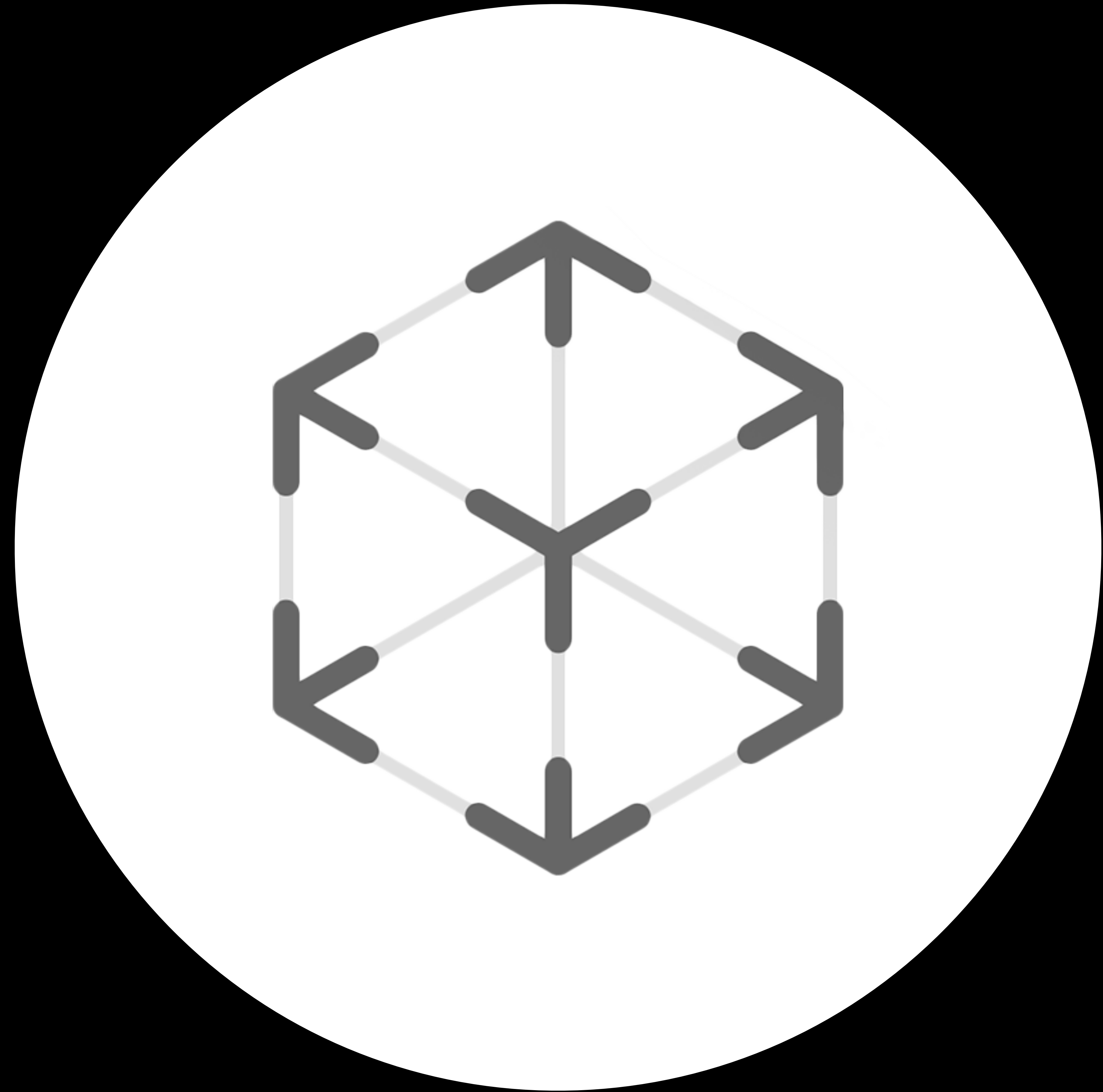


#WWDC19

Working with USD

Denis Kovacs, GPU Software Engineer
Chloe Moore, GPU Software Engineer











iOS Integration



Messages



Mail



Notes



News



Safari



Files



3rd Party Apps



USD

What is USD?

Workflows

USD essentials

USD highlights

What is USD?

Workflows

USD essentials

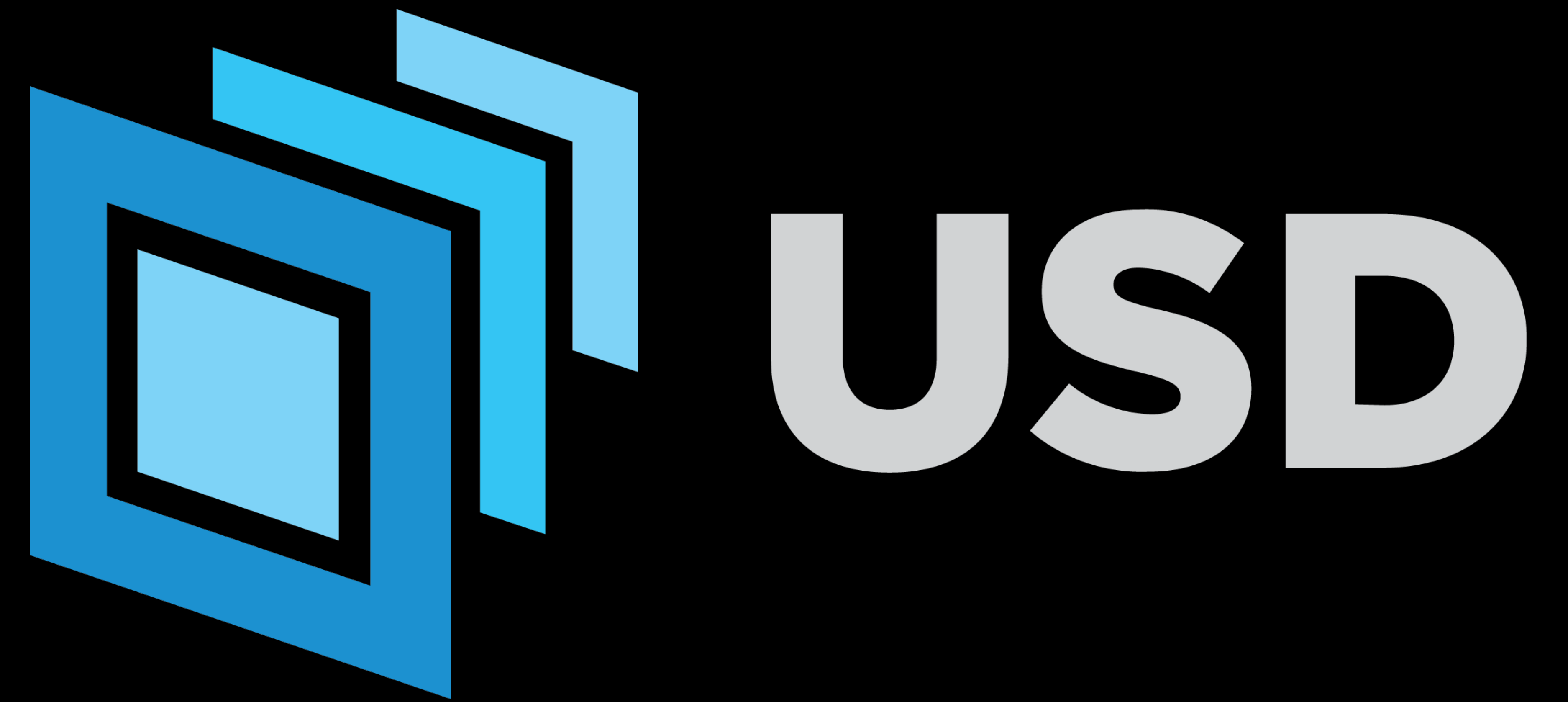
USD highlights

Universal Scene Description

Developed by Pixar

- 3D file format
- C++ library with Python bindings
 - Read/write
 - Composition engine
 - And much more...

Focus on speed, scalability, collaboration



USD File Types



Plain text



Binary



Plain text or binary

USD File Types



USDA

Plain text



USDC

Binary



USD

Plain text or binary



USDZ

Archive

What Is usdz?

Distribution format for USD

Compact single file

Optimized for sharing

Basis of AR Quick Look

Supported on iOS, macOS, and tvOS



Anatomy of usdz

Uncompressed archive

- 64-byte aligned files (memory-mapping)

Scene description file(s)

- .usda, .usd, .usdc, .usdz

Textures

- .png or .jpeg



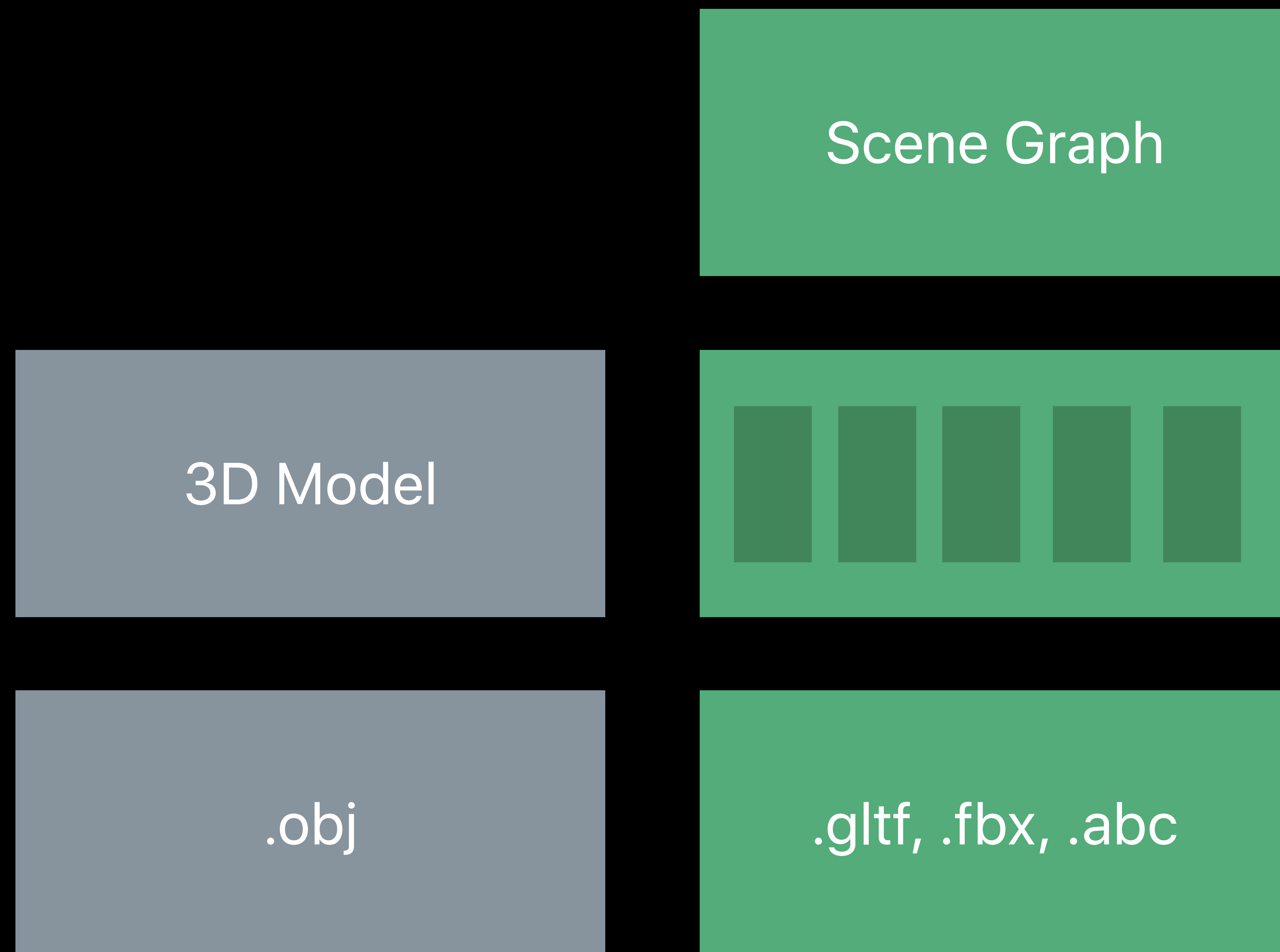
model.usdz

3D Formats

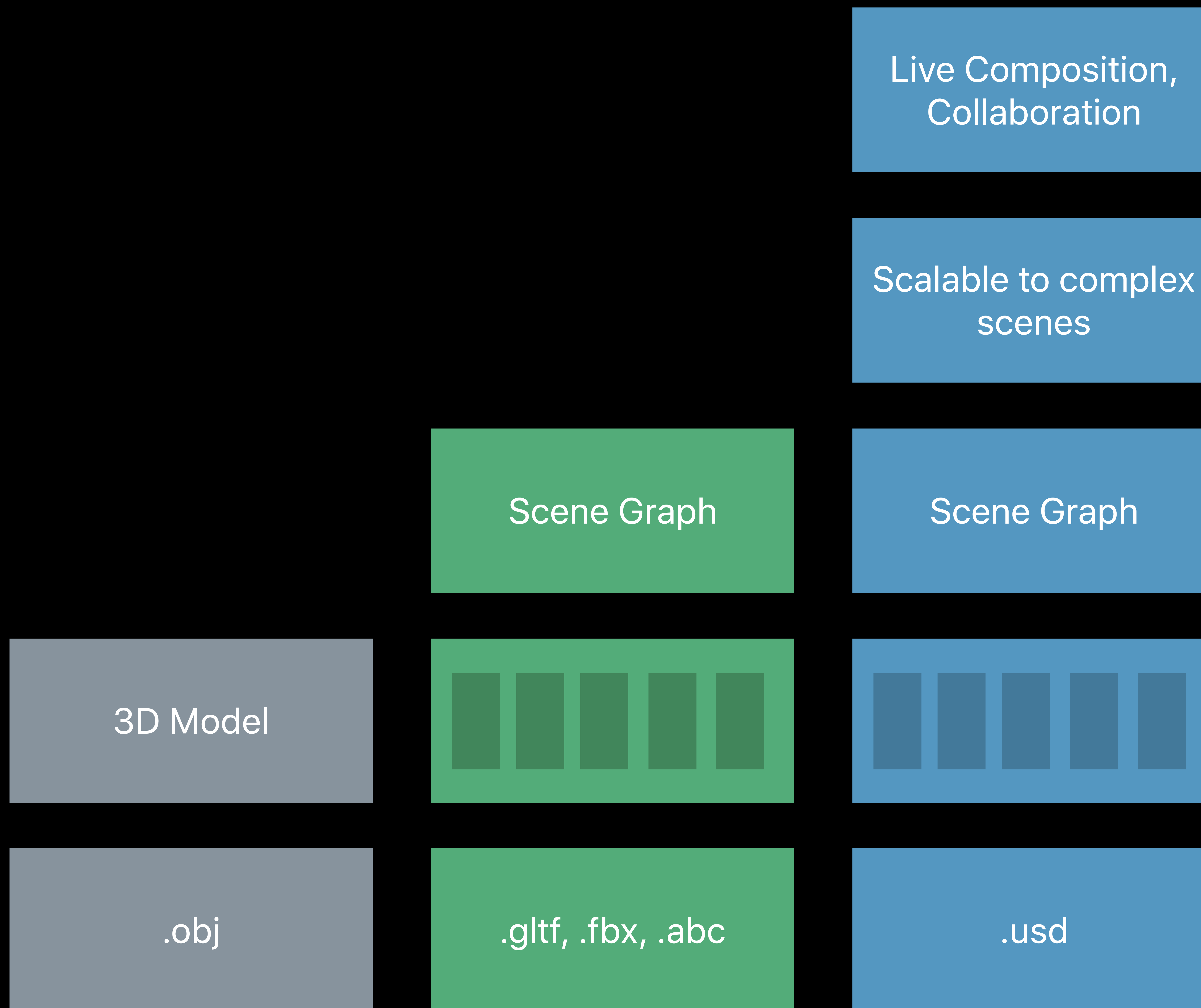
3D Model

.obj

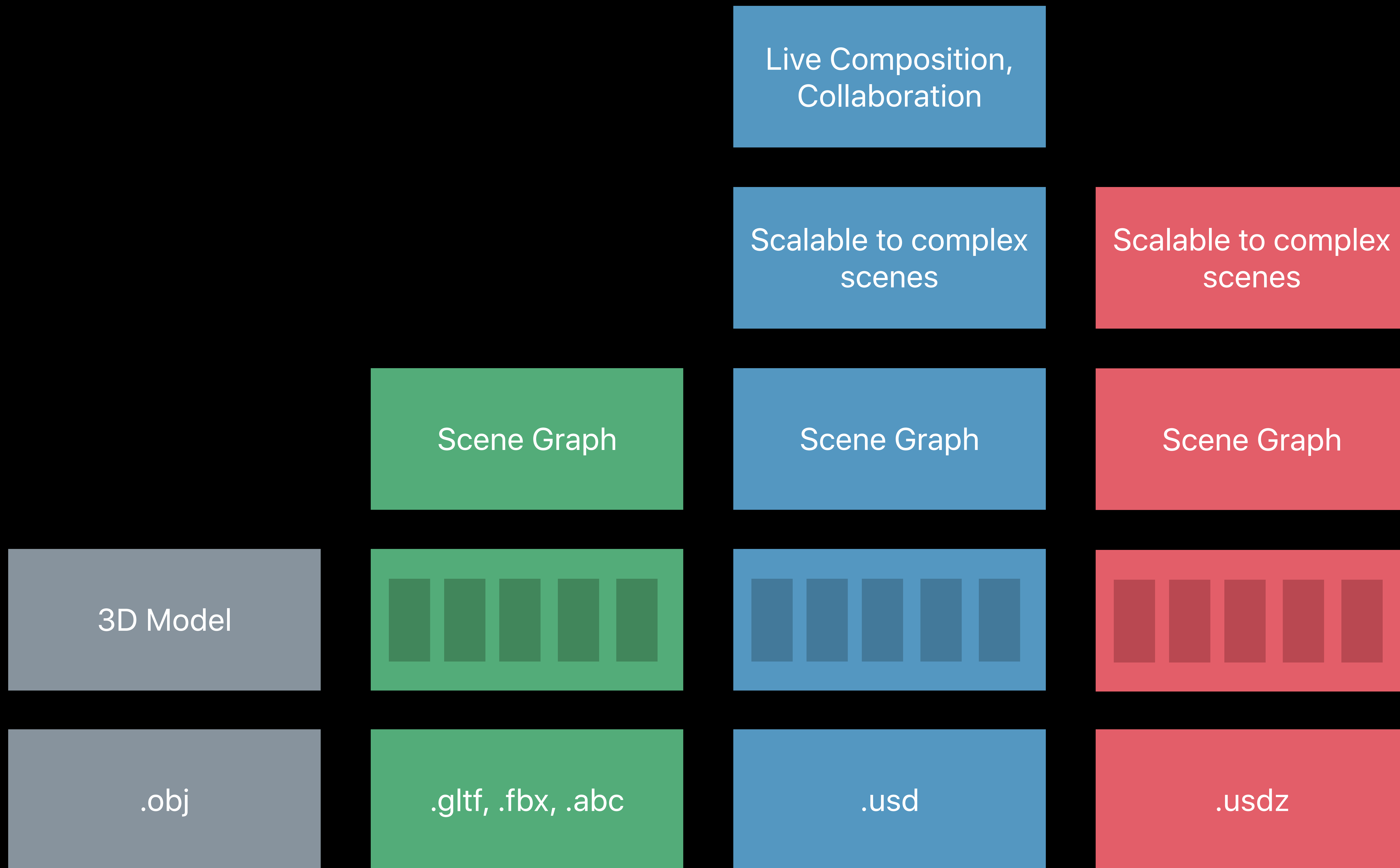
3D Formats



3D Formats



3D Formats



What is USD?

Workflows

USD essentials

USD highlights

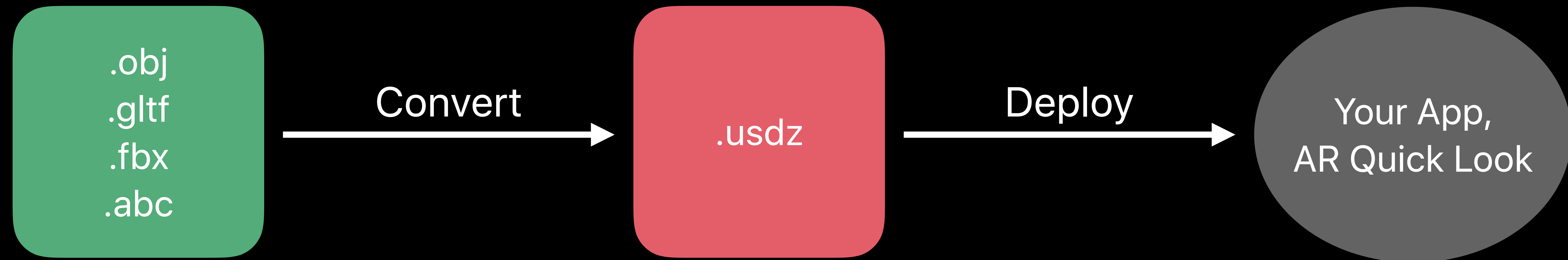
Convert

Create

Convert

Create

Converting 3D Content



usdzconvert



NEW

New command-line tool

.obj, .fbx, .glTF, .abc, etc.

Performs asset validation

Python based

Platform independent


```
% usdzconvert esprit.gltf
```

```
usdzconvert: converting file: esprit.gltf
```

```
usdzconvert: converted usdz file: esprit.usdz
```

```
usdARKitChecker: [Pass] esprit.usdz
```

```
% usdzconvert -h
```

```
[...]
```

```
% usdzconvert esprit.obj -diffuseColor d.png -normal n.png -metallic m.png -roughness r.png
```

```
[...]
```

```
% usdzconvert esprit.gltf
usdzconvert: converting file: esprit.gltf
usdzconvert: converted usdz file: esprit.usdz
usdARKitChecker: [Pass] esprit.usdz
```

```
% usdzconvert -h
[...]
```

```
% usdzconvert esprit.obj -diffuseColor d.png -normal n.png -metallic m.png -roughness r.png
[...]
```

```
% usdzconvert esprit.gltf
usdzconvert: converting file: esprit.gltf
usdzconvert: converted usdz file: esprit.usdz
usdARKitChecker: [Pass] esprit.usdz
```

```
% usdzconvert -h
[...]
```

```
% usdzconvert esprit.obj -diffuseColor d.png -normal n.png -metallic m.png -roughness r.png
[...]
```

```
% usdzconvert esprit.gltf
usdzconvert: converting file: esprit.gltf
usdzconvert: converted usdz file: esprit.usdz
usdARKitChecker: [Pass] esprit.usdz
```


```
% usdzconvert -h
[...]
```

```
% usdzconvert esprit.obj -diffuseColor d.png -normal n.png -metallic m.png -roughness r.png
[...]
```


developer.apple.com

Augmented Reality


Overview ARKit 3 Reality Composer AR Quick Look Resources




[Bethesda Gear ↗](#)



[Nomatic ↗](#)



[FURNI ↗](#)



usdz Tools

Download essential Python-based tools for generating, validating, and inspecting usdz files. Also includes a converter that creates usdz from other 3D file formats along with Pixar's USD library and sample scripts.

[Download usdz tools ↴](#)


Apple Developer > Augmented Reality > AR Quick Look

developer.apple.com/augmented-reality/quick-look/


developer.apple.com

Augmented Reality


Overview ARKit 3 Reality Composer AR Quick Look Resources




[Bethesda Gear ↗](#)



[Nomatic ↗](#)



[FURNI ↗](#)



usdz Tools

Download essential Python-based tools for generating, validating, and inspecting usdz files. Also includes a converter that creates usdz from other 3D file formats along with Pixar's USD library and sample scripts.

[Download usdz tools ↴](#)

Developer > Augmented Reality > AR Quick Look

developer.apple.com/augmented-reality/quick-look/

```
% usdcat model.usdz
[...]

% usdtree model.usdz
/
  `--Parent [def Xform] (kind = component)
    |--CubeChild [def Cube]
    |--SphereChild [def Sphere]
    `--Materials [def Scope]
      `--Glass [def Material]

% usdchecker model.usdz --arkit
Success!

% fixOpacity model.usdz
[...]
```

```
% usdcat model.usdz
```

```
[...]
```

```
% usdtree model.usdz
```

```
/
```

```
  |--Parent [def Xform] (kind = component)
```

```
    |--CubeChild [def Cube]
```

```
    |--SphereChild [def Sphere]
```

```
    |--Materials [def Scope]
```

```
      |--Glass [def Material]
```

```
% usdchecker model.usdz --arkit
```

```
Success!
```

```
% fixOpacity model.usdz
```

```
[...]
```



```
% usdcat model.usdz
```

```
[...]
```

```
% usdtree model.usdz
```

```
/
```

```
  `--Parent [def Xform] (kind = component)
```

```
    |--CubeChild [def Cube]
```

```
    |--SphereChild [def Sphere]
```

```
    `--Materials [def Scope]
```

```
      `--Glass [def Material]
```

```
% usdchecker model.usdz --arkit
```

```
Success!
```

```
% fixOpacity model.usdz
```

```
[...]
```

```
% usdcat model.usdz
```

```
[...]
```

```
% usdtree model.usdz
```

```
/
```

```
`--Parent [def Xform] (kind = component)
```

```
  |--CubeChild [def Cube]
```

```
  |--SphereChild [def Sphere]
```

```
  `--Materials [def Scope]
```

```
    `--Glass [def Material]
```

```
% usdchecker model.usdz --arkit
```

```
Success!
```

```
% fixOpacity model.usdz
```

```
[...]
```

```
% usdcat model.usdz
[...]
```

```
% usdtree model.usdz
/
  `--Parent [def Xform] (kind = component)
    |--CubeChild [def Cube]
    |--SphereChild [def Sphere]
    `--Materials [def Scope]
      `--Glass [def Material]
```

```
% usdchecker model.usdz --arkit
Success!
```

```
% fixOpacity model.usdz
[...]
```

Demo

Chloe Moore, GPU Software Engineer

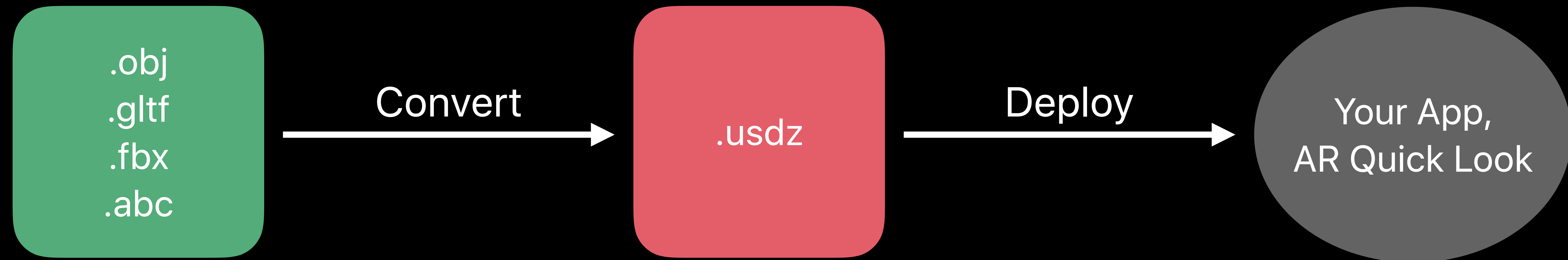
Convert

Create

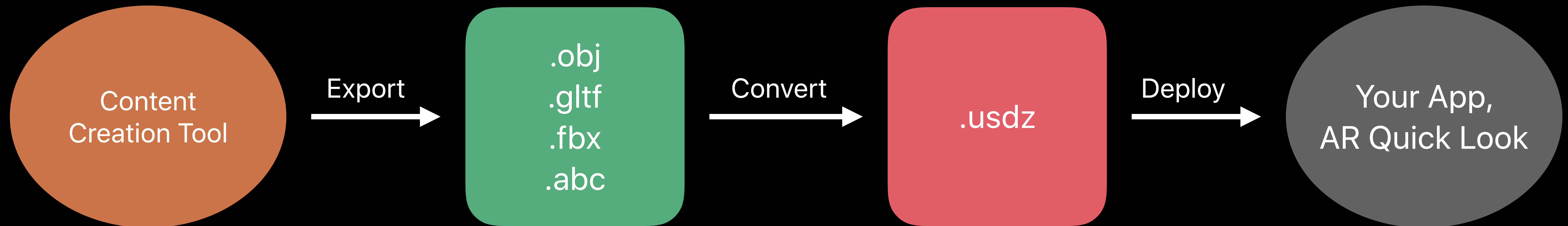
Convert

Create

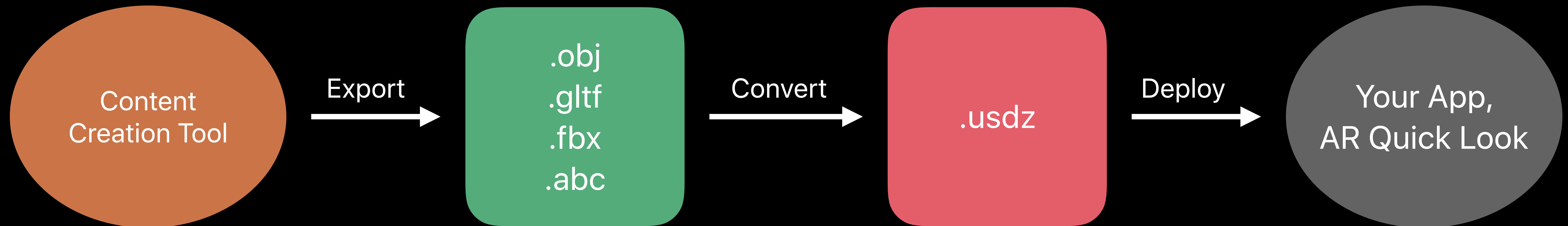
Creating 3D Content



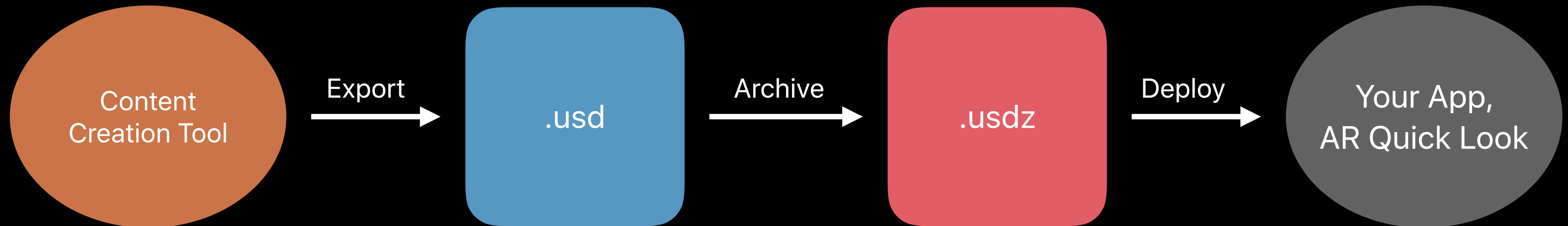
Creating 3D Content

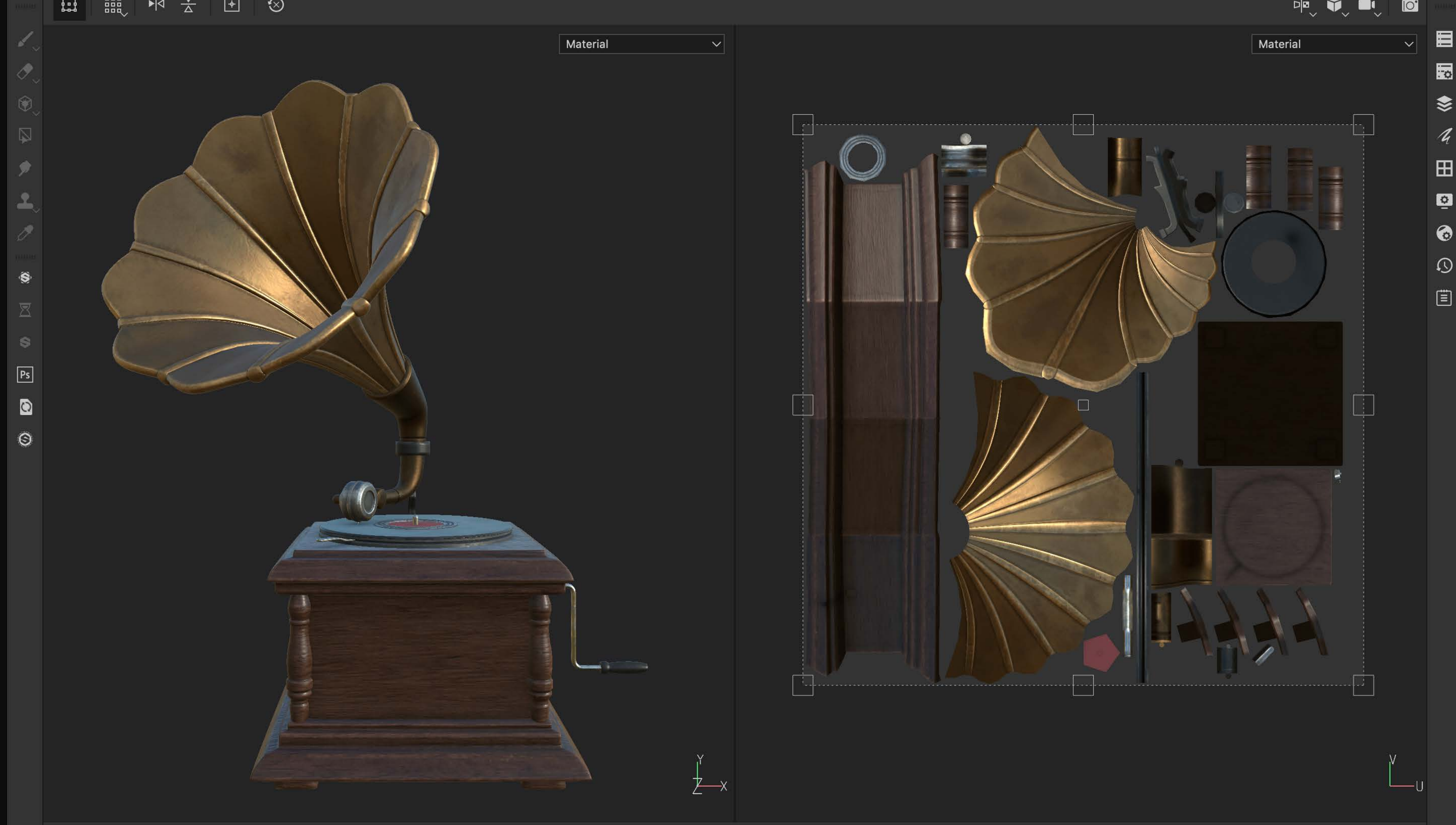


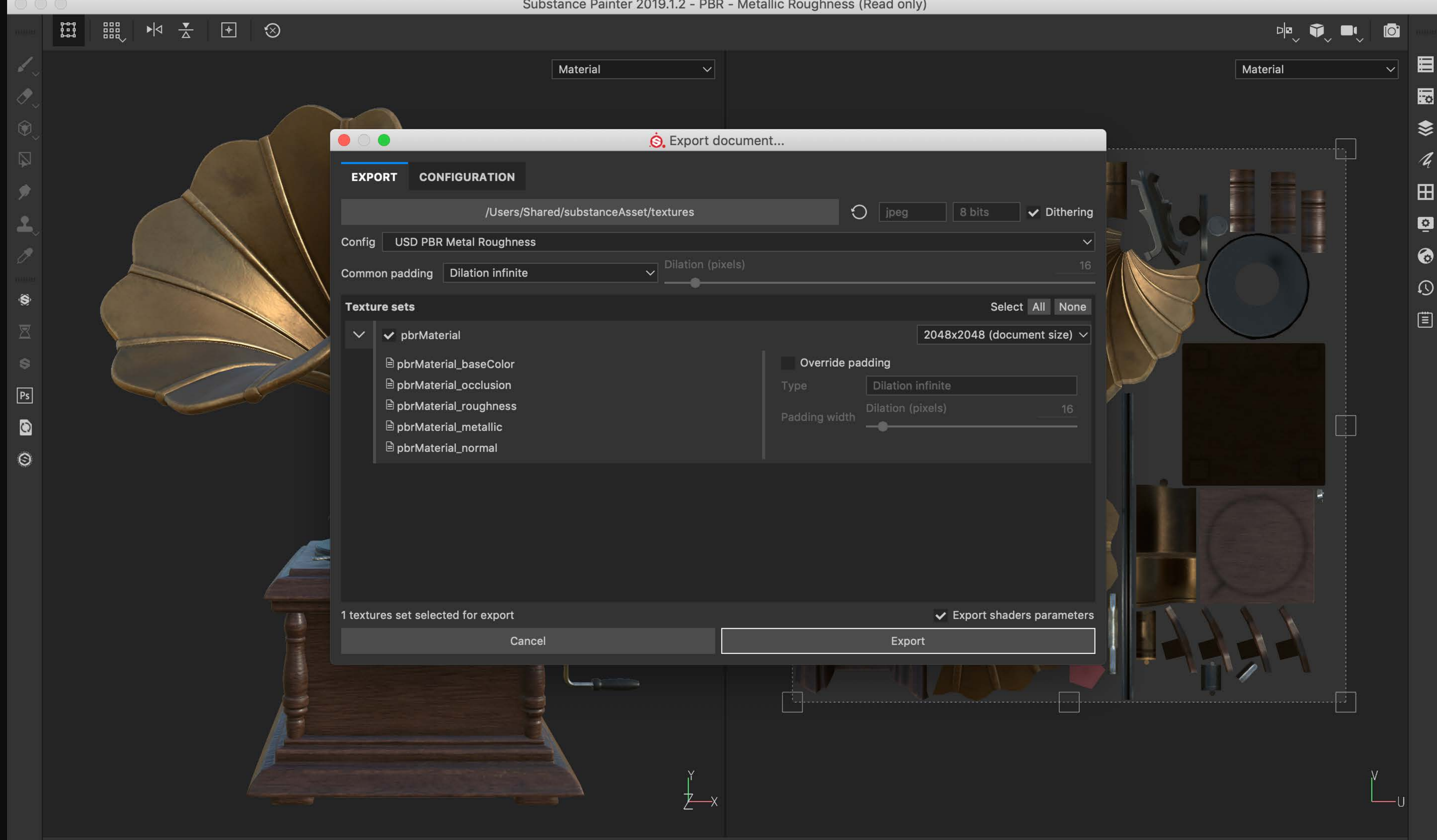
Creating 3D Content



Creating 3D Content







File Edit Create Select Modify Display Windows Mesh Edit Mesh Mesh Tools Mesh Display Curves Surfaces Deform UV Generate Cache Arnold Help

Workspace: Maya Classic*

Outliner

Display Show Help

Search...

- persp
- top
- front
- side
- Stratocaster
- defaultLightSet
- defaultObjectSet

View Shading Lighting Show Renderer Panels

0.00 1.00 sRGB gamma

persp

Select Tool: select an object

The image shows a screenshot of the Autodesk Maya software interface on a MacBook Pro. The main viewport displays a 3D model of a Stratocaster guitar on a stand, rendered in a perspective view. The interface includes a menu bar at the top with options like File, Edit, Create, Select, Modify, Display, Windows, Mesh, Edit Mesh, Mesh Tools, Mesh Display, Curves, Surfaces, Deform, UV, Generate, Cache, Arnold, and Help. Below the menu bar is a toolbar with various icons for navigation and editing. On the left side, there is an Outliner panel with a search bar and a list of objects: persp, top, front, side, Stratocaster, defaultLightSet, and defaultObjectSet. The bottom status bar shows 'Select Tool: select an object'.

MacBook Pro

Outliner

Display Show Help

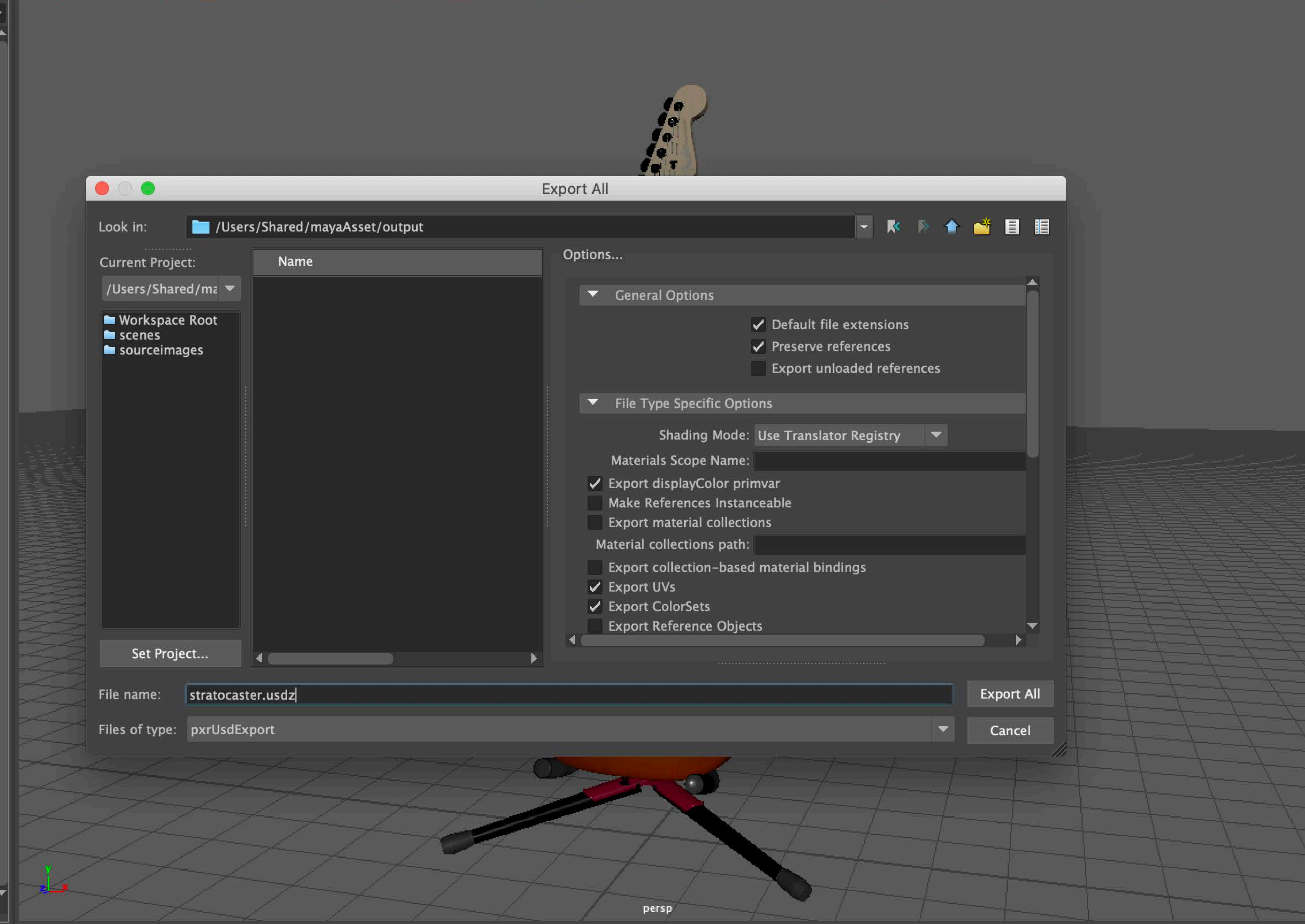
Search...

- persp
- top
- front
- side
- Stratocaster
- defaultLightSet
- defaultObjectSet

M

View Shading Lighting Show Renderer Panels

0.00 1.00 sRGB gamma



Export All

Look in: /Users/Shared/mayaAsset/output

Current Project: /Users/Shared/mayaAsset

Name

Options...

General Options

- Default file extensions
- Preserve references
- Export unloaded references

File Type Specific Options

Shading Mode: Use Translator Registry

Materials Scope Name:

- Export displayColor primvar
- Make References Instanceable
- Export material collections

Material collections path:

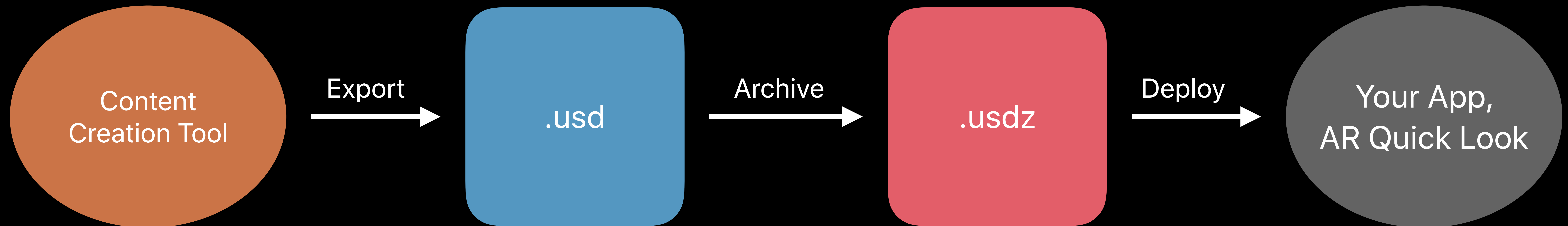
- Export collection-based material bindings
- Export UVs
- Export ColorSets
- Export Reference Objects

File name: stratocaster.usdz

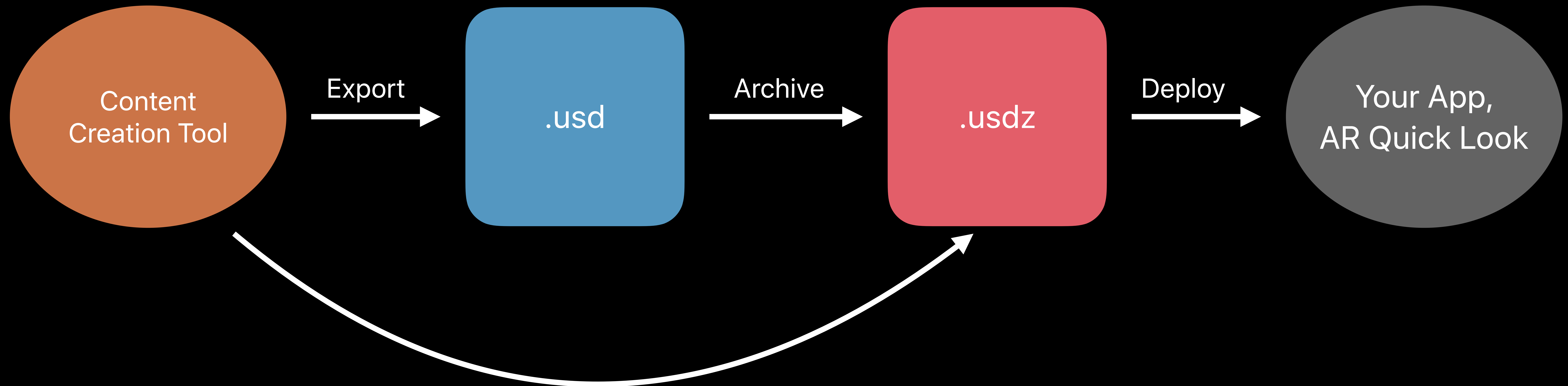
Files of type: pxrUsdExport

Export All Cancel

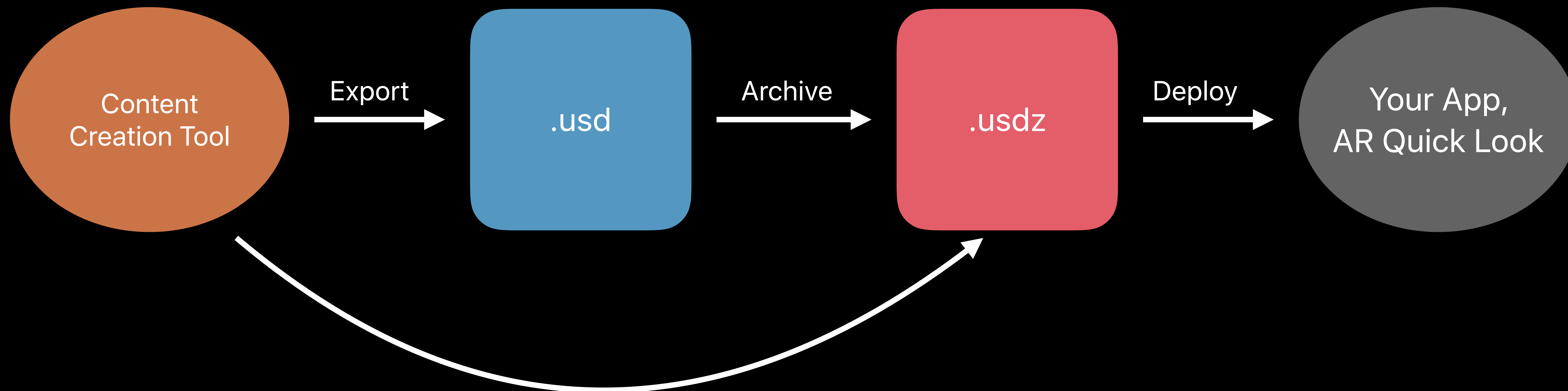
Creating 3D Content



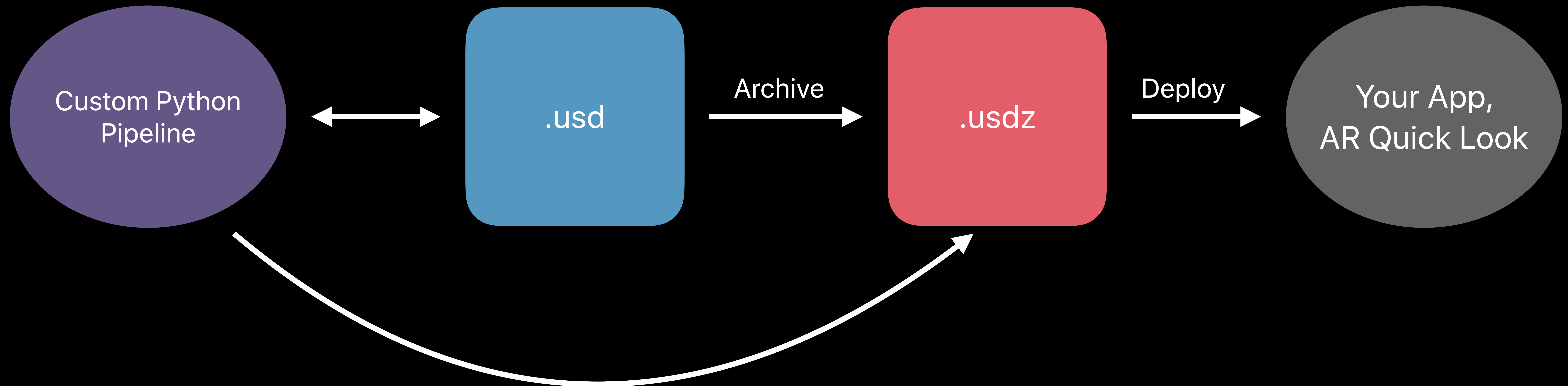
Creating 3D Content



Creating 3D Content



Creating 3D Content



Exporting usdz from SceneKit

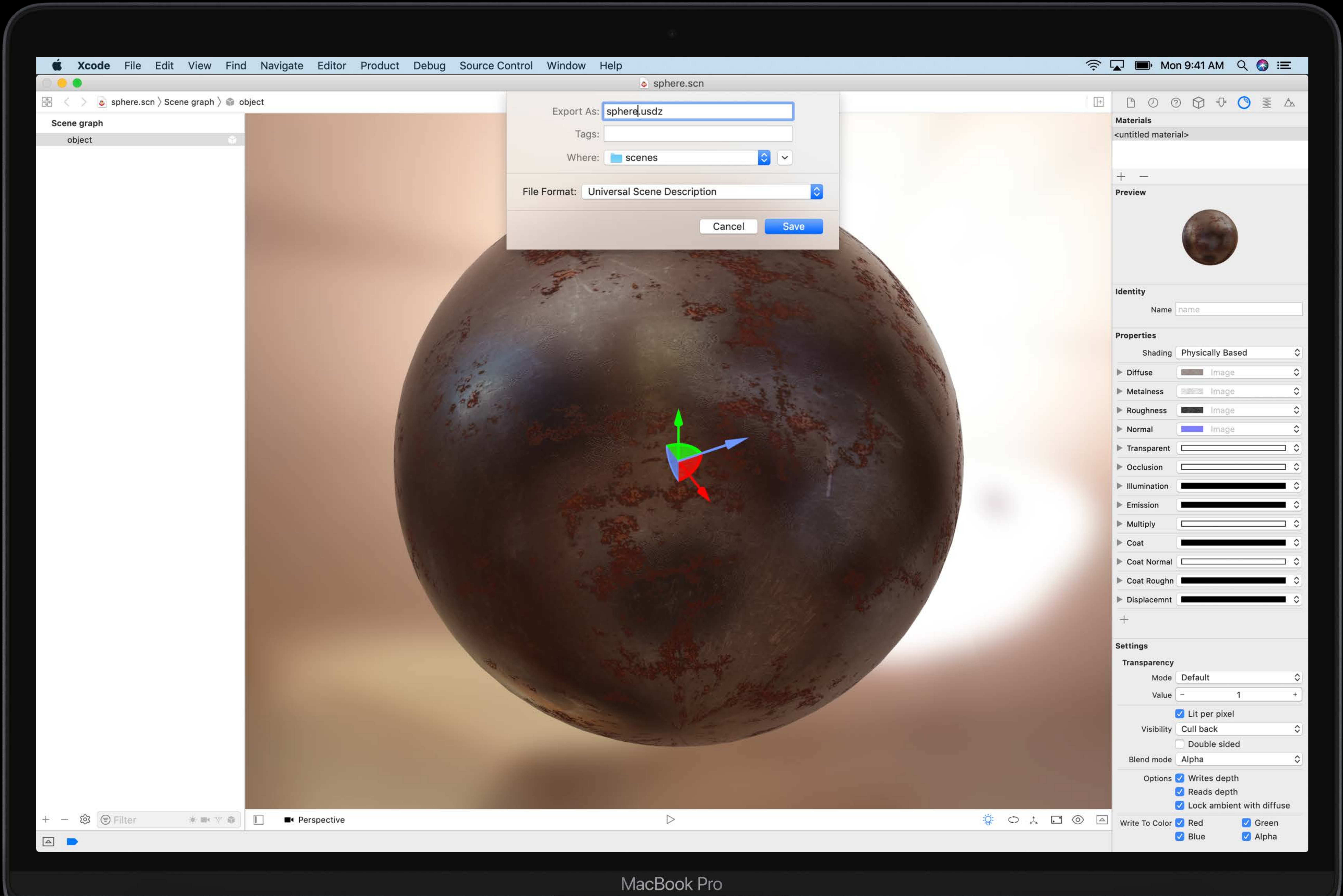
NEW

Create or load SCNScene

Export .usdz

```
let scene = SCNScene(...)
...
scene.write(to: baseUrl.appendingPathComponent("file.usdz"), delegate: nil)
```

NEW



USD Features

Hydra API Schema Purpose AssetInfo Specializes
Class Time-Varying Properties UsdPreviewSurface Variant Geometry Subset
Model API Over Path Translation Indexed Mesh Attribute VariantSet
Inherits UsdLux SubLayer Layer Offset Namespace Prim
Shader Graph Interpolation Assembly Connection
Crate File Layer Stack Edit Target PointInstancer Composition Engine
TimeCode Proxy
Subdivision Surface **USD Features** Instance Override
Layer Scene Graph Time-Varying Topology Instancing Mesh Value Resolution
Custom Data Collection Session Layer Default Value Transform Stack
Material Library Skeletal Animation Payload Custom Schema
Reference Subtree Skinning Visibility Metadata
PseudoRoot List Editing Kind Vertex Animation Stage
Value Clip Relationship TimeSample Flatten Asset Resolver

USD Features

Hydra

API Schema

Purpose

AssetInfo

Specializes

Class

Time-Varying Properties

UsdPreviewSurface

Variant

Geometry Subset

Model API

Over

Path Translation

Indexed Mesh Attribute

VariantSet

Inherits

UsdLux

SubLayer

Layer Offset

Namespace

Prim

Shader Graph

Interpolation

Assembly

Connection

Crate File

Layer Stack

Edit Target

PointInstancer

Composition Engine

TimeCode

Proxy

Subdivision Surface

Instance Override

Layer

Scene Graph

Time-Varying Topology

Instancing

Mesh

Value Resolution

Custom Data

Collection

Session Layer

Default Value

Transform Stack

Material Library

Skeletal Animation

Payload

Custom Schema

Reference

Subtree Skinning

Visibility

Metadata

PseudoRoot

List Editing

Kind

Vertex Animation

Stage

Value Clip

Relationship

TimeSample

Flatten

Asset Resolver

What is USD?

Workflows

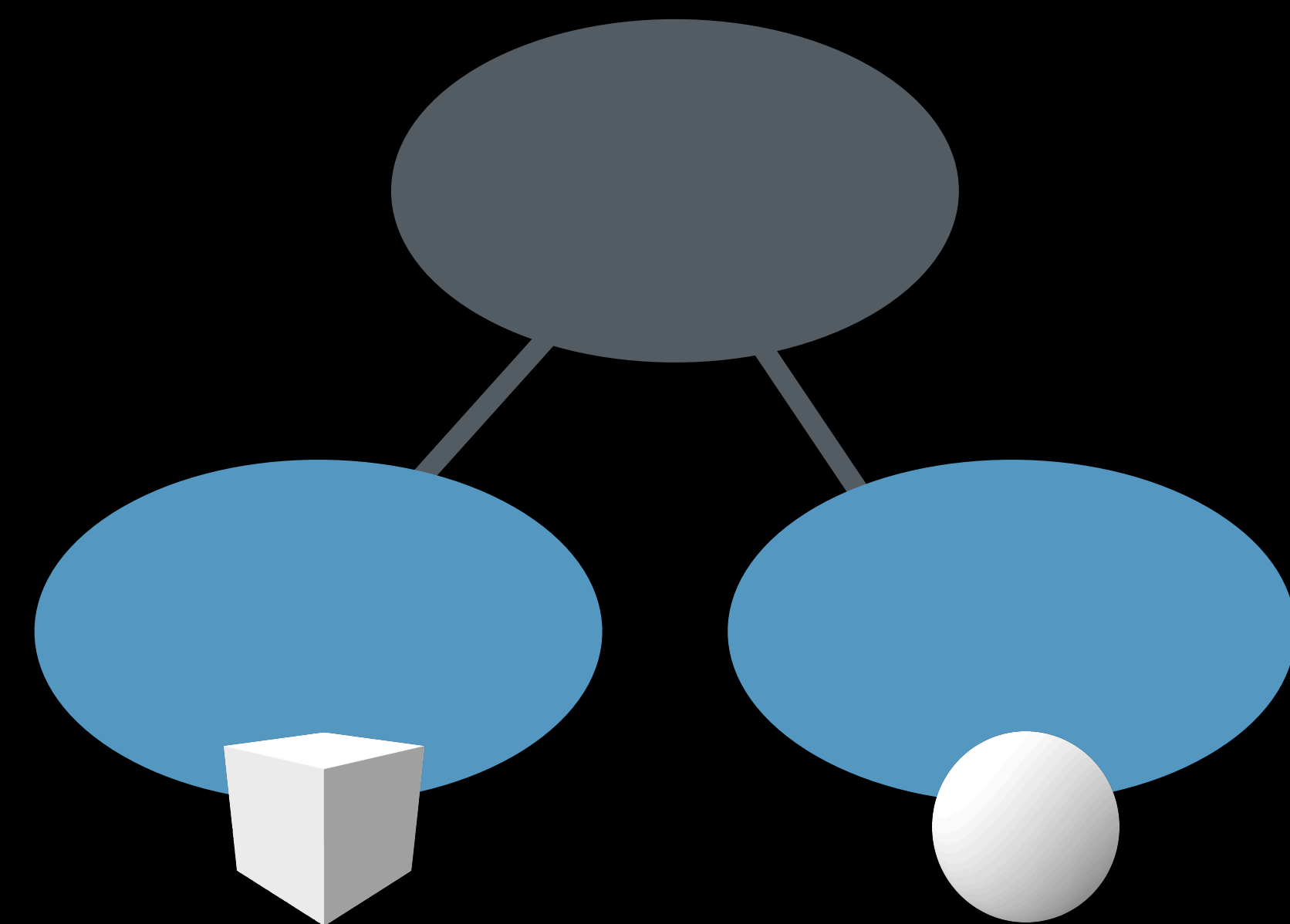
USD essentials

USD highlights

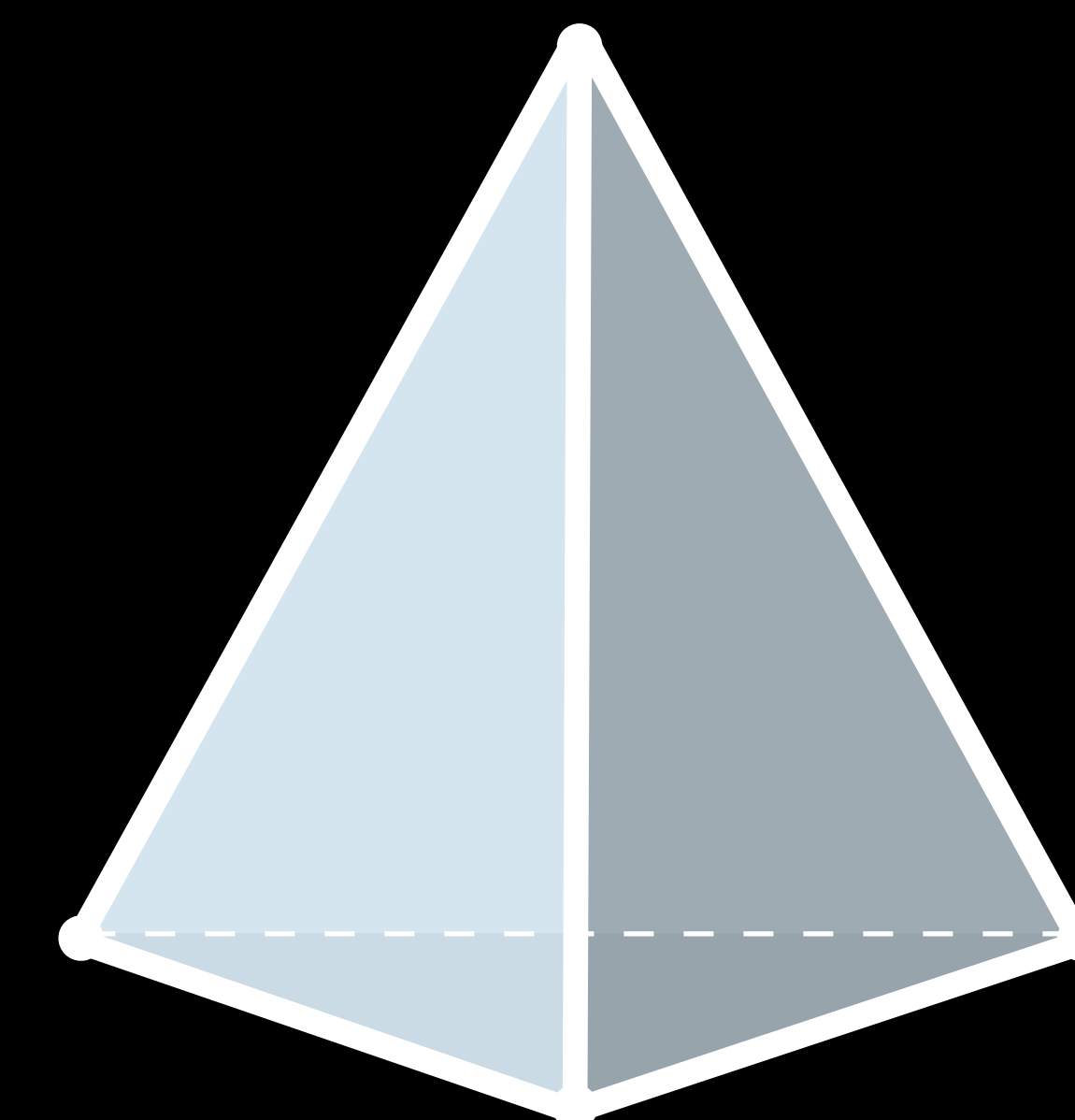
USD Essentials



Basic File Structure



Scene Graph



Mesh Data

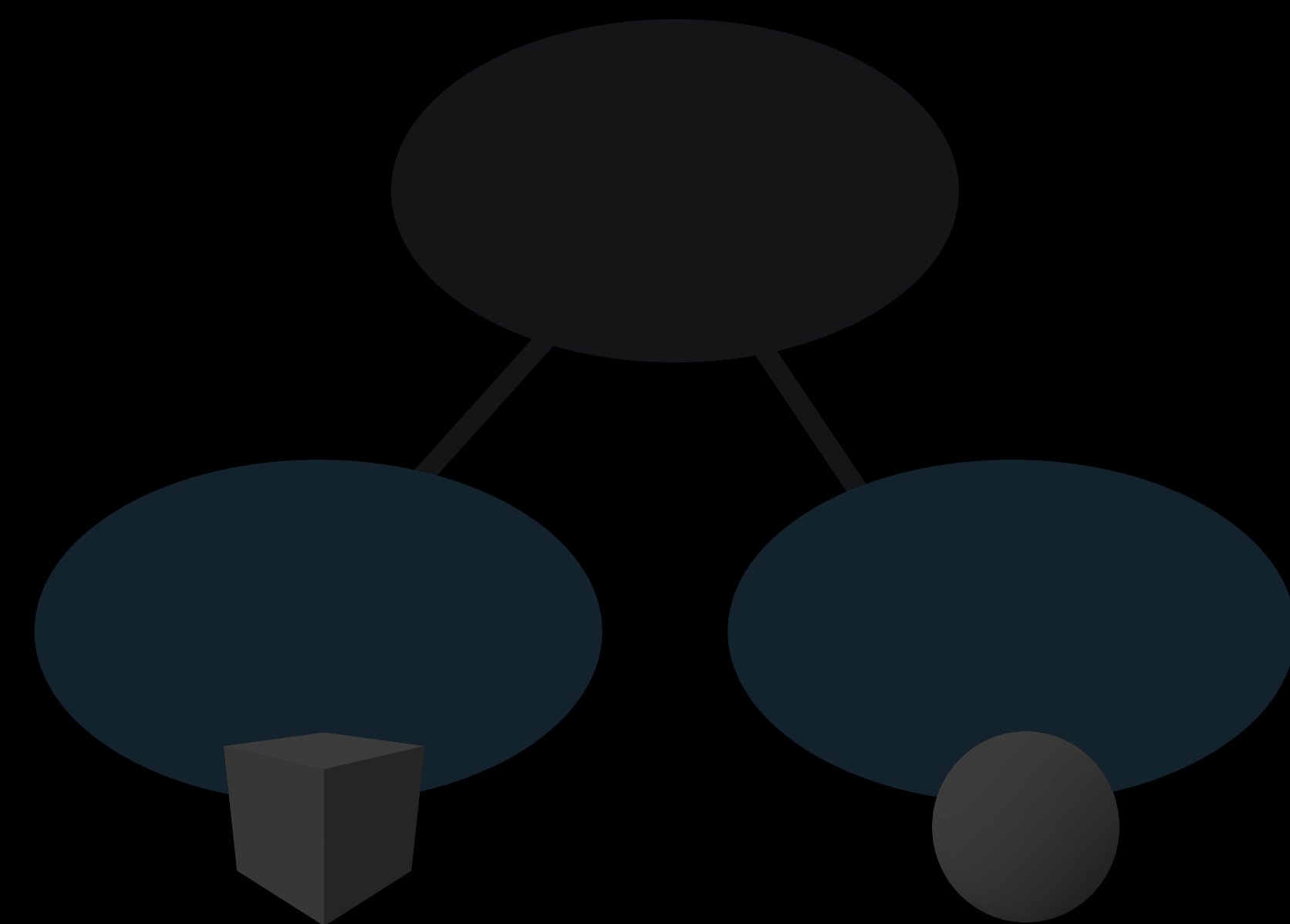


Materials

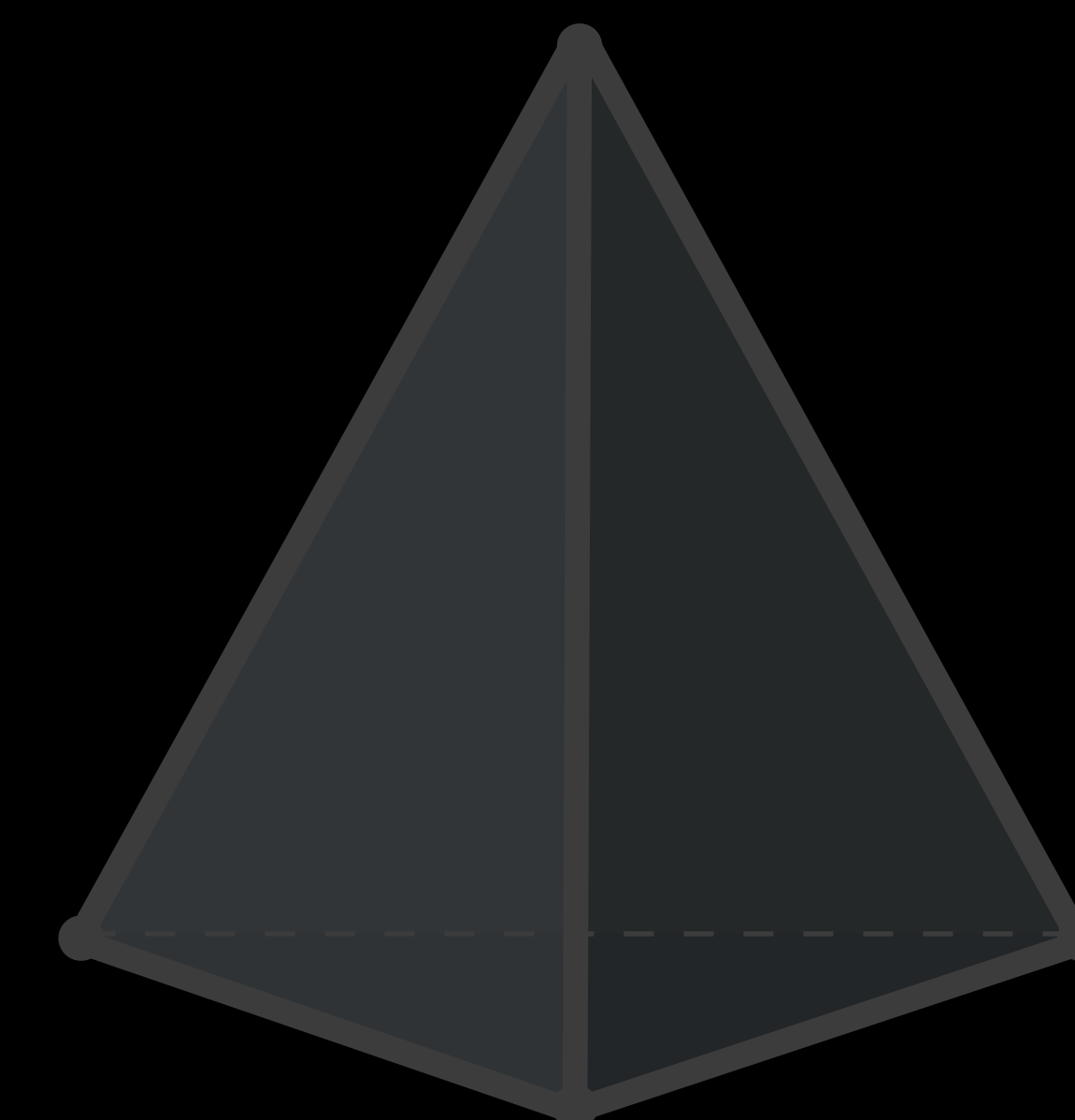
USD Essentials



Basic File Structure



Scene Graph



Mesh Data



Materials

File Structure

```
#usda 1.0
```

File Structure — Prims

```
#usda 1.0
```

```
def Xform "simpleMesh"
```

```
{
```

```
  def Mesh "cube"
```

```
  {
```

```
  }
```

```
}
```

File Structure — Properties

```
#usda 1.0
```

```
def Xform "simpleMesh"
```

```
{
```

```
  def Mesh "cube"
```

```
  {
```

```
    point3f[] points = [(0,0,0), (1,0,0), (1,1,0), (0,1,0)]
```

```
    normal3f[] normals = [(0,0,1)]
```

```
    ...
```

```
  }
```

```
}
```

File Structure — Metadata

```
#usda 1.0
(
  upAxis = "Y"
)

def Xform "simpleMesh"

{
  def Mesh "cube"
  {
    point3f[] points = [(0,0,0), (1,0,0), (1,1,0), (0,1,0)]
    normal3f[] normals = [(0,0,1)]

    ...
  }
}
```

File Structure — Metadata

```
#usda 1.0
(
  upAxis = "Y"
)

def Xform "simpleMesh" (
  kind = "component"
)
{
  def Mesh "cube"
  {
    point3f[] points = [(0,0,0), (1,0,0), (1,1,0), (0,1,0)]
    normal3f[] normals = [(0,0,1)]

    ...
  }
}
```

File Structure — Metadata

```
#usda 1.0
(
  upAxis = "Y"
)

def Xform "simpleMesh" (
  kind = "component"
)
{
  def Mesh "cube"
  {
    point3f[] points = [(0,0,0), (1,0,0), (1,1,0), (0,1,0)]
    normal3f[] normals = [(0,0,1)] (
      interpolation = "uniform"
    )
    ...
  }
}
```


Object Access Through Paths

```
#usda 1.0
(
  upAxis = "Y"
)

def Xform "simpleMesh" (
  kind = "component"
)
{
  def Mesh "cube" ← /simpleMesh/cube
  {
    point3f[] points = [(0,0,0), (1,0,0), (1,1,0), (0,1,0)]
    normal3f[] normals = [(0,0,1)] (
      interpolation = "uniform"
    )
    ...
  }
}
```

Object Access Through Paths

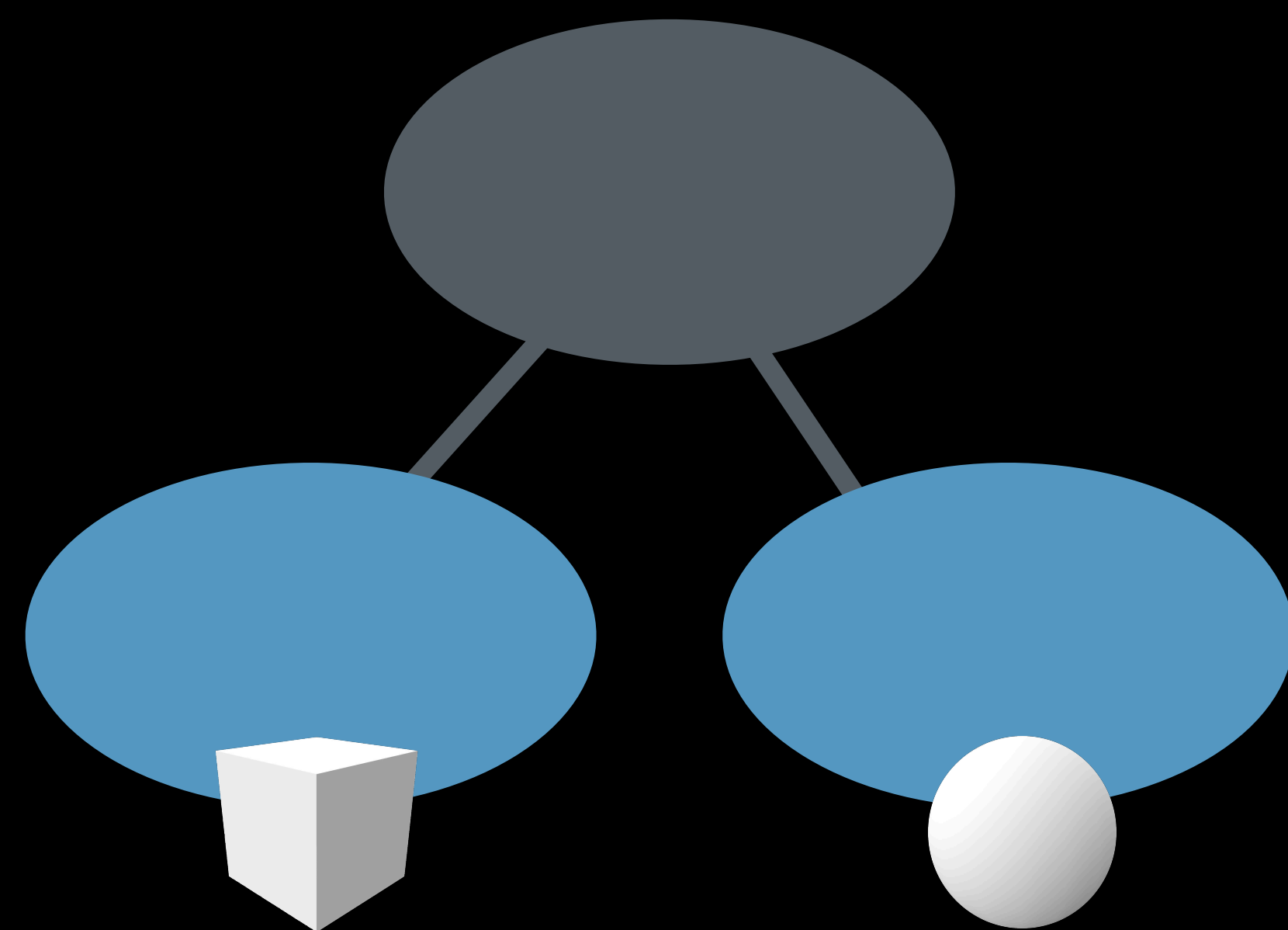
```
#usda 1.0
(
  upAxis = "Y"
)

def Xform "simpleMesh" (
  kind = "component"
)
{
  def Mesh "cube"
  {
    point3f[] points = [(0,0,0), (1,0,0), (1,1,0), (0,1,0)]
    normal3f[] normals = [(0,0,1)] ( ← /simpleMesh/cube.normals
      interpolation = "uniform"
    )
    ...
  }
}
```

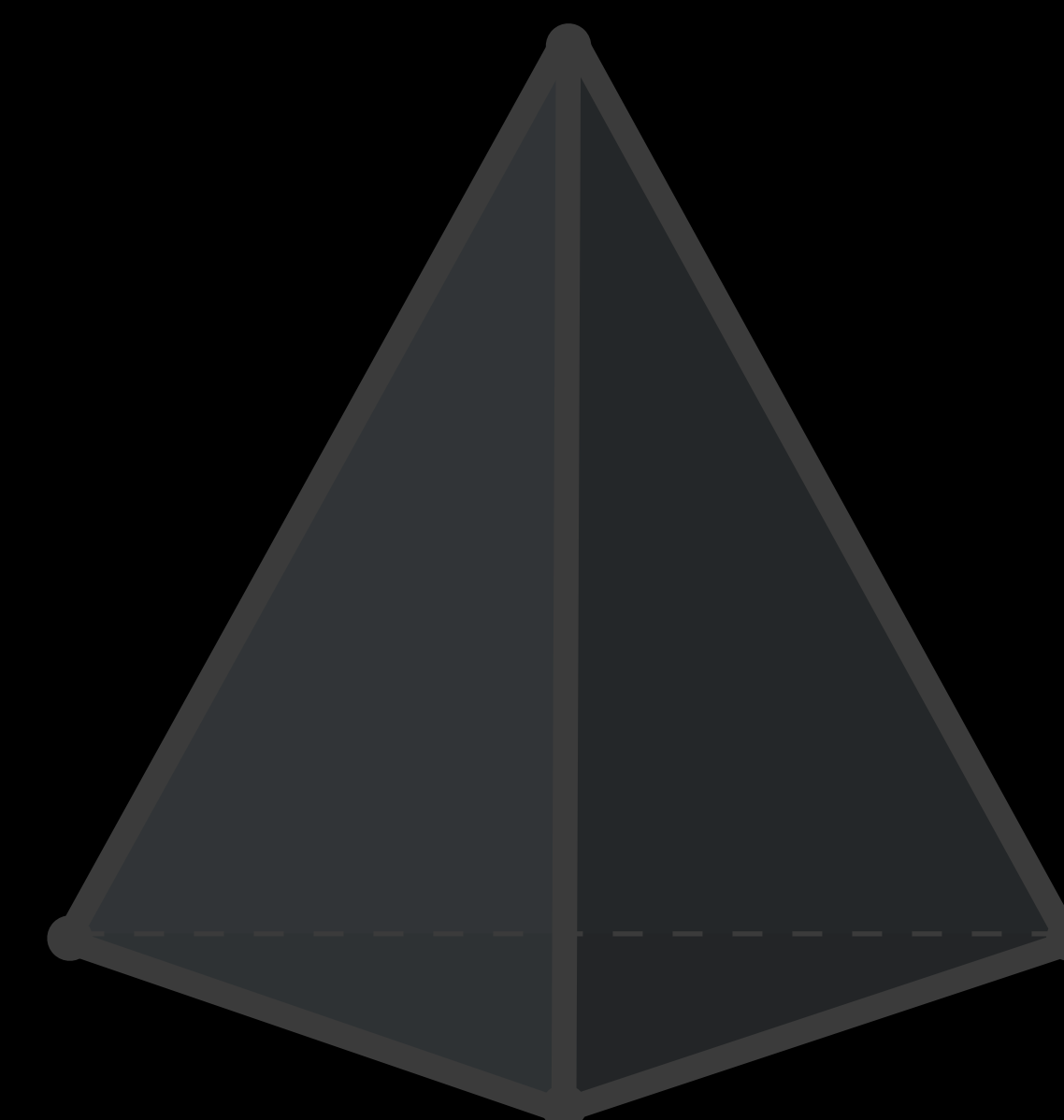
USD Essentials



Basic File Structure



Scene Graph



Mesh Data

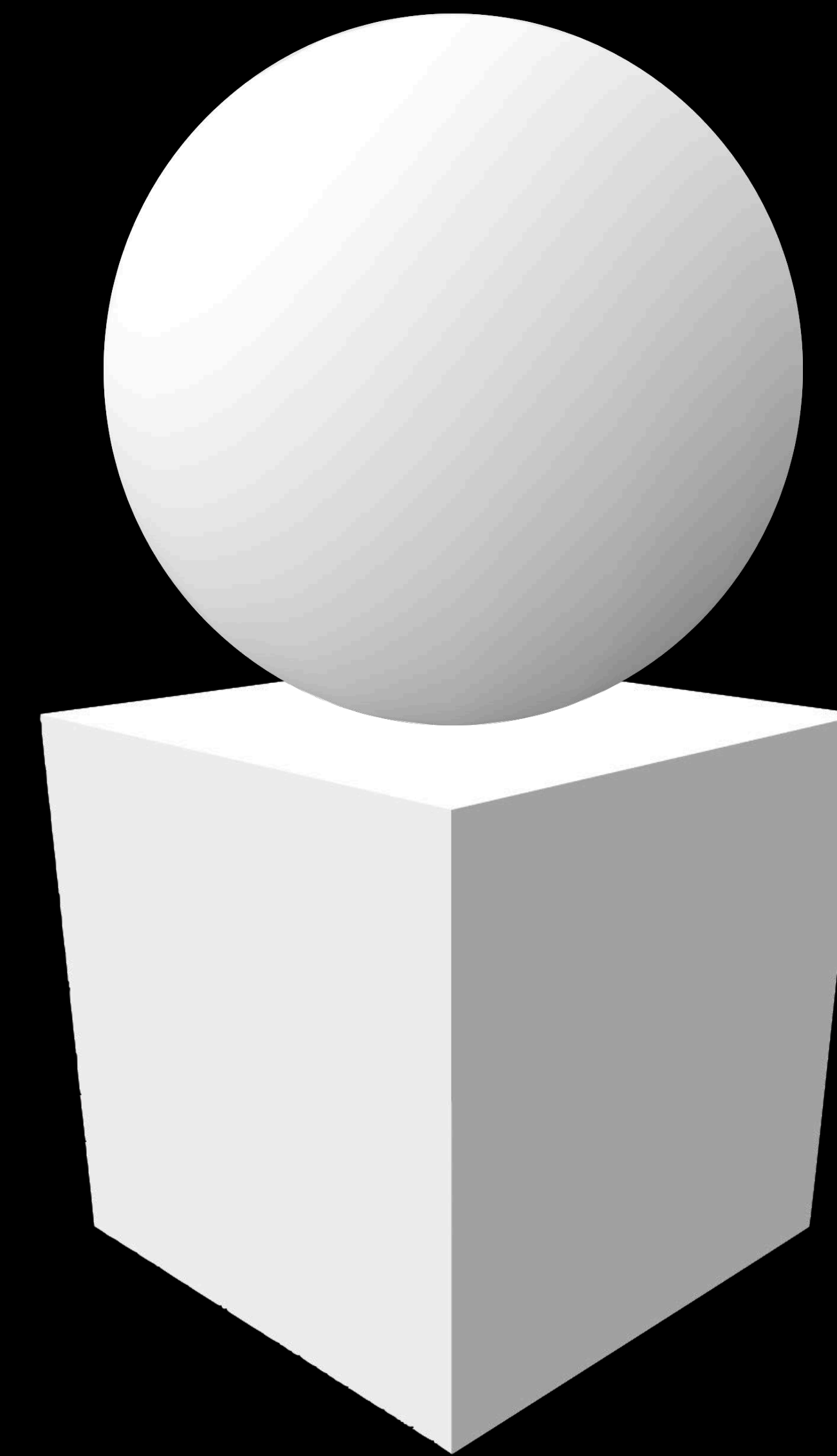
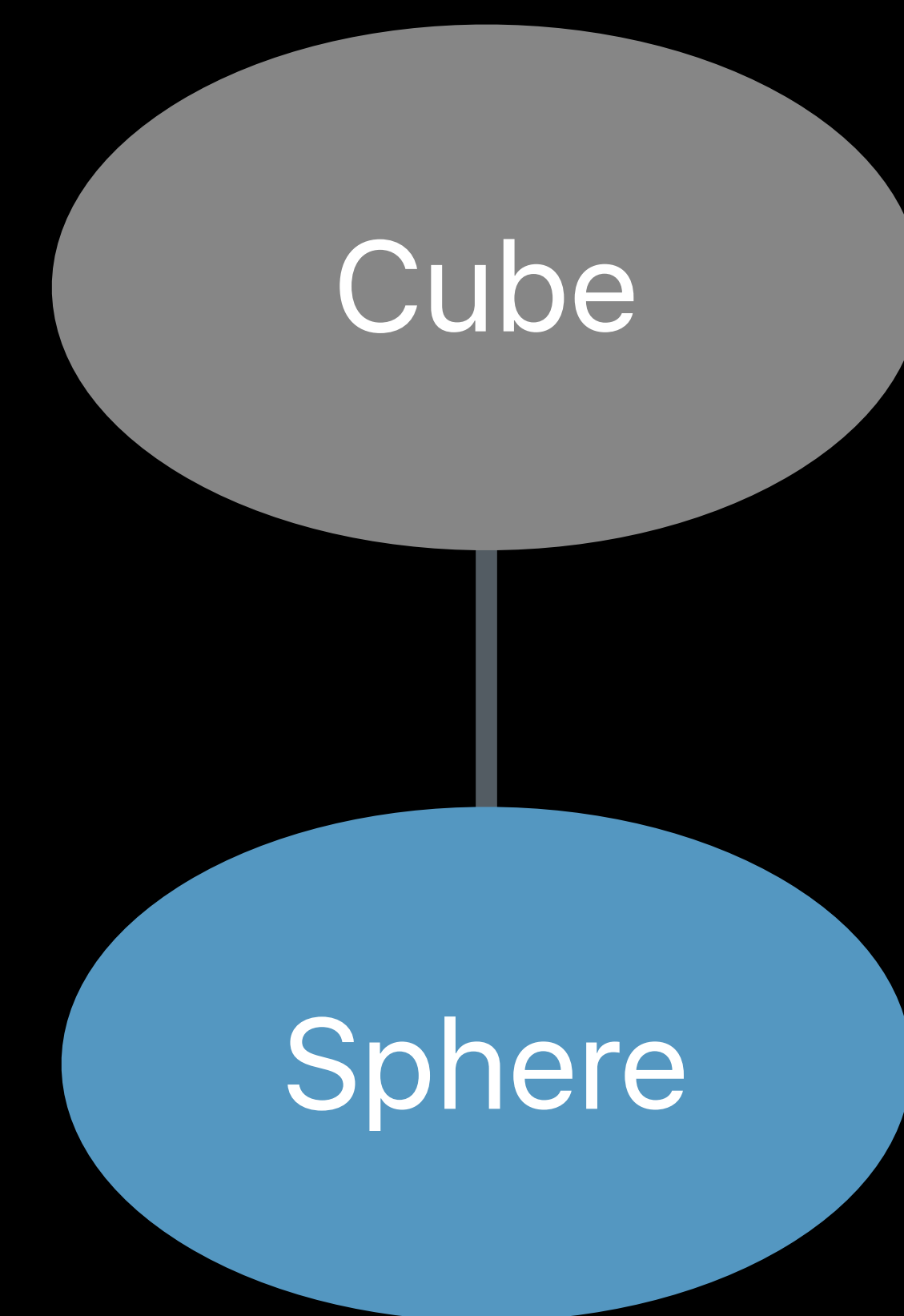


Materials

Scene Graph

Scene graph defines object hierarchy

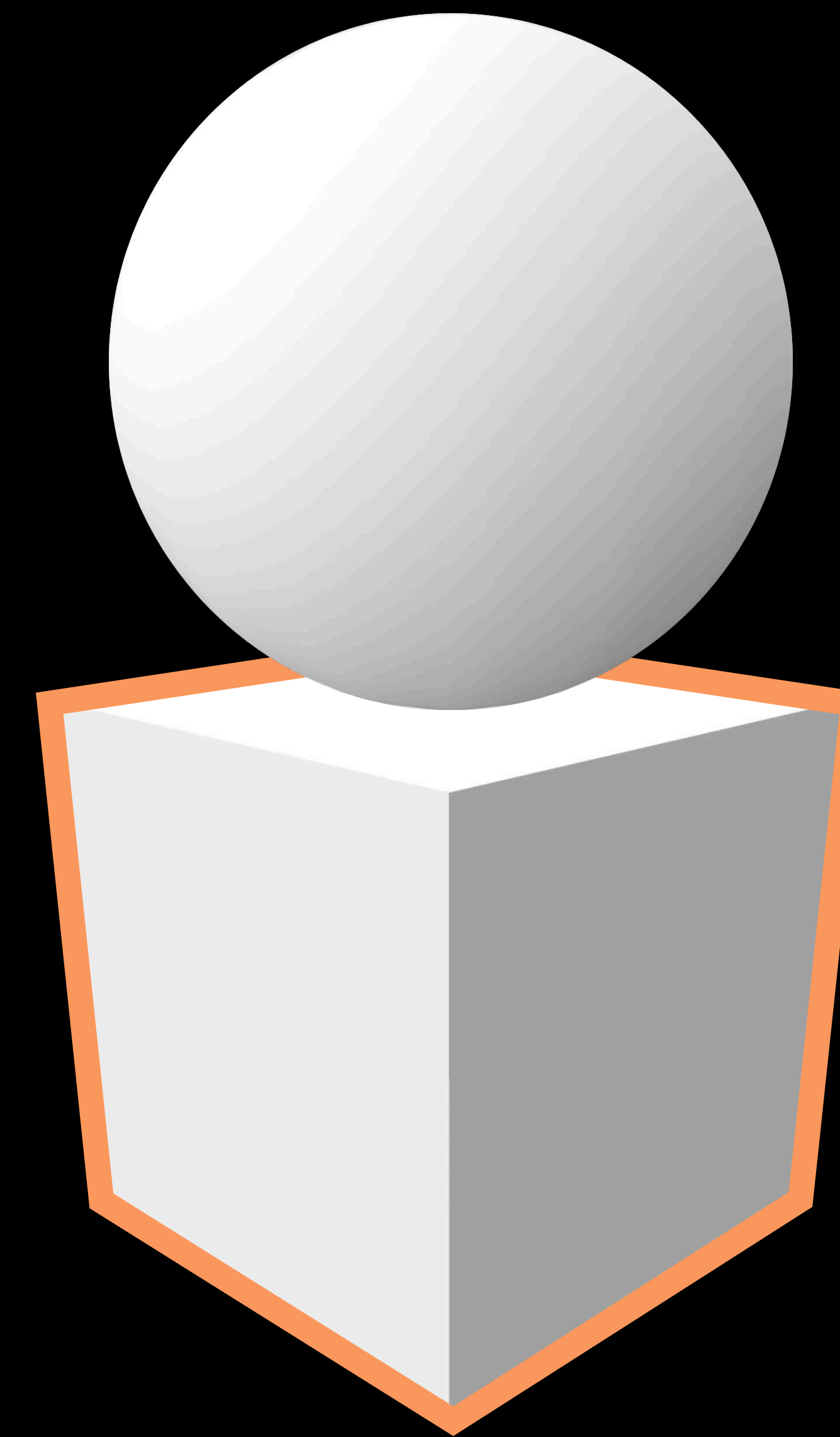
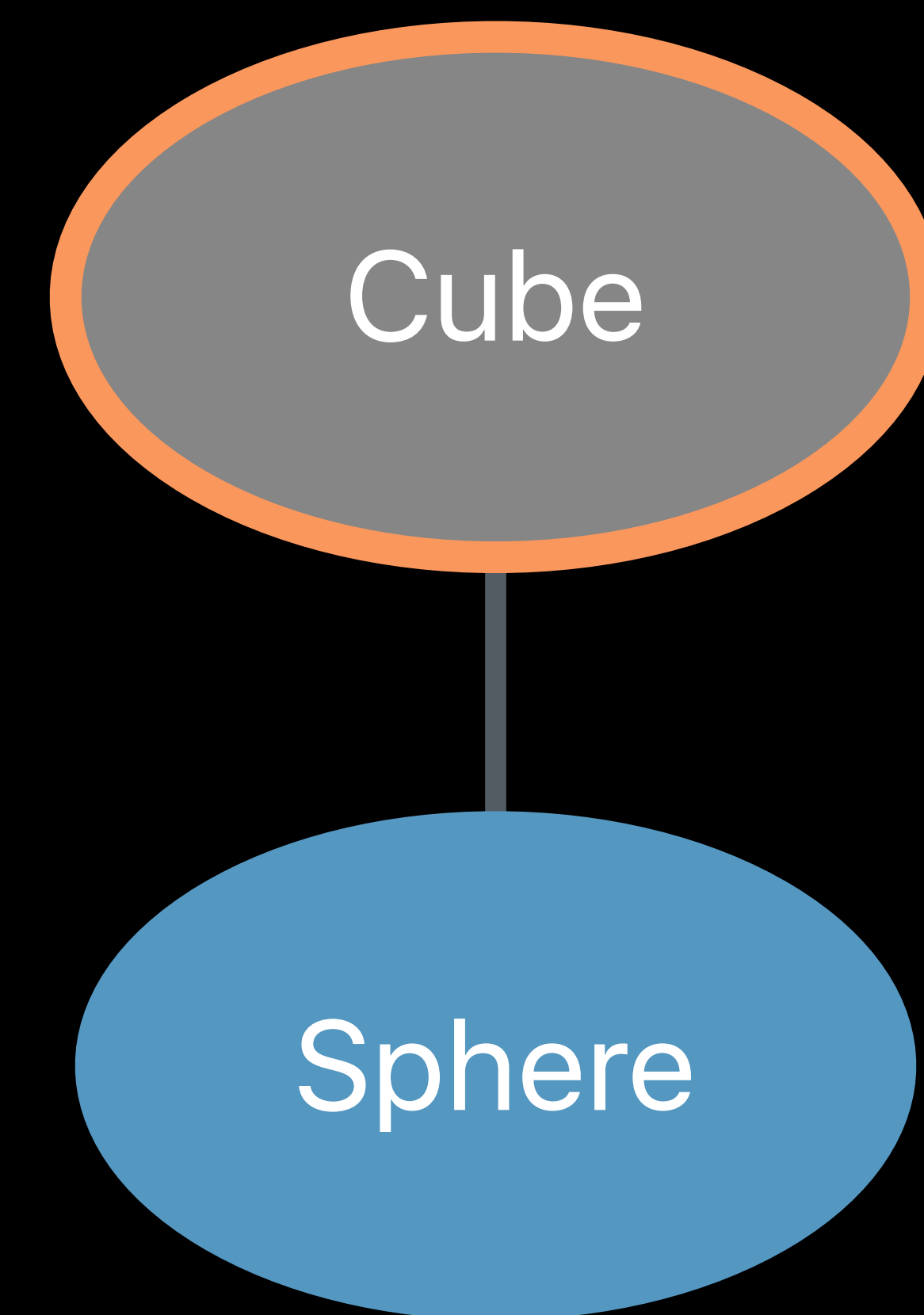
Transforms on a parent also affect its children



Scene Graph

Scene graph defines object hierarchy

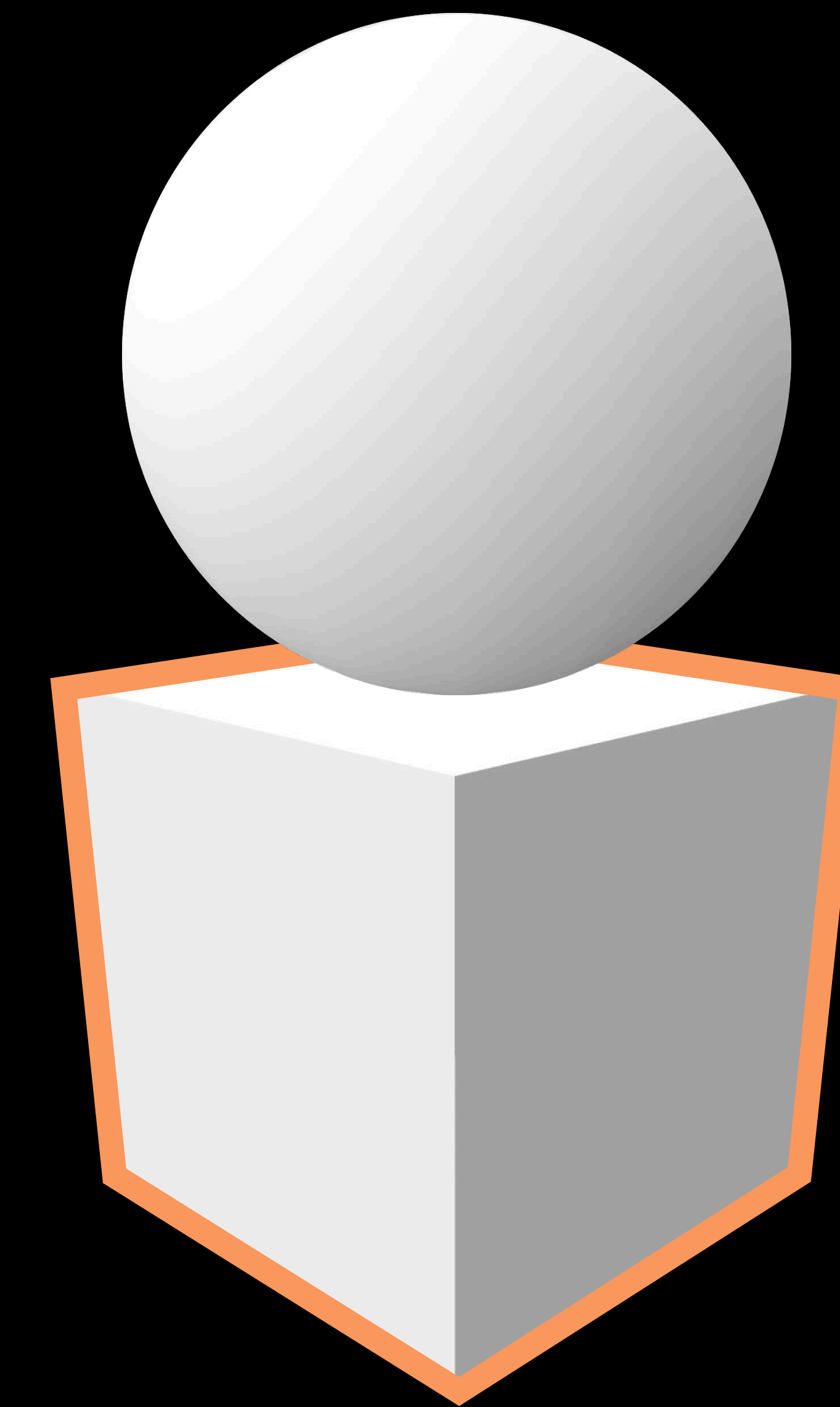
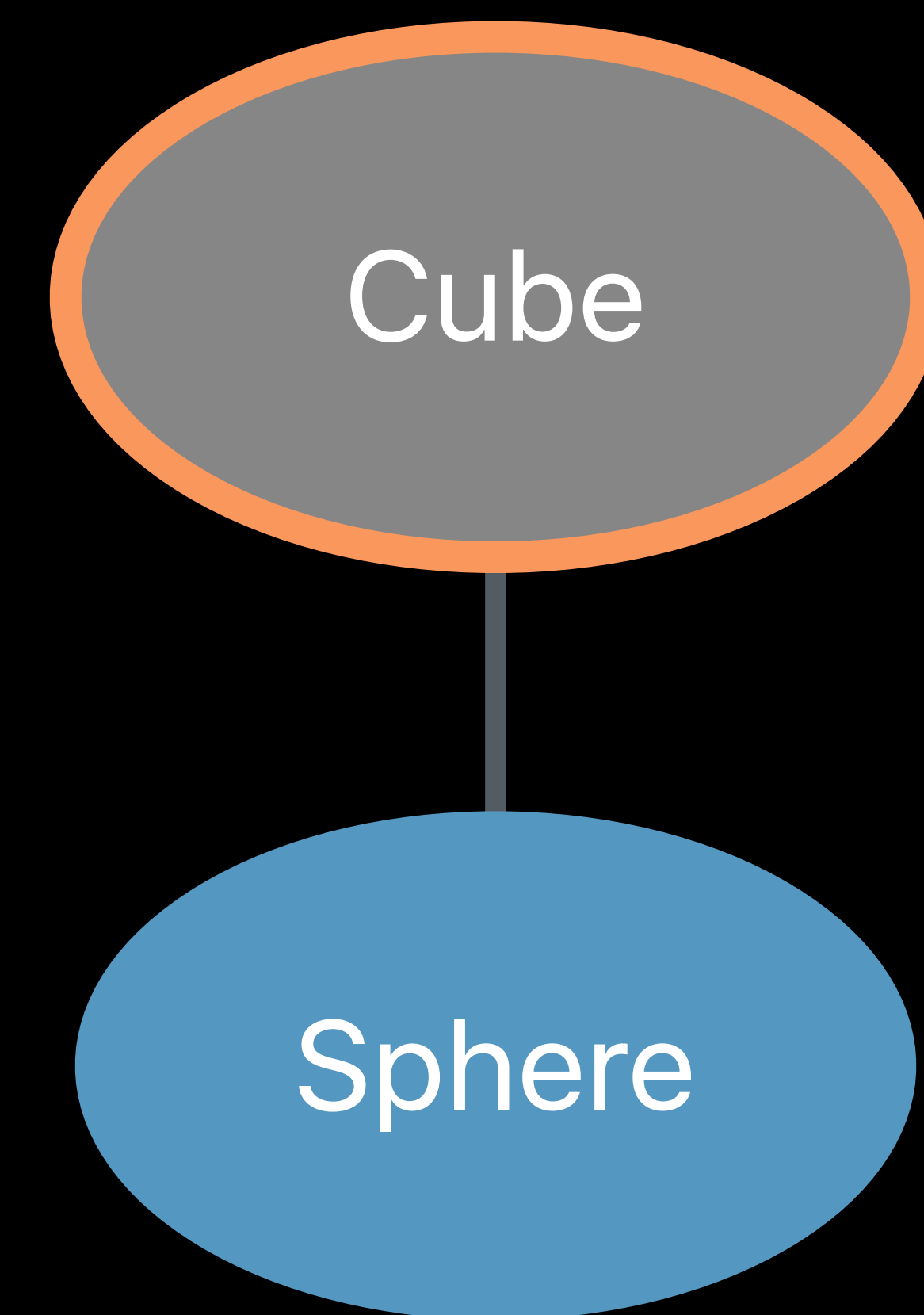
Transforms on a parent also affect its children



Scene Graph

Scene graph defines object hierarchy

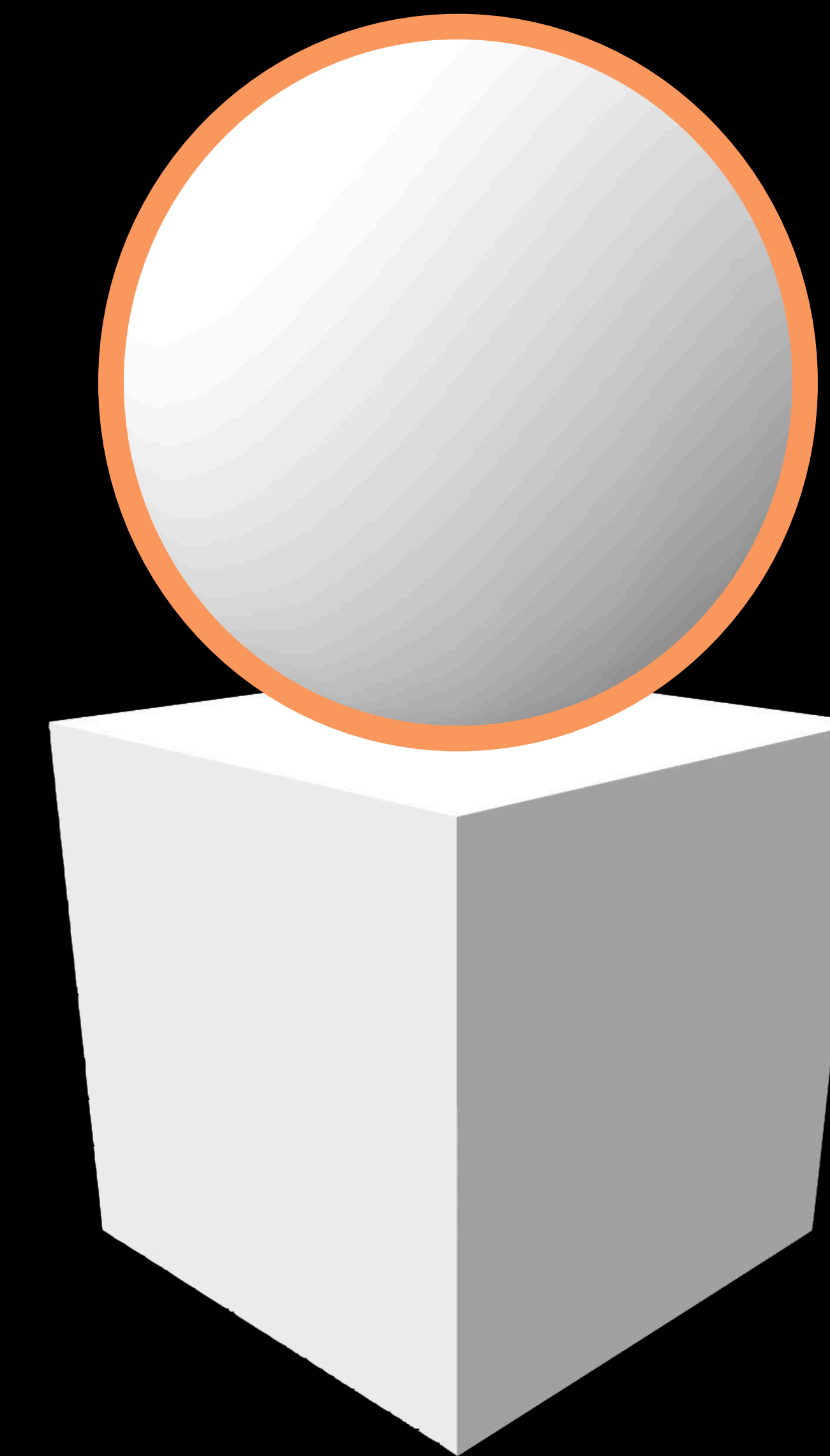
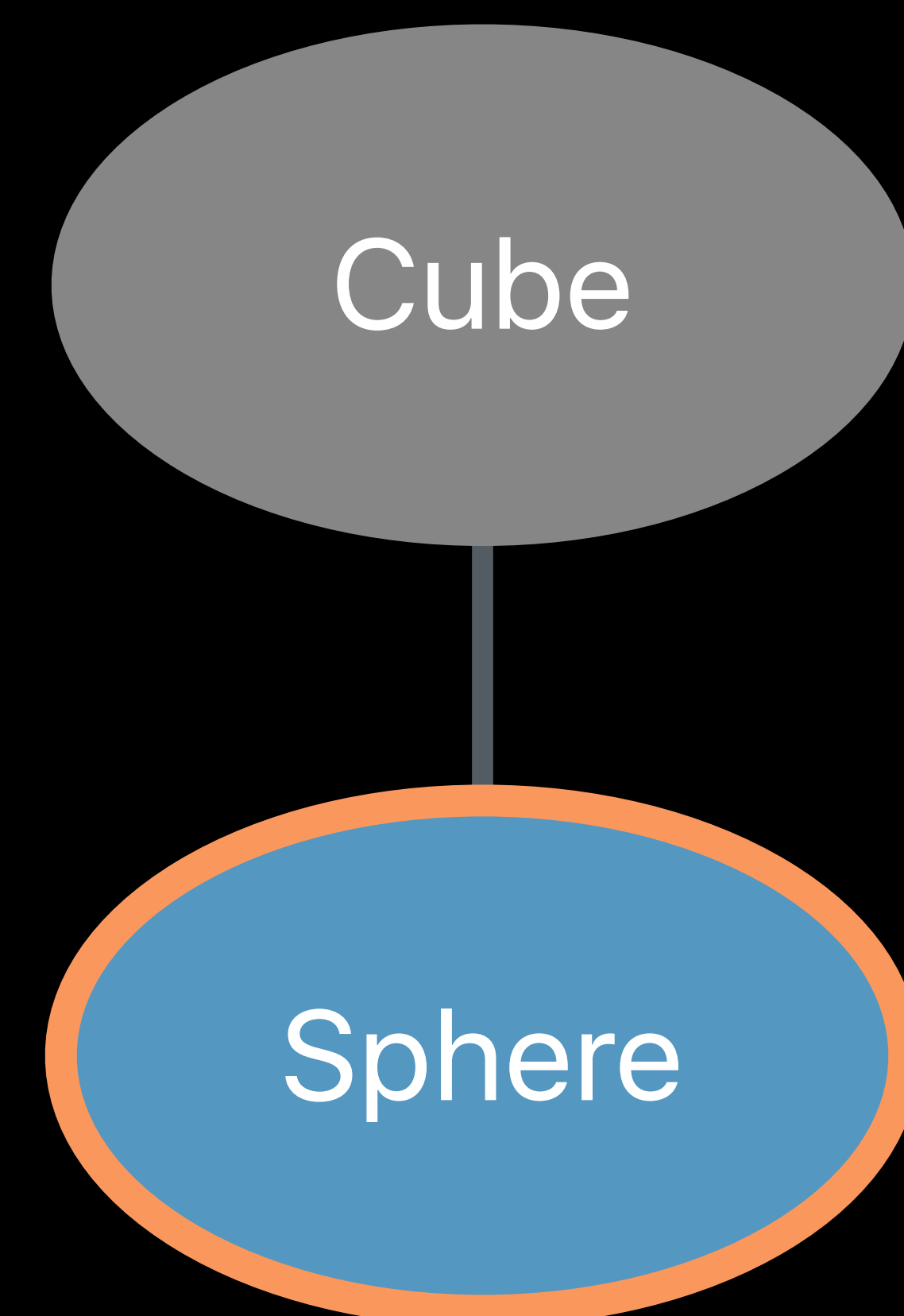
Transforms on a parent also affect its children



Scene Graph

Scene graph defines object hierarchy

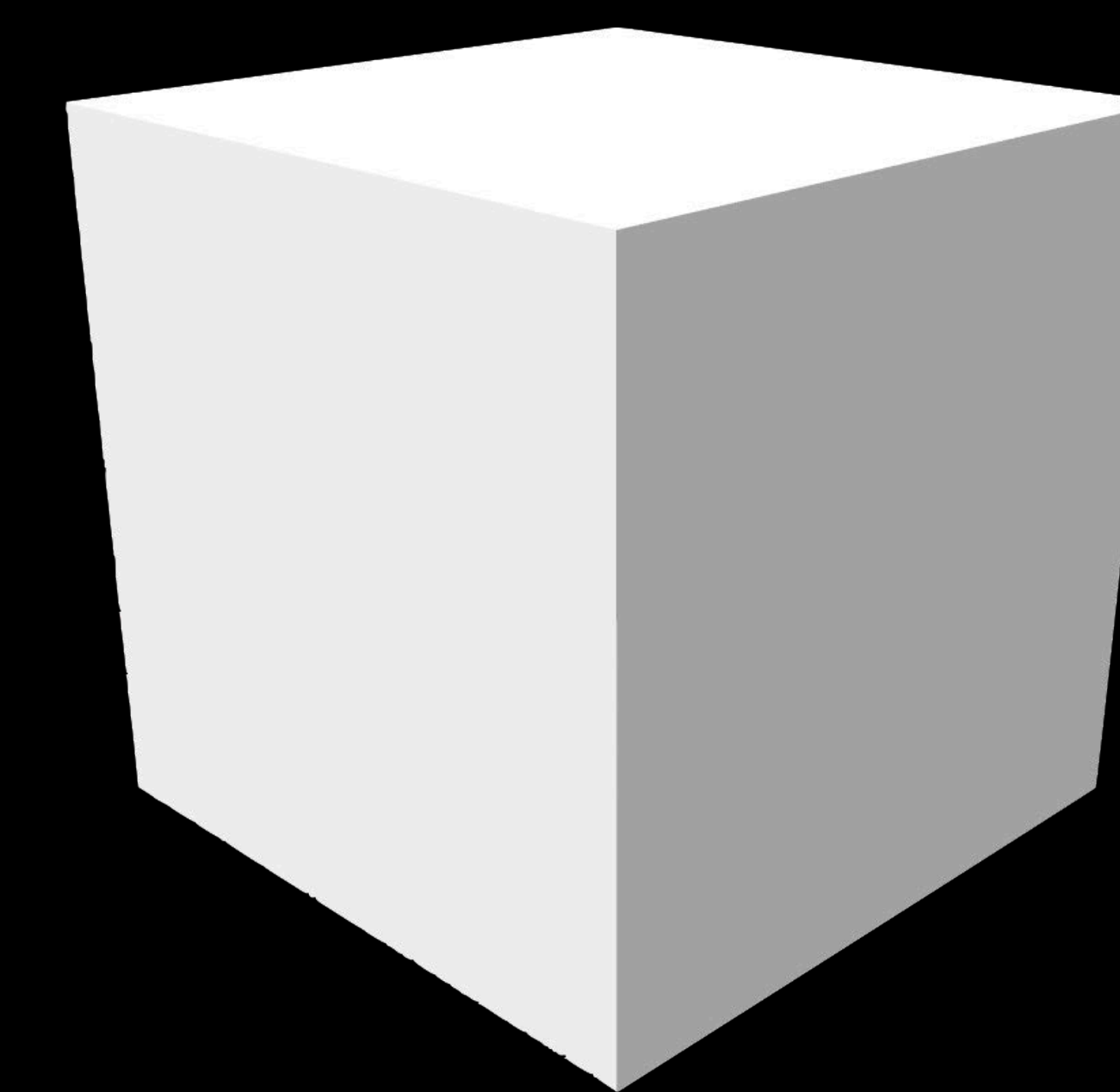
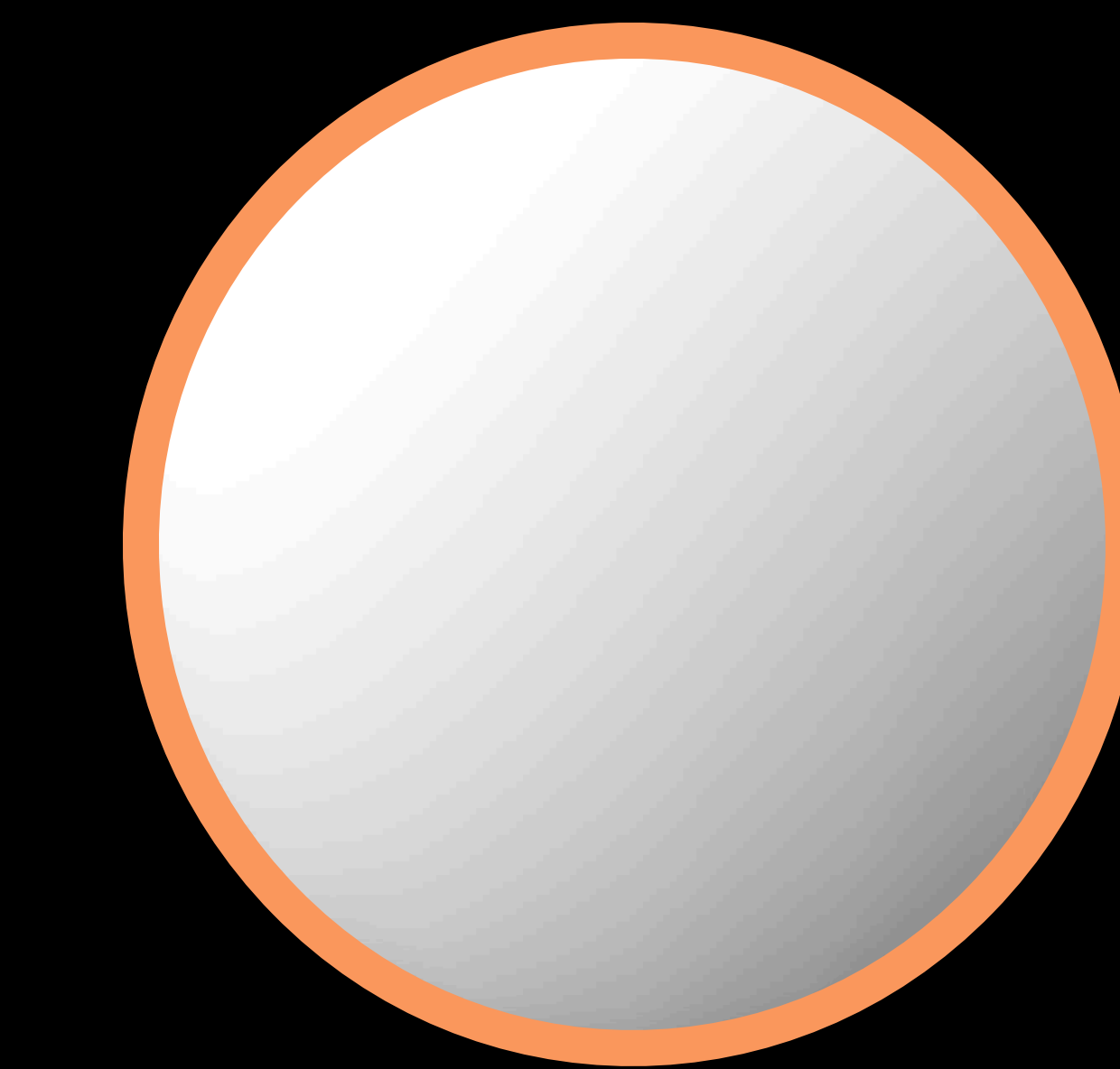
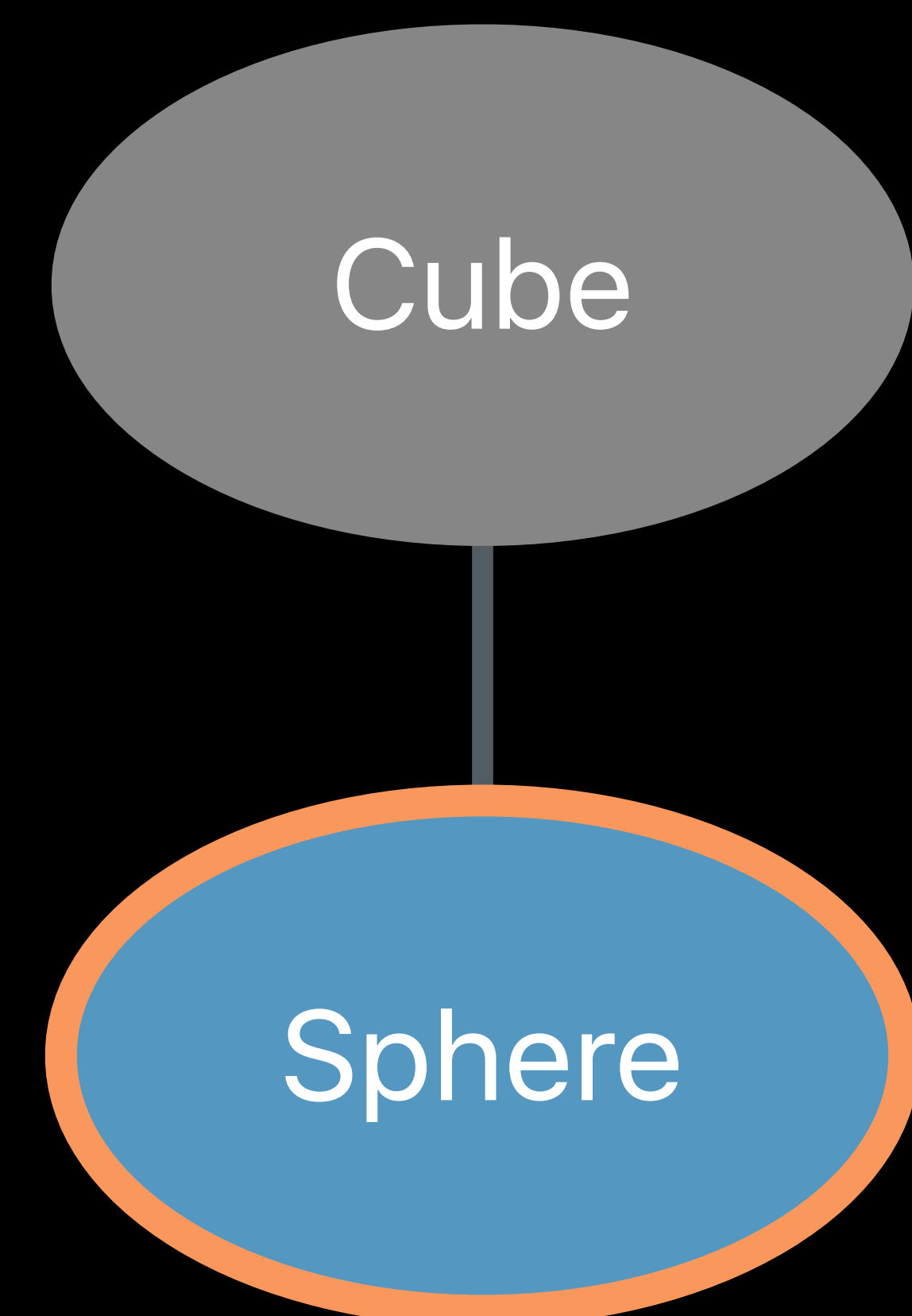
Transforms on a parent also affect its children



Scene Graph

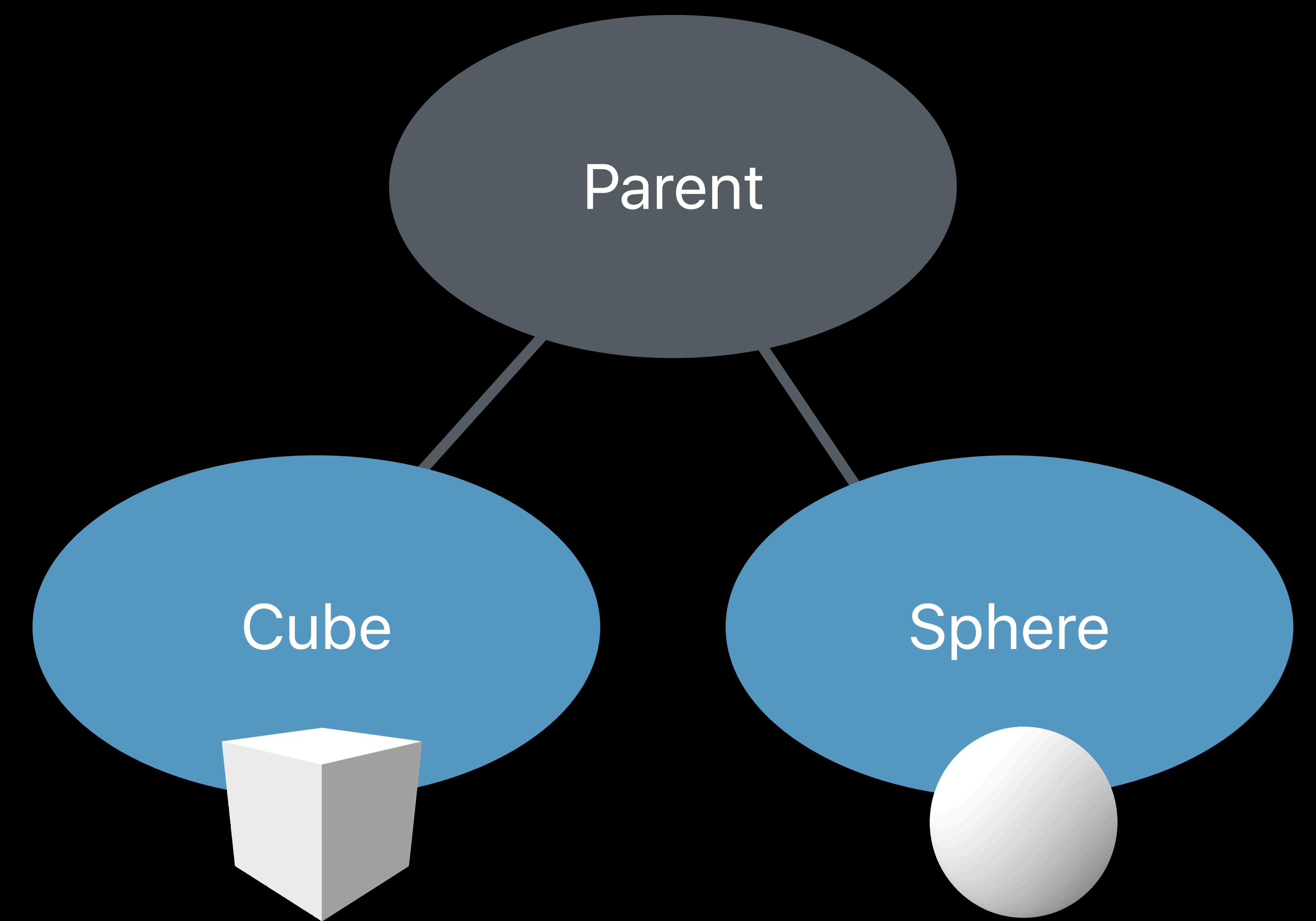
Scene graph defines object hierarchy

Transforms on a parent also affect its children



Scene Graph

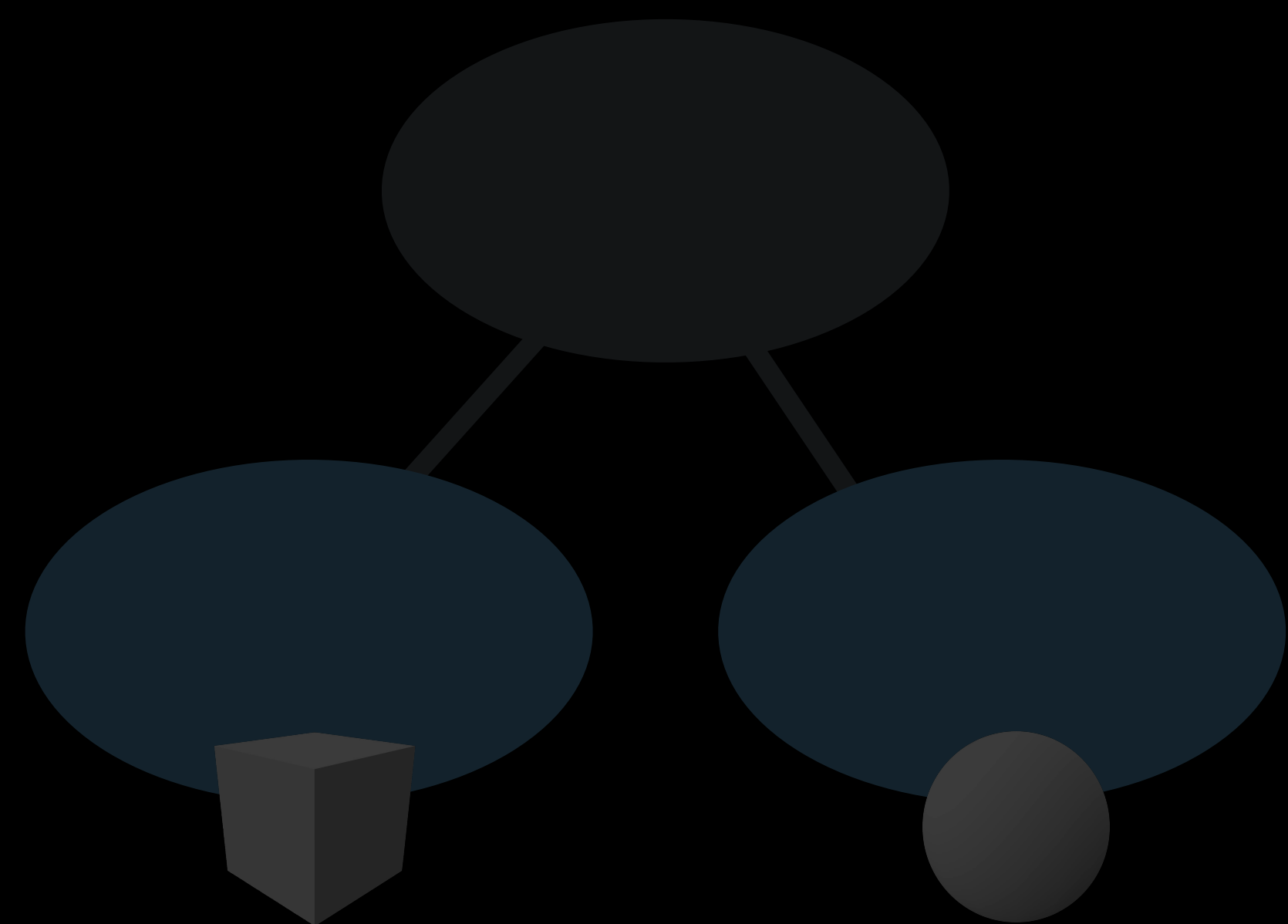
```
def Xform "Parent"  
{  
  def Cube "Cube"  
  {  
  }  
  def Sphere "Sphere"  
  {  
  }  
  def Scope "Materials"  
  {  
    def Material "Glass"  
    {  
    }  
  }  
}
```



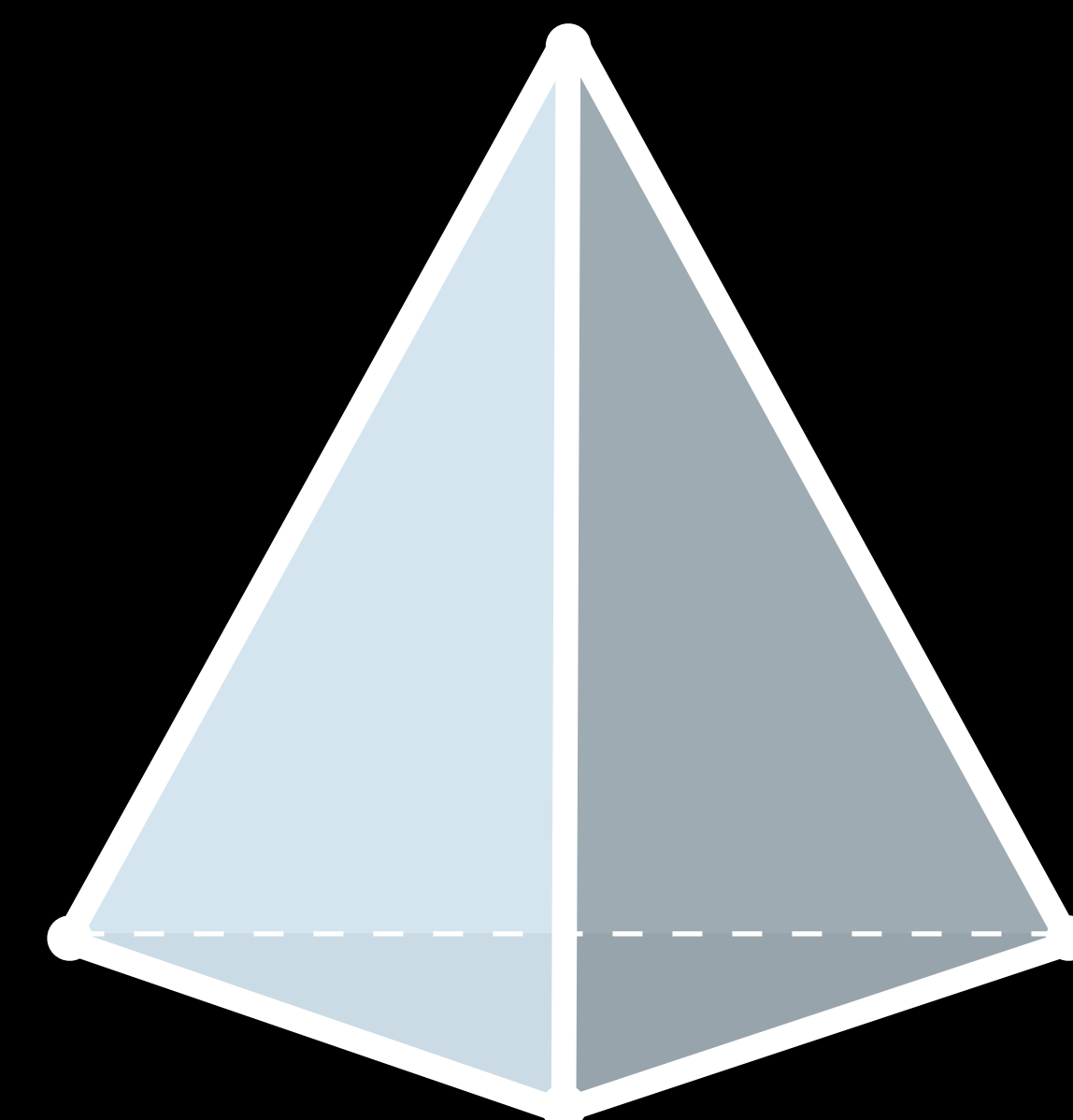
USD Essentials



Basic File Structure



Scene Graph



Mesh Data

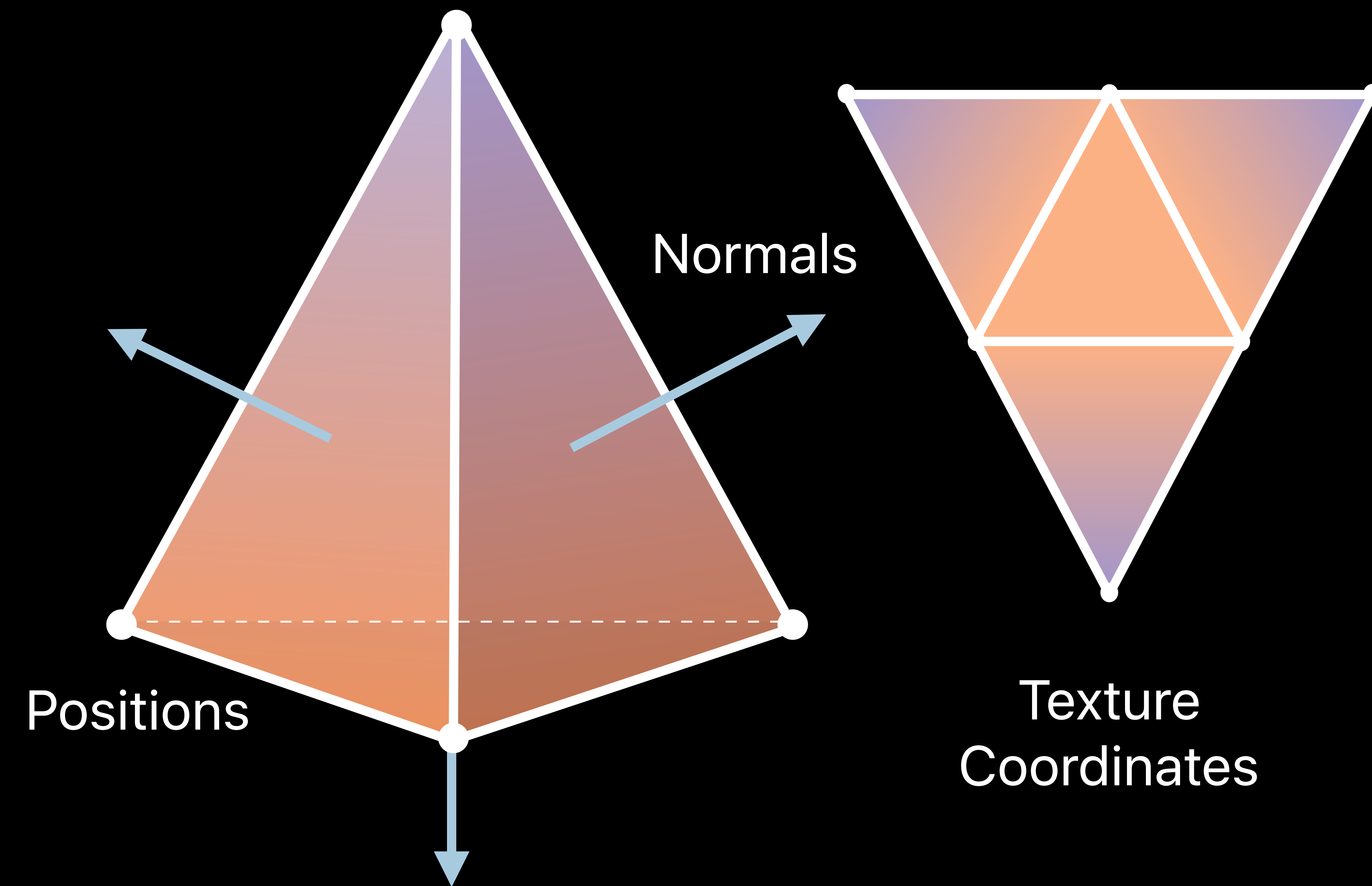


Materials

Mesh Data

Mesh attributes

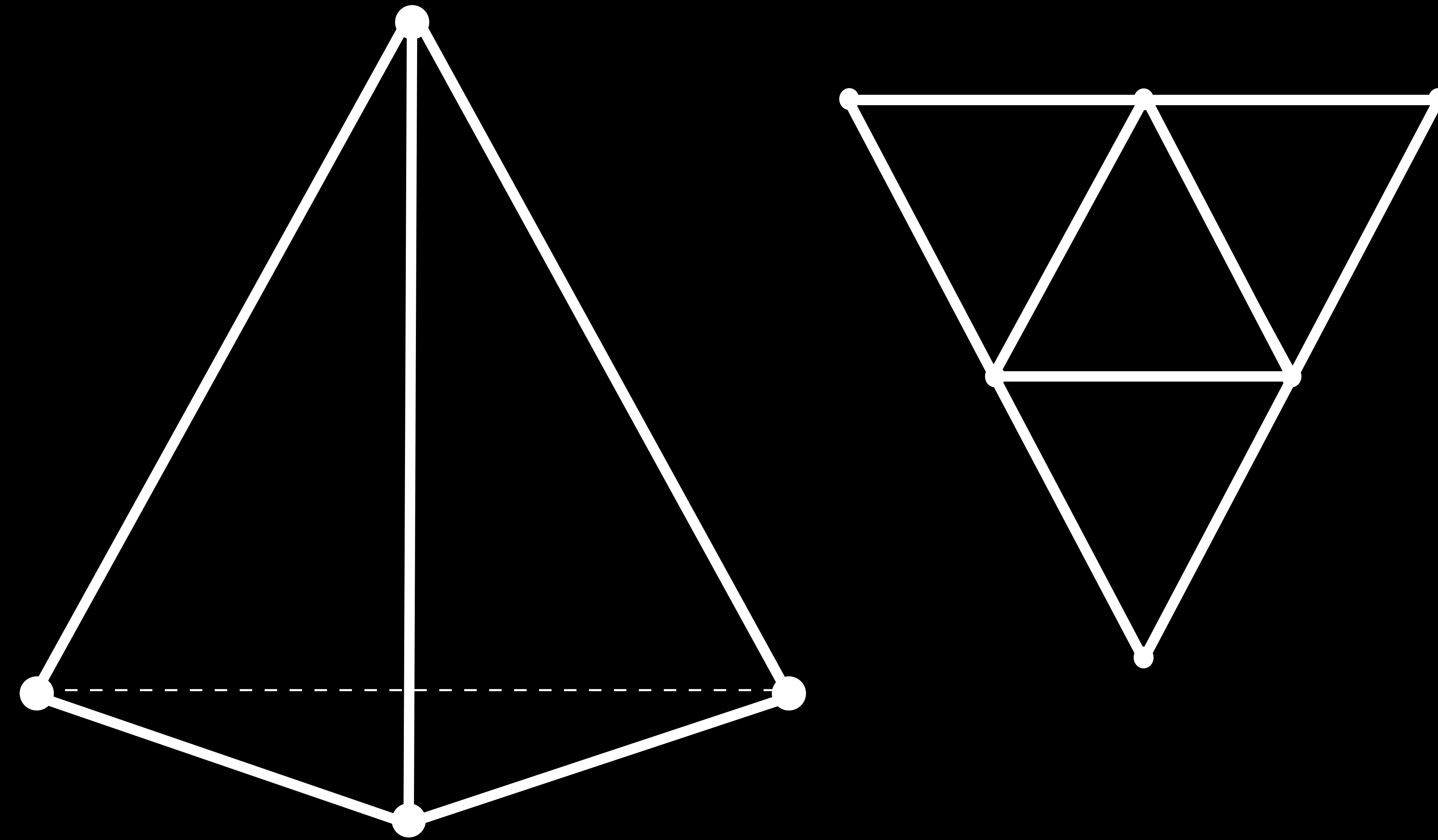
Mesh connectivity



Mesh Data

Mesh attributes

Mesh connectivity



```
def Mesh "Tetrahedron"
{
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
    int[] faceVertexCounts = [3, 3, 3, 3]
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
        interpolation = "uniform"
    )

    float2[] primvars:st = [(0.75,0.43),(0.5,0.87),(0.25,0.43),
                          (1,0.87),(0,0.87),(0.5,0)] (
        interpolation = "faceVarying"
    )

    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

    uniform token subdivisionScheme = "none"
}
```

```
def Mesh "Tetrahedron"  
{  
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]  
    int[] faceVertexCounts = [3, 3, 3, 3]  
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]  
  
    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),  
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (  
        interpolation = "uniform"  
    )  
  
    float2[] primvars:st = [(0.75,0.43), (0.5,0.87), (0.25,0.43),  
                          (1,0.87), (0,0.87), (0.5,0)] (  
        interpolation = "faceVarying"  
    )  
  
    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]  
  
    uniform token subdivisionScheme = "none"  
}
```

```

def Mesh "Tetrahedron"
{
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
    int[] faceVertexCounts = [3, 3, 3, 3]
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

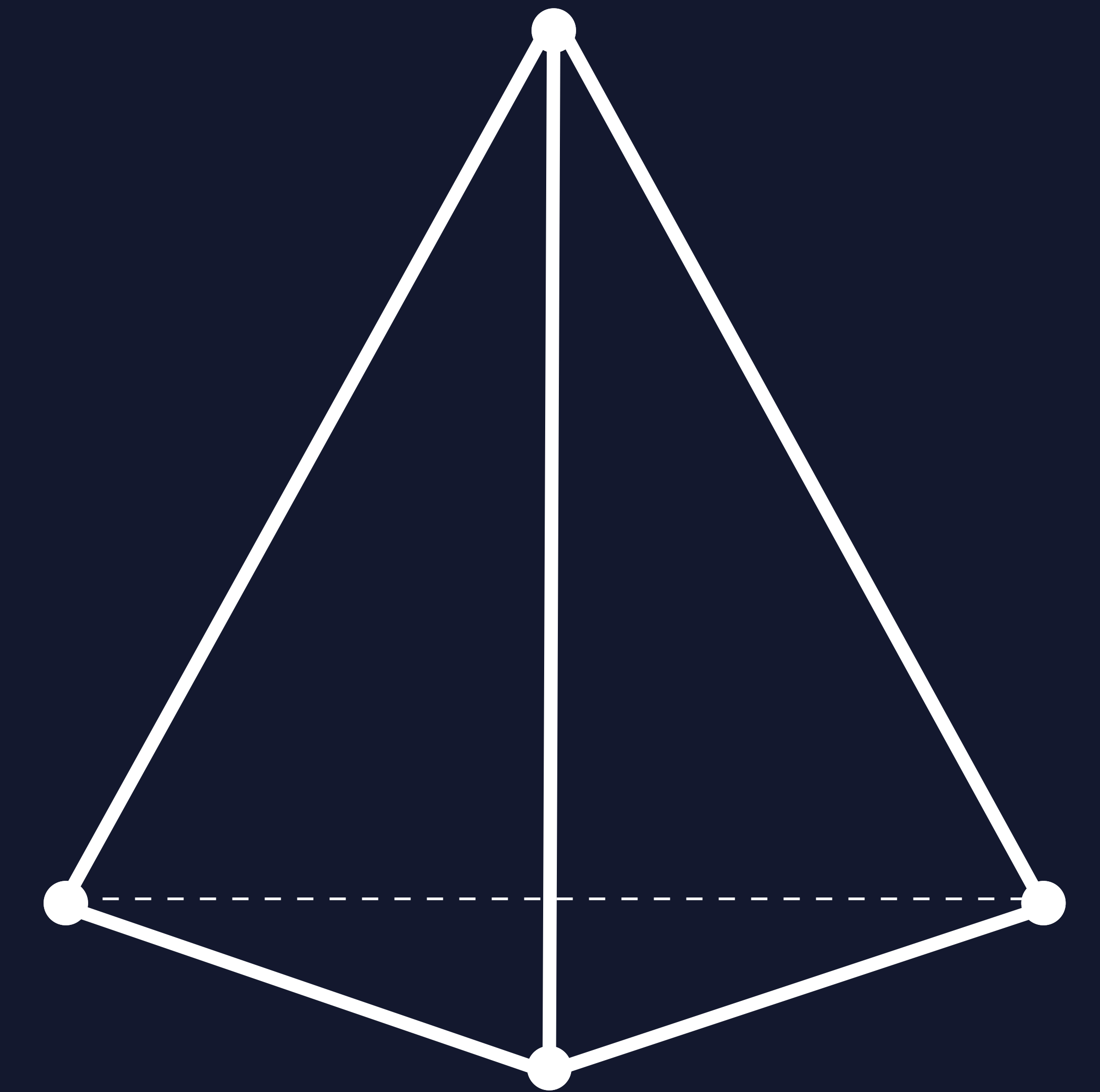
    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
        interpolation = "uniform"
    )

    float2[] primvars:st = [(0.75,0.43),(0.5,0.87),(0.25,0.43),
                          (1,0.87),(0,0.87),(0.5,0)] (
        interpolation = "faceVarying"
    )

    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

    uniform token subdivisionScheme = "none"
}

```



```

def Mesh "Tetrahedron"
{
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
    int[] faceVertexCounts = [3, 3, 3, 3]
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

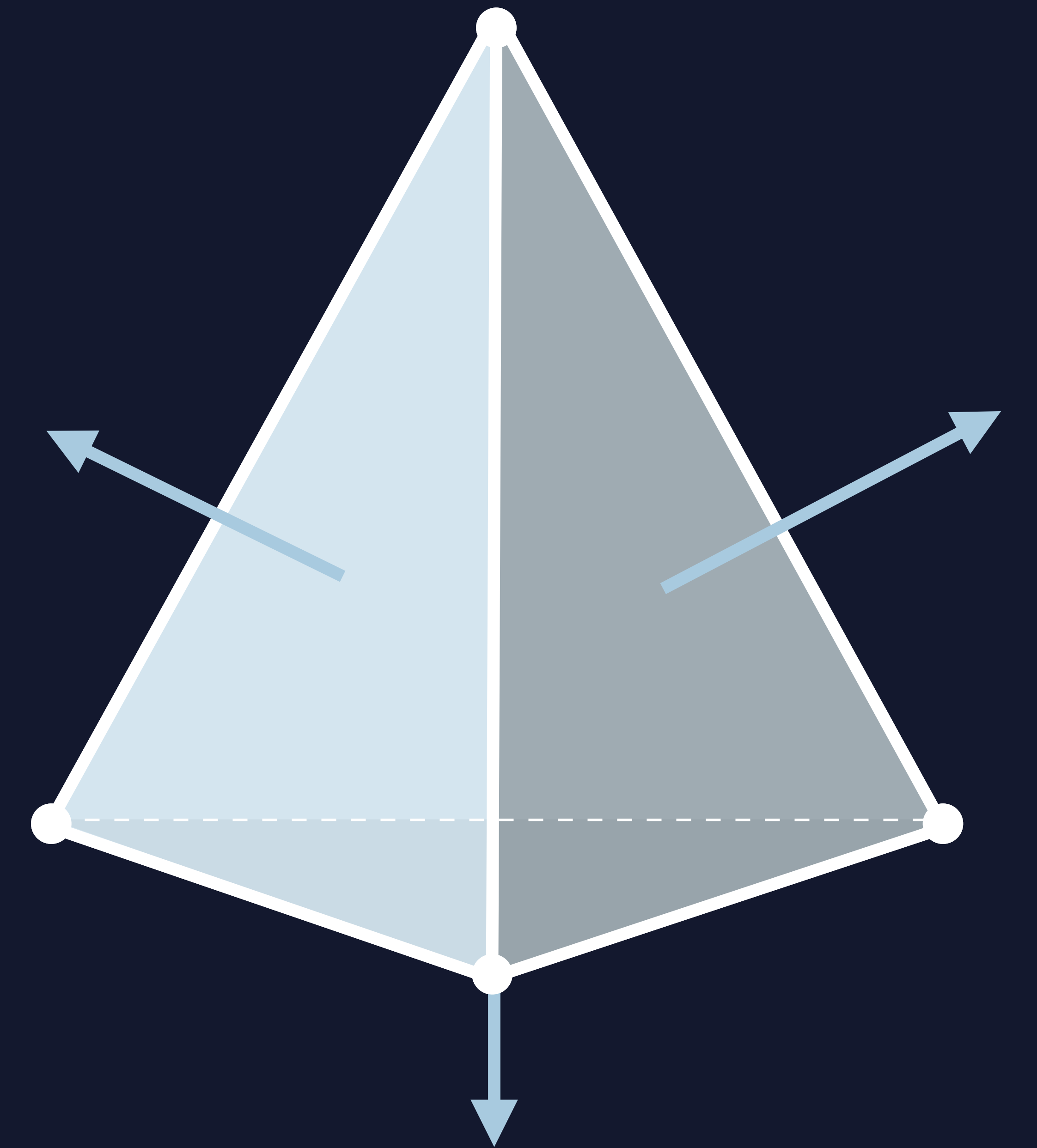
    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
        interpolation = "uniform"
    )

    float2[] primvars:st = [(0.75,0.43), (0.5,0.87), (0.25,0.43),
                          (1,0.87), (0,0.87), (0.5,0)] (
        interpolation = "faceVarying"
    )

    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

    uniform token subdivisionScheme = "none"
}

```




```

def Mesh "Tetrahedron"
{
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
    int[] faceVertexCounts = [3, 3, 3, 3]
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

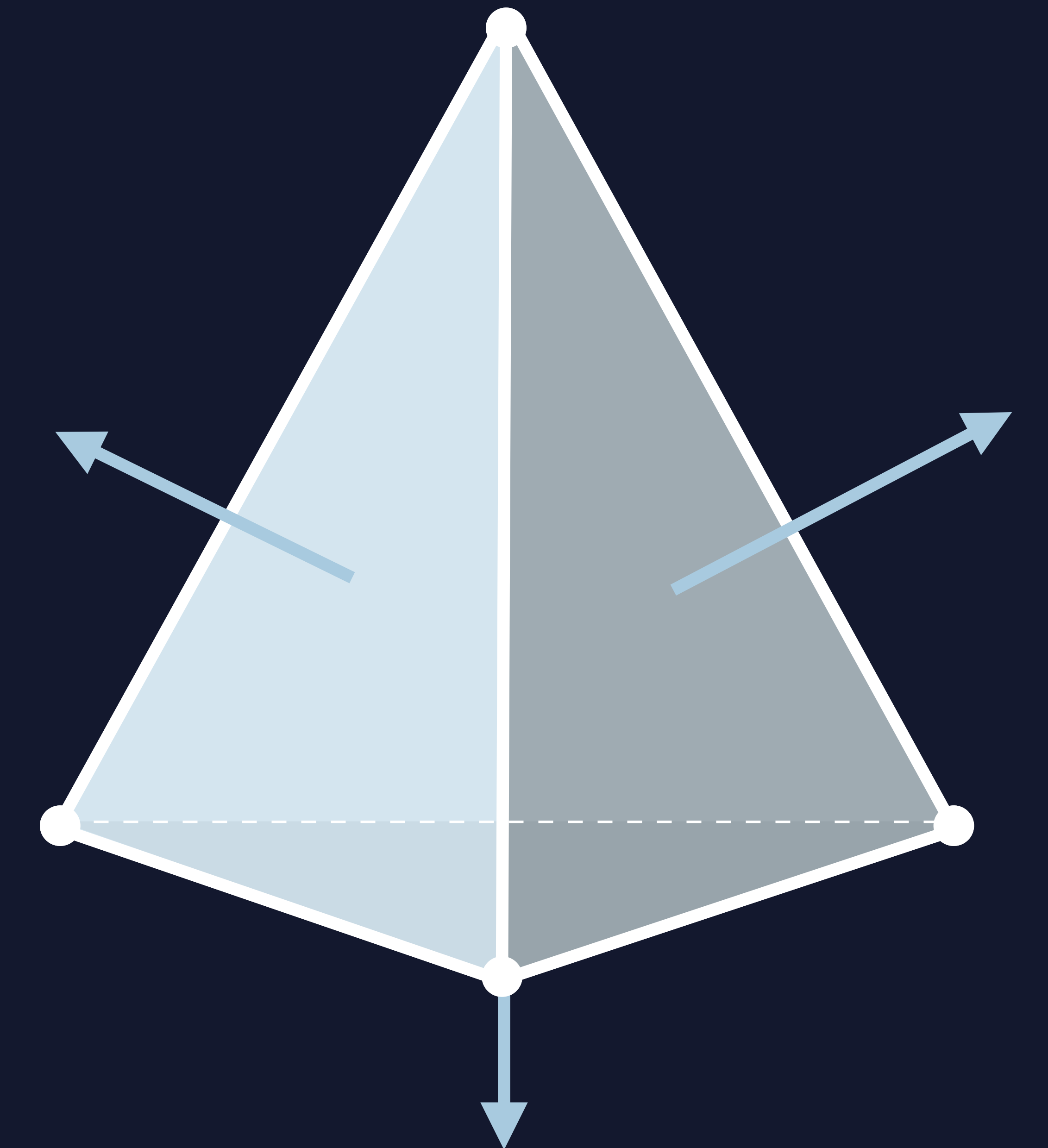
    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
        interpolation = "uniform" ← per-face attribute
    )

    float2[] primvars:st = [(0.75,0.43), (0.5,0.87), (0.25,0.43),
                          (1,0.87), (0,0.87), (0.5,0)] (
        interpolation = "faceVarying"
    )

    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

    uniform token subdivisionScheme = "none"
}

```



```

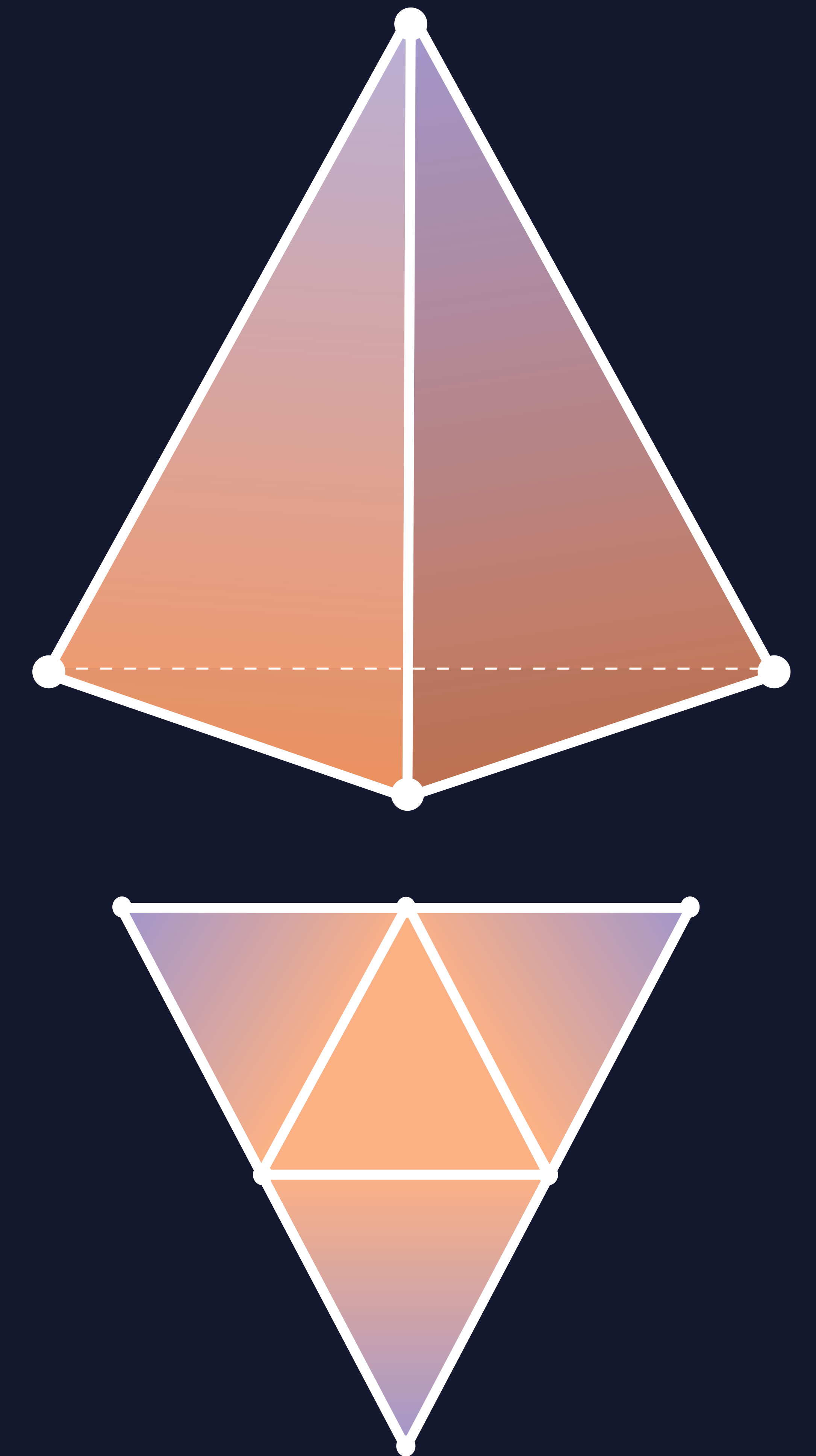
def Mesh "Tetrahedron"
{
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
    int[] faceVertexCounts = [3, 3, 3, 3]
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
        interpolation = "uniform"
    )

    float2[] primvars:st = [(0.75,0.43),(0.5,0.87),(0.25,0.43),
                          (1,0.87),(0,0.87),(0.5,0)] (
        interpolation = "faceVarying"
    )
    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

    uniform token subdivisionScheme = "none"
}

```



```

def Mesh "Tetrahedron"
{
  point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
  int[] faceVertexCounts = [3, 3, 3, 3]
  int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

  normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                        (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
    interpolation = "uniform"
  )
}

```

```

float2[] primvars:st = [(0.75,0.43), (0.5,0.87), (0.25,0.43),
                       (1,0.87), (0,0.87), (0.5,0)] (
  interpolation = "faceVarying" ← per-face per-vertex attribute
)
int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

```

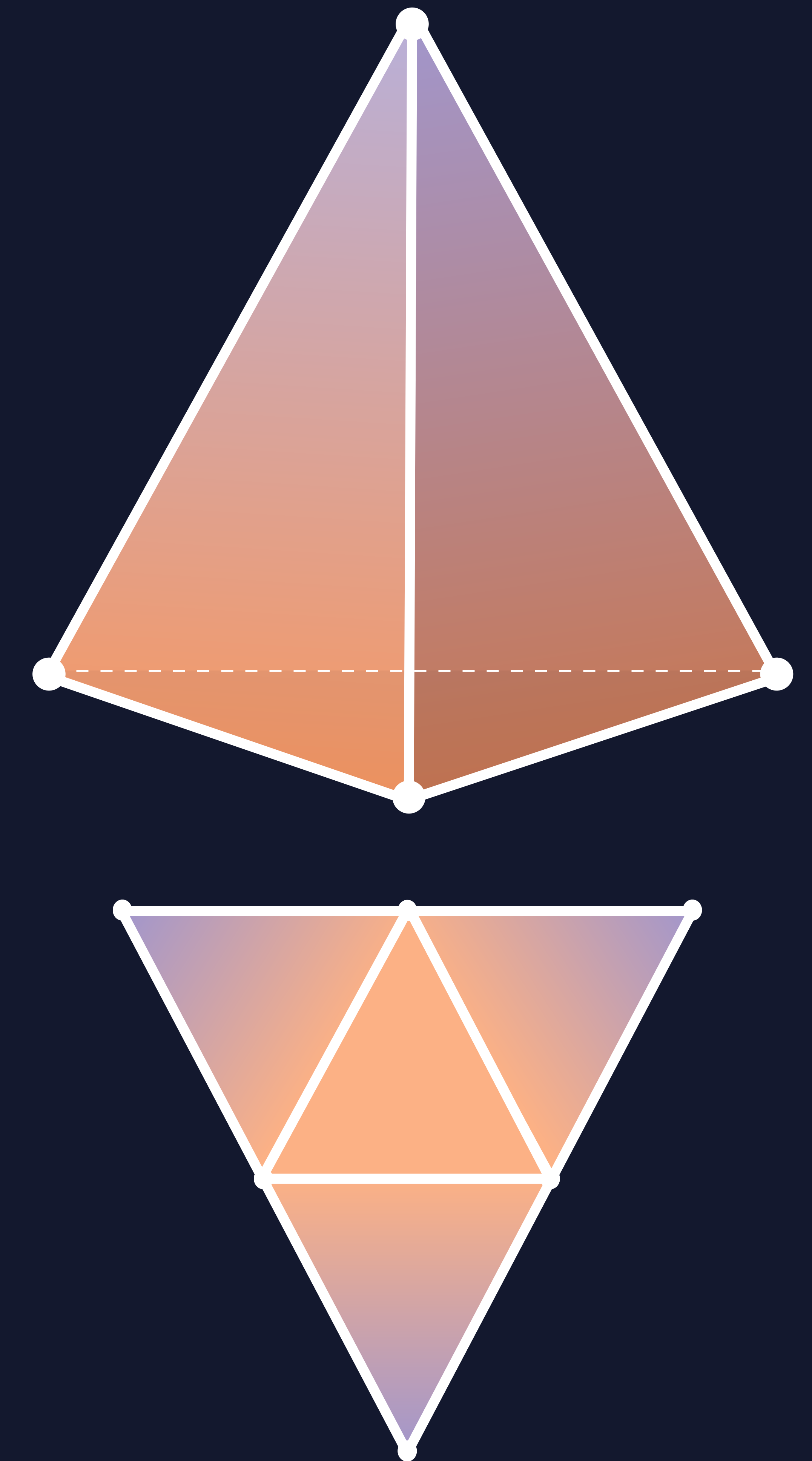
```

uniform token subdivisionScheme = "none"

```

```

}
```



```

def Mesh "Tetrahedron"
{
    point3f[] points = [(-1,-1,1), (-1,1,-1), (1,-1,-1), (1,1,1)]
    int[] faceVertexCounts = [3, 3, 3, 3]
    int[] faceVertexIndices = [1,2,0, 3,2,1, 3,0,2, 3,1,0]

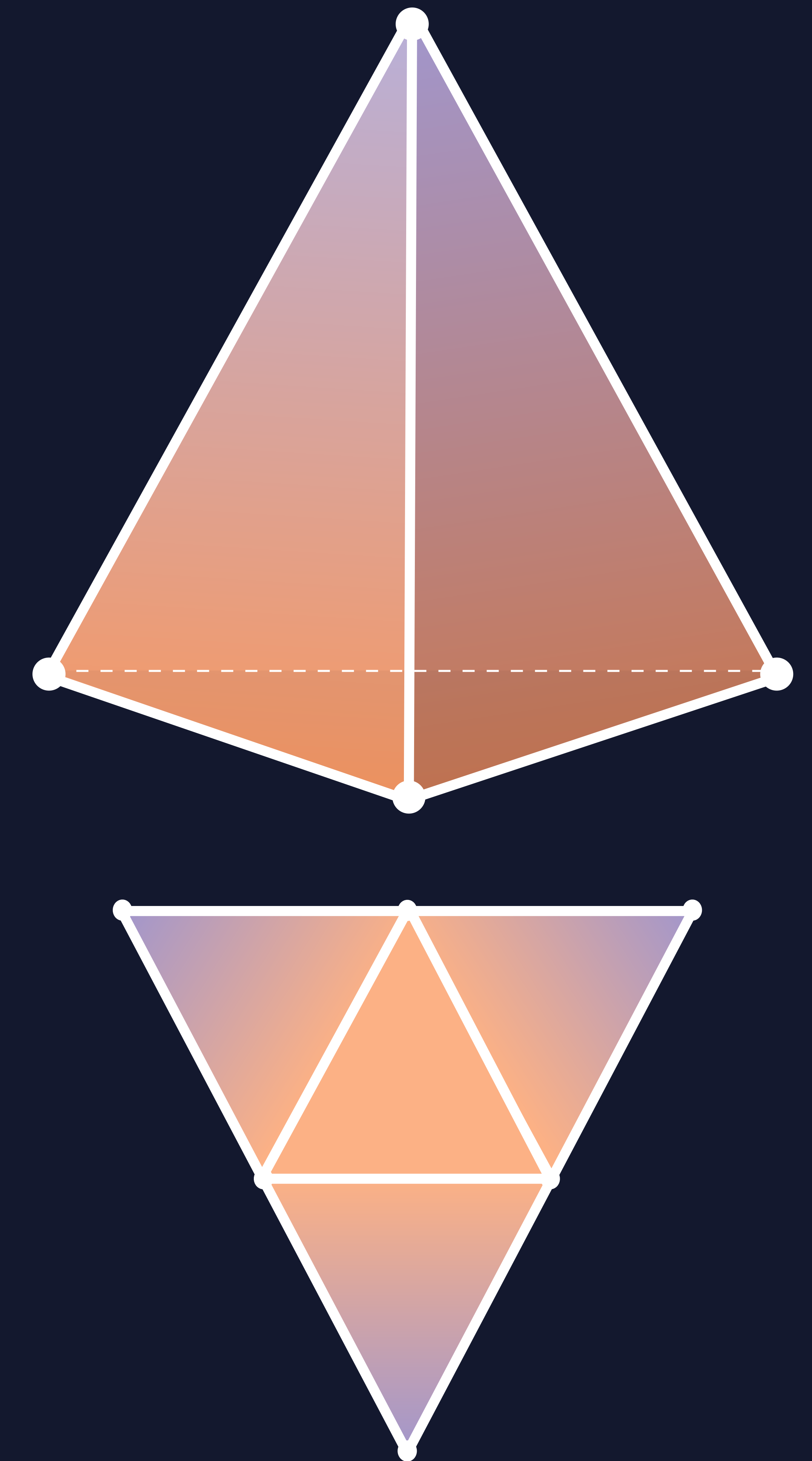
    normal3f[] normals = [(-0.58,-0.58,-0.58), (0.58,0.58,-0.58),
                          (0.58,-0.58,0.58), (-0.58,0.58,0.58)] (
        interpolation = "uniform"
    )

    float2[] primvars:st = [(0.75,0.43),(0.5,0.87),(0.25,0.43),
                          (1,0.87),(0,0.87),(0.5,0)] (
        interpolation = "faceVarying"
    )

    int[] primvars:st:indices = [0,1,2, 3,1,0, 4,2,1, 5,0,2]

    uniform token subdivisionScheme = "none"
}

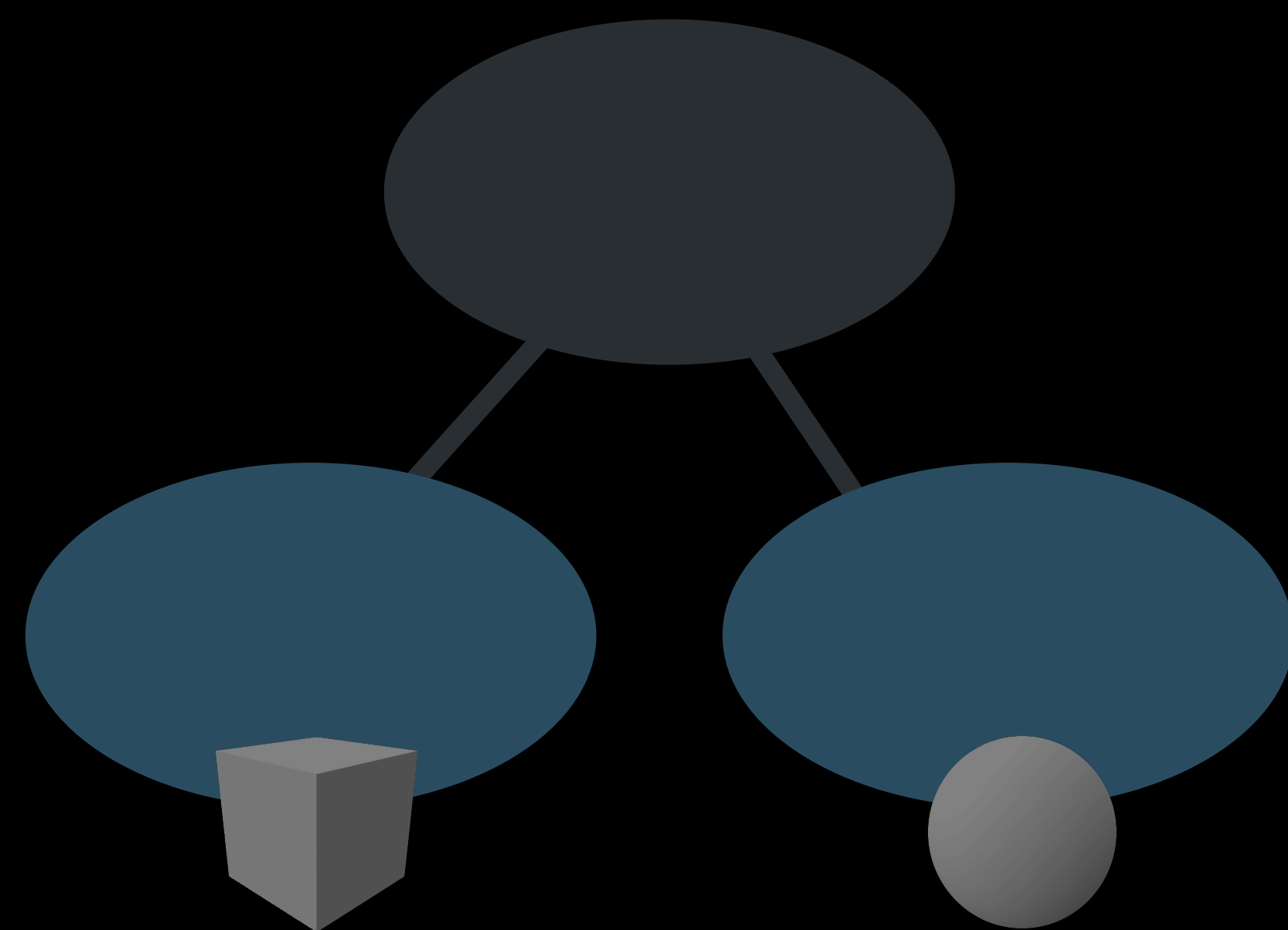
```



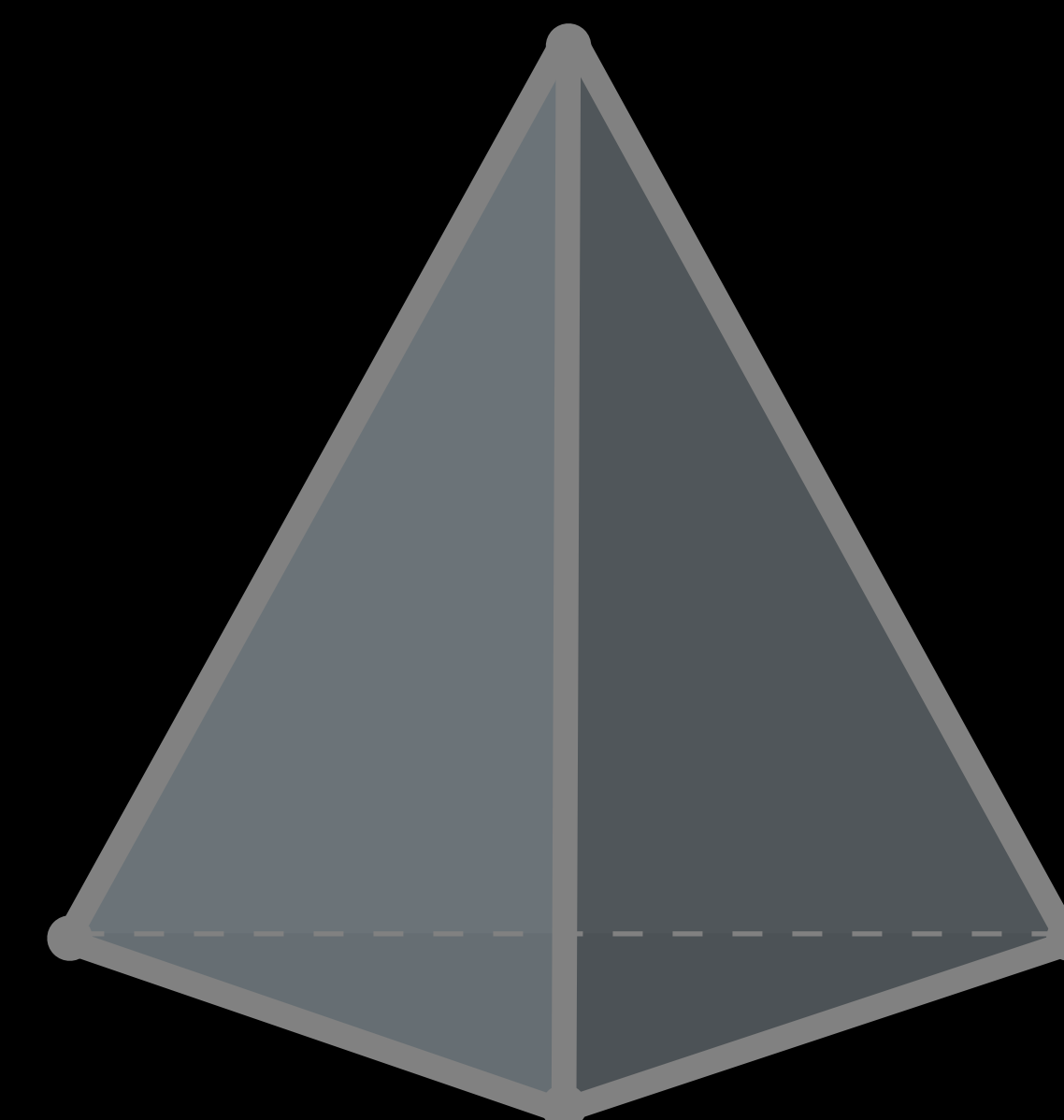
USD Essentials



Basic File Structure



Scene Graph



Mesh Data



Materials

Materials

UsdPreviewSurface

- Schema for realistic real-time rendering

Physically based material description

Supported workflows

- Metallic-roughness
- Specular-roughness

Material Properties

UsdPreviewSurface

Example

```
diffuseColor = (1,1,1)
```



Material Properties

UsdPreviewSurface

Example

```
diffuseColor = (1,1,1)  
normal = normals.png  
occlusion = occlusion.png
```



Material Properties

UsdPreviewSurface

Example

```
diffuseColor = (1,1,1)  
normal = normals.png  
occlusion = occlusion.png  
metallic = 1
```



Material Properties

UsdPreviewSurface

Example

```
diffuseColor = (1,1,1)  
normal = normals.png  
occlusion = occlusion.png  
metallic = 1  
roughness = roughness.png
```



Material Properties

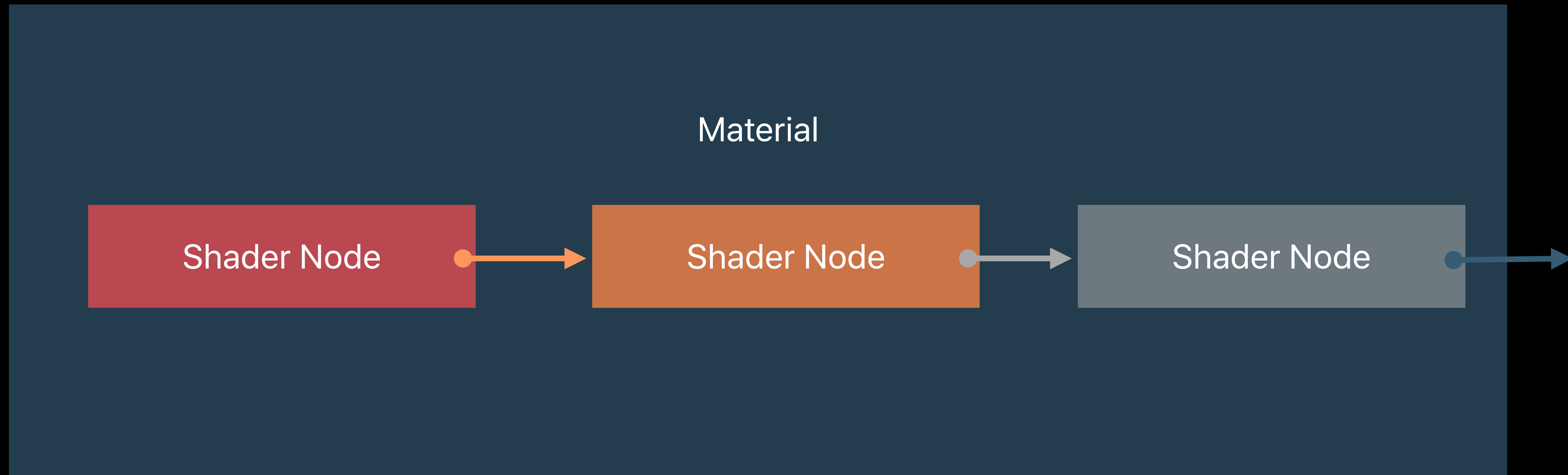
UsdPreviewSurface

Example

```
diffuseColor = diffuse.png  
normal = normals.png  
occlusion = occlusion.png  
metallic = metallic.png  
roughness = roughness.png
```



Shader Node Graph

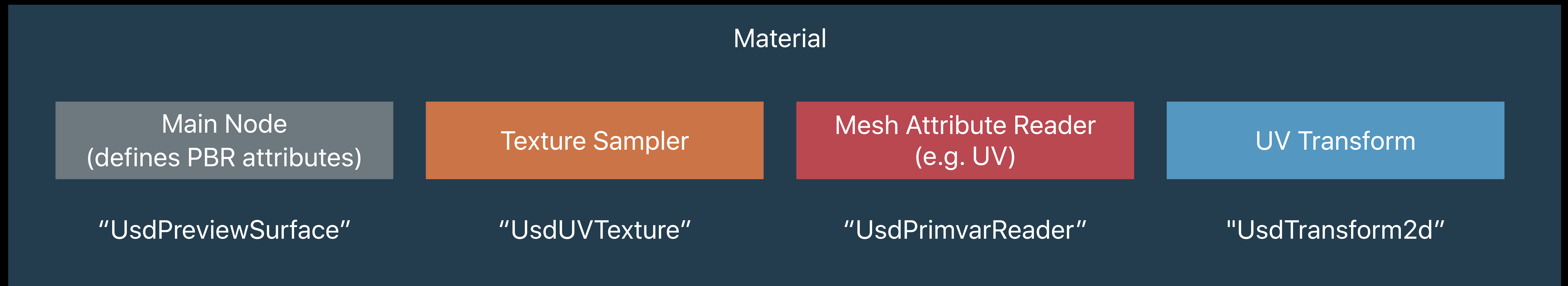


Shader Node Graph

UsdPreviewSurface

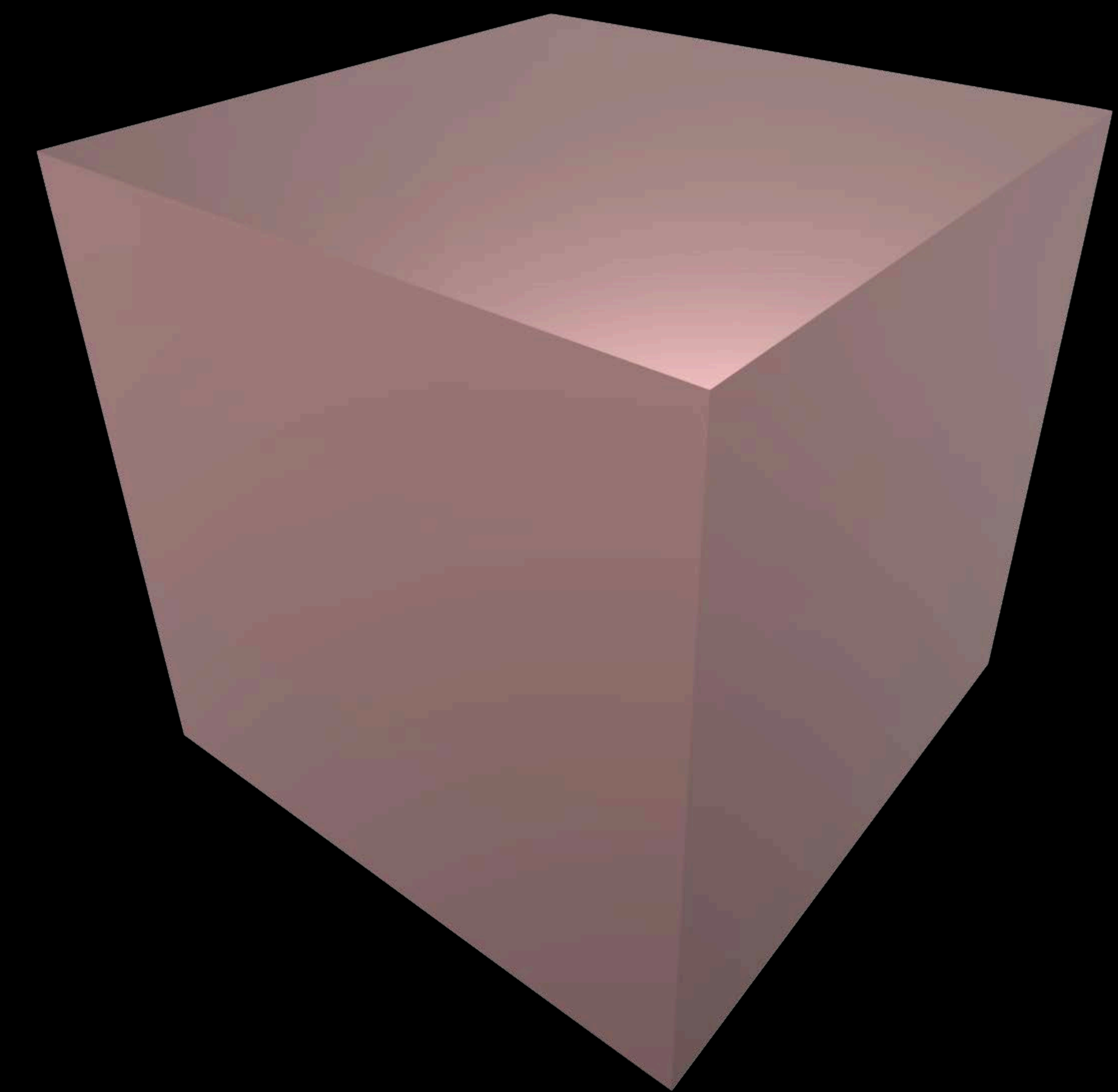
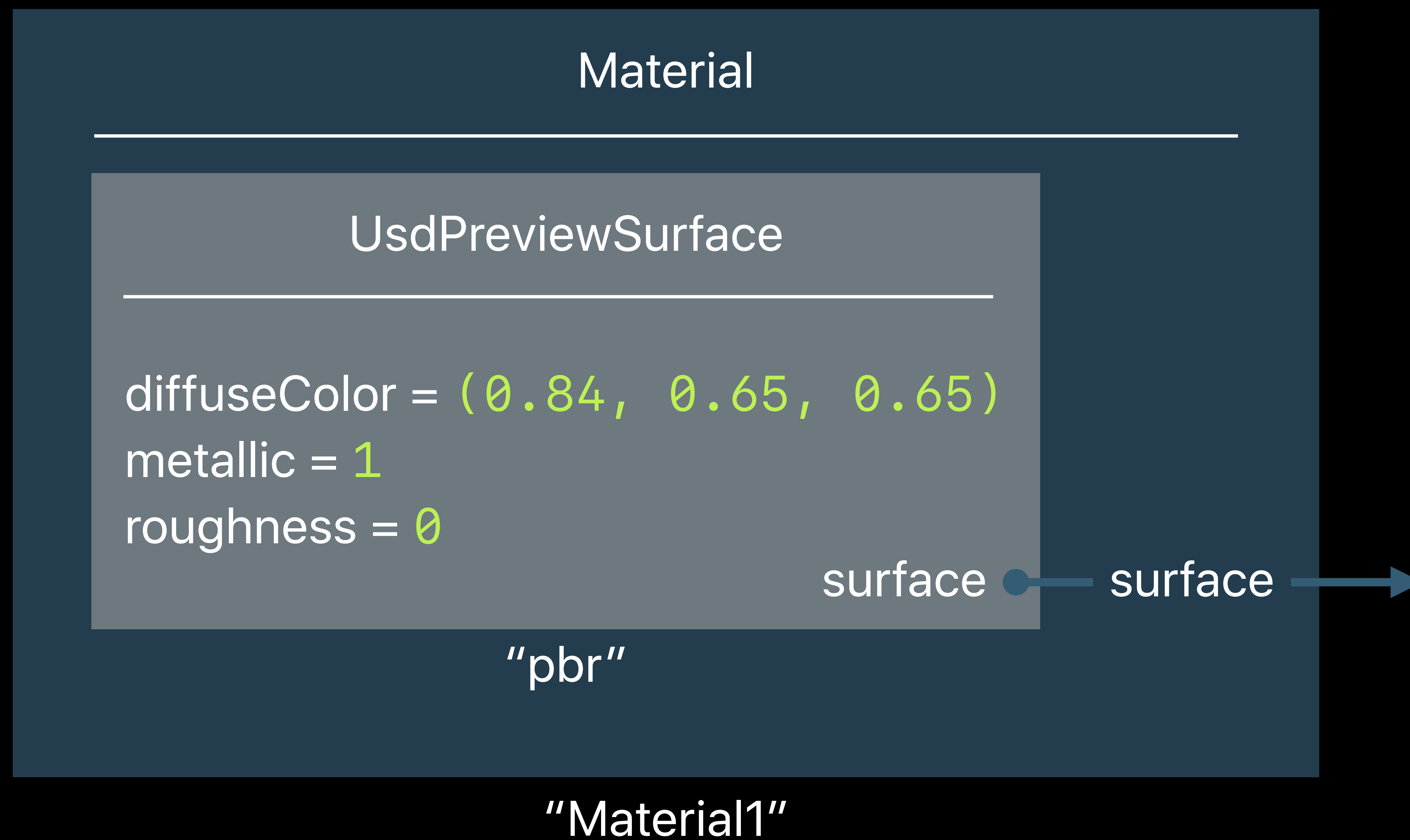
Simple shader node graph structure

Four node types



Example

Constant material property



Example

Constant material property

```
def Material "Material1"  
{  
  def Shader "pbr"  
  {  
    uniform token info:id = "UsdPreviewSurface"  
    color3f inputs:diffuseColor = (0.84, 0.65, 0.65)  
    float inputs:metallic = 1  
    float inputs:roughness = 0  
    token outputs:surface  
  }  
  token outputs:surface.connect = <../Material1/pbr.outputs:surface>  
}
```

Example

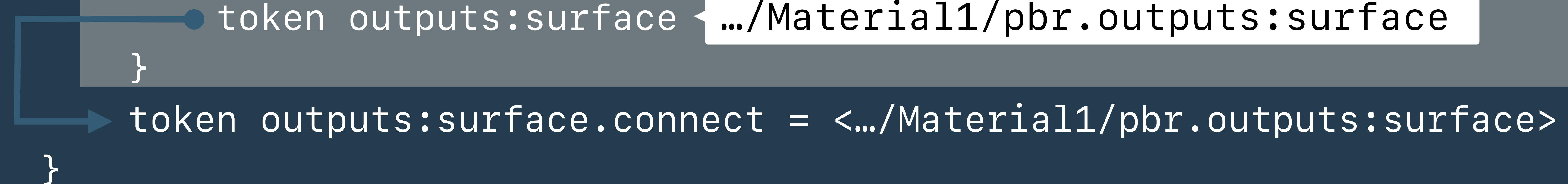
Constant material property

```
def Material "Material1"  
{  
  def Shader "pbr"  
  {  
    uniform token info:id = "UsdPreviewSurface"  
    color3f inputs:diffuseColor = (0.84, 0.65, 0.65)  
    float inputs:metallic = 1  
    float inputs:roughness = 0  
    token outputs:surface  
  }  
  token outputs:surface.connect = <../Material1/pbr.outputs:surface>  
}
```


Example

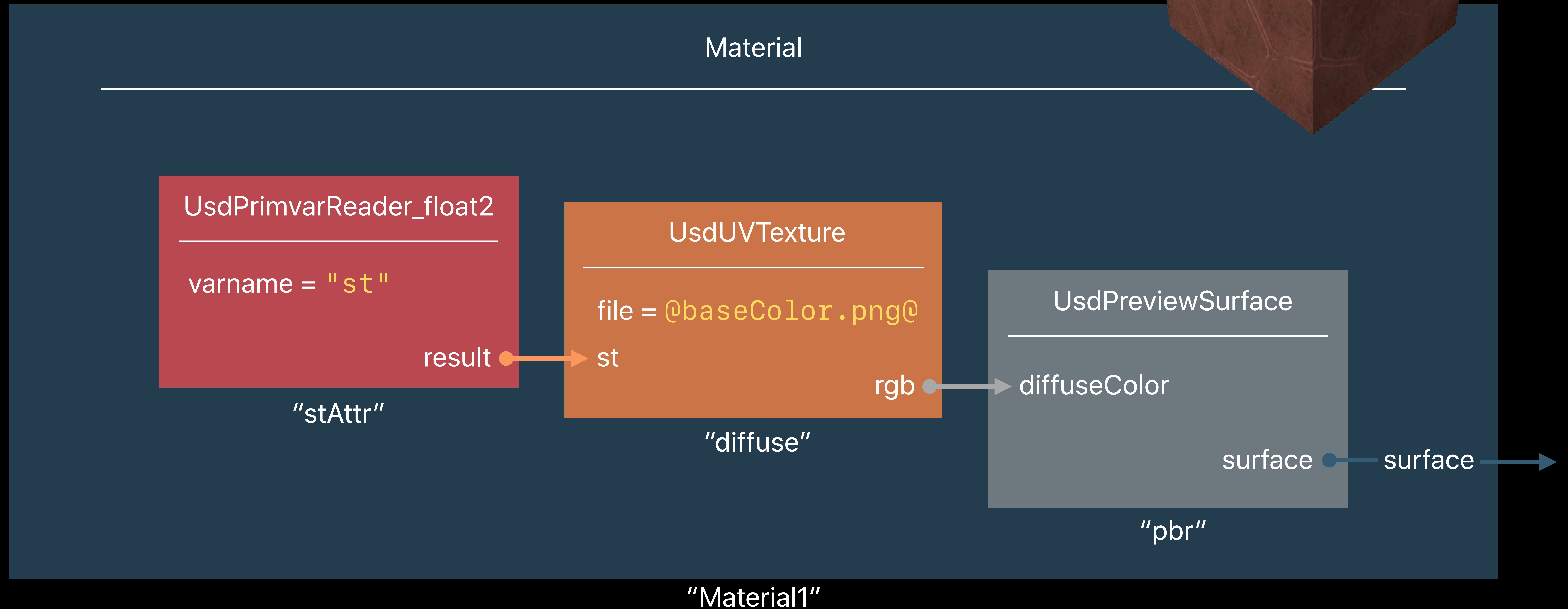
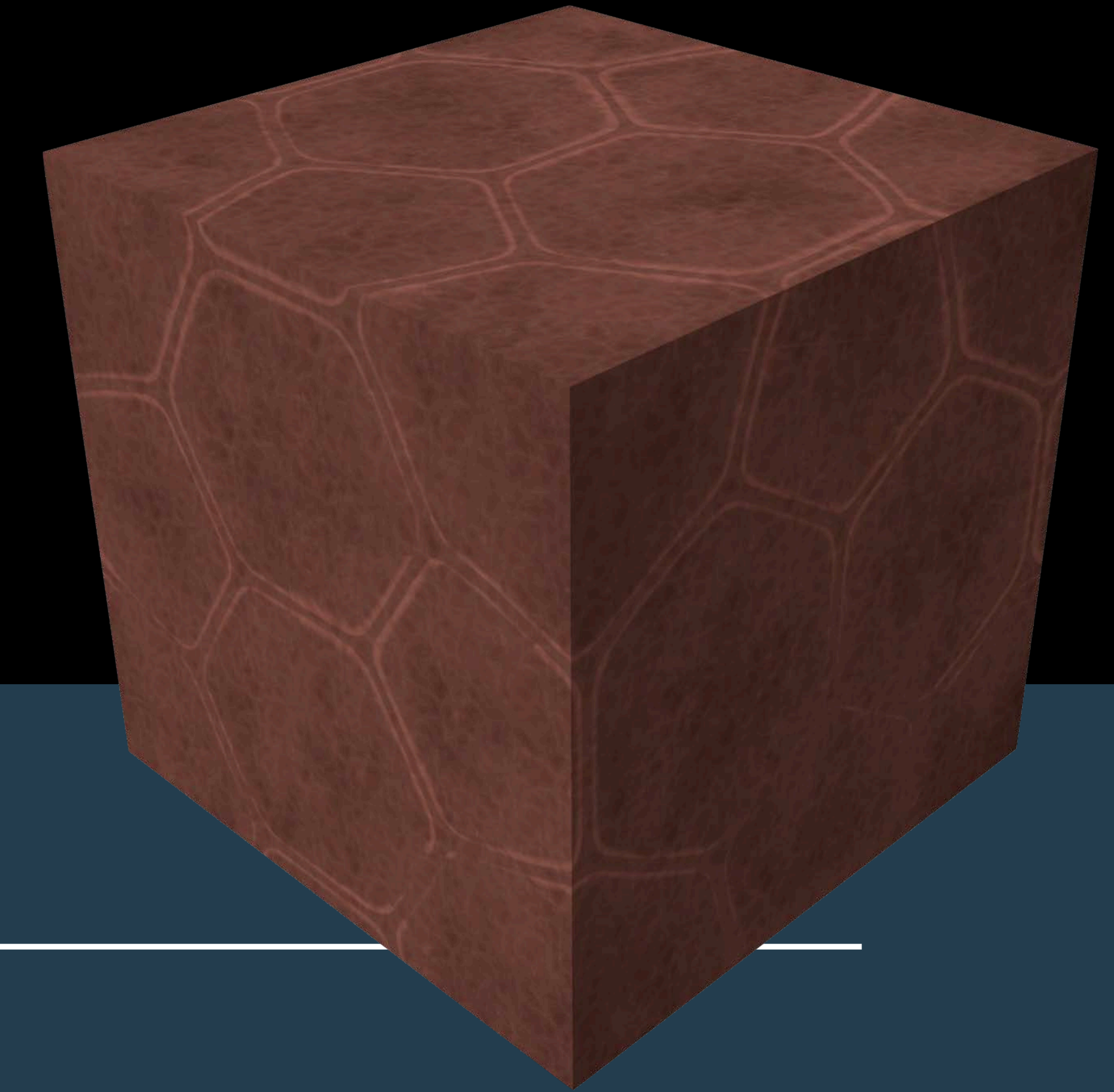
Constant material property

```
def Material "Material1"  
{  
  def Shader "pbr"  
  {  
    uniform token info:id = "UsdPreviewSurface"  
    color3f inputs:diffuseColor = (0.84, 0.65, 0.65)  
    float inputs:metallic = 1  
    float inputs:roughness = 0  
    token outputs:surface ← .../Material1/pbr.outputs:surface  
  }  
  token outputs:surface.connect = <.../Material1/pbr.outputs:surface>  
}
```



Example

Texture material property



```
def Material "Material1"
{
  def Shader "stAttr"
  {
    uniform token info:id = "UsdPrimvarReader_float2"
    token inputs:varname = "st"
    float2 outputs:result
  }
  def Shader "diffuse"
  {
    uniform token info:id = "UsdUVTexture"
    asset inputs:file = @baseColor.png@
    float2 inputs:st.connect = <../Material1/stAttr.outputs:result>
    float3 outputs:rgb
  }
  def Shader "pbr"
  {
    uniform token info:id = "UsdPreviewSurface"
    color3f inputs:diffuseColor.connect = <../Material1/diffuse.outputs:rgb>
    token outputs:surface
  }
  token outputs:surface.connect = <../Material1/pbr.outputs:surface>
}
}
```

```
def Material "Material1"
{
  def Shader "stAttr"
  {
    uniform token info:id = "UsdPrimvarReader_float2"
    token inputs:varname = "st"
    float2 outputs:result
  }
  def Shader "diffuse"
  {
    uniform token info:id = "UsdUVTexture"
    asset inputs:file = @baseColor.png@
    float2 inputs:st.connect = <../Material1/stAttr.outputs:result>
    float3 outputs:rgb
  }
  def Shader "pbr"
  {
    uniform token info:id = "UsdPreviewSurface"
    color3f inputs:diffuseColor.connect = <../Material1/diffuse.outputs:rgb>
    token outputs:surface
  }
  token outputs:surface.connect = <../Material1/pbr.outputs:surface>
}
}
```

```
def Material "Material1"
{
  def Shader "stAttr"
  {
    uniform token info:id = "UsdPrimvarReader_float2"
    token inputs:varname = "st"
    float2 outputs:result
  }

  def Shader "diffuse"
  {
    uniform token info:id = "UsdUVTexture"
    asset inputs:file = @baseColor.png@
    float2 inputs:st.connect = <../Material1/stAttr.outputs:result>
    float3 outputs:rgb
  }

  def Shader "pbr"
  {
    uniform token info:id = "UsdPreviewSurface"
    color3f inputs:diffuseColor.connect = <../Material1/diffuse.outputs:rgb>
    token outputs:surface
  }
  token outputs:surface.connect = <../Material1/pbr.outputs:surface>
}
}
```

```
def Material "Material1"
{
  def Shader "stAttr"
  {
    uniform token info:id = "UsdPrimvarReader_float2"
    token inputs:varname = "st"
    float2 outputs:result
  }

  def Shader "diffuse"
  {
    uniform token info:id = "UsdUVTexture"
    asset inputs:file = @baseColor.png@
    float2 inputs:st.connect = <../Material1/stAttr.outputs:result>
    float3 outputs:rgb
  }

  def Shader "pbr"
  {
    uniform token info:id = "UsdPreviewSurface"
    color3f inputs:diffuseColor.connect = <../Material1/diffuse.outputs:rgb>
    token outputs:surface
  }

  token outputs:surface.connect = <../Material1/pbr.outputs:surface>
}
}
```

```
def Material "Material1"
{
  def Shader "stAttr"
  {
    uniform token info:id = "UsdPrimvarReader_float2"
    token inputs:varname = "st"
    float2 outputs:result
  }
  def Shader "diffuse"
  {
    uniform token info:id = "UsdUVTexture"
    asset inputs:file = @baseColor.png@
    float2 inputs:st.connect = <../Material1/stAttr.outputs:result>
    float3 outputs:rgb
  }
  def Shader "pbr"
  {
    uniform token info:id = "UsdPreviewSurface"
    color3f inputs:diffuseColor.connect = <../Material1/diffuse.outputs:rgb>
    token outputs:surface
  }
  token outputs:surface.connect = <../Material1/pbr.outputs:surface>
}
}
```

The diagram illustrates the data flow between three shaders defined within a material:

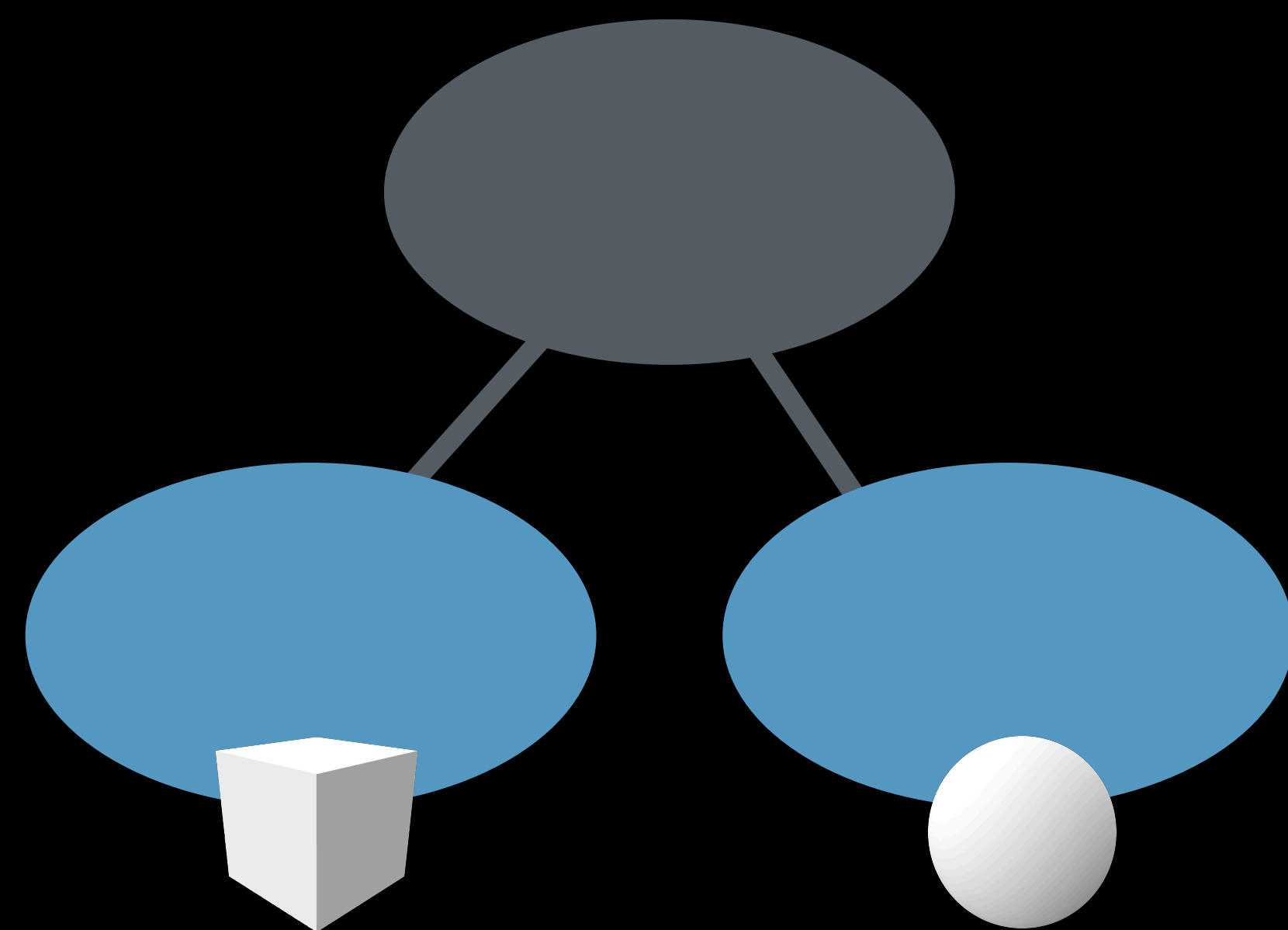
- stAttr Shader (Red background):** Outputs a `float2` result. A red dot is placed on the `float2 outputs:result` line.
- diffuse Shader (Orange background):** Takes a `float2` input from `stAttr` (indicated by an orange arrow) and outputs a `float3` rgb. A red dot is placed on the `float3 outputs:rgb` line.
- pbr Shader (Grey background):** Takes a `color3f` input from the `diffuse` shader (indicated by a grey arrow) and outputs a `token` surface. A blue dot is placed on the `token outputs:surface` line.

Finally, the material's `token outputs:surface` is connected to the `pbr` shader's `token outputs:surface` (indicated by a blue arrow).

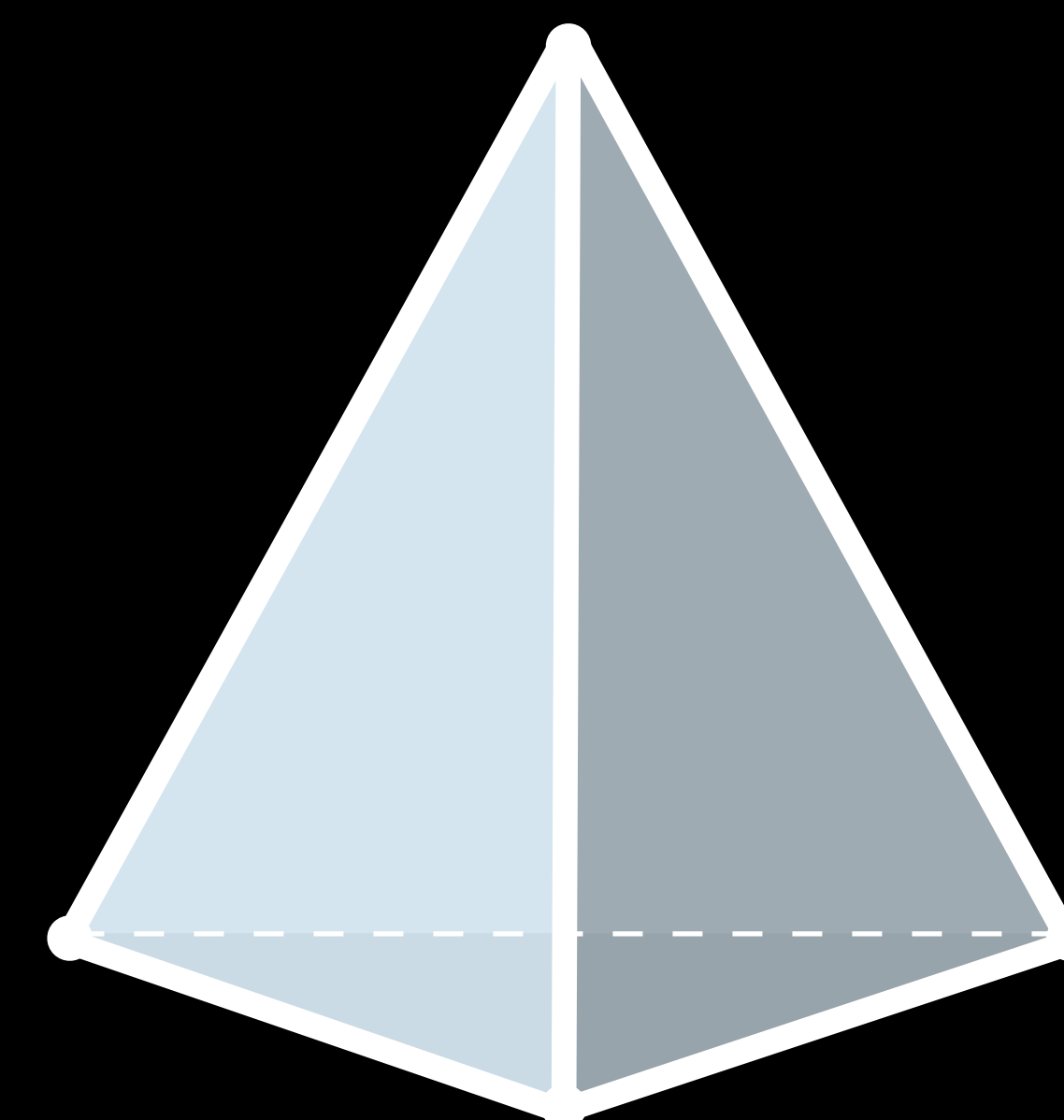
USD Essentials



Basic File Structure



Scene Graph



Mesh Data



Materials

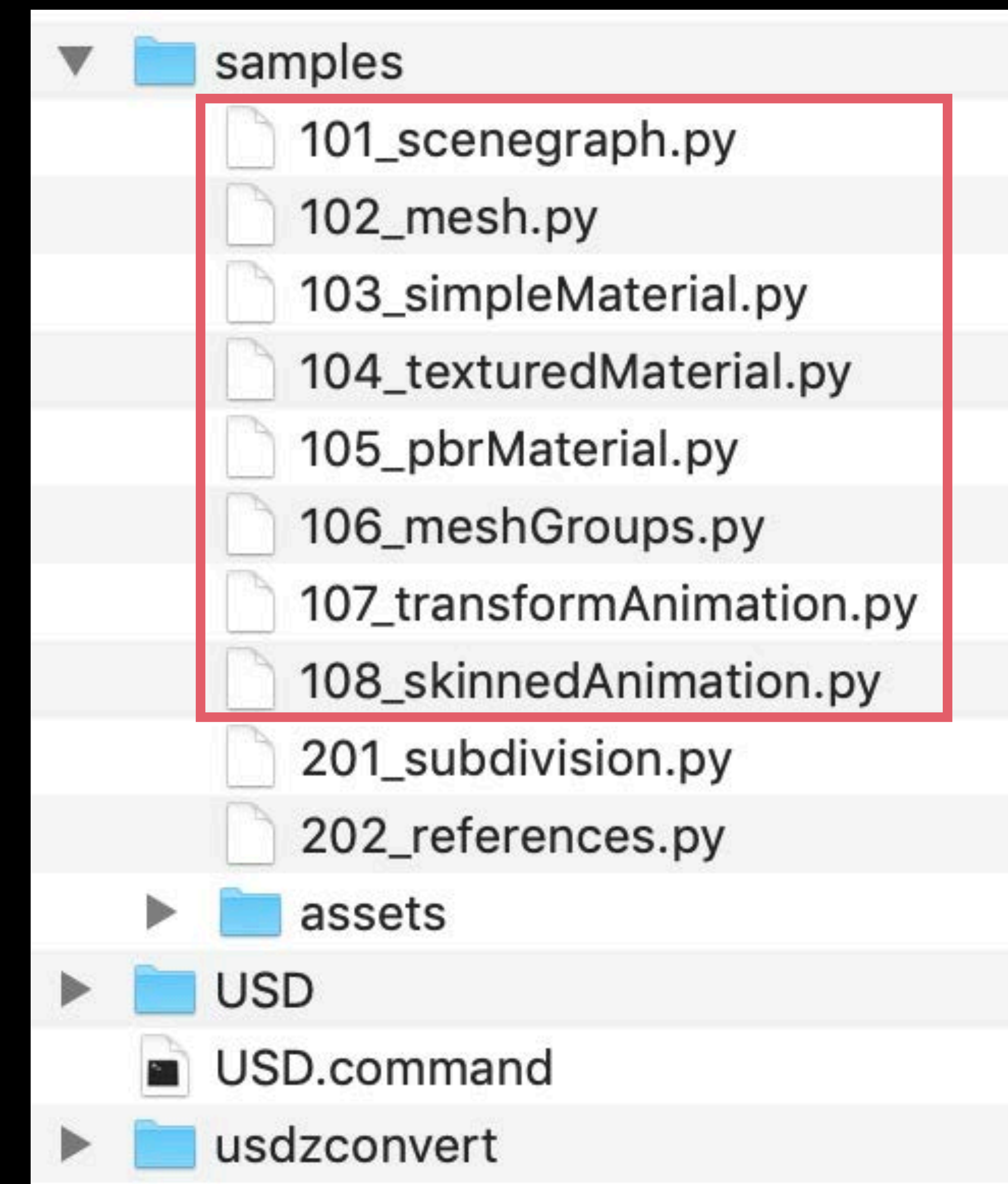
Python Samples

USD essentials

NEW

“Samples” folder

- Scene graph
- Mesh
- Material and mesh groups
- Transform animations
- Skinning and skeletal animations

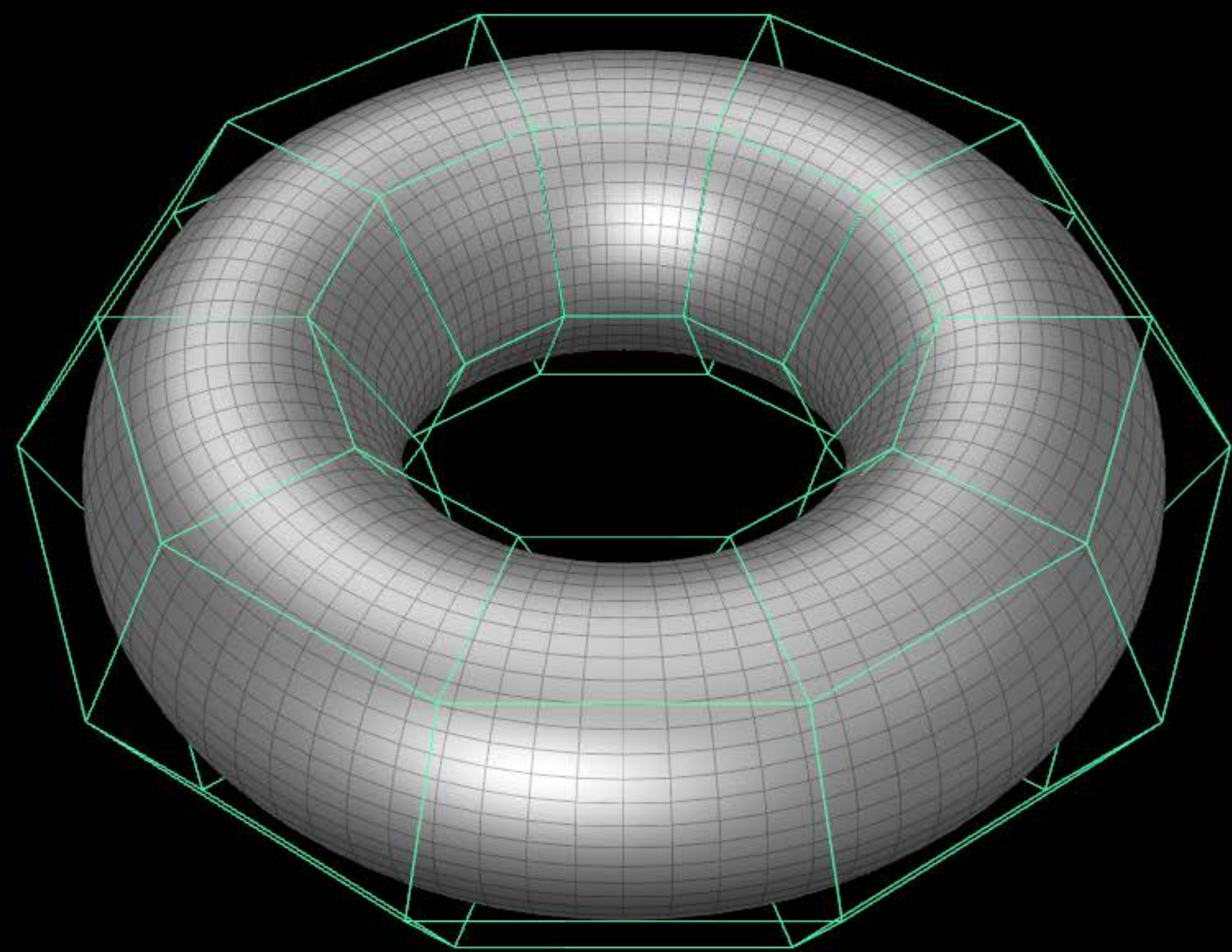


What is USD?

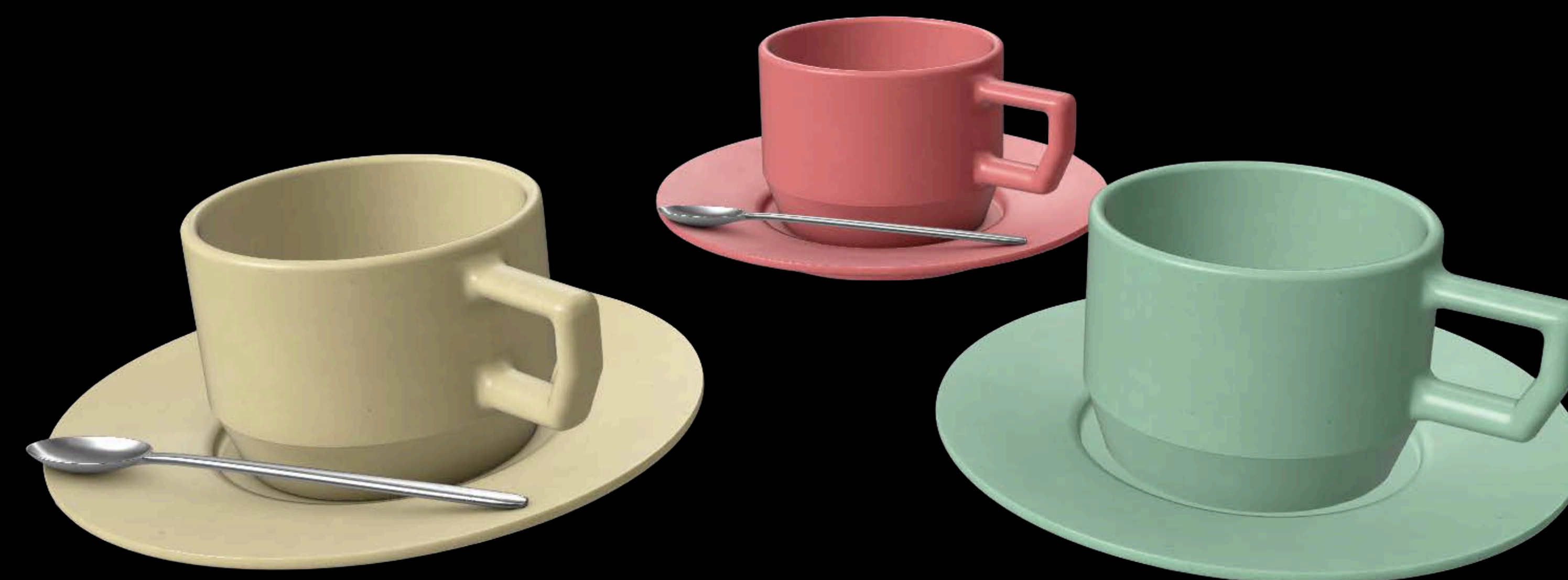
Workflows

USD essentials

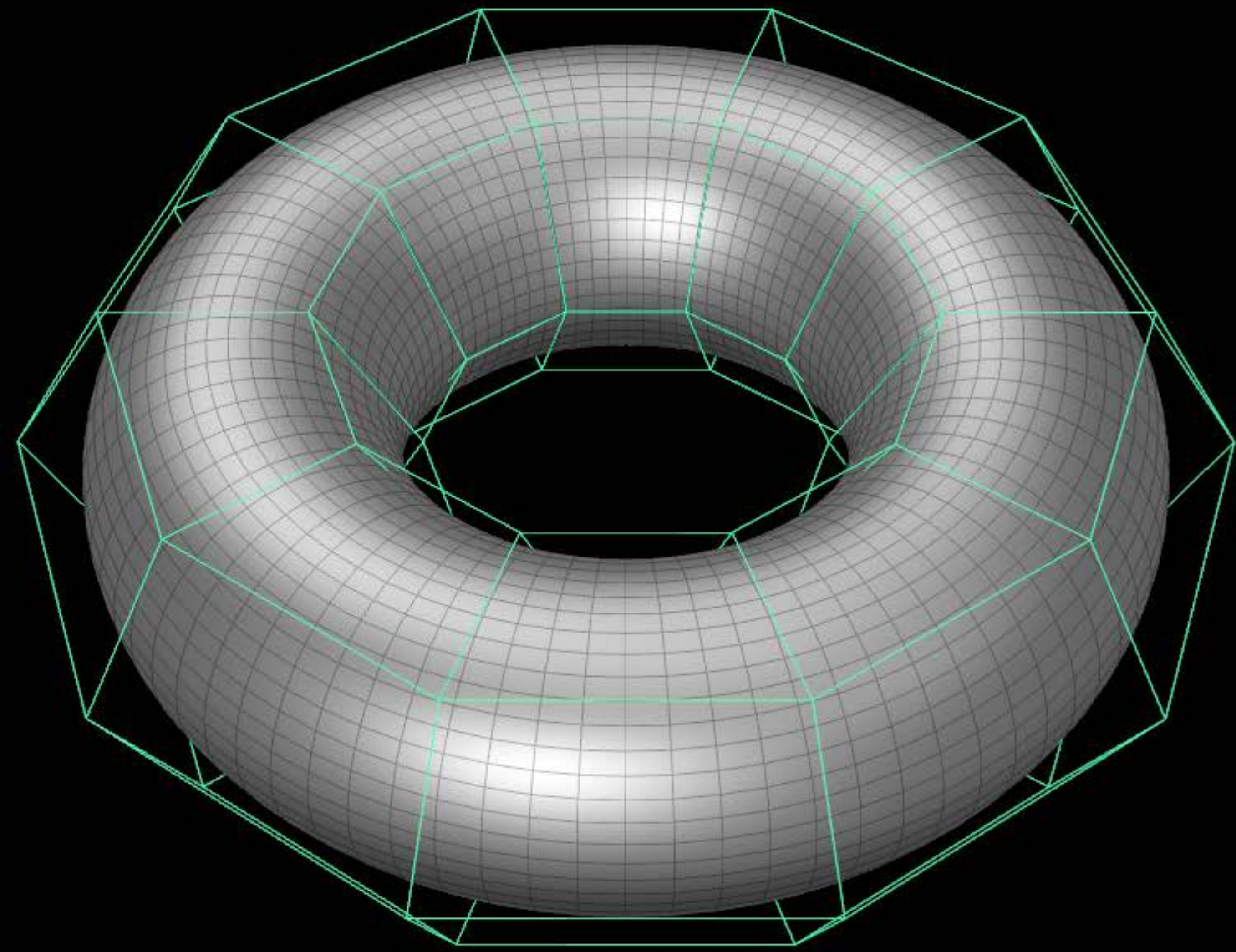
USD highlights



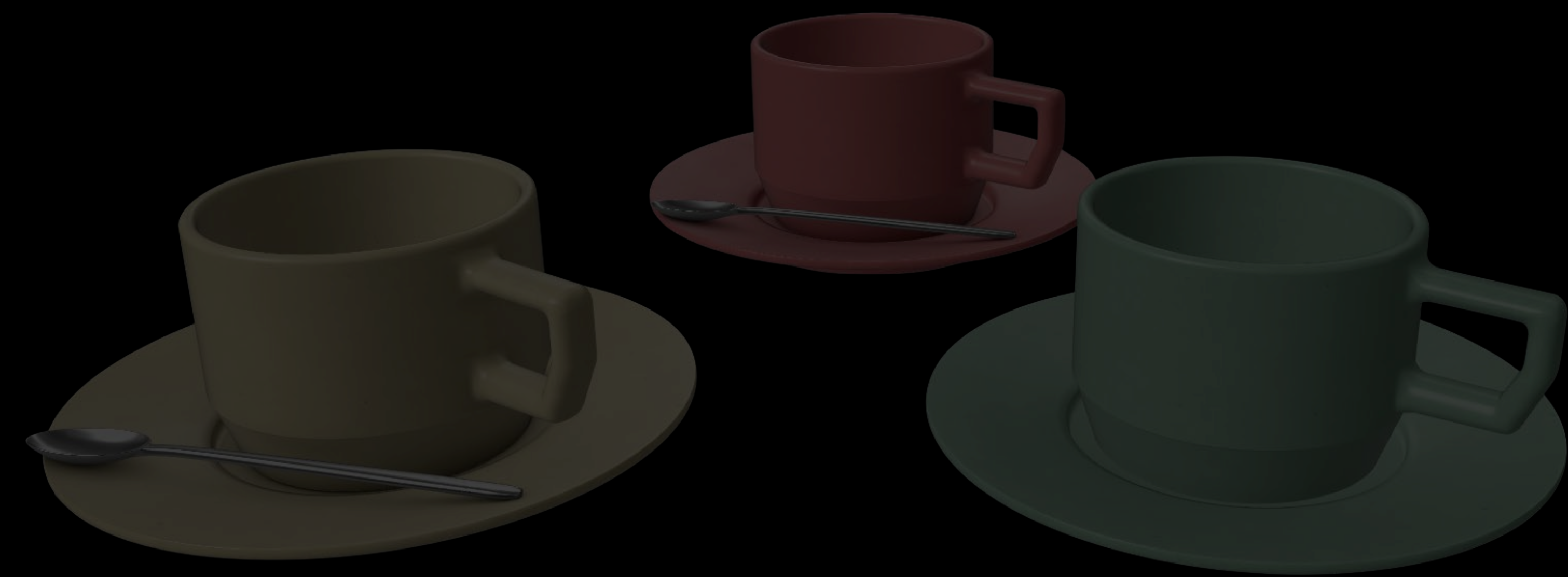
Subdivision Surfaces



Composition Engine



Subdivision Surfaces



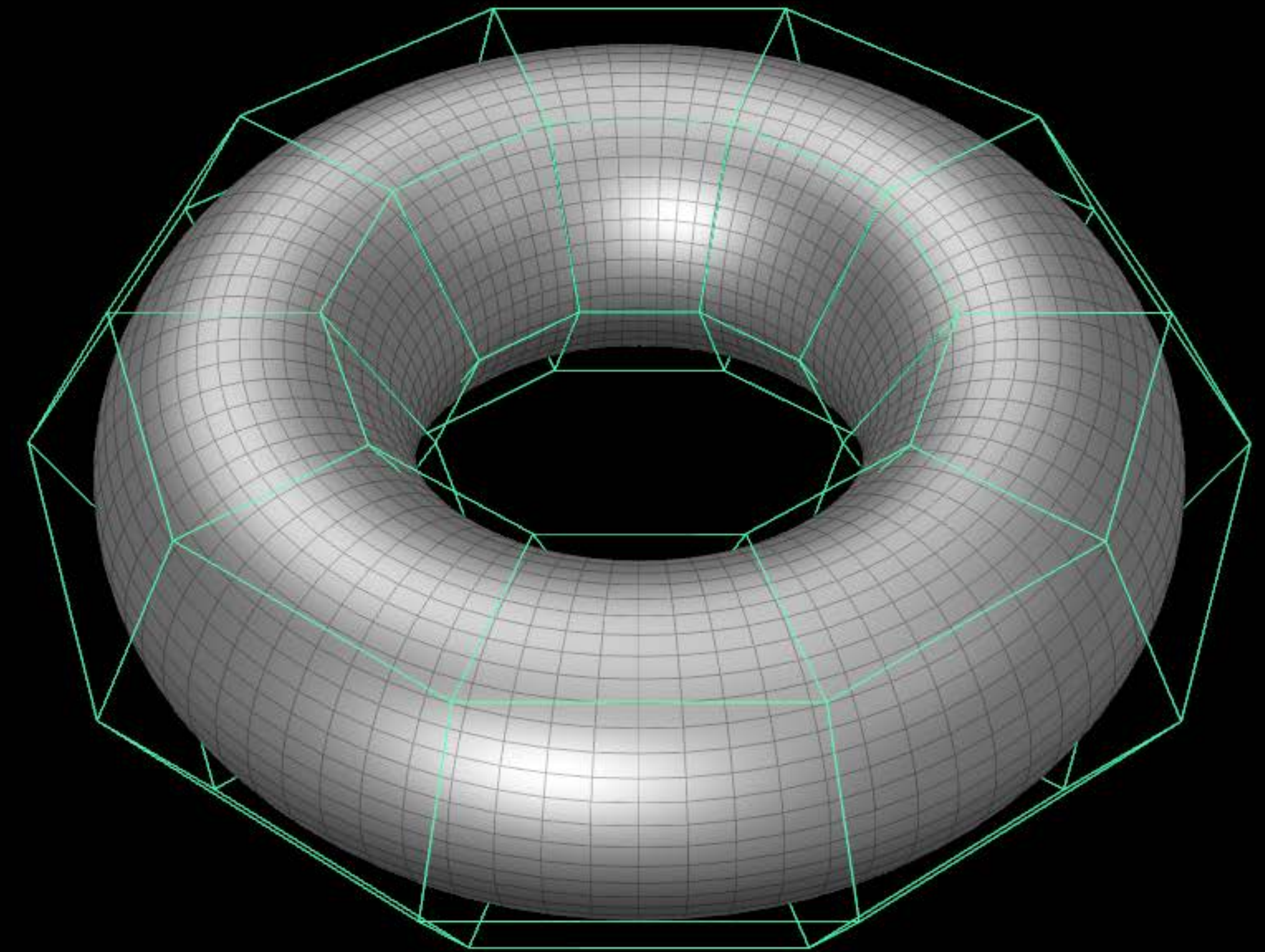
Composition Engine

Subdivision Surfaces

Efficient representation of curved surfaces

Dynamic tessellation
(e.g. based on distance to camera)

Great for animated surfaces





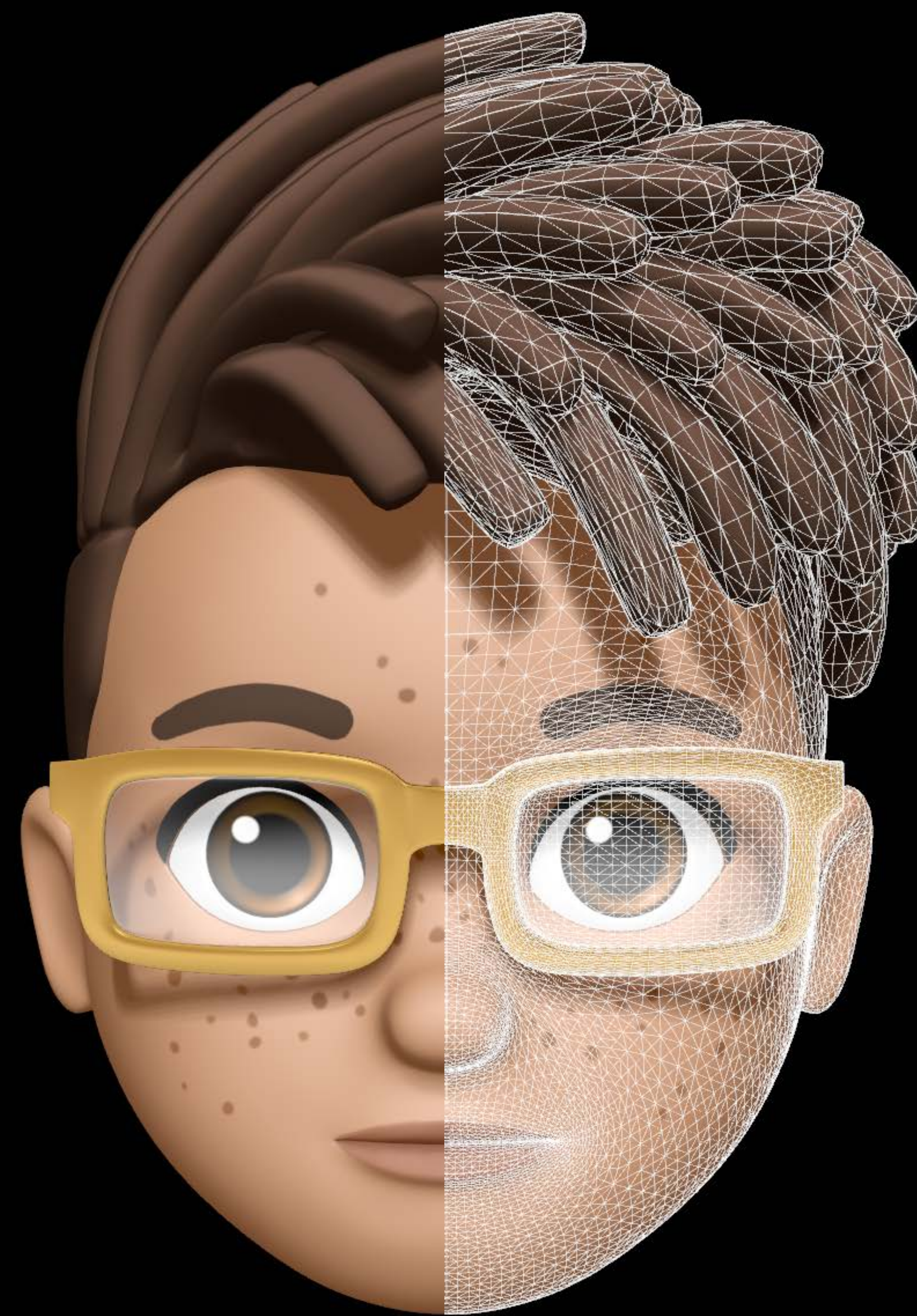
No Subdivision



Subdivision



No Subdivision



Subdivision

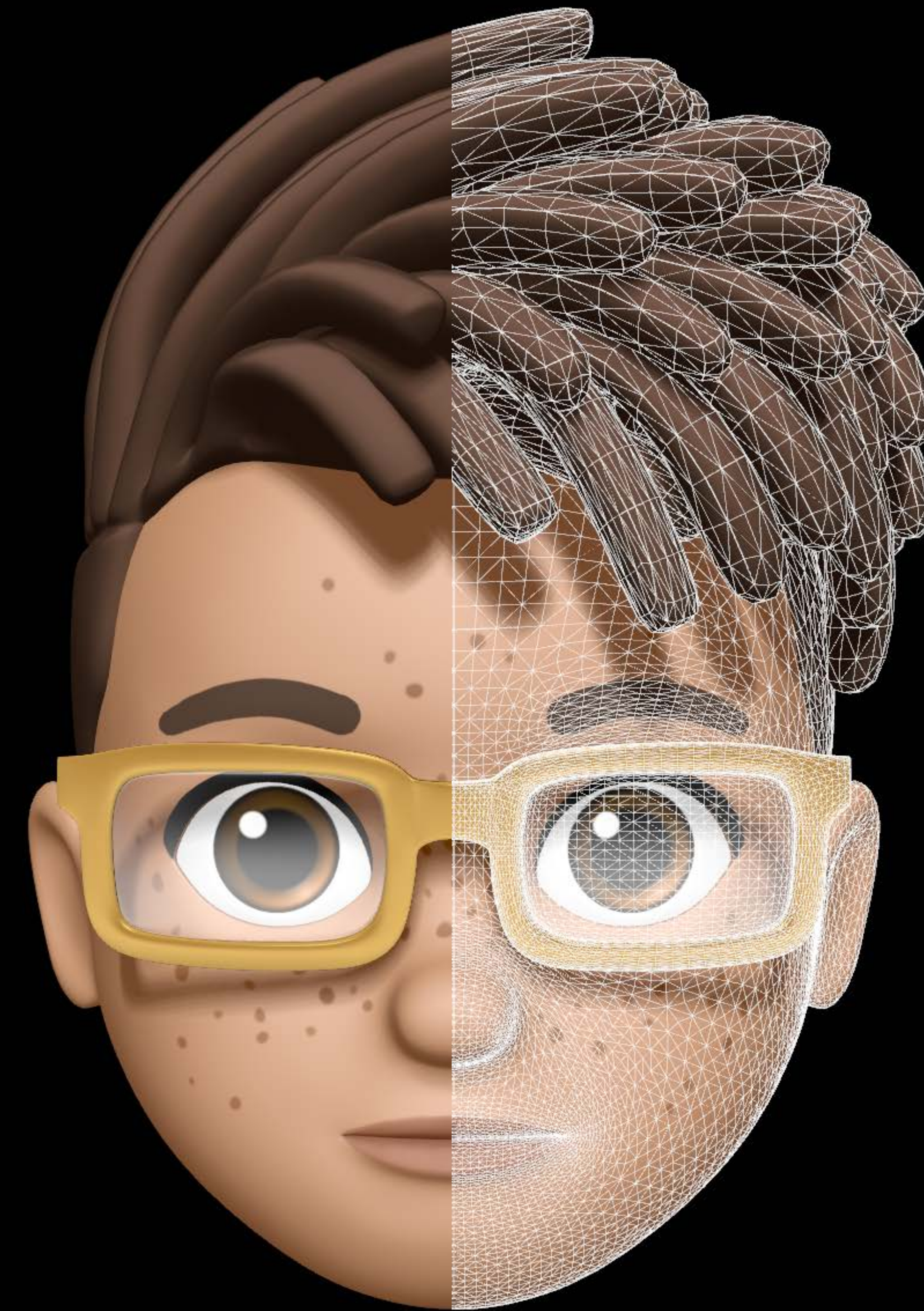
OpenSubdiv

Industry standard for subdivision surfaces

Developed by Pixar

Metal shaders for efficient GPU evaluation

Basis for subdivision surfaces in SceneKit

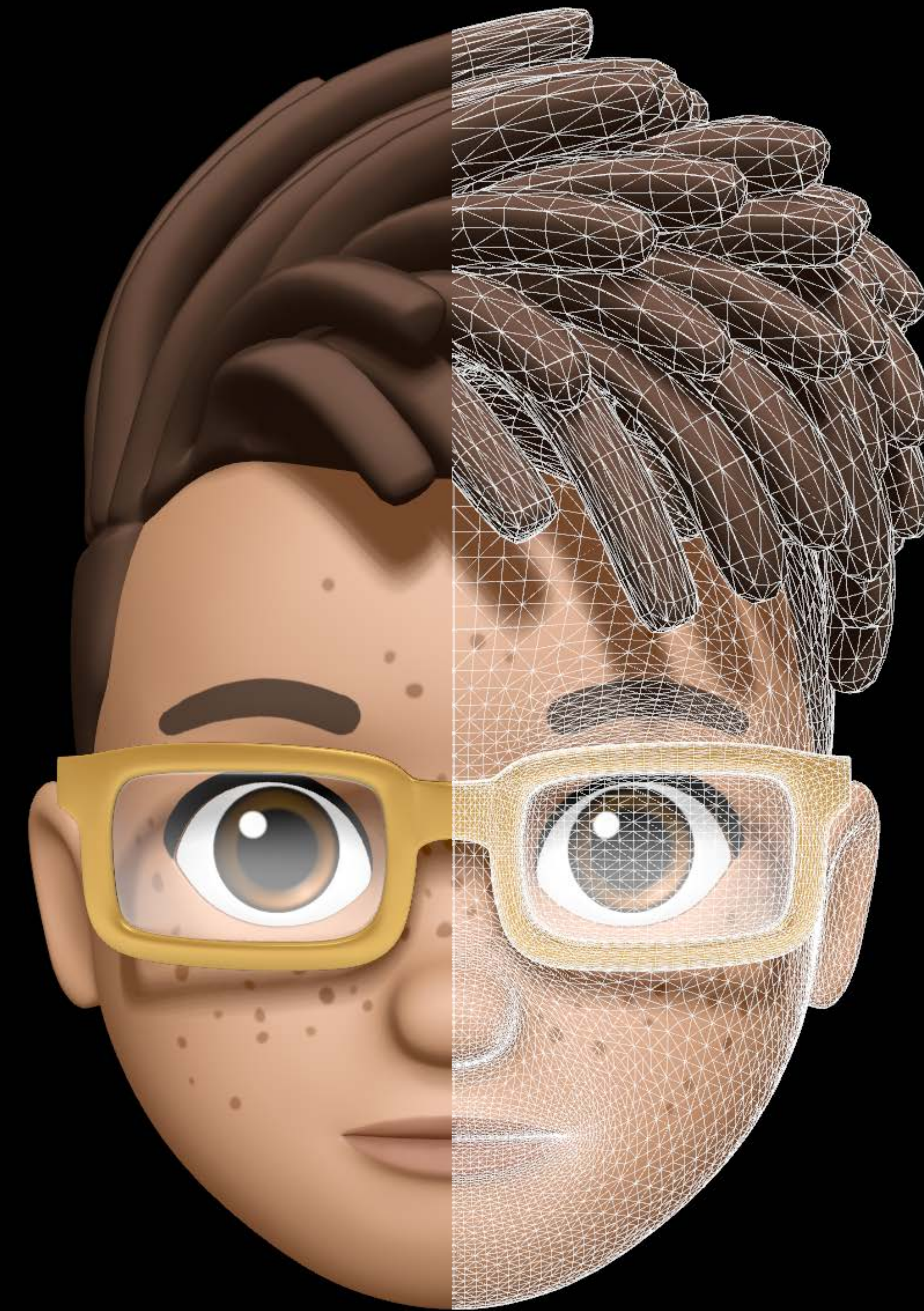


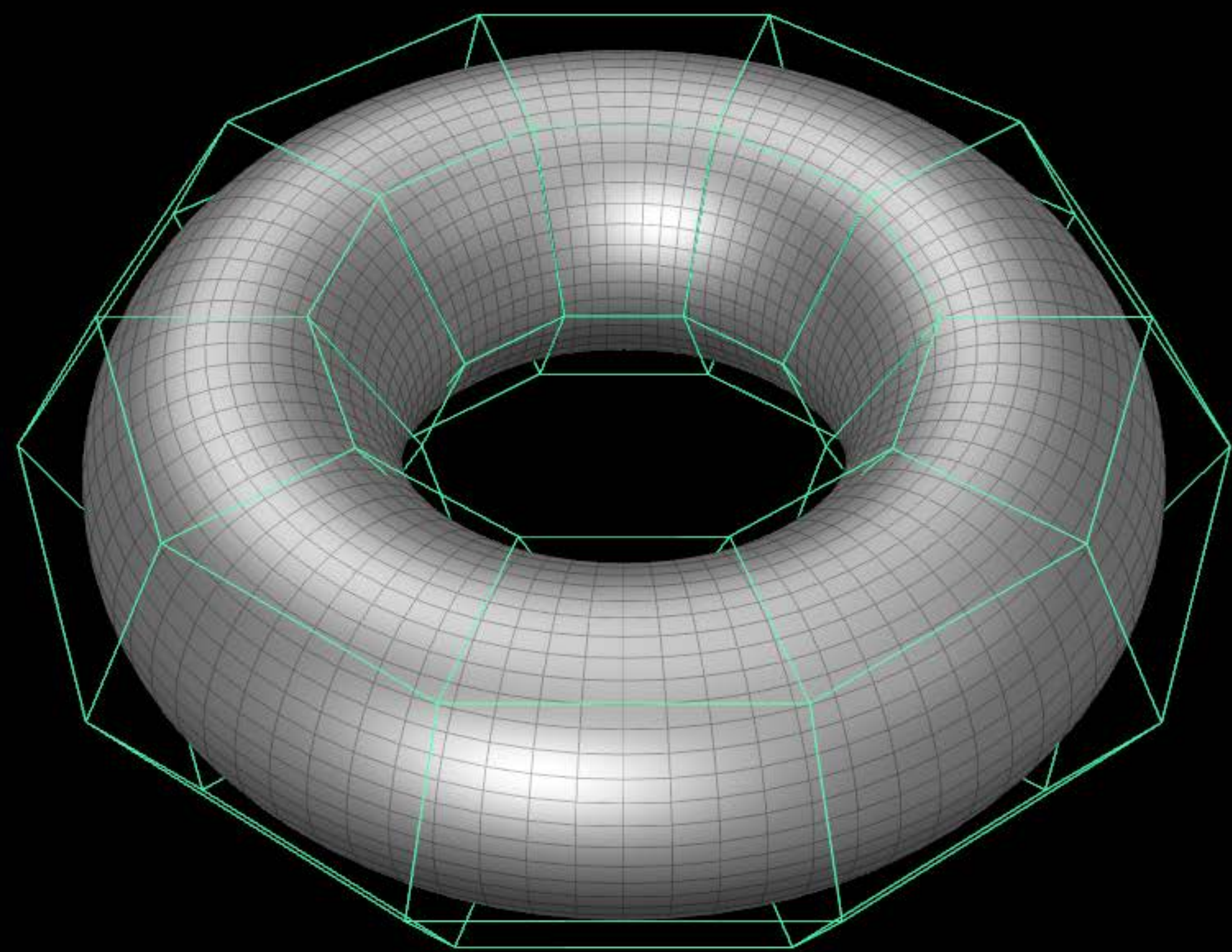
USD and OpenSubdiv

USD has unified description for

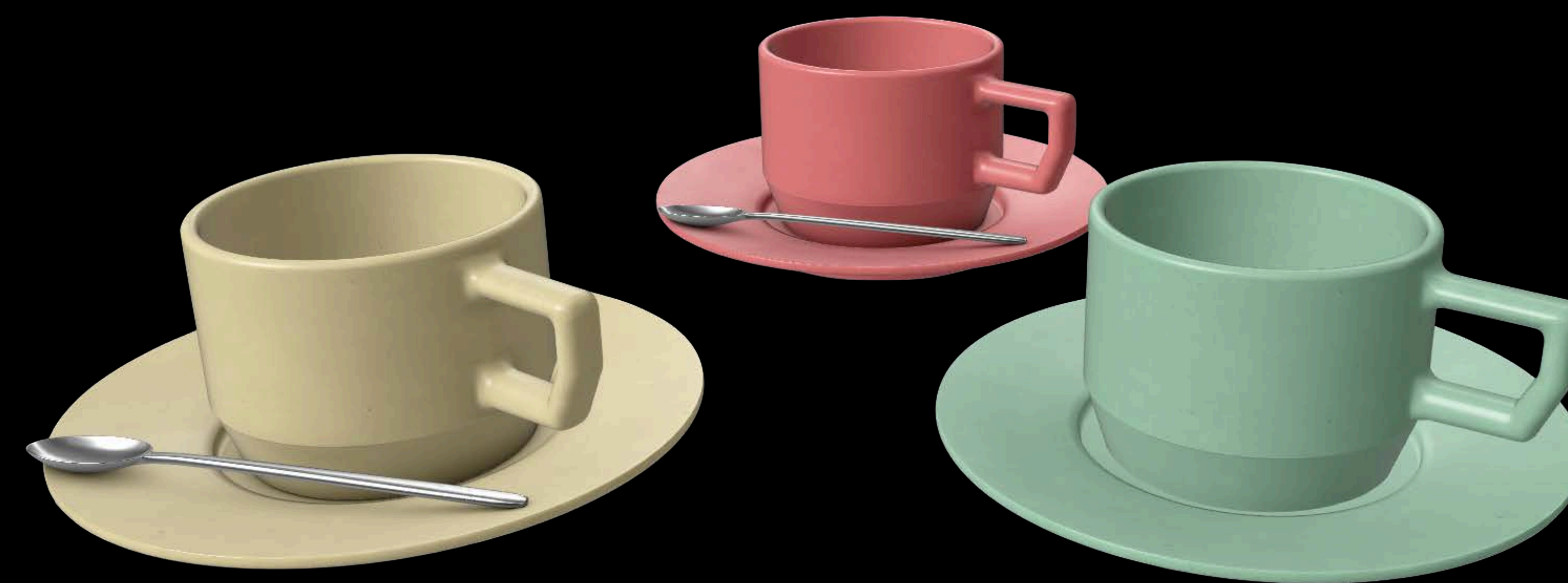
- Polygonal mesh data
- Subdivision surface data

Subdivision-specific properties

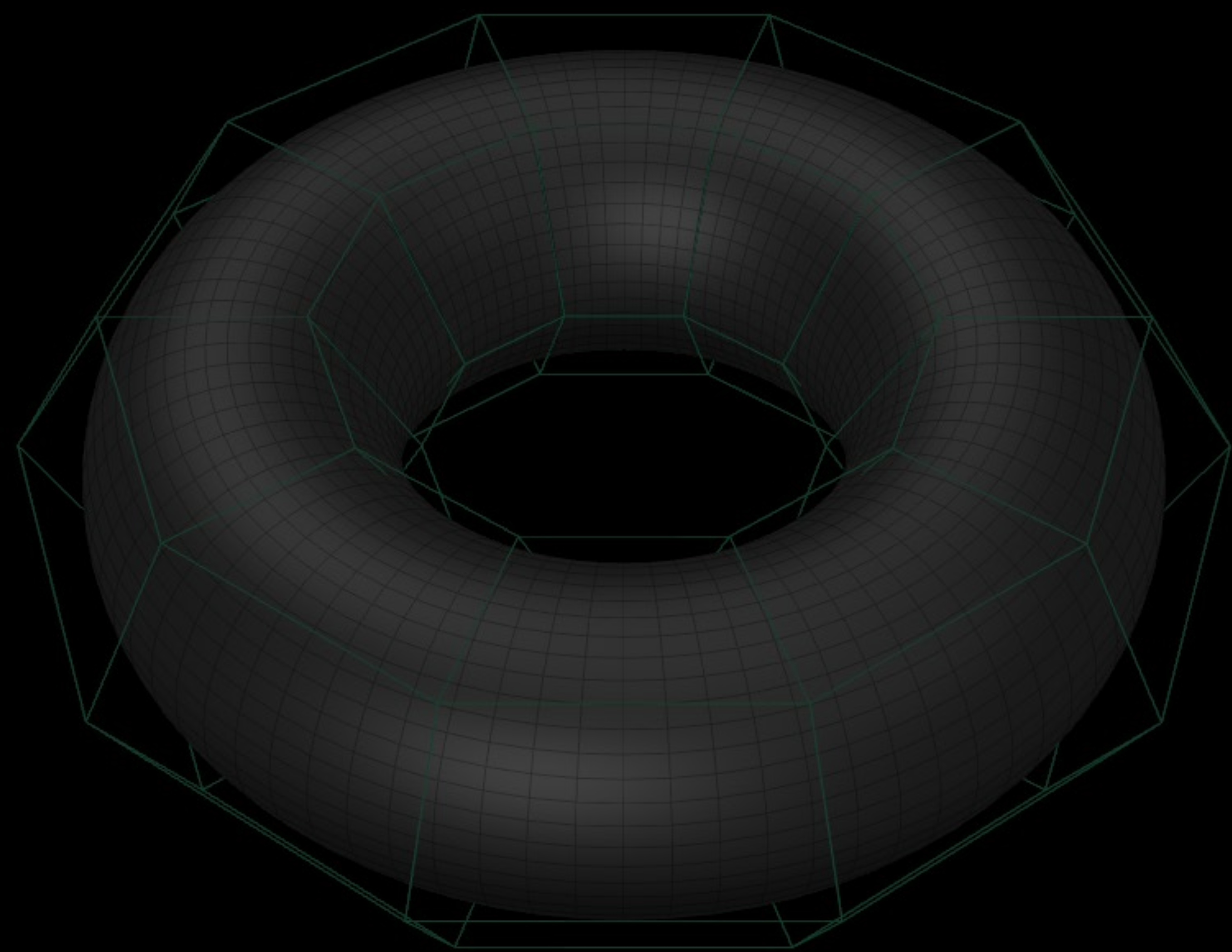




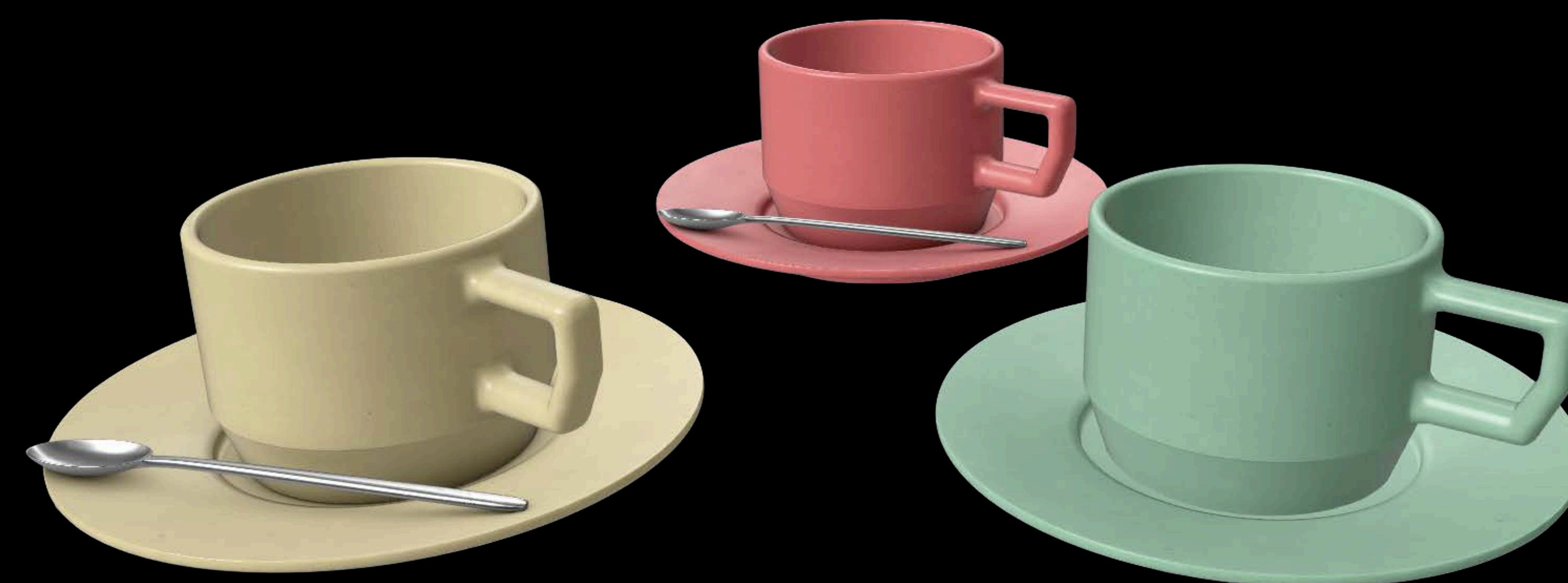
Subdivision Surfaces



Composition Engine



Subdivision Surfaces



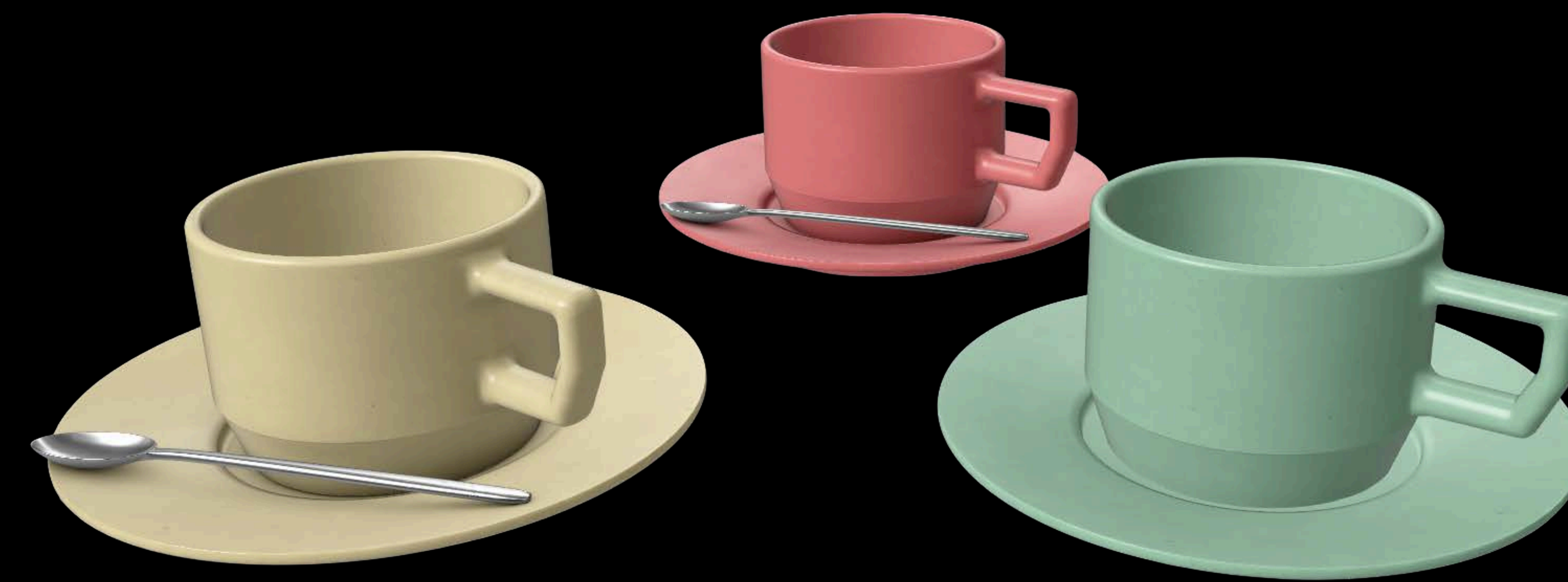
Composition Engine

Composition Engine

Powerful authoring tool

Enables efficient collaboration

Example: references

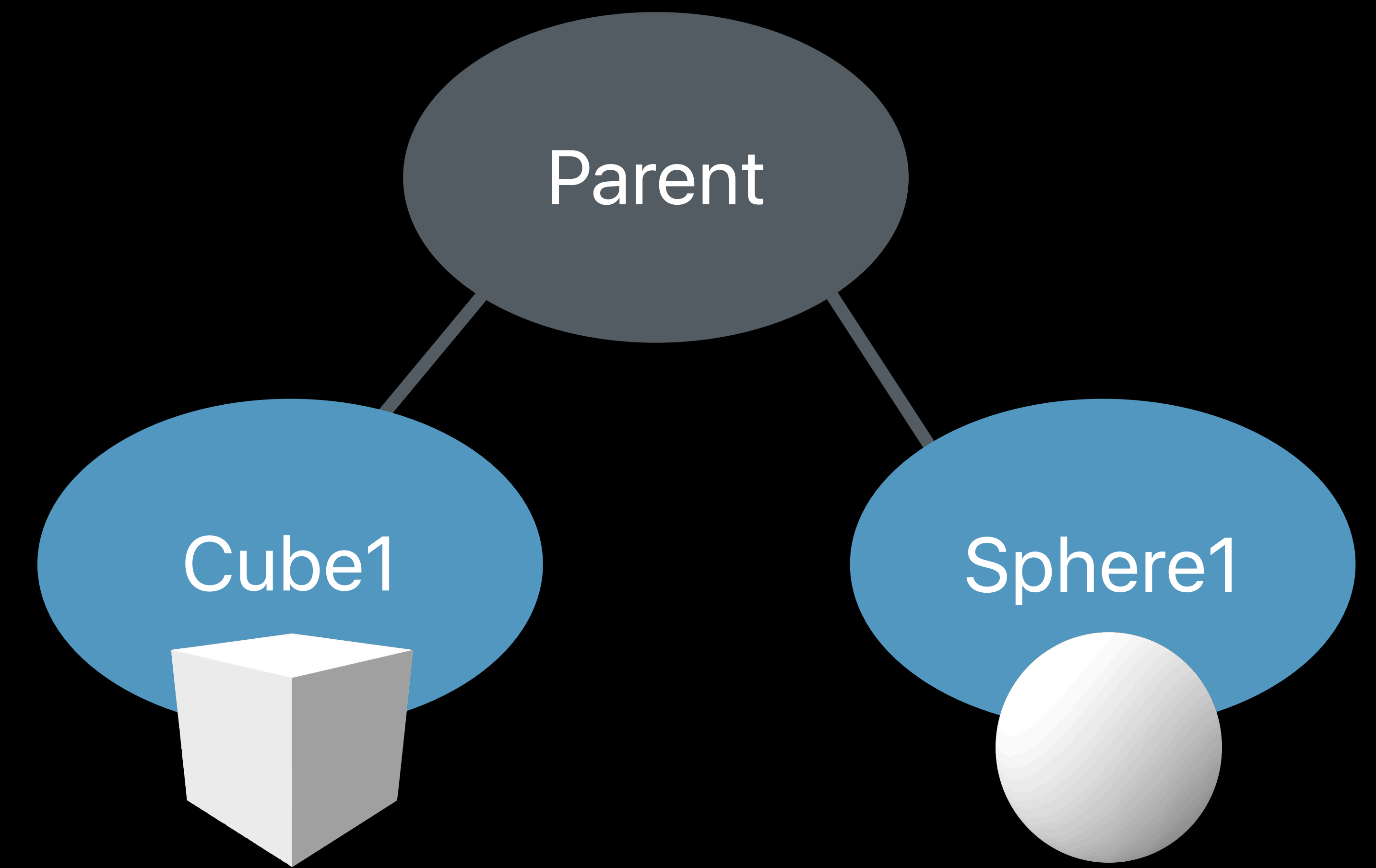


Scene Graph

Parent

Cube1

Sphere1



Duplicate Subgraphs

Parent

Child1

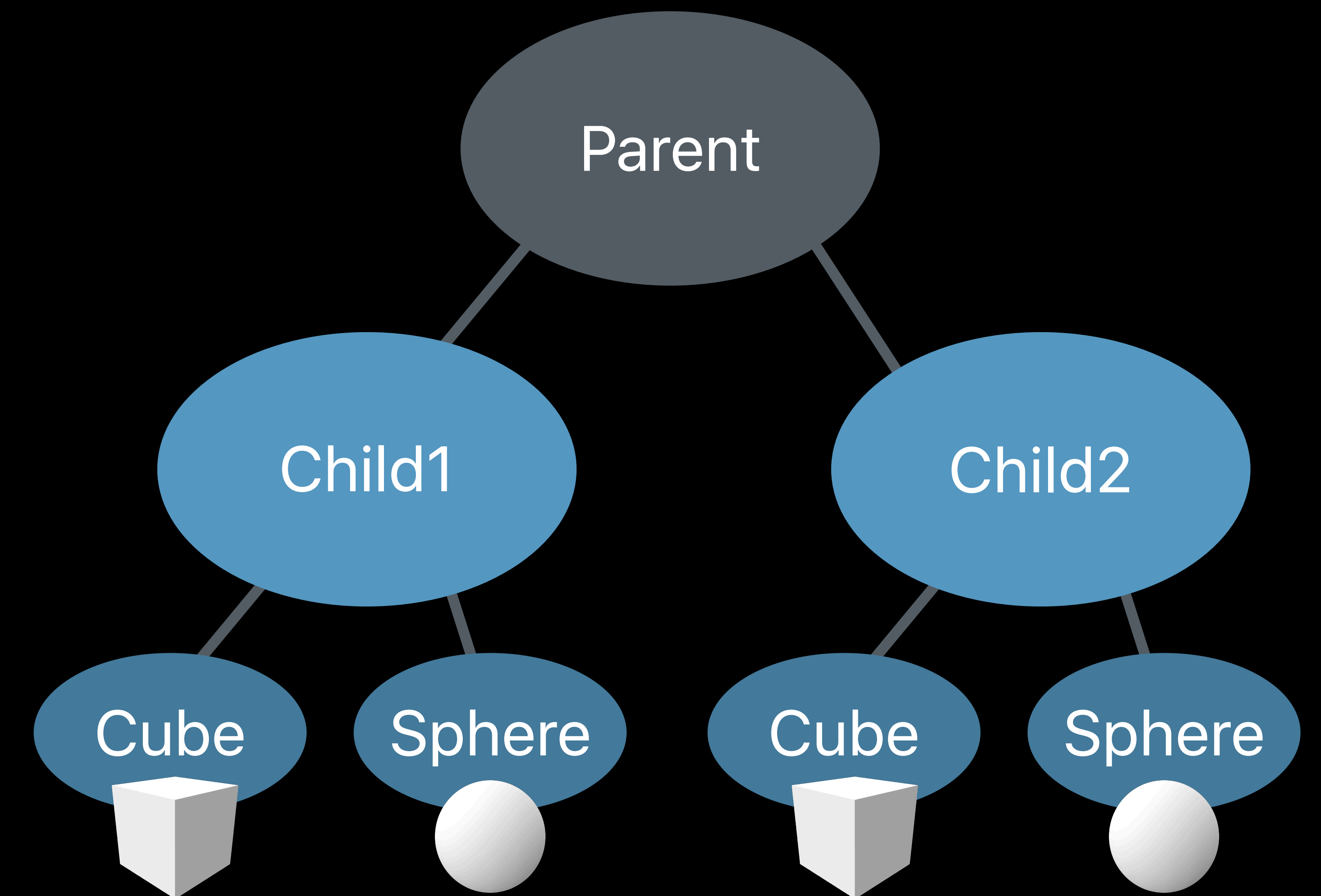
Cube

Sphere

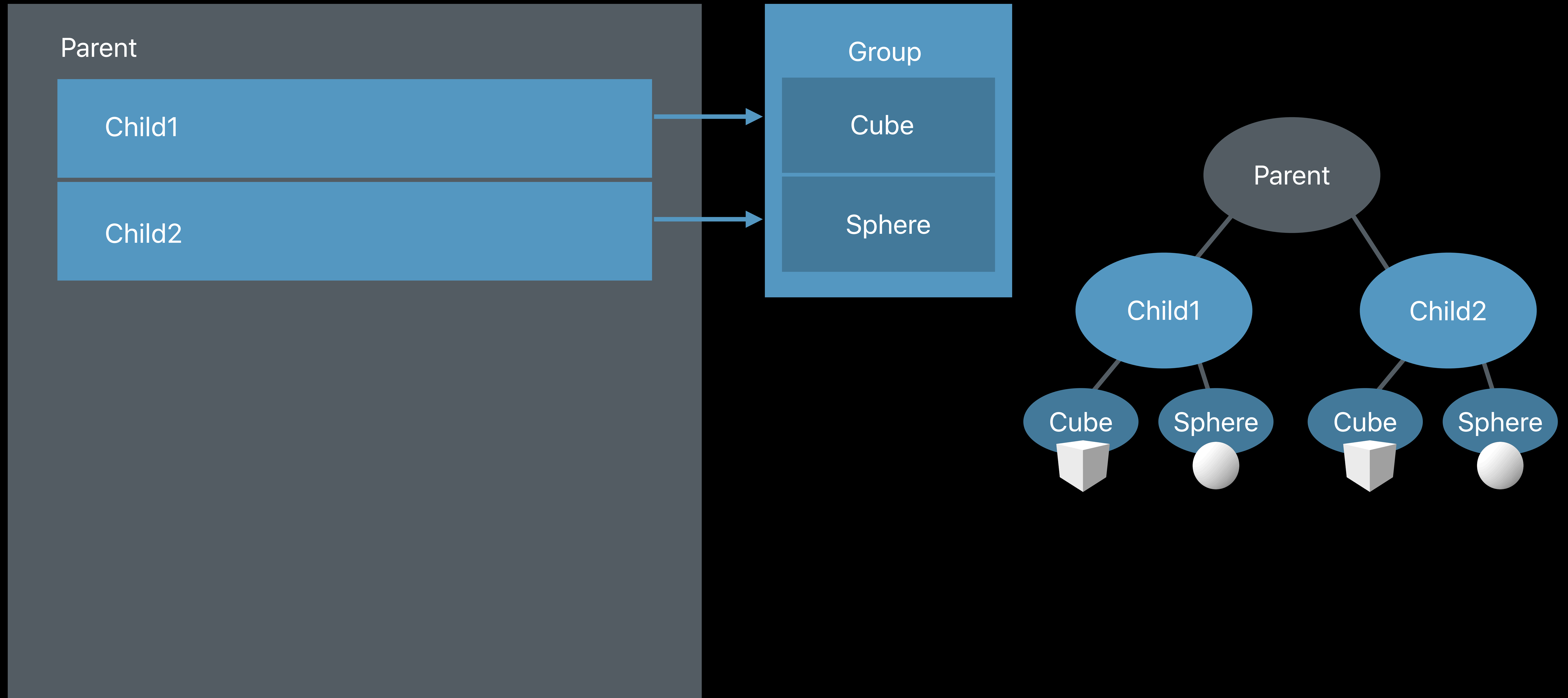
Child2

Cube

Sphere

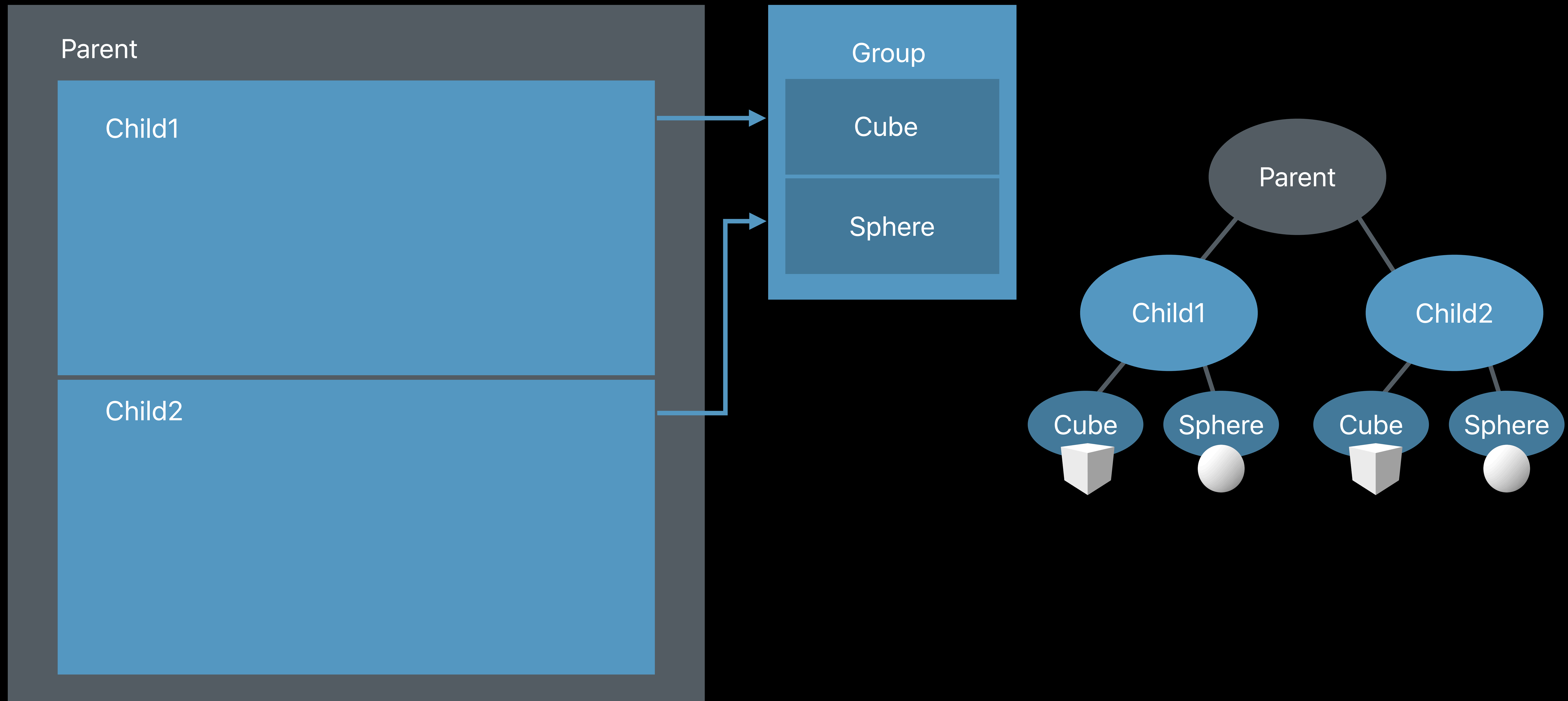


References



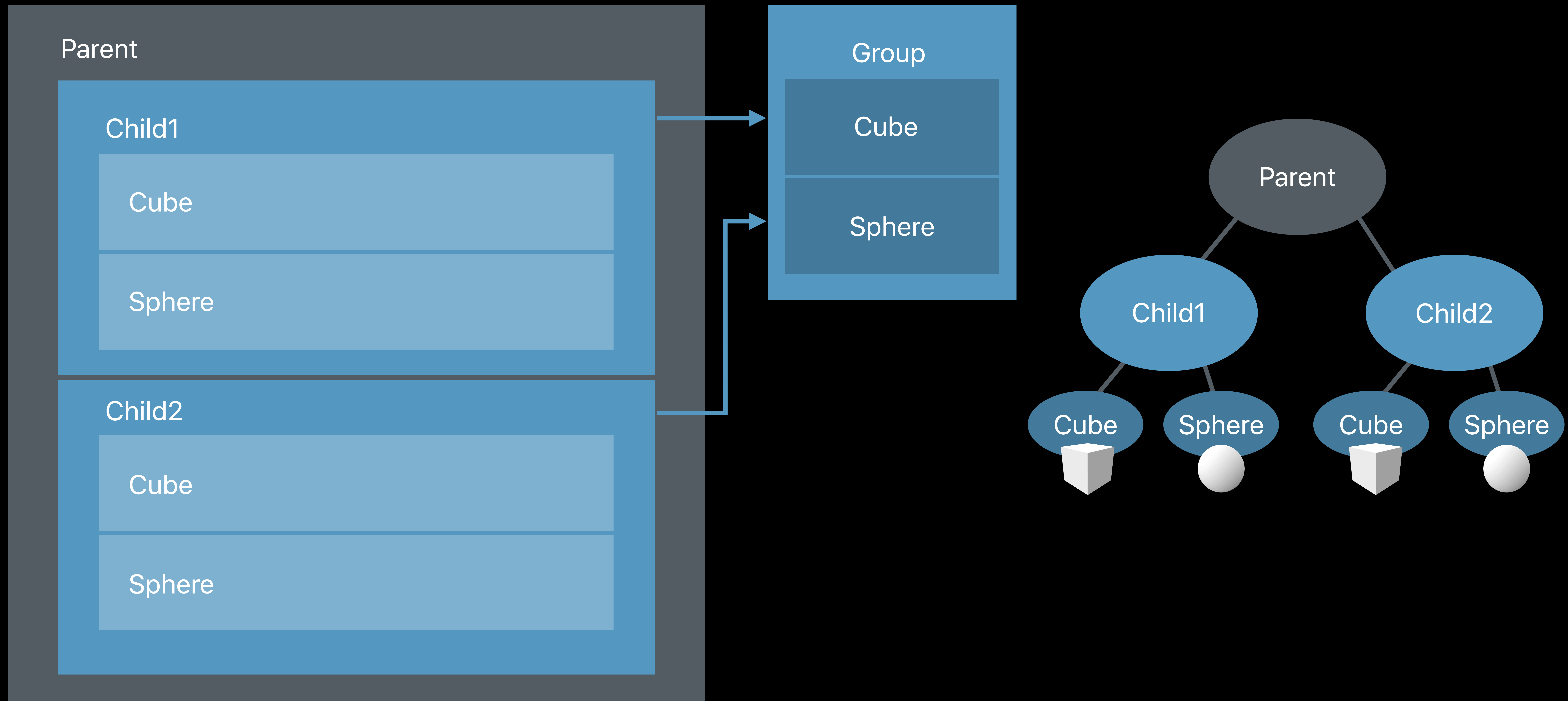
Composition Engine

References



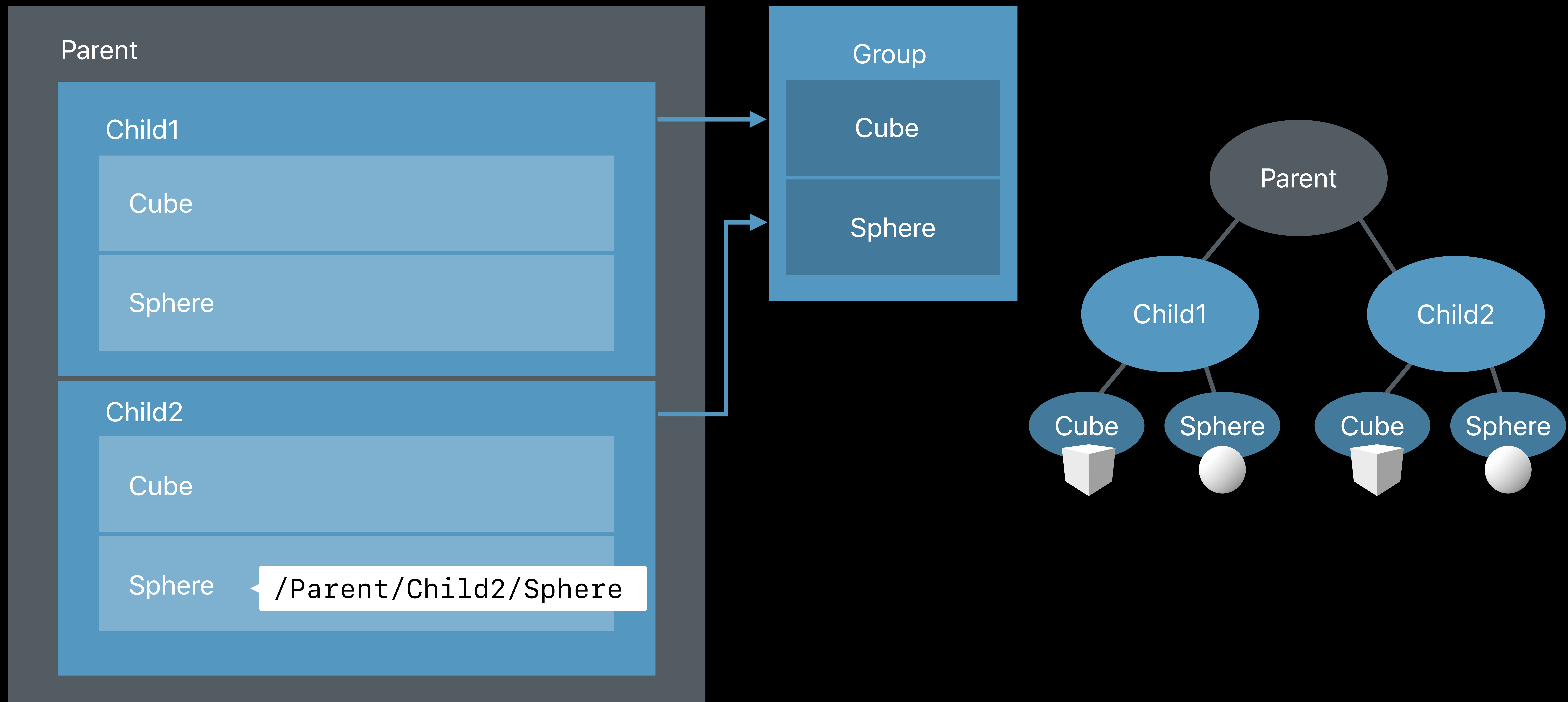
Composition Engine

References



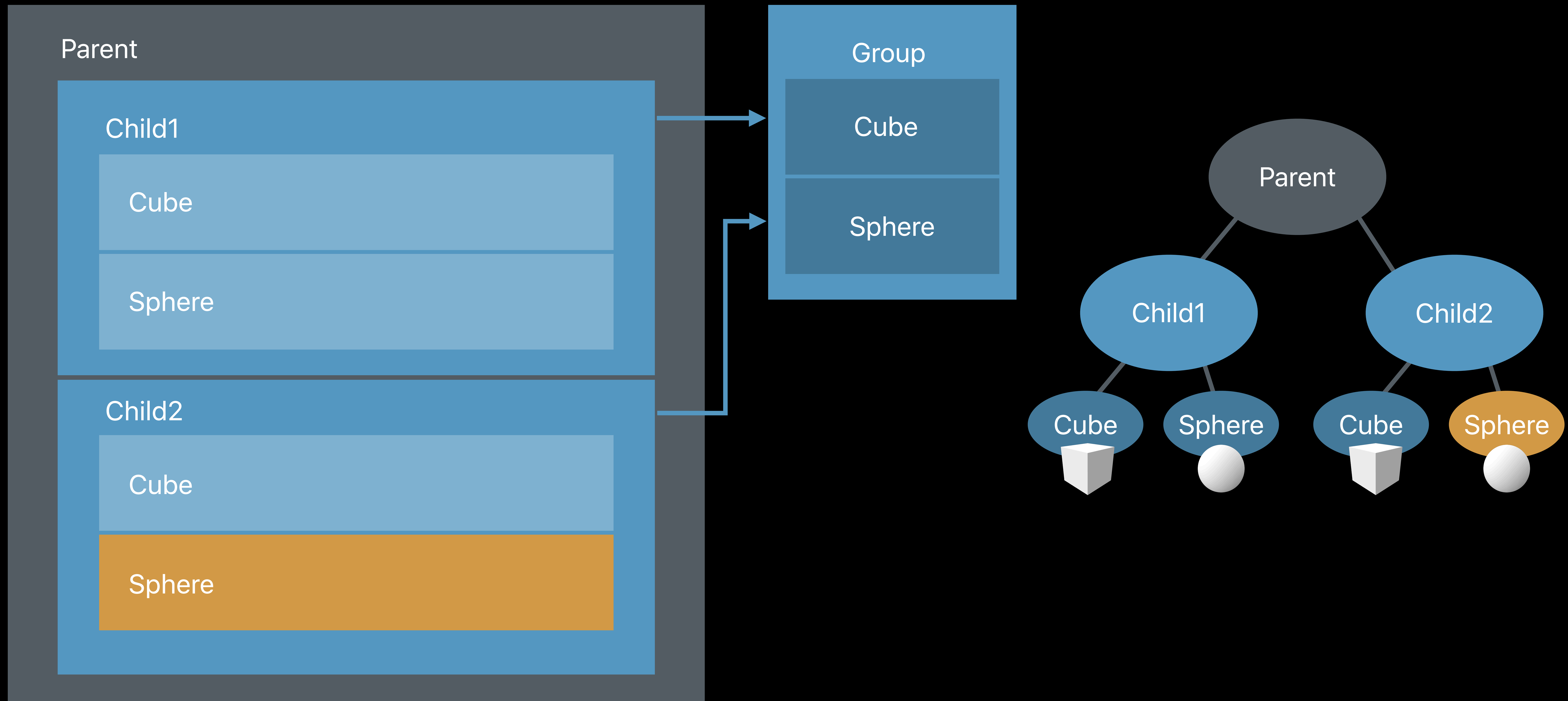
Composition Engine

References



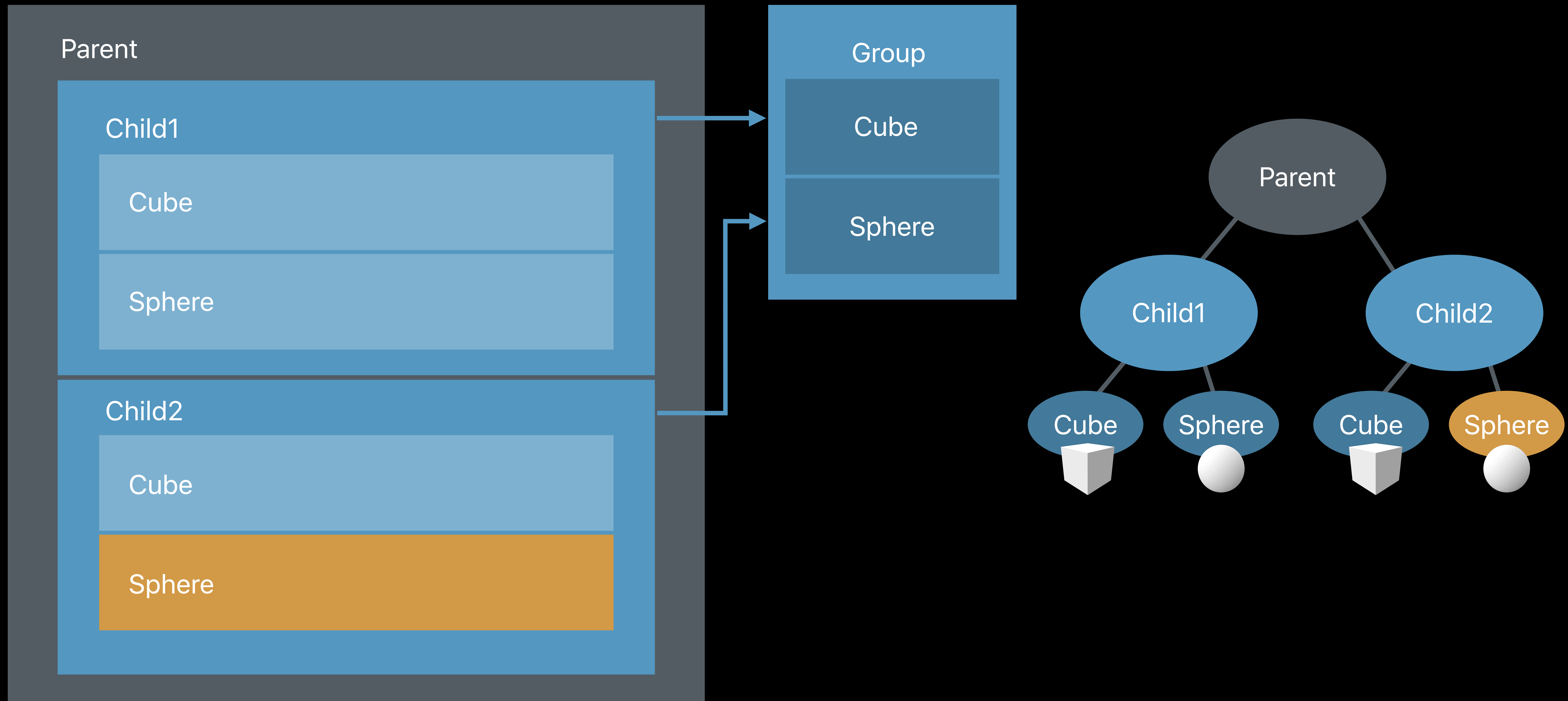
Composition Engine

Overrides



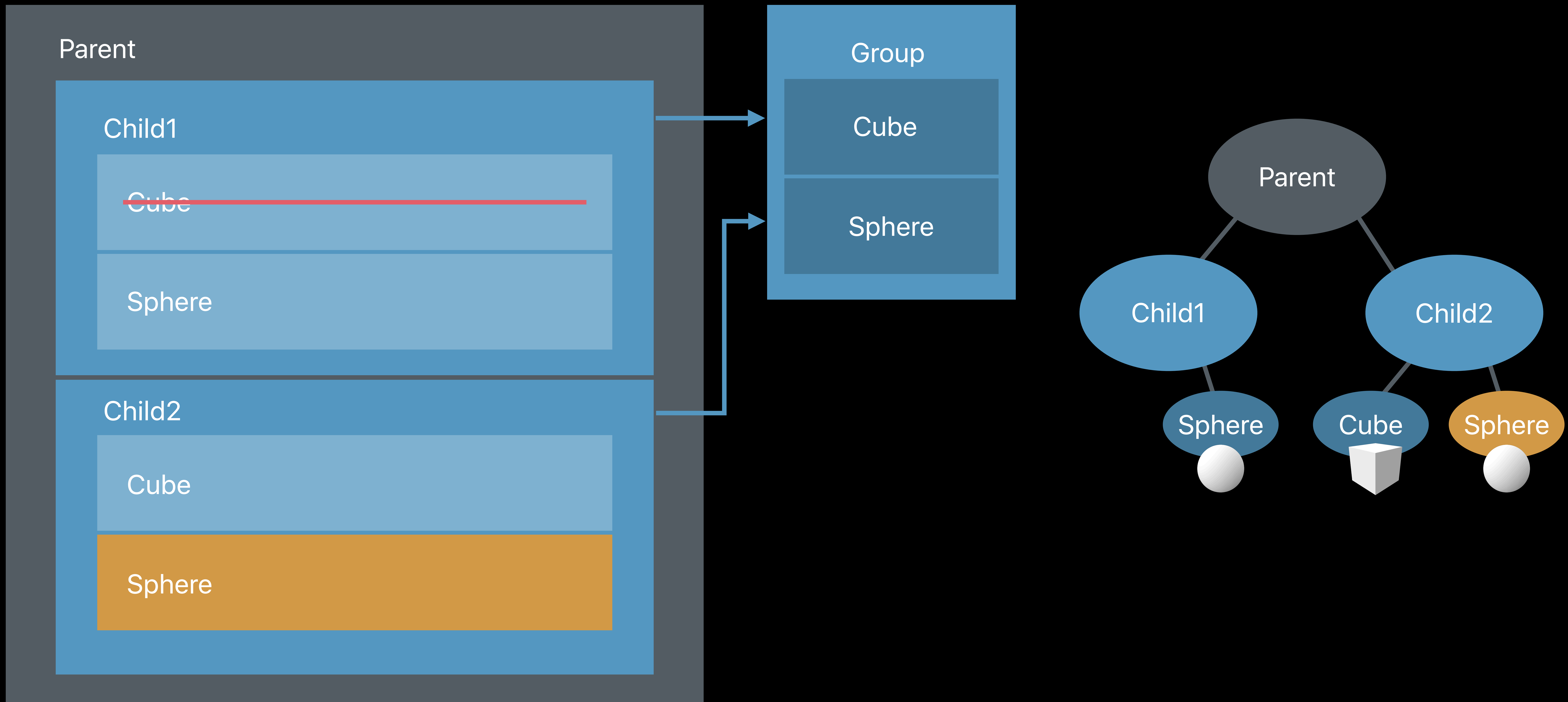
Composition Engine

Overrides



Composition Engine

Overrides





Kitchen_set

Name	Date Modified
01_Kitchen_set.usd	Jul 28, 2017 at 5:35 PM
assets	Today at 6:20 PM
Ball	Jul 28, 2017 at 5:35 PM
Book	Jul 28, 2017 at 5:35 PM
Bottle	Jul 28, 2017 at 5:35 PM
BottleB	Jul 28, 2017 at 5:35 PM
Bowl	Jul 28, 2017 at 5:35 PM
BreadBag	Jul 28, 2017 at 5:35 PM
Broom	Jul 28, 2017 at 5:35 PM
CastIron	Jul 28, 2017 at 5:35 PM
CeilingLight	Jul 28, 2017 at 5:35 PM
CerealBox	Jul 28, 2017 at 5:35 PM
Chair	Jul 28, 2017 at 5:35 PM
Chair_payload.usd	Jul 28, 2017 at 5:35 PM
Chair.geom.usd	Jul 28, 2017 at 5:35 PM
Chair.usd	Jul 28, 2017 at 5:35 PM
Cheerio	Jul 28, 2017 at 5:35 PM
Clock	Jul 28, 2017 at 5:35 PM
Crayon	Jul 28, 2017 at 5:35 PM
Cup	Jul 28, 2017 at 5:35 PM
DinnerMat	Jul 28, 2017 at 5:35 PM
DustPan	Jul 28, 2017 at 5:35 PM
FlowerPot	Jul 28, 2017 at 5:35 PM
FoldingTable	Jul 28, 2017 at 5:35 PM
Fork	Jul 28, 2017 at 5:35 PM
FramePicture	Jul 28, 2017 at 5:35 PM
FramePictureOval	Jul 28, 2017 at 5:35 PM
GarlicRope	Jul 28, 2017 at 5:35 PM
HandTowel	Jul 28, 2017 at 5:35 PM
Hanger	Jul 28, 2017 at 5:35 PM
Hook	Jul 28, 2017 at 5:35 PM
Iron	Jul 28, 2017 at 5:35 PM
IronBoard	Jul 28, 2017 at 5:35 PM
Jar	Jul 28, 2017 at 5:35 PM
Kitchen	Jul 28, 2017 at 5:35 PM
KitchenTable	Jul 28, 2017 at 5:35 PM
Knife	Jul 28, 2017 at 5:35 PM
MeasuringCup	Jul 28, 2017 at 5:35 PM
MeasuringSpoon	Jul 28, 2017 at 5:35 PM
Mixer	Jul 28, 2017 at 5:35 PM
Nail	Jul 28, 2017 at 5:35 PM
OilBottle	Jul 28, 2017 at 5:35 PM



Kitchen_set

Name	Date Modified
01_Kitchen_set.usd	Jul 28, 2017 at 5:35 PM
assets	Today at 6:14 PM
Ball	Jul 28, 2017 at 5:35 PM
Book	Jul 28, 2017 at 5:35 PM
Bottle	Jul 28, 2017 at 5:35 PM
BottleB	Jul 28, 2017 at 5:35 PM
Bowl	Jul 28, 2017 at 5:35 PM
BreadBag	Jul 28, 2017 at 5:35 PM
Broom	Jul 28, 2017 at 5:35 PM
CastIron	Jul 28, 2017 at 5:35 PM
CeilingLight	Jul 28, 2017 at 5:35 PM
CerealBox	Jul 28, 2017 at 5:35 PM
Chair	Jul 28, 2017 at 5:35 PM
Chair_payload.usd	Jul 28, 2017 at 5:35 PM
Chair.geom.usd	Jul 28, 2017 at 5:35 PM
Chair.usd	Jul 28, 2017 at 5:35 PM
Cheerio	Jul 28, 2017 at 5:35 PM
Clock	Jul 28, 2017 at 5:35 PM
Crayon	Jul 28, 2017 at 5:35 PM
Cup	Jul 28, 2017 at 5:35 PM
DinnerMat	Jul 28, 2017 at 5:35 PM
DustPan	Jul 28, 2017 at 5:35 PM
FlowerPot	Jul 28, 2017 at 5:35 PM
FoldingTable	Jul 28, 2017 at 5:35 PM
Fork	Jul 28, 2017 at 5:35 PM
FramePicture	Jul 28, 2017 at 5:35 PM
FramePictureOval	Jul 28, 2017 at 5:35 PM
GarlicRope	Jul 28, 2017 at 5:35 PM
HandTowel	Jul 28, 2017 at 5:35 PM
Hanger	Jul 28, 2017 at 5:35 PM
Hook	Jul 28, 2017 at 5:35 PM
Iron	Jul 28, 2017 at 5:35 PM
IronBoard	Jul 28, 2017 at 5:35 PM
Jar	Jul 28, 2017 at 5:35 PM
Kitchen	Jul 28, 2017 at 5:35 PM
KitchenTable	Jul 28, 2017 at 5:35 PM
Knife	Jul 28, 2017 at 5:35 PM
MeasuringCup	Jul 28, 2017 at 5:35 PM
MeasuringSpoon	Jul 28, 2017 at 5:35 PM
Mixer	Jul 28, 2017 at 5:35 PM
Nail	Jul 28, 2017 at 5:35 PM
OilBottle	Jul 28, 2017 at 5:35 PM



Kitchen_set

Name	Date Modified
01_Kitchen_set.usd	Jul 28, 2017 at 5:35 PM
assets	Today at 6:17 PM
Ball	Jul 28, 2017 at 5:35 PM
Book	Jul 28, 2017 at 5:35 PM
Bottle	Jul 28, 2017 at 5:35 PM
BottleB	Jul 28, 2017 at 5:35 PM
Bowl	Jul 28, 2017 at 5:35 PM
BreadBag	Jul 28, 2017 at 5:35 PM
Broom	Jul 28, 2017 at 5:35 PM
CastIron	Jul 28, 2017 at 5:35 PM
CeilingLight	Jul 28, 2017 at 5:35 PM
CerealBox	Jul 28, 2017 at 5:35 PM
Chair	Today at 5:21 PM
Chair_payload.usd	Today at 5:21 PM
Chair.geom.usd	Today at 5:21 PM
Chair.usd	Today at 5:21 PM
textures	Today at 5:21 PM
Cheerio	Jul 28, 2017 at 5:35 PM
Clock	Jul 28, 2017 at 5:35 PM
Crayon	Jul 28, 2017 at 5:35 PM
Cup	Jul 28, 2017 at 5:35 PM
DinnerMat	Jul 28, 2017 at 5:35 PM
DustPan	Jul 28, 2017 at 5:35 PM
FlowerPot	Jul 28, 2017 at 5:35 PM
FoldingTable	Jul 28, 2017 at 5:35 PM
Fork	Jul 28, 2017 at 5:35 PM
FramePicture	Jul 28, 2017 at 5:35 PM
FramePictureOval	Jul 28, 2017 at 5:35 PM
GarlicRope	Jul 28, 2017 at 5:35 PM
HandTowel	Jul 28, 2017 at 5:35 PM
Hanger	Jul 28, 2017 at 5:35 PM
Hook	Jul 28, 2017 at 5:35 PM
Iron	Jul 28, 2017 at 5:35 PM
IronBoard	Jul 28, 2017 at 5:35 PM
Jar	Jul 28, 2017 at 5:35 PM
Kitchen	Jul 28, 2017 at 5:35 PM
KitchenTable	Jul 28, 2017 at 5:35 PM
Knife	Jul 28, 2017 at 5:35 PM
MeasuringCup	Jul 28, 2017 at 5:35 PM
MeasuringSpoon	Jul 28, 2017 at 5:35 PM
Mixer	Jul 28, 2017 at 5:35 PM
Nail	Jul 28, 2017 at 5:35 PM



Kitchen_set

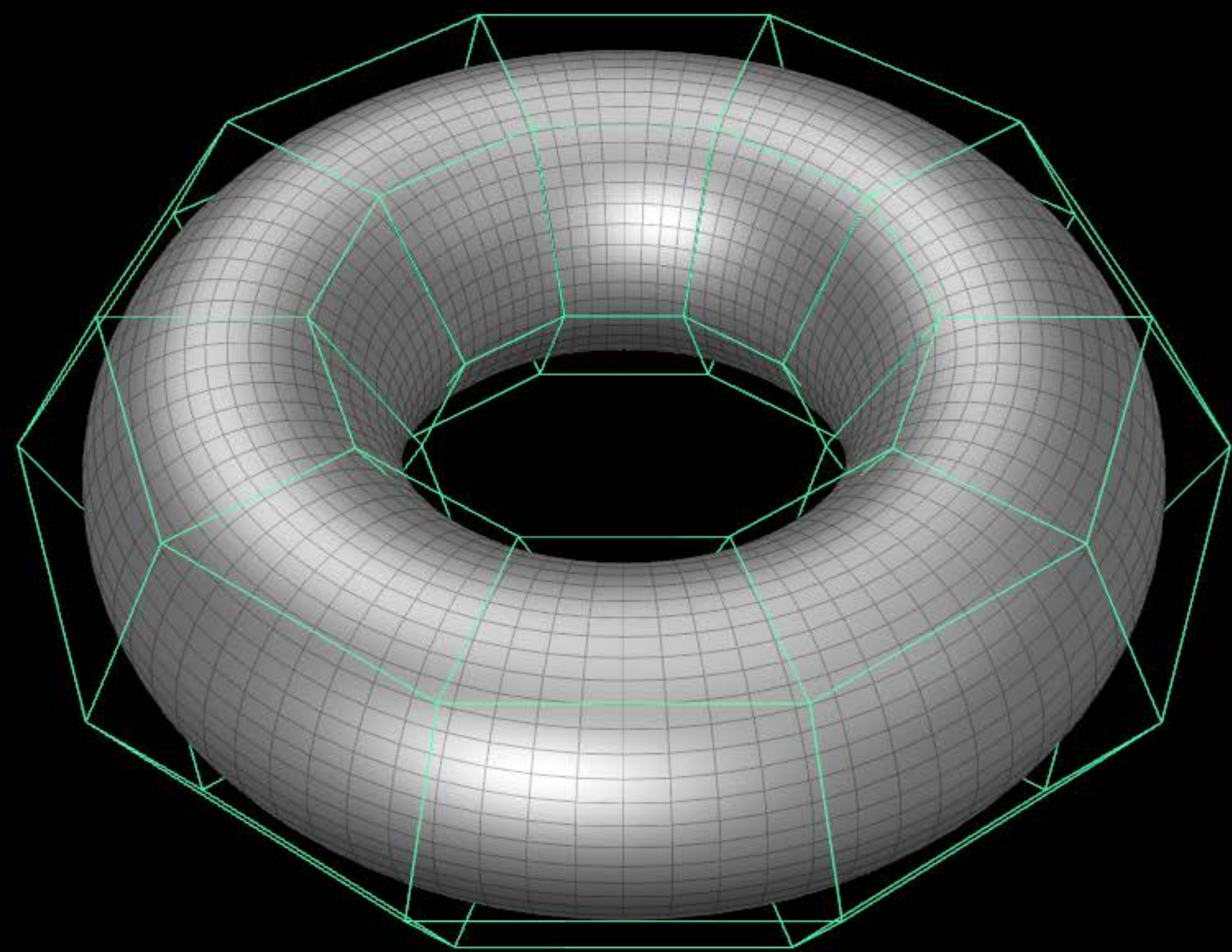
Name	Date Modified
01_Kitchen_set.usd	Jul 28, 2017 at 5:35 PM
assets	Today at 6:17 PM
Ball	Jul 28, 2017 at 5:35 PM
Book	Jul 28, 2017 at 5:35 PM
Bottle	Jul 28, 2017 at 5:35 PM
BottleB	Jul 28, 2017 at 5:35 PM
Bowl	Jul 28, 2017 at 5:35 PM
BreadBag	Jul 28, 2017 at 5:35 PM
Broom	Jul 28, 2017 at 5:35 PM
CastIron	Jul 28, 2017 at 5:35 PM
CeilingLight	Jul 28, 2017 at 5:35 PM
CerealBox	Jul 28, 2017 at 5:35 PM
Chair	Today at 5:21 PM
Chair_payload.usd	Today at 5:21 PM
Chair.geom.usd	Today at 5:21 PM
Chair.usd	Today at 5:21 PM
textures	Today at 5:21 PM
Cheerio	Jul 28, 2017 at 5:35 PM
Clock	Jul 28, 2017 at 5:35 PM
Crayon	Jul 28, 2017 at 5:35 PM
Cup	Jul 28, 2017 at 5:35 PM
DinnerMat	Jul 28, 2017 at 5:35 PM
DustPan	Jul 28, 2017 at 5:35 PM
FlowerPot	Jul 28, 2017 at 5:35 PM
FoldingTable	Jul 28, 2017 at 5:35 PM
Fork	Jul 28, 2017 at 5:35 PM
FramePicture	Jul 28, 2017 at 5:35 PM
FramePictureOval	Jul 28, 2017 at 5:35 PM
GarlicRope	Jul 28, 2017 at 5:35 PM
HandTowel	Jul 28, 2017 at 5:35 PM
Hanger	Jul 28, 2017 at 5:35 PM
Hook	Jul 28, 2017 at 5:35 PM
Iron	Jul 28, 2017 at 5:35 PM
IronBoard	Jul 28, 2017 at 5:35 PM
Jar	Jul 28, 2017 at 5:35 PM
Kitchen	Jul 28, 2017 at 5:35 PM
KitchenTable	Jul 28, 2017 at 5:35 PM
Knife	Jul 28, 2017 at 5:35 PM
MeasuringCup	Jul 28, 2017 at 5:35 PM
MeasuringSpoon	Jul 28, 2017 at 5:35 PM
Mixer	Jul 28, 2017 at 5:35 PM
Nail	Jul 28, 2017 at 5:35 PM

01_Kitchen_set.overrides.usd Open with Xcode-beta

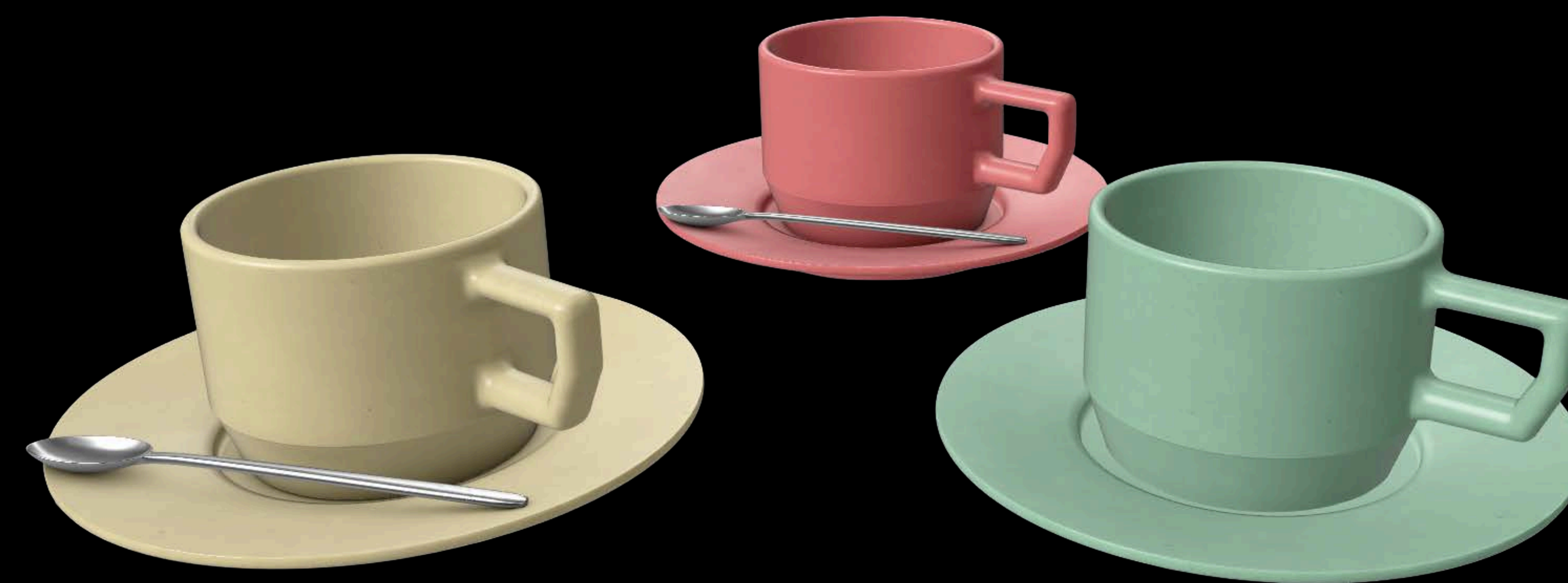


Kitchen_set

Name	Date Modified
01_Kitchen_set.overrides.usd	Today at 5:24 PM
01_Kitchen_set.usd	Jul 28, 2017 at 5:35 PM
assets	Today at 6:21 PM
Ball	Jul 28, 2017 at 5:35 PM
Book	Jul 28, 2017 at 5:35 PM
Bottle	Jul 28, 2017 at 5:35 PM
BottleB	Jul 28, 2017 at 5:35 PM
Bowl	Jul 28, 2017 at 5:35 PM
BreadBag	Jul 28, 2017 at 5:35 PM
Broom	Jul 28, 2017 at 5:35 PM
Castron	Jul 28, 2017 at 5:35 PM
CeilingLight	Jul 28, 2017 at 5:35 PM
CerealBox	Jul 28, 2017 at 5:35 PM
Chair	Today at 5:21 PM
Chair_payload.usd	Today at 5:21 PM
Chair.geom.usd	Today at 5:21 PM
Chair.usd	Today at 5:21 PM
textures	Today at 5:21 PM
Cheerio	Jul 28, 2017 at 5:35 PM
Clock	Jul 28, 2017 at 5:35 PM
Crayon	Jul 28, 2017 at 5:35 PM
Cup	Jul 28, 2017 at 5:35 PM
DinnerMat	Jul 28, 2017 at 5:35 PM
DustPan	Jul 28, 2017 at 5:35 PM
FlowerPot	Jul 28, 2017 at 5:35 PM
FoldingTable	Jul 28, 2017 at 5:35 PM
Fork	Jul 28, 2017 at 5:35 PM
FramePicture	Jul 28, 2017 at 5:35 PM
FramePictureOval	Jul 28, 2017 at 5:35 PM
GarlicRope	Jul 28, 2017 at 5:35 PM
HandTowel	Jul 28, 2017 at 5:35 PM
Hanger	Jul 28, 2017 at 5:35 PM
Hook	Jul 28, 2017 at 5:35 PM
Iron	Jul 28, 2017 at 5:35 PM
IronBoard	Jul 28, 2017 at 5:35 PM
Jar	Jul 28, 2017 at 5:35 PM
Kitchen	Jul 28, 2017 at 5:35 PM
KitchenTable	Jul 28, 2017 at 5:35 PM
Knife	Jul 28, 2017 at 5:35 PM
MeasuringCup	Jul 28, 2017 at 5:35 PM
MeasuringSpoon	Jul 28, 2017 at 5:35 PM
Mixer	Jul 28, 2017 at 5:35 PM



Subdivision Surfaces



Composition Engine

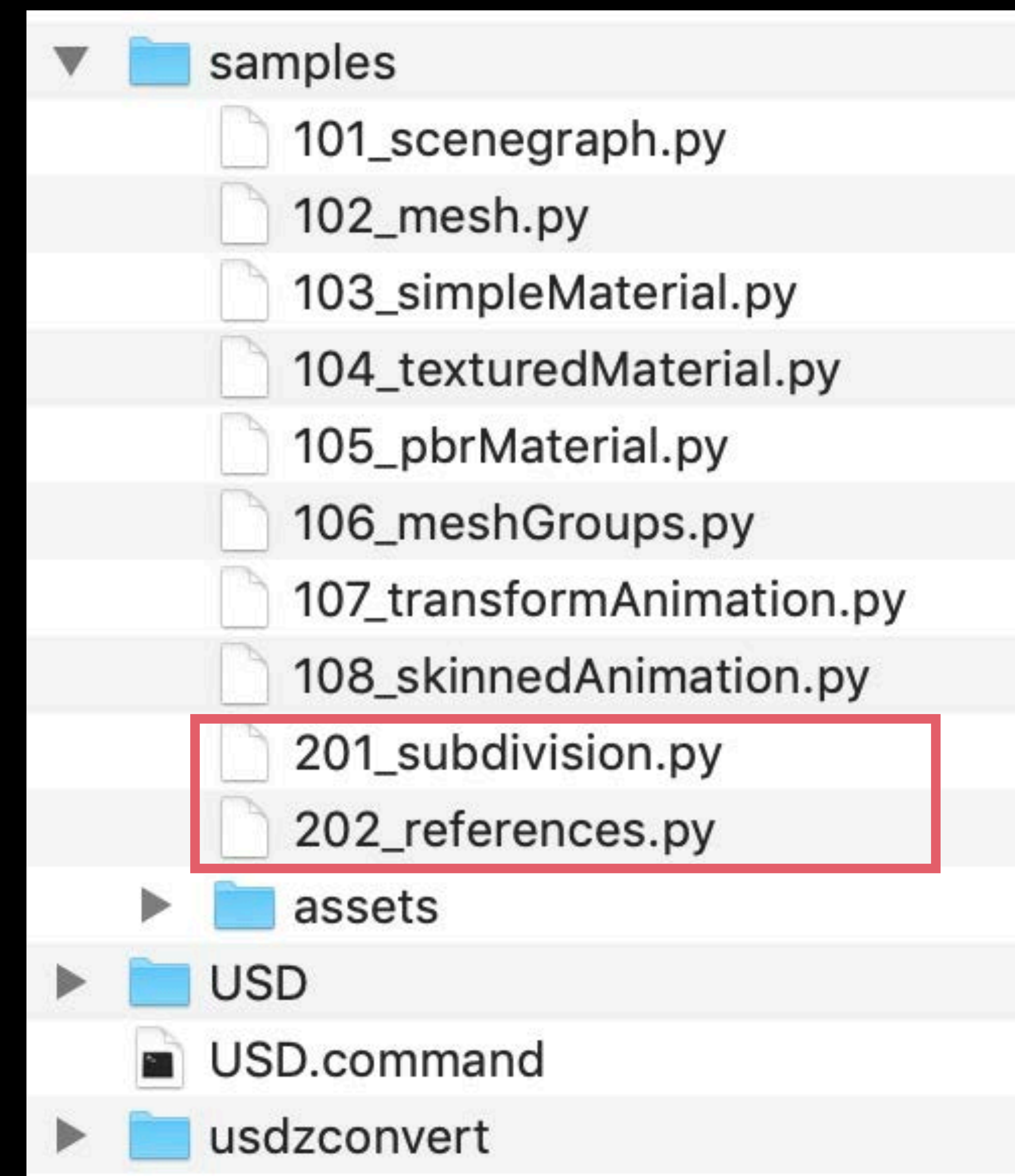
Python Samples

USD highlights

NEW

“Samples” folder

- Subdivision surface with creases
- References with overrides



Summary

Universal Scene Description

- Powerful library for authoring and deploying 3D content

Workflows to create and convert assets to usdz

- usdzconvert converts .obj, .glTF, .fbx, .abc...

Download Python usdz tools

- Includes usdzconvert, pre-compiled USD library and sample scripts

More Information

developer.apple.com/wwdc19/602

USD and usdz Lab

Wednesday, 3:00

Advances in AR Quick Look

Friday, 9:00

