AutoCAD® Map 3D 2013 Platform API

Autodesk

Overview of the Geospatial Platform API

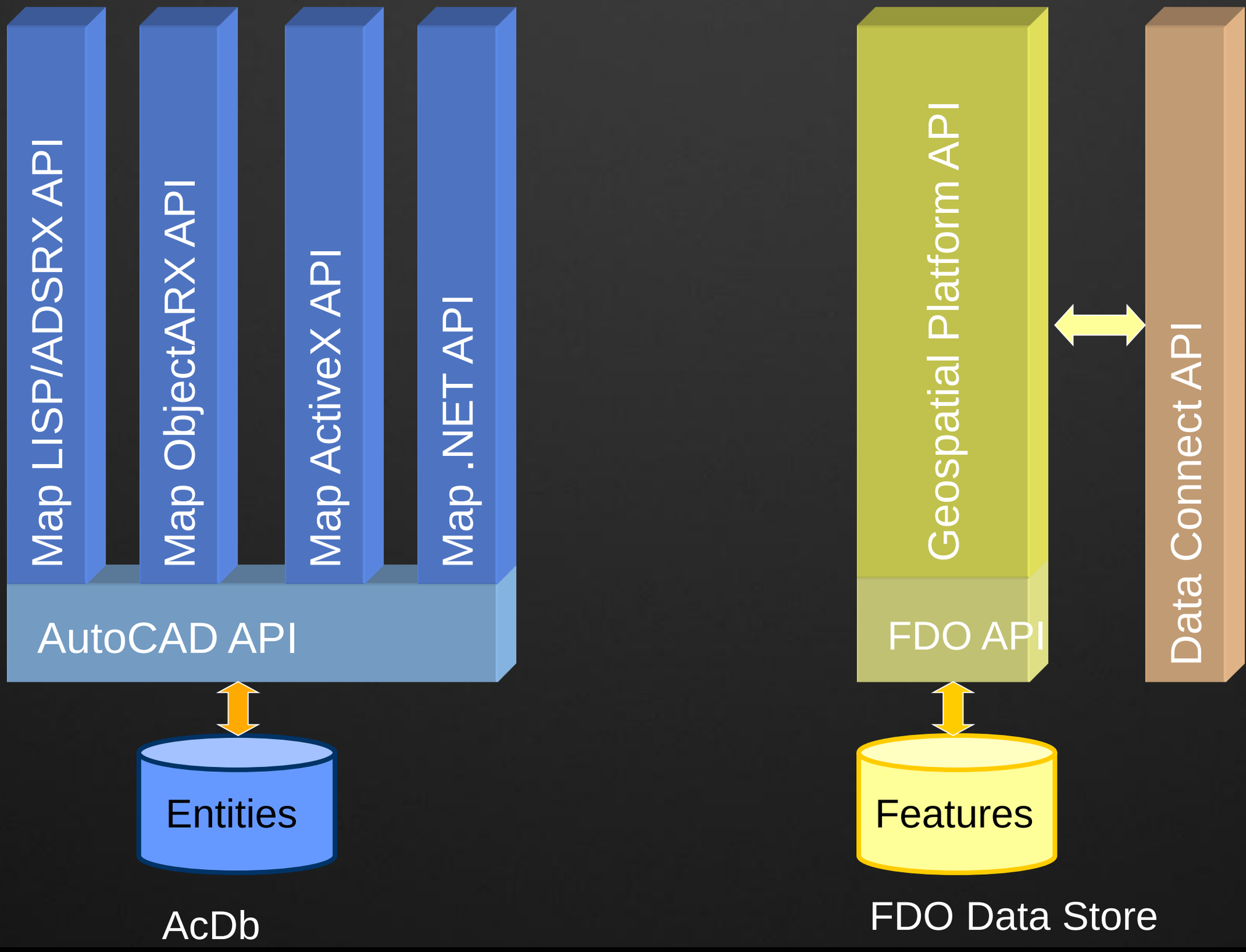
Autodesk

Contents

- Map API Overview
- Introduction to the Geospatial Platform API
- Benefits of the Geospatial Platform API
- Differences between Infrastructure Map Server (MapGuide Enterprise) and Map 3D Geospatial Platform APIs

Autodesk

AutoCAD Map 3D API



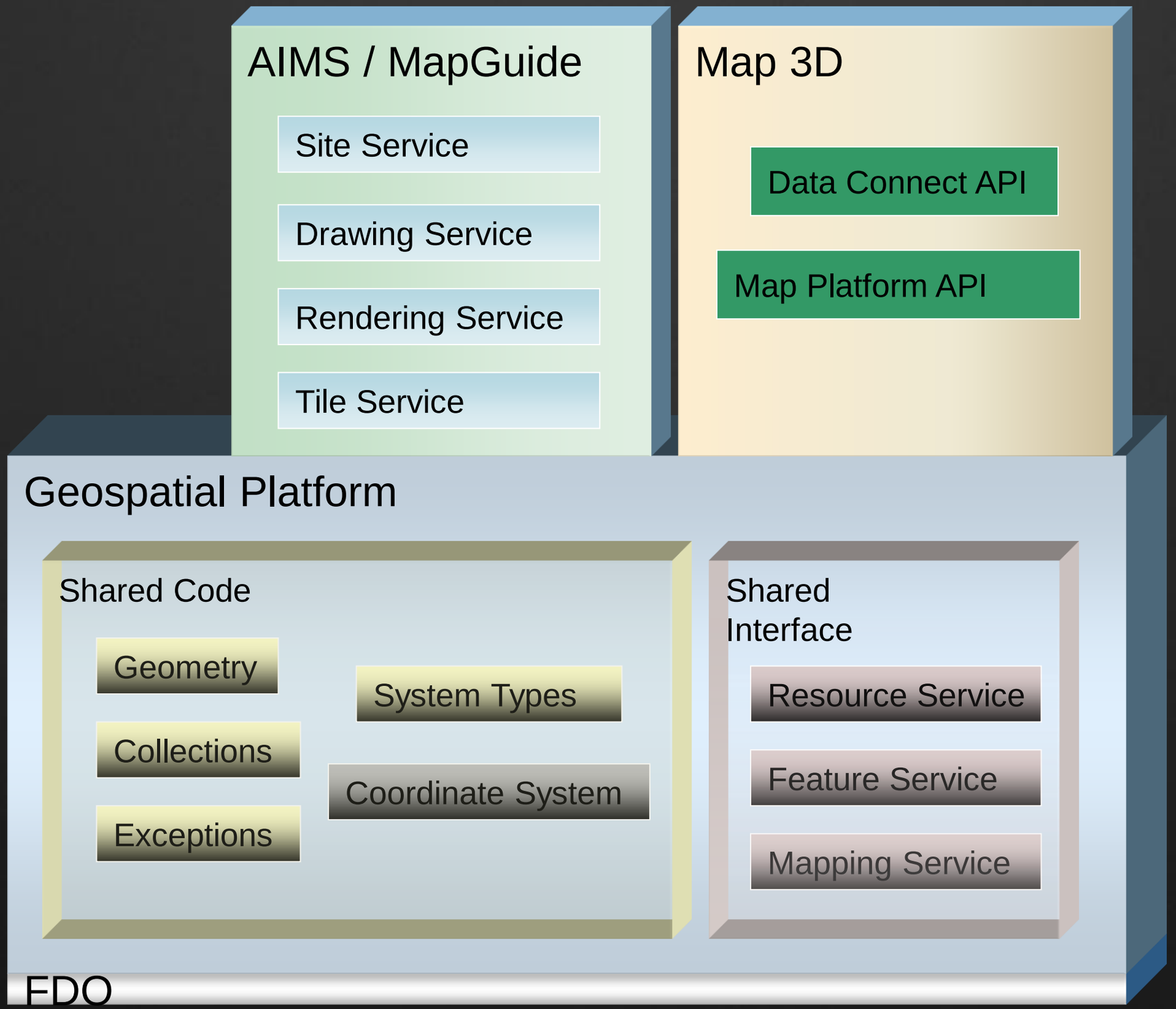
Autodesk

AutoCAD Map 3D Platform API

- Consists of the Geospatial Platform API and its Map 3D-specific extension
  - Several modules shared with AIMS / MGE
- Based on FDO API
  - access to features through FDO Providers
- .NET API

Autodesk

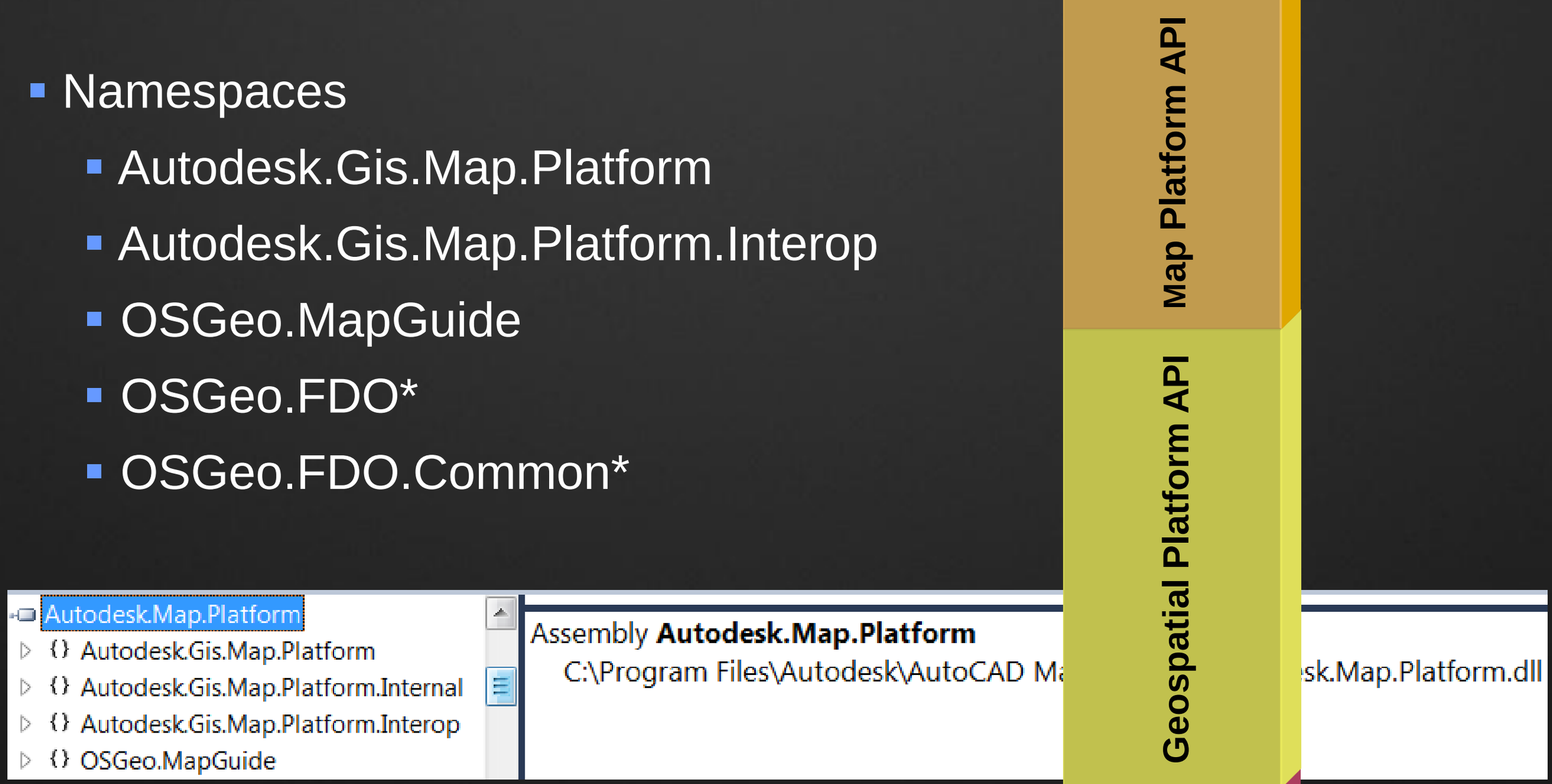
Architecture



Autodesk

AutoCAD Map 3D Platform API

- Namespaces
  - Autodesk.Gis.Map.Platform
  - Autodesk.Gis.Map.Platform.Interop
  - OSGeo.MapGuide
  - OSGeo.FDO*
  - OSGeo.FDO.Common*



Autodesk

API Components

Resource Service

Feature Service

Feature-Entity Service

Map and Layers

Geometry

Coordinate System

Collections

Common

Exceptions

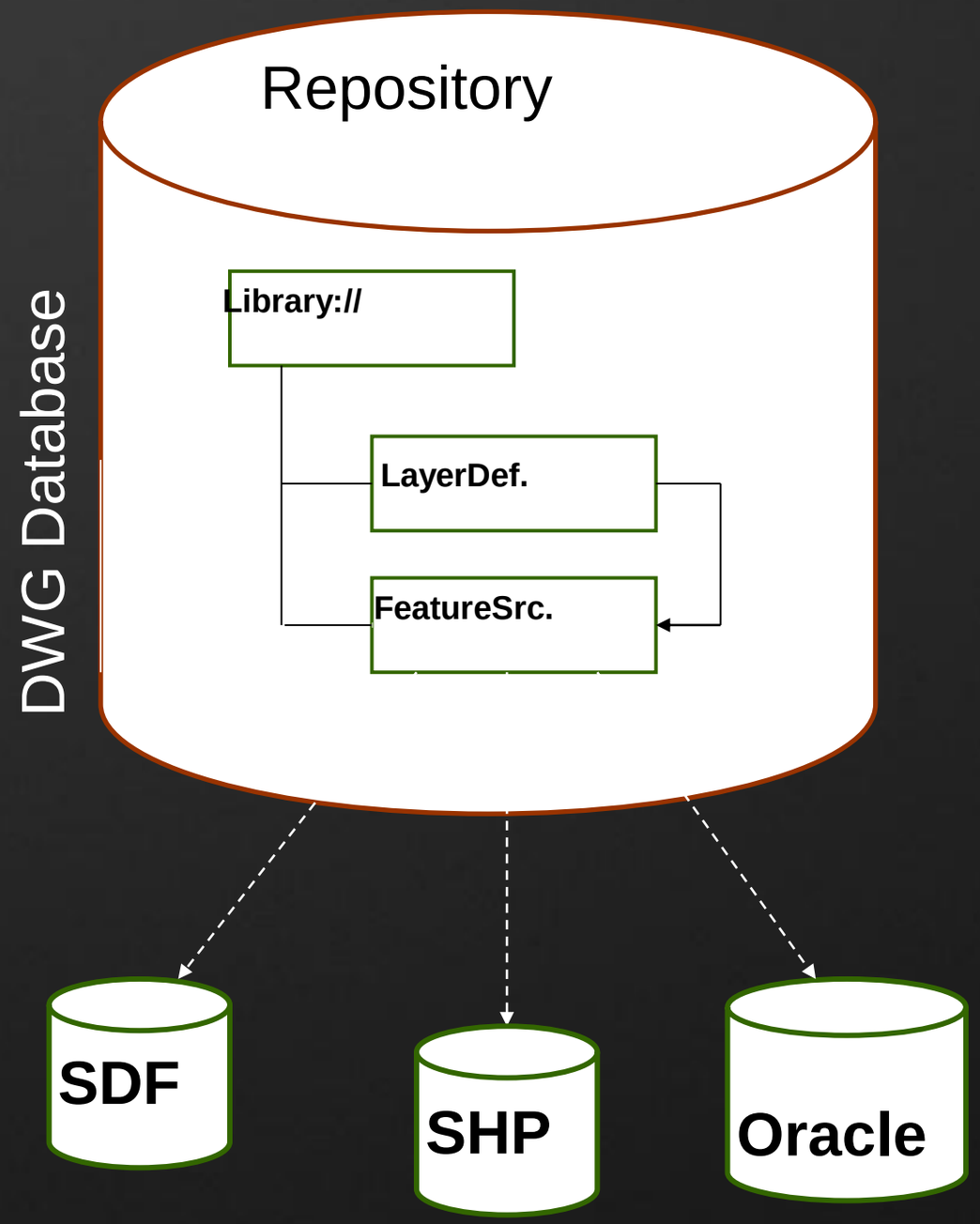
Resource Service

Resources

- Files and configuration information required for map and layer display
- Layer definition and Feature source
- Stored in drawings or drawing templates and are specific to the drawing

Resource Service

- Used for manipulating resources
  - Adding resources to Map
  - Copying resources
  - Checking for existence of resources
  - Getting the contents of a resource



Autodesk

Resource Service

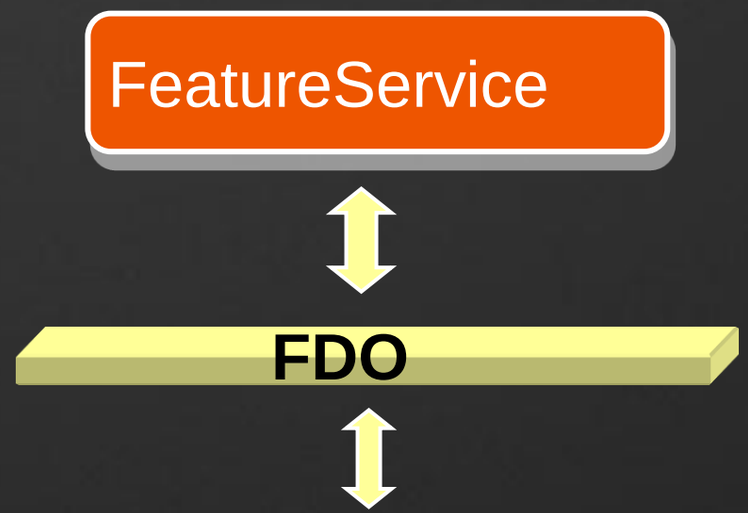
Resource Identifier

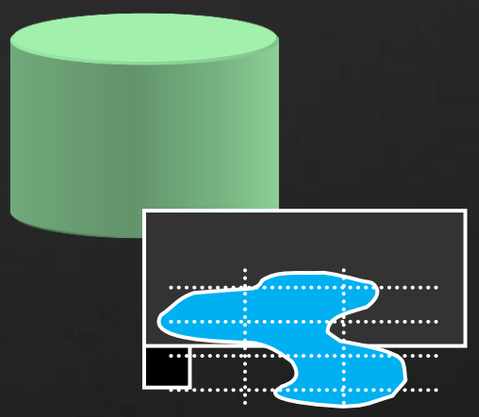
- String that uniquely identifies a resource

```
// using Platform API
using OSGeo.MapGuide;

// Get the resource service
AcMapResourceService resService = (AcMapResourceService )
AcMapServiceFactory.GetService (MgServiceType.ResourceService);

// Create a resource Id for a layer definition resource
MgResourceIdentifier resId = new
MgResourceIdentifier("Library://Data/Raster/Redding.LayerDefinition");

MgByteReader byteRdr = null;

// If the layer definition exists output it
if resServ.ResourceExists(resId)
{
  byteRdr = resServ.GetResourceContent(id);
  MgByteSink byteSink = new MgByteSink(byteRdr);
  byteSink.ToFile(@"c:\ReddingLayerDef.Xml");
}
```

Feature Service

- Provides access to FDO functionality
  - Providers and Capabilities
  - Creating feature sources
  - Querying for features
  - Creating and modifying features
- Enhanced in AutoCAD Map with events to track feature creation, modification and deletion





Autodesk

Feature Service

Selecting Features

```
// using platform API
using OSGeo.MapGuide;

// Get the feature service
AcMapFeatureService featureService =
AcMapServiceFactory.GetService(ServiceType.FeatureService) as AcMapFeatureService;

// Resource identifier
MgResourceIdentifier resId = new MgResourceIdentifier(
      "Library://SDF_1.FeatureSource");
String className="SDF_2_Schema:Hauptstrassen";

// Define filter for selection
MgFeatureQueryOptions options = new MgFeatureQueryOptions();
options.SetFilter("Year > 1999");

// Select features
MgFeatureReader ftrReader = featureService.SelectFeatures(resId, className, options);
```

Feature-Entity Service

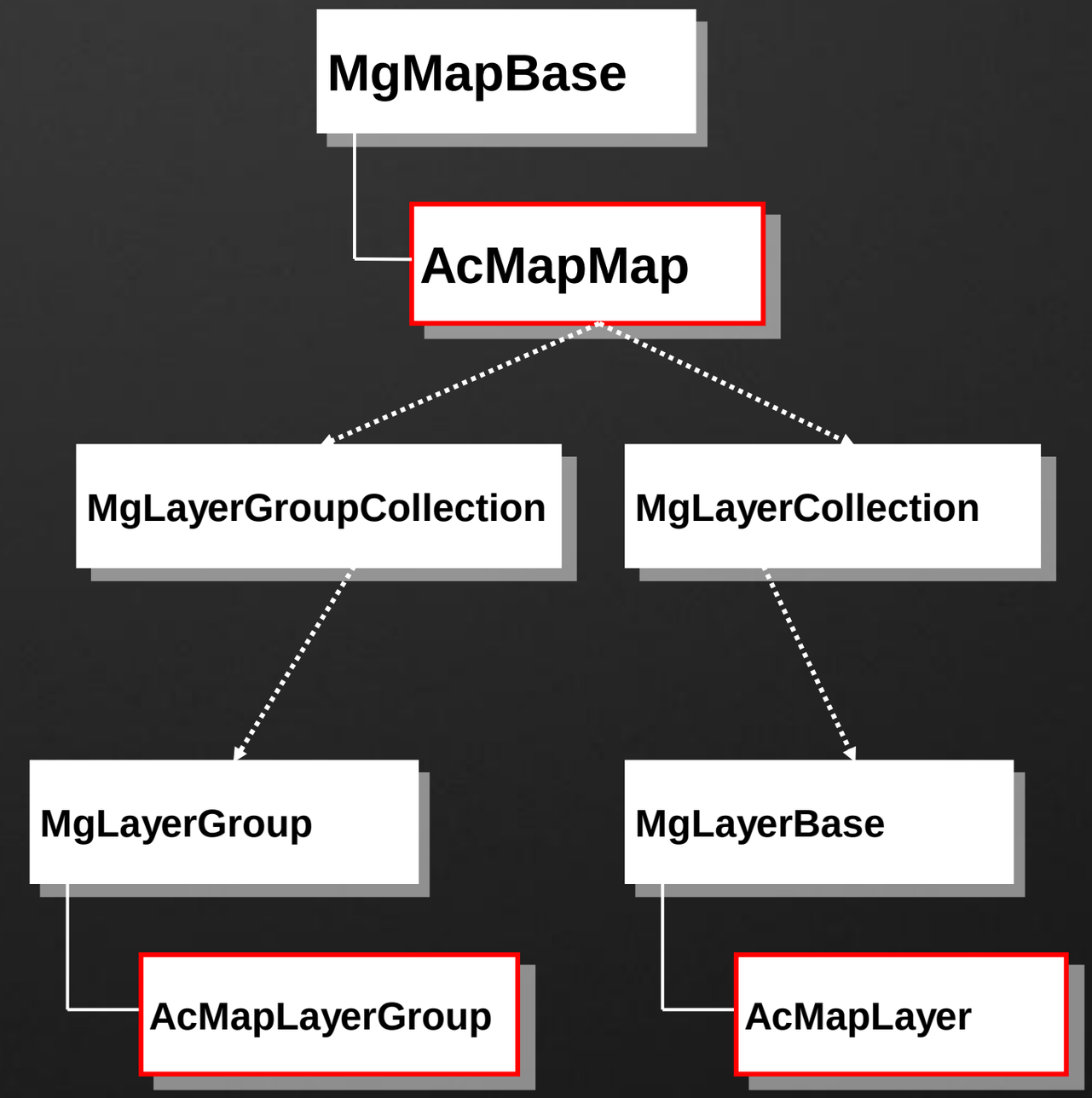
- Enables AutoCAD Selection API support for Features
  - Selection set management
- Support for feature highlighting

AcMapFeatureEntityService

```
//select feature on Map
PromptSelectionResult selResult = ed.GetSelection();
if (selResult.Status == PromptStatus.OK)
{
    SelectionSet selSet = selResult.Value;
    MgSelectionBase selectionBase = AcMapFeatureEntityService.GetSelection(selSet);
    // Deal with Selected Features
}
```

Mapping Service

- Provides access to map and map layers
  - Get and set map and layer properties
  - Root class AcMapMap extends MgMapBase



Mapping Service

Accessing Map and Layer objects

```
AcMapMap currentMap = AcMapMap.GetCurrentMap()  ;
MgLayerCollection layers = currentMap.GetLayers();
MgLayerGroupCollection layerGroups = currentMap.GetLayerGroups();

//Iterate through layers and toggle visibility
foreach (AcMapLayer layer in layers)
{
    bool visibility = IsVisible();
    layer.SetVisible(!visibility);
}
```

Geometry

- Contains geometric object types and operations to manipulate them
  - Supports
    - spatial comparisons between objects
    - creation of new objects based on the intersection, difference, or union of existing objects
    - creation of buffers around objects

MgPoint

MgMultiPoint

MgLineString

MgMultiLineString

MgPolygon

MgMultiPolygon

MgCurveString

MgMultiCurveString

MgCurvePolygon

MgMultiCurvePolygon

MgMultiGeometry

Coordinate Systems

- Provides unified access to coordinate system information
- Enables coordinate system transformations

MgCoordinateSystem

MgCoordinateSystemType

MgCoordinateSystemTransform

MgCoordinateSystemMeasure

MgCoordinateSystemFactory

Utility Classes

Common

- The Common classes provide a set of utility classes

MgByte

MgByteReader

MgByteSink

MgByteSource

MgService

MgColor

MgMimeType

MgDateTime

MgFeatureInformation

MgWarning

Autodesk

Utility Classes

MgByteSource

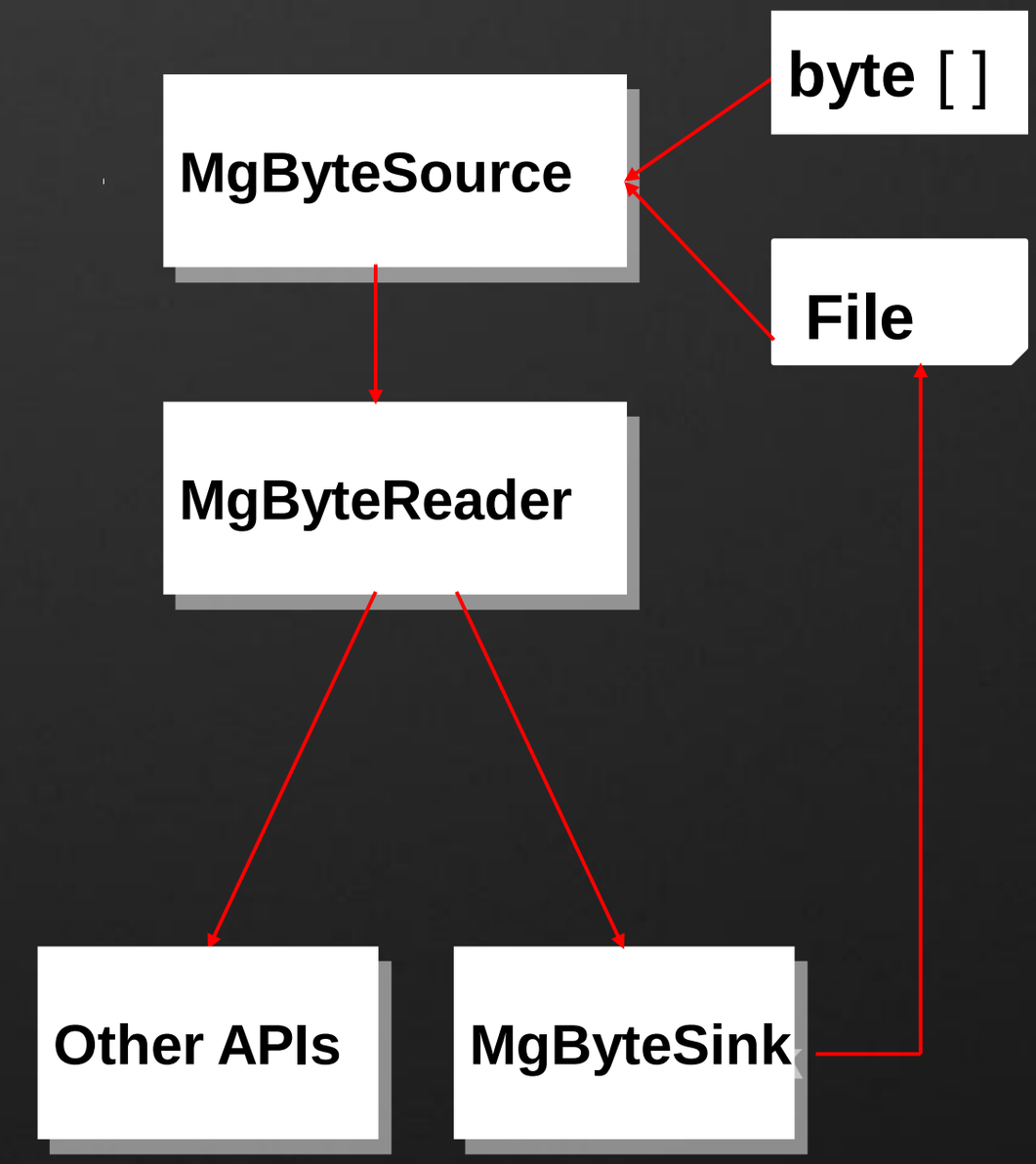
- Byte data source with a specific content type
- Can be initialized from an array of bytes in memory or a file

MgByteReader

- Reads data from MgByteSource
  - MgByteSource::GetReader()

MgByteSink

- Used for writing MgByteReader data to file
  - Construct with MgByteReader and call ToFile()

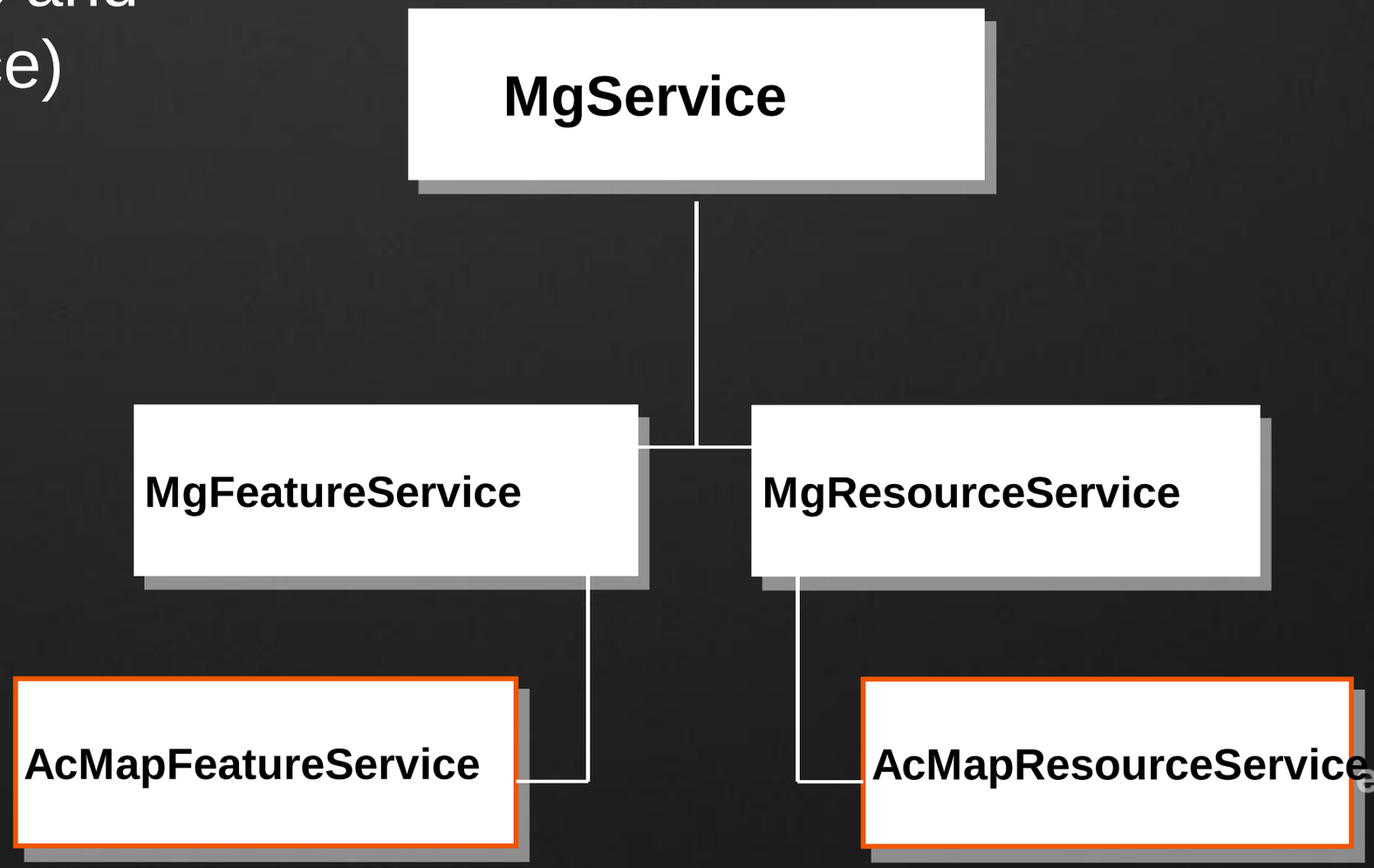


Autodesk

Utility Classes

MgService

- Base class for all Services (AcMapFeatureService and AcMapResourceService)



Autodesk

Utility Classes

Collections

- Container for a set of objects of the same type

MgCollection

MgIntCollection

MgStringCollection

MgPointCollection

MgPolygonCollection

MgPropertyCollection

MgLineStringCollection

MgLinerRingCollection

MgCoordinateCollection

MgCurveRingCollection

MgGeometryCollection

MgFeatureSchemaCollection

Utility Classes

Exceptions

- A set of classes representing all possible exceptions that can be generated by the application.
- 100+ classes

Autodesk

Benefits of the Geospatial Platform API

- Programming against features without worrying about any AcDb constructs
- Highly scalable way of working with features
- Build common code for Geoprocessing tools (like buffer) that can be shared between Map and MapGuide
- Directly access data in various data sources
- Added data access and functionality enhancements through Open Source community

AIMS / MGE and Map 3D Geospatial Platform APIs - Differences

- No DrawingSource, LoadProcedure, MapDefinition, PrintLayout, SymbolSet, or WebLayout resources
- Use AcMapServiceFactory.GetService() to create service in AutoCAD Map
- Resources are stored in DWG in AutoCAD Map
- No session repository in Map, only Library://
- No resource headers in Map
- Methods relating to the unimplemented resources (above) are not in AutoCAD Map