



2016 Esri China
Developer Summit

ArcGIS平台的影像管理技术

慕晓燕

Content

Imagery 与 ArcGIS

大规模影像管理解决方案

影像即服务: Image Service



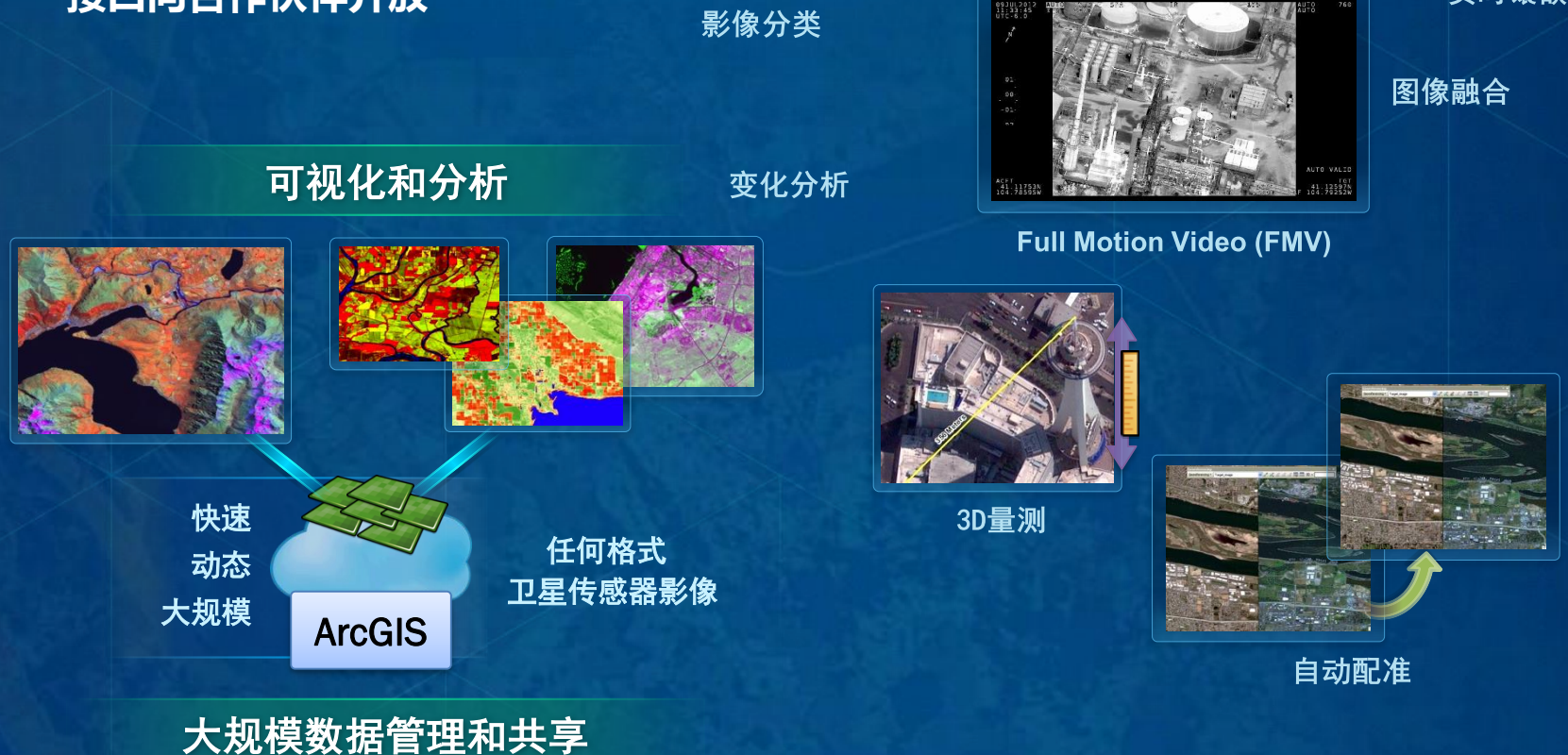


2016 Esri China
Developer Summit

Imagery 与 ArcGIS

ArcGIS是一个影像平台

- 支持多种通用影像处理任务
- 接口向合作伙伴开放



ArcGIS 影像信息模型

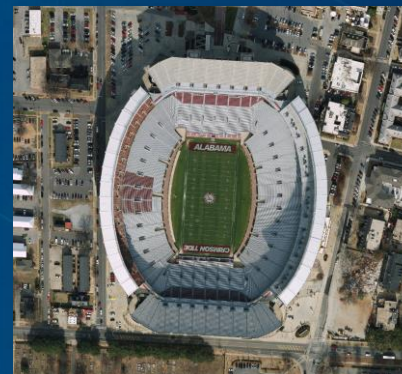
- Raster Dataset
- Raster Type
- Raster Function
- Raster Product
- Mosaic Dataset
- Image Service



ArcGIS 影像信息模型

Raster Dataset

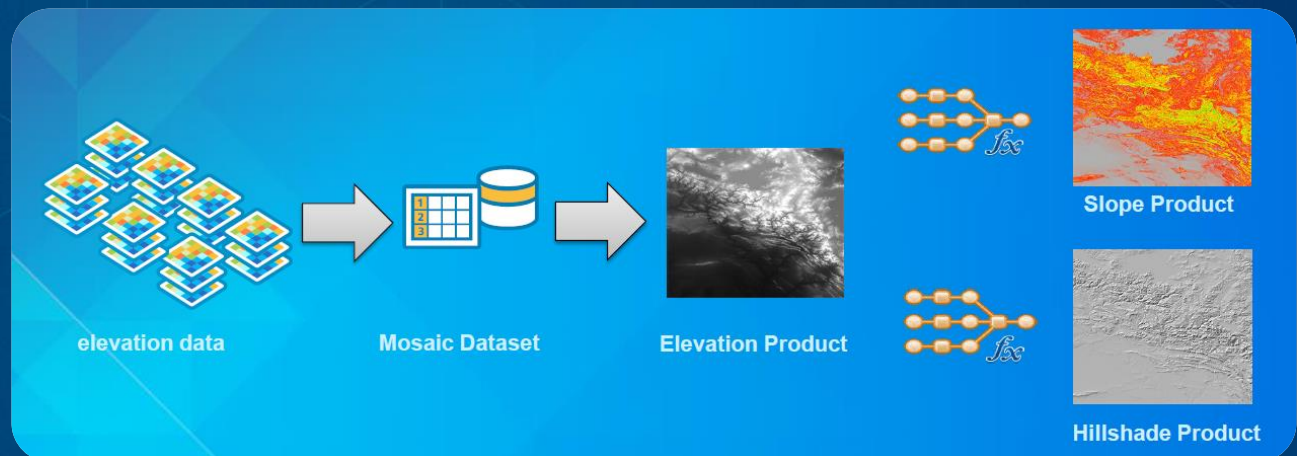
- ArcGIS支持单幅影像的信息模型
- 模型完善
 - 1 或 N 个波段
 - 每一波段 1 - 64 bit
 - 压缩或未压缩
 - 80+ 种支持的数据格式
 - 金字塔 (rrd, ovr, internal...)
- 直接读取 —— 不需要数据转换



ArcGIS 影像信息模型

Raster Function

- 实时处理、按需处理
- 自定义影像处理链
- 节省时间: 处理结果实时显示
- 节省空间: 不产生中间冗余数据
- 50+ Raster Function



ArcGIS 影像信息模型

Raster Type

- 定义了像元存储和元数据结构
- 定义了ArcGIS读取影像的规则
- 定义了默认函数处理链
- 针对特定传感器模型和参数
- 40+ Raster Type



✓ Applanix	✓ GeoEye-1	✓ Landsat 1-5	✓ Pleiades-1	✓ WorldView-1
✓ CDRG	✓ HRE	✓ TM	✓ Quickbird	✓ WorldView-2
✓ ECRG	✓ IKONOS	✓ Landsat 7	✓ RapidEye	✓ WorldView-3
✓ CIB	✓ ISAT	✓ ETM+	✓ Radarsat 2	
✓ DMCii	✓ Kompsat-2	✓ Landsat 8	✓ SOCET (SUP)	New in 10.4
✓ DTED	✓ Landsat 1-5	✓ LAS	✓ SPOT 5	✓ SPOT-7
✓ Formosat-2	✓ MSS	✓ NITF	✓ SPOT 6	✓ UAV/UAS

ArcGIS 影像信息模型

中国卫星栅格类型

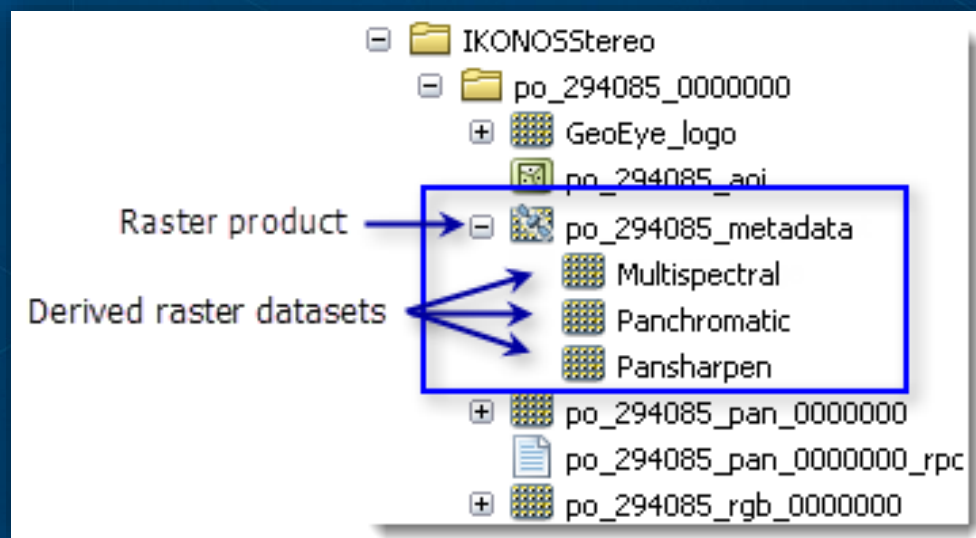
10.4版本之后无需插件

卫星	Raster Type
环境卫星	HJ 1A/1B CCD
资源一号	ZY02C HRC
	ZY02C PMS
资源三号	ZY3-CRESDA
	ZY3-SASMAC
高分一号	GF-1 PMS
	GF-1 WFV
高分二号	GF2 PMS ●
天绘一号	TH-01 ●

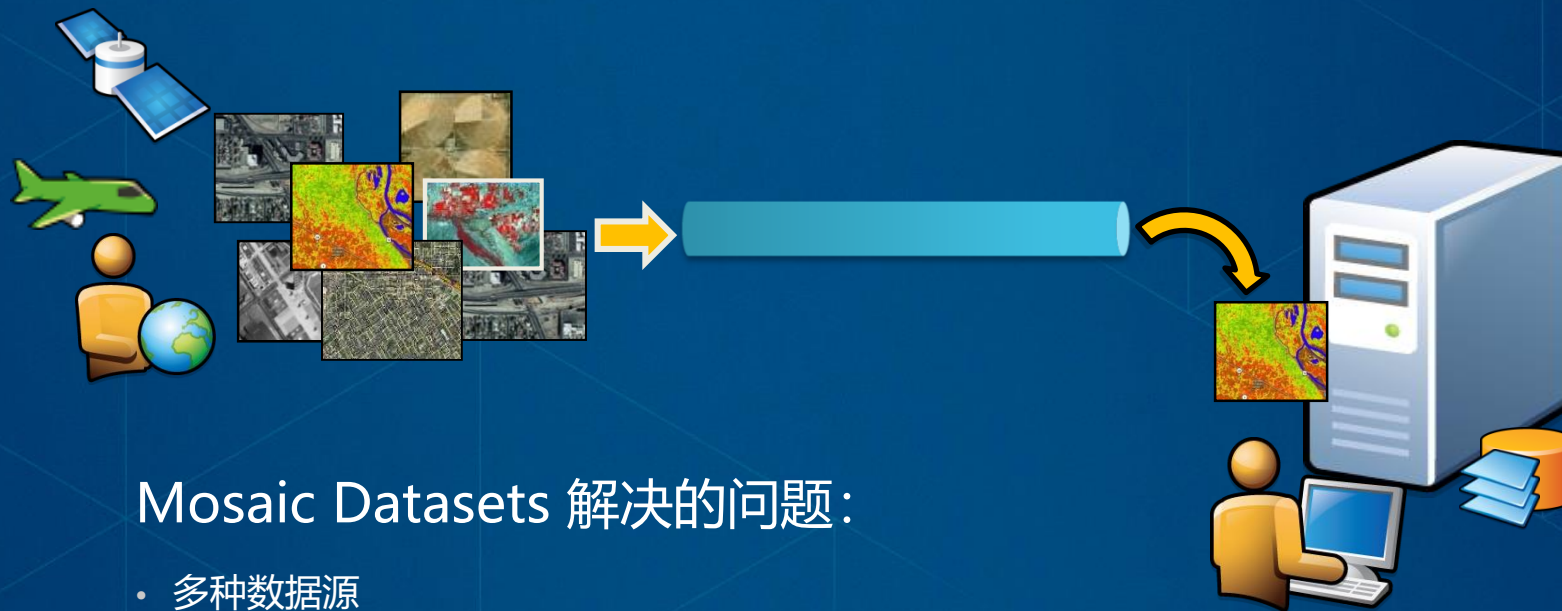
10.4 新增

Raster Product

- 基于 Raster Type
- 快速、简单地使用影像数据
- 用户无需了解影像产品中每一个文件的用途



Mosaic Dataset

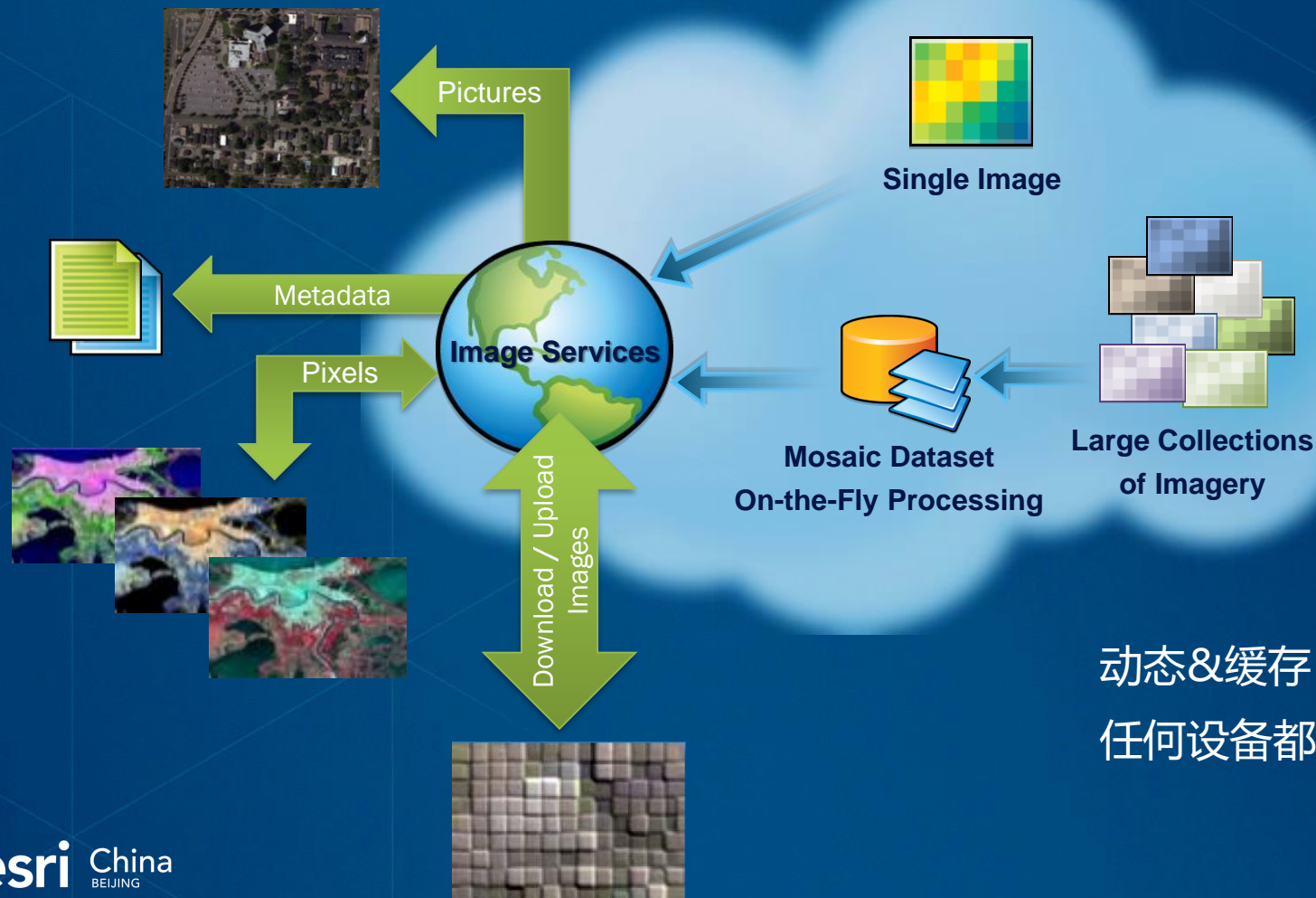


Mosaic Datasets 解决的问题：

- 多种数据源
- 多种数据格式
- 不断增加的波段个数和像元位数
- 极大规模的影像数据
- 元数据
- 多变的影像预处理流程
- 减少了中间数据
- 快速编目

ArcGIS 影像信息模型

Image Service



动态&缓存

任何设备都可以访问

ArcGIS影像信息模型的特点

- 面向对象 和 基于接口的 *Objects, Interfaces*
- 跨产品的 *Desktop, Server, Mobile,...*
- 接口丰富稳定
- 通过公开的APIs访问
 - **ArcObjects**: C/C++, C#, Java
 - **ArcPy**: Python
 - **Web API**: Javascript, Silverlight, Flex
 - **Native API**: iOS, Android, .Net



2016 Esri China
Developer Summit

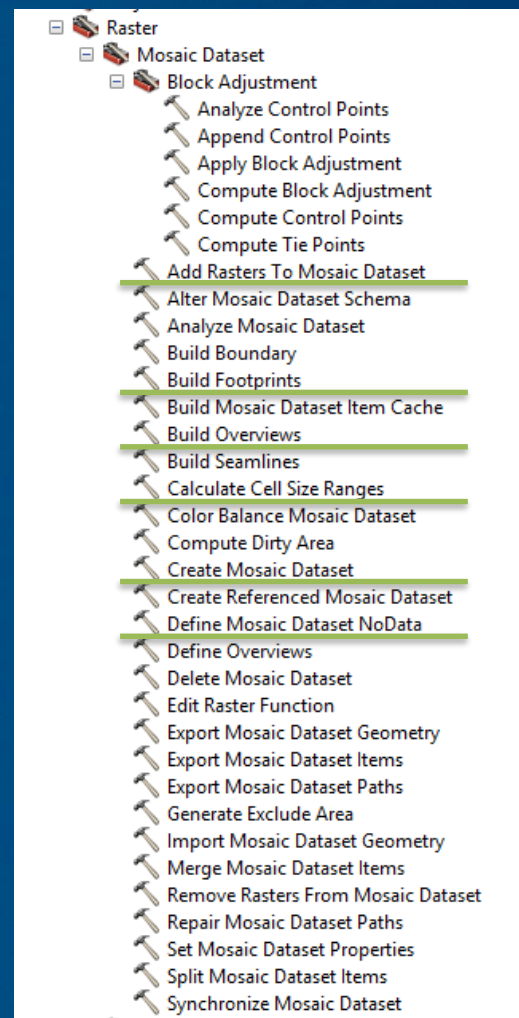
大规模影像管理解决方案

如何管理大规模影像呢？

GP工具

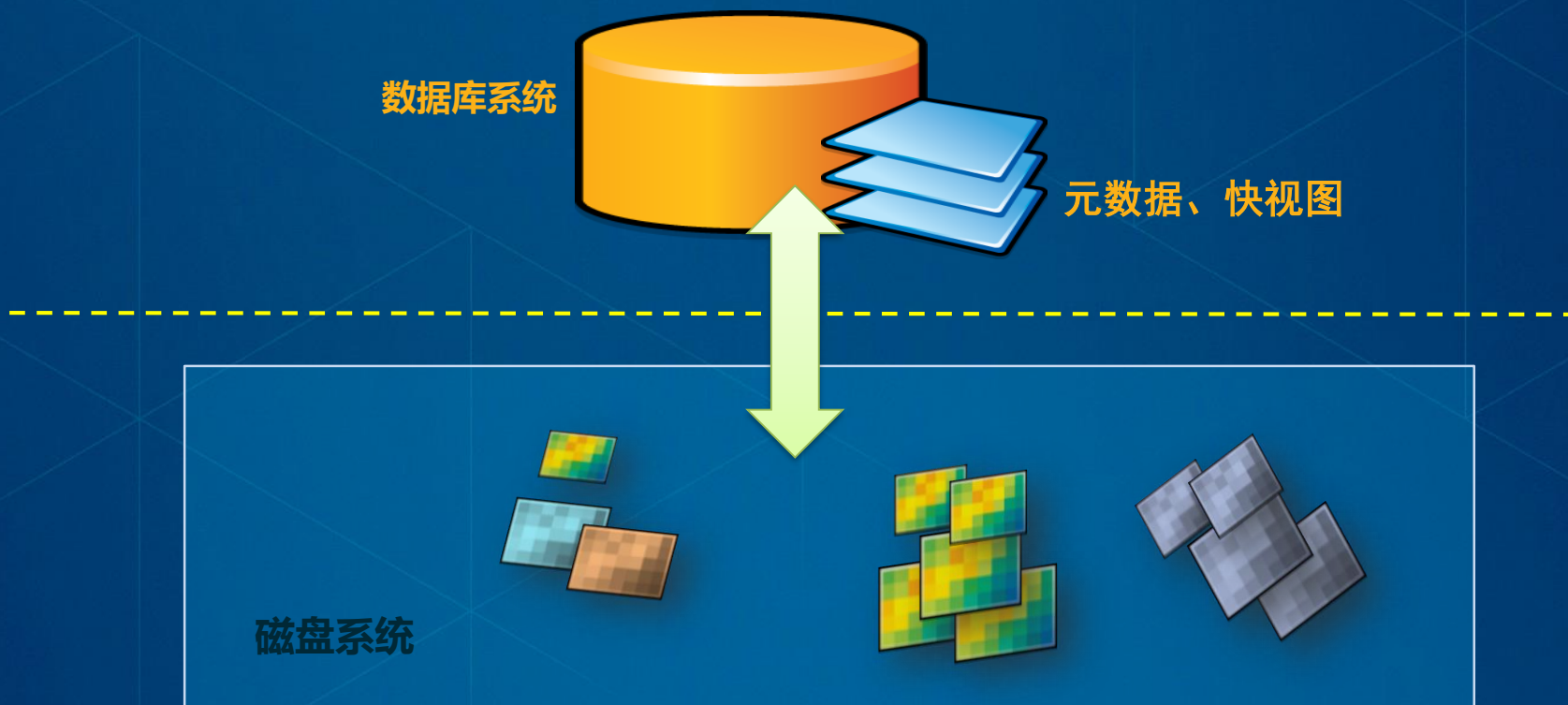


- 创建
 - Create Mosaic Dataset
 - Add Raster to Mosaic Dataset
 -
- 修改
 - Define Mosaic Dataset Nodata
 - Building Footprints
 -
- 增强
 - Build Seamlines
 - Color Balance Mosaic Dataset
 -




如何管理大规模的影像呢？

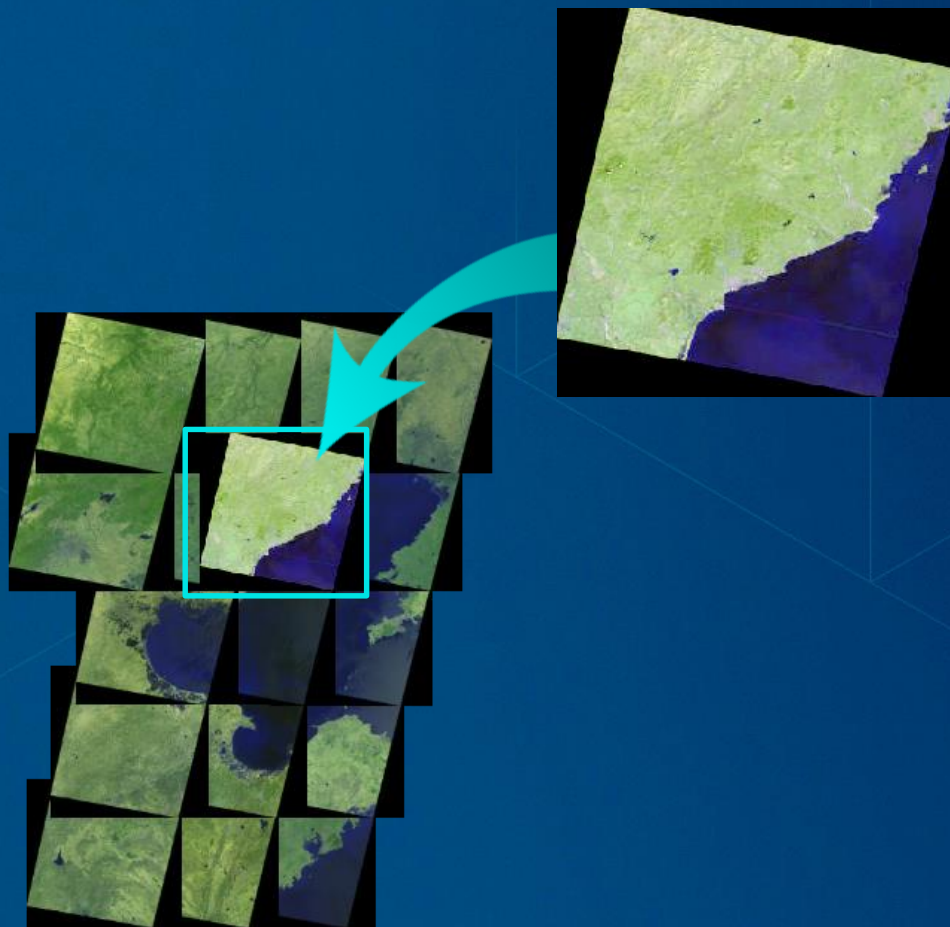
Mosaic Dataset 存储



如何管理大规模的影像呢？

典型的工作流程

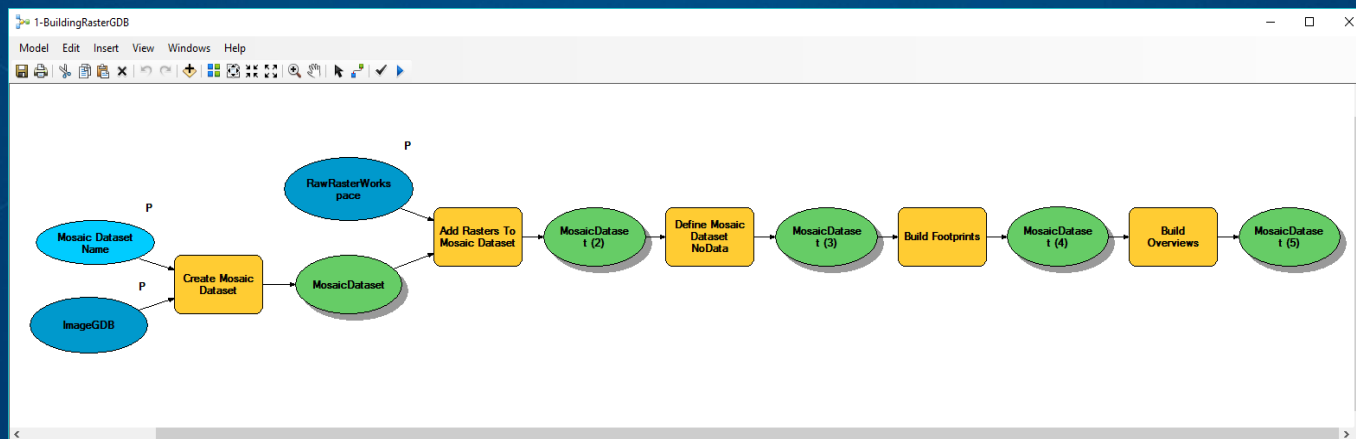
- 创建 Mosaic Dataset 
- 加入影像
 - 计算显示比例
 - 构建数据边界
- 设置无效值
- 构建轮廓线
- 构建 overview
- 编辑 属性信息



如何管理大规模的影像呢？

自动化工作流程

通过 ModelBuilder 或 Python 自动化执行



```
# 创建镶嵌数据集
arcpy.CreateMosaicDataset_management(_ImageGDB_gdb, MosaicDataset, CoordinateSystem, product_definition_="None")
print "Step1 Create Mosaic Dataset - Done"

# 加入栅格
arcpy.AddRastersToMosaicDataset_management(_MosaicDataset_,"Raster Dataset",_SourceRasterWorkspace_,"UPDATE_CELL_SIZES", "UPDATE_BOUNDARY",
                                           "NO_OVERVIEWS",_,"0",_1500,_,"SUBFOLDERS",_,"ALLOW_DUPLICATES",_,"NO_PYRAMIDS",_,"NO_STATISTICS",
                                           "NO_THUMBNAILS",_,"NO_FORCE_SPATIAL_REFERENCE",_,"NO_STATISTICS", "")
print "Step2 Add Rasters ToMosaic Dataset - Done"

# 除黑边 -- 定义无效值
arcpy.DefineMosaicDatasetNoData_management(_MosaicDataset, "4", "ALL_BANDS 0", "", "", "#,")
print "Step3 Define Mosaic Dataset NoData - Done"

# 重新构建轮廓线
arcpy.BuildFootprints_management(MosaicDataset, "", "RADIOMETRY", "1", "255", "4", "1000", "NO_MAINTAIN_EDGES",
                                "SKIP_DERIVED_IMAGES", "UPDATE_BOUNDARY", "2000", "100", "NONE", "", "20", "1")
print "Step4 Rebuild Build Footprints - Done"

# 匀色处理
arcpy.ColorBalanceMosaicDataset_management(MosaicDataset, "DODGING", "SECOND_ORDER", "", "", "NONE", "1", "")
print "Step5 Color Balancing - Done"

# 创建概视图
arcpy.BuildOverviews_management(_MosaicDataset, "", "DEFINE_MISSING_TILES", "GENERATE_OVERVIEWS",
                                "GENERATE_MISSING_IMAGES", "REGENERATE_STALE_IMAGES")
print "Step6 Build Overviews - Done"
```

ArcPy中的影像数据管理方式

ArcPy & Raster

<code>arcpy.<ToolName></code>	访问所有的栅格GP工具
<code>arcpy.ListRasters</code>	列出工作空间中的栅格数据集
<code>arcpy.Describe</code>	获取栅格数据集的通用属性 获取镶嵌数据集的默认属性
<code>arcpy.da.<cursor></code>	访问或修改镶嵌数据集的栅格项目的字段值
<code>arcpy.sa.*</code>	影像分析功能



2016 Esri China
Developer Summit

Demo

使用 Python 管理大规模影像数据

动态实时处理





2016 Esri China
Developer Summit

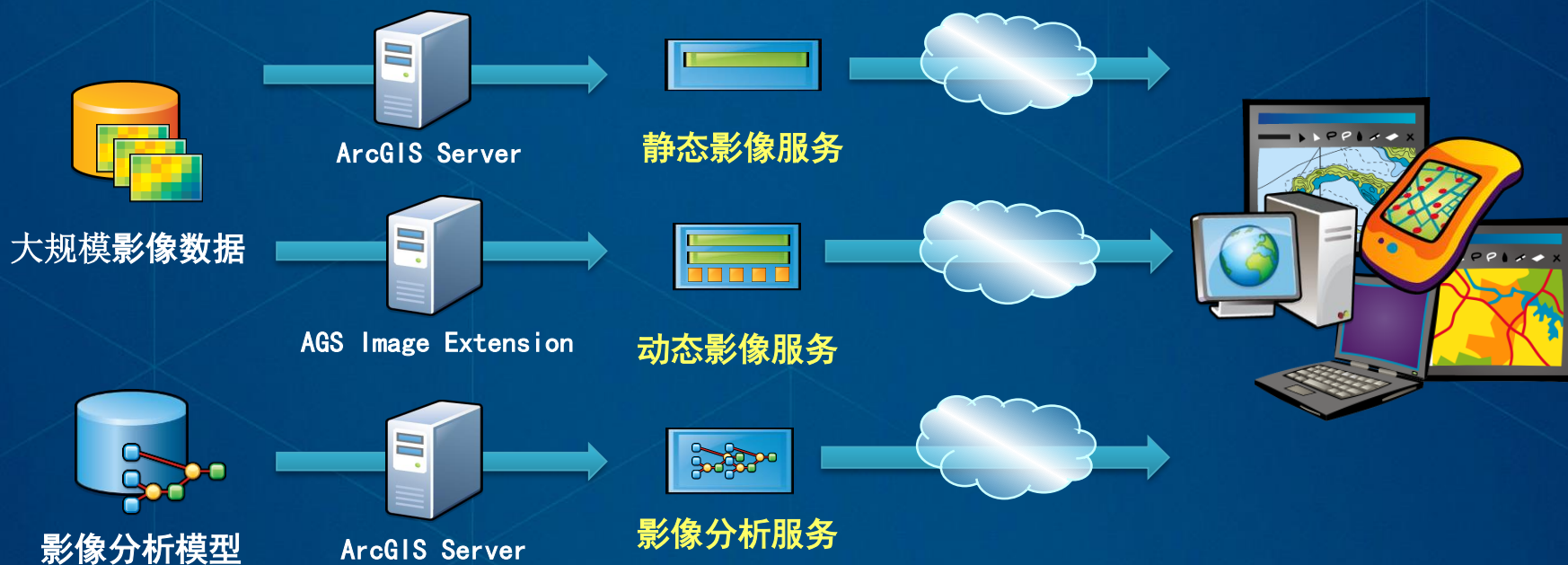
Demo 动态实时处理



2016 Esri China
Developer Summit

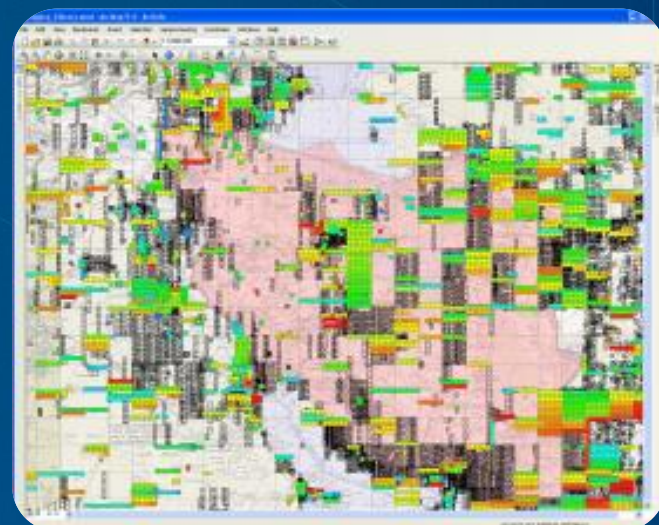
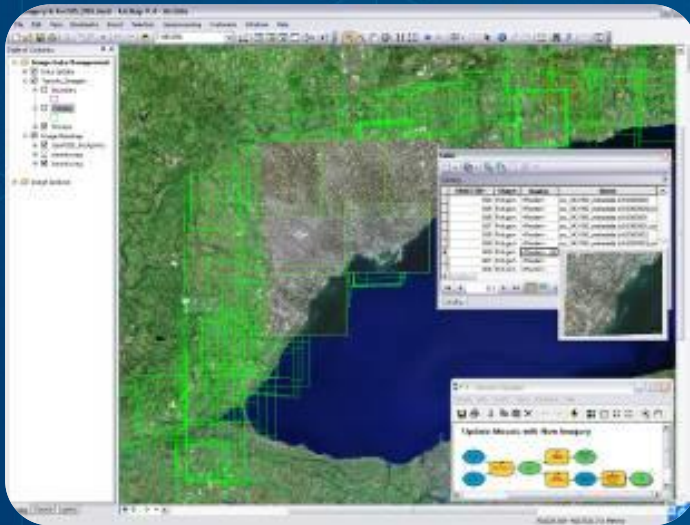
影像即服务：Image Service

任何应用都可以快速访问影像数据并进行影像处理



能用影像服务做什么？

- 使用作为一个影像 (visual analysis)
- 使用作为栅格数据 (pixel analysis)
- 使用作为一个 catalog (mosaic dataset)



影像服务能提供什么？

- 影像的快速显示
- 数据输出
 - 像素值, 原始影像, 处理的影像
- 镶嵌和编目
- 动态的影像处理
- 影像的量测
- 影像服务的编辑
 - 添加, 删除, 修改
- 支持OGC的标准 WCS, WMS, WMTS, KML

影像服务的数据源

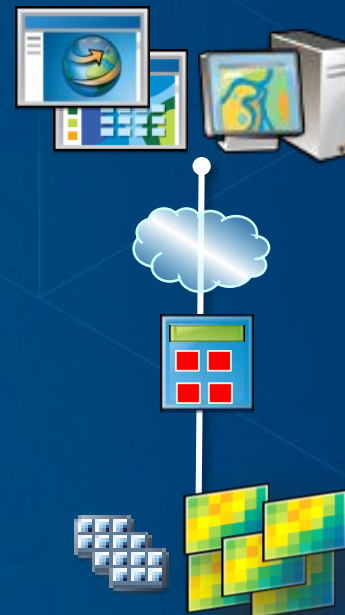
- **Raster Layer**

- 定义了 “on-the-fly” 处理
- 如果包含镶嵌函数，需要 ArcGIS Image Extension

- **Raster datasets**

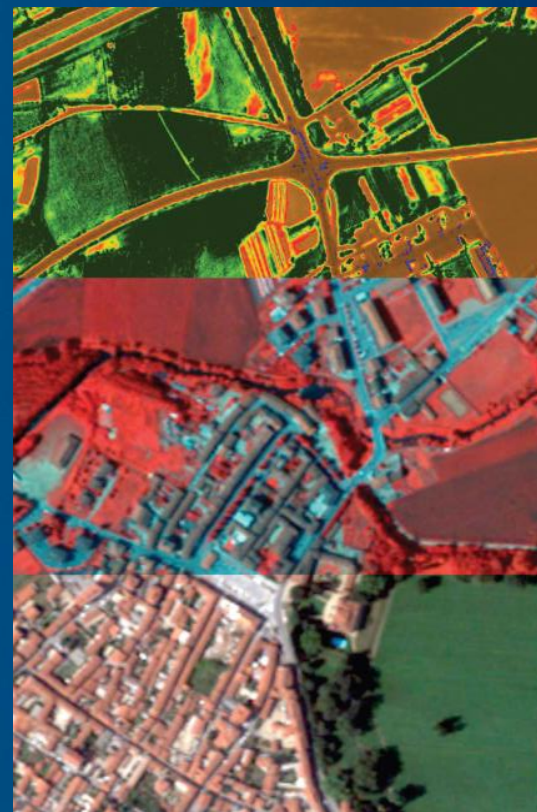
- **Mosaic datasets**

- 需要 ArcGIS Image Extension



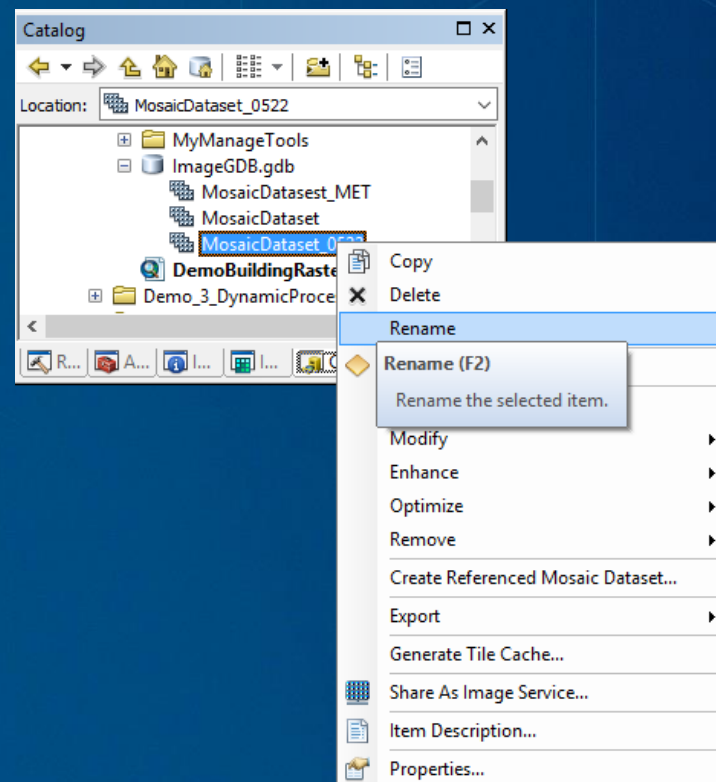
动态影像服务

- 海量影像访问、查询和分发
 - 元数据查询
 - 原始影像下载
 - 影像在线调阅
 - 服务器端动态处理
- 支持时态感知
- 作为背景底图，用于影像分析
 - 例如ArcGIS Online中的Landsat服务



如何发布影像服务?

1. 连接到server
2. 注册数据位置 (shared/duplicated)
3. 导航到相应的 dataset 并且共享为 Image Service
4. 定义相关信息
 - Server, 服务名称
5. 修改capabilities和参数
 - 一些能力是针对特定的数据的
6. *Register with ArcGIS Online
7. 分析
8. 发布



通过Python高效发布服务

Step1 创建GIS服务器连接文件

```
arcpy.mapping.CreateGISServerConnectionFile(connecttype,  
                                              out_folder_path,  
                                              out_name,  
                                              server_url,  
                                              "ARCGIS_SERVER",  
                                              use_staging_folder,  
                                              staging_folder_path,  
                                              username,  
                                              password,  
                                              "SAVE_USERNAME")
```

Step2 创建影像服务定义草稿文件(.sddraft)

```
arcpy.CreateImageSDDraft(mdpath, sddraft, service, 'ARCGIS_SERVER',  
                          copy_data_to_server=False)
```

通过Python高效发布服务

Step3 分析服务定义草稿

```
analysis = arcpy.mapping.AnalyzeForSD(sddraft)

print(u"        分析服务定义草稿结果:")
for key in list(analysis.keys()):
    print("        ---{}---".format(key.upper()))
    for (message, code), layerlist in analysis[key].items():
        print("        (CODE {}) {} ".format(code, message))
```

通过Python高效发布服务

Step4 过渡 sddraft 到服务定义文件sd

```
outSDfile = os.path.join(outdir,service+".sd")  
  
arcpy.StageService_server(sddraft,outSDfile )
```

Step5 将服务定义文件发布到服务器

```
arcpy.UploadServiceDefinition_server(inSdFile, inServer, inServiceName)
```




2016 Esri China
Developer Summit

Demo

通过Python高效发布服务

如何访问影像服务?

- ArcGIS Desktop
- Web APIs
- ArcGIS.com
- REST, SOAP
- WMS, WCS, KML
- 3rd Party Application

影像应用越来越广泛

各种各样的行业应用需求

HORIZONTAL MARKETS



VERTICAL INDUSTRIES





2016 Esri China
Developer Summit

Thanks !



The Geographic Approach



分 享 地 理 价 值

