

2011 Esri China  
Developer Summit

# 2011Esri中国开发者大会

## 在Web应用中使用 Geoprocessing服务

Esri中国（北京）有限公司 王坚

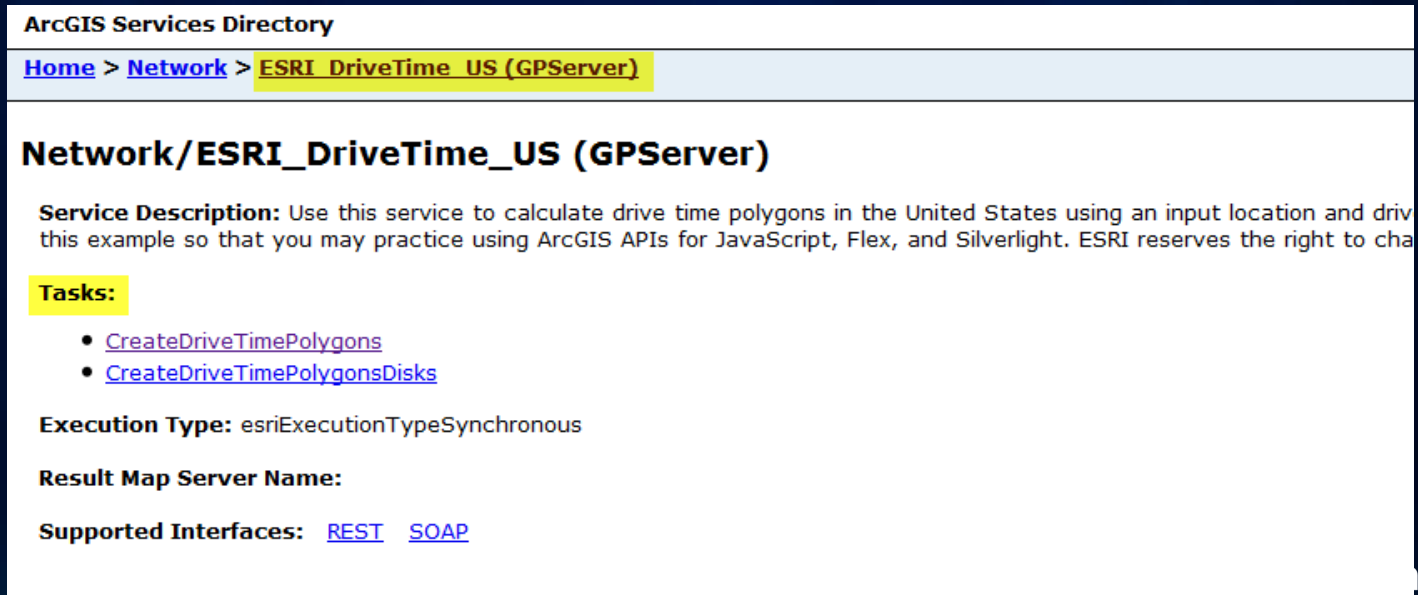
# Geoprocessing服务

- 通过ArcGIS Server把Geoprocessing 工具箱发布成 Geoprocessing服务
- 工具箱里的每个工具，脚本和模型都成为一个task



# 什么是Geoprocessing Task ?

- Geoprocessing task就是访问Geoprocessing功能的REST地址
  - 格式 : ArcGIS REST Service Directory/GPServer/<Task>
  - 比如 :



ArcGIS Services Directory

[Home](#) > [Network](#) > [ESRI DriveTime\\_US \(GPServer\)](#)

## Network/ESRI\_DriveTime\_US (GPServer)

**Service Description:** Use this service to calculate drive time polygons in the United States using an input location and drive time. This example is provided for educational purposes only. ESRI reserves the right to change this example so that you may practice using ArcGIS APIs for JavaScript, Flex, and Silverlight. ESRI reserves the right to change this example so that you may practice using ArcGIS APIs for JavaScript, Flex, and Silverlight. ESRI reserves the right to change this example so that you may practice using ArcGIS APIs for JavaScript, Flex, and Silverlight.

**Tasks:**

- [CreateDriveTimePolygons](#)
- [CreateDriveTimePolygonsDisks](#)

**Execution Type:** esriExecutionTypeSynchronous

**Result Map Server Name:**

**Supported Interfaces:** [REST](#) [SOAP](#)

# Task操作

- Task支持两种类型的操作
  - Execute : 适用于执行较快的处理
    - 执行模式 : 同步
  - Submit Job : 适用于需要较长时间的处理 ( 10秒以上 )
    - 执行模式 : 异步

**Supported Operations:** [Execute Task](#)

**Supported Operations:** [Submit Job](#)

# Task参数

- 每个Task都有自己的参数
- 参数可以分为两类：
  - 输入参数
  - 输出参数



# Task参数类型

- FeatureSet
- RecordSet(表格)
- Raster
- Linear Unit(比如100 Km,50 Miles...)
- File(.xml, .gpx, .txt, .zip ...)
- Integer, Long, Double
- String
- Date
- Boolean

# Demo:CreateDriveTimePolygons

ArcGIS Services Directory

## Task: CreateDriveTimePolygons

**Display Name:** CreateDriveTimePolygons

**Category:**

**Help URL:** [http://sampleserver1b.arcgisonline.com/arcgisoutput/Network\\_ESRI\\_DriveTime\\_US/CreateDriveTimePolygons.htm](http://sampleserver1b.arcgisonline.com/arcgisoutput/Network_ESRI_DriveTime_US/CreateDriveTimePolygons.htm)

**Execution Type:** esriExecutionTypeSynchronous

**Parameters:**

**Execution Type:** esriExecutionTypeSynchronous

参数1 :

参数2 :

参数3 :

**Parameter:** Input\_Location

**Parameter:** Drive\_Times

**Parameter:** Output\_Drive\_Time\_Polygons

**Data Type:** GPFeatureRecordSetLayer

**Display Name:** Output Drive Time Polygons

**Direction:** esriGPPParameterDirectionOutput

**Default Value:**

**Geometry Type:** esriGeometryPolygon

**Spatial Reference:** 4326

**Fields:**

- FID (Type: esriFieldTypeOID, Alias: FID)
- Shape (Type: esriFieldTypeGeometry, Alias: Shape)
- FacilityID (Type: esriFieldTypeInteger, Alias: FacilityID)
- Name (Type: esriFieldTypeString, Alias: Name)
- FromBreak (Type: esriFieldTypeDouble, Alias: FromBreak)
- ToBreak (Type: esriFieldTypeDouble, Alias: ToBreak)
- Shape\_Length (Type: esriFieldTypeDouble, Alias: Shape\_Length)
- Shape\_Area (Type: esriFieldTypeDouble, Alias: Shape\_Area)

**Parameter Type:** esriGPPParameterTypeRequired

**Category:**



享地理价值

# Web应用里的Task

```
function onMapClick(event) {  
    var map = new esri.Map("map");  
    var tl = ledmapServiceLayer = new  
    esri.Layer({url: "http://services.esri.com/arcgis/rest/services/Esri_Imagery_Series/ImageServer",  
    map: map, addLayer: tl, ledmapServiceLayer});  
    function getDriveTimePolys(results) {  
        var features = results[0].features;  
        for (var f=0, l=features.length; f<l; f++) {  
            var feature = features[f];  
            if (feature.attributes["type"] == "Road") {  
                var polys = feature.attributes["polys"];  
                var polySymbol = new esri.PolySymbol({color: "#FF0000", width: 2});  
                new dojo.Color(polySymbol.getColor(), 0, 0, 0, 0.5);  
                polysymbol.setColor(polySymbol.getColor());  
                feature.setSymbol(polySymbol);  
            }  
            else if (feature.attributes["type"] == "Water") {  
                polysymbol.setColor(polySymbol.getColor(), 0, 0, 0, 0.5);  
                polysymbol.setWidth(2);  
                feature.setSymbol(polySymbol);  
            }  
        }  
    }  
    new dojo.Color(polySymbol.getColor(), 0, 0, 0, 0.5);  
    polysymbol.setColor(polySymbol.getColor());  
    polysymbol.setWidth(2);  
    feature.setSymbol(polySymbol);  
}
```



# Web应用里的Task

- Geoprocessing Tasks可以在所有的ArcGIS Web API中使用，步骤如下
  - 初始化Tasks
  - 设置参数并执行Task
  - 获取结果



# Flex 代码

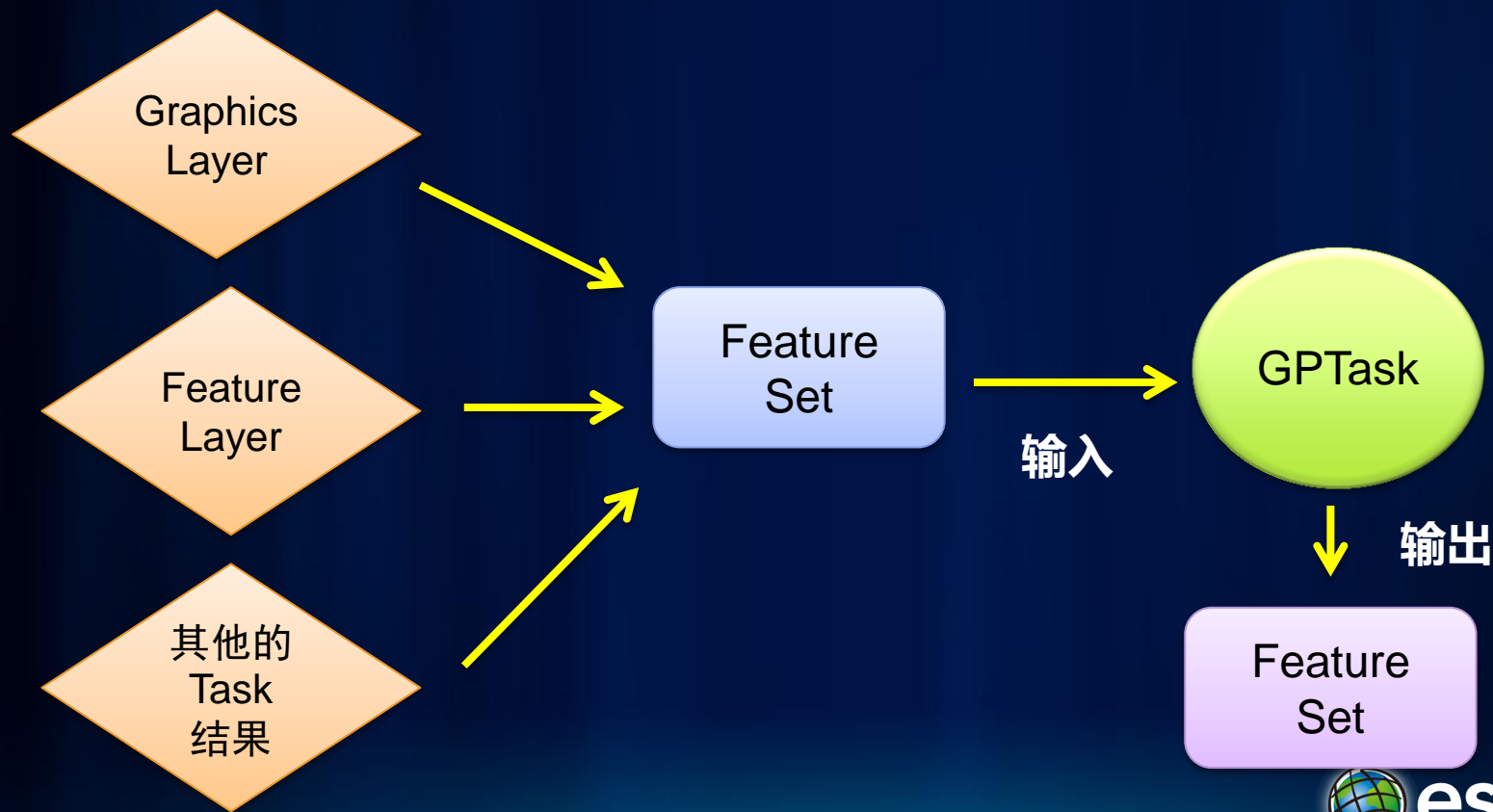
- 1、初始化GP Task
- 2、设置参数并执行
- 3、获取结果并显示

```
function onResult(gpResult:ExecuteResult,token:Object = null):void
{
    var pv:ParameterValue = gpResult.results[0];
    var fs:FeatureSet = pv.value as FeatureSet;
    graphicsLayer.graphicProvider = fs.features;
}
...
<esri:UniqueValueRenderer
id="uniqueValueRenderer" attribute="ToBreak">
    <esri:UniqueValueInfo symbol="{rFill}" value="1"/>
    <esri:UniqueValueInfo symbol="{gFill}" value="2"/>
    <esri:UniqueValueInfo symbol="{bFill}" value="3"/>
</esri:UniqueValueRenderer>
<esri:GraphicsLayer id="graphicsLayer"
renderer="{uniqueValueRenderer}"/>
```



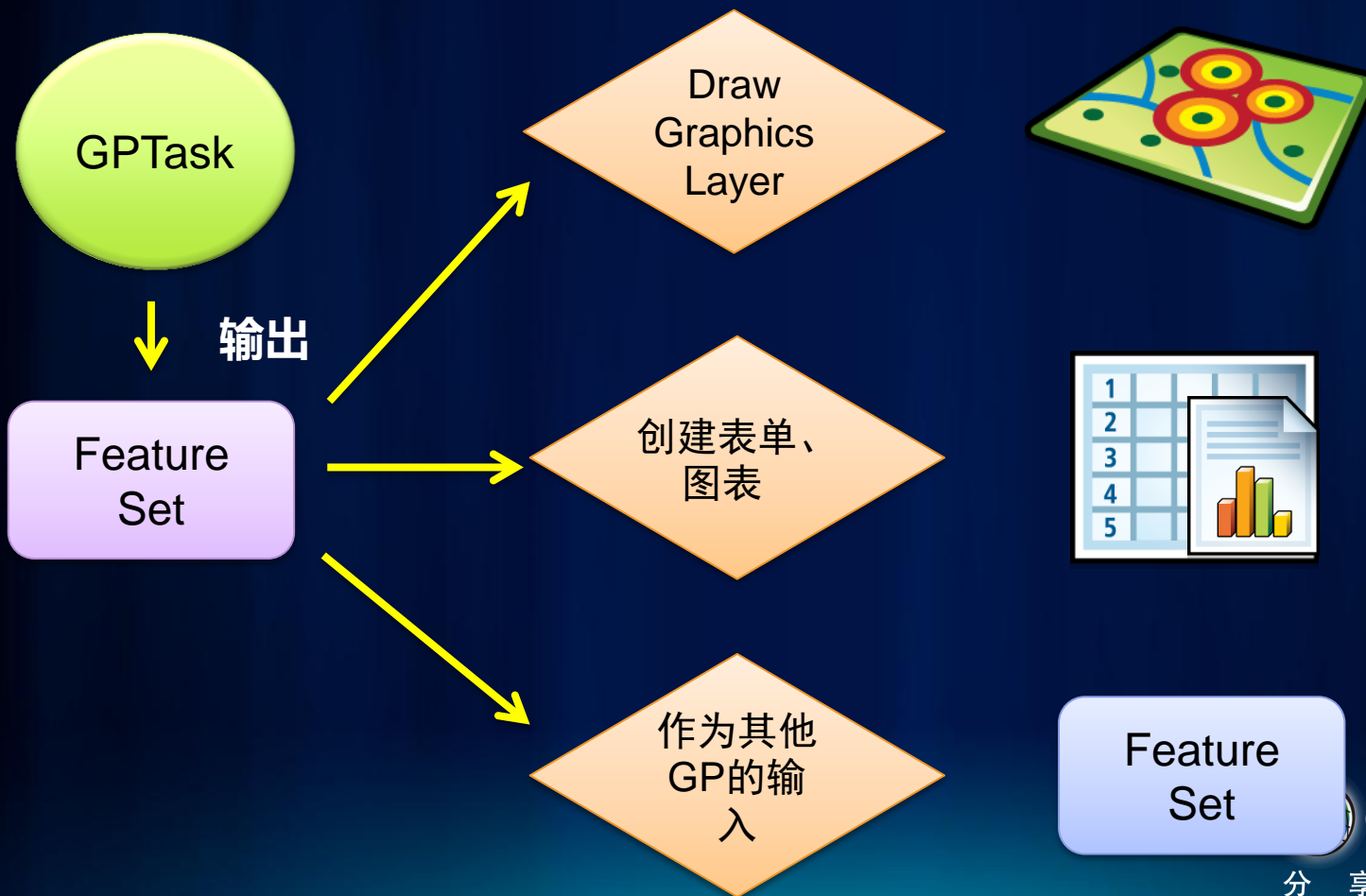
# Web应用模式1

- 输入FeatrueSet ， 输出FeatureSet

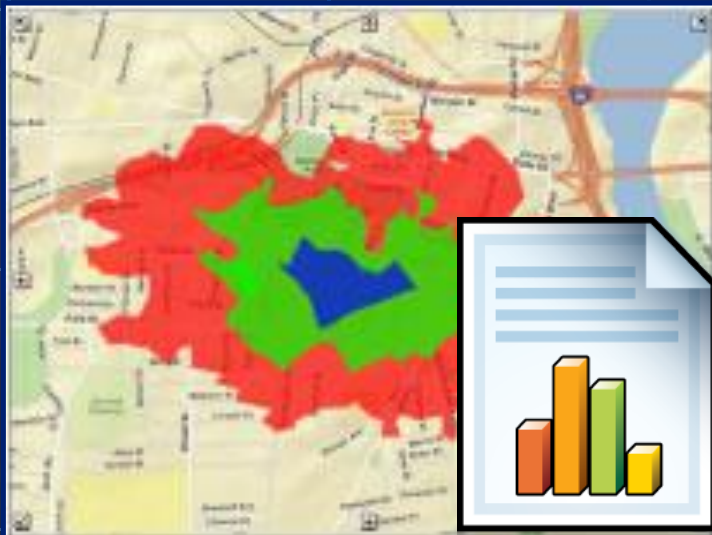


# Web应用模式1

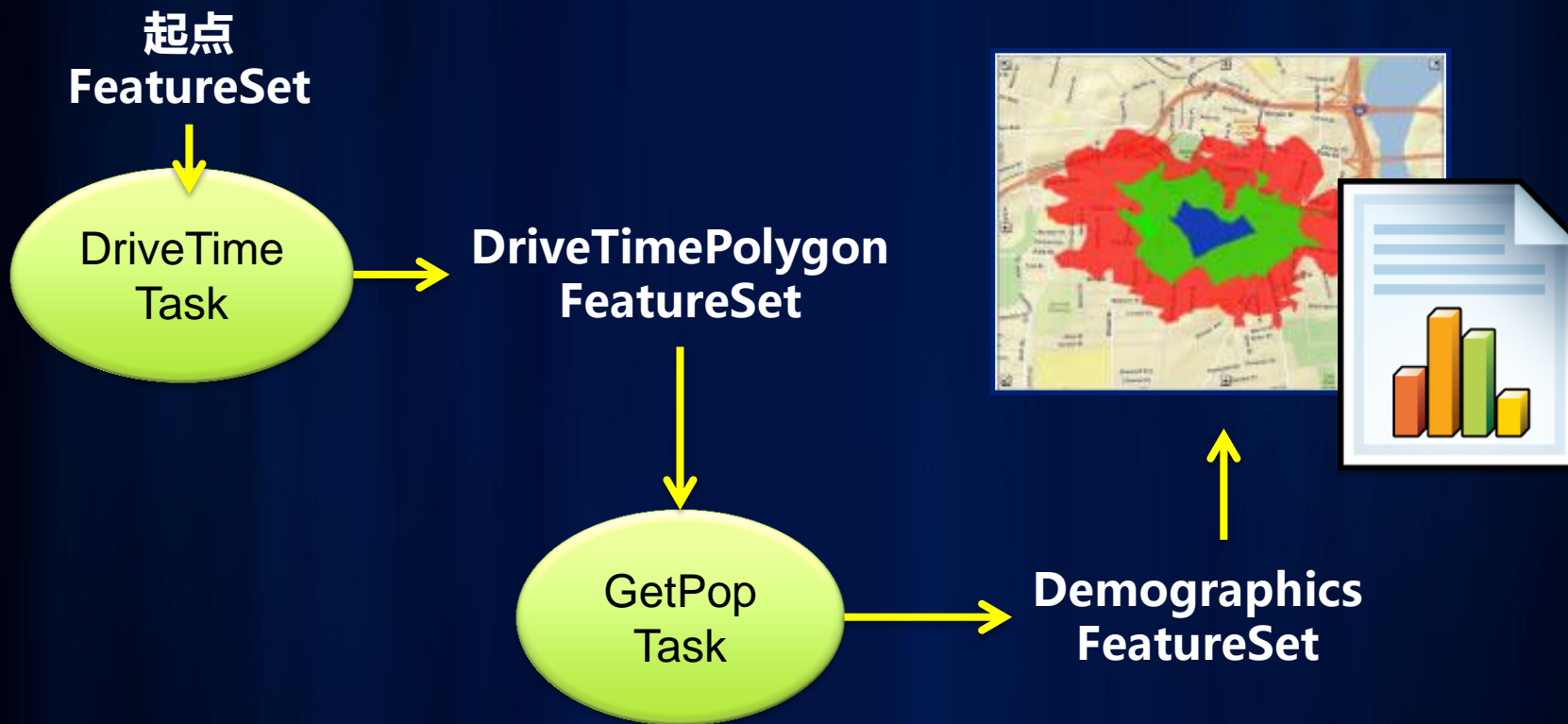
- 输入FeatrueSet ， 输出FeatureSet



# Demo: DriveTimePolygon



# Demo流程图



# 小结

- Geoprocessing Task
- Web程序模式1
  - 输入FeatureSet
  - 输出FeatureSet

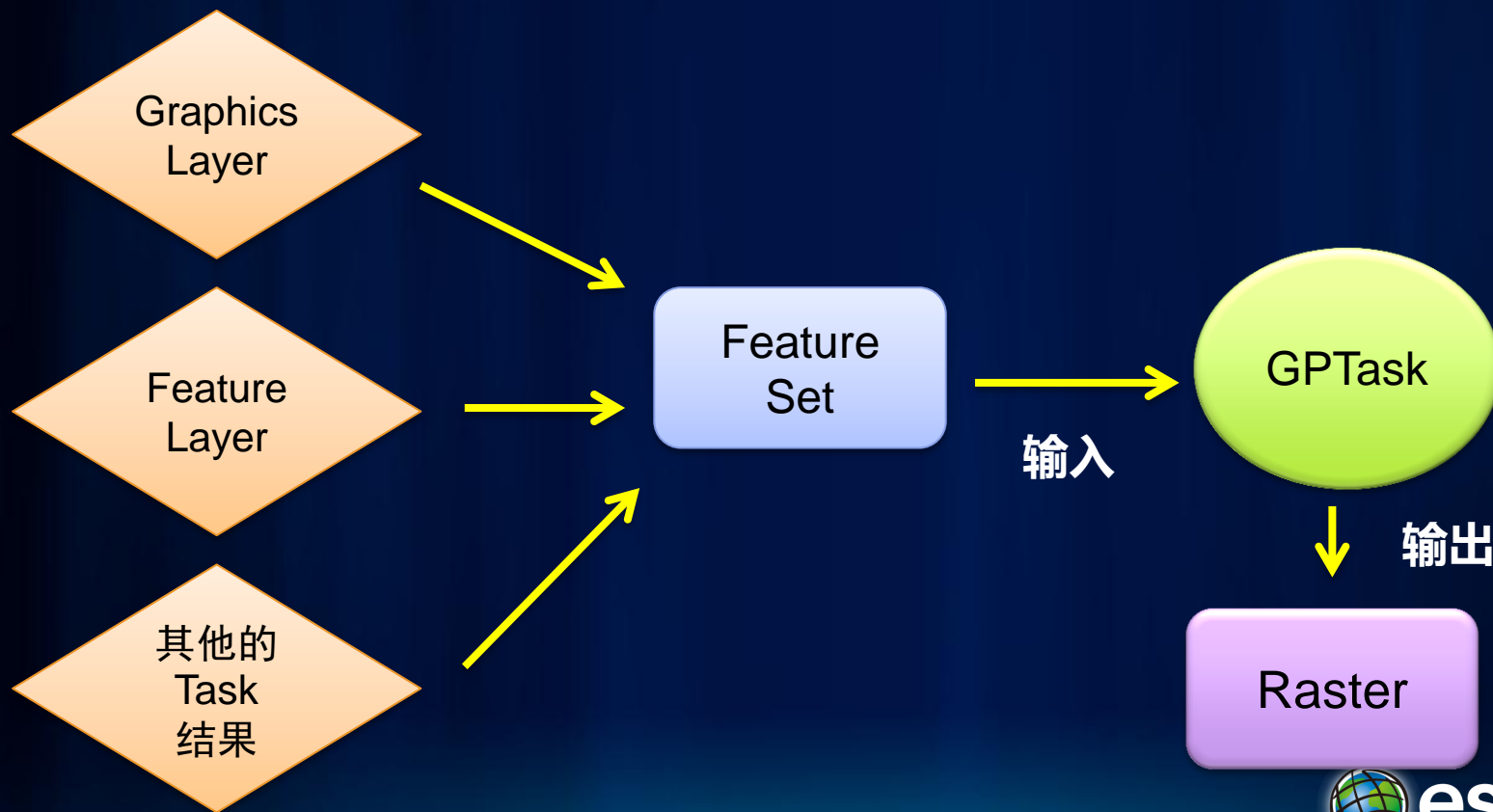


执行方式：同步  
支持的操作：Execute



## Web应用模式2

- 输入FeatureSet，输出Raster



# Web应用模式2

- 如何在网络地图上添加Raster？
  - 问题：Web API不能绘制栅格数据
  - 解决办法：使用Result Map Server

ArcGIS Services Directory

[Home](#) > [SanFranciscoCrime \(GPServer\)](#)

## SanFranciscoCrime (GPServer)

**Service Description:**

**Tasks:**

- [CrimeHotSpot](#)
- [CrimeHotSpotQuery](#)

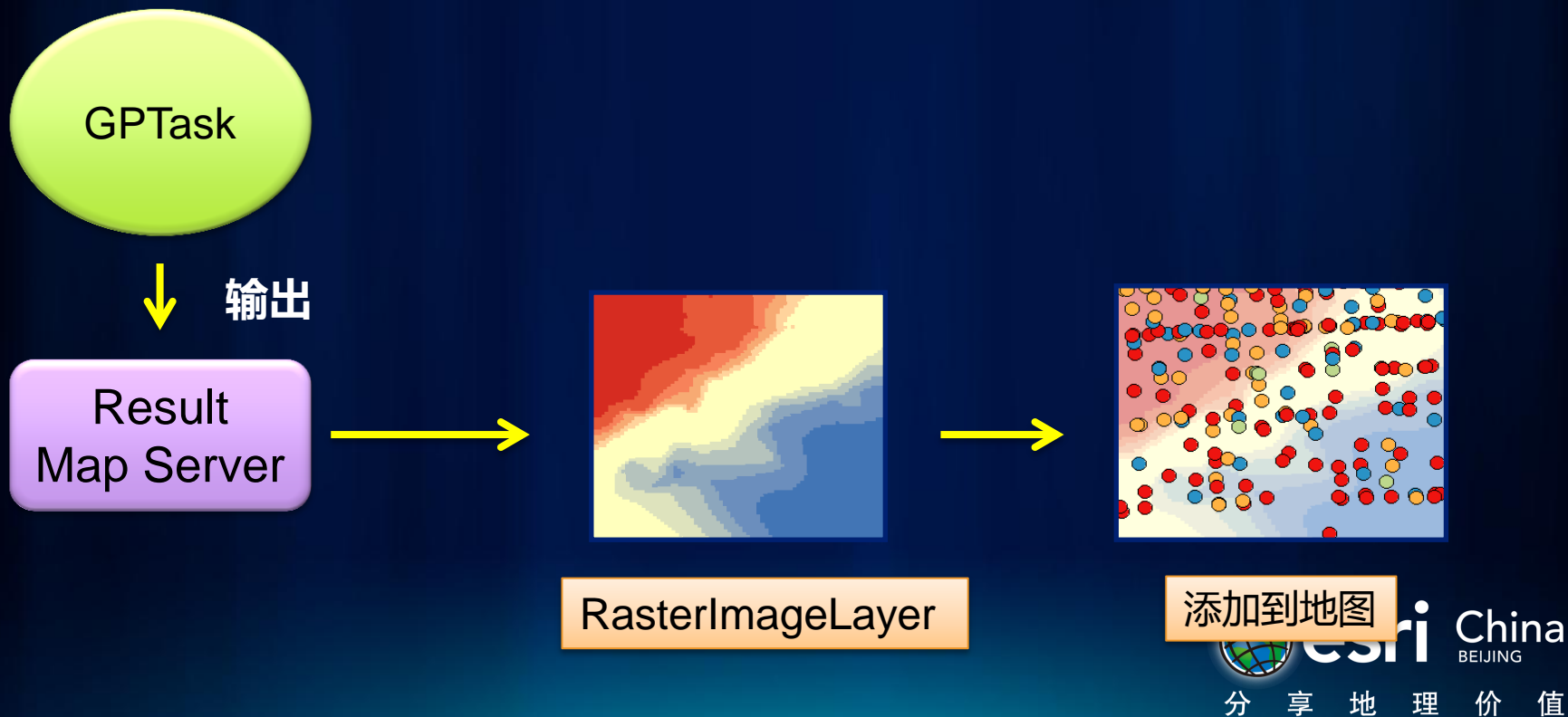
**Execution Type:** esriExecutionTypeAsynchronous

**Result Map Server Name:** SanFranciscoCrime

**Supported Interfaces:** [REST](#) [SOAP](#)

## 结果作为地图服务

- Result Map Server返回一个Image图层
- 把这个图层添加到地图上用来显示结果



# Flex代码

1、初始化GP Task

2、设置参数并执行GP Task

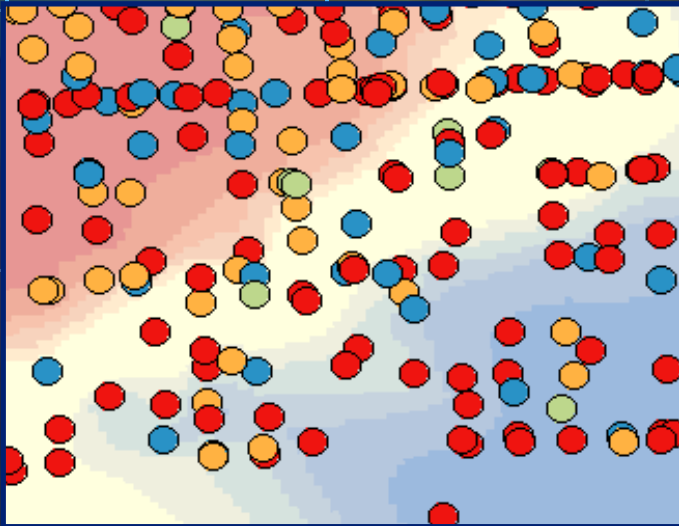
3、执行成功后，请求Result Map Server，作为ImageLayer

```
private function onResult(event:JobInfo):void{  
    var jobId:String = event.jobId;  
    var jobStatus:String = event.jobStatus;  
    if(jobStatus==JobInfo.STATUS_SUCCEEDED){  
        var imgLyr:GPResultImageLayer =  
crimeGP.getResultImageLayer(jobId,"Hotspot");  
    }  
}
```

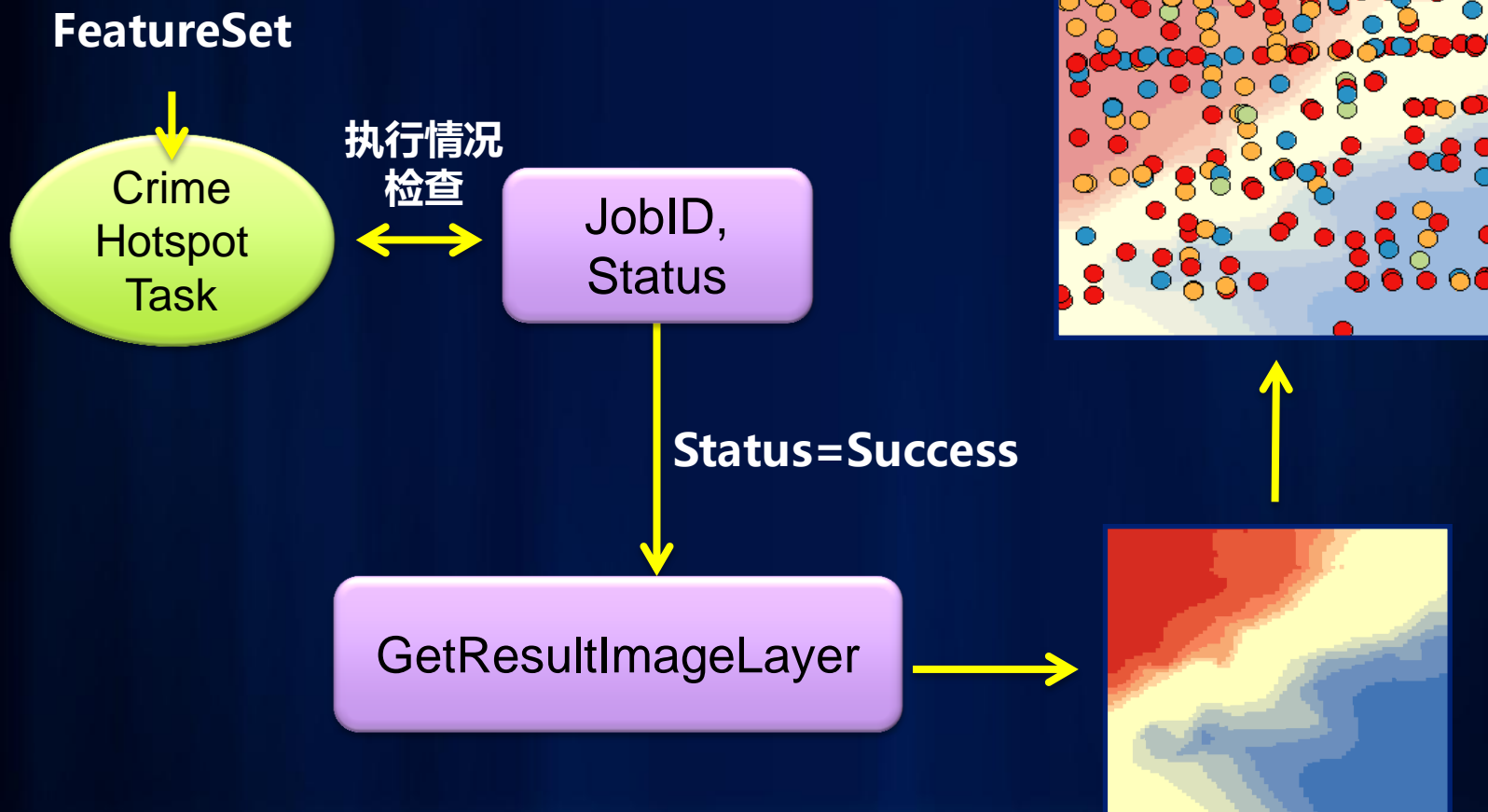
4、添加Image Layer

```
myMap.addLayer(imgLyr);
```

## Demo: CrimeHotSpot



# Demo流程图



# 小结

- Geoprocessing Task
- Web程序模式1
  - 输入FeatureSet
  - 输出FeatureSet
- Web程序模式2
  - 输入FeatureSet
  - 输出栅格
    - 返回Result Map Server
    - 获取Image Layer
    - 添加到地图

执行方式：同步  
支持的操作：Execute

执行方式：异步  
支持的操作：Submit Job

# Result Map Server

```
function onInit() {  
    var map = new esri.Map("map");  
    var tl = ledmapServiceLayer = new  
    esri.Layer({url: "http://www.esri.com/arcgis/rest/services/ESRI_Imagery/MapServer",  
    title: "ESRI Imagery",  
    type: "Image"});  
    map.addLayer(tl);  
}  
  
function getDriveTimePolys(results) {  
    var features = results(0).features;  
    for (var f=0; f<features.length; f++) {  
        var feature = features[f];  
        if (feature.type === "Polygon") {  
            var polys = feature.geometry.paths;  
            for (var p=0; p<polys.length; p++) {  
                var poly = polys[p];  
                var color = poly.symbol.color;  
                new dojo.Color({r: 0, g: 0, b: 50}).  
                setAlpha(0.5);  
                polys[p].symbol = new esri.Symbol({color: color, type: "Solid"});  
            }  
        }  
    }  
}
```

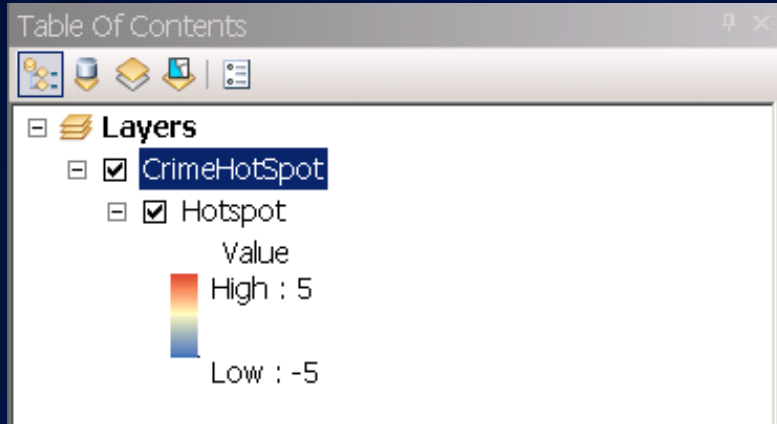


# GP的结果作为地图服务

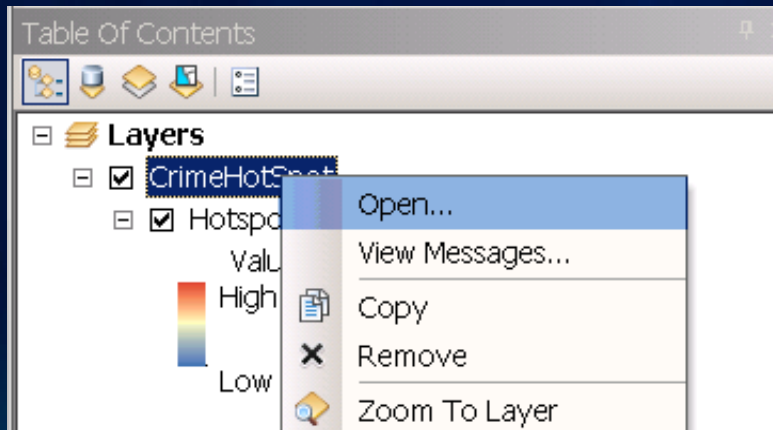
- 用这种方式的三个理由：
  - Web客户端无法绘制结果（比如栅格）
  - 输出数据太大不适合下载
  - 更好的绘图效果（可以使用各种符号）
- 如何发布这样的工具？
  - 在ArcMap里创建工具图层
  - 保存mxd
  - 使用“Publish to ArcGIS Server”发布成地图服务
  - 勾选“Geoprocessing”功能

# 工具图层

- 第一步：把工具或模型拖到图层列表

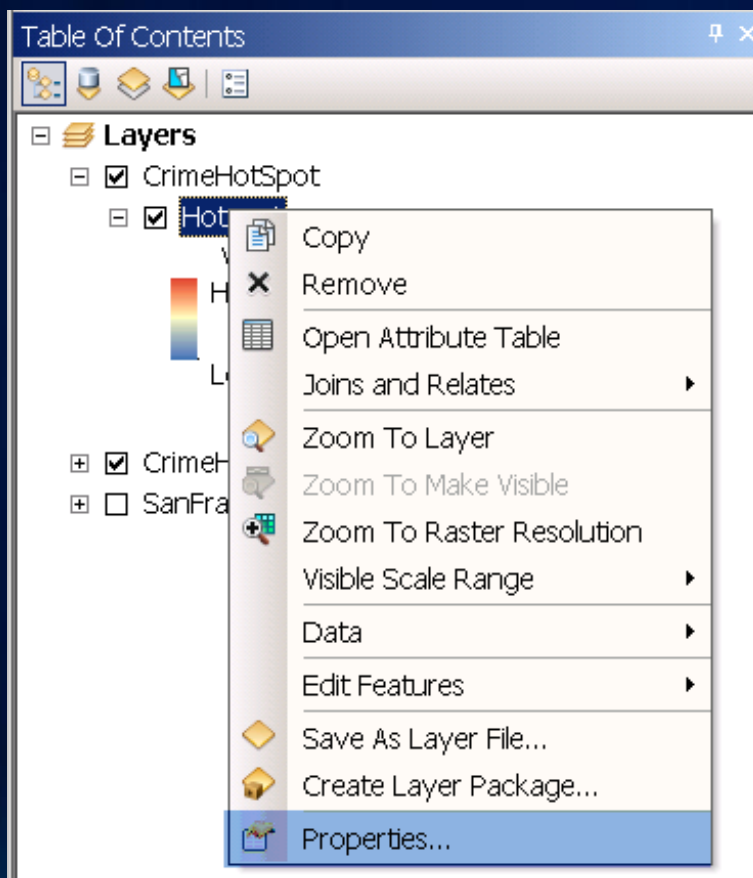


- 第二步：打开工具并运行



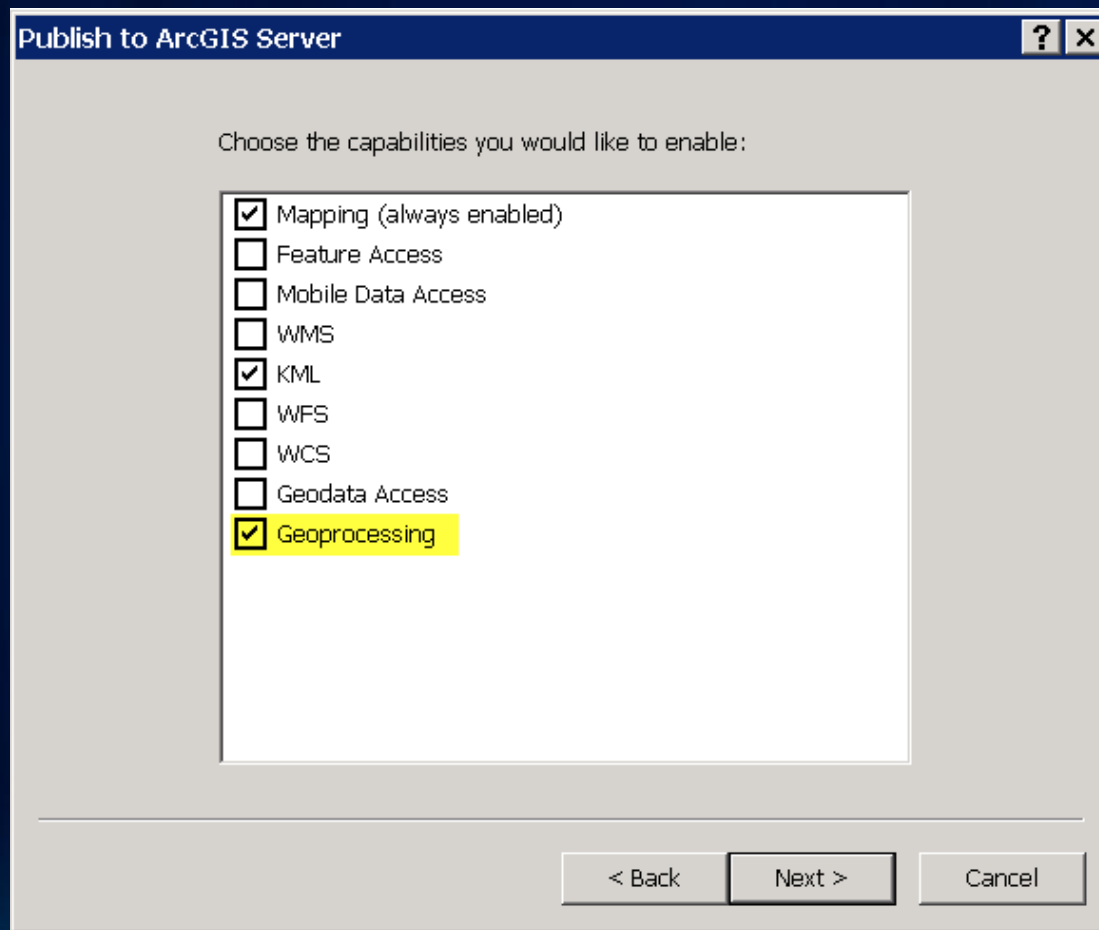
# 工具图层

- 第三步：设置结果的符号



# 工具图层

- 第四步：发布成地图服务，勾选Geoprocessing功能



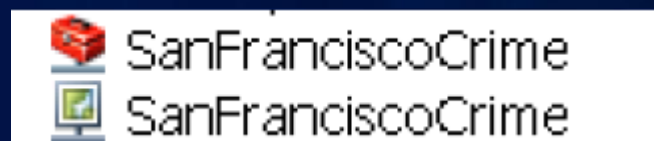
## 发布带工具图层的地图服务

- 会创建一个Geoprocessing服务和一个Result Map Service
- 地图文档里的工具图层就是**Geoprocessing task**

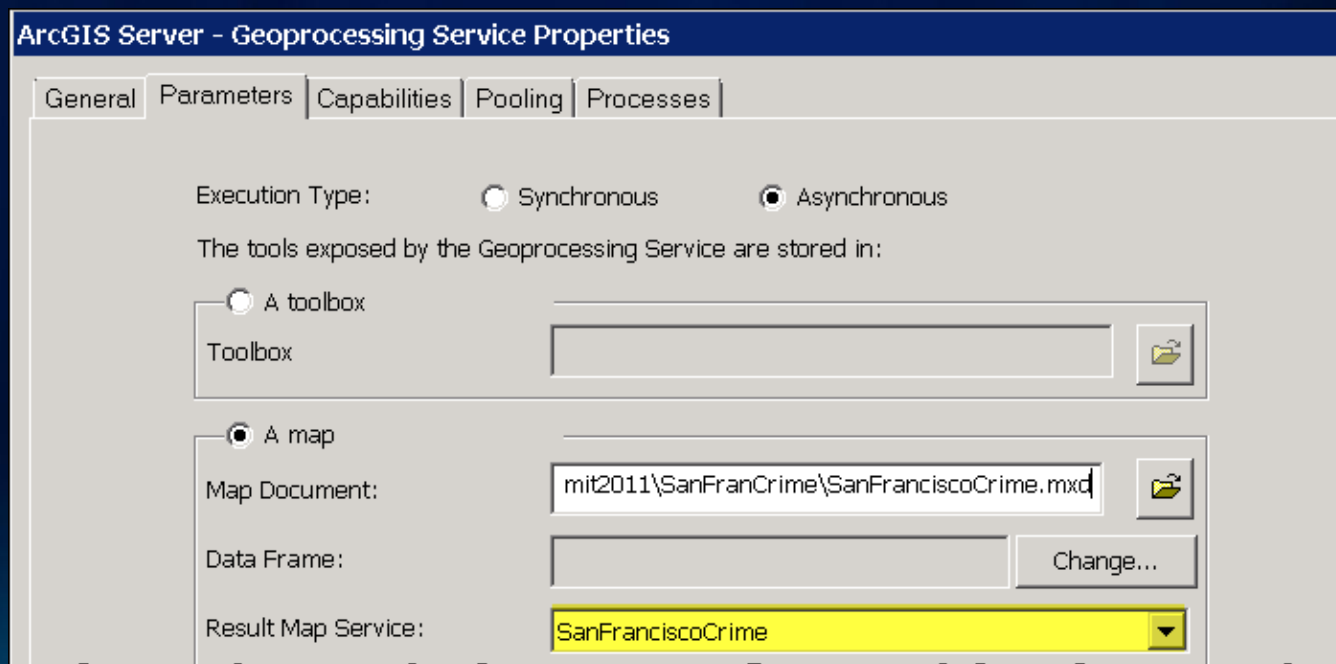


# 发布工具图层

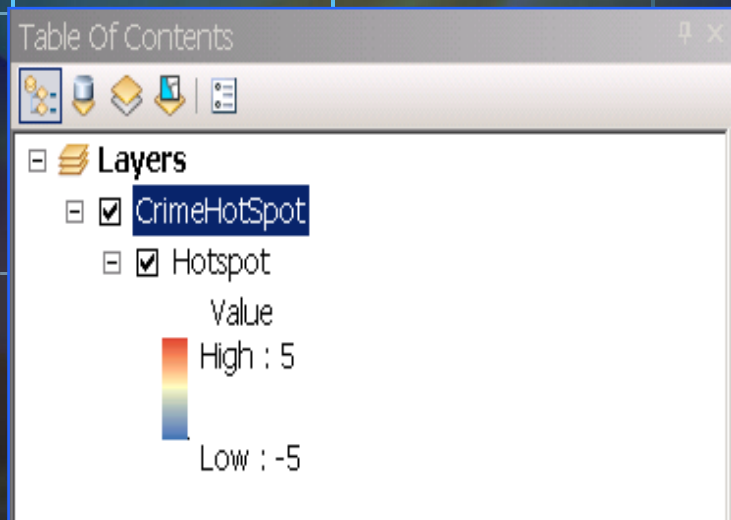
- Geoprocessing服务和地图服务是同名的



- Geoprocessing服务的属性对话框里可以看到Result Map Service



# Demo: 工具图层



# 小结

- Geoprocessing Task
- Web程序模式1
  - 输入FeatureSet
  - 输出FeatureSet
- Web程序模式2
  - 输入FeatureSet
  - 输出栅格
    - 返回Result Map Server
    - 获取Image Layer
    - 添加到地图
- 发布工具图层

执行方式：同步  
支持的操作：Execute

执行方式：异步  
支持的操作：Submit Job



# 模式3

- 输入文件，输出文件

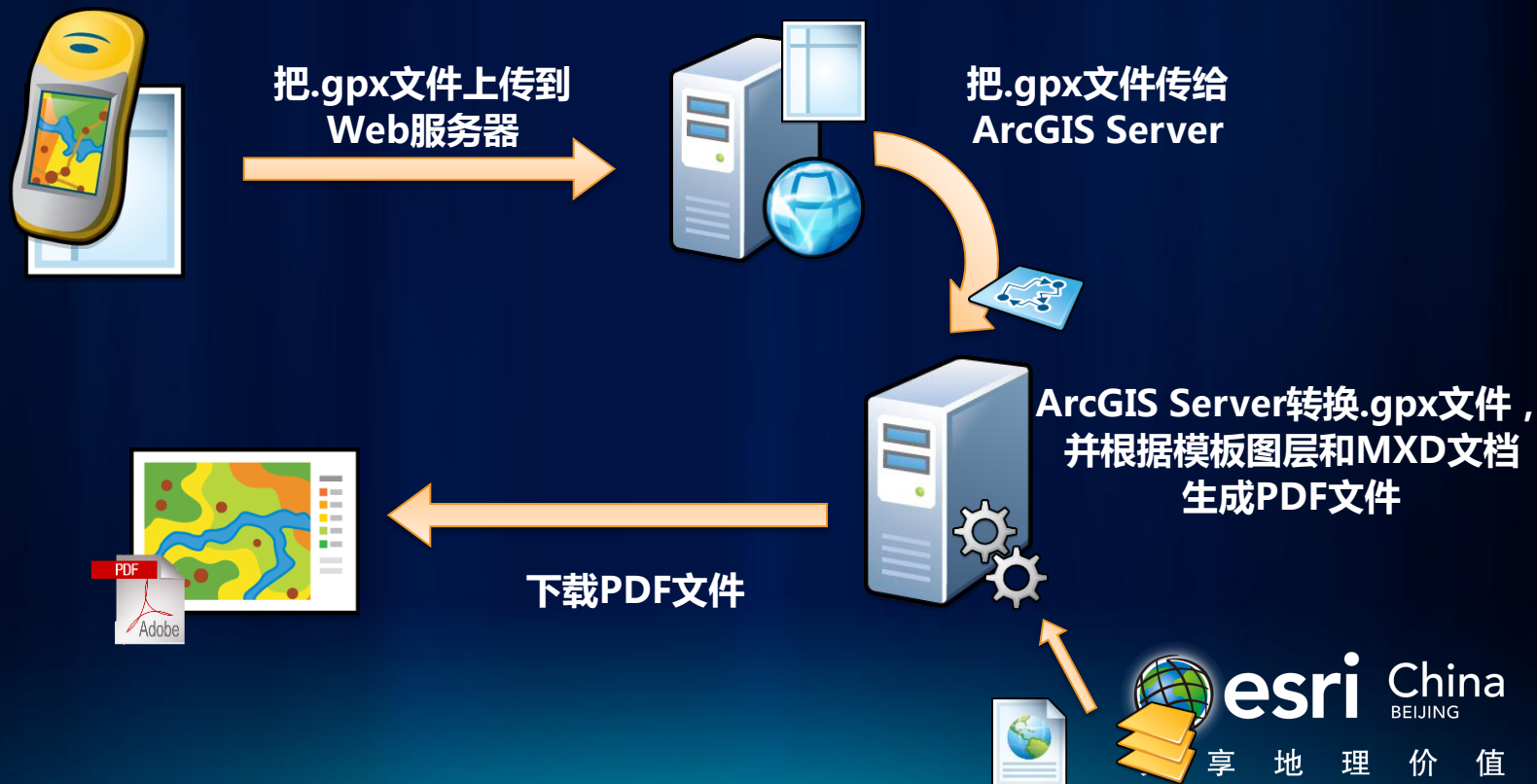


# 把GPX文件转换成PDF

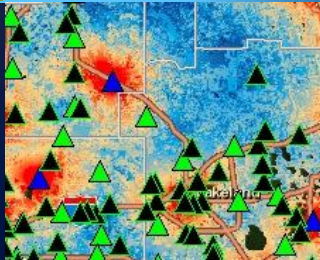
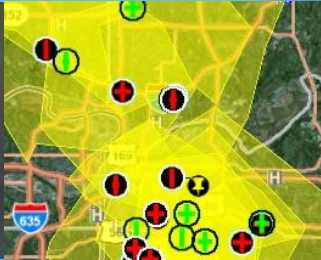
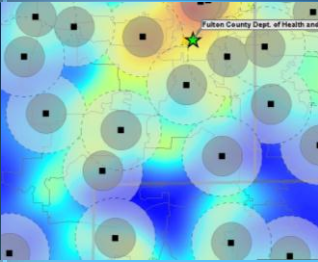
```
function print() {  
    var map = new esri.Map("map");  
    var tl = ledmapService.layer = new  
    esri.Layer({url: "http://services.esri.com/arcgis/rest/services/Esri_Imagery  
    /MapServer", map: .addLayer(tl ledmapService)});  
}  
  
function getDriveTimePolys(results) {  
    var features = results[0].features;  
    for (var f=0; f<features.length; f++) {  
        var feature = features[f];  
        if (feature.type === "Polygon") {  
            var polys = feature.geometry.paths;  
            for (var i=0; i<polys.length; i++) {  
                var poly = polys[i];  
                var polySymbol = new esri.PolySymbol({color: "red", width: 5});  
                new dojo.Color(polySymbol).setRGB(0, 0, 0.5);  
                feature.setSymbol(polySymbol);  
            }  
        }  
    }  
    new dojo.Color(polySymbol).setRGB(0, 0, 0.5);  
    feature.setSymbol(polySymbol);  
}
```

# 概要

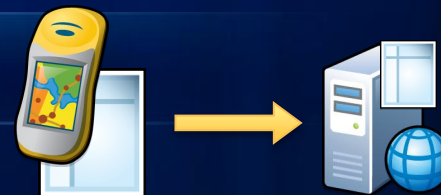
- 处理文件上传
- Python脚本把GPX转换成要素
- 用arcpy.Mapping创建高质量的PDF



# Demo:GPX转换成PDF



# 文件上传



- 文件上传可以由多种方式实现
  - PHP , ASP.NET , FTP , JSP等
- Demo实现方式
  - jQuery FileUploader ( AJAX方式 )
  - JSP

# GPX文件转换成要素



- Python脚本读取GPX文件并转换成要素类
  - 通过Python的“elementtree”读取XML
  - 通过arcpy.InsertCursor用X/Y坐标构建要素
- 输入：文件名（字符串）
  - 脚本里通过硬编码的方式设置了文件上传的路径，因此只需要文件名就可以构建完整的路径
- 其他格式的转换也是类似的（比如KML，或其他文本）

# arcpy.Mapping

- 可以打开和操作MXD和图层文件的模块
- 自动创建和输出地图
- 批量更新数据源
- 可以设置输出和渲染让这个模块在GP服务中发挥强大的作用

## arcpy.Mapping输出PDF



- 预先创建好MXD布局和图层文件，只要输入数据就可以生成想要的结果
  - 可以在服务中使用制图效果
- 可以根据服务的输入参数选择合适的MXD和图层文件
- 可以在地图中插入标题、图层名和备注
- 客户端可以下载输出的PDF



## 更多的参考资料

- 本地工作目录:
- <http://esriurl.com/localjob> 搜索: "local job"
- 输入、输出的参数类型:
- <http://esriurl.com/inNout> 搜索: "input output"
- 编辑GP服务的内容列表:
- <http://esriurl.com/2280> 搜索: "Checklist for authoring"



分 享 地 理 价 值