



Working with the ArcGIS Server Web Services API

Sud Menon, Julio Andrade

Presentation outline

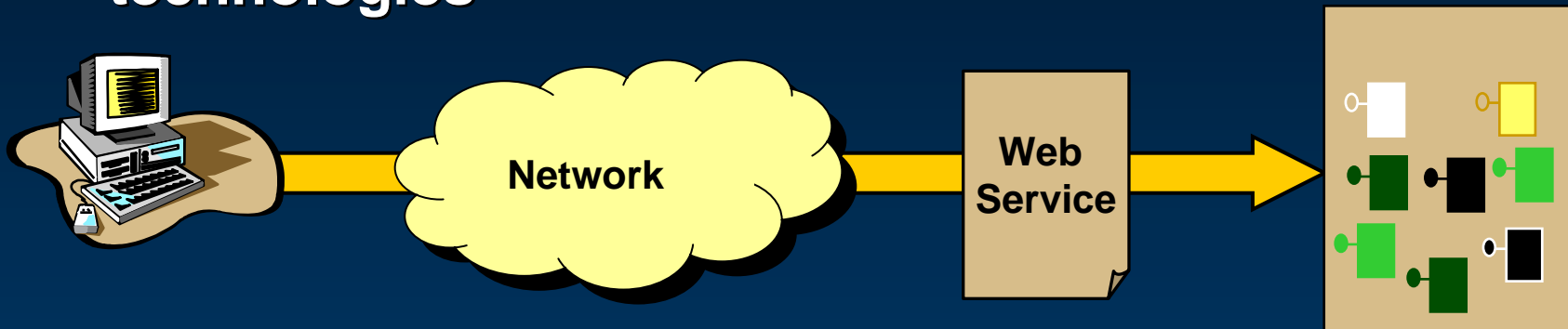
- **Brief overview of ArcGIS Server GIS Web Services**
- **Description of 9.2 GIS Web Services**
 - Functional capabilities of each service
 - The SOAP API for the service
 - Sample applications that consume the service
- **Under the Hood**
 - Implementation details of interest to you

Our assumptions

- You have a high level understanding of :
 - ArcGIS Server
 - The ArcGIS Information Model
 - Web services
- You can read C# code

What Are Web services?

- **Network accessible interface or application functionality, built using standard Internet technologies**



- **Web service technology stack**

Discovery	UDDI
Description	WSDL
XML Messaging	SOAP,XML
Transport	HTTP,SMTP,FTP

Services can be used to Answer Questions

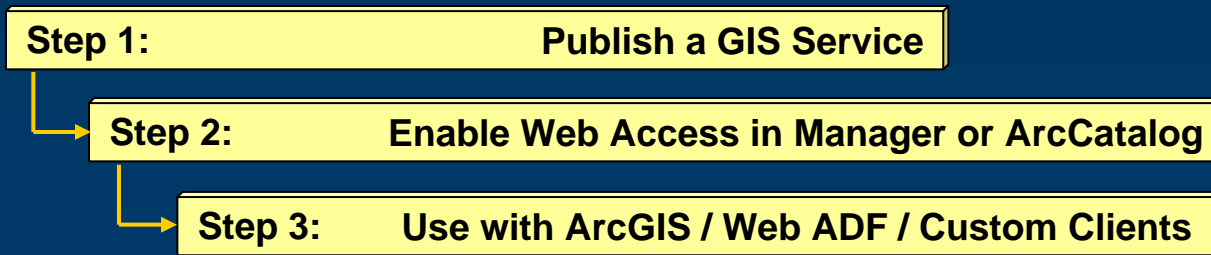
The image displays three overlapping web service dialog boxes. The top-left box is titled "Driving Directions" and contains fields for "Country" (United States), "Starting location" (1495 Easton Rd, Warrington, PA 18976), and "End Location" (28955 Live Oak Canyon Rd, Redlands). It also includes options for "Return the distances in" (Miles) and "Route options" (Fastest), along with a "Get Directions" button and a link to "Tips on getting directions". The top-right box is titled "Find Place" and has a text input field containing "Mammoth Cave, Kentucky", a "Country (optional)" dropdown menu set to "<Not Specified>", a "Place finding tips" link, and a "Find" button. The bottom box is titled "Find Address" and features a "Country" dropdown menu set to "United States", a "Street Address or Intersection" text input field with "1495 Easton Road", and three separate input fields for "City" (Warrington), "State" (PA), and "ZIP Code" (18976). It also includes an "Address finding tips" link and a "Find" button.

GIS Web Services

- **ArcGIS Server includes a rich set of out of the box GIS Web Services**
- **SOAP**
 - 2D Map Service
 - 3D Map / Globe Service
 - Geocoding Service
 - Network Analysis Service
 - Geoprocessing Service
 - Geodata Service
 - Mobile Data Service
- **OGC**
 - WMS Service
- **KML**
 - KML Service
- **Publish and Deploy using Manager**
 - no programming required

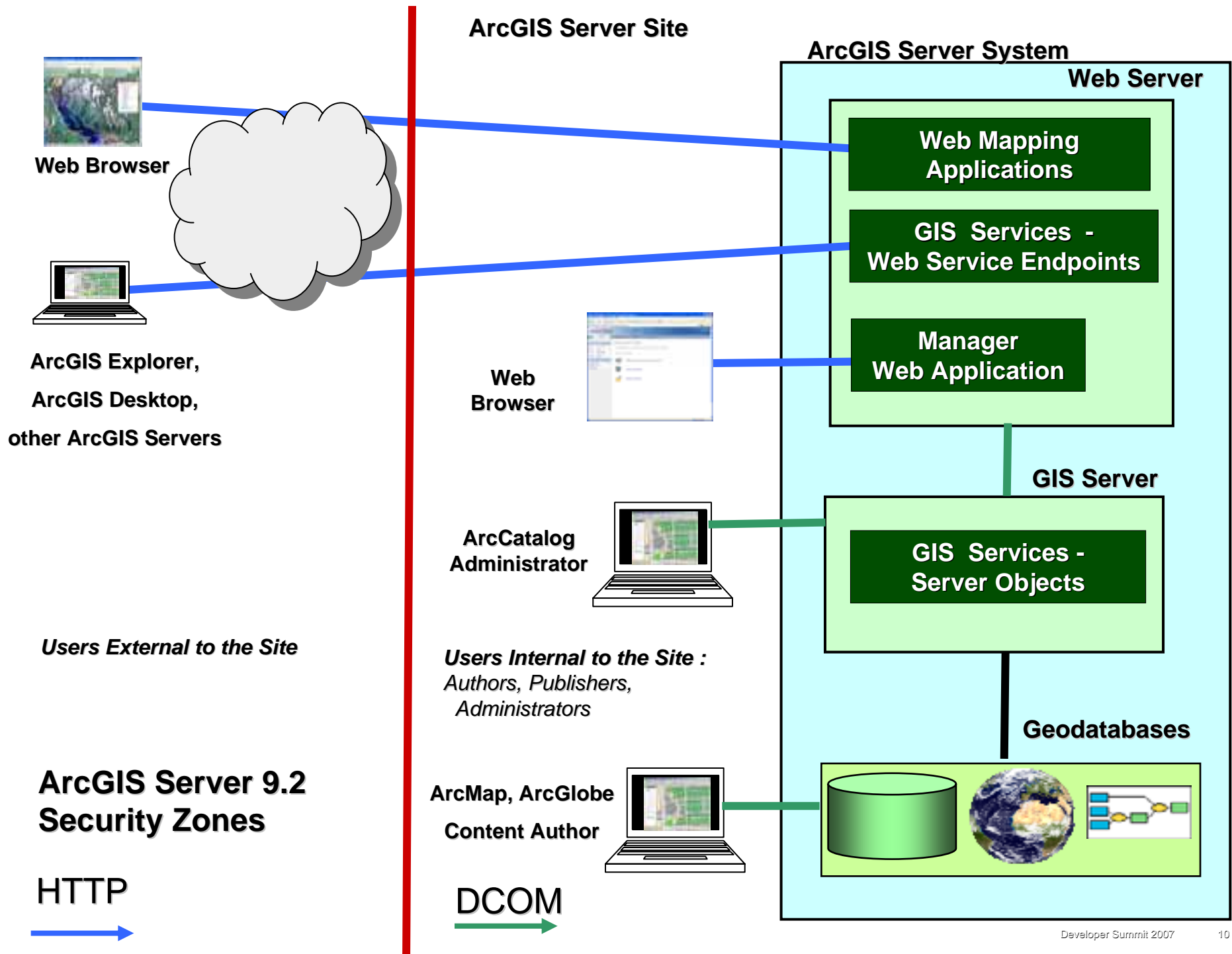
ArcGIS Server – Publishing GIS Web Services

- Author GIS resources (eg maps, locators) using desktop.
- Publish GIS resources to create GIS Services.
- A GIS Service may have multiple capabilities
 - Eg Mapping, Network Analysis
- Each capability is exposed to consumers as a GIS Web Service accessible over HTTP



ArcGIS Server GIS Web Services - Out of the Box Clients

- ArcGIS Explorer and ArcGIS Desktop are out of the box Desktop Clients
- You can use Manager wizards to build Web Mapping Applications that consume GIS Web Services
 - no programming required

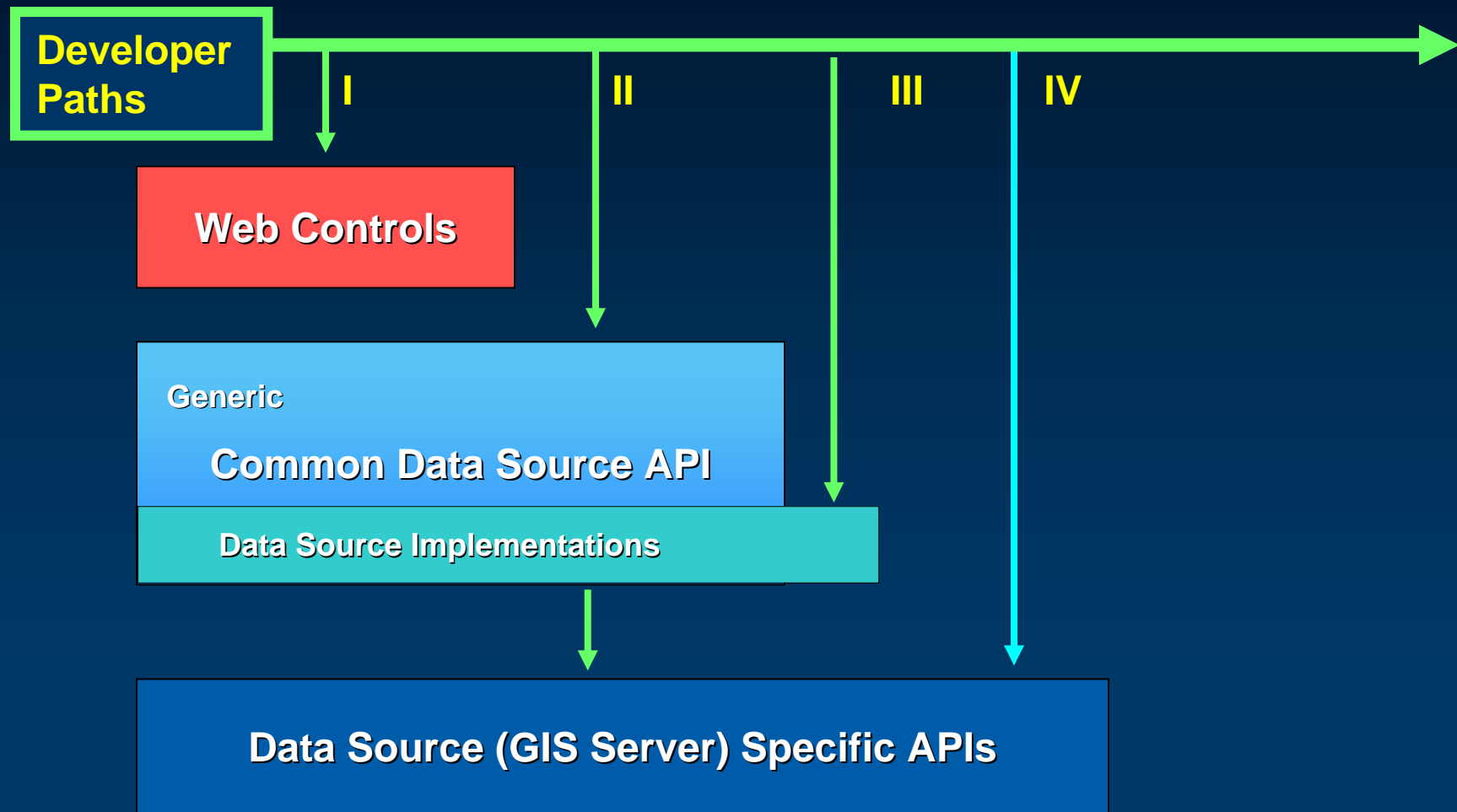


Options for Developing Custom Client Applications that use GIS Web Services

- **Work with the Web ADF object model (for web apps)**
 - Work with the Web ADF common object model and controls
 - Use the ArcGIS Server Internet data source
- **Work with ArcGIS Engine (for desktop apps)**
 - Use the ArcEngine object model and controls
 - Use the GIS Client ArcObjects library and built-in ArcObjects proxies for each web service
- **Work directly with the web service api**
 - The focus of this presentation
 - You can use pre-generated proxies that ship with the .Net and Java Web ADFs, eg `ESRI.ArcGIS.ADF.ArcGISServer`
 - You can generate the proxies yourself using your favourite wsdl tool

Web ADF Development Paths

- This talk focuses on path IV

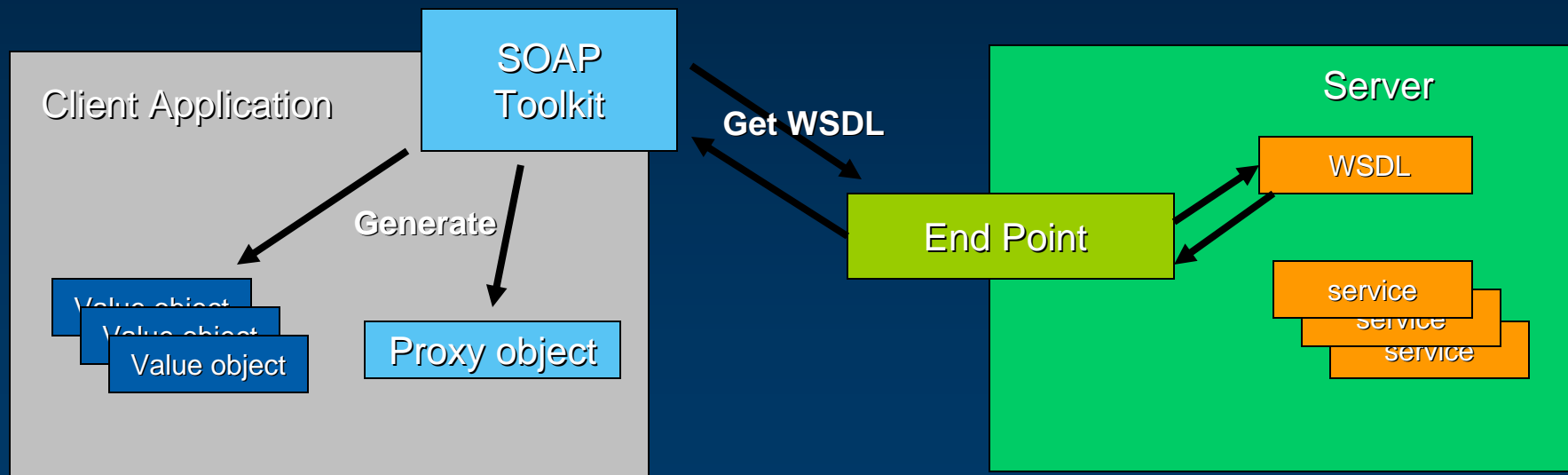


The ArcGIS Server Web Services API

- **Coarse Grained**
- **Stateless**
- **Described by a wsdl**
- **Methods take serializable Value objects (simple structures) as parameters**
- **Value Objects are based on existing well known ArcObjects types**
 - **Eg Color, Symbol, Graphic Element, Geometries, DataElements**
- **Straightforward**
- **Powerful**

Working with a SOAP service

- You work with Proxies and Value objects
 - Generated from a WSDL using a SOAP toolkit



ArcGIS Server : Web Service URLs

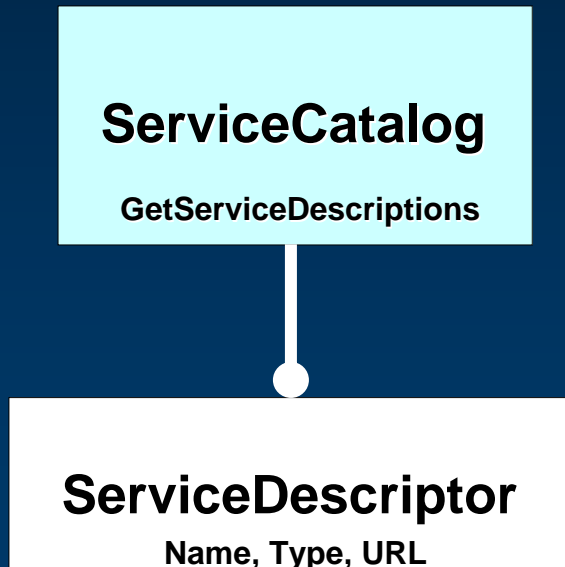
- A standard scheme for all service URLs :
- `http://<web server hostname>/<arcgis instance>/services/<folder>/<servicename>/<servicetype>`
- <http://gis.mybiz.com/arcgis/services/usa/mapserver>
- <http://gis.mybiz.com/arcgis/services/northernca/SanFranciscoLocator/geocodeserver>
- <http://gis.mybiz.com/arcgis/services/usa/mapserver/naserver>
- To get the wsdl :
- <http://gis.mybiz.com/arcgis/services/usa/mapserver?wsdl>

Discovering Services – The Service Catalog Web Service

Every ArcGIS Server has a Service Catalog Web Service :

`http://<web server hostname>/<arcgis instance>/services?wsdl`

- What Services do you have ?



```
WSDL Method
<xs:element name="GetServiceDescriptions">
  <xs:complexType/>
</xs:element>

Response
<xs:element name="GetServiceDescriptionsResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ServiceDescriptions"
        type="ArrayOfServiceDescription"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

Value Object
<xs:complexType name="ArrayOfServiceDescription">
  <xs:sequence>
    <xs:element name="ServiceDescription"
      type="ServiceDescription"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

Value Object
<xs:complexType name="ServiceDescription">
  <xs:sequence>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Type" type="xs:string"/>
    <xs:element name="Url" type="xs:string"/>
  </xs:sequence>
</xs:complexType>
```

Map Service

- **Based on a Map Document (.mxd) authored using ArcMap**
- **Can be used to generate map images in a variety of image formats**
- **Callers can supply additional information to be overlaid on the map**
 - **Graphics**
 - **Selection IDs and buffer distances**
- **Can be used to identify features**
- **Can be used to query features**
 - **Results returned as Geodatabase record sets**
 - **Easily marshaled by value**
 - **Can be converted to ADO.NET datasets**

Map Service

- **Maps can be pre-rendered and cached for performance resulting in a tiled map service that has a fixed set of scales**
- **Methods to discover if**
 - **A map is a tiled fixed scale map**
 - **If the map tiles correspond to single fused images or if there are separate tiles for each layer**
- **Methods to get the tiling scheme**
- **Methods to fetch tiles**
- **Tiles can also be obtained directly via HTTP Get.**

Working with the Map Service

```
// Drawing the map - generating map images

MapImage mi = ms.ExportMapImage (mapDescription, imageDescription);

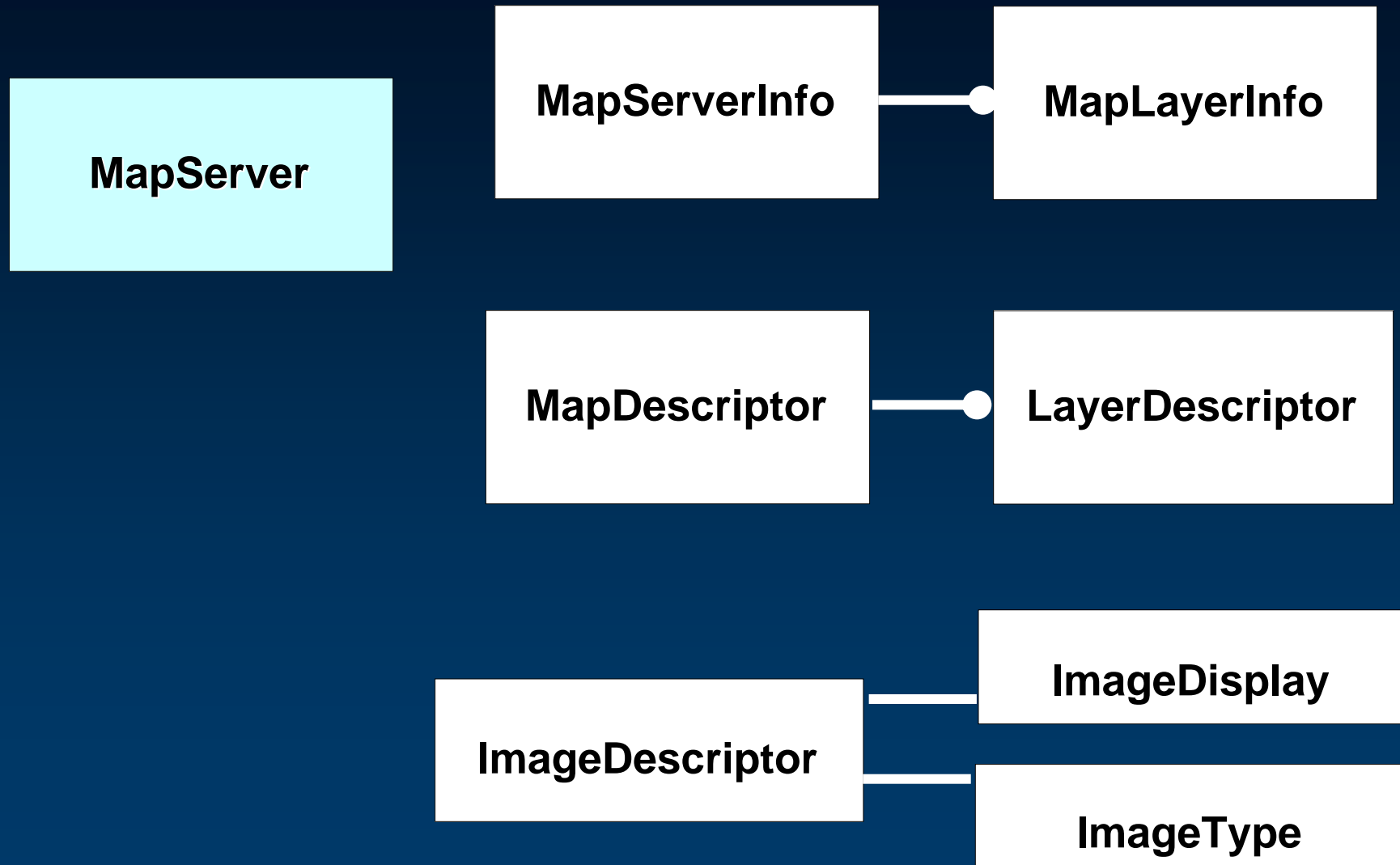
// Querying the map

Recordset rs = ms.QueryFeatureData(mapName, layerID, queryFilter);

FidSet fs = ms.QueryFeatureIDs(mapName, layerID, queryFilter);

MapServerFindResult[] results
    = ms.Find(mapDescription, mapImageDisplay, searchString,
              expression, searchFields, findOption, layerIDs);
```

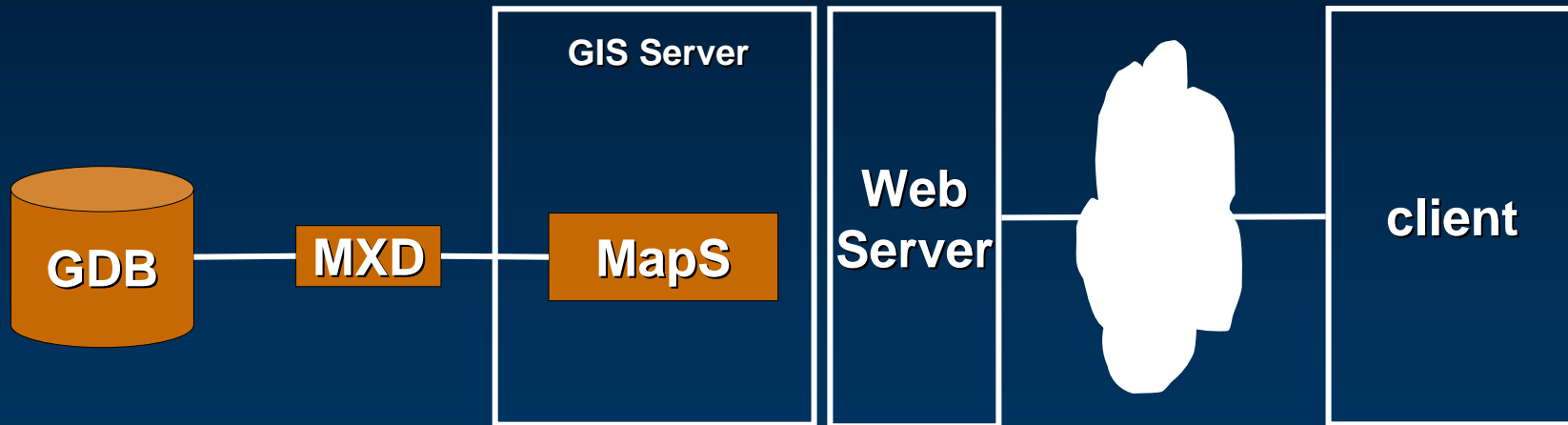
Map Service Object Model - Drawing



Map Service Methods

MapCount	No of maps in the service
MapName	Name of a map, by index
GetServerInfo	Retrieves Information about a specified map
ExportMapImage	Generates an image based on a specified map description
GetLegendInfo	Returns legend information for specified layers
Identify	Returns a collection of Identify Results
Find	Returns a collection of Find Results
QueryFeatureData	Returns a recordset of features based on a query filter
FromMapPoints	Converts from map to screen/image coordinates
ToMapPoints	Converts from screen/image to map coordinates

Map Service demo



Drawing a Map

```
MapServerWS.ImageType it = new MapServerWS.ImageType();

it.ImageFormat = MapServerWS.esriImageFormat.esriImageJPG;
it.ImageReturnType = MapServerWS.esriImageReturnType.esriImageReturnMimeData;

idisp = new MapServerWS.ImageDisplay();
idisp.ImageHeight = 400;
idisp.ImageWidth = 552;
idisp.ImageDPI = 150;

MapServerWS.ImageDescription pID = new MapServerWS.ImageDescription();
pID.ImageDisplay = idisp; pID.ImageType = it;

MapServerWS.MapImage pMI = ms.ExportMapImage(pMapDescription, pID);

System.IO.Stream pStream = new
System.IO.MemoryStream((byte[])pMI.ImageData);
pImage = Image.FromStream(pStream);

pictureBox1.Image = pImage;

pictureBox1.Refresh();
```

MapServerInfo Object :

Information on the map as published.

Name	The name of the map
MapLayerInfos	Information on the layers in the map
DefaultMapDescription	The DefaultMapDescription - as published
FullExtent	The full extent of the map
Bookmarks	The published bookmarks

MapLayerInfo Object :

Information on a layer, as published.

Name	The name of the layer
ID	The ID for the layer
IsComposite	True if a composite layer
FullExtent	The full extent of the map
ParentLayerID	ID of parent layer, if child of a composite layer
Sublayers	IDs of sublayers if a composite layer
Fields	The fields for the layer
Extent	The extent for the layer
MinScale	The minimum visible scale for the layer
MaxScale	The maximum visible scale for the layer

MapDescription Object :

Properties used to control the drawing of the map

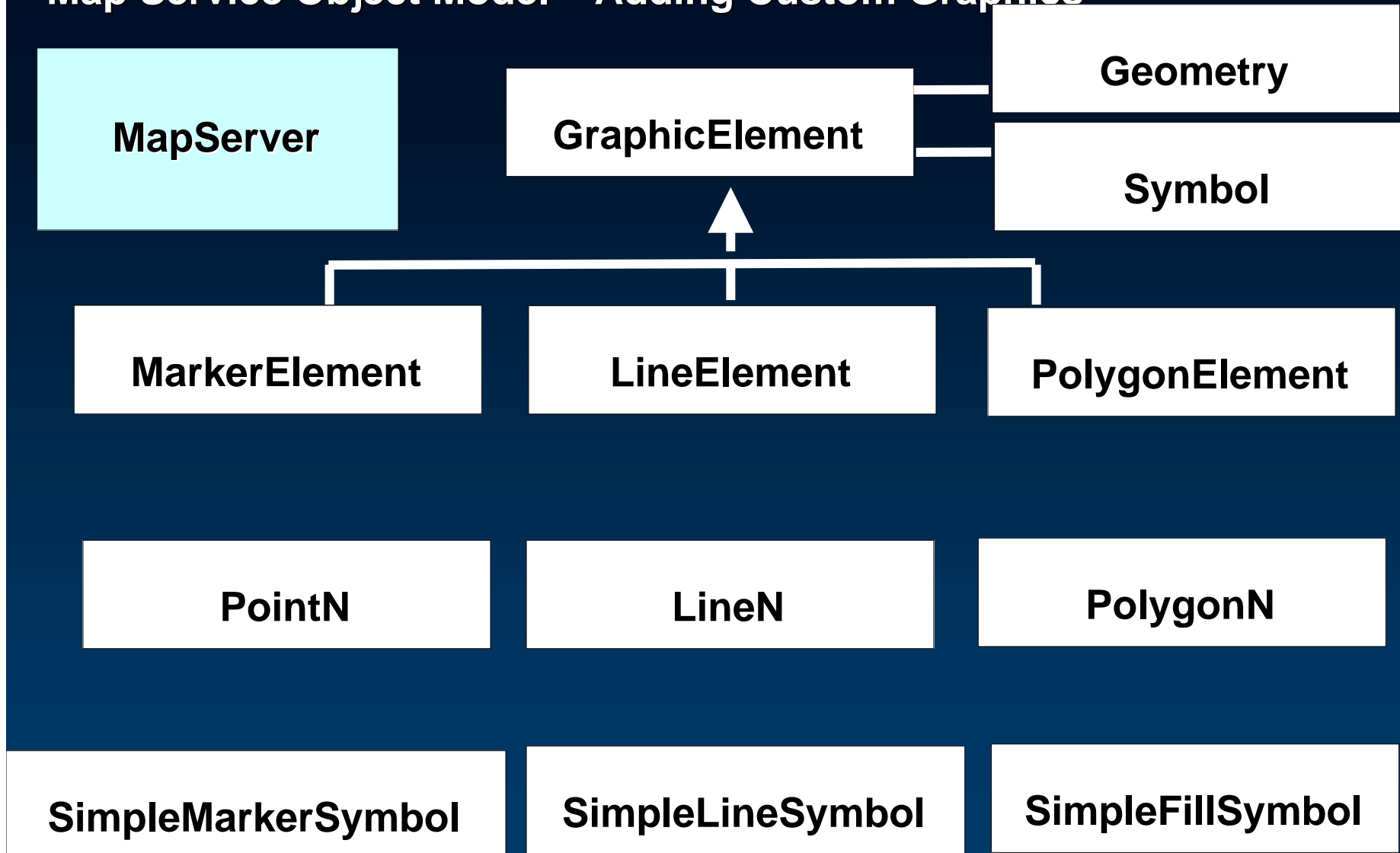
Name	The name of the map
LayerDescriptions	A collection of layer descriptions
CustomGraphics	CustomGraphics to be drawn on the map
SpatialReference	The spatial reference in which to draw the map
TransparentColor	The color that is transparent

LayerDescription Object :

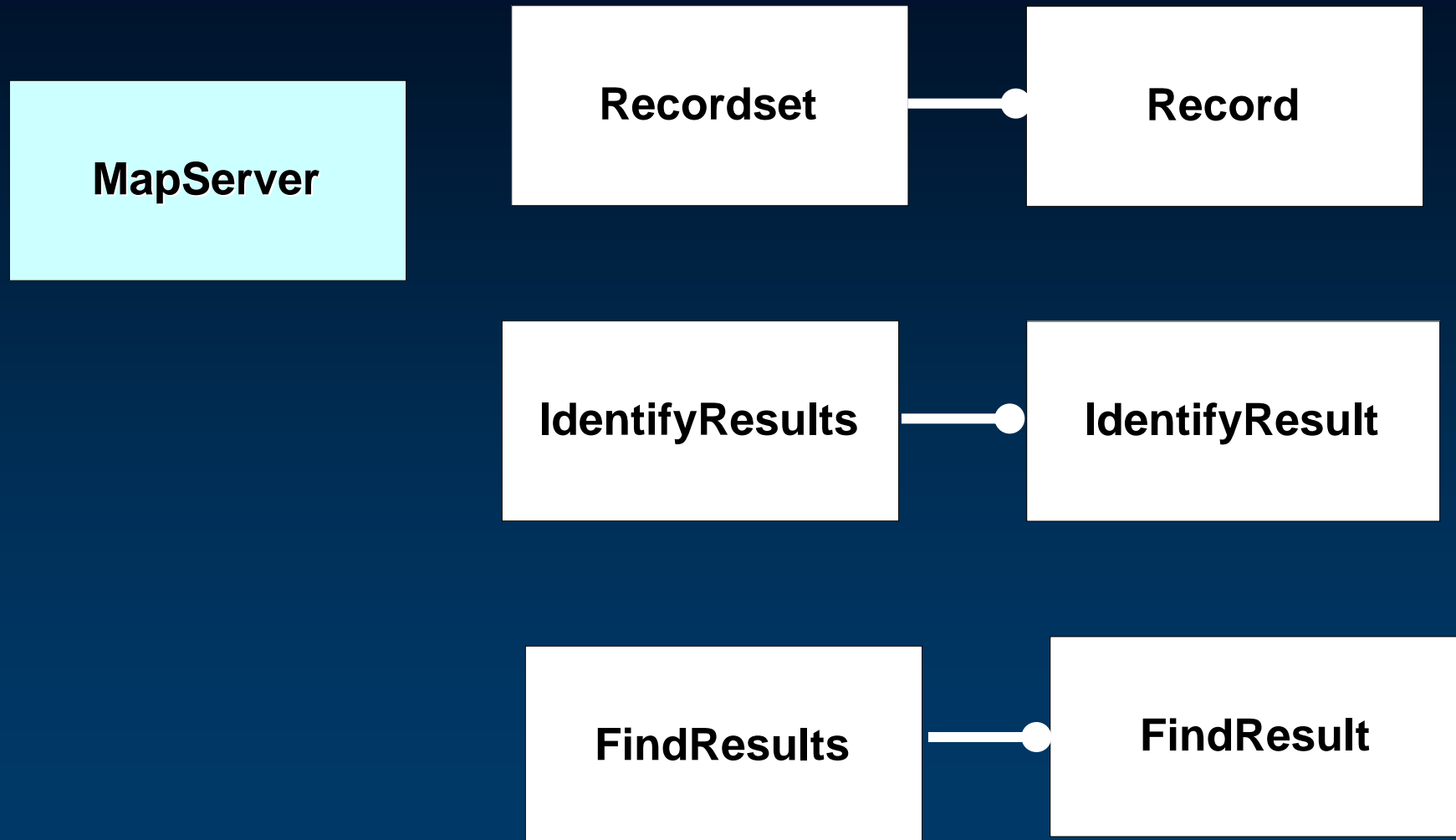
Properties used to control the drawing of a layer

LayerID	The ID for the layer
DefinitionExpression	The definition expression for the layer
Visible	Is the Layer Visible
ShowLabels	Should labels be drawn
SelectionFeatures	Selected feature ids
SelectionColor	Color to be used for selected features

Map Service Object Model – Adding Custom Graphics

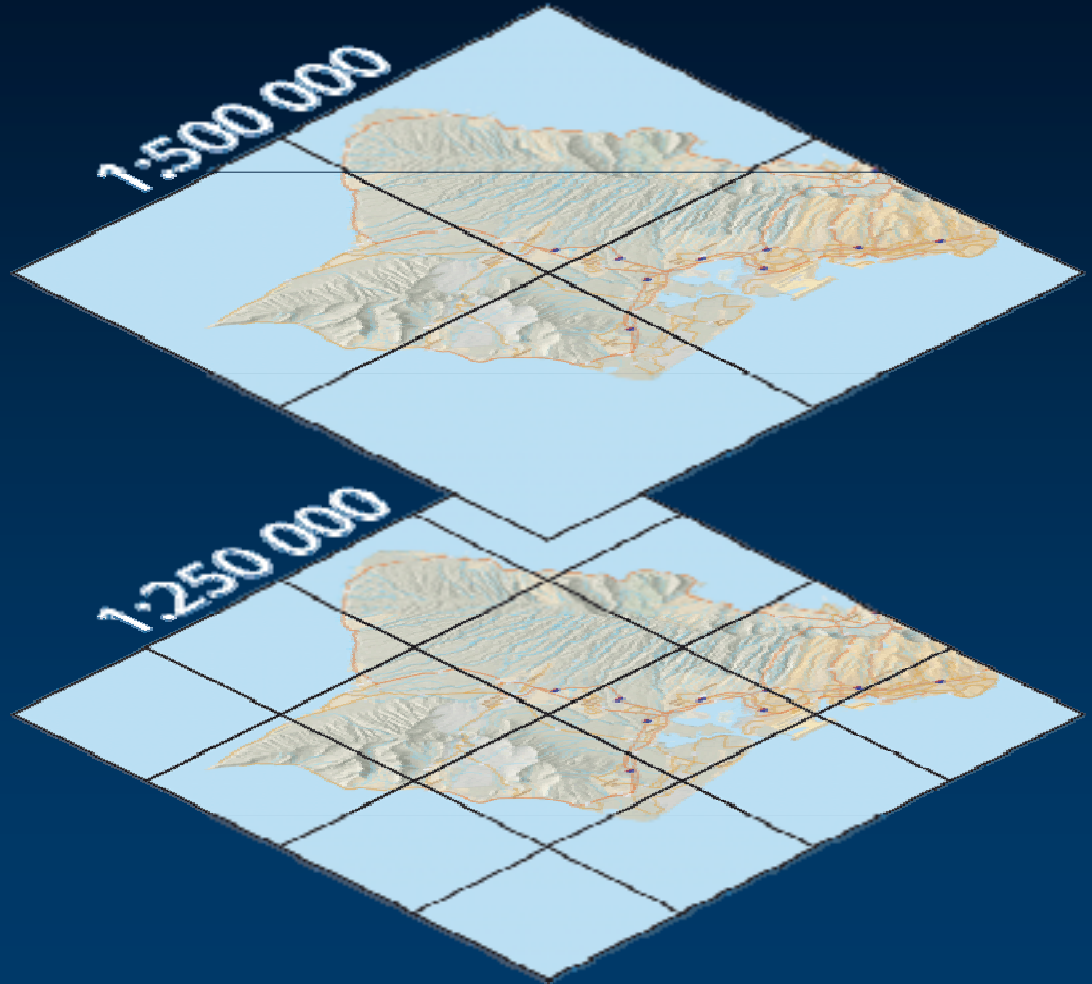


Map Service Object Model – Querying Features



Cached Map Services

- Tiles pre-rendered at fixed scales
- Rapid display of static base maps
- Richer symbols and more information



Map Service Object Model – Tiled Map Services



Map Service - Methods for working with tiled map services

<code>IsFixedScaleMap</code>	Indicates that the map has a fixed set of LODs
<code>HasSingleFusedMapCache</code>	Indicates that the map has a single cache that prefetches all layers
<code>GetTileCacheInfo</code>	Gets the tiling scheme for the map
<code>GetMapTile</code>	Gets the specified tile for the map given tile <i>level</i> , <i>row</i> and <i>column</i> coordinates
<code>GetVirtualCacheDirectory</code>	Gets the virtual cache directory for the map, if present

Tile Cache Info Object :

Properties that describe a tiling scheme

LODInfos	The levels of detail in the cache {id, scale, resolution}
Origin	The origin for the tiling grid
TileCols	The number of columns in a tile
TileRows	The number of rows in a tile
DPI	Resolution in dots per inch
SpatialReference	The spatial reference of the cache

```
tileWidth = tileCols * LODResolution  
  
tileColumn = floor ((x - xorig) / tileWidth)  
  
tileHeight = tileRows * LODResolution  
  
tileRow = floor ((y - yorig) / tileHeigh)
```

ArcGIS Server - GIS Web Services and ArcObjects

- Each ArcGIS Server web service has a corresponding coarse grained ArcObject (a Server Object) which supports a logically equivalent ArcObjects interface

Web Service	ArcObjects - ServerObject	ArcObjects - Interface
Map Service	MapServer	IMapServer
Geoprocessing Service	GPServer	IGPServer
Network Analysis Service	NAServer	INAServer
...

Drawing a map – equivalent ArcObject model

MapServer

IMapServer : IUnknown	
■	DefaultMapName: String
■	DocumentInfo: IPropertySet
■	MapCount: Long
■	MapName (in Index: Long): String
←	ComputeDistance (in MapName: String, in fromPoint: IPoint, in toPoint: IPoint, in Units: esriUnits): Double
←	ComputeScale (in mapDesc: IMapDescription, in mapDisplay: IImageDisplay): Double
←	ExportMapImage (in mapDesc: IMapDescription, in imageDesc: IImageDescription): IMapImage
←	Find (in mapDesc: IMapDescription, in MapImage: IImageDisplay, in searchString: String, in Contains: Boolean, in searchFields: String, in option: esriFindOption, in layerIds: ILongArray): IMapServerFindResults
←	FromMapPoints (in MapDescription: IMapDescription, in mapDisplay: IImageDisplay, in mapPoints: IPointCollection, screenXValues: ILongArray, screenYValues: ILongArray)
←	GetLegendInfo (in MapName: String, in layerIds: ILongArray, in patch: IMapServerLegendPatch, in imgType: IImageType): IMapServerLegendInfos
←	GetServerInfo (in MapName: String): IMapServerInfo
←	GetSQLSyntaxInfo (in MapName: String, in LayerID: Long): ISQLSyntaxInfo
←	GetSupportedImageReturnTypes: esriImageReturnType
←	Identify (in mapDesc: IMapDescription, in MapImage: IImageDisplay, in searchShape: IGeometry, in Tolerance: Long, in option: esriIdentifyOption, in layerIds: ILongArray): IMapServerIdentifyResults
←	QueryFeatureCount (in MapName: String, in LayerID: Long, in Filter: IQueryFilter): Long
←	QueryFeatureData (in MapName: String, in LayerID: Long, in Filter: IQueryFilter): IRecordSet
←	QueryFeatureIDs (in MapName: String, in LayerID: Long, in Filter: IQueryFilter): IFIDSet
←	QueryHyperlinks (in mapDesc: IMapDescription, in MapImage: IImageDisplay, in layerIds: ILongArray): IMapServerHyperlinks
←	ToMapPoints (in MapDescription: IMapDescription, in mapDisplay: IImageDisplay, in screenXValues: ILongArray, in screenYValues: ILongArray): IPointCollection

MapDescription

IMapDescription2 : IUnknown	
■	BackgroundSymbol: IFillSymbol
■	CustomGraphics: IGraphicElements
■	LayerDescriptions: ILayerDescriptions
■	MapArea: IMapArea
■	MapRotation: Double
■	Name: String
■	SelectionColor: IColor
■	SpatialReference: ISpatialReference
■	TransparentColor: IColor

MapArea

IMapArea : IUnknown	
■	Extent: IEnvelope

MapExtent

IMapExtent : IUnknown	
■	Extent: IEnvelope

ImageDescription

IImageDescription : IUnknown	
■	Display: IImageDisplay
■	Type: IImageType

LayerDescription

ILayerDescription : IUnknown	
■	DefinitionExpression: String
■	ID: Long
■	ScaleSymbols: Boolean
■	SelectionBufferDistance: Double
■	SelectionColor: IColor
■	SelectionFeatures: IFIDSet
■	SelectionSymbol: ISymbol
■	SetSelectionSymbol: Boolean
■	ShowLabels: Boolean
■	ShowSelectionBuffer: Boolean
■	Visible: Boolean

ILayerDescription2 : ILayerDescription

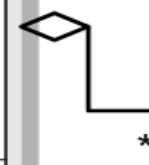
■	SelectionBufferSymbol: IFillSymbol
■	SourceID: String

ImageDisplay

IImageDisplay : IUnknown	
■	DeviceResolution: Double
■	Height: Long
■	Width: Long

ImageType

IImageType : IUnknown	
■	Format: esriImageFormat
■	ReturnType: esriImageReturnType



ArcGIS Server - GIS Web Services and ArcObjects

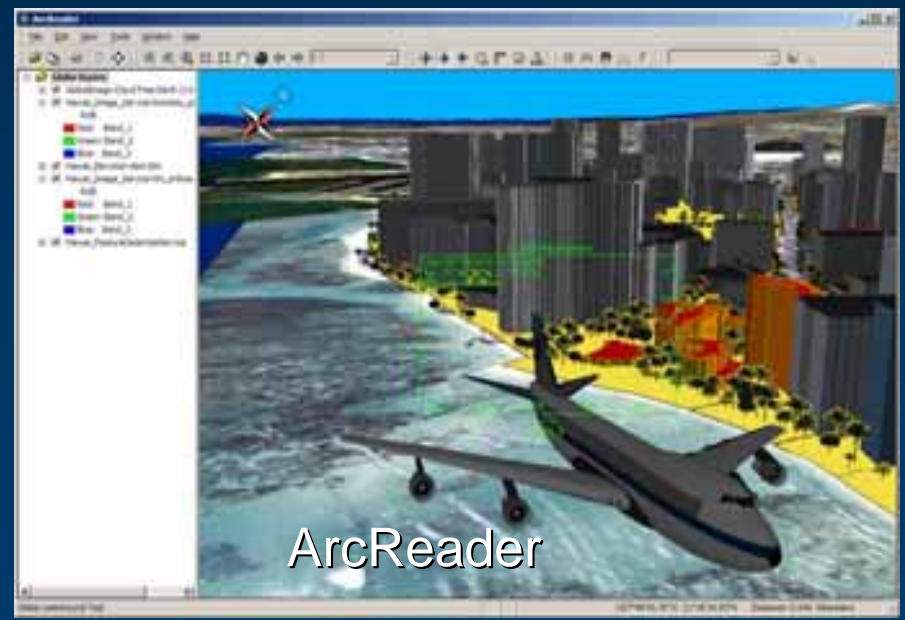
- The ArcObjects object model for a service is available on both the server (ServerObject running within the ServerContext) *and on the client.*
- The client object model consists of a server object proxy and all the value types
- The GISClient assembly can be used to connect to a server and obtain proxies.

Globe Services

- Efficiently deliver *globe layers* for display on a 3D Map / Globe
- Deliver *animations*
- Support *identify* and *find features* functionality
- Support web based access
- Clients can mix and match different services

Applications that work with Globe Services

- Any globe enabled ArcGIS Client
 - ArcGIS Explorer
 - ArcGIS Desktop Applications - ArcGlobe and ArcCatalog
 - ArcReader
 - ArcGIS Engine Applications based on the GlobeControl



Globe Service – Publishers View

- Based on a Globe document (.3dd) authored using ArcGlobe
- Exposes a set of individually consumable Globe Layers
- Returns tiles for each layer that are optimized for display on a 3D Globe
- Tiles are in the Cube projection and use a fixed tiling scheme.
- ArcGIS Explorer, ArcGlobe and the Globe control have the ability to drape these tiles on the globe

Types of Globe Layers Delivered by a Globe Service

- **Elevation**
- **Imagery**
- **Draped 2D Vectors**
- **Draped 2D Text**
- **3D Vectors (eg. buildings)**
- **3D Graphics and Text (bill boarded)**

Globe Service – Developer's View

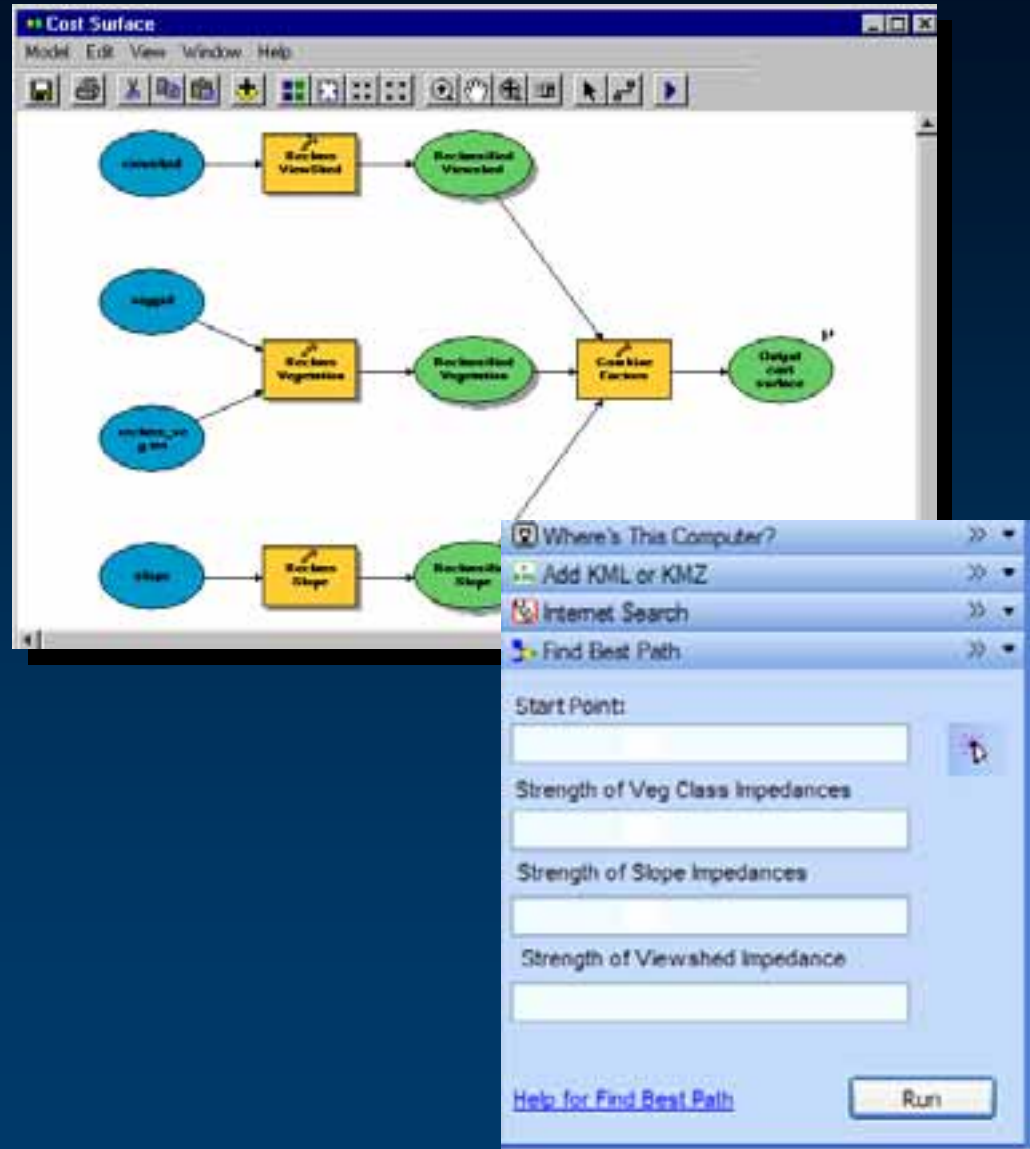
- Intended to be used within an ArcEngine application that is based on the Globe Control
- Use the ArcObjects API for working with ArcGIS Server

ArcGIS Server 9.2 Geoprocessing

- Allows organizations to centralize both data and processing on the server
- Allows organizations with spatial data to expose spatial analysis functionality against that data as easy to use tasks that can be invoked by less experienced users
- Allows GIS Analysts to easily author and publish geoprocessing models

Publishing Geoprocessing Models

- Author models using Desktop
- Publish models creating Geoprocessing Services
- Published models are exposed as *tasks*
- Tasks are tools that deliver specific GIS functionality
- Consume tasks in
 - Web Applications
 - ArcGIS Explorer
 - ArcGIS Desktop



Working directly with the Geoprocessing Service

- Methods to :
 - List the tools
 - Get information on the parameters for each tool
 - Name, Type, Default Value, ...
 - Execute a tool synchronously
 - Execute a tool asynchronously
 - Returns a Job ID
 - Methods to discover status of pending jobs
 - Methods to download results for completed jobs
 - Methods to draw results for completed jobs on the server using an associated map service
- Note :
 - Web ADF, Explorer and Desktop include a Geoprocessing Task that will autogenerate UI dialogs for any published GP tool.

Geoprocessing Service - Parameters

- **Supported parameter types**
 - String
 - Long
 - Double
 - Boolean
 - Linear unit
 - Feature record set
 - Table record set
 - Raster data
 - Map Layer on the server

Geoprocessing Service Properties

- **Execution Mode :**

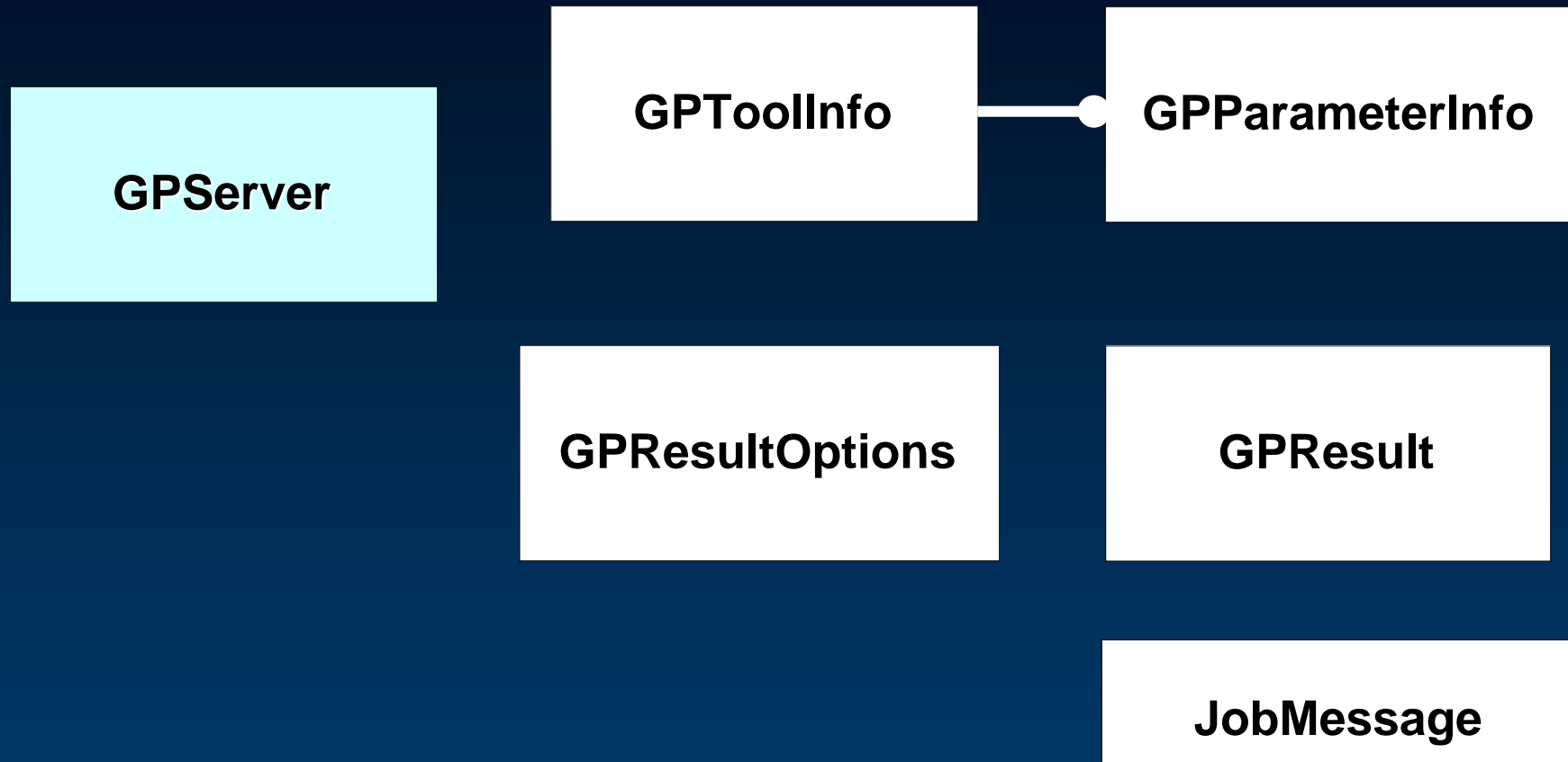
- **Synchronous**

- Results calculated and returned immediately by value
 - Appropriate for fast tools with small results

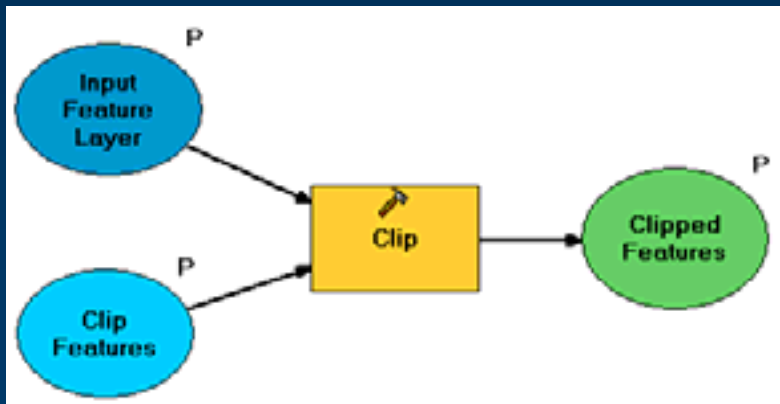
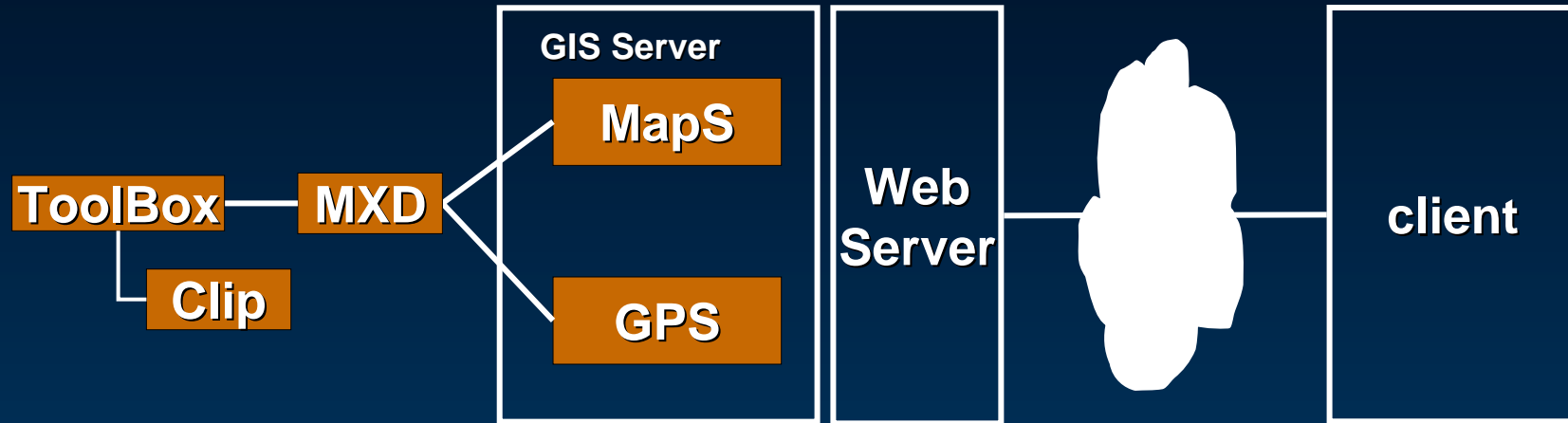
- **Asynchronous**

- Allows client to submit a job and come back later to fetch or draw results
 - Results are saved on the server
 - large results are best left on the server

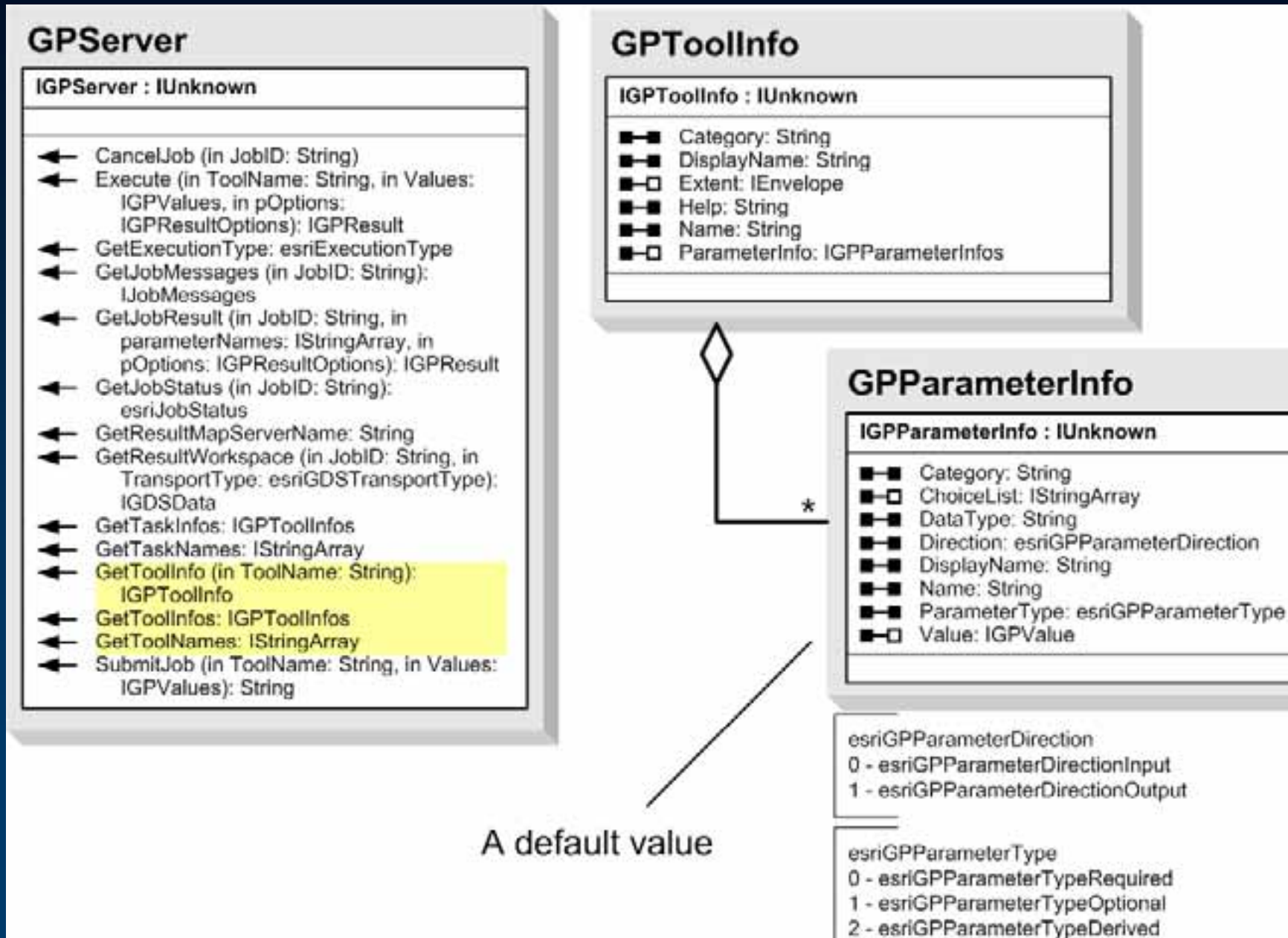
Geoprocessing Service – Object Model



Geoprocessing Demo – interrogate a service, use a clip tool in the service



Getting tool information



Getting tool information

```
GPToolInfo ti = m_gps.GetToolInfo(toolName);
StringBuilder sb = new StringBuilder(1024);
sb.Append("Name: " + ti.Name + "\r\n");
sb.Append("Help: " + ti.Help + "\r\n");
sb.Append("Extent: " + DescExtent((EnvelopeN)ti.Extent) + "\r\n");
sb.Append("NumParms: " + ti.ParameterInfo.GetLength(0).ToString() + "\r\n");
foreach(GPParameterInfo pi in ti.ParameterInfo)
{
    sb.Append("\tName: " + pi.Name + "\r\n");
    sb.Append("\tDisplayName: " + pi.DisplayName + "\r\n");
    sb.Append("\tDataType: " + pi.DataType + "\r\n");
    sb.Append("\tDirection: " + pi.Direction.ToString() + "\r\n");
    sb.Append("\tCategory: " + pi.Category + "\r\n");

    if (pi.ChoiceList != null)
    {
        sb.Append("\tChoiceList:\n");
        foreach(string choice in pi.ChoiceList)
            sb.Append("\t\t" + choice + "\r\n");
    }
}
MessageBox.Show(sb.ToString(), "ToolInfo");
```

Executing a tool

GPServer

IGPServer : IUnknown

- ← CancelJob (in JobID: String)
- ← Execute (in ToolName: String, in Values: IGPValues, in pOptions: IGPResultOptions): IGPResult
- ← GetExecutionType: esriExecutionType
- ← GetJobMessages (in JobID: String): IJobMessages
- ← GetJobResult (in JobID: String, in parameterNames: IStringArray, in pOptions: IGPResultOptions): IGPResult
- ← GetJobStatus (in JobID: String): esriJobStatus
- ← GetResultMapServerName: String
- ← GetResultWorkspace (in JobID: String, in TransportType: esriGDSTransportType): IGSDData
- ← GetTaskInfos: IGPToolInfos
- ← GetTaskNames: IStringArray
- ← GetToolInfo (in ToolName: String): IGPToolInfo
- ← GetToolInfos: IGPToolInfos
- ← GetToolNames: IStringArray
- ← SubmitJob (in ToolName: String, in Values: IGPValues): String

esriExecutionType

- 1 - esriExecutionTypeAsynchronous
- 2 - esriExecutionTypeSynchronous

esriJobStatus

- 0 - esriJobNew
- 1 - esriJobSubmitted
- 2 - esriJobWaiting
- 3 - esriJobExecuting
- 4 - esriJobSucceeded

GPResultOptions

IGPResultOptions : IUnknown

- DensifyFeatures: Boolean
- TransportType: esriGDSTransportType

GPResult

IGPResult : IUnknown

- Messages: IJobMessages
- Values: IGPValues

JobMessage

IJobMessage : IUnknown

- Description: String
- MessageType: esriJobMessageType

esriJobMessageType

- 0 - esriJobMessageTypeInformative
- 1 - esriJobMessageTypeWarning
- 2 - esriJobMessageTypeError
- 3 - esriJobMessageTypeEmpty
- 4 - esriJobMessageTypeAbort

Geodata Services and Replication

- **Geodata Services allow you to publish a geodatabase so that it can be accessed remotely over the Web**
- **ArcGIS Desktop can be used as a client with both local and remote geodatabases in order to**
 - **Extract data**
 - **Create replicas**
 - **Synchronize replicas**

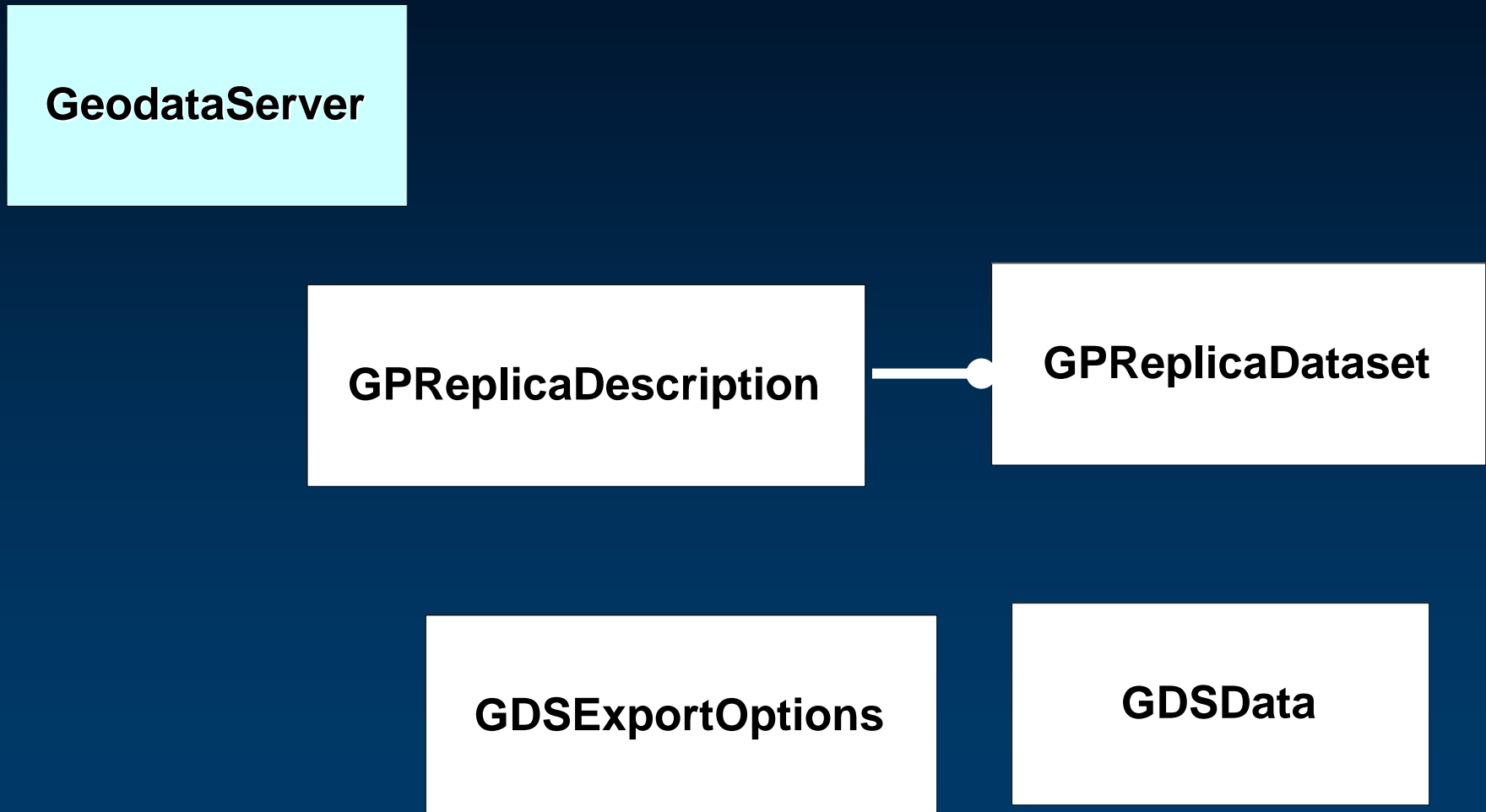
Working with Geodata Services in ArcGIS Desktop

- **All the distributed database tools work with local geodatabase connections as well as with geodata services (“ remote geodatabase connections “)**
- **GIS Analysts can also use GP scripting to setup automated synchronization jobs that automatically synchronize geodatabases at periodic intervals**

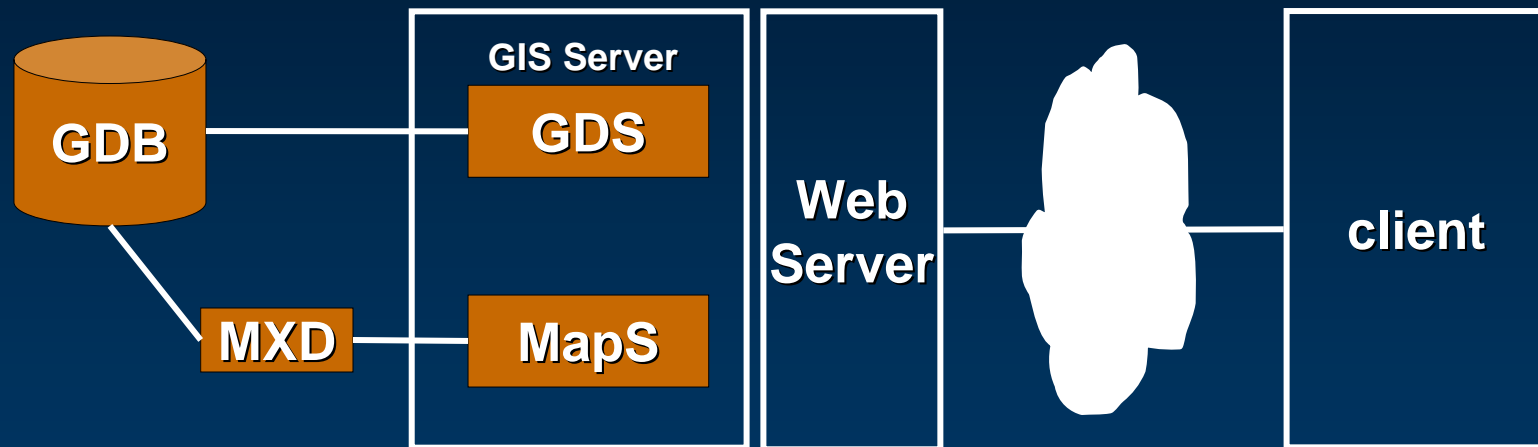
Geodata Service

- Methods for
 - browsing datasets
 - exporting datasets to XML
 - creating check-out, 1 way and 2 way replicas that can be downloaded and edited
 - exporting and importing replica changes.

Geodata Service – Object Model



Geodata Service demo – data extraction



Geodata Service - Creating a replica

GeoDataServer

IGeoDataServer : IUnknown

- DataElements (in pBrowseOptions: IDEBrowseOptions): IDataElements
- DefaultWorkingVersion: String
- MaxRecordCount: Long
- Replicas: IGPReplicas
- Versions: IGPVersionInfos
- WrappedWorkspaceType: esriWorkspaceType

← CompareReplicaSchema (in relativeReplicaSchemaDoc: IGSDData, in TransportType: esriGDSTransportType): IGSDData

← **CreateReplica (in parentVersion: String, in ReplicaName: String, in replicaDesc: IGPReplicaDescription, in repOptions: IGPReplicaOptions, in exportOptions: IGDSExportOptions, in TransportType: esriGDSTransportType): IGSDData**

← ExpandReplicaDatasets (in repDatasets: IGPReplicaDatasets): IGPReplicaDatasets

← ExportAcknowledgement (in ReplicaName: String, in TransportType: esriGDSTransportType): IGSDData

← ExportReplicaDataChanges (in ReplicaName: String, in options: IGDSExportOptions, in TransportType: esriGDSTransportType, in generationsToExport: esriExportGenerationsOption, in switchRole: Boolean): IGSDData

← ExportReplicaSchema (in ReplicaName: String, in TransportType: esriGDSTransportType): IGSDData

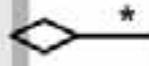
← ExtractData (in versionName: String, in replicaDesc: IGPReplicaDescription, in

esriReplicaModelType
 1 - esriModelTypeSimple
 2 - esriModelTypeFullGeodatabase

GPReplicaDescription

IGPReplicaDescription : IUnknown

- ModelType: esriReplicaModelType
- QueryGeometry: IGeometry
- ReplicaDatasets: IGPReplicaDatasets
- SingleGeneration: Boolean
- SpatialRelation: esriSpatialRelEnum
- TransferRelatedObjects: Boolean



GPReplicaDataset

IGPReplicaDataset : IUnknown

- DatasetType: esriDatasetType
- DefQuery: String
- IsPrivate: Boolean
- Name: String
- RelExtractDirection: esriRelExtractDirect
- RowsType: esriRowsType
- SelectionIDs: ILongArray
- UseGeometry: Boolean

GDSEExportOptions

IGDSExportOptions : IUnknown

- BinaryGeometry: Boolean
- Compressed: Boolean
- ExportFormat: esriGDSEExportFormat

esriGDSEExportFormat
 0 - esriGDSEExportFormatPersonalGdb
 1 - esriGDSEExportFormatXml
 2 - esriGDSEExportFormatFileGDB

GDSDData

IGSDData : IUnknown

- Compressed: Boolean
- EmbeddedData: BYTE
- TransportType: esriGDSTransportType
- URL: String

esriGDSTransportType
 0 - esriGDSTransportTypeEmbedded
 1 - esriGDSTransportTypeUri

Geodata Service - Creating a replica

```
//create array of datasets
GPReplicaDataset[] repDSs = new GPReplicaDataset[2];

//fill it up w/datasets to replicate
GPReplicaDataset repDS = new GPReplicaDataset();
repDS.DatasetName = "Parcels";
repDS.DatasetType = esriDatasetType.esriDTFeatureClass;
//selection IDs, UseGeometry, DefQuery...
repDSs[0] = repDS;

repDS = new GPReplicaDataset();
repDS.DatasetName = "Buildings";
repDS.DatasetType = esriDatasetType.esriDTFeatureClass;
repDSs[1] = repDS;

//setup a replicaDescription
GPReplicaDescription repDesc = new GPReplicaDescription();
repDesc.GPReplicaDatasets = repDSs;
repDesc.QueryGeometry = myEnvelope;
repDesc.SingleGeneration = true; //ie checkOut ...
```

Geodata Service - Creating a replica

```
//extra options
GPReplicaOptions repOpts = new GPReplicaOptions();
repOpts.AccessType = esriReplicaAccessType.esriReplicaBothReadWrite;

//how to format the data
GDSExportOptions expOpts = new GDSExportOptions();
expOpts.ExportFormat = esriGDSExportFormat.esriGDSExportFormatFileGDB;

//how to transfer the data
esriGDSTransportType transpType =
    esriGDSTransportType.esriGDSTransportTypeUrl;

//create the replica
GDSData data =
    m_gds.CreateReplica(sdeVersion, repName, repDesc, repOpts, expOpts,
        transpType);

DownloadFile(data.URL, outFGdb);

//m_gds.ExtractData with similar signature will extract the data (no replica
    created)
```

Geodata Service – Synchronizing Replicas

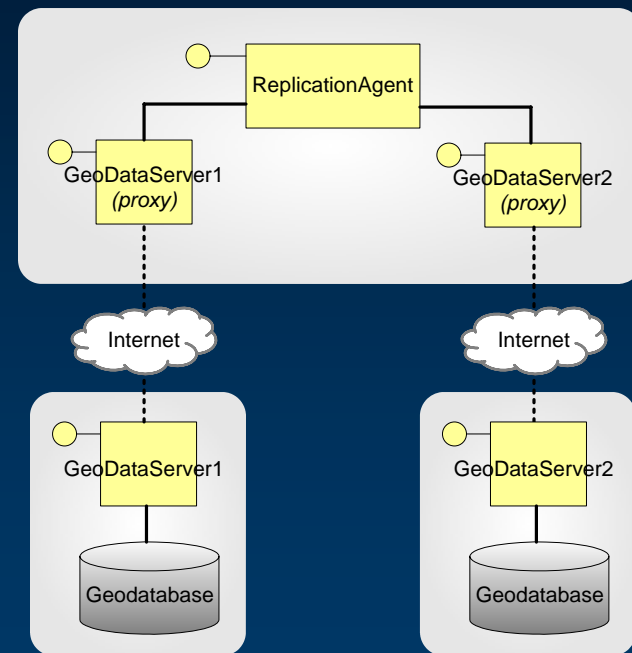
```
//export from source gds
GDSDData changes
  = src_gds.ExportReplicaDataChanges(repName, expOpts, transpType,
    esriExportGenerationsOption.esriExportGenerationsAll, switchRoleYes);

//import into destination gds
dest_gds.ImportReplicaDataChanges(
  esriGDSReplicaImportSource.esriGDSReplicaImportSourceDeltaFileGDB,
  esriReplicaRecPolicyType.esriResolveConflictsInFavorOfDatabaseChanges,
  colLevelYes, changes);

//repeat in the opposite direction
```

Developing Desktop and ArcEngine Clients to the GeodataService

- Use the `GeoDataServer` and `ReplicationAgent` classes that are part of the `GeodatabaseDistributedArcObjects` assembly
- `ReplicationAgent` gives you single call synchronization



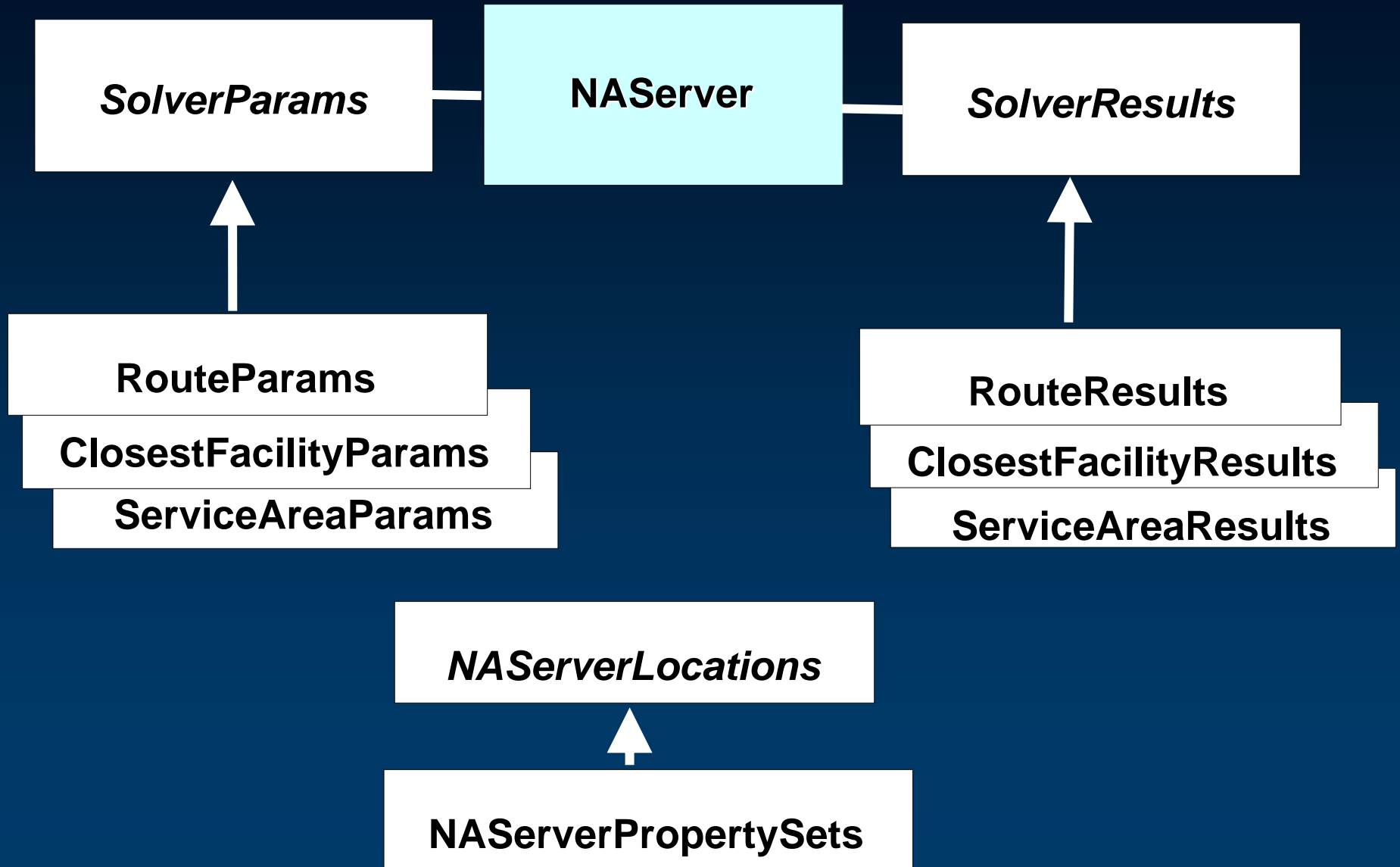
Geocode Service

- Based on a Locator stored in the file system or in a Geodatabase
- 9.2 has new high performance locators
- Methods to perform single address and batch geocoding
- Methods to perform reverse geocoding
- Support for different address styles

Network Analysis Service

- Based on publishing a map containing network analysis layers
- A network analysis layer represents a specific analysis (Routing, Service Area, Closest Facility) to be performed against a specific network dataset (for eg. LA Streets)
- Methods to :
 - Discover the parameters of the analysis solver
 - Solve
 - Get Results (paths, directions etc)
- Callers supply input parameters (for eg stops, barriers) using basic web service types such as PropertySets and RecordSets.

Network Analysis Service – Object Model



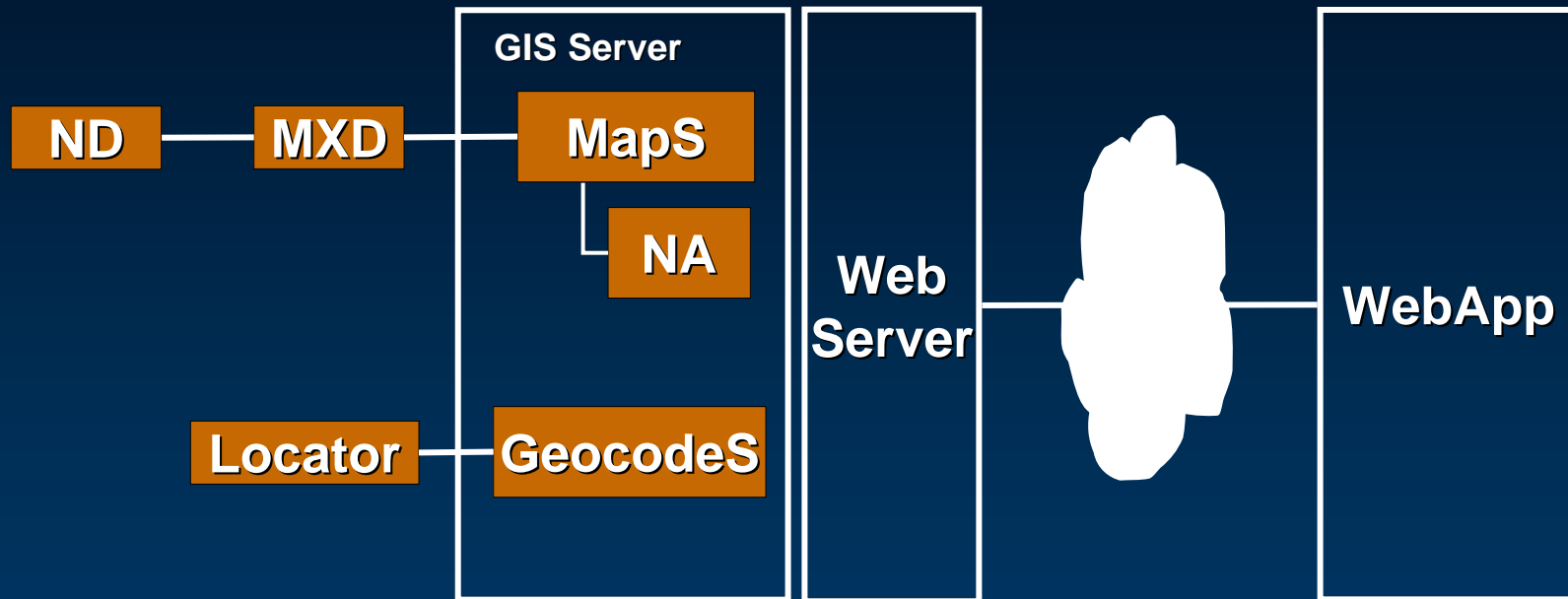
NetworkAnalysis Service Methods

<code>GetNALayerNames</code>	Array of Network Analysis Layers of a particular type
<code>GetNetworkDescription</code>	Returns the Network Dataset Data Element for the specified layer
<code>GetSolverParameters</code>	Default Solver Parameters for the specified Network Analysis Layer
<code>Solve</code>	Perform Network Analysis based on the specified solver parameters

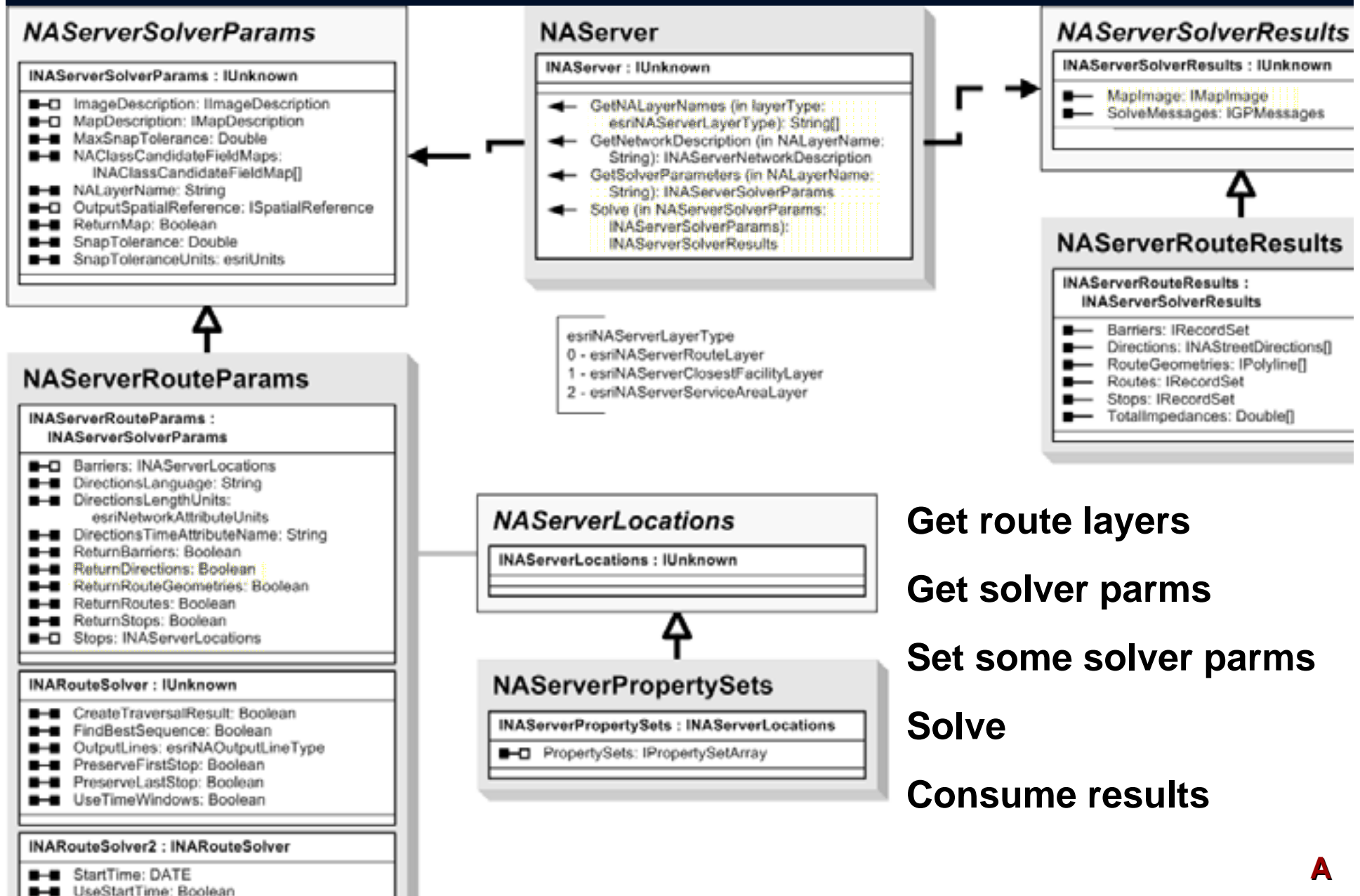
- **Finding a Route :**

- Get Route Network Analysis Layer
- Get Default Route Solver Parameters
- Specify Desired Route Solver Parameters
 - Barriers, Stops, FindBestSequence, UseTimeWindows, ...
- Solve
- Consume Results

Geocoding and Network Analysis Service demo - routing

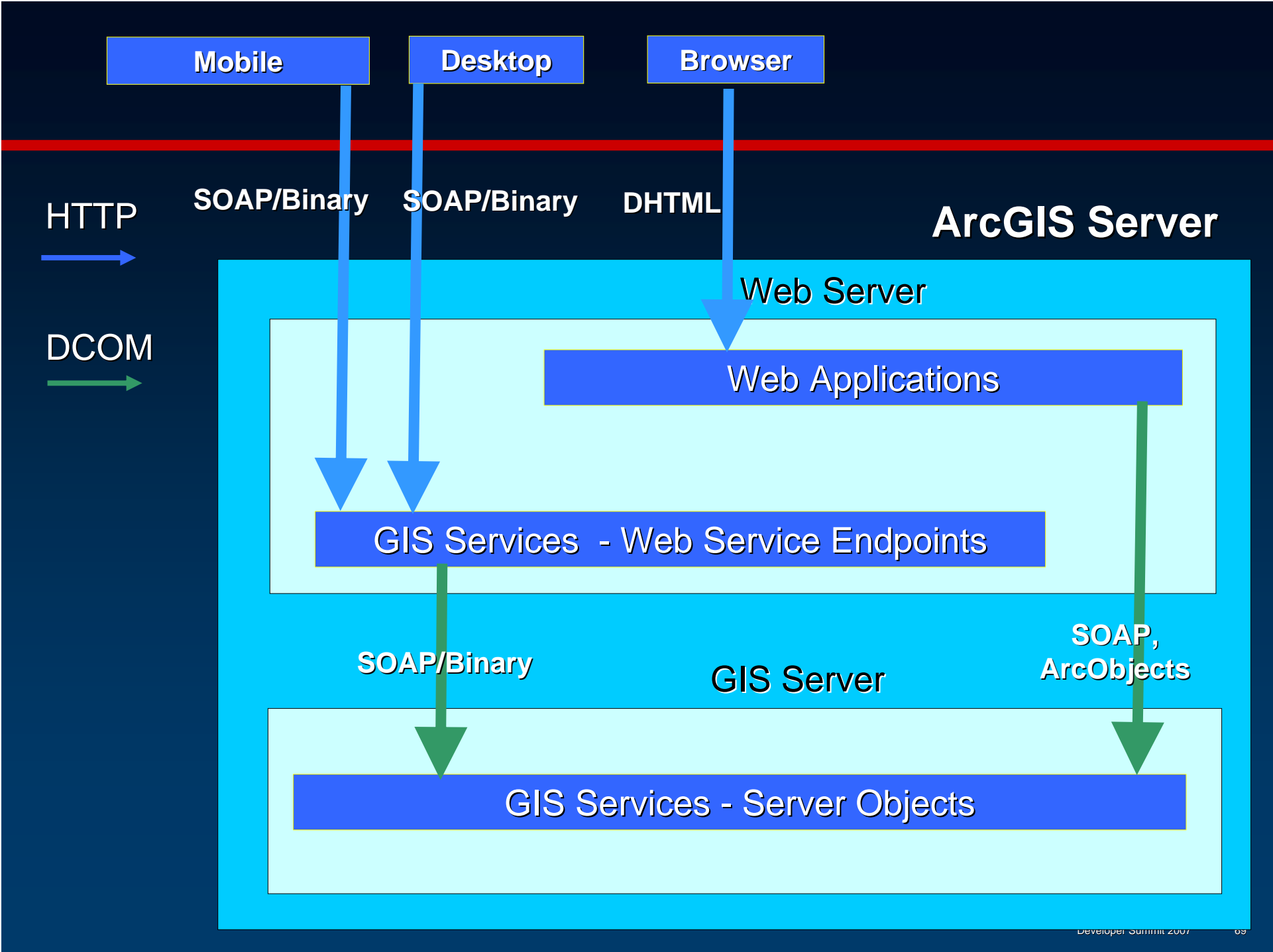


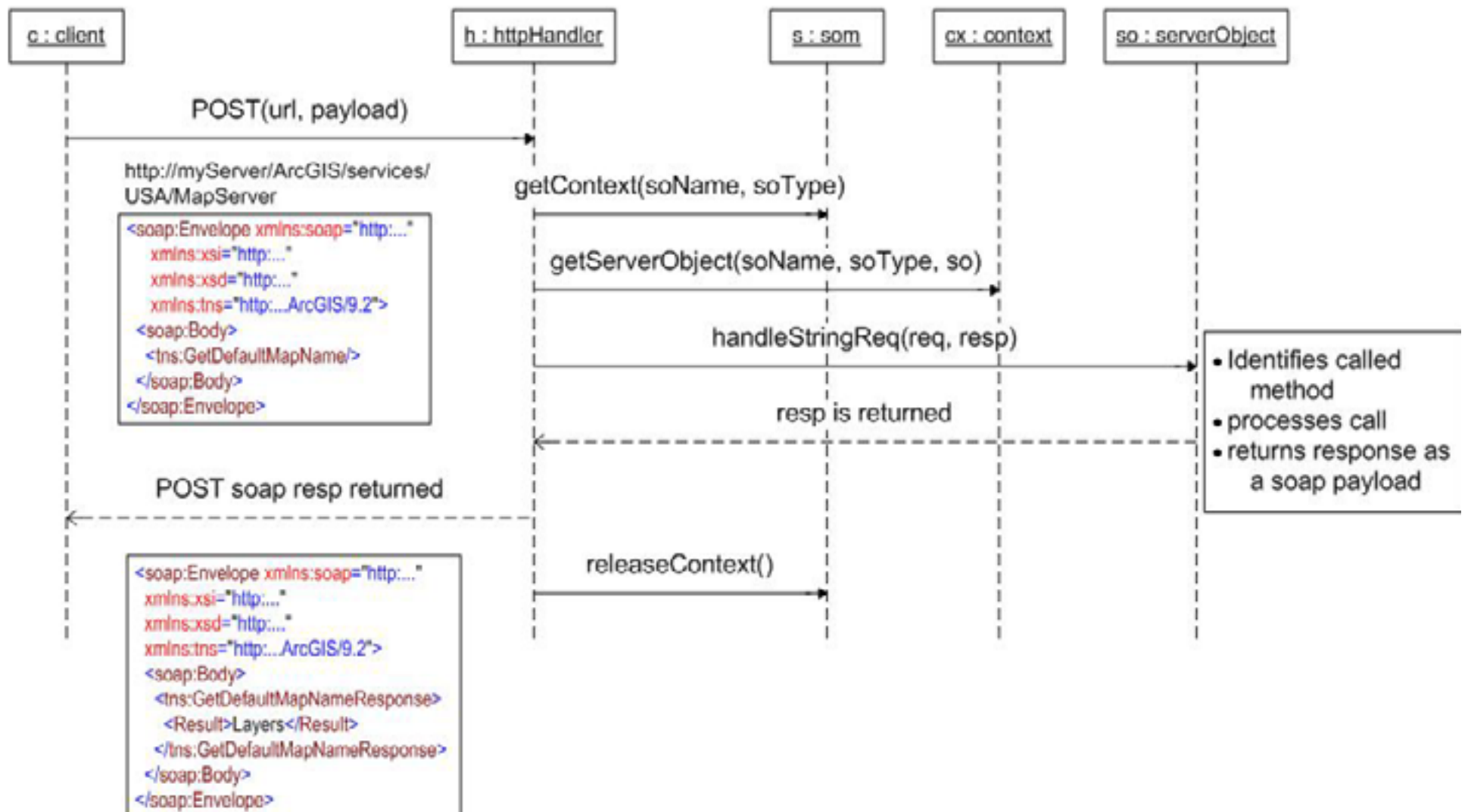
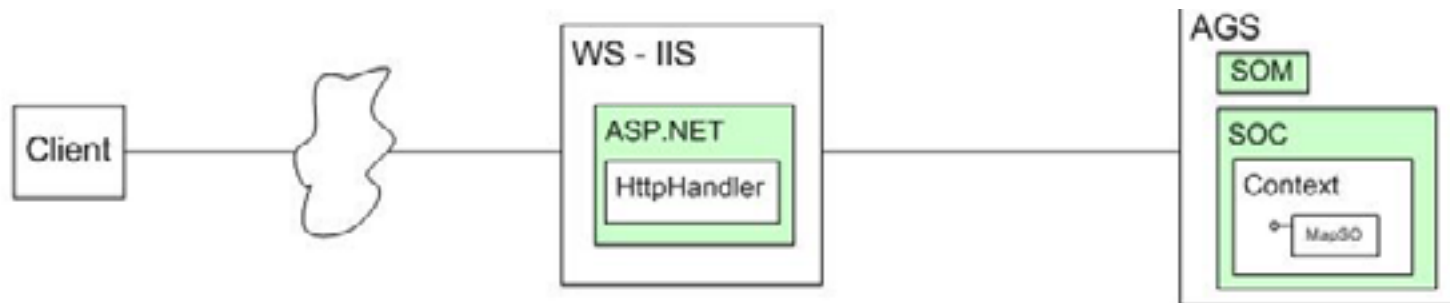
Finding a route



ArcGIS Server Web Services – Under the Hood

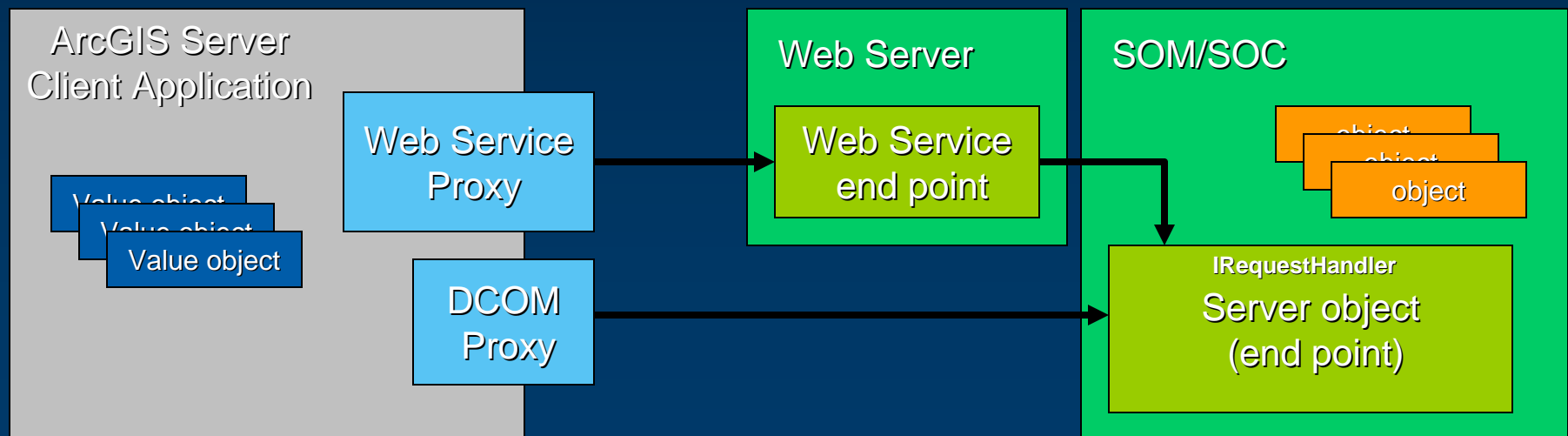
- SOAP requests received by the web server are handled by a custom HTTP handler [.Net] / Servlet [Java] and forwarded to the corresponding server object in the GIS Server for processing.
- **ArcGIS includes its own SOAP stack and XML serialization framework for ArcObjects**
- All parameter objects implement IXMLSerialize
- SOAP requests [the SOAP body] is deserialized and processed by the server object
- Results are serialized into the corresponding SOAP response





Working with SOAP requests over the LAN

- Web ADF includes both a Web proxy as well as a LAN proxy
- Web service proxy generated from WSDL - MapServerProxy
 - Used for SOAP / HTTP
- LAN proxy is also included - MapServerDCOMProxy
 - Used for SOAP / LAN
 - Uses the IRequestHandler interface to submit SOAP requests directly to the server object



Presentation materials

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

Further questions?

- **TECH-TALK AREAS**

- **What:** Further opportunity to discuss questions and concerns with presenters and subject matter experts

- **Where:**

- **When:** during the next 30 minutes

- **ESRI Showcase**

- **Meet the teams**

- **ESRI Developers Network (EDN) website**

- <http://edn.esri.com>