

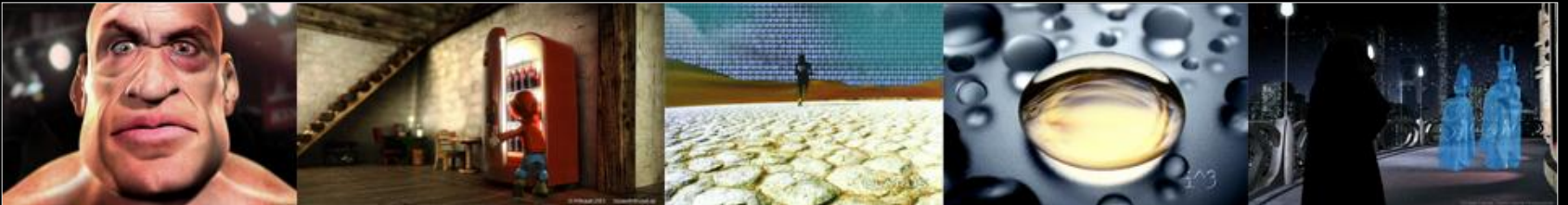
Dependency Graph and Nodes – Day 5

***Kristine Middlemiss**, Senior Developer Consultant
Autodesk Developer Network (ADN)*

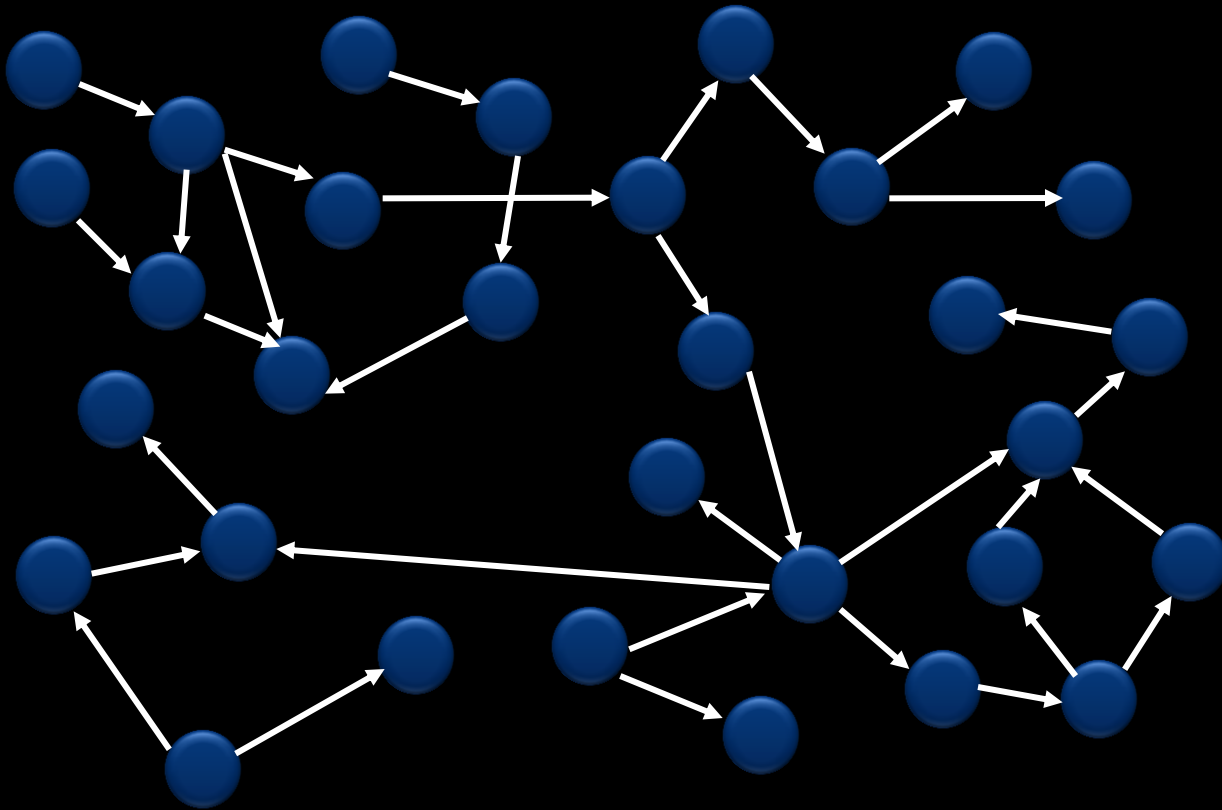
Autodesk®

Agenda

- Components of Maya Node and API classes
- Building Maya Custom Node with MPxNode



Dependency Graph



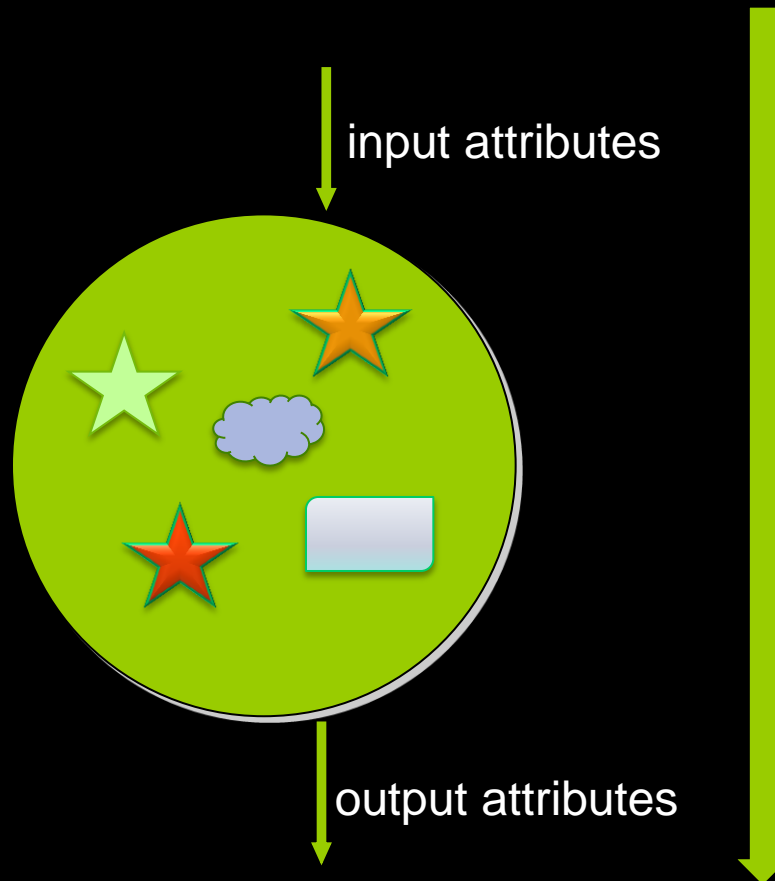
What does a node do?

- Know its own attributes
- Store data efficiently in “datablocks”
- Accept input, compute, generate output
- Connect with other nodes through connections

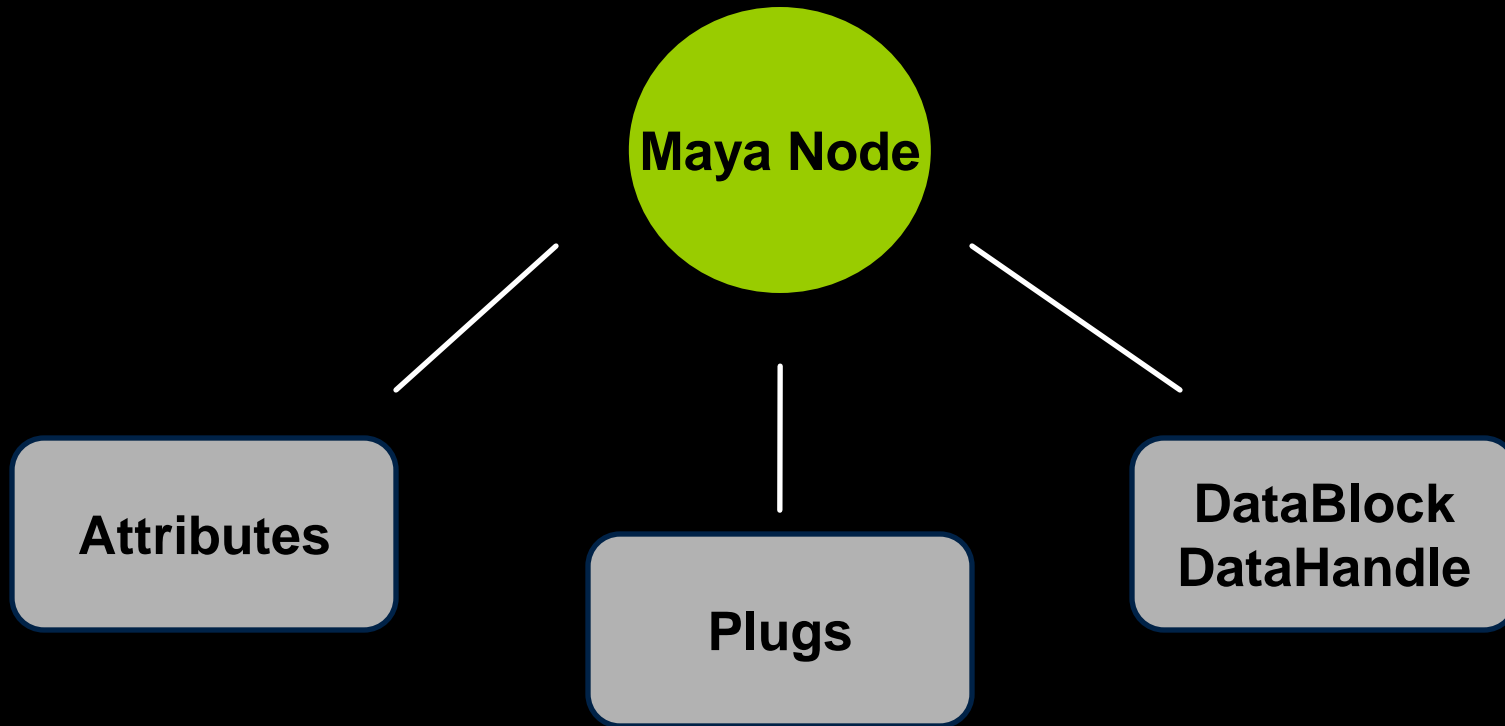


What does a node do?

Accept input data, compute output data



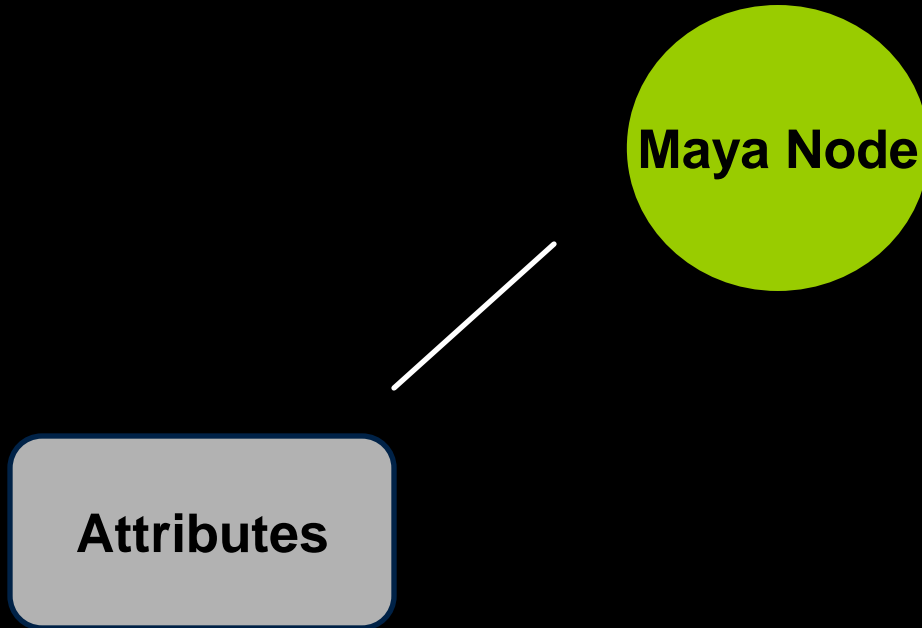
Maya Node Structure



Attributes



Attributes



Attributes

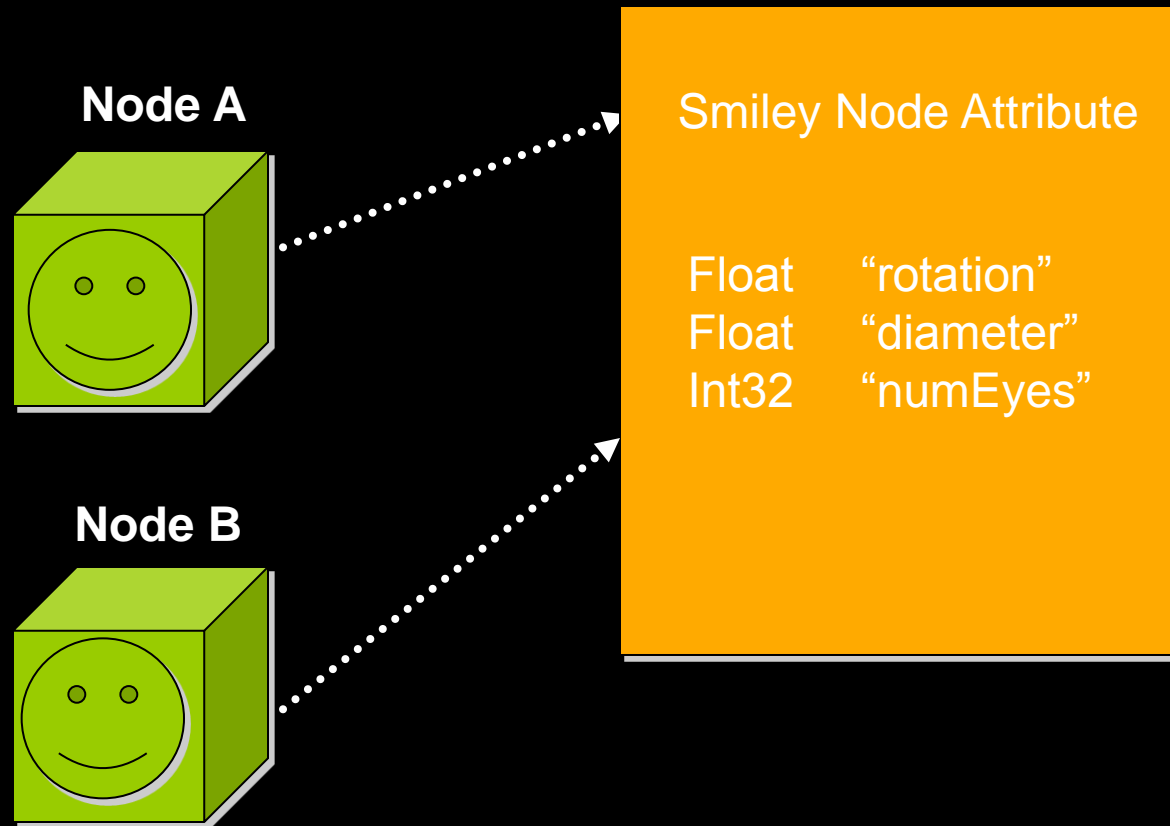
- Describe data that belongs to nodes of a given type

Smiley Node Attribute

Float	“rotation”
Float	“diameter”
Int32	“numEyes”

Attributes

- Attributes are shared among nodes of the same type **and** all derived node types



Node and their Attributes

All nodes		Hierarchy	Tags
By first letter		A B C D E F G H I J K L M N O P Q R S T U V W X	
By substring(s)		mesh	

Attributes (144)

Attribute quick index omitted (too many attributes to show them all).

Long name (short name)	Type	Default	Flags
inMesh (i) The input (creation) mesh.	mesh	NULL	
outMesh (o) The output (local space) mesh.	mesh	NULL	
outGeometryClean (ogc) Tells if aOutMesh is only dirty due to tweak change.	Message	n/a	
cachedInMesh (ci) Internal cached input mesh used only for file I/O. This attribute is for internal use and should never be directly manipulated by the user. If it is modified the results will be unpredictable.	mesh	NULL	
worldMesh (w) The world space meshes.	mesh	NULL	
outSmoothMesh (os) The smoothed output (local space) mesh.	mesh	NULL	
cachedSmoothMesh (cs) The cached smoothed mesh, that holds all the meshes from different levels of subdivision	mesh	NULL	
smoothWarn (sw) Check for reasonable face count before smoothing.	bool	true	
smoothLevel (lev) The level of smoothing to perform on the mesh.	short	2	
continuity (co)	float	1	

Name

Data Type

Structure

Property

Attributes

Define the interface of the node including

- **Names** of the inputs and outputs
- **Data** accepted by the inputs and outputs
- **Structure** of the inputs and outputs
- **Properties** of the inputs and outputs



Attribute Data Types

- Basic
 - Numeric (float, int, etc.)
 - String
 - Matrix
 - Etc.
- Complex
 - Mesh
 - NurbsSurface
 - Generic (accepts more than one type)
 - Etc.

Attribute Structures

Attribute Data Types will fit into one of these four:

1. Simple
2. Compound
3. Array
4. Compound Array

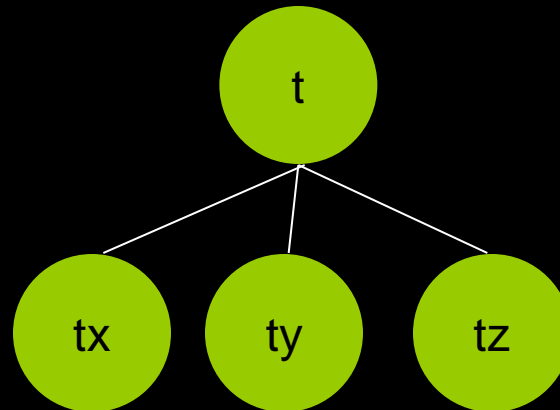
Attribute Structures

1. Simple

- handles a single piece of data
- Example: `polySphere1.radius`

2. Compound

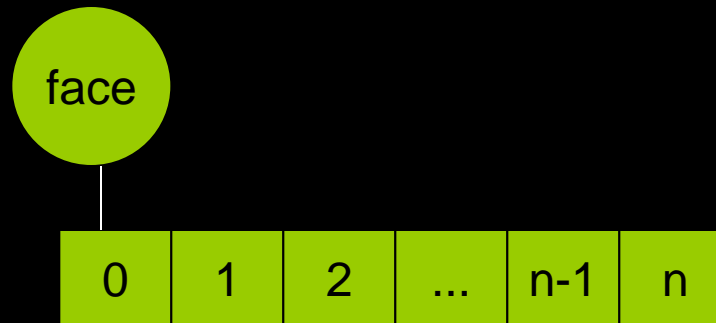
- 1 or more attributes grouped under a parent attribute
- Example: `translate` is a compound with children (`tx`, `ty`, `tz`)



Attribute Structures

3. Array (also referred to as a Multi)

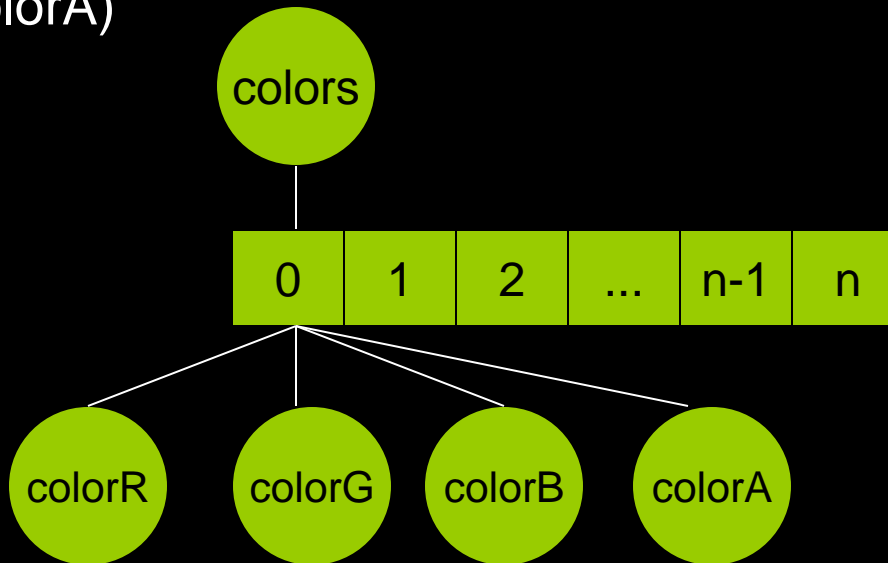
- An array of simple data
- Example: `pSphereShape1.face`



Attribute Structures

4. Compound array

- An array of compound data
- Example: mesh colors with children (colorR, colorG, colorB, colorA)



Each Attribute has Properties

By default, attributes are:

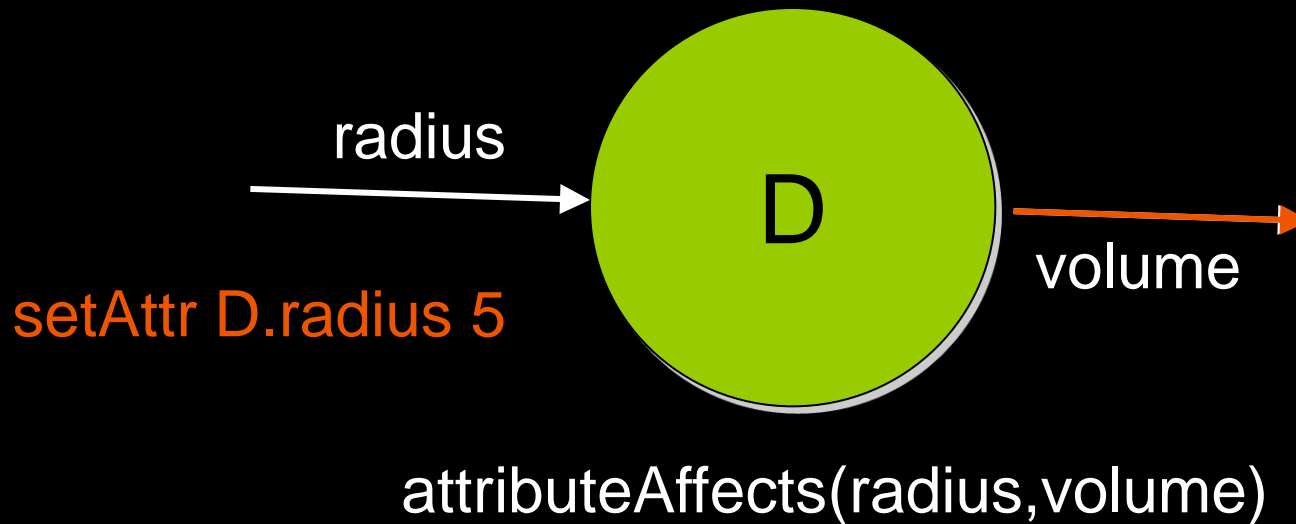
- Readable.
- Writable.
- Connectable.
- Storable.
- Cached.
- Not arrays.
- Have indices that matter.
- Do not use an array builder.
- Not keyable.
- Not hidden.
- Not used as colors.
- Not indeterminant.
- Set to disconnect behavior kNothing.

Attribute Properties

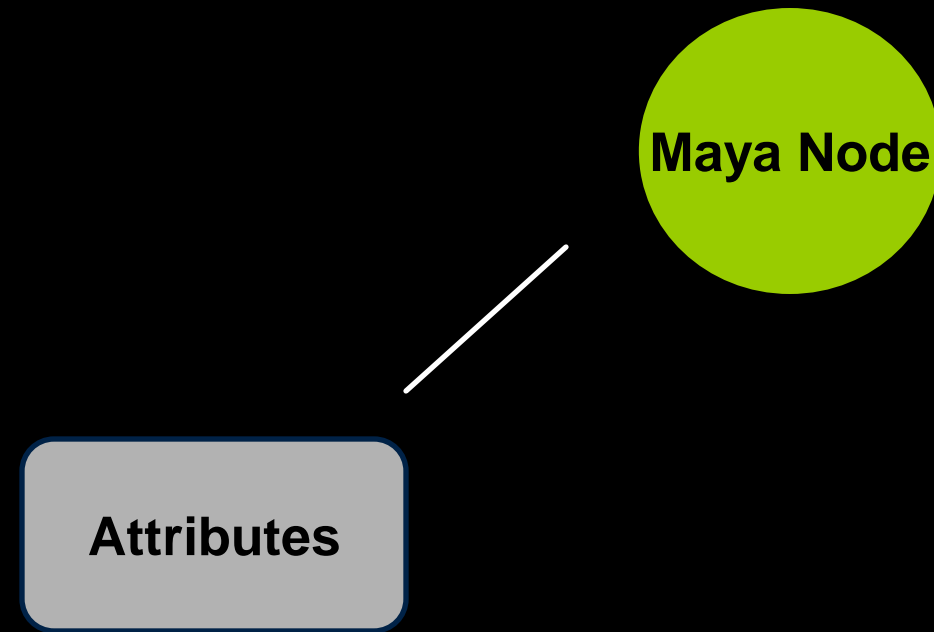
- Readable
 - Attribute can be connected as a source
 - Example: polyCylinder1.output
- Writable
 - Attribute can be connected as a destination
 - Example: pCylinderShape1.inMesh
- Read/Write
 - Attribute can be connected as both a source and a destination
 - Example: pCylinder1.tx

Attribute Affects

- Maya has no concept of input and output attribute
- Once attributes are created on a node, an “attributeAffects” relationship needs to be setup to denote a dependency (indicating input and outputs)



API Classes for Attributes



MFnAttribute

API Classes for Attributes

- Base Class: MFnAttribute
- Most Common Used Child Classes
 - MFnNumericAttribute
 - MFnCompoundAttribute
 - MFnTypedAttribute
 - MFnMatrixAttribute
 - MFnGenericAttribute

Steps to Create an Attribute

Define a Python function containing these steps:

1. create attribute using MFnAttribute*
2. set the attribute properties
3. add to the node
4. setup any attribute affects relationships

MFnNumericAttribute

- Function set for simple numeric value attribute

```
import maya.OpenMaya as OpenMaya
```

```
....
```

```
nAttr = OpenMaya.MFnNumericAttribute()
```

```
simpleNode.input = nAttr.create( "input", "in", OpenMaya.MFnNumericData.kFloat, 0.0 )
```

```
nAttr.setWritable(1)
```

```
nAttr.setStorable(1)
```

```
nAttr.setKeyable(1)
```

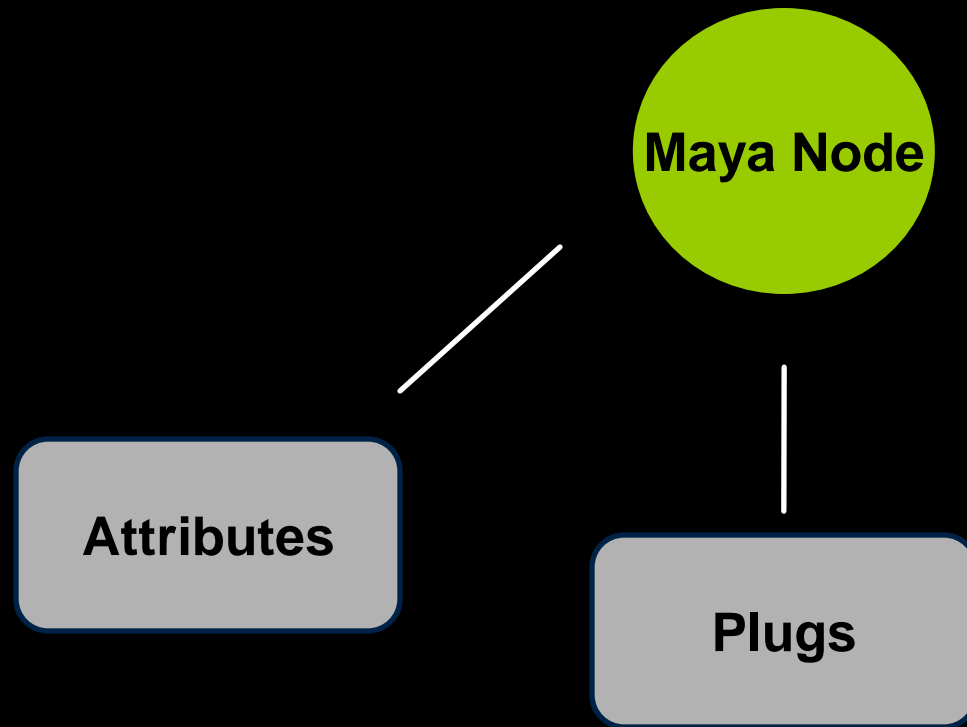
```
simpleNode.addAttribute(simpleNode.input)
```

```
simpleNode.attributeAffects(simpleNode.input, <output> )
```


Plugs



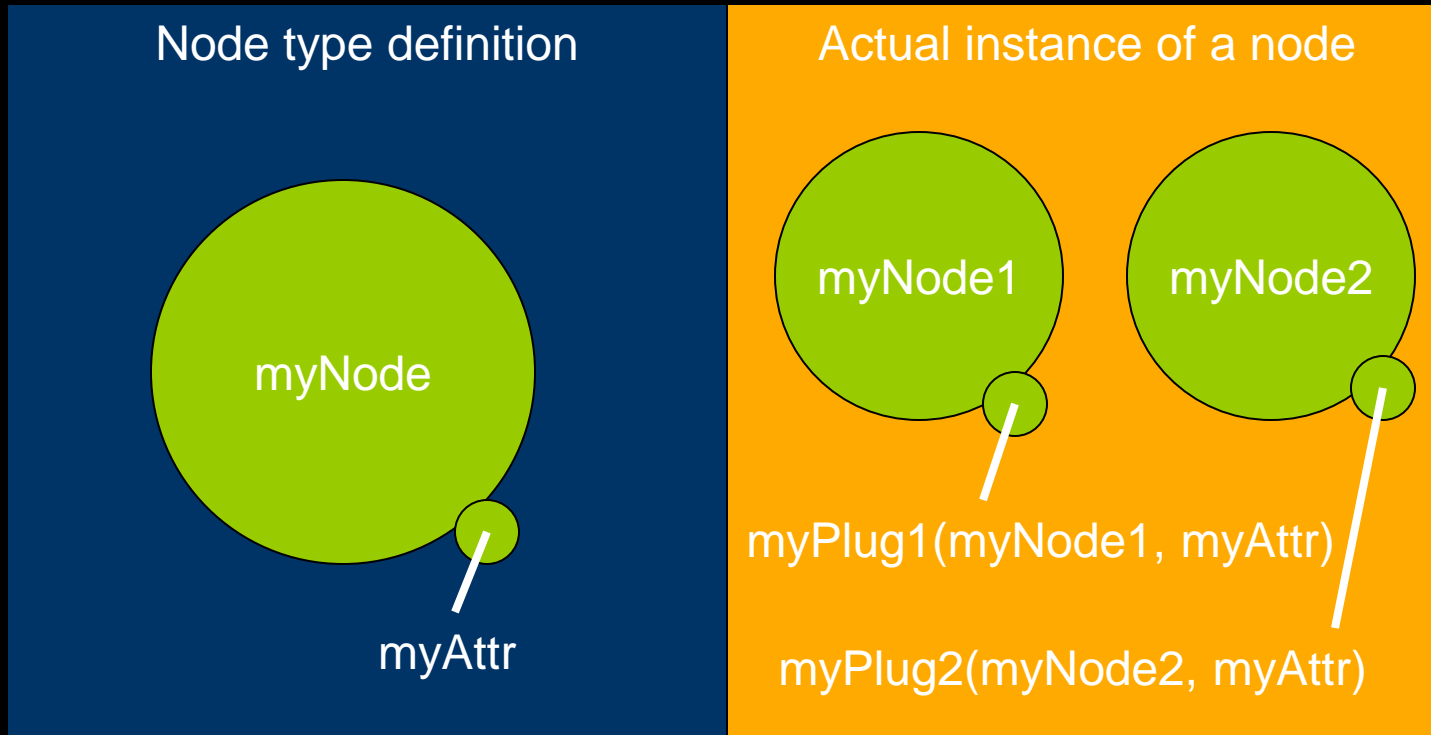
Plugs



MFnAttribute

Plugs

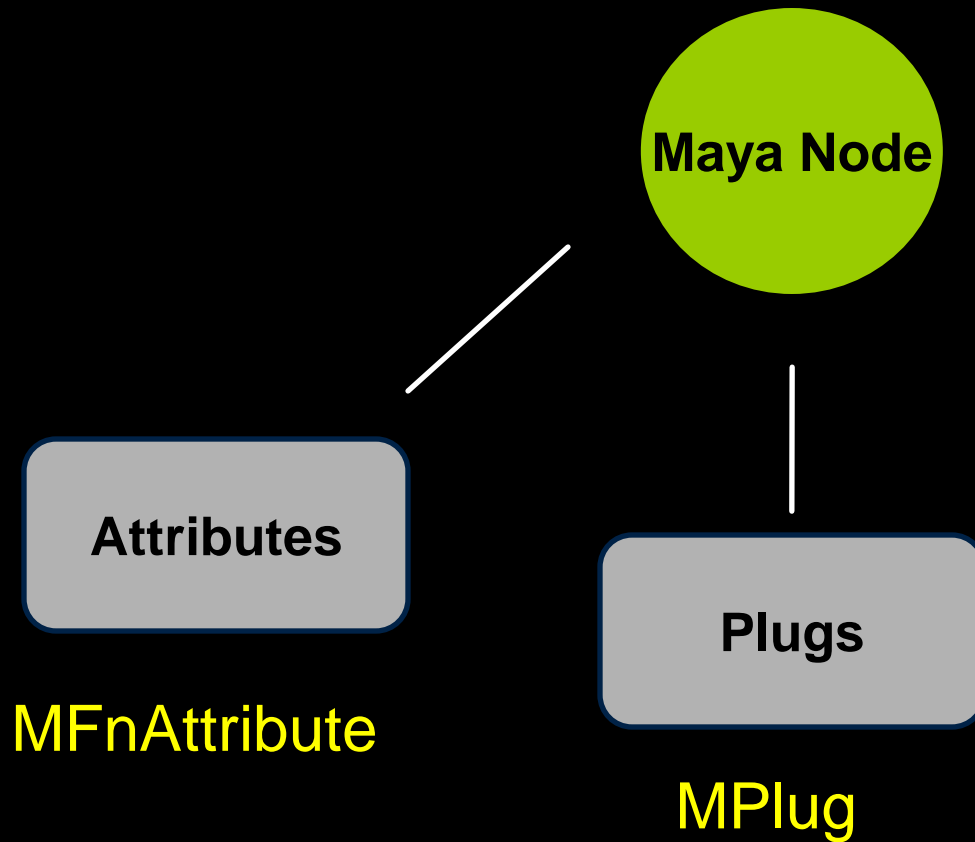
- Pointer to an attribute on a specific node (ie. a specific instance of an attribute)



Plugs

- Plugs can be used to:
 - query a value
 - set a value
 - create a connection
 - remove a connection
 - query connection(s)
- Does not store attribute data

API Classes for Plugs



API Class for Plug: MPlug

Attribute and Node Operations:

- `MPlug::MPlug (const MObject &node, const MObject &attribute)`
- `MObject MPlug::attribute (MStatus * ReturnStatus)`
- `MObject MPlug::node (MStatus * ReturnStatus)`

Operations for Compound Attribute and Connections:

- `MPlug MPlug::parent (MStatus * ReturnStatus)`
- `MPlug MPlug::child (unsigned int index, MStatus *ReturnStatus)`
- `bool MPlug::connectedTo (MPlugArray & array, bool asDst, bool asSrc, MStatus * ReturnStatus = NULL)`

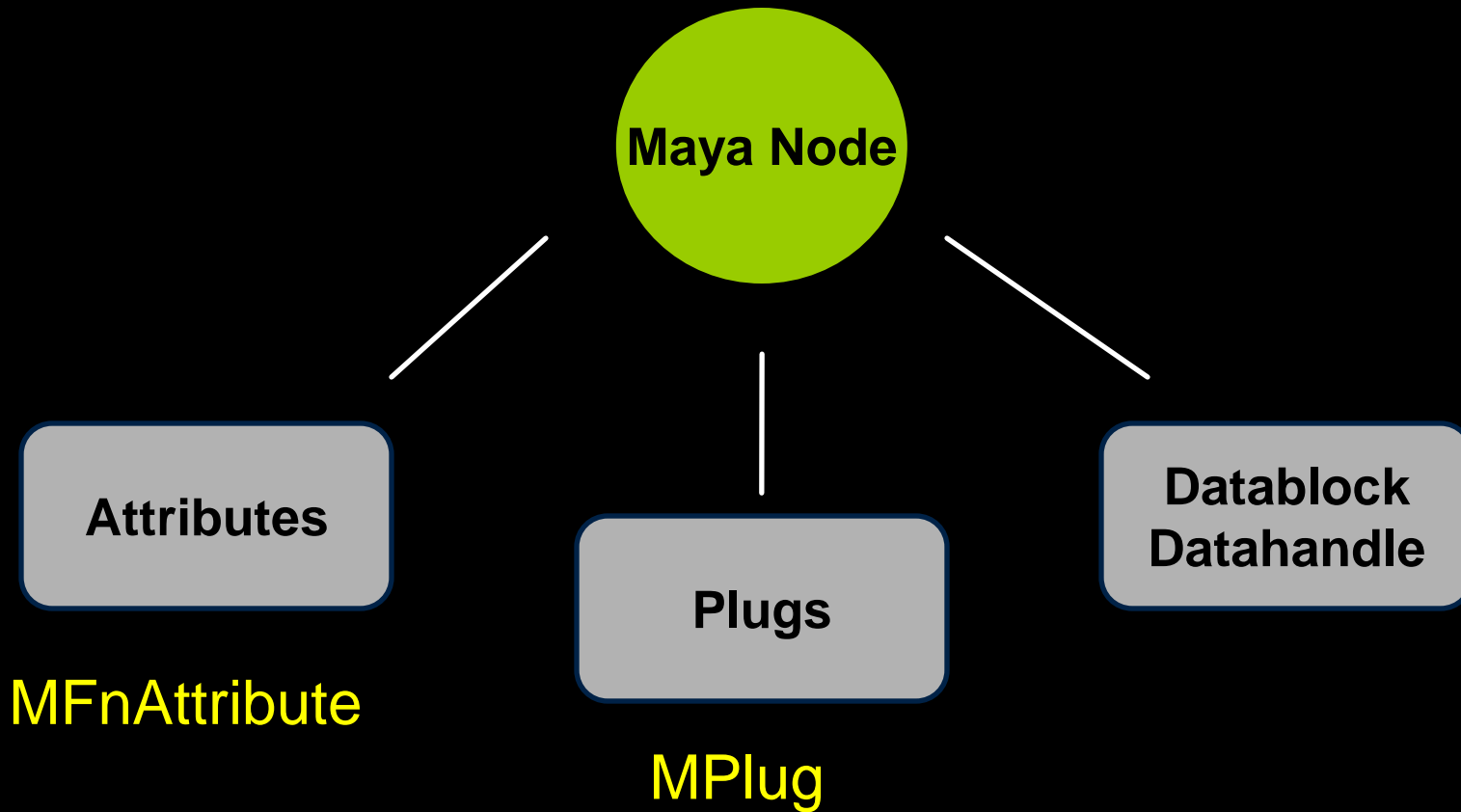
Manipulation of Data

- `MStatus MPlug::getValue (double & val, MDGContext & ctx)`
- `MStatus MPlug::setValue (double val)`

Datablocks

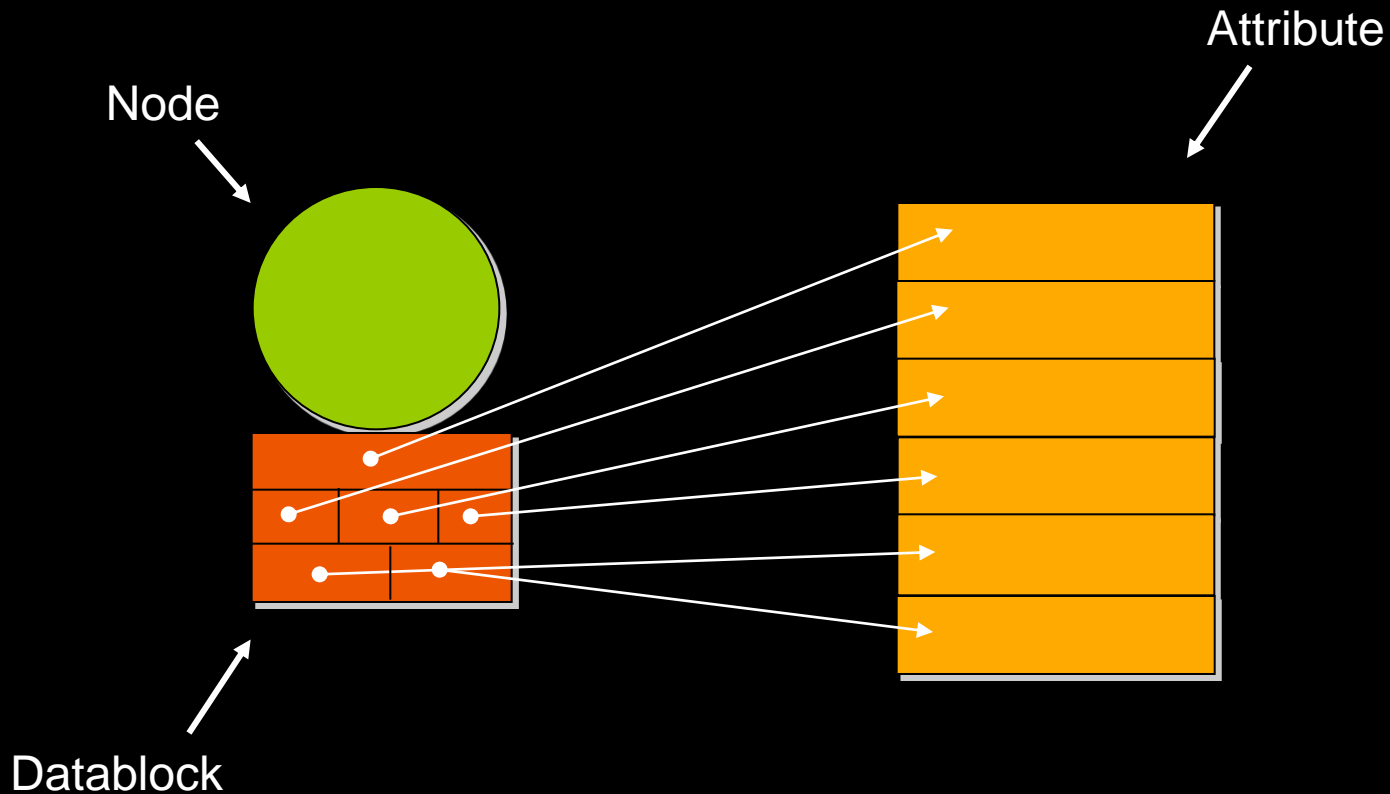


Datablocks



Datablocks

- Node stores data for every attribute (arrays are special, more details later)

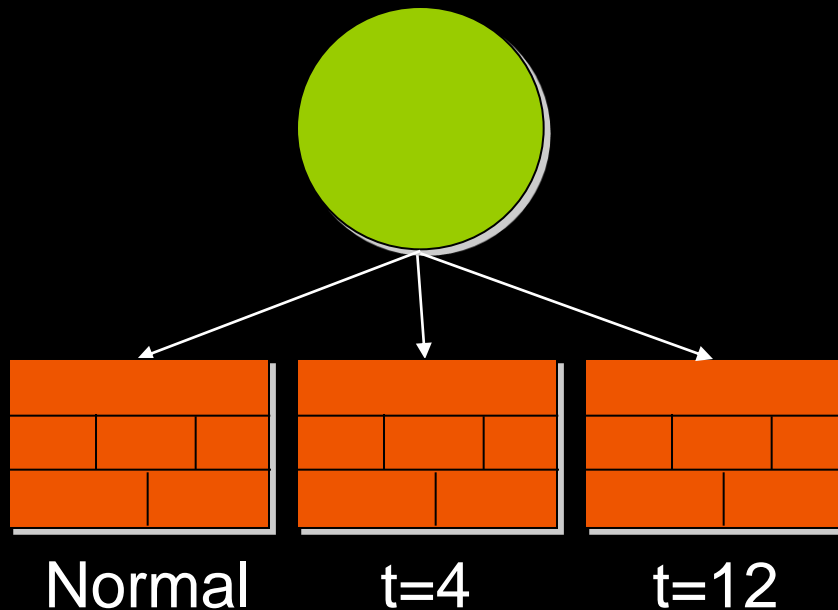


Datablocks & Datahandles

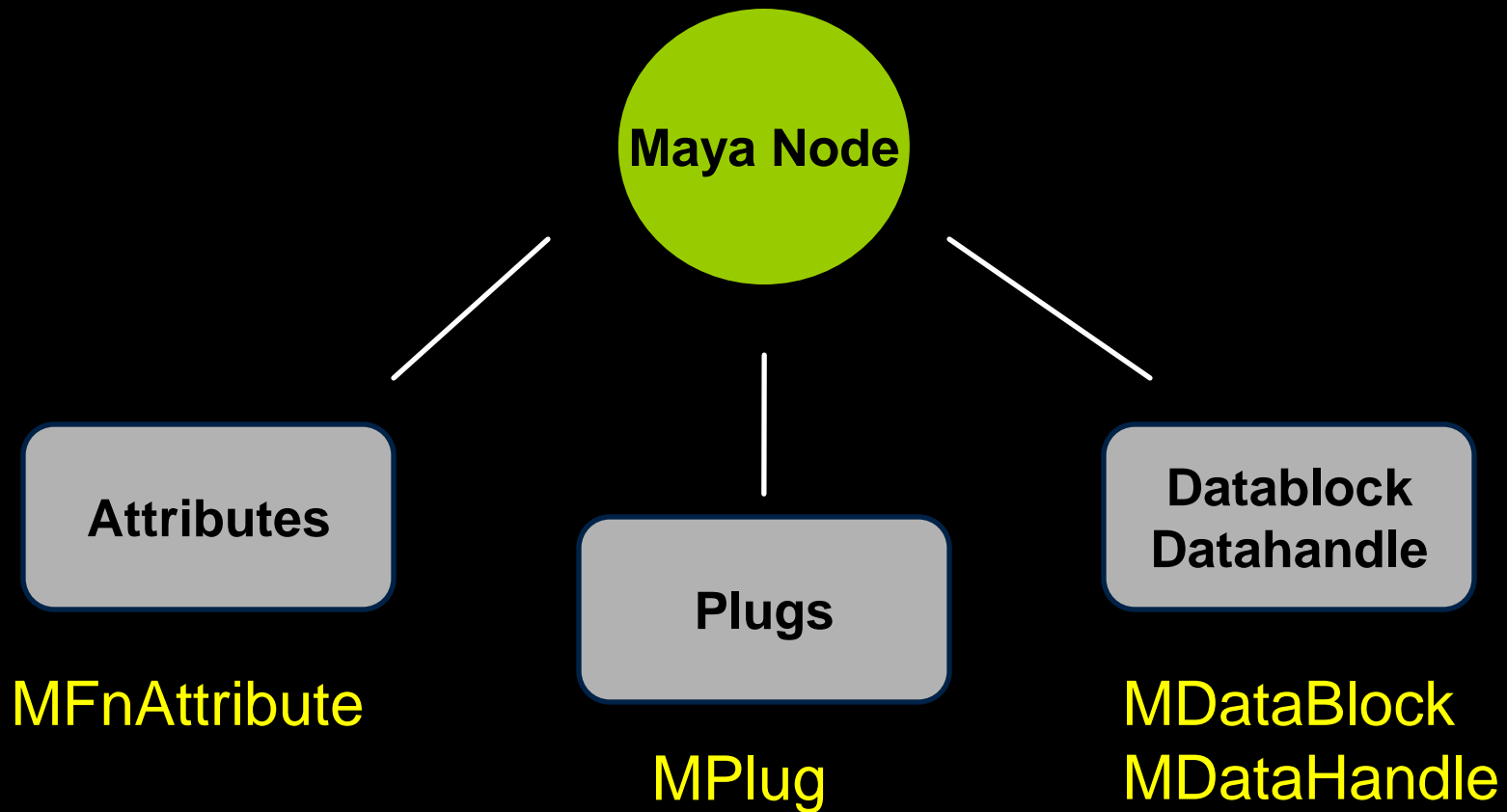
- Datablock is the actual storage for the input and output data of a node
- For every non-array attribute, datablock stores:
 - Data
 - Dirty/clean status
- Data handles are lightweight pointers into the data in the datablock

Datablocks

- Nodes hold an array of datablocks
 - One datablock for a normal evaluation (using current time)
 - More datablocks for timed contexts (specified times)



Maya Node Structure: API Classes



API Classes for Datablock

- MDataBlock

- only valid during compute()
- Pointers to data block should not be retained after compute()

```
MStatus MPxNode::compute(const MPlug& plug, MDataBlock& dataBlock)
```

```
MDataHandle MDataBlock::inputValue(const MPlug & plug, MStatus * ReturnStatus)
```

```
MDataHandle MDataBlock::outputValue ( const MPlug & plug, MStatus * ReturnStatus)
```

```
MDGContext MDataBlock::setContext (const MDGContext & ctx )
```

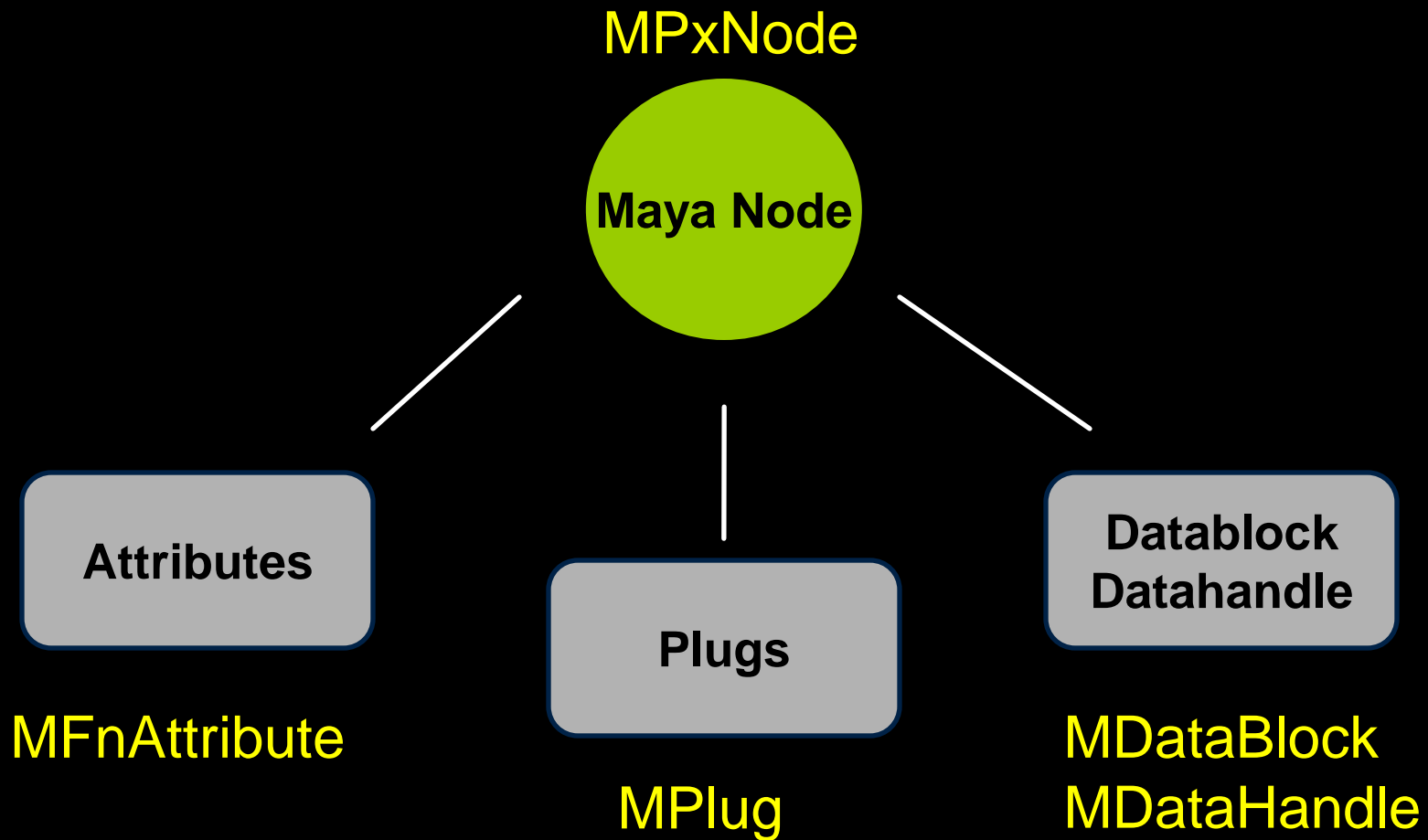
- MDataHandle

- a smart pointer for information contained in data block

```
double & MDataHandle::asDouble ( )
```

```
void MDataHandle::setDouble ( double )
```

Maya Node Structure: API Classes



Custom Node Implementation



Custom DG Nodes in Maya

- Entirely new operations
 - MPxNode
- Extend existing Maya nodes
 - MPxDeformerNode
 - MPxFieldNode
 - MPxEmitterNode
 - MPxSpringNode
 - MPxIkSolverNode
 - MPxHwShaderNode

MPxNode Registration

- Every node requires a unique identifier.

```
MTypeId myNode::id( 0x80000 );
```



- For plug-ins that you intend to share between sites
 - Will require a globally unique ID issued to you by Autodesk.
 - IDs are allocated in blocks of 64/128/256/512.
 - Contact ADN M&E for unique global IDs.

MPxNode Registration

- To register your node in Maya:

```
kPluginNodeTypeName = "simpleNode"  
simpleNodeId = OpenMaya.MTypeId(0x80001)
```

```
def initializePlugin(mobject):  
    mplugin = OpenMayaMPx.MFnPlugin(mobject)  
    try:  
        mplugin.registerNode( kPluginNodeTypeName, simpleNodeId, nodeCreator, nodeInitializer )  
    except:  
        sys.stderr.write( "Failed to register node: %s" % kPluginNodeTypeName )  
        raise
```

- To deregister your node

```
def uninitializePlugin(mobject):  
    mplugin = OpenMayaMPx.MFnPlugin(mobject)  
    try:  
        mplugin.deregisterNode( simpleNodeId )  
    except:  
        sys.stderr.write( "Failed to deregister node: %s" % kPluginNodeTypeName )  
        raise
```

MPxNode Creator Function

The creator method is called to return a new instance of the node

```
def nodeCreator():  
    return OpenMayaMPx.asMPxPtr( simpleNode() )
```

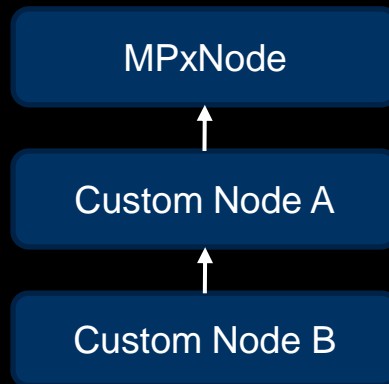
MPxNode initialize Function

Reminder || Override this method to define the attribute interface for your node:

1. create the attribute
2. set the attribute's flags
3. add the attribute to the node
4. define attribute relationships
5. Inherit attributes if necessary

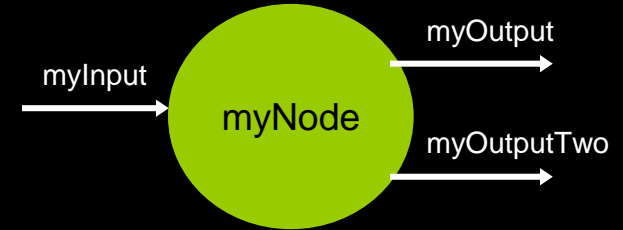
Attribute Operations

- `MStatus MPxNode::addAttribute (const MObject & attr)`
- `MStatus MPxNode::attributeAffects (const MObject & whenChanges, const MObject & isAffected)`
- `MStatus MPxNode::inheritAttributesFrom (const MString & parentClassName)`



MPxNode

- MPxNode::initialize()



```
def nodeInitializer():
```

```
    nAttr = OpenMaya.MFnNumericAttribute()
```

```
    simpleNode.myInputAttr = nAttr.create( "myInput", "mi", OpenMaya.MFnNumericData.kFloat, 1.0 )
    nAttr.setStorable(1)
```

```
    simpleNode.myOutputAttr = nAttr.create("myOutput", "mo", OpenMaya.MFnNumericData.kFloat, 1.0)
    nAttr.setStorable(1)
```

```
    simpleNode.myOutputAttrTwo = nAttr.create("myOutputTwo", "motwo",
OpenMaya.MFnNumericData.kFloat, 1.0)
    nAttr.setStorable(1)
```

```
    simpleNode.addAttribute( simpleNode.input )
    simpleNode.addAttribute( simpleNode.myOutputAttr)
    simpleNode.addAttribute( simpleNode.myOutputAttrTwo)
```

MPxNode::addAttribute()

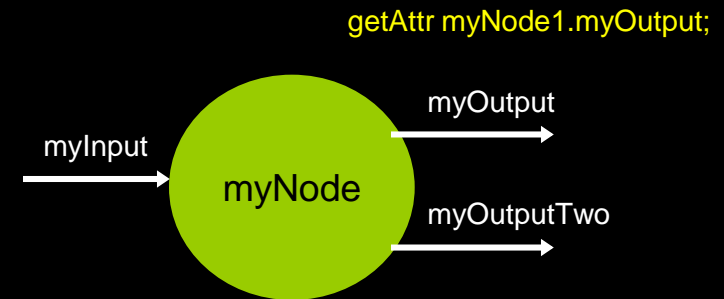
MPxNode::attributeAffects()

```
    simpleNode.attributeAffects(simpleNode.myInputAttr, simpleNode.myOutputAttr);
```

MPxNode

- `MPxNode::compute()`
called when the node is asked to evaluate an output

```
def compute( self, plug, dataBlock):  
    if ( plug == simpleNode.myOutputAttr):  
        # your compute algorithm for output1  
  
    if (plug == simpleNode.myOutputAttrTwo):  
        # your compute algorithm for output2
```

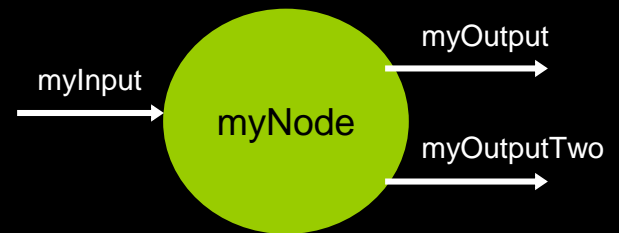


MPxNode

- MPxNode::compute()

```
def compute(self,plug,dataBlock):  
    # ...  
    if( plug == simpleNode.myOutputAttr ):  
  
        inputData = dataBlock.inputValue( simpleNode.myInputAttr )  
        inputValue = inputData.asFloat()  
  
        outputHandle = dataBlock.outputValue( simpleNode.myOutputAttr )  
        outputHandle.setFloat(inputValue * 2 );  
  
        dataBlock.setClean( plug )  
        return OpenMaya.MStatus.kSuccess  
  
    if (plug == simpleNode.myOutputAttrTwo):  
        #...  
  
    return OpenMaya.MStatus.kUnknownParameter
```

myOutput = myInput * 2;



OpenMaya.kUnknownParameter

- One gotcha with MStatus codes as exceptions
- Some DG routines return MStatus::kUnknownParameter (e.g. compute)
- In Python, this DOES NOT map to an exception
- MStatus::kUnknownParameter maps to the Python constant maya.OpenMaya.kUnknownParameter

Workshop Session



Example: simpleNode

In this example, we implement a custom node with two attributes: “input”, ”output”. Whenever the “input” attribute value changes, the “output” attribute will always be the input value multiplied by 2.

Example: sinNode

In this example, we implement a custom node with two attributes: “input”, ”output”. Whenever the “input” attribute value changes, the “output” attribute will calculate the sin of the number.

- Don't forget to import the math module

Example: transCircleNode

In this example, we implement a custom node `transCircleNode`, it takes in a compound input translate attribute “inputTranslate”, and output a compound translate attribute “outputTranslate”, the value of `outputTranslate` is the value of `inputTranslate` plus the value of a circular movement based on current time frame.

- Use the MEL script provided “`transCircleNode.mel`” to set and test your new node

Autodesk