

2010 Esri 中国区域用户大会



ArcGIS Server 10 SOE 新特性 及应用开发

秦桐

讲座提纲

1. SOE概念模型及发展历程
2. SOE的运行机制
3. SOE的开发方法
4. SOE的实际意义

概念的SOE



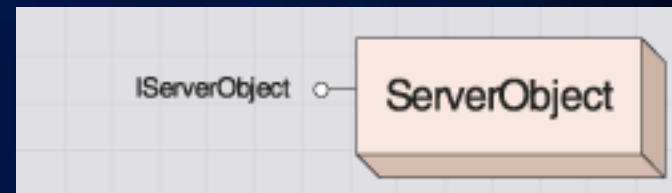
什么是SOE

SOE=

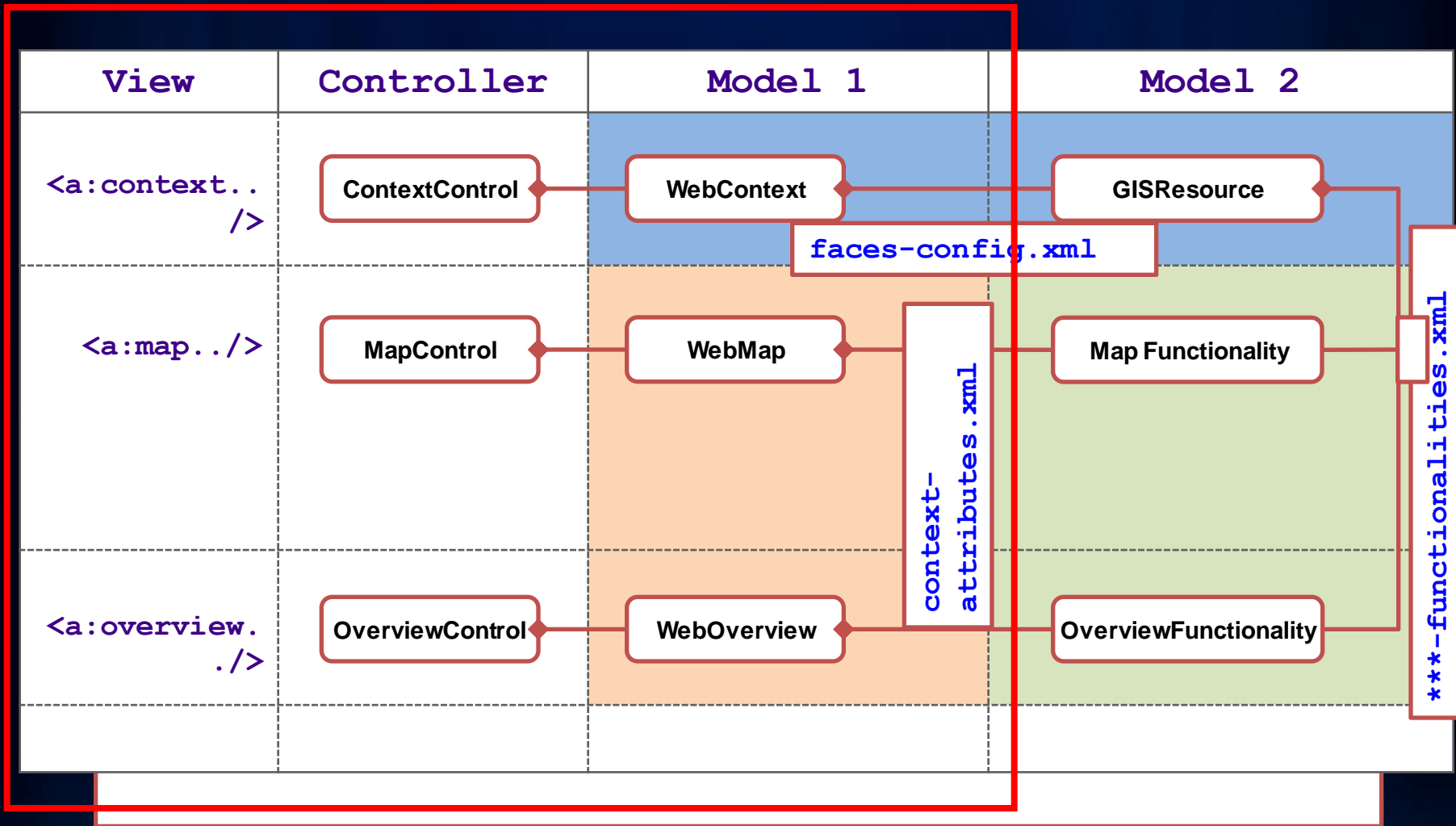
SO + E

SO—Server Object

- Server Objects是粗粒度的AO组件，是一个可以执行特定操作的高级对象，它隐藏了细粒度的ArcObject。Server Objects的粗粒度接口支持对于像绘制地图或地理编码等整体性任务。Server objects也有SOAP接口，可以把server objects暴露成Web Service，从而可以被客户端使用。



Web ADF MVC层次图



客户端Server Object生命周期

- `new ServerInitializer().initializeServer(domain, user, password);`
- `ServerConnection connection = new ServerConnection();`
- `connection.connect(host);`
- `IServerObjectManager som = connection.getServerObjectManager();`
- `IServerContext scGC = som.createServerContext("portland", "MapServer");`
- `GeocodeServer geocodeServer = (GeocodeServer) scGC.getServerObject();`
- `//操作GeocodeServer对象。`
- `scGC.releaseContext();`

Server
初始化

创建连
接

获取
SOM
对象

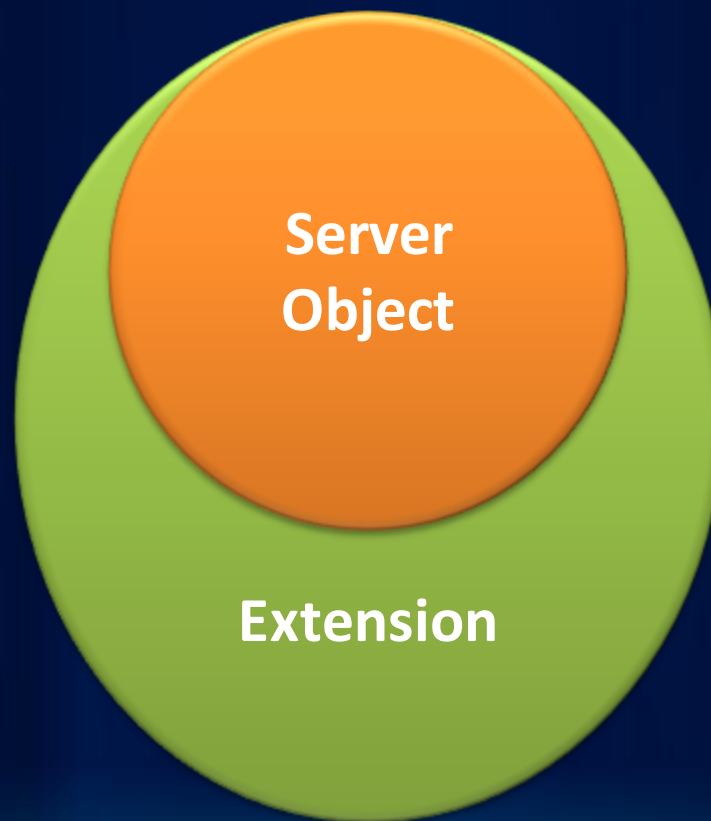
获取上
下文

获取
Server
Object

操作
SO

释放资
源

SOE



SOE溯源

- .Net

- 9.2中首次推出，在随后的9.3.1中引入了SOAP接口

- Java

- 9.3.1中引入SOE，实现可实现DCOM架构SOAP接口
- 10中可实现REST架构下的SOE

SOE

- Server Object Extension

服务器对象扩展，一个SOE（Server Object Extension）就是一个包含一组方法的类，这个类可以在SOC中运行，并且可以被web应用程序调用。

SOE运行机制概述



ArcGIS Server架构



桌面应用程序

Java/.NET 控制台

Java/.NET 图形界面

网络应用程序

Java Web ADF

.NET Web ADF

DCOM

SOAP

SOAP

DCOM

网络服务处理程序 (SOAP)

ArcGIS Server

SOM

(管理服务, 用户和网络地图应用程序)

SO

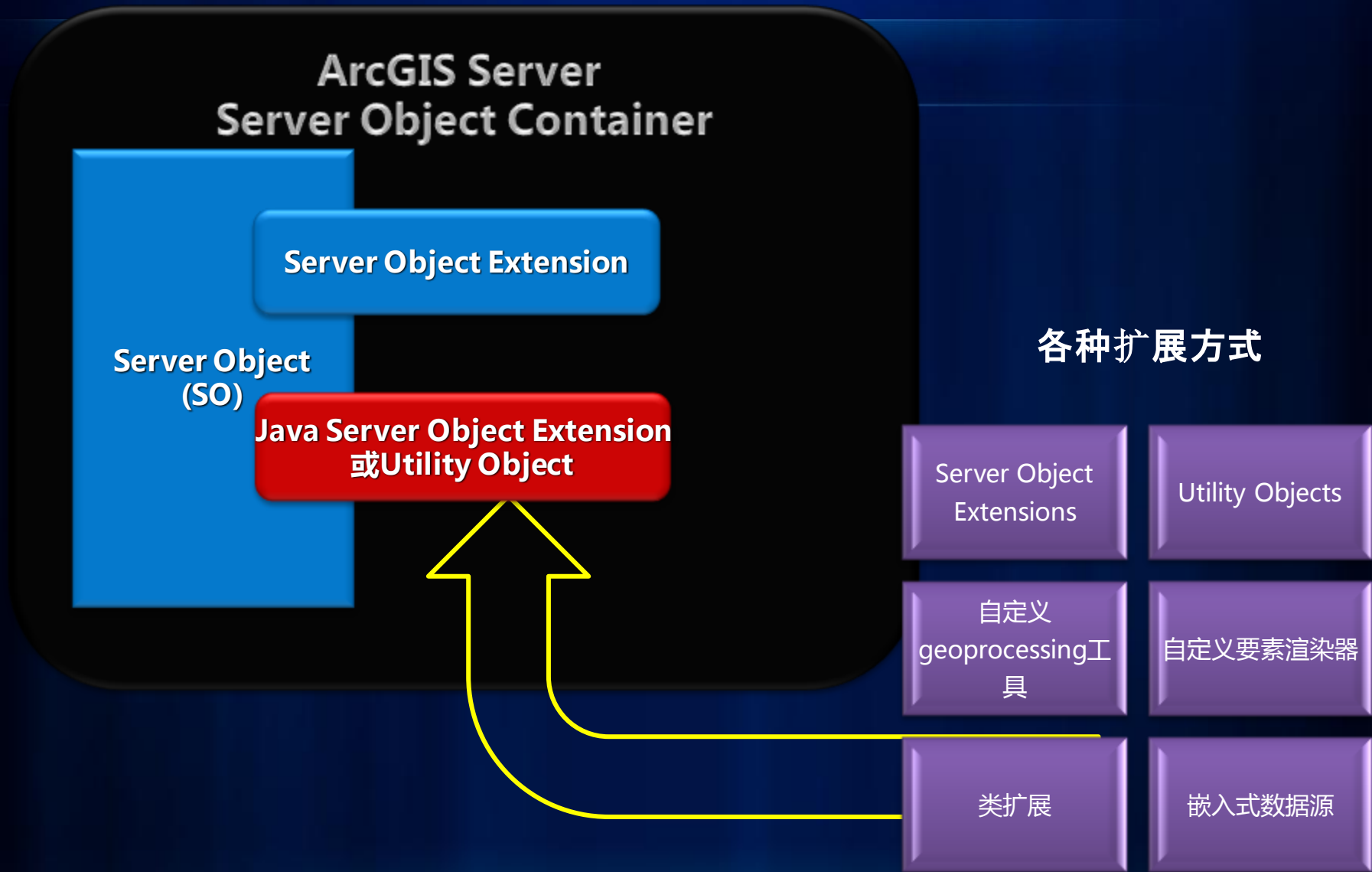
SOE

Server Object Container

SO

SOE

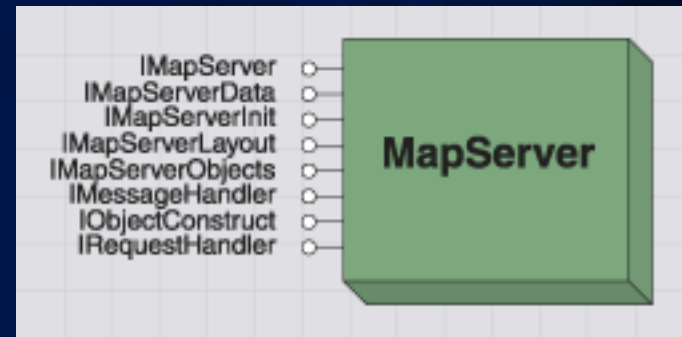
Server Object Container



Java扩展ArcGIS Server示意图

MapServer

- MapServer是一个粗粒度对象，
- 提供了对地图文档内容的访问，以及查询和绘制地图的方法。
- SOE到10.0版本为止只能应用到MapServer上



扩展ArcGIS工作流程



开发GIS Server扩展

Server Object Extension (SOE)

- 是能够封装业务逻辑（自定义渲染器，类扩展等）的特殊“Wrapper”
- 客户端利用和配置扩展的统一方式
 - 优点
 - » 不需被显式创建
 - » 可以缓存信息
 - » 本地/远程两种方式管理
 - » 可以利用SOC进程 (生命周期通知，日志记录，访问和操作)
 - 缺点
 - » 只与一种Server Object类型或配置绑定

SOE详细概念

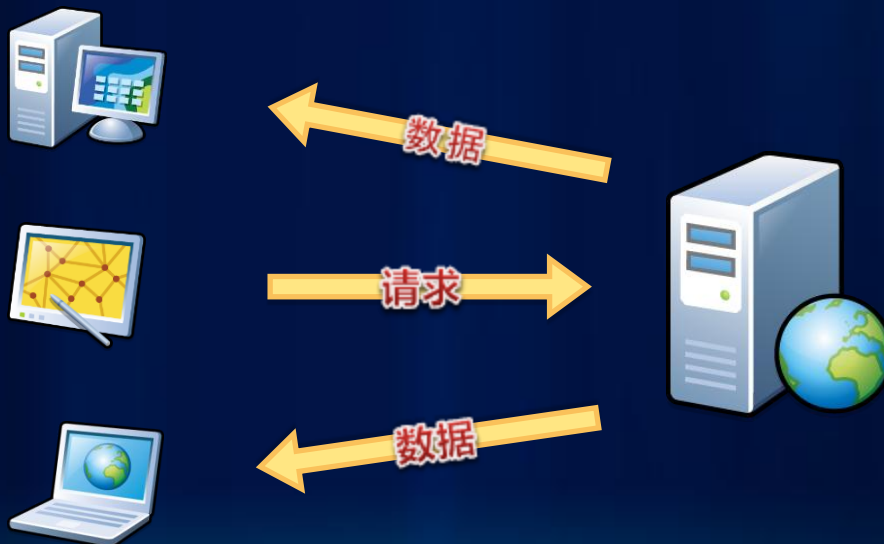
- 可以扩展MapServer Object，从而建立自定义的ArcGIS行为到Server Object中，而不必在Web层建立调用AO代理的Java业务逻辑。这样省去了跨进程调用的资源开销，提高了效率。
- SOE随着Server Object进程的创建而创建并初始化，对于每个请求，都被重用。
- 只要Server Object进程存在，SOEs进程也就存在。可以缓存被不同请求重用的信息。

SOE自定义行为

● SOE 自定义行为:

- 当`ServerContext` 被创建和释放时执行
- 可通过修改属性来修改SOE Runtime行为
- 在自定义接口中声明的函数可以创建被SOE实现的新AO类型
- 可以基于安全规则配置“操作”的访问权限
- 有SOE的操作日志

哪个是基于SOE下的服务模式？



SOE的优势

- ✓ 性能 ++! ——Up to 75%!
- ✓ 无状态，可重用

SOE的优势

- 1. 将代码模块化，降低耦合度和开发难度。SOE允许将底层ArcObjects业务逻辑代码和展现层代码分离。也就是将前后台分开，编写SOE的程序员不需要了解JSF，JSF开发者不需要知道AO就可以开发出很优秀的系统。

- 2. 性能提升。在某些场合下使用SOE性能会得到极大提升。

跨进程调用：很多项目需要往返调用，很多情况下，server处理请求不需要很长时间，但是网络开销极其影响性能。

细粒度ArcObjects：同理，细粒度ArcObjects通常需要很多往返调用，有必要使用SOE。

数据分析：如果要从服务器端传输数据为应用程序分析，确实有必要考虑用Geoprocessing工具或SOE。例如，想要获取一组属性值的中值，用JSF编写了查询和计算代码，将代码迁移到SOE之后，运行时间减少了75%。因为从SOC到客户端的唯一数据传输时间只是把数据结果呈现给用户，客户端上传了必要的的数据之后，无需再从客户端往返调用任何信息给服务器端。

- 3. SOE (REST&SOAP) 支持多种富客户端API。利用ESRI的Javascript, Flex和Silverlight等API都可以开发SOE的客户端。

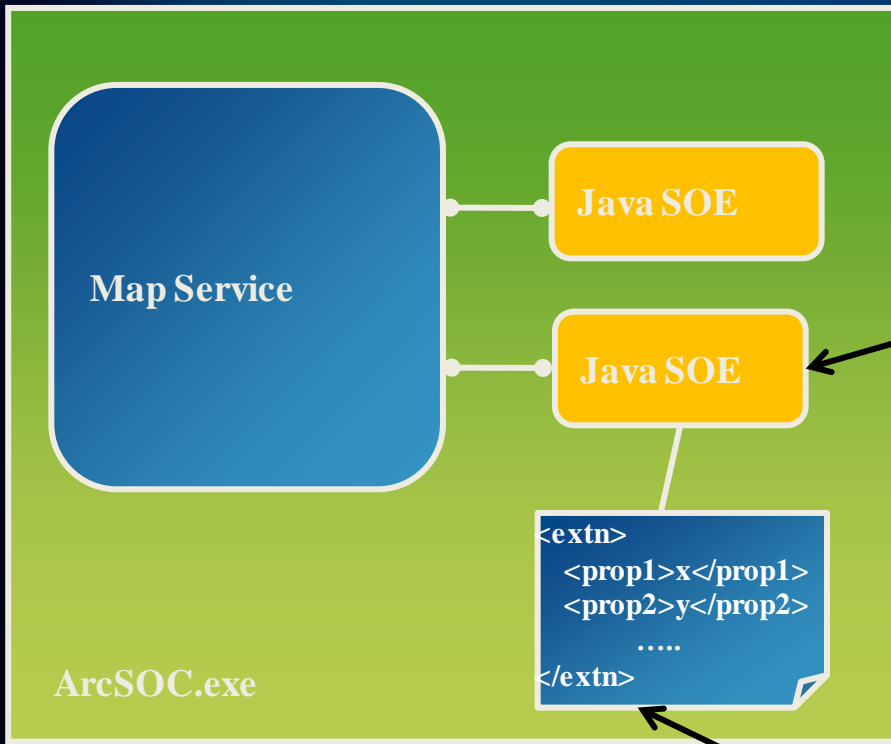
SOE workflows



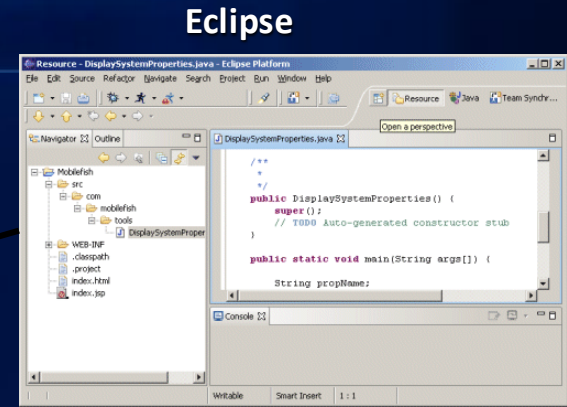
SOE workflow

- 开发
- 部署
- 管理
- 使用

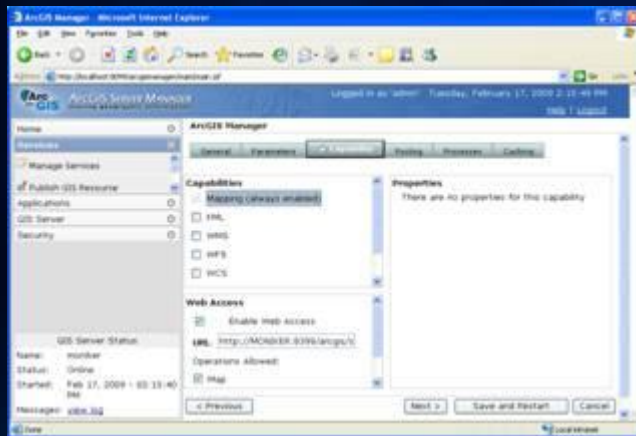




调试Java SOE



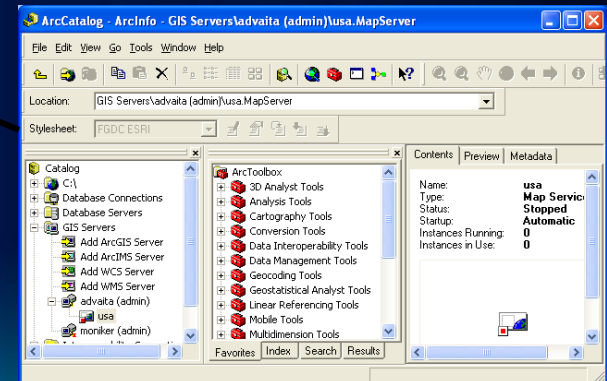
配置 Java SOE



Manager



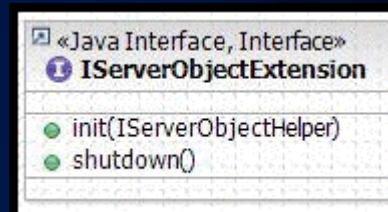
ArcCatalog



开发Server Object Extensions

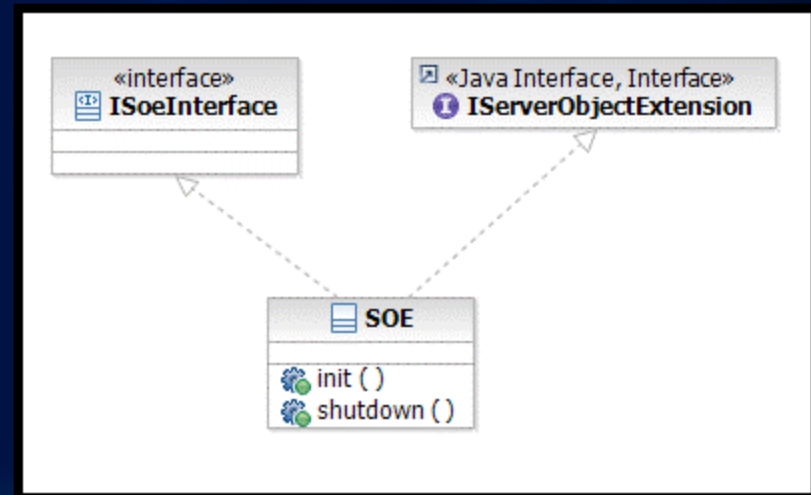
● 处理SOE对象

- 要自定义SOE，需要实现接口 *com.esri.arcgis.server.IServerObjectExtension*
- 这是所有SOE必须支持的强制接口，包含两个方法：*init()* 和 *shutdown()*.
 - *init()* 方法只在SOE被创建时被调用一次
 - *shutdown()* 方法把Server Object上下文的关闭消息通知SOE，只被调用一次
- 这个接口被Server Object用来管理SOE的生命周期



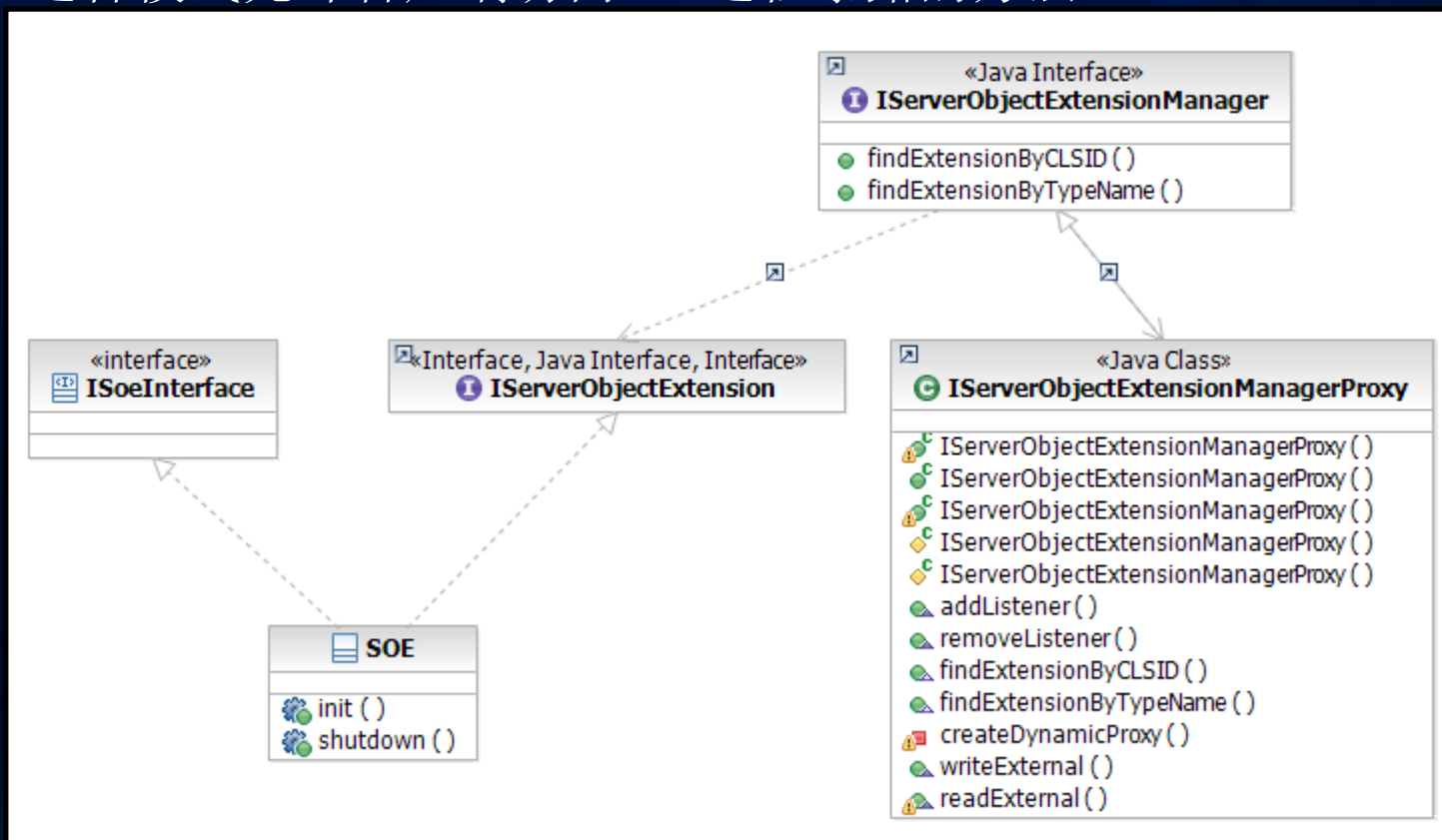
SOE 模式(Pattern)

- 开发SOE的通用模式是创建一个SOE类，这个类实现强制接口 *IServerObjectExtension* 和自定义功能接口 *ISoeInterface*



SOE 模式

- SOE实现指定的AO接口，以提供指定业务逻辑的自定义ArcGIS扩展接口
- 这种模式允许客户端访问SOE远程暴露的方法



@ArcGISExtension 笔记

- 自动暴露到ArcGIS平台

```
// Custom Interface
@ArcGISExtension public interface ISoeInterface{
    public void mySoeFoo();
}

// SOE class
@ArcGISExtension public class SOE implements IServerObjectExtension, IMySoeInterface{
    // IServerObjectExtension
    public void init(IServerObjectHelper arg0) throws IOException,
        AutomationException{
        // Called once when the instance of the SOE is created
    }
    public void shutdown() throws IOException, AutomationException{
        // Called once when the SOE's context is shut down
    }
    public void mySoeFoo(){
        // Instantiate custom interface
    }
}
```

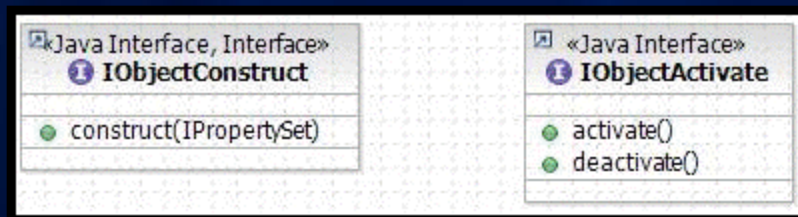
扩展方式 (*@ArcGISExtension*)

- 自定义要素渲染工具
- 类扩展
- 嵌入式数据源
- Utility Object
- Server object

可选接口

可实现的Java SOE可选接口包括

- » *com.esri.arcgis.system.IObjectConstruct*
- » *com.esri.arcgis.system.IObjectActivate*
- » *com.esri.arcgis.system.IRequestHandler*



IObjectConstruct

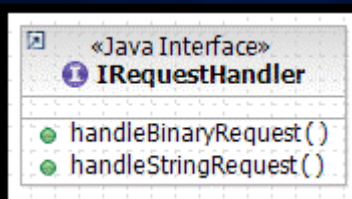
- 附加的初始化逻辑
- 配置属性
- 此接口包含了唯一的construct()方法
 - construct()方法在init()方法之后只被调用一次
 - 应该把任何高消耗的初始化逻辑包含进construct()方法中来

IObjectActivate

- 每一次Server Context被获取或释放时的附加逻辑
- *IObjectActivate* 是SOE的可选接口，包含两个方法*activate()* 和*deactivate()*.
 - 每次客户端调用Server Object上下文中*CreateServerContext()*时，都会调用*activate()*方法
 - 客户端释放上下文时会调用*deactivate()*方法

IRequestHandler

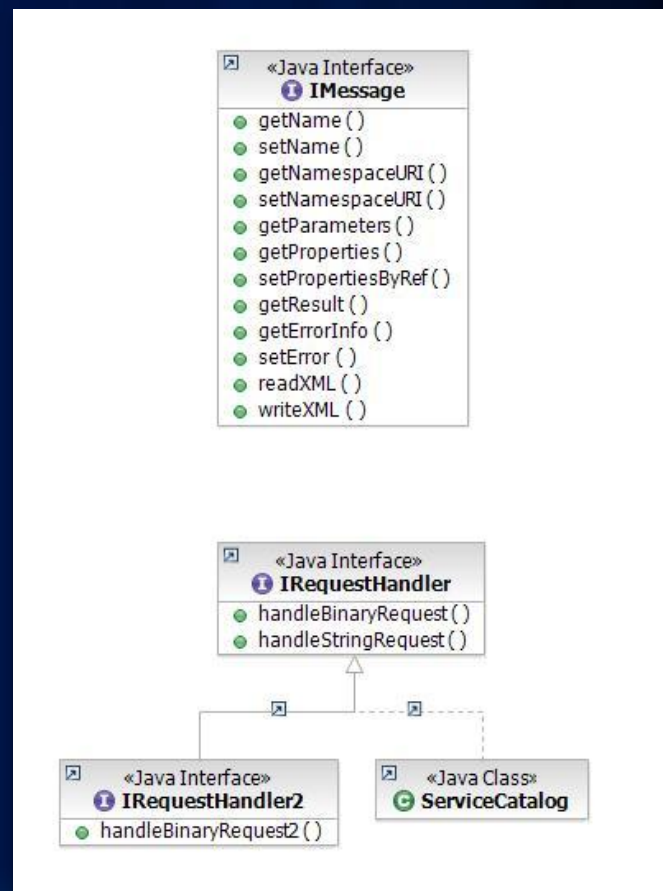
- 通过接受并响应SOAP请求，实现对SOE的web访问
- 本接口提供两个方法：



- *handleBinaryRequest()*: 处理二进制数据请求
- *handleStringRequest()*: 处理字符串请求

处理SOAP 请求

1. 解析SOAP请求
2. 确定客户端的具体请求
3. 创建包含请求信息的SOAP响应



处理REST请求： *IRESTRequestHandler*

- SOE要实现Rest Web Service，必须实现IRESTRequestHandler接口。

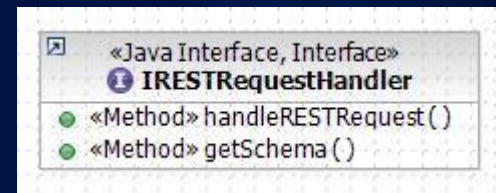
- 包含两个方法：

- handleRESTRequest()

- 处理对资源操作调用的请求
- 返回对资源的进程级别描述

- getSchema()

- 以JSON格式返回SOE的资源列表



ArcGIS Server 10新特性



SOE开发方式比较

	9.3.1		10
SOE开发方式	DCOM	SOAP	REST
开发难度	高	较高	较易
使用难度	高	较高	较易

handleRESTRequest()

- 参数：
 - capabilities
 - resourceName
 - **operationName**
 - operationInput
 - outputFormat
 - requestProperties
 - responseProperties

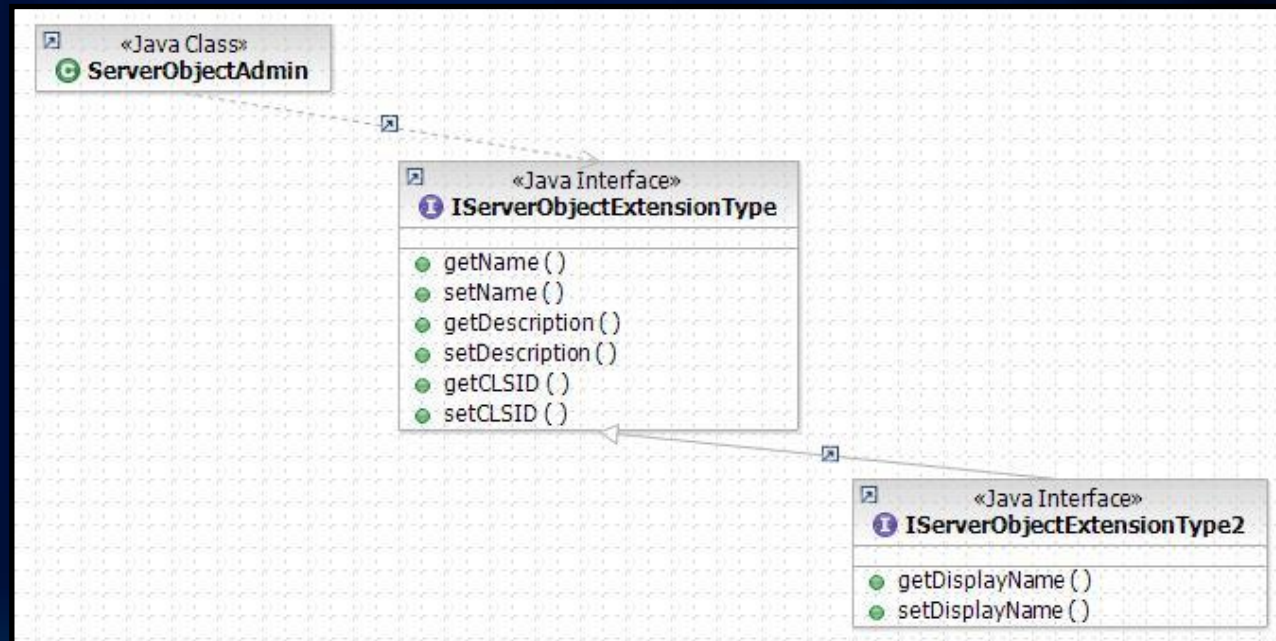
日志记录

- 所有SOE都可将消息记录到ArcGIS Server的日志中（可选）
- *ServerUtilities*是一个可选SOE类，唯一方法*getServerLogger()*通过*ILog2*接口返回给ArcGIS Server log文件
- 如果你想要SOE记录消息到Server日志中，必须通过*ILog2*接口



部署和管理Server Object Extensions

1. 部署SOE到ArcGIS Server上
2. 将Map Server Object附加到SOE中
3. 启用SOE



使用Server Object Extensions

● 通过多种方式使用SOE



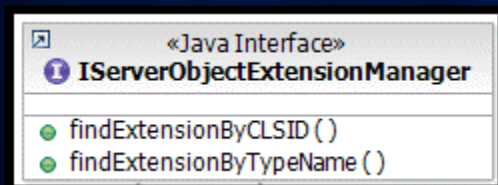
- 控制台应用程序



- 基于Web的应用程序

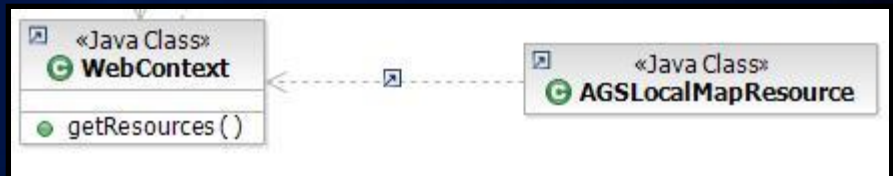
控制台应用程序中使用SOE

1. 获取Server Context
2. 通过服务上下文访问Server Object
3. 在Mapserver上调用
findExtensionByTypeName(soeName)



在Java Web ADF中调用SOE

- WebADF 通过 *WebContext* 引用 GISResources 和如 WebMap 等 GIS 业务对象的属性
- 如果你可以通过 *WebContext* 访问 Server Object，则获取 SOE 的形式与在控制台应用程序中的相同。



SOE 演示

基于Flex API调用SOE

- 界面友好，美观
- 性能高
- 与REST完美对接

SOE是万能的吗？

- SOE与Server Object共享线程资源，减小了跨线程调用的开销。
- 可以为地图服务量身定做，集中业务逻辑
- 同时增加了业务耦合度，生命周期与Server Object完全相同。一旦服务器进程崩溃，SOE的功能即失效，甚至影响其他业务。
- 在某些情况下，Utility Object性能更高

SOE的不适用场景

- 对于与Server Object生命周期无关的应用，尽量避免使用SOE。
- 在数据库操作等场合可以采用Utility Object等扩展方式。

SOE特性总结

- SOE的本质是一个特殊的寄生服务器对象（Server Object），与其宿主地图服务共享服务器资源与访问方式。
- 大幅减少客户端与服务端端的细粒度跨线程调用，将降低应用程序复杂度，明确区分了提交请求与处理请求的分工。



esri **China**
BEIJING

谢谢!

敬请关注：

李宗倍——《ArcGIS 10 中的OGC服务》

15:00-15:50