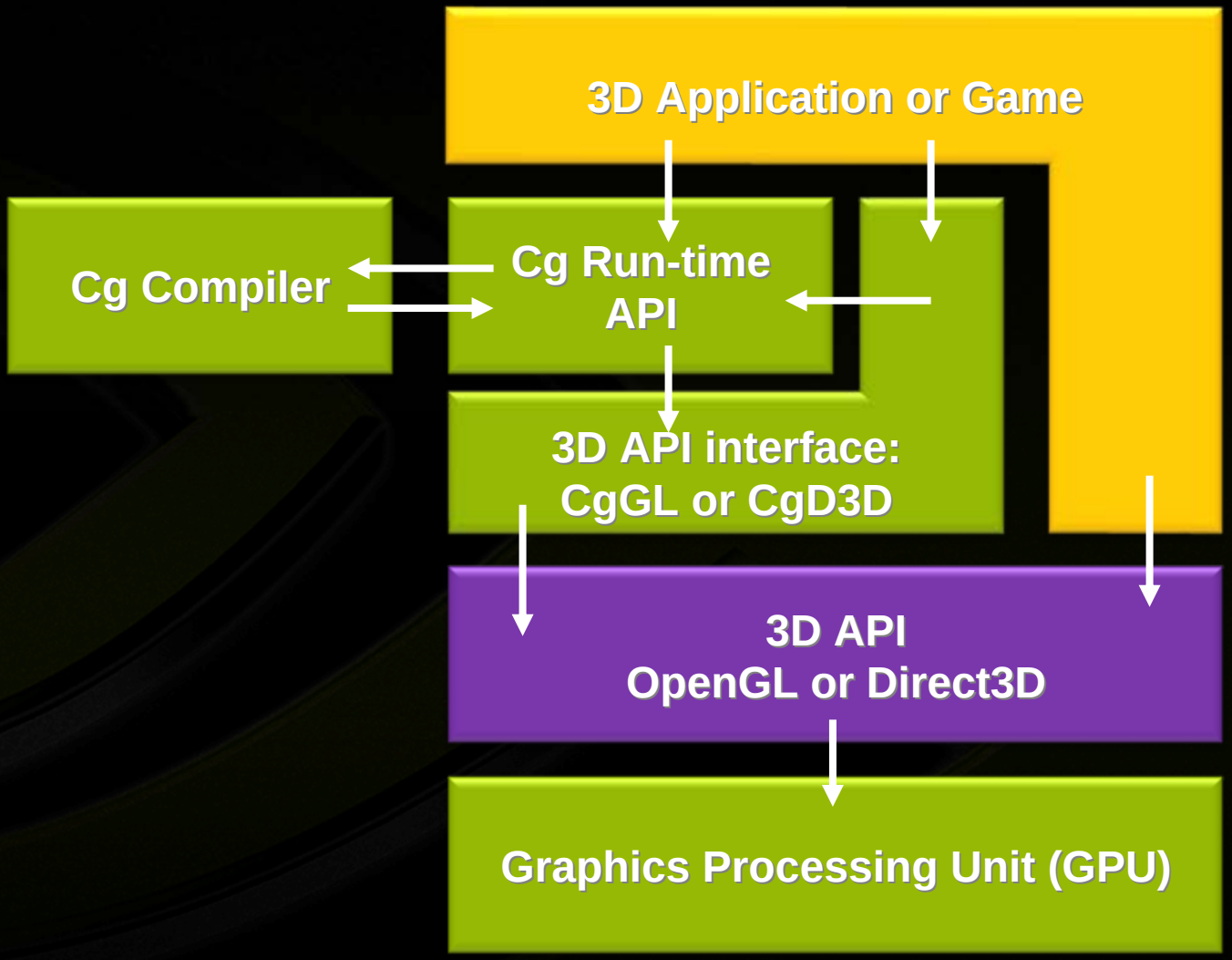
Cg/CgFX developed for developers and artists to make the material definition easier and more effective. Special effects can we written in a special effect file format which directly links into the Cg runtime for easy parameter and tweakable handling.

- Cross platform and API-independent
- GPU Shading language inspired by C
- Exposing latest HW features
- Part of 3D content creation tool chains - professional apps and games

Write your shaders in Cg/CgFX and deploy them to any API or platform

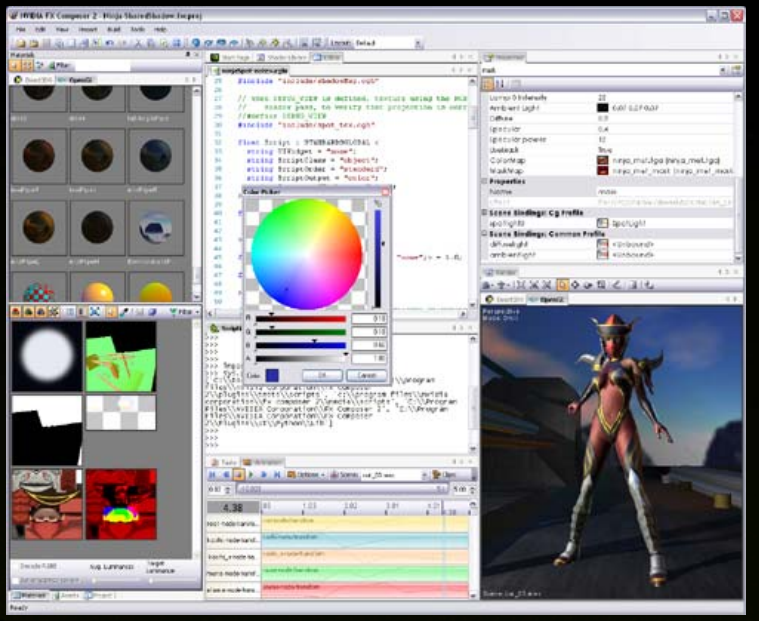


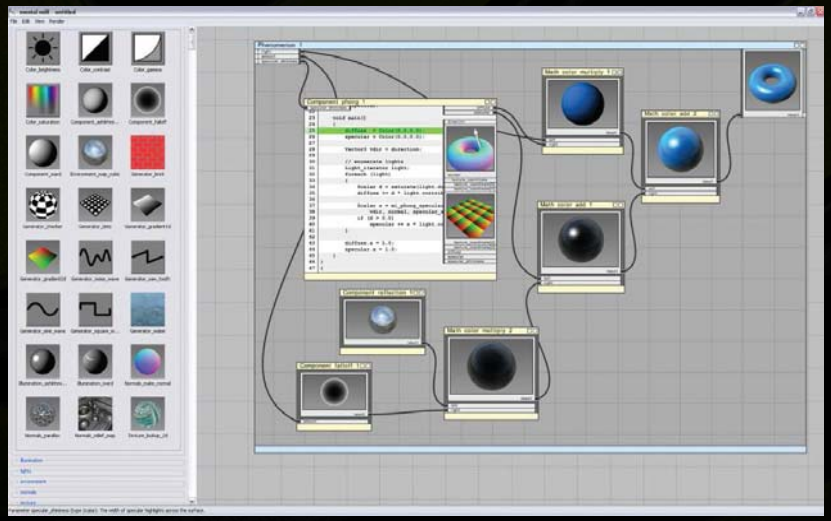
Cg software stack



Shader Authoring Tool Integration

- Shader authoring tools
- FXComposer
- mental mill

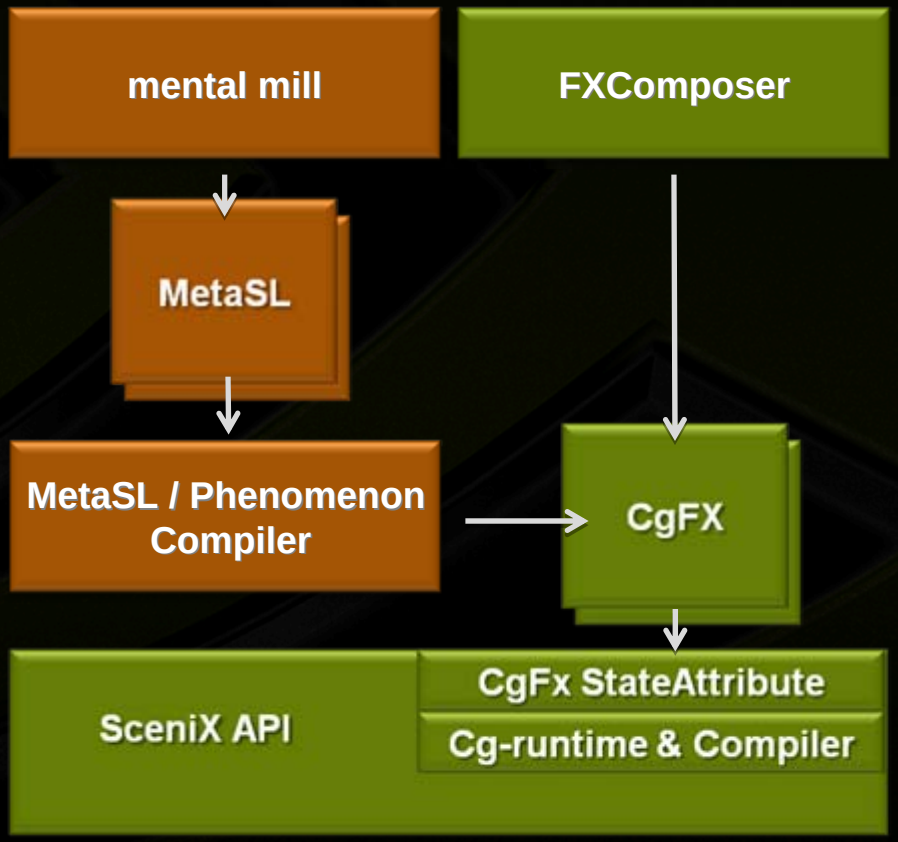




- Visual programming
- Drag & Drop
- Debugging

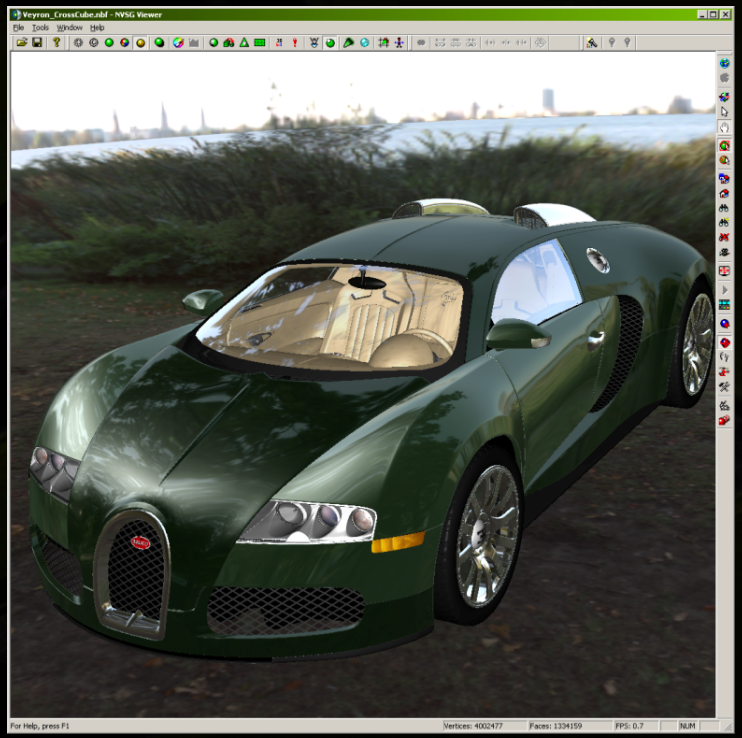
MetaSL support with SceniX

- MetaSL will be translated to a valid CgFX shader
- The translated CgFX can use the whole Cg framework in SceniX



Demo – SceniX + Material

- Sample Viewer created with SceniX
- OpenGL with CgFX material handling



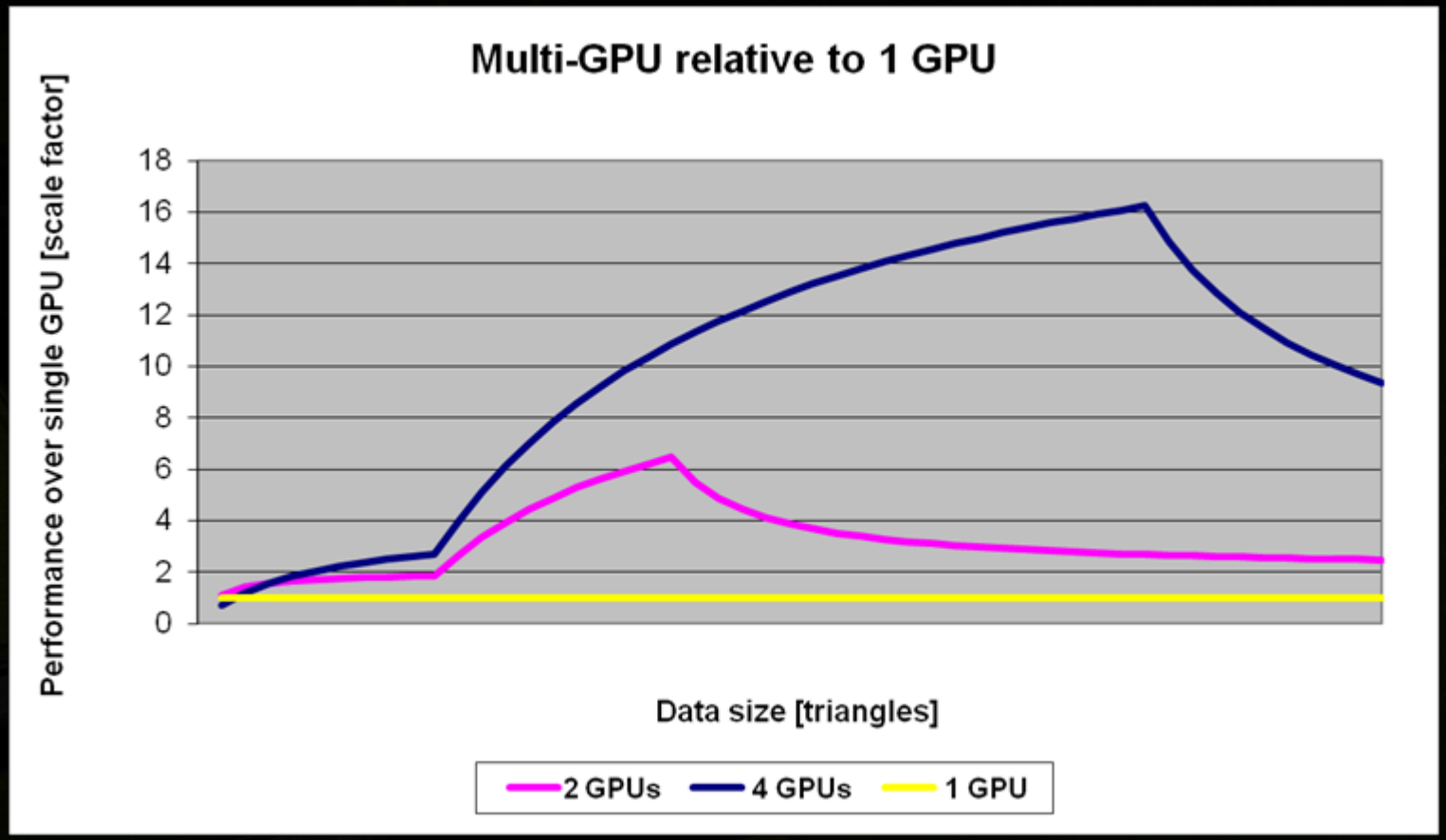
NVIDIA® CompleX™ scene scaling engine

busting through the 4GB ceiling

CompleX makes massive scenes interactive by easily integrating with OpenGL applications for fully leveraging the memory and performance across Quadro Plex GPUs

- Automatically distributes the geometry load across GPUs and composites a seamless end result
- Can leverage up to 8 GPUs and **32 GB** of memory; using 4 Quadro Plex D2 or 2 Quadro Plex S4 systems
- Direct support for SceniX, OpenSceneGraph and (soon) Open Inventor
- CompleX SDK for custom integration with *any* OpenGL scene graph
- "Developer Mode" works with any two matching Quadro FX boards

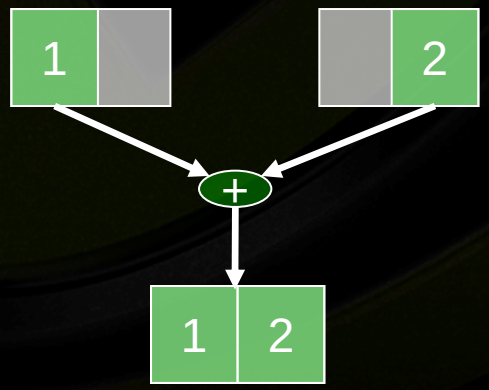
CompleX: Performance Scalability



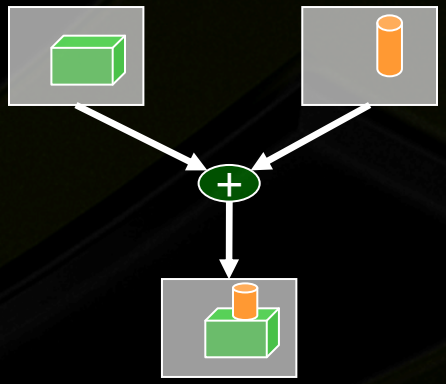
CompleX SDK - Compositor

Image compositor for sort-first & sort-last based applications

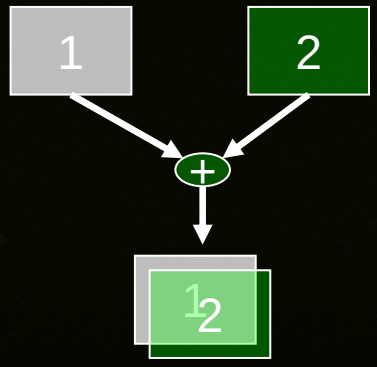
- Screen tiling, alpha and depth based compositing approaches
- Platforms: Win 64, Linux 64
- Compositor implementation based on latest technologies, minimal migration effort for applications



Screen Tiling



Depth Compositing

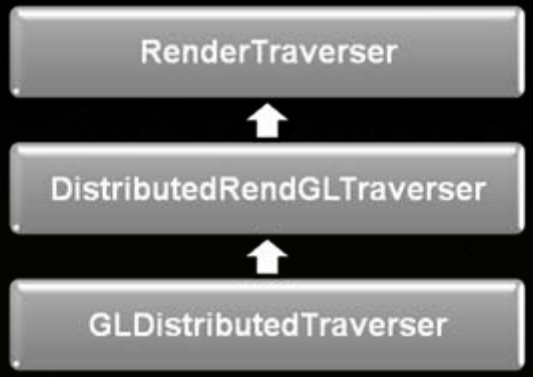


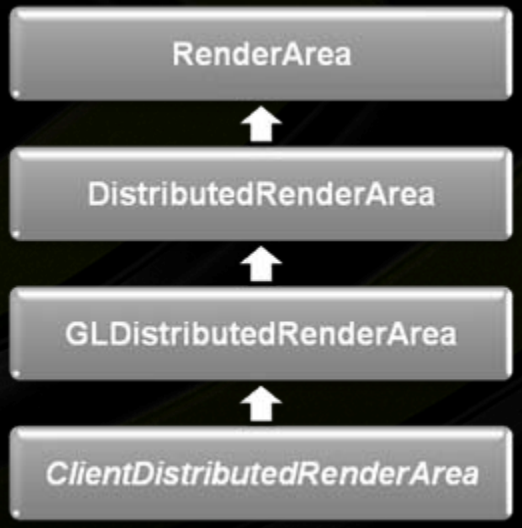
Alpha Compositing

Distributed rendering with SceniX

GLDistributedRenderArea

- Clients must derive from this and integrate with windowing system
- Optionally select GPUs to be used
- Uses MGPUSDK for image composition





DistributionTraverser

- Assigns GPUs to handle scene graph objects based on distribution scheme
- Distribution scheme assigns objects to optimally balance load on GPUs

GLDistributedTraverser

- Multiple instances (one per GPU)
- Instances run in parallel
- Each instance only renders objects assigned to its GPU
- Triggers image composition when ready

Demo Video – SceniX & CompleX

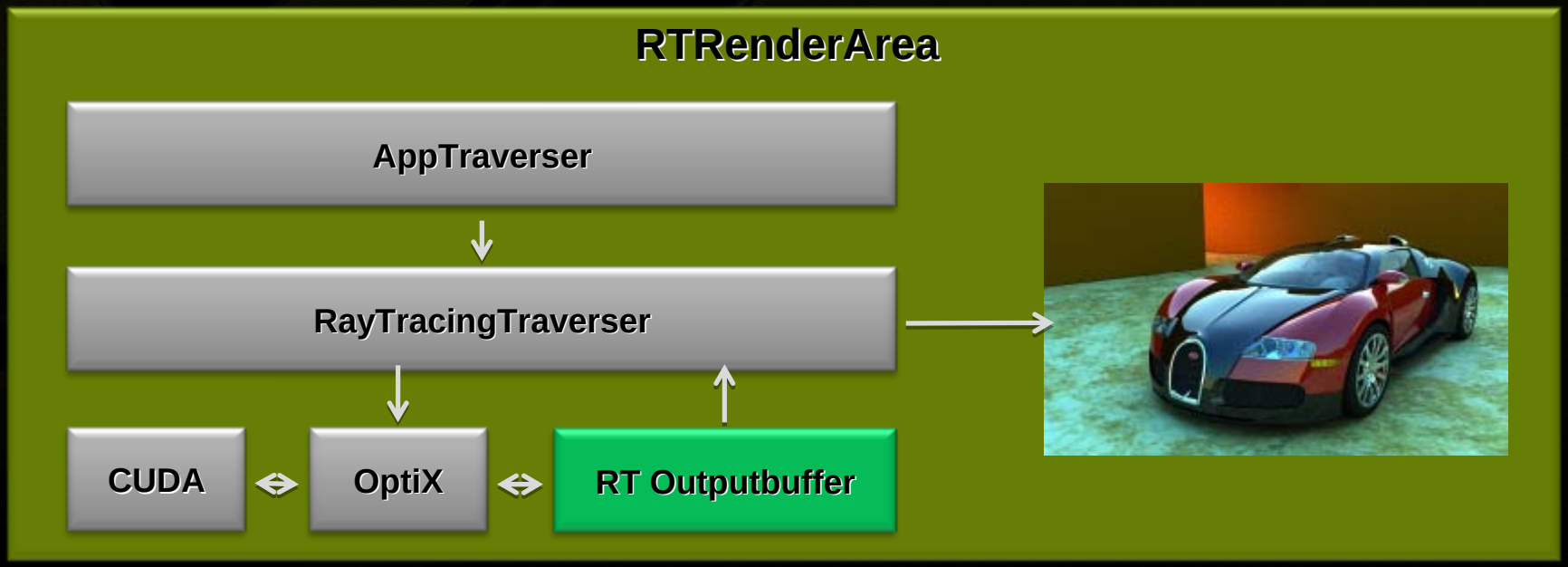
- Massive Dataset Rendering
  - Geometry data
  - Volume data

NVIDIA® OptiX™ ray tracing engine elevating interactive realism

- Programmable GPU ray tracing via a C-based SDK, akin to a programmable graphics pipeline (e.g., OpenGL)
- Easy adoption of GPU ray tracing for Application developers
- Transparent exploitation of future GPU advances
- Easily paired with GPU raster rendering for +speed
- OptiX: **interactivity** **physically correct**: mental images

Basic Structure of a Ray Tracing Render Area in SceniX

- Render Area creates e.g. OpenGL output
- RayTracingTraverser
  - Creates an RT-Outputbuffer
  - Traverses the scene graph
  - OptiX ray traces via CUDA into the RTOutputbuffer
  - Maps to OpenGL Output



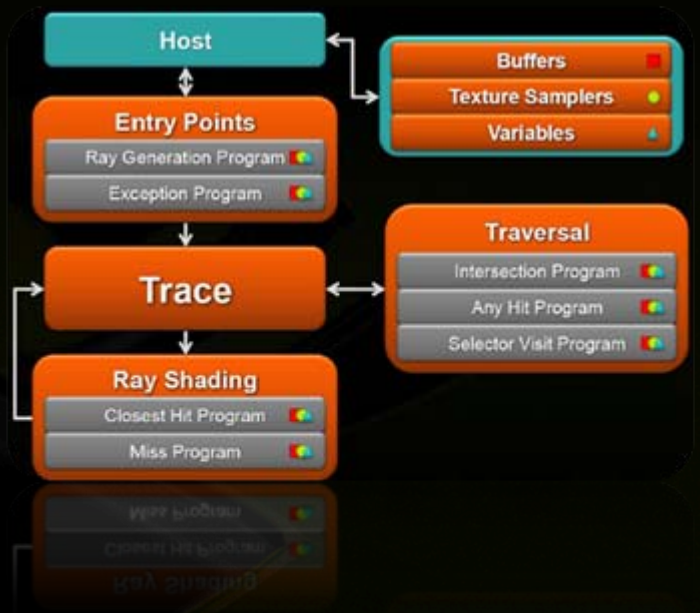
Shader Types

Shader Domain scene-wide

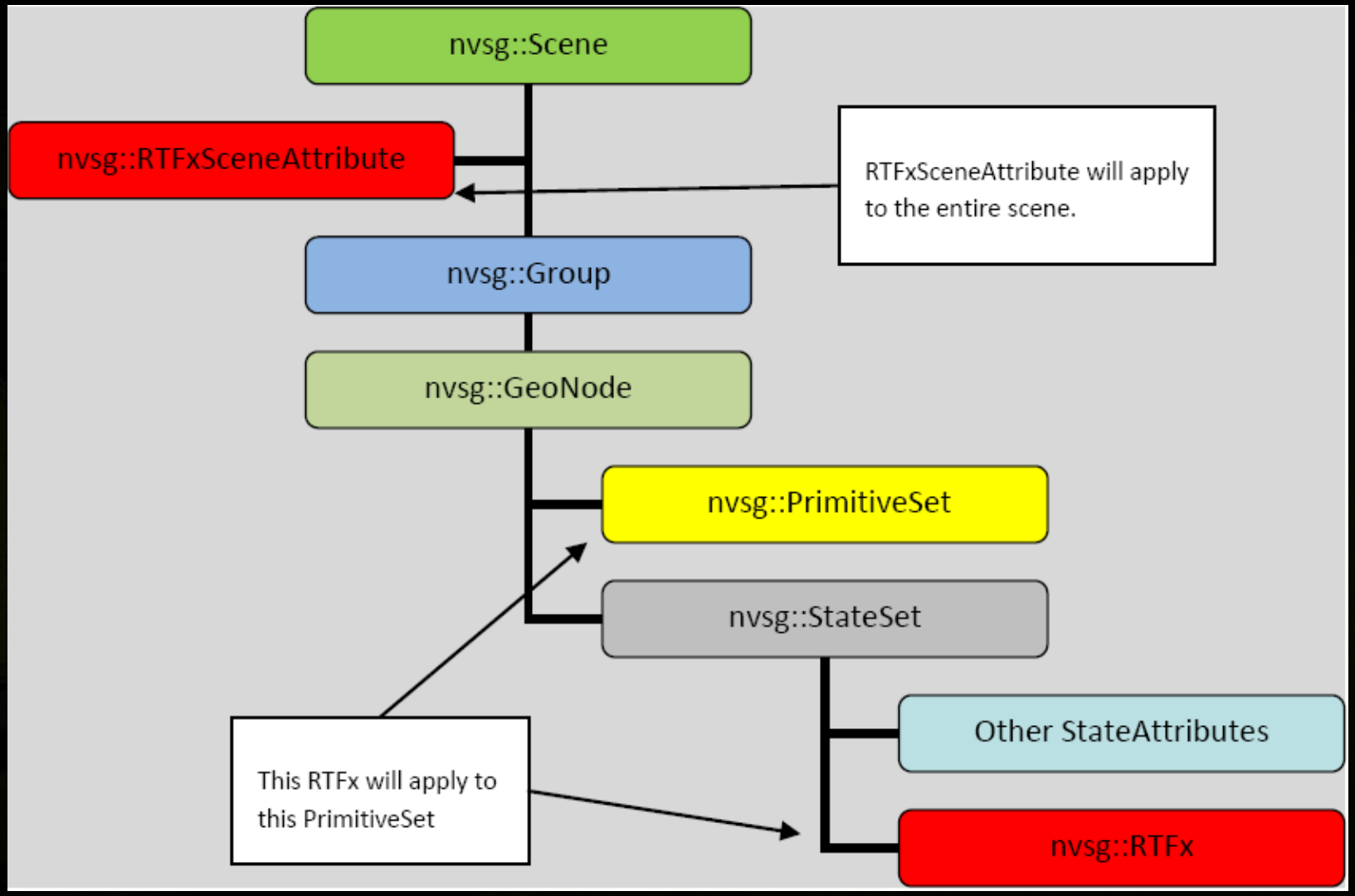
- RayGenerationPrograms
- ExceptionPrograms
- MissPrograms

Shader Domain local

- IntersectionPrograms
- BoundingBoxPrograms
- AnyHitPrograms
- ClosestHitPrograms

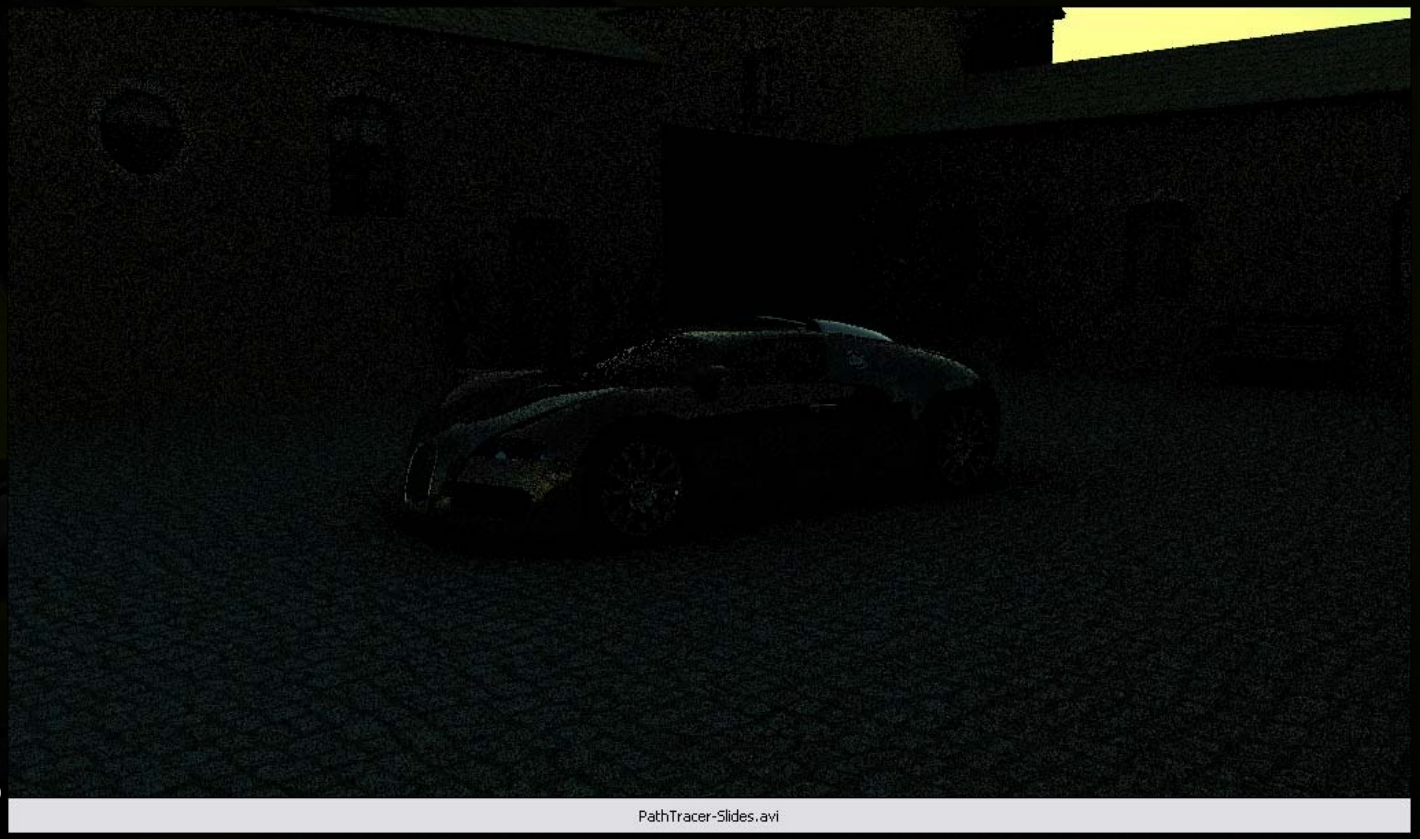


Scene Layout



Demo Video – SceniX & OptiX

- SceniX/OptiX based Path Tracer
- Global illumination
- Progressive refinement of the final image for photo quality
- Art Path: 3ds Max & Maya – Collada – SceniX/OptiX



Thanks!