



Advanced ArcGIS Engine Java

Divesh Goyal

Ranjit Iyer

Introductions ...

- **Who are we?**
 - ArcGIS Engine Java Dev team members.
- **Who are you?**
 - ArcGIS Desktop developers/users?
 - Current ArcGIS Engine developers?
 - Interested in extending Desktop with Java
 - Target Platforms?
 - Preferred IDE?

Our assumptions

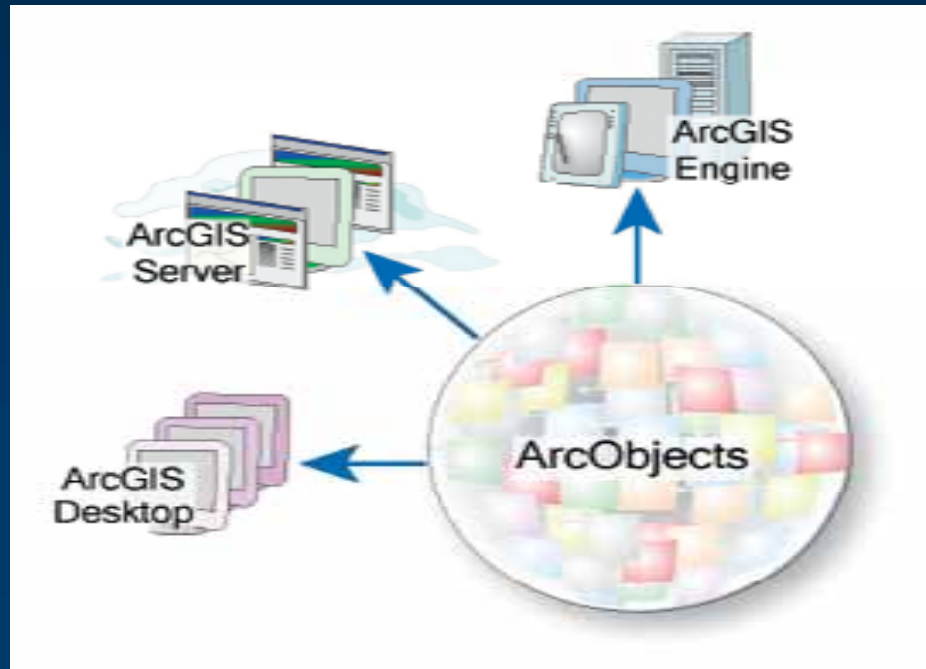
- You are familiar with `ArcObjects` in Java
- You are intrigued by `AutomationException`

Agenda

- **ArcGIS Engine Architecture**
- Programming with ArcObjects
- Geoprocessing with ArcObjects
- ProxyGen
- Back to the future

What are ArcObjects

- Core GIS components written in C++/COM
- Includes coarse-grained objects for ease of development
- Can run in Desktop and Server containers
- Library of objects accessible through...
 - Java
 - C++
 - .Net



COM before the storm

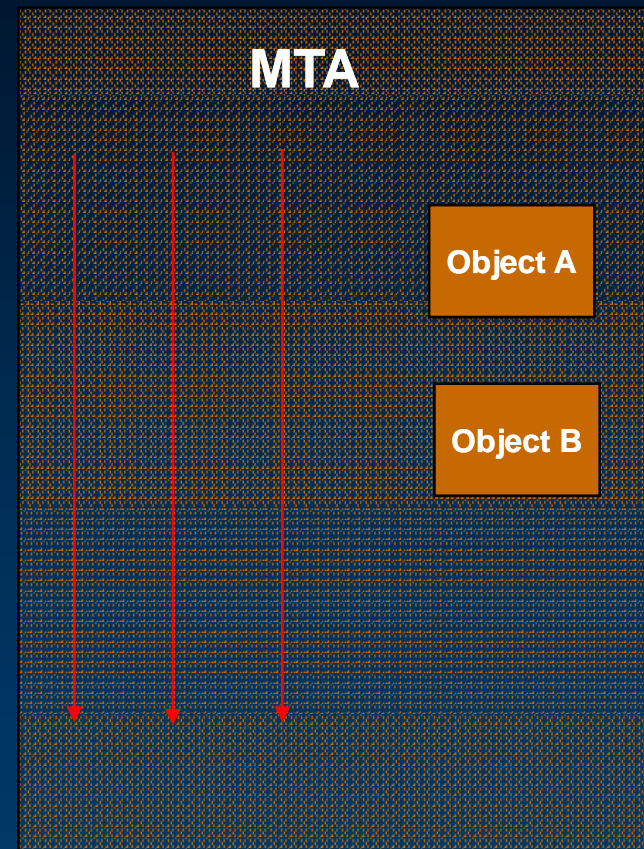
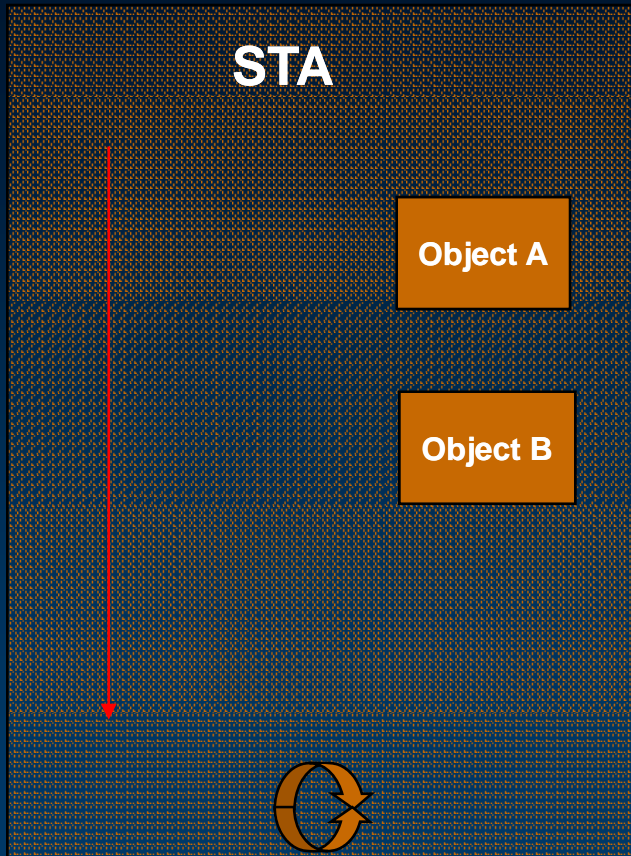
- **COM is both a specification and runtime infrastructure for developing component software**
- **The specification propounds**
 - Interface based programming
 - Unique identifiers (GUID)
 - Protocol for connecting software components
 - Garbage collection via reference counting
- **The runtime provides**
 - Discovering and instantiating objects
 - Marshalling support for communicating components
 - Concurrency and synchronization

A Minimal ArcObject

- Has a unique ID (GUID)
- Implements the *IUnknown* interface
 - QueryInterface()
 - AddRef()
 - Release()
- Publishes it's threading model
 - Thread unsafe (STA)
 - Thread safe (MTA)

Apartments

- A logical boundary for every COM object



ArcObjects live in an STA

Consuming ArcObjects

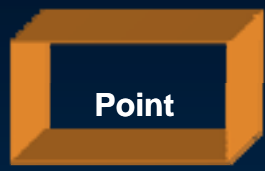
- Client enters an apartment (STA)
- Creates an ArcObject using the GUID
- Explicitly gets an interface to work with
- Increments the reference count
- Calls methods on the interface
- Decrements the reference count
- Exits the apartment

```
// Java  
Point p = new Point();  
p.setX(10);
```

Java-COM Interop Bridge

- **Provides a Java API for ArcObjects**
 - Generates Java classes for every ArcObjects class (Point)
 - Generates Java interfaces for every ArcObjects interface (IPoint)
 - Generates Java Proxy classes for every interface (IPointProxy)
- **Allows Java applications to create ArcObjects and call methods on them using JNI technology**

```
// Java code snippet
EngineInitializer.initializeEngine();
Point p = new Point();
p.setX(10);
```



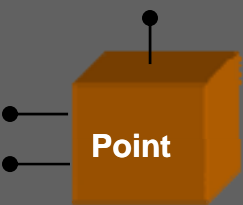
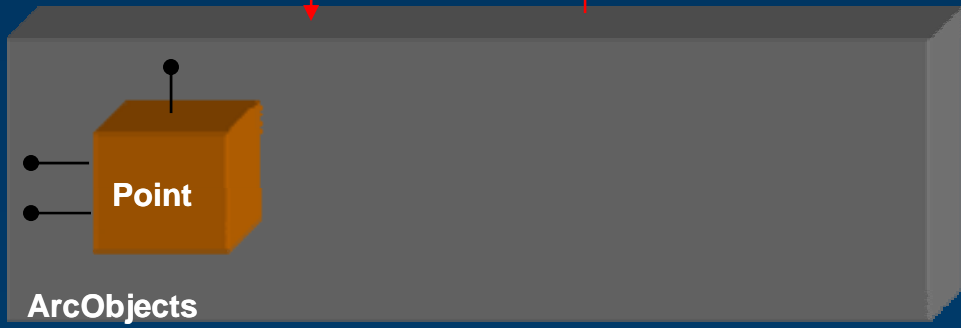
Java



Java-COM Bridge

JNI / C++

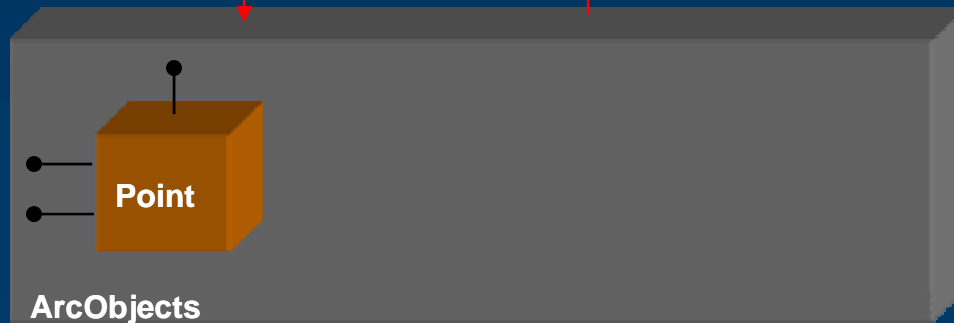
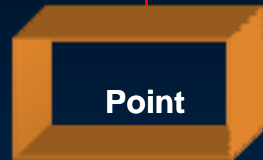
CoCreateInstance(CLSID_Point)



ArcObjects

STA

```
// Java code snippet
EngineInitializer.initializeEngine();
Point p = new Point();
p.setX(10);
```

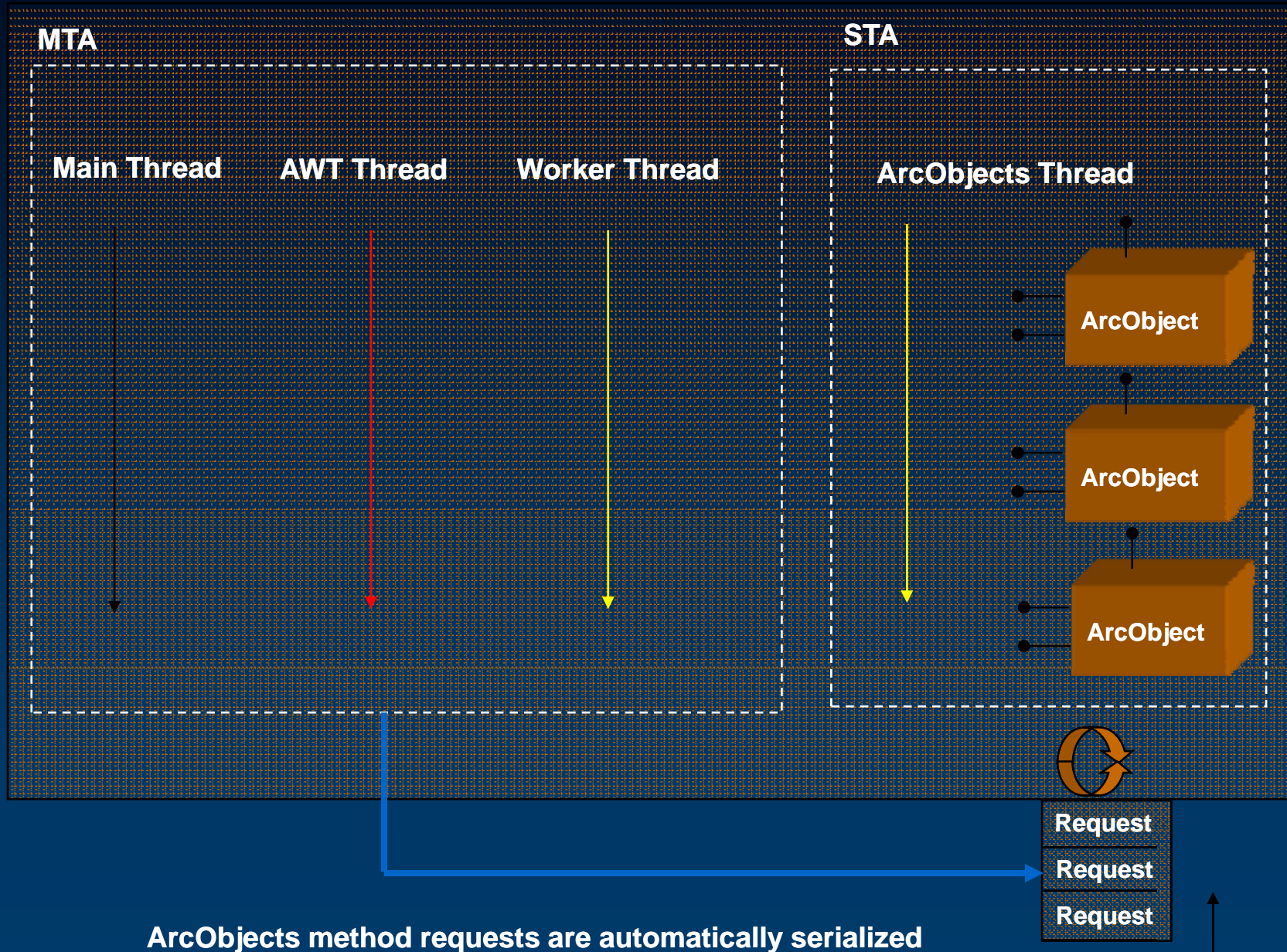


1. Ask for an interface
2. Marshal input arguments
3. Set up call stack
4. Make the method call

Unmarshal output arguments

STA

Architecture of a Visual-Beans application



Agenda

- ArcGIS Engine Architecture
- **Programming with Arcobjects**
 - **IxxxProxy / Casting**
 - **Bridge / GEN Interfaces**
 - **in / out parameters**
 - **value / reference parameters**
 - **AutomationException**
- Geoprocessing with Arcobjects
- ProxyGen
- Back to the future

IxxxProxy classes

- The API contains an IxxxProxy class for every Ixxx interface
- The IxxxProxy class implements the Ixxx interface
- Two-fold usage
 - Perform explicit casts between interfaces
 - Create interface types as return values

```
IGeometry geom = feature.getShape();
```

IFeatureProxy

IGeometryProxy

Java-COM Bridge

IGeometry

IGeometry

IPoint

Point

ArcObjects

STA

IxxxProxy classes...

```
// Obtained the IGeometryProxy
IGeometry geom = feature.getShape();

// Perform an explicit cast to IPoint
IPoint point = new IPointProxy(geom);

// Call a method
point.setX(10);
```

- Java-style casting obviates most of the Proxy classes

Casting/instanceof support

- Java-style casting and *instanceof* support available at 9.2 for 90% of ArcObjects

```
// Getting the shape of a feature
IGeometry geometry = feature.getShape();
if(geometry instanceof IPoint){
    IPoint p = (IPoint)geometry;
}
```

- Applicable to all ArcObjects classes that have an empty constructor
- Use the IxxxProxy classes where casting is not possible

GEN and Bridge Interfaces

- **Certain ArcObjects interface methods take parameters that the Java-COM bridge does not support**
- **GEN Interfaces:**
 - Created to substitute incompatible interfaces
 - Ex. IEnvelopeGEN replaces and deprecates IEnvelope
- **BRIDGE Interfaces:**
 - A collection of incompatible interface methods
 - Ex. IGeometryBridge replaces and deprecates certain methods from IPointCollection, ISegmentCollection, IGeometryCollection

GEN and Bridge Interfaces

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

com.esri.arcgis.geometry

Interface IEnvelope

All Superinterfaces:

[IGeometry](#), java.io.Serializable

All Known Subinterfaces:

[IEnvelope2](#)

All Known Implementing Classes:

[Envelope](#), [IEnvelope2Proxy](#), [IEnvelopeProxy](#)

Deprecated. *This interface uses C style arrays which are not supported in the ArcGIS API for Java. It is replaced by [IEnvelopeGEN](#). You can use the GEN interface proxy's Object-constructor to cast an instance of this interface to its GEN equivalent. Any ArcGIS class that implements this interface also implements the GEN interface.*

```
public interface IEnvelope
extends IGeometry, java.io.Serializable
```

Provides access to methods and properties of envelopes.

Product Availability

Available with ArcGIS Engine, ArcGIS Desktop, and ArcGIS Server.

Method Signatures – multiple outputs

- Some ArcObjects methods expect input parameters to be place holders for return values
 - Indicated as “out” parameter in the Javadoc
 - Passed as a single element array

```
// Gets the next feature ID from the set
FIDSet fidSet = ...
int fID[] = new int[1];
fidSet.next(fid);
```

Method Signatures – multiple outputs...

- Some methods return multiple result values

```
int x           = point.getX(); // [in]
int y           = point.getY(); // [in]
int itemType [] = new int[1];    // [in/out]
IBasicMap basicmap [] = new IBasicMap[1]; // [in/out]
ILayer layer [] = new ILayer[1]; // [in/out]
Object pIndex [] = new Object[1]; // [in/out]
Object other [] = new Object[1]; // [in/out]

// Returns the item clicked on in the TOCControl
toc.hitTest(x, y, itemType, basicmap, layer, pIndex,
            other);
```

Setting properties on ArcObjects

- Two ways to set object references as properties on ArcObjects
 - By value – eg, `ISegmentCollection::setRectangle();`
 - By reference – eg, `IGeometry::setSpatialReferenceByRef();`

```
// Set property by reference
IGeometry geometry;
UnknownCoordinateSystem spRef = new
    UnknownCoordinateSystem();

geom.setSpatialReferenceByRef(spRef);

// DOES affect the state of geometry
spRef.setMResolution(interval);
```

Cleaner.release()

- Explicitly release ArcObjects references
- Java-COM Bridge performs garbage collection most times
 - Relies on JVM's garbage collector

```
// Iterate through features
for(int i=0; i<featureCount; i++)
{
    IFeature feature = cursor.nextFeature();

    ...

    // Release the feature after using it
    Cleaner.release(feature);
}
```

AutomationException

- No Types for exception-events in ArcObjects
- Exceptions are reported as a hexadecimal number – HRESULT
- The Java-COM bridge wraps the HRESULT in an AutomationException class
- Error description is provided when available on the ArcObject. If not...

```
AutomationException: 0x80004005 - Unspecified error  
  at com.esri.arcgis.interop.NativeObjRef.nativeVtblInvoke(Native Method)  
  at com.esri.arcgis.interop.NativeObjRef.a(Unknown Source)  
  at com.esri.arcgis.interop.DispatchVtblInvoke(Unknown Source)
```

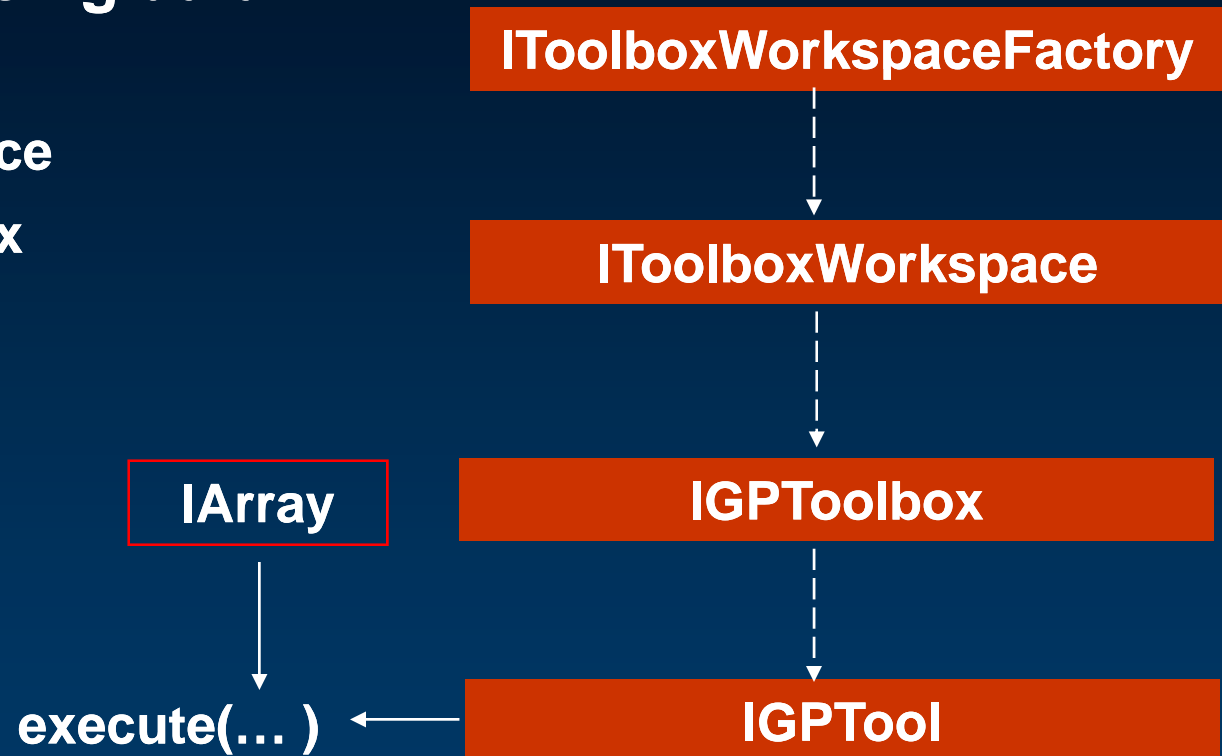
Agenda

- ArcGIS Engine Architecture
- Programming with Arcobjects
- **Geoprocessing with Arcobjects**
 - **Fine-grained API**
 - **Geoprocessor**
 - **Geoprocessor with Tool Wrappers**
 - **Geoprocessing with ArcGIS Server**
 - **Wrappers for custom tools & models**
- ProxyGen
- Back to the future

Fine Grained Geoprocessing API

- **Similar to accessing data**

- create a Factory
- get the Workspace
- open the Toolbox
- get the tool



Fine Grained API

```
//Instantiate the factory
ToolboxWorkspaceFactory factory = new ToolboxWorkspaceFactory();
//Get the workspace
IToolboxWorkspace toolboxWorkspace = factory.openFromFile(C:/Program
  Files/ArcGIS/ArcToolbox/Toolboxes", 0);
//Open the toolbox
IGPToolbox toolbox = toolboxWorkspace.openToolbox("Analysis Tools.tbx");
//Get the tool
IGPTool tool = toolbox.openTool("Clip");

//Set the parameters
Array values = new Array();
GPString clippee = new GPString(); clippee.setValue("C:/mydata/input.shp");
GPString clipper = new GPString(); clipper.setValue("C:/mydata/clipper.shp");
GPString clipped = new GPString(); clipped.setValue("C:/result/output.shp");
values.add(clippee);values.add(clipper);values.add(clipped);

//Execute the tool
tool.execute(values,null,null,null);
```

Geoprocessor

- Façade to the Geoprocessing subsystem
- Preconfigured with tools in system toolboxes
- Add custom and server toolboxes

“ToolName_ToolboxAlias”, IVariantArray

execute(...)

Geoprocessor

Geoprocessing : Geoprocessor

```
//Set tool parameters in the correct sequence
VarArray values = new VarArray();
values.add("C:/mydata/input.shp");
//Set any optional parameters as null
//values.add(null);

values.add("c:/mydata/clipper.shp");
values.add("c:/result/output.shp");

//Instantiate the Geoprocessor and execute
Geoprocessor geoprocessor = new Geoprocessor();
IGeoprocessorResult result =
    geoprocessor.execute("Clip_analysis",values,null);
```

Geoprocessing : Tool Wrappers

- A Wrapper for every standard Tool
- Wrappers have a Type
- Wrappers know about Tool parameters
- Wrappers extend AbstractGPTool
- Create Wrappers for custom Tools and Models



Geoprocessing : Tool Wrappers

```
import com.esri.arcgis.geoprocessing.tools.analysis.tools;  
  
...  
  
//Instantiate the Clip tool wrapper  
Clip clipTool = new Clip();  
  
//Set tool parameters  
clipTool.setInFeatures("C:/mydata/input.shp");  
clipTool.setClipFeatures("c:/mydata/clipper.shp");  
clipTool.setOutputFeatureClass("c:/result/output.shp");  
  
//Instantiate the Geoprocessor and execute  
Geoprocessor geoprocessor = new Geoprocessor();  
IGeoprocessorResult result =  
    geoprocessor.execute(clipTool,null);
```

DEMO : Creating hillshade

Putting it in perspective

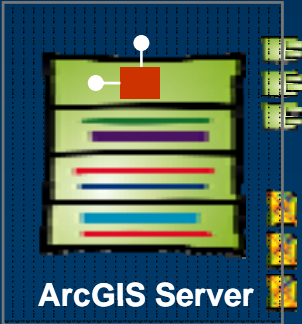
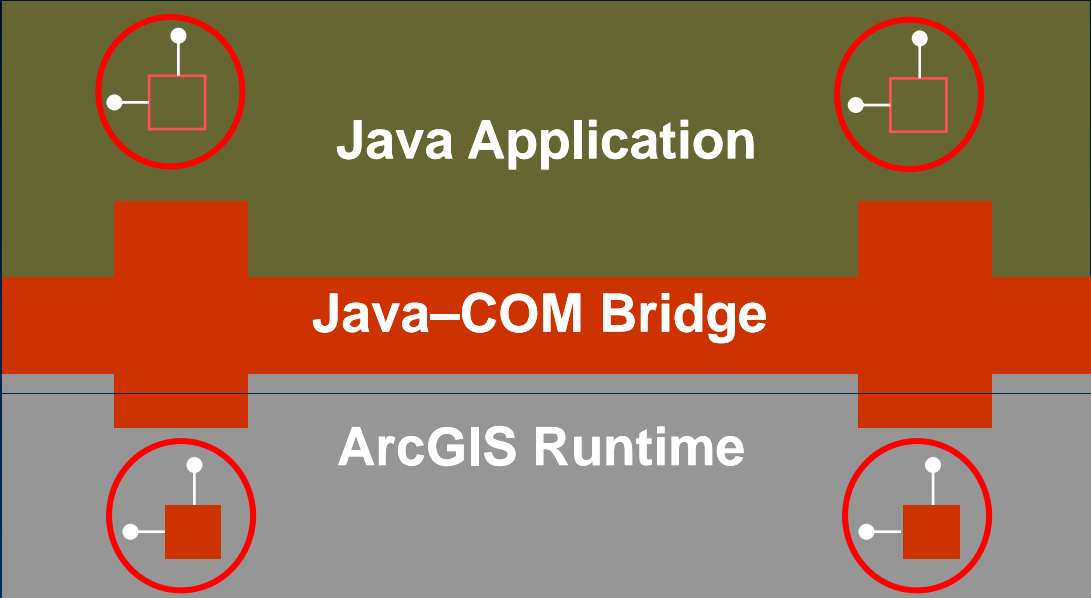
- **Geoprocessor makes code compact**
- **Tool Wrappers make setting parameters easy**

Tool Parameters

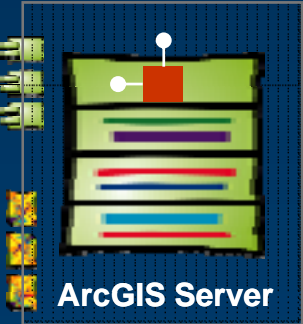
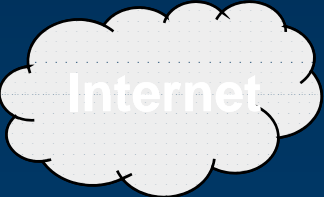
- **Strings**
- **ArcObjects**
- **IGPValue objects**
 - **DE*** : represent data
 - **GP*** : represent other information
- **GPUutilities**

Geoprocessing : with ArcGIS Server

- **Synchronous / Asynchronous execution of Tools**
- **Cannot use Tool Wrappers**
- **Using the Geoprocessor**
 - **Add server toolbox**
- **Using GPServer**



LAN



Using Geoprocessor with ArcGIS Server

```
//Instantiate the geoprocessor
Geoprocessor gp = new Geoprocessor();

//Add the server toolbox
gp.addToolbox("server;folder/service"); //Local
gp.addToolbox("http://server:8399/arcgis/services;folder/service") //Internet

VarArray array = ... //Tool Parameters

IGeoprocessorResult result =
    gp.execute("toolname_tbxalias",array);
```

Using GPServer with ArcGIS Server

```
AGSServerConnectionFactory factory =  
    new AGSServerConnectionFactory();  
IAGSServerConnection connection =  
    factory.open(props, 0);  
IAGSEnumServerObjectName enumName =  
    connection.getServerObjectNames();  
IAGSServerObjectName name = enumName.next();  
IGPServer gpServer = (IGPServer)((IName)name).open();  
  
GPValues values = new GPValues();  
...  
IGPResult result =  
    gpServer.execute("toolname_tbxalias", values, null);
```

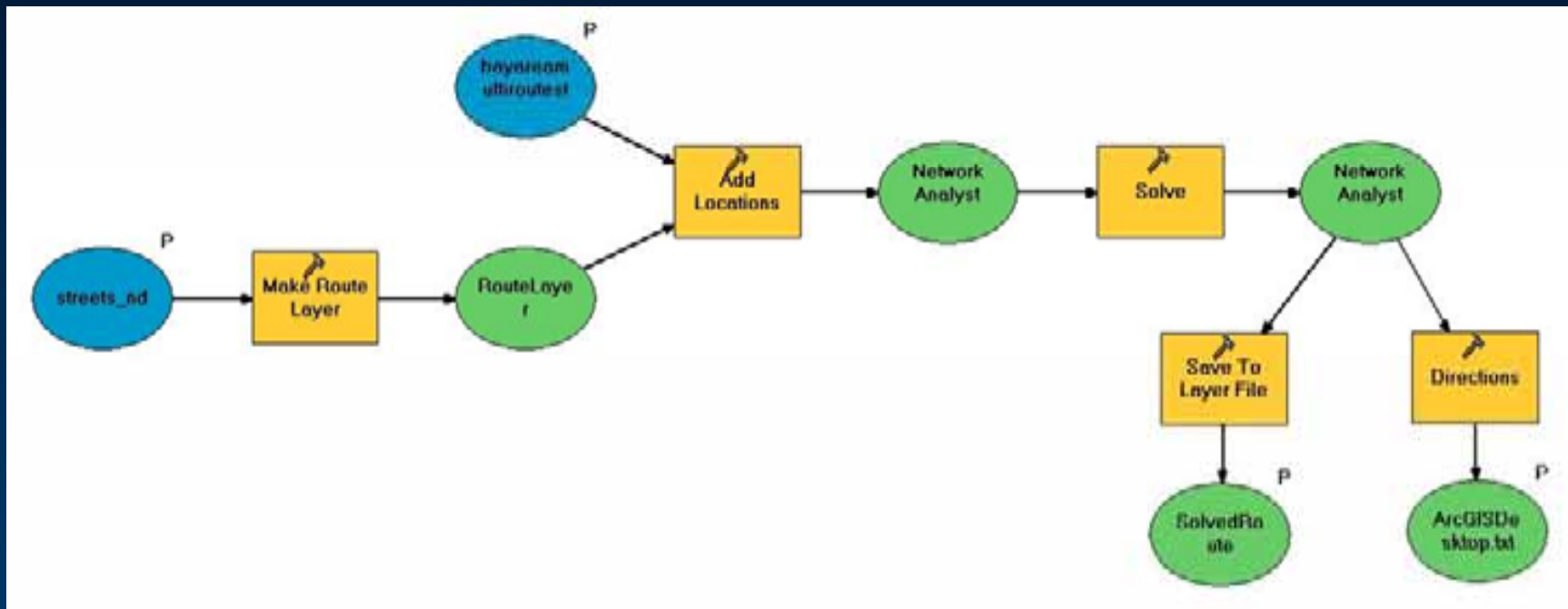
Using Asynchronous ArcGIS Services

- **Geoprocessor::execute()**
 - Check the IGPResultInfo for status of job
 - Geoprocessor::makeResult(String jobid)
- **GPServer::submitJob()**
 - GPServer::getJobResult()

Custom Tools and Models

- **Author tools, scripts and models in ArcGIS Desktop**
- **Add them to a custom toolbox**
- **Generate Java Wrappers**
 - Eclipse ArcGIS plugin
 - Command line
- **Invoke from Java Application**

Demo : Invoke custom routing model from Java



Agenda

- ArcGIS Engine Architecture
- Programming with Arcobjects
- Geoprocessing with Arcobjects
- **ProxyGen**
- Back to the future

Plug-in Data Sources

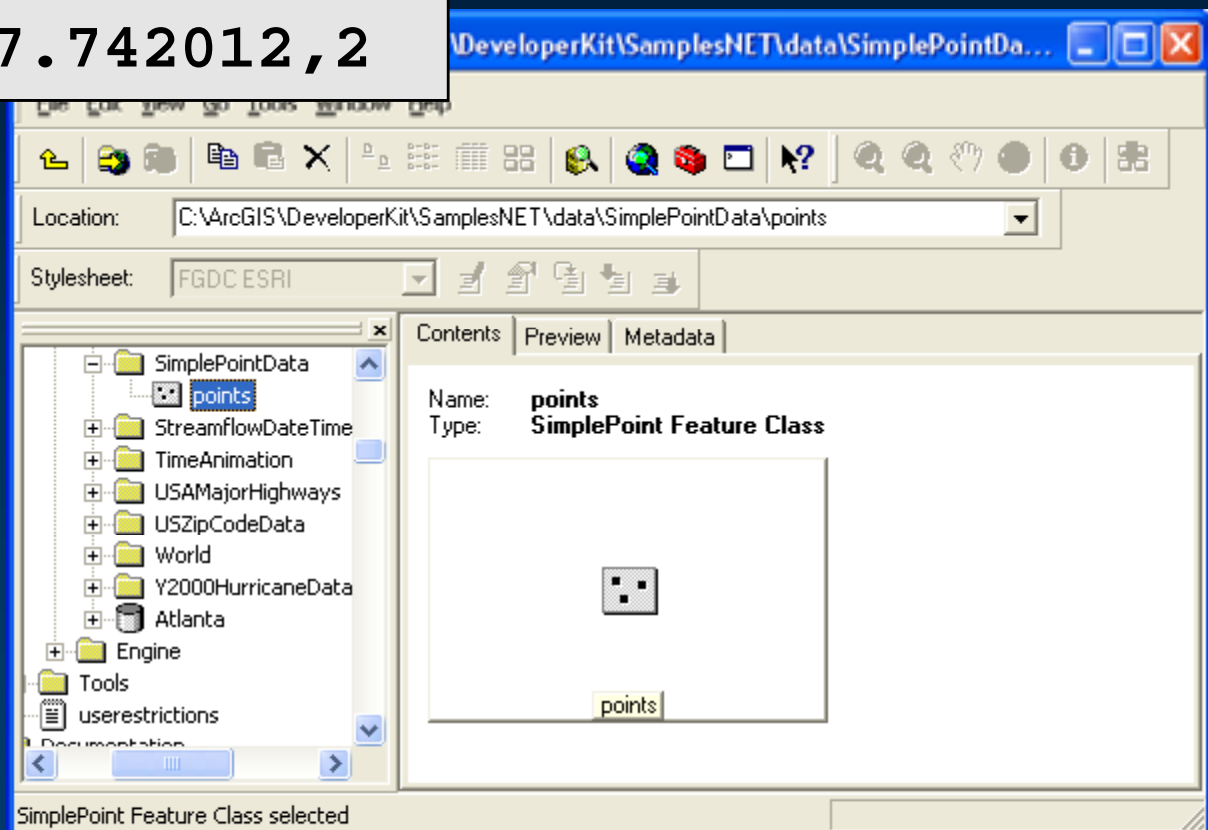
- **Provide ArcGIS support for proprietary data format**
- **Integrates new data format completely into ArcGIS**
 - Render, select, query, label and join
- **Use standard geodatabase API**
 - IWorkspace and IFeatureClass
- **Supports tables, feature classes and featureDatasets**
- **In Java, we have limited extensibility of the API, yet, so we will use legacy code for creating a plug-in data source**

Extending ArcObjects - ProxyGen

- **Extend ArcObjects using custom COM components and access them from Java**
- **When is this useful?**
 - Extend ArcObjects (Custom layers, plug-in data sources...)
 - Integrate legacy or third party COM libraries
 - Improve performance - replace many fine-grained calls with a single call to a coarse grained COM component

SimplePoint Plug-in Datasource

```
# x-coord, y-coord, seq  
-122.492587, 37.781514, 1  
-122.497241, 37.742012, 2
```



Extending ArcObjects - ProxyGen

- **Procedure:**

- Build custom COM component using C++ or .Net
- Run ProxyGen tool on type library to generate java proxies
- Use the proxies in your java code

- **ProxyGen Usage:**

- Specify type library, java package name, and output directory in mapping.txt
- Run from cmd.exe: *ProxyGen mapping.txt*
- Integrate the generated proxies in your java project

Agenda

- ArcGIS Engine Architecture
- Programming with Arcobjects
- Geoprocessing with Arcobjects
- ProxyGen
- **Back to the future**

Back to the future

- **Extending the ArcGIS framework using Java**
 - Custom commands and tools for ArcMap
 - Custom GP tools
 - Server Object Extensions
 - Geodatabase Extensions
 - Custom Layers, Renderers and Features

Presentation materials

- PowerPoint presentation and code are posted on the conference web site
 - <http://www.esri.com/events/devsummit/index.html>
- EDN – downloads and videos

Further questions?

- **TECH-TALK AREA 1**
 - in the Community Center (Oasis 3)
 - during the next 30 minutes
- **ESRI Developers Network (EDN) website**
 - <http://edn.esri.com>

Thank you for coming!
See you next year