

Xflow

Declarative Data Processing in XML3D



German Research
Center for Artificial
Intelligence GmbH



UNIVERSITÄT
DES
SAARLANDES

Beyond static geometry

- Different type of animations

- Rigid Transformations ✓

- Morphing ✗

- Skinning ✗

Use group transformations

Need for vertex processing

- Several technologies

- CPU, GPU
(vertex shaders, CUDA, OpenCL)

Make it Platform Independent!

Go for Generic

- Avoid specialized elements
 - i.e. specification blow up
- Find generic solution to describe data processing
 - Should be at least as powerful as vertex shaders in GLSL / HLSL
- Still: Platform Independent!

~~<bone> <skinning>
<skeleton> <morphing>
<blendshape>~~



<data script="..." >

Introducing: Xflow

- Mechanism for declarative data processing
- Describe **dataflow process** inside xml3d documents
 - Composition of data fields
 - Data modification through scripts / shaders

<data>

A simple mesh

```
<mesh type="triangles" >
  <int name="index" >0 1 2 1 2 3 ...</int>
  <float3 name="position" >1.0 2.0 -0.2 ...</float3>
  <float3 name="normal" >0.0 1.0 0.0 ...</float3>
</mesh>
```

=

```
<mesh type="triangles" >
  <data>
    <int name="index" >0 1 2 1 2 3 ...</int>
    <float3 name="position" >1.0 2.0 -0.2 ...</float3>
    <float3 name="normal" >0.0 1.0 0.0 ...</float3>
  </data>
</mesh>
```

Instantiate with <data>

```
<defs>
  <data id="meshData" >
    <int name="index" >0 1 2 1 2 3 ...</int>
    <float3 name="position" >1.0 2.0 -0.2 ...</float3>
    <float3 name="normal" >0.0 1.0 0.0 ...</float3>
  </data>
</defs>
...

<mesh type="triangles" src="#meshData" />
<group style="transform: ... " >
  <mesh type="triangles" src="#meshData" />
</group>
```

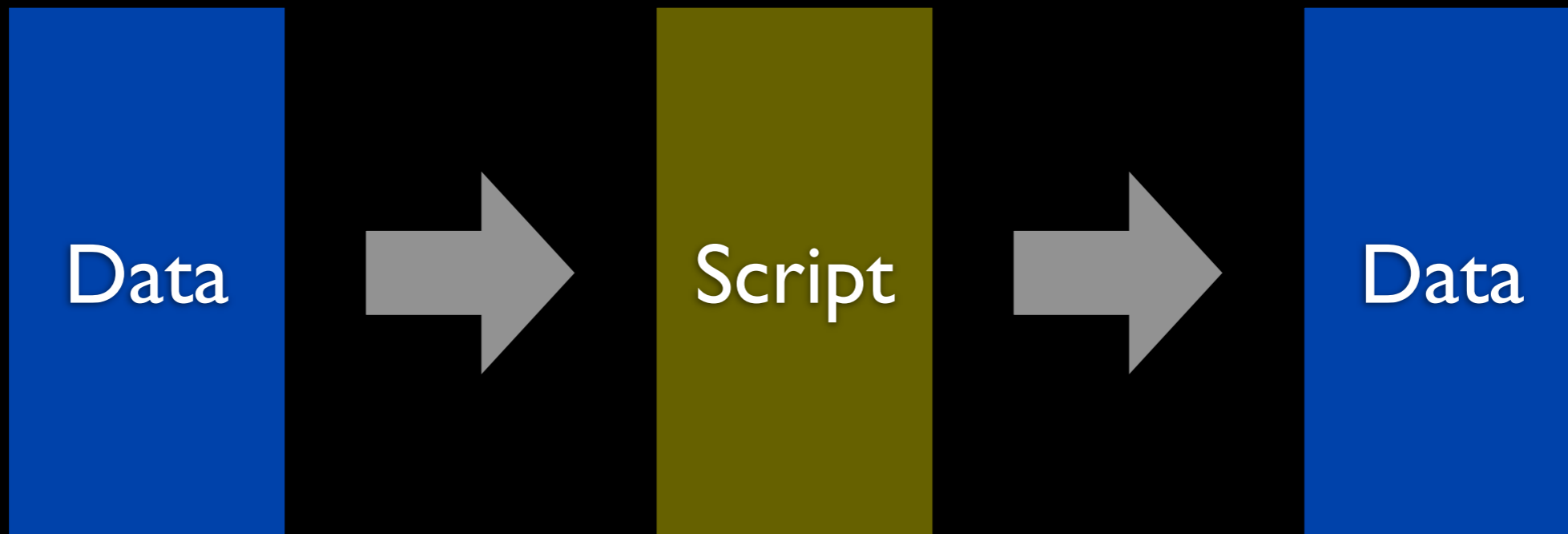
Reuse part of data

```

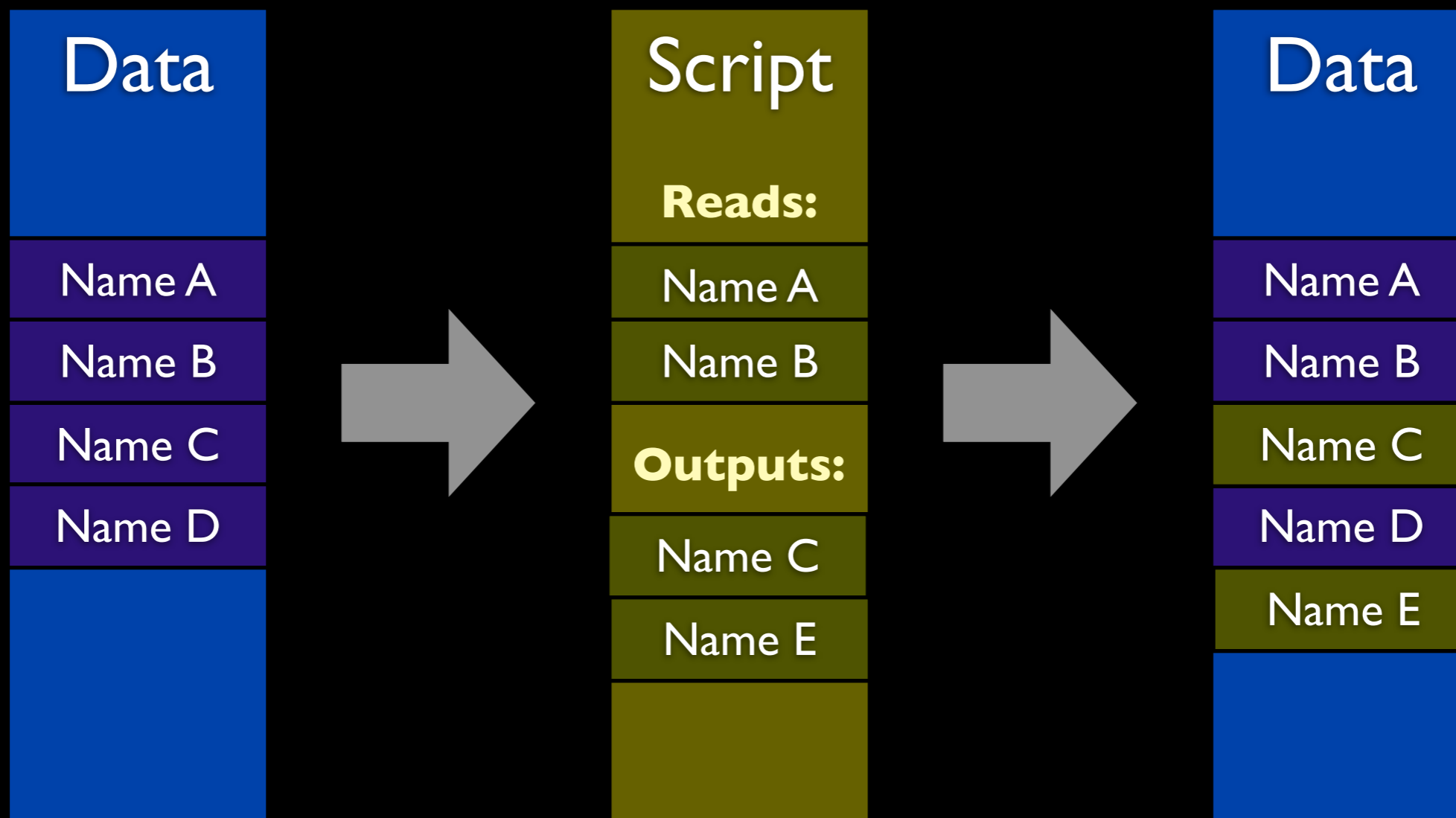
<defs>
  <data id="basicMeshData" >
    <int name="index" >0 1 2 1 2 3 ...</int>
    <float3 name="position" >1.0 2.0 -0.2 ...</float3>
    <float3 name="normal" >0.0 1.0 0.0 ...</float3>
  </data>
</defs>
...
<mesh type="triangles">
  <data src="#basicMeshData" />
  <float3 name="color" >1.0 0.7 0.1 ...</float>
</mesh>
...
<mesh type="triangles">
  <data src="#basicMeshData" />
  <float3 name="color" >0.2 0.7 0.9 ...</float>
</mesh>

```

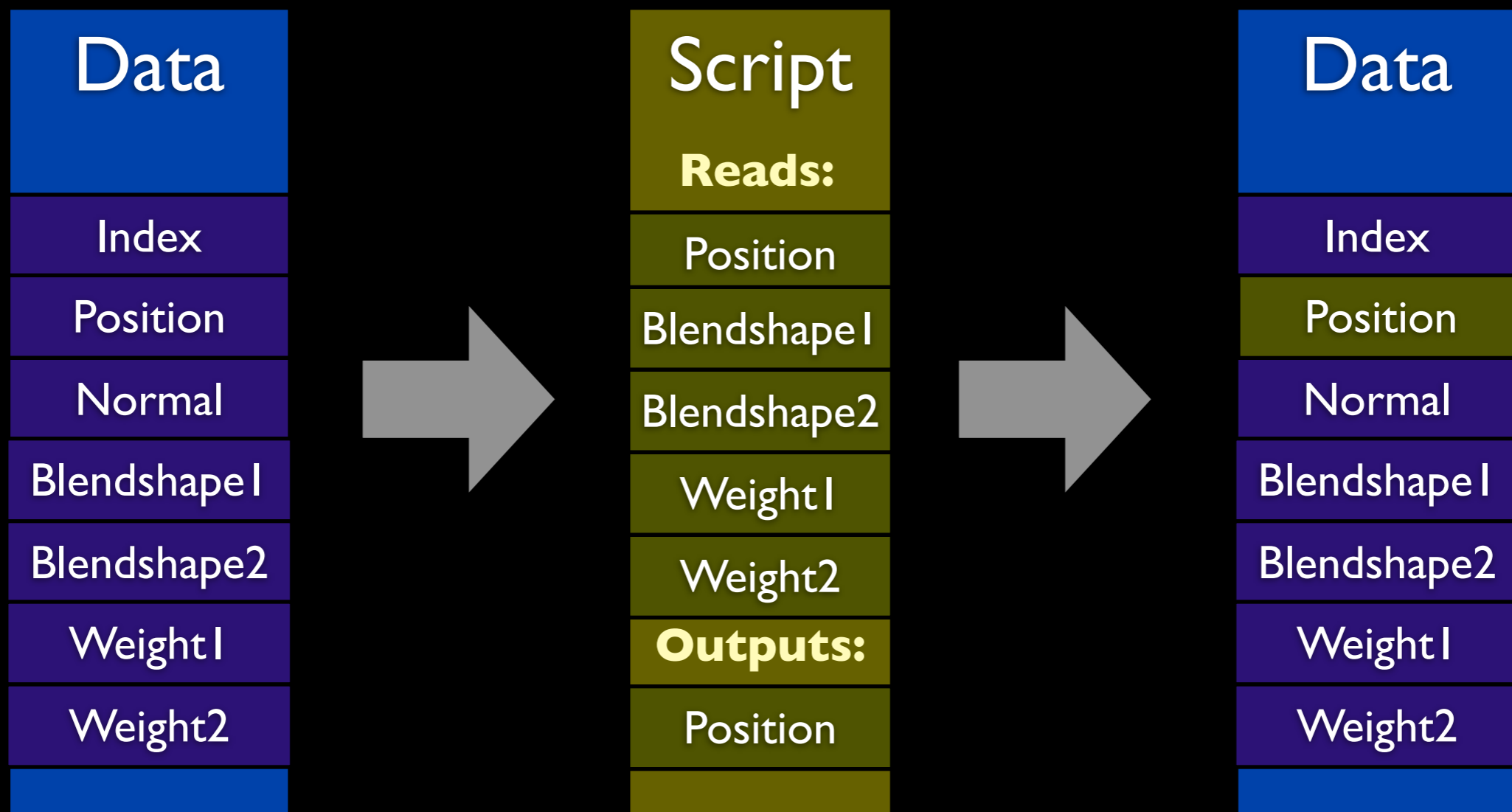
Data Scripts



Data Scripts in Detail



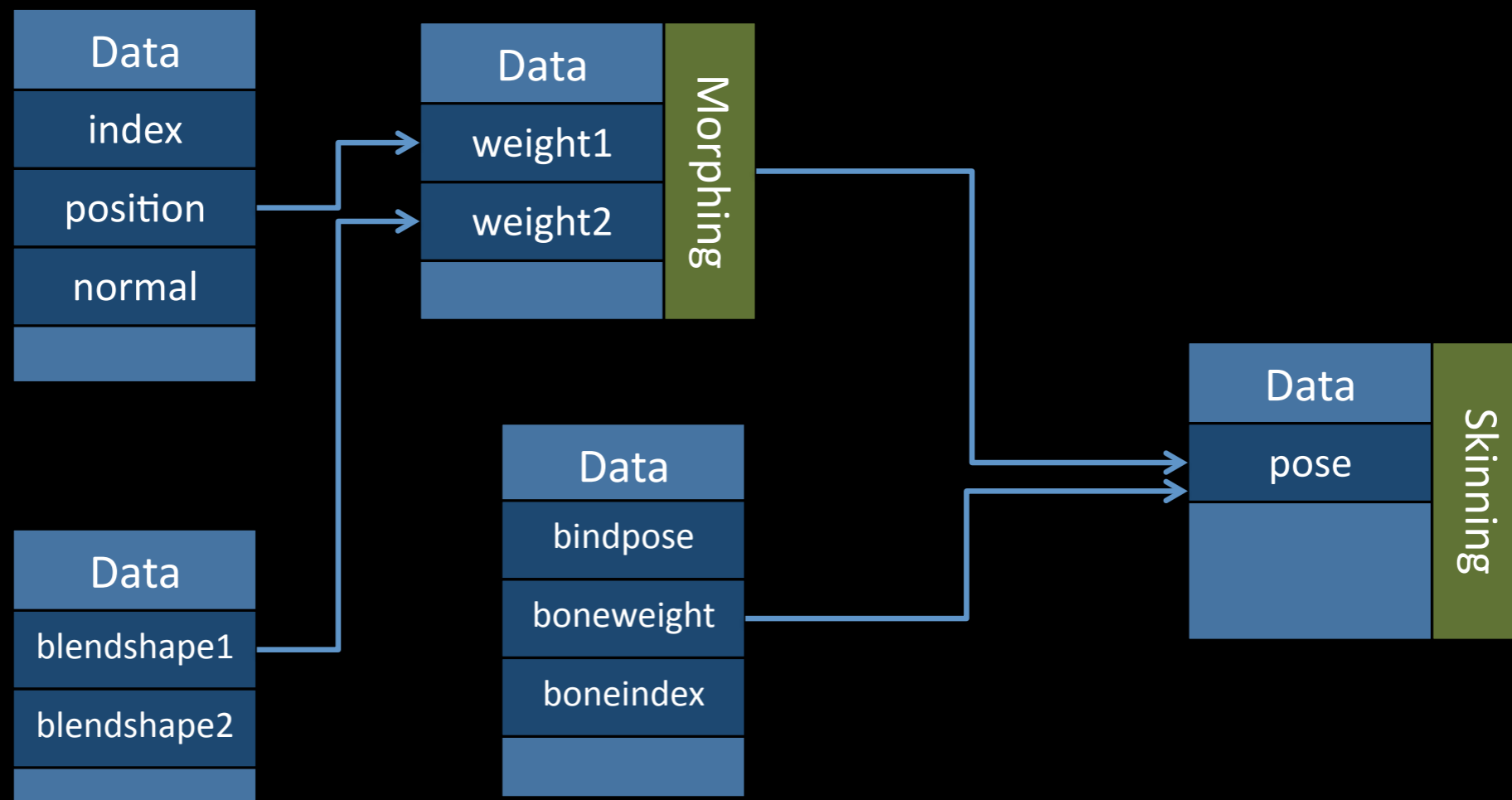
Example: Morphing



Script attached to <data>

```
<data id="meshData" script="morphing.xxx" >  
  <int name="index" >0 1 2 1 2 3 ...</int>  
  <float3 name="position" >1.0 2.0 -0.2 ...</float3>  
  <float3 name="normal" >0.0 1.0 0.0 ...</float3>  
  <float3 name="blendshape1" >1.5 2.3 -0.4 ...</float3>  
  <float3 name="blendshape2" >1.2 1.9 0.1 ...</float3>  
  <float name="weight1" >0.3</float3>  
  <float name="weight2" >0.5</float3>  
</data>
```

Complex Data Flow



... In Markup

```
<mesh type="triangles" >  
  <data script="skinning.xxx" >  
    <data script="morphing.xxx" >  
      <data src="#meshData" />  
      <data src="#blendData" />  
      <float3 name="weight1" >0.8</float3>  
      <float3 name="weight2" >0.2</float3>  
    </data>  
    <data src="#skinningData" />  
    <float4x4 name="pose">...</float4x4>  
  </data>  
</mesh>
```

More about Data Scripts

- Platform independent description
 - Use AnySL
- Safe exposure of parallel computing via OpenCL / CUDA
- Script execution not bound to specific hardware
 - Can be executed on GPU or CPU depending on context and renderer

More Applications

- Displacement Mapping
 - Add Texture to Data Element
- Not limited on vertex data
 - Use for Image Processing
- Use content of <data> for shader parameters
 - Process textures and forward to shader

External Data

```
<data src="external.xxx" />
```

Can be any file that
can be converted to a
“name” => “value”
table



Thank you!

www.xml3d.org