Effects framework for OpenGL and OpenGL ES

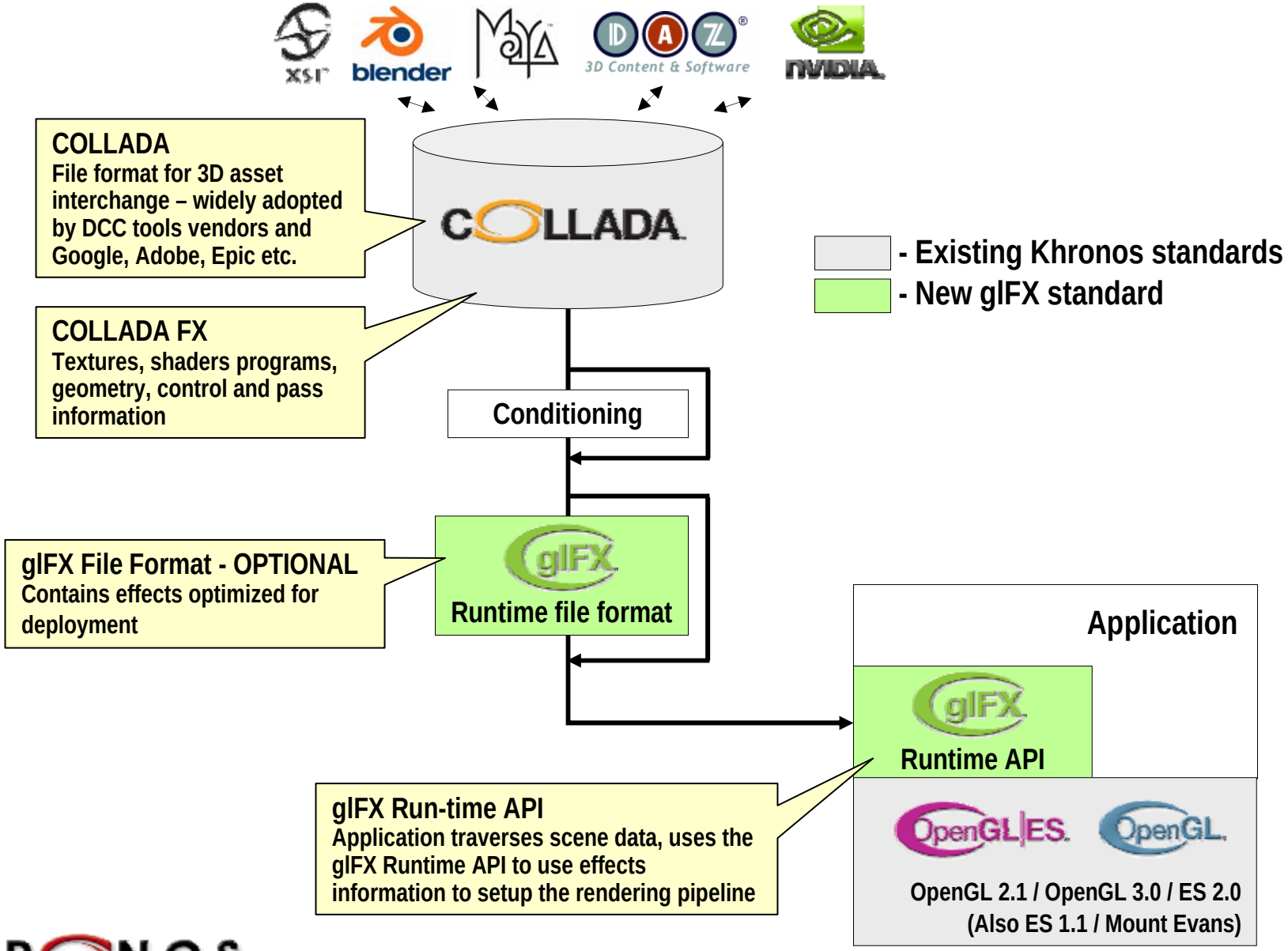
Neil Trevett
Vice President Embedded Content, NVIDIA
President, Khronos Group

OpenGL Needs an Effects Framework

- **An effects framework enables rendering effects to be easily deployed**
  - Cg has CgFX – Microsoft has FX – we need an open standard
  - NVIDIA has contributed CgFX and associated IP as an input to creating glFX
- **glFX API - a high level description of rendering effects**
  - Enables effects to be built *procedurally* in the run-time application
  - NOT just loading effects from a particular file format
- **glFX will define an OPTIONAL run-time file format**
  - Not mandated – perhaps not even official standard – but useful to exercise and test the API
- **Tight integration with content tools through COLLADA FX compatibility**
  - Multi pass, techniques, full screen effects etc.
- **Primary run-time targets are OpenGL 2.1, OpenGL 3.0 AND OpenGL ES 2.0**
  - Optimizing for OpenGL 3.0 object model
  - OpenGL ES 1.1 also supported – also Mount Evans in the future

glFX is the last piece of the toolchain
- to create first class FX in a variety of tools
- and deploy those FX on desktop AND embedded systems

glFX – Effects Framework



Deliverables

- **C language run-time glFX API specification**
  - For creating and manipulating effects and effects instances
- **A OPTIONAL glFX run-time file format – perhaps not official standard**
  - To contain effects in a form optimized for run-time deployment
- **Conformance Tests**
  - For the glFX API - using the glFX run-time file format
  - Leveraging COLLADA FX conformance tests wherever possible
- **Open Source examples – Berkeley license**
  - COLLADA FX to glFX file format run-time compiler
  - Uses of the glFX API to procedurally build effects –with glFX run-time files and without
  - glFX run-time file player
- **Delivery in 2008**
  - Still planning detailed milestones

Call for Participation in glFX

All Khronos members are invited to join the glFX Working Group
Else – join Khronos – and have a voice in how glFX evolves!