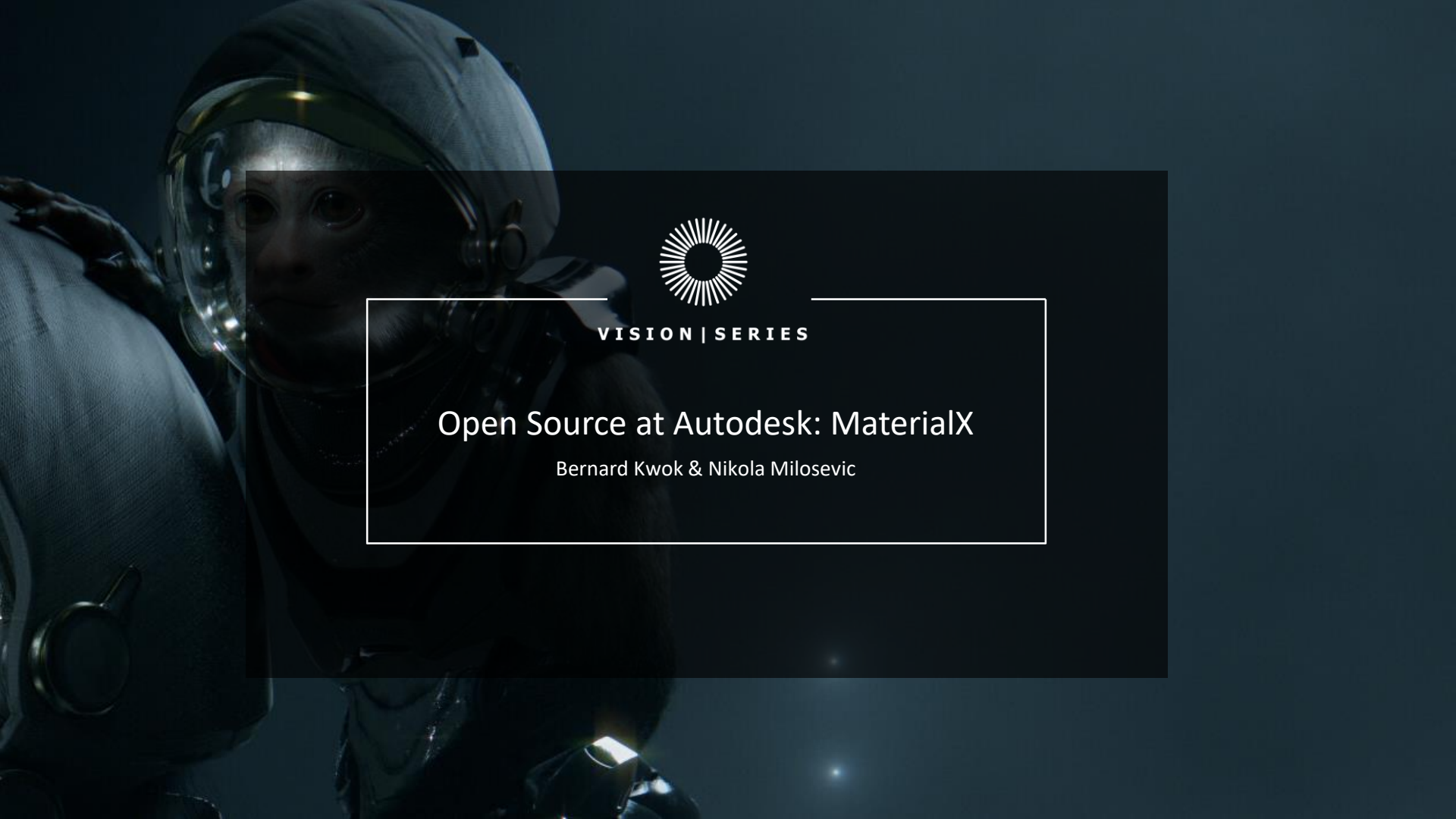


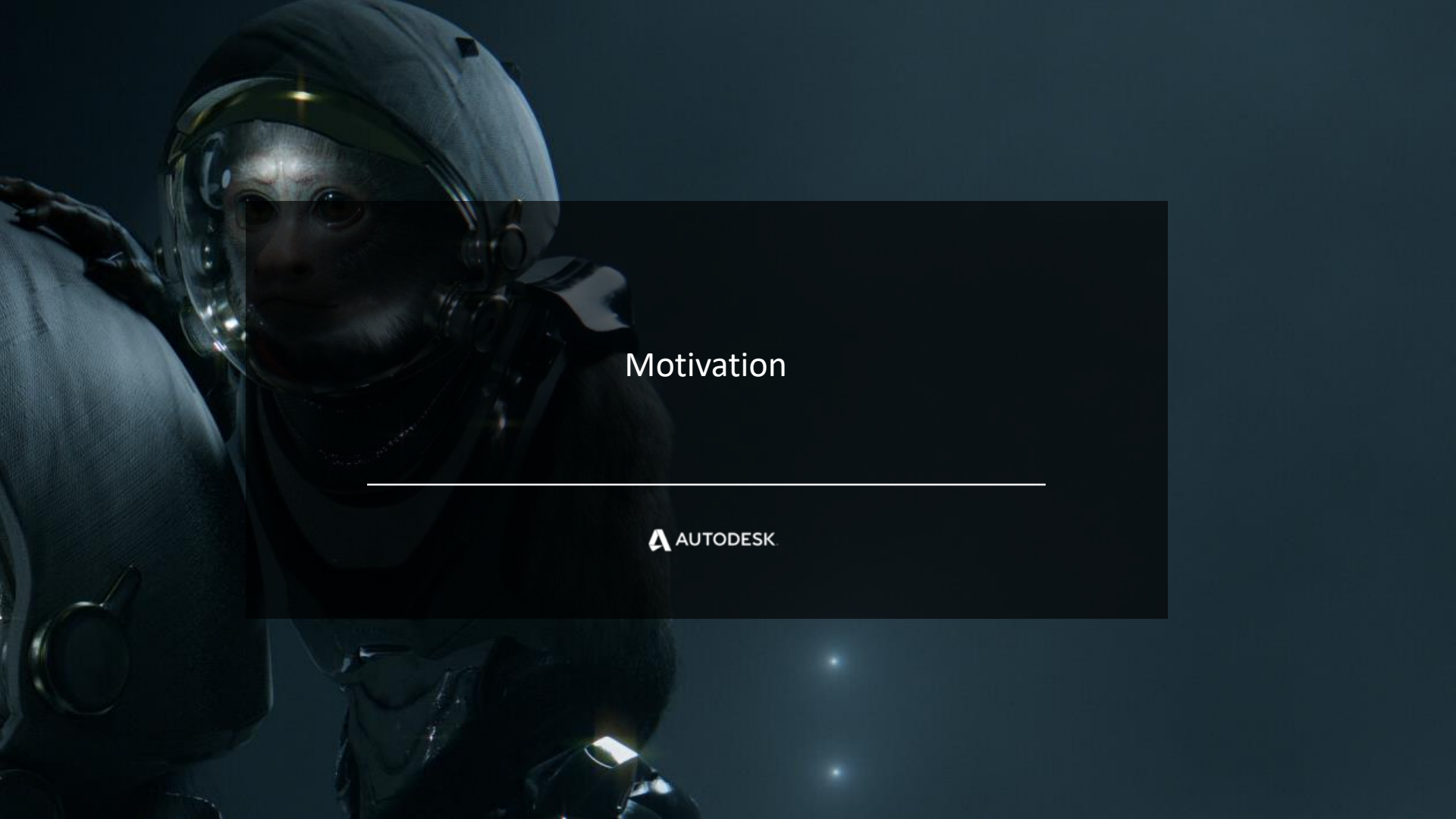
VISION | SERIES

Open Source at Autodesk: MaterialX

Bernard Kwok & Nikola Milosevic

Outline

- Motivation
- Introduction to MaterialX
- What’s new since last year
- Future work
- LookdevX Demo



Motivation

AUTODESK

Material interop challenges in Film & VFX

- Multiple studios working on the same film, even the same shot, using different tools and renderers
- Need to share assets between different vendors
- Replicating the look of a complex asset from scratch is difficult, even with lots of references
- Requires tedious, error prone, manual work
- In addition, a need to share assets between real-time and offline renderers

Material interop is a big challenge for Product Design & AEC as well

"This is our biggest problem in our 3D/CG production.

We can transfer 3D models over to almost any application, but we can't move the materials in a good way, with the high fidelity we would like to"

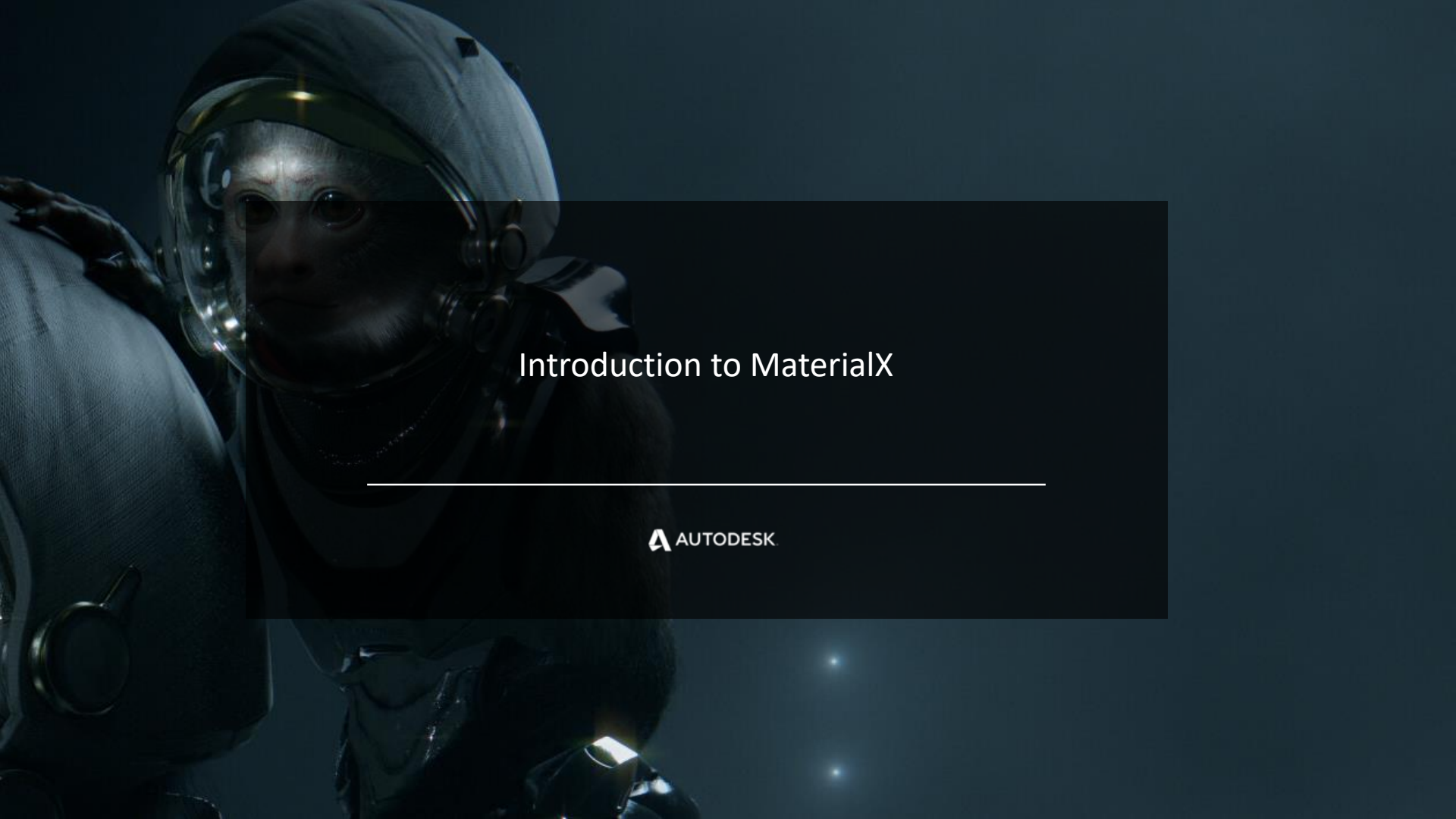
Martin Enthed
Digital Manager
IKEA Communications AB

See Martin's talk from Autodesk University 2017:
https://vimeo.com/243860738

"We see material interop as a fundamental challenge in our workflow.

What we need is a render neutral PBR based material, which allows us to exchange content between not only all the applications in our production pipeline, but with all the applications and tools our customers work with"

Mark Kauffman
Technical Lead of Project Visualization
WSP USA



Introduction to MaterialX

AUTODESK

MaterialX – An open standard for creation and transportation of complete material setups

- Express materials independently of application or renderer
- Patterns and texturing networks, and complex layered materials
- Materials assignments, looks and variations
- Launched at Lucasfilm in 2012
- First used on *Star Wars: The Force Awakens* in 2015

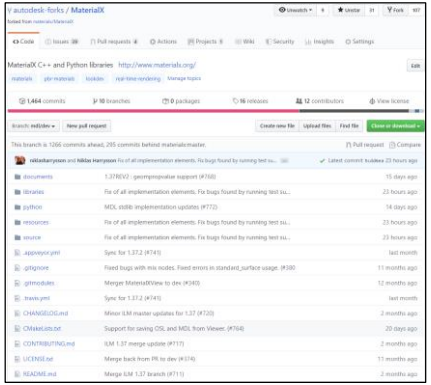
MaterialX – Collaboration

- AMG (Abstract Material Graph) started at Autodesk around 2013, with similar goals
- Lucasfilm and Autodesk joined forces in 2016 to collaborate on a single open standard
- MaterialX released open-source in 2017 https://github.com/materialx/MaterialX

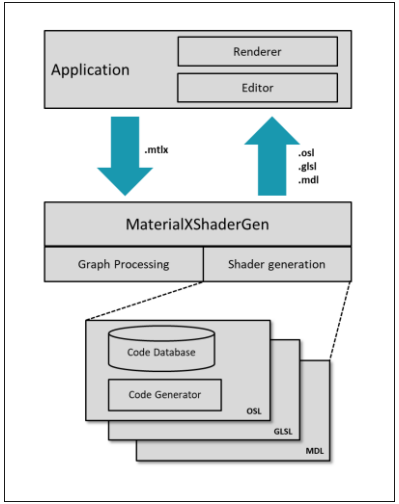
Our Contributions

Collaborate on specification and implementation

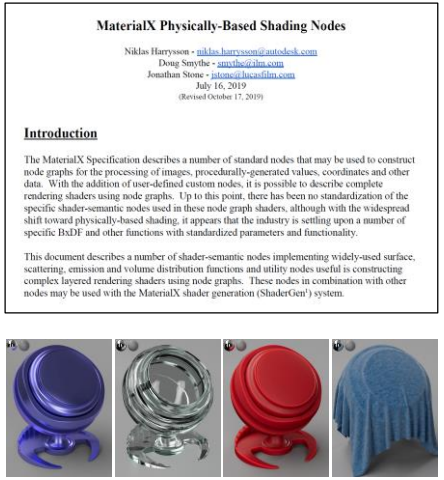
GitHub



Shader Generation



Physically Based Shading nodes



MaterialX Physically-Based Shading Nodes

Niklas Harrysson - niklas.harrysson@autodesk.com
Doug Smythe - smythe@zlm.com
Jonathan Stone - jstone@lucasfilm.com
July 16, 2019
(Revised October 17, 2019)

Introduction

The MaterialX Specification describes a number of standard nodes that may be used to construct node graphs for the processing of images, procedurally-generated values, coordinates and other data. With the addition of user-defined custom nodes, it is possible to describe complete rendering shaders using node graphs. Up to this point, there has been no standardization of the specific shader-semantic nodes used in these node graph shaders, although with the widespread shift toward physically-based shading, it appears that the industry is settling upon a number of specific BxDF and other functions with standardized parameters and functionality.

This document describes a number of shader-semantic nodes implementing widely-used surface, scattering, emission and volume distribution functions and utility nodes useful is constructing complex layered rendering shaders using node graphs. These nodes in combination with other nodes may be used with the MaterialX shader generation (ShaderGen[1]) system.

https://github.com/autodesk-forks/MaterialX

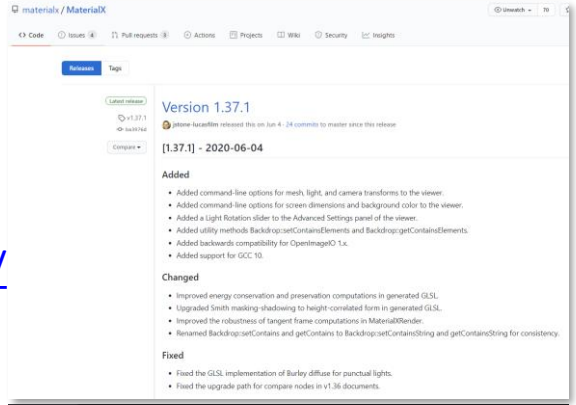
What’s New Since Last Year

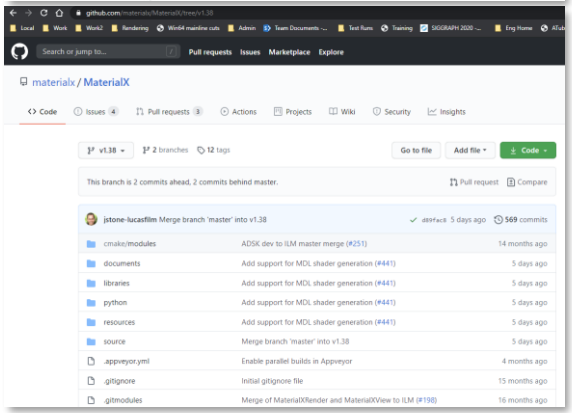
AUTODESK

MaterialX Releases

- 1.37 Revision 2 Release (https://github.com/materialx/MaterialX/releases/tag/v1.37.1)

- Specification Proposal for 1.38 (http://www.materialx.org/Specification.html)





Real-World Units

```
<unittypedef name="distance"/>
<unitdef name="UD_stdlib_distance" unittype="distance">
  <unit name="micron" scale="0.000001"/>
  <unit name="millimeter" scale="0.001"/>
  <unit name="centimeter" scale="0.01"/>
  <unit name="meter" scale="1.0"/>
  <unit name="kilometer" scale="1000.0"/>
  <unit name="inch" scale="0.0254"/>
  <unit name="foot" scale="0.3048"/>
  <unit name="yard" scale="0.9144"/>
  <unit name="mile" scale="1609.34"/>
</unitdef>
```

Autodesk Materials using real-world units. Shown in MaterialXView.

Node Parameters are Now Inputs

V1.37: Many float/color/vector node arguments were unconnectable `<parameter>`s

Having separate parameters and inputs is annoying and leads to code duplication.

V1.38: Almost all float/color/vector node arguments are now `<input>`s

V1.38: The `<parameter>` element itself is now replaced by `<input>` with a new "`uniform`" attribute in the nodedef:

```
<nodedef name="ND_image_float" node="image">
   <input name="file" type="filename" uniform="true"/>
   …
```

Material Nodes

Previously:

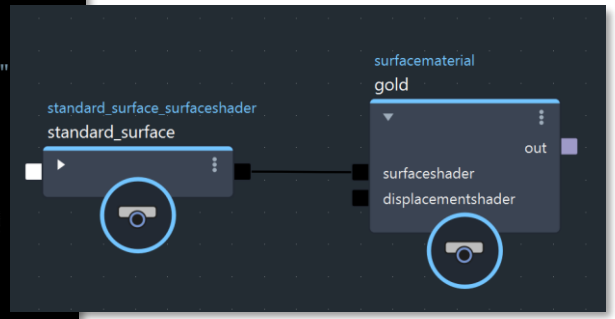
. Materials use special <material> elements, with <shaderref>s and <bindinput>s

```
<material name="gold">
  <shaderref name="SR_gold" node="standard_surface">
    <bindinput name="base" type="float" value="1"/>
    <bindinput name="base_color" type="color3" value="0.944, 0.776, 0.373"/>
    <bindinput name="specular" type="float" value="1"/>
    <bindinput name="specular_color" type="color3" value="0.998, 0.981, 0.751"
  </shaderref>
</material>
```

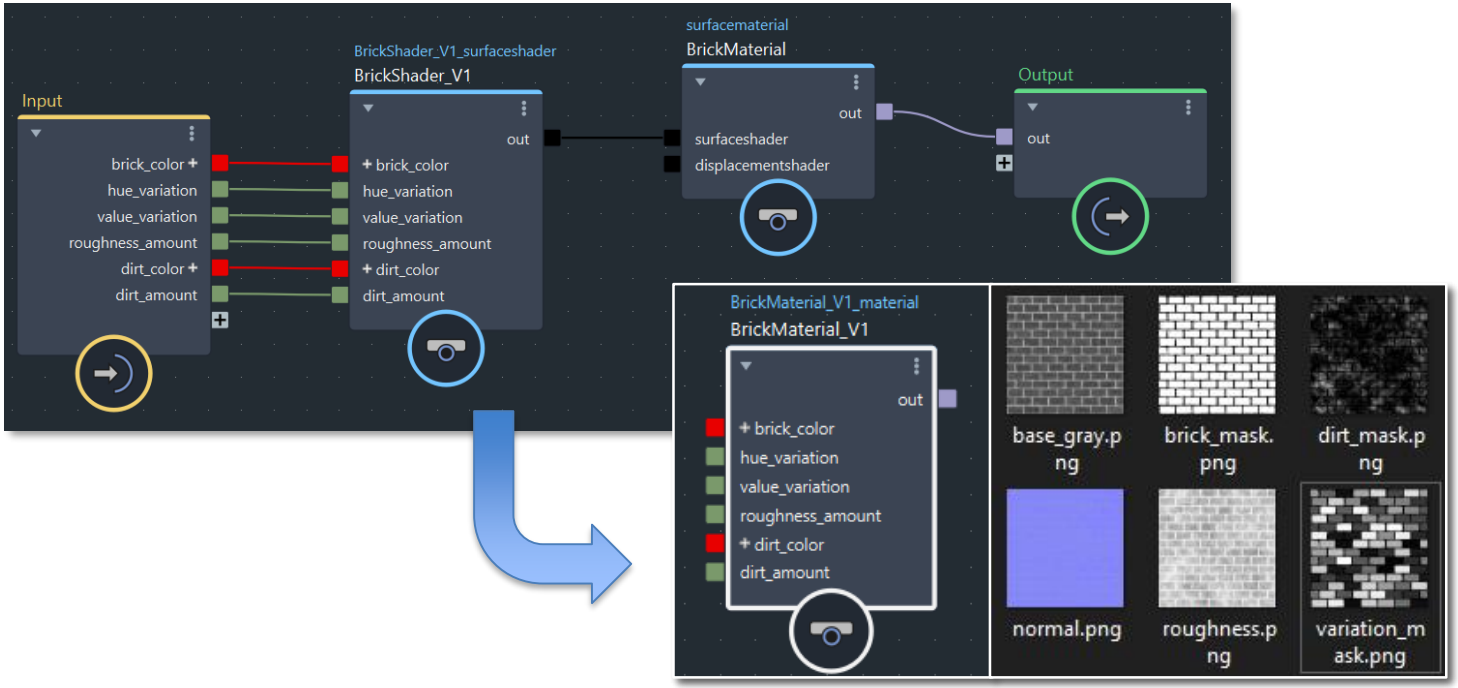
v1.38:

. Materials are just regular nodes with regular inputs connecting to shader nodes

```
<nodedef name="ND_surfacemtl" node="surfacematerial"
  <input name="surfaceshader> type="surfaceshader" value=""/>
  <input name="displacementshader> type="displacementshader" value=""/>
  <output name="out" type="material"/>
</nodedef>

<surfacematerial name="gold">
  <input name="surfaceshader" type="surfaceshader" nodename="goldsrf"/>
  <input name="displacementshader" type="displacementshader" nodename="dsp"/>
</surfacematerial>
```



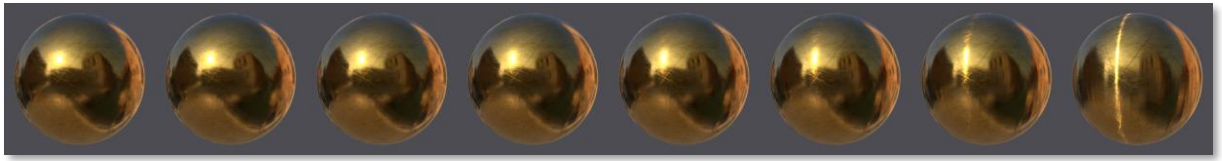
Definition Creation



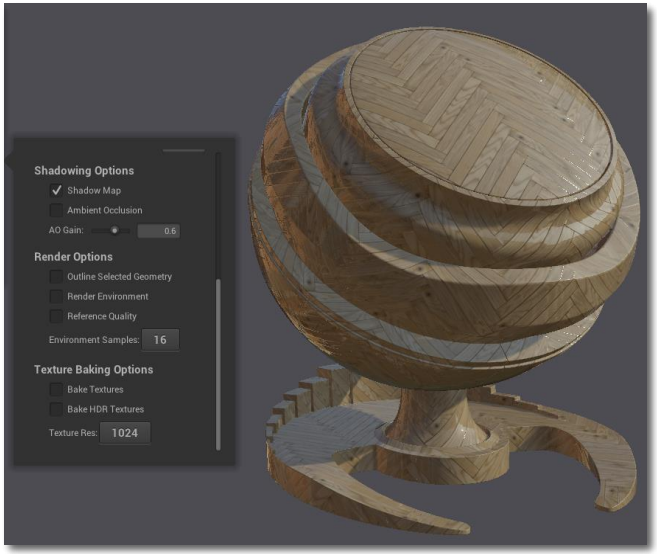
Brick material definition published based on MTLX document, and textures created in Substance Designer.

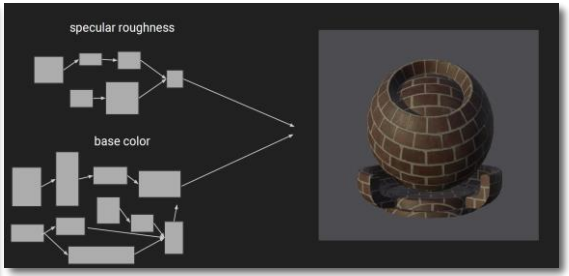
MaterialX Viewer Updates

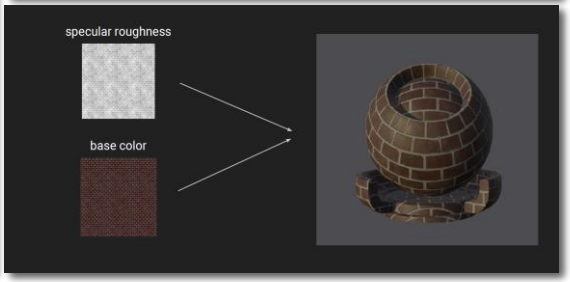
Wedge Rendering



Texture Baking

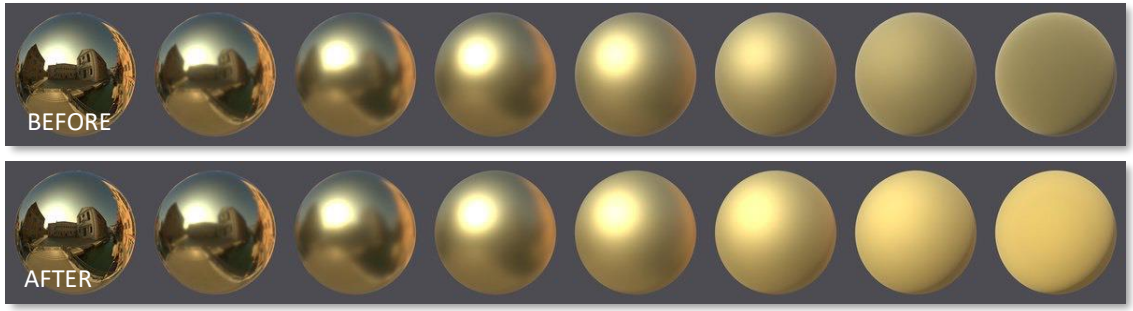






Physically Based Shading – BSDF Improvements

Multiple Scattering Energy compensation

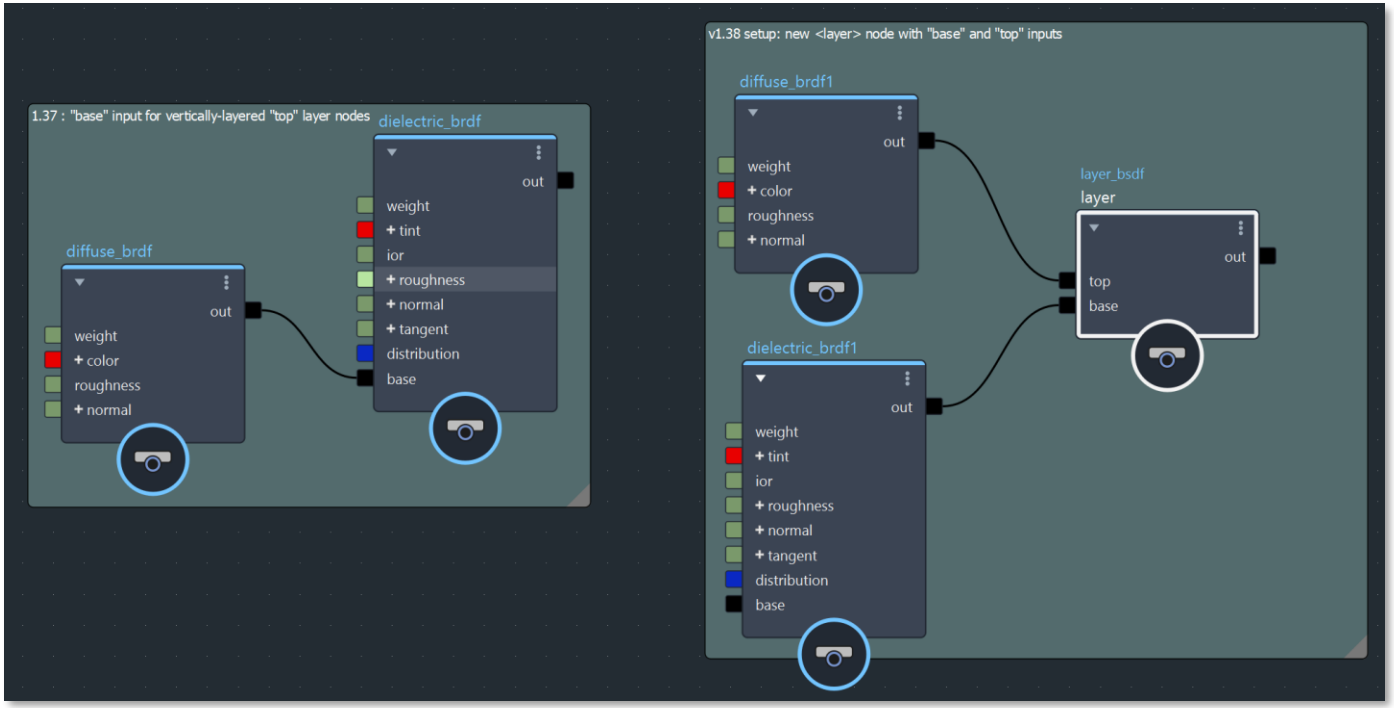


Sheen BRDF for cloth / backscattering

Thin-Film BRDF for iridescence effects

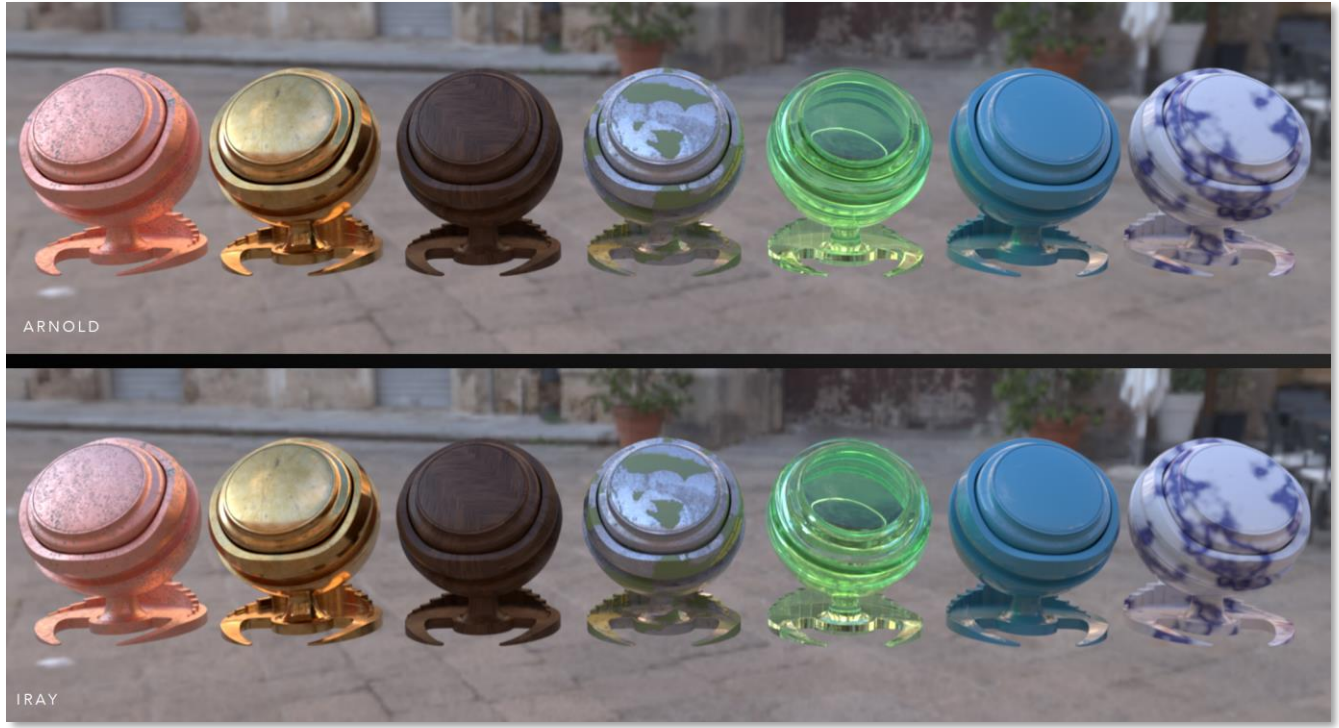
https://www.materialx.org/assets/MaterialX.v1.37REV2.PBRSpec.pdf

Physically Based Shading – Layering Operator



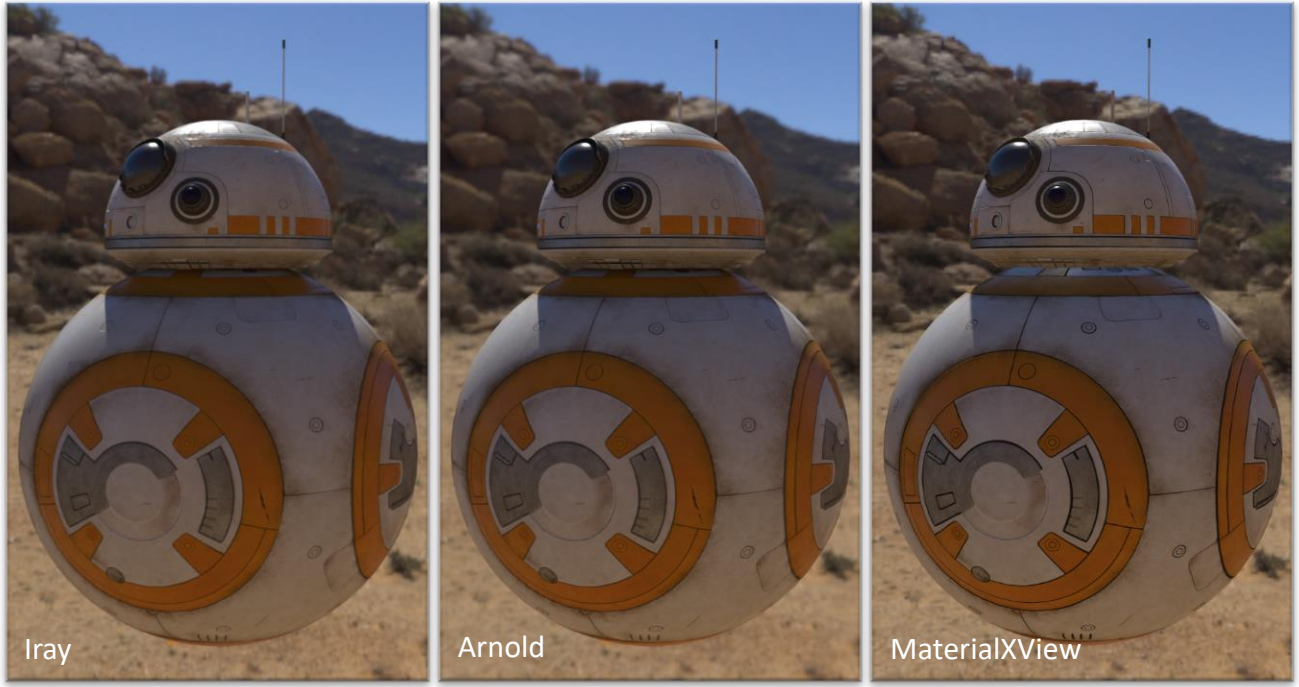
https://www.materialx.org/assets/MaterialX.v1.37REV2.PBRSpec.pdf

MDL Shader Generation

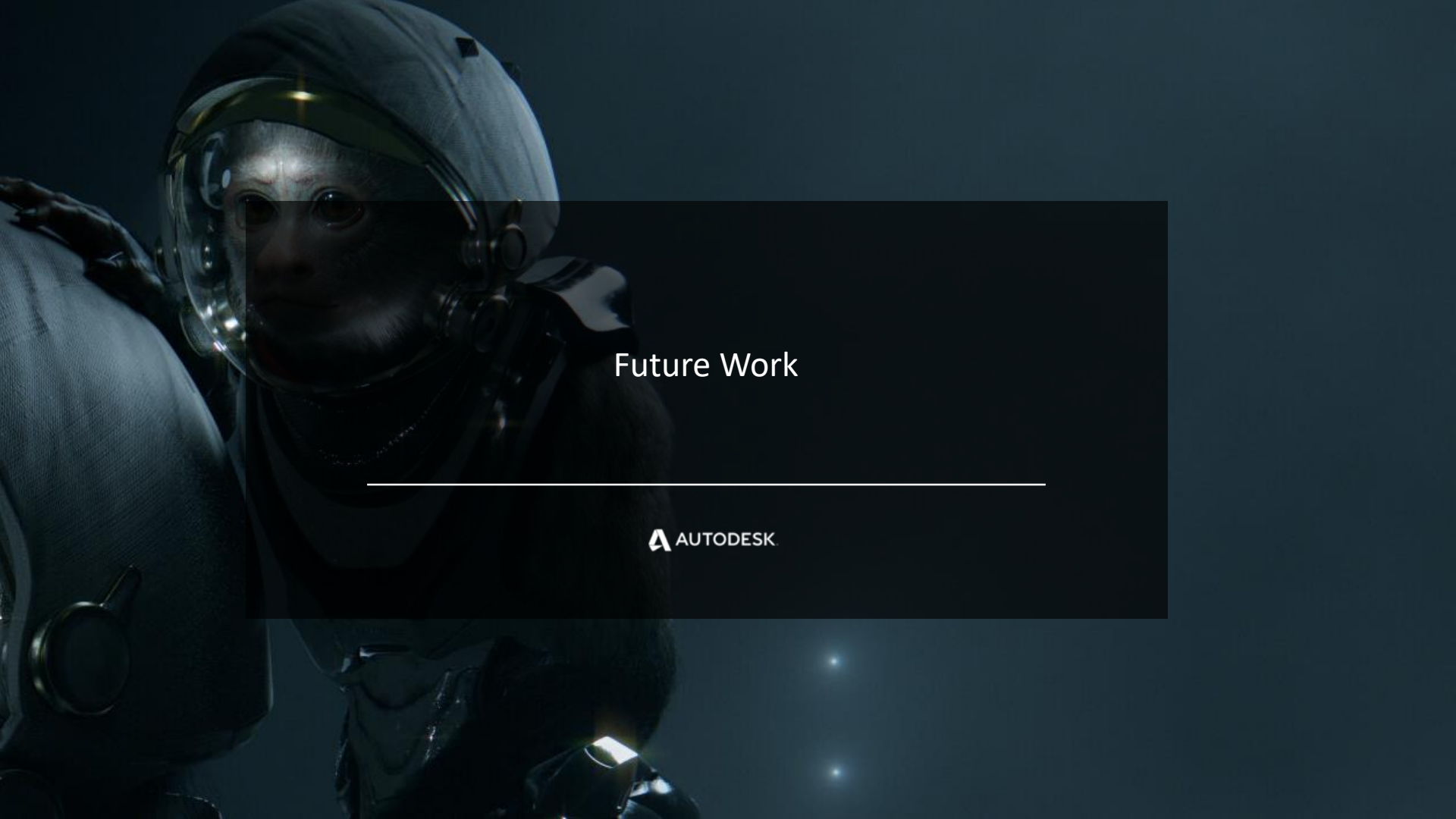


https://www.nvidia.com/en-us/gtc/on-demand/?search=s21469

MDL Shader Generation



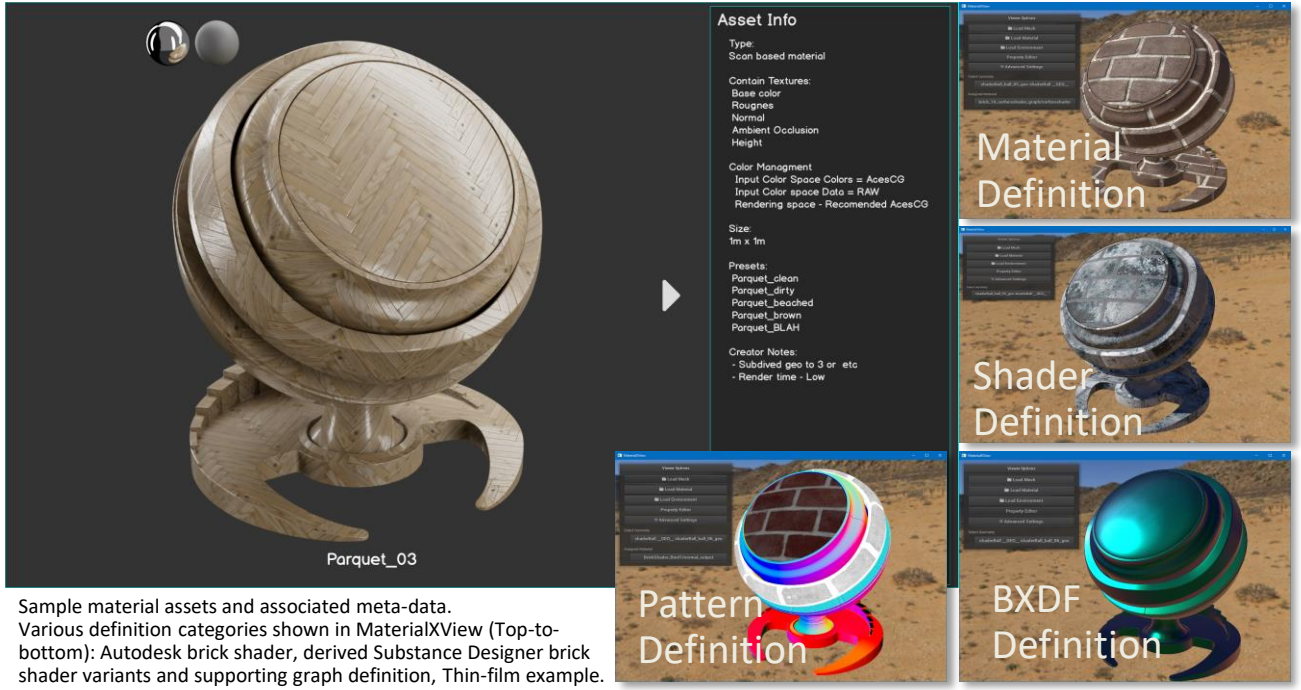
https://www.nvidia.com/en-us/gtc/on-demand/?search=s21469



Future Work

AUTODESK

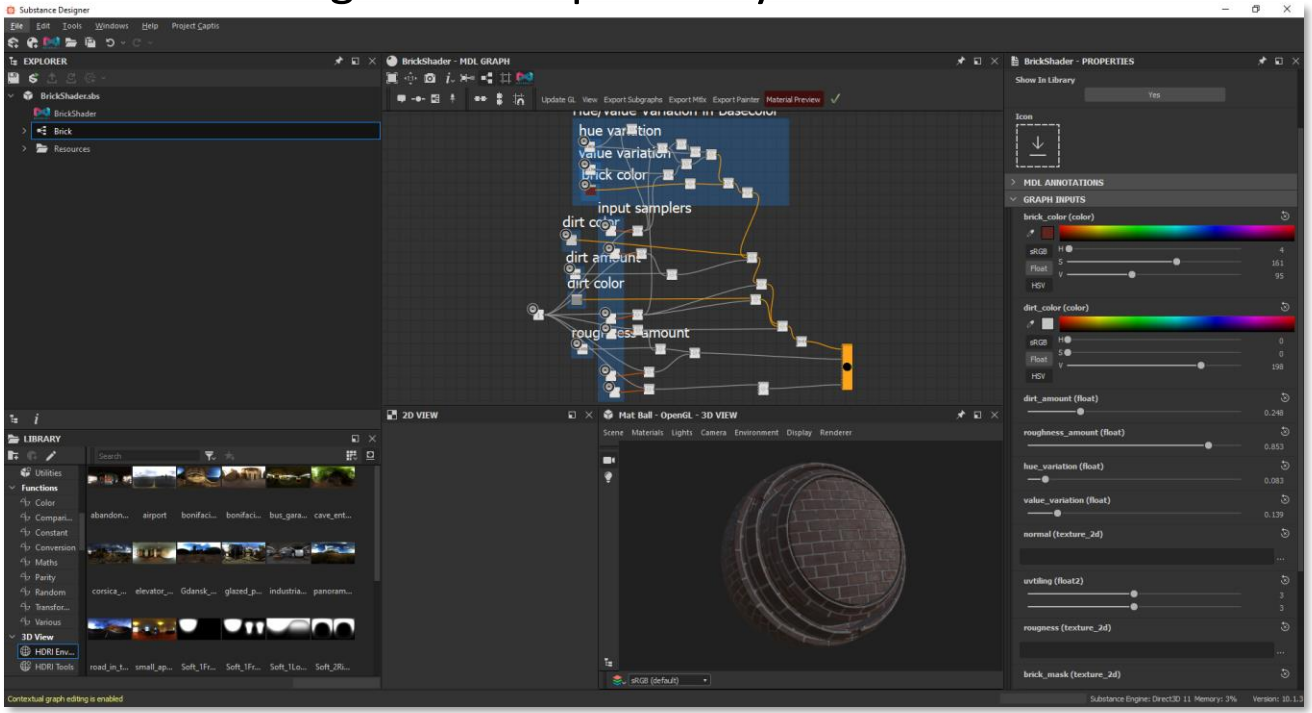
Libraries and Definitions



Sample material assets and associated meta-data.
Various definition categories shown in MaterialXView (Top-to-bottom): Autodesk brick shader, derived Substance Designer brick shader variants and supporting graph definition, Thin-film example.
Geometry assets created in Maya.

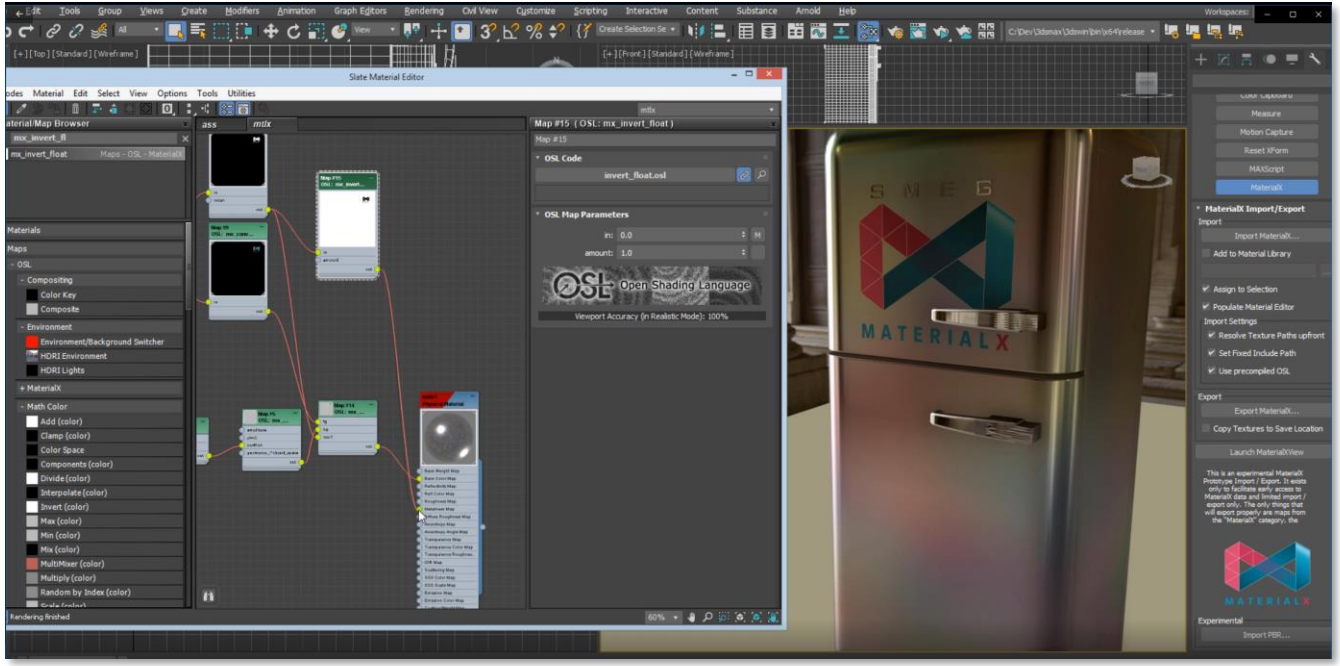
Autodesk Material Library

Substance Designer Interoperability

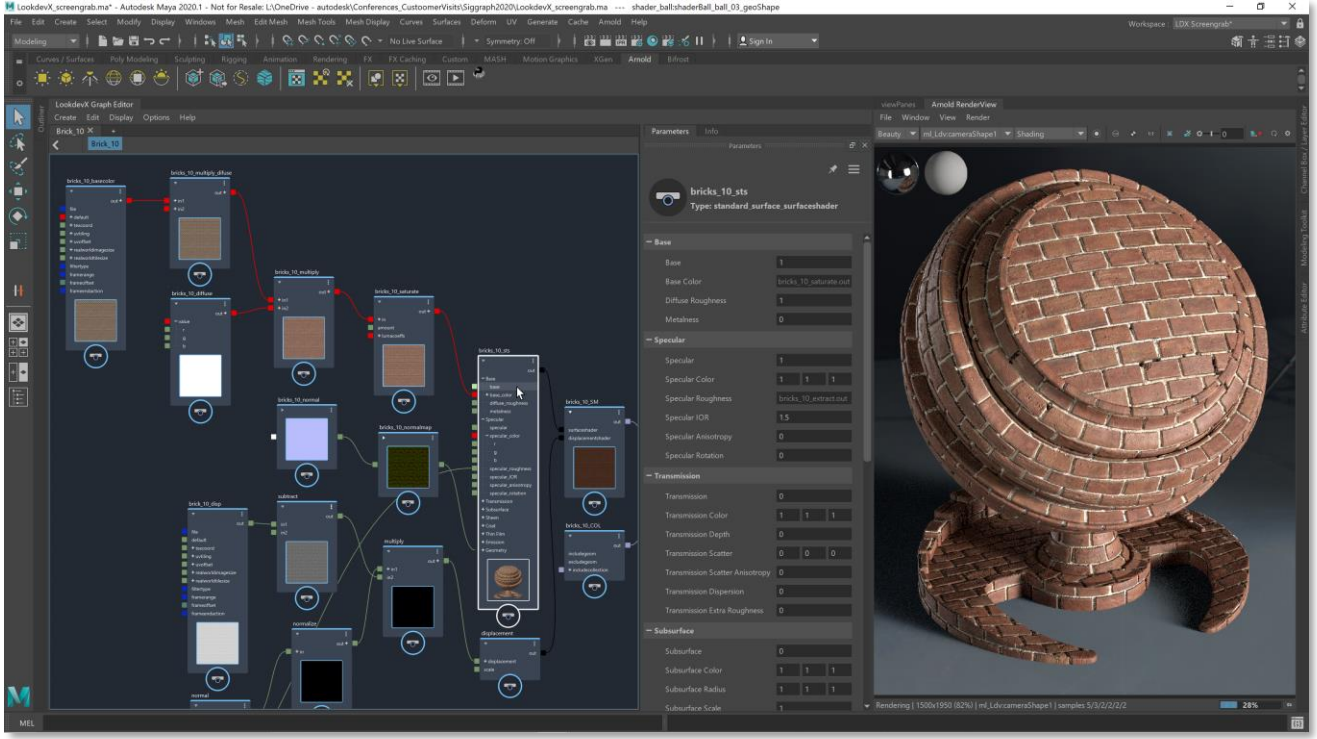


https://share.substance3d.com/libraries/6111

3ds Max Interoperability



Maya and Arnold Interoperability



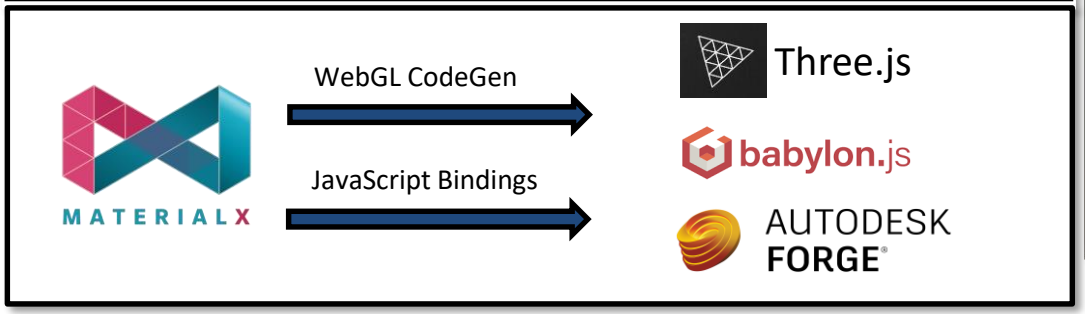
Shader Generation: SPIR-V

Early results. See [Autodesk Vision Series](#) Session: *3ds Max: Open Standards & Next Generation Viewport Framework*

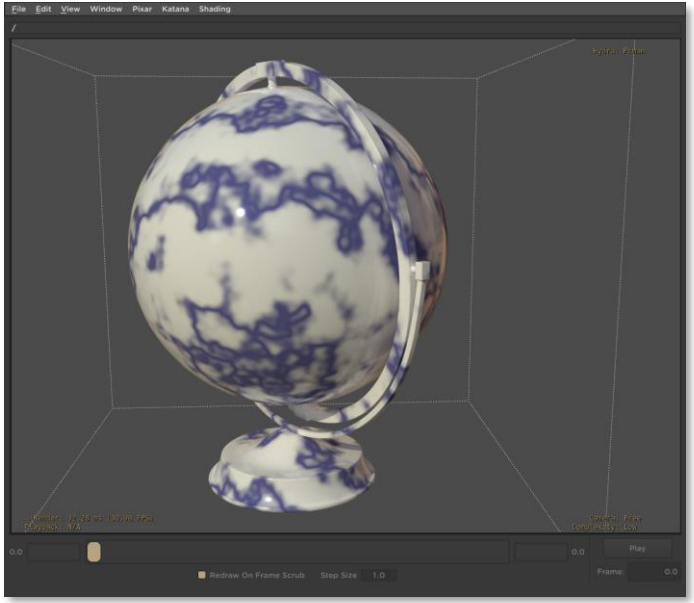
Web Support

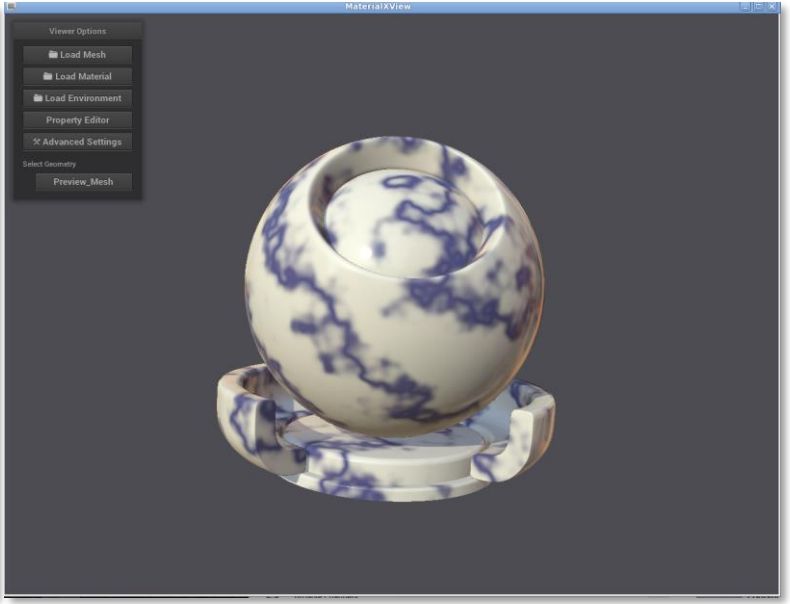
[Javascript support in progress](#) GitHub

```
import Module from 'JsMaterialX.js’;

let mx, doc;
mx = await initiMaterialX();
doc = mx.createDocument();

let nodeGraph;
nodeGraph = doc.addNodeGraph();

let output1, output2, constant, image;
constant = nodeGraph.addNode('constant');
image = nodeGraph.addNode('image');
output = nodeGraph.addOutput();
output.setConnectedNode(image);
```



MaterialX in USD/Hydra





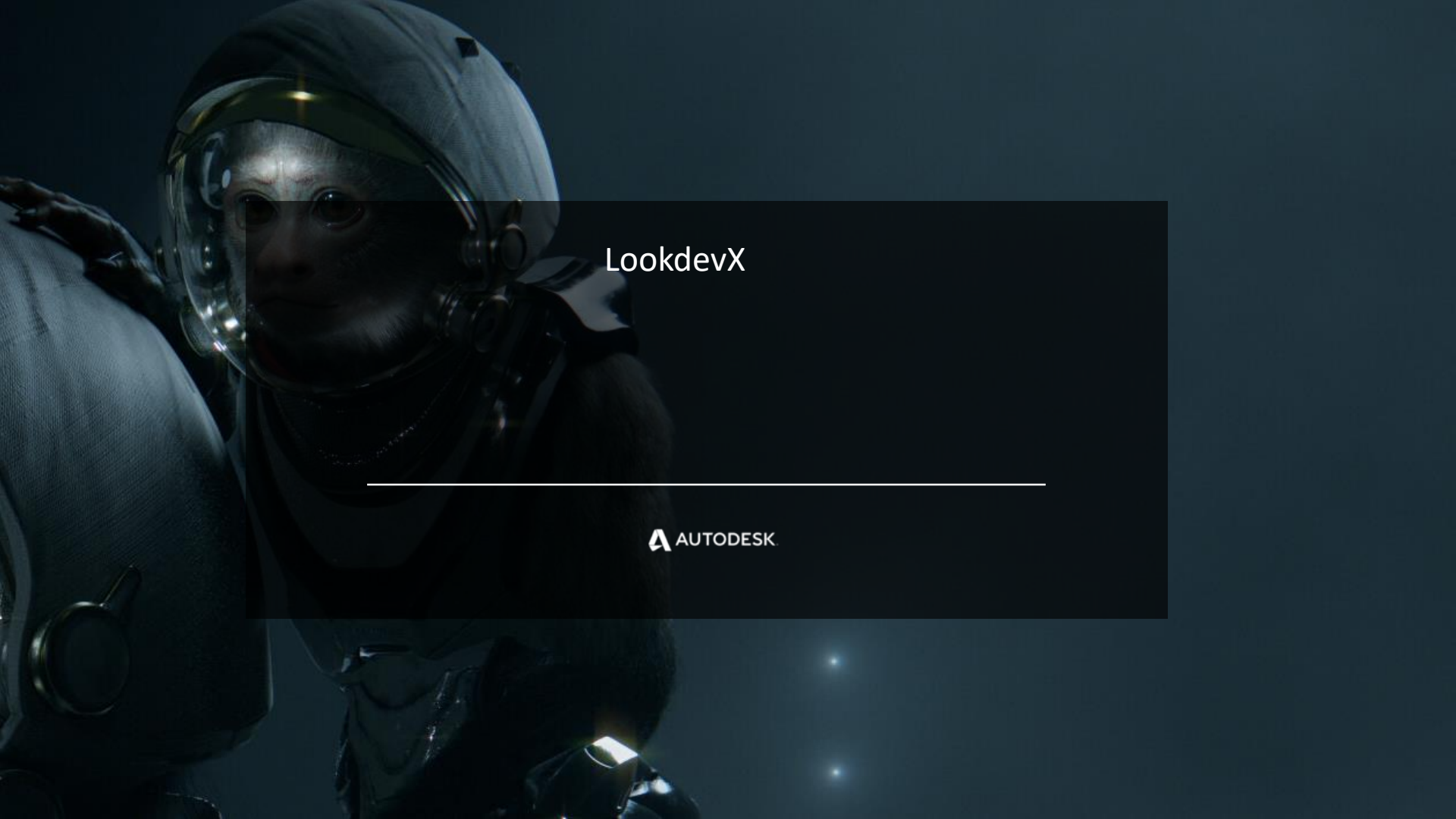
Material rendering in hdPRman (left) and MaterialXView (right) © Pixar

MaterialX in USD/Hydra

MaterialX and Color Management

+

OCIO-2 integration



LookdevX

AUTODESK

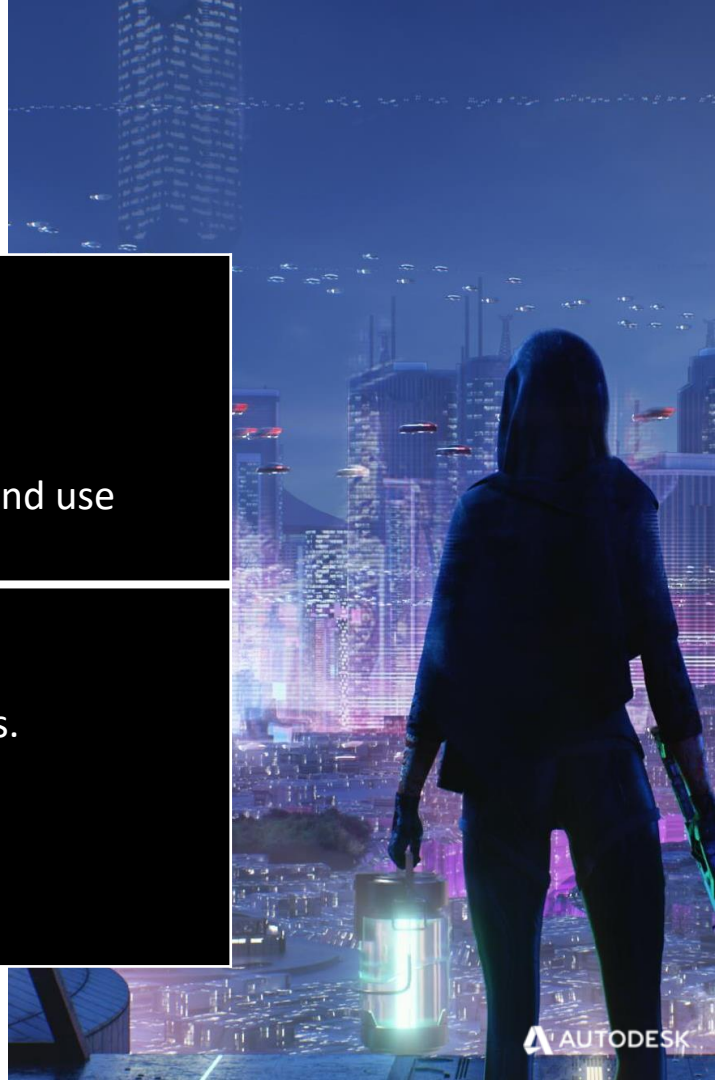
What is LookdevX?

Users

- Tool for interactive nodegraph and definition creation
- Tool for Interactive material and look creation
- Able to integrate into a DCC for material assignment and use scene content.

Integrators

- Interoperability between DCCs and 3rd party renderers.
- Currently an Inner source project
- Built as a shared component with reusable modules.



Acknowledgements

- *Autodesk MaterialX*: Niklas Harrysson, Jonathan Feldstein, Adam Felt, Ashwin Bhat, Nicolas Savva, Fedor Nikolayev, Tom Varik, Henrik Edström, Eric Bourque
- *Autodesk 3ds Max*: Zap Andersson, Neil Hazzard
- *Autodesk Arnold*: Orn Gunnarsson, Krishnan Chunangad Ramachandran
- *nVidia MDL*: Lutz Kettner, Jan Jordan, Kai Rohmer and Sandra Pappenguth
- *Adobe Substance*: David Larsson
- *Lucasfilm*: Doug Smythe, Jonathan Stone


AUTODESK®