

通过Java ADF开发和部署 ArcGIS Server应用

林青 (Cherry Lin)

ESRI



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

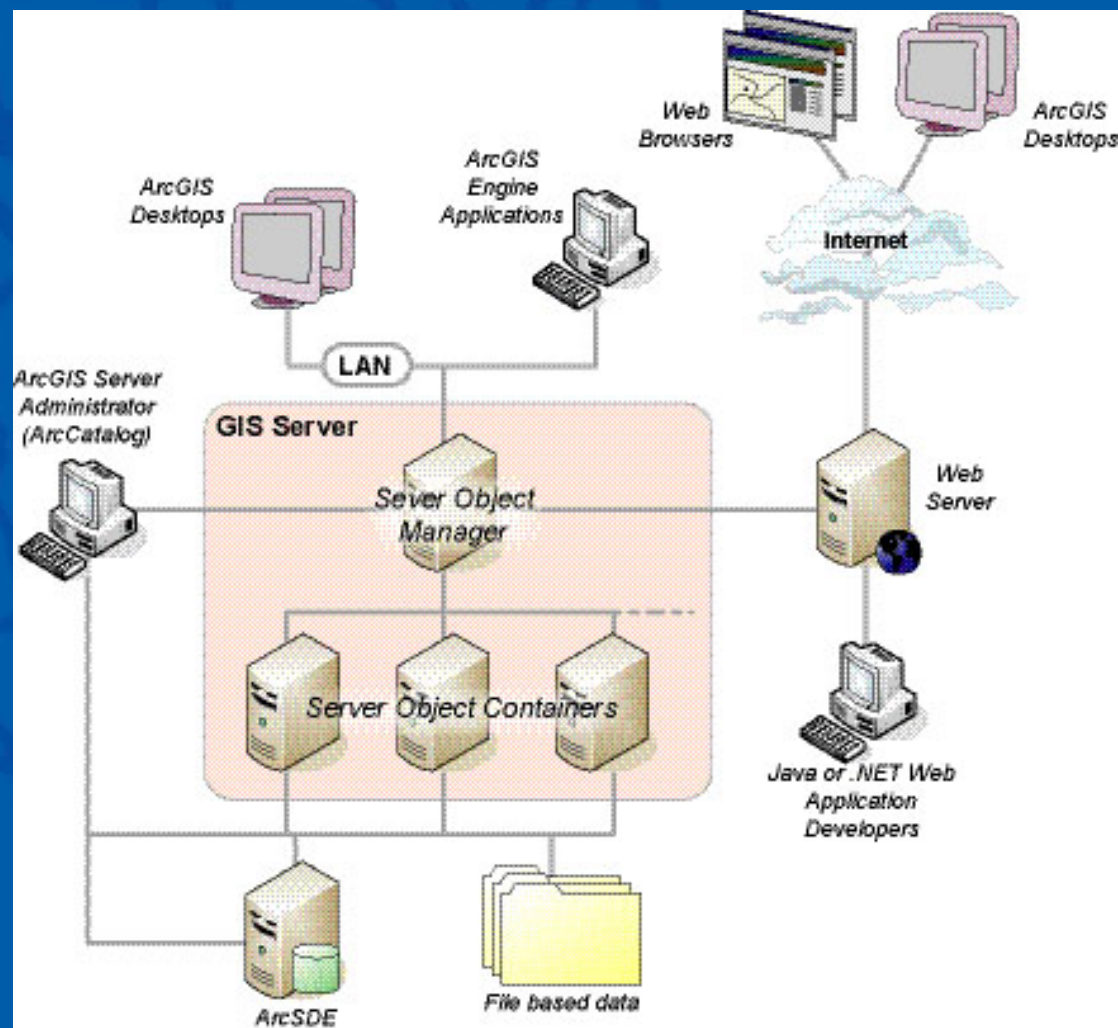
总结

演示

问题？



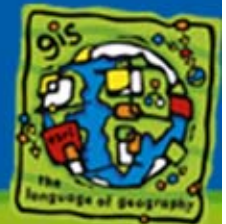
综述 – ArcGIS Server 系统



什么是 ADF (Application Development Framework)?

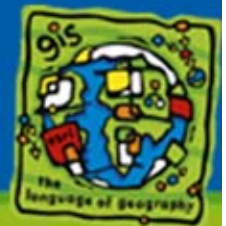
一个框架: 用于开发 ArcGIS Web 应用和 Web services

程序开发人员应用 .NET 和 Java 的接口



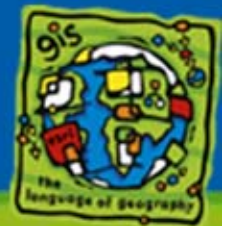
为什么要使用 ADF ?

简单便捷地开发GIS Web 应用
利用Web 模板快速地开发和部署
可连到 ArcObjects 的所有功能
可帮助管理连接 GIS server 和 Server objects
提供通向常见的绘图内容的用户界面



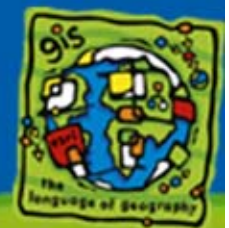
Java ADF 系统开发综述

- GIS Server 容纳提供GIS 功能和数据的 server objects
- J2EE 应用能够通过 Java Application Development Framework (ADF) 来使用这些 server objects
- Java ADF 适用与多种级别的程序开发人员
- Java ADF 支持不同J2EE级别中的使用
 - Client – Server usage
 - Web Container
 - EJB Container

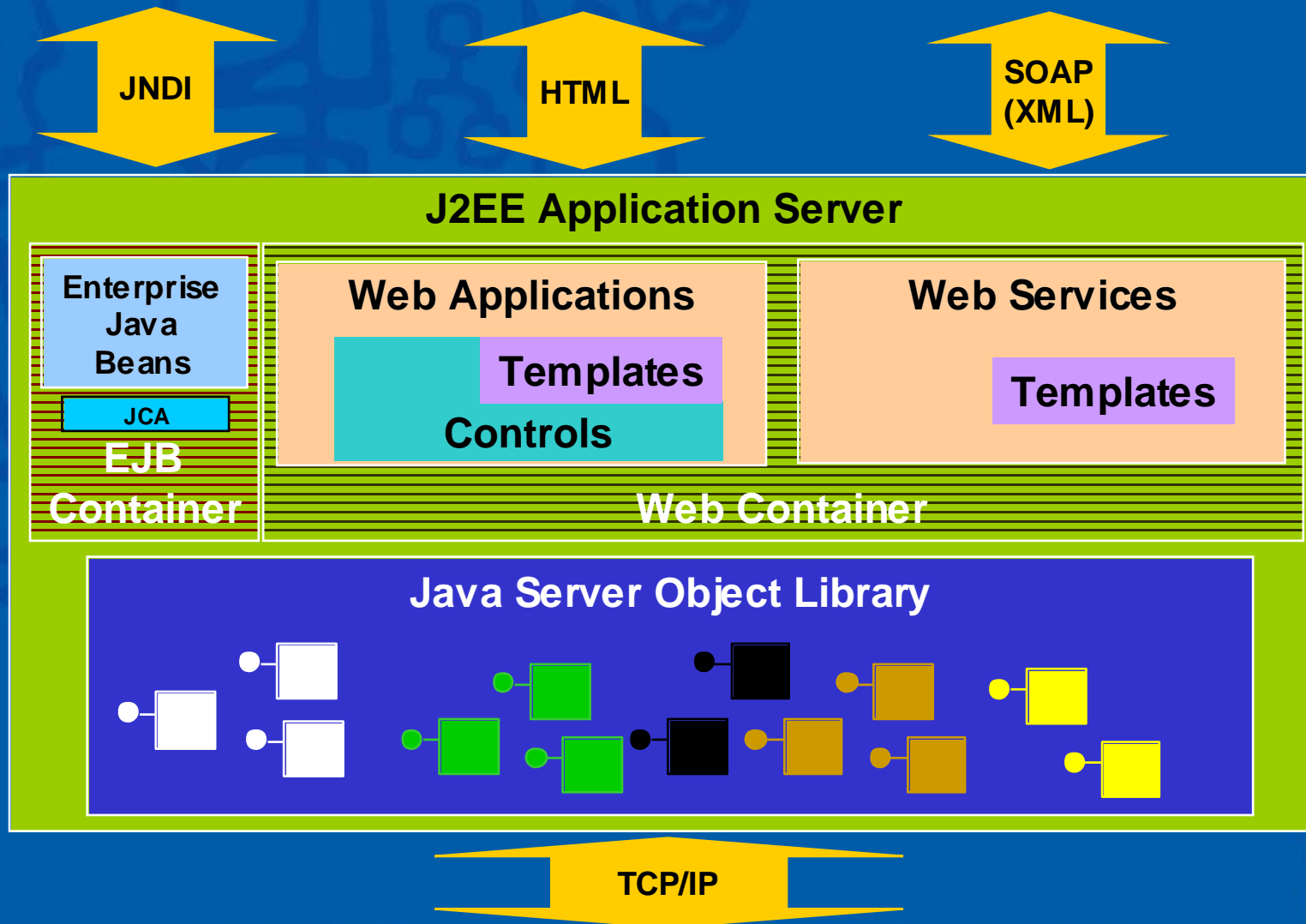


Java ADF 系统开发综述

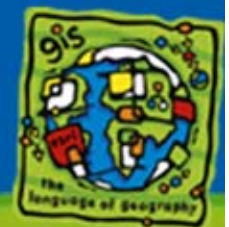
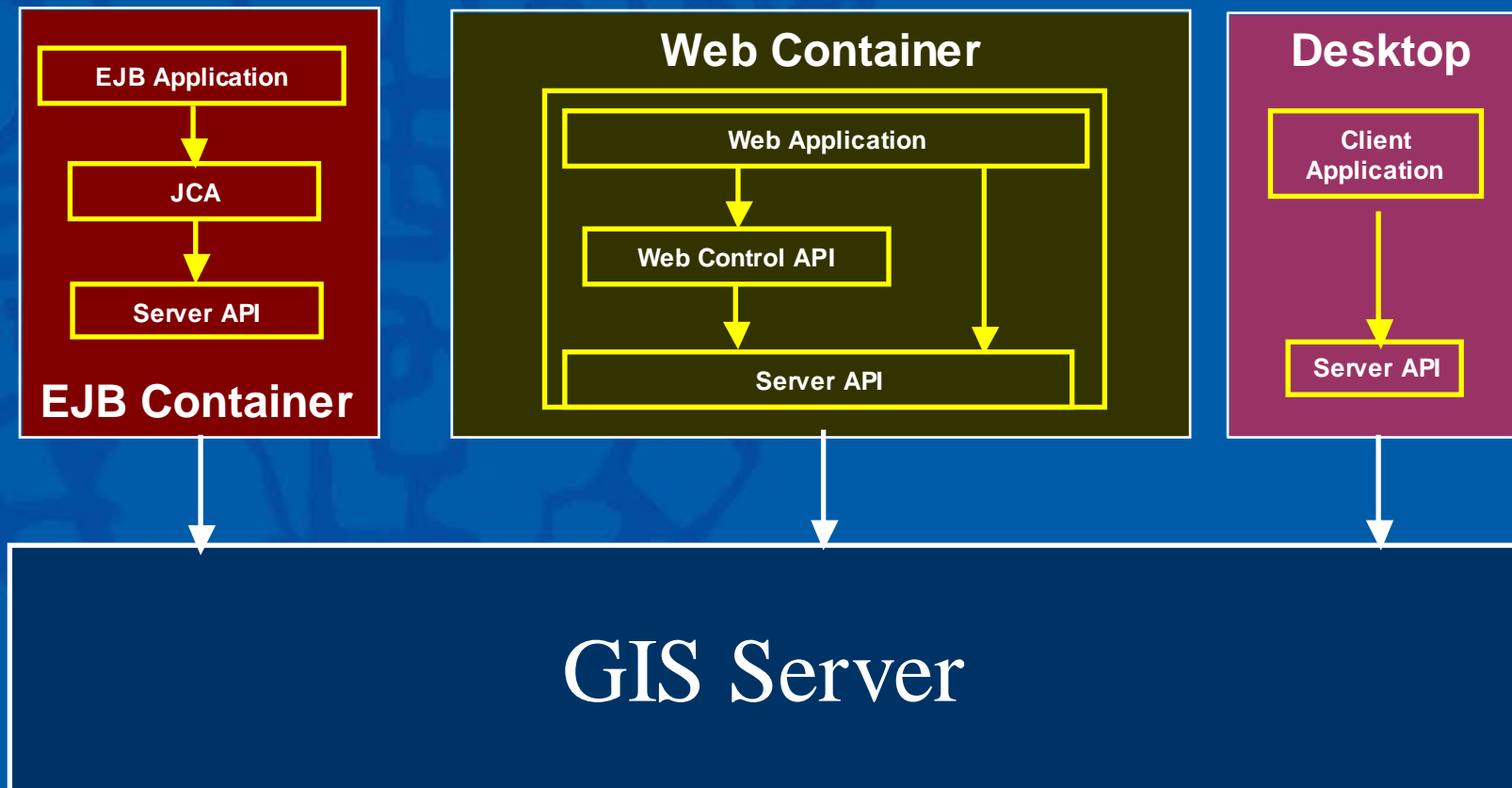
- JAVA ADF的产品构成
 - Class library
 - Web Services
 - Web 控件 (Controls)
 - Web 模板 (Templates)
 - J2EE Connector Architecture (JCA) Resource Adapter (RA) for EJBs
- 支持多种操作平台 (用于开发和部署)
- 支持多种 IDE
- 支持多种J2EE 开发服务器 (Application Server)
- 协调多种工业标准



Java ADF 构成



编写程序的不同级别



内容

综述

ArcGIS Server Java API

Web 控件和模板

Web Services

开发中的重要概念

进一步开发

总结

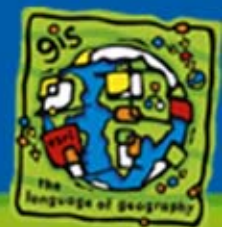
演示

问题？



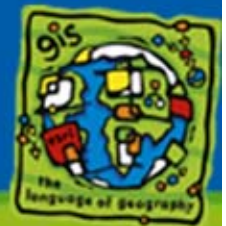
起步

- 怎样远程编辑ArcObjects
- 理解 Server 构造
- Java编程
 - JSP
 - Servlets
 - 其它内容
- 开发与管理的网络应用服务器 (Web Application Server) 中的应用开发



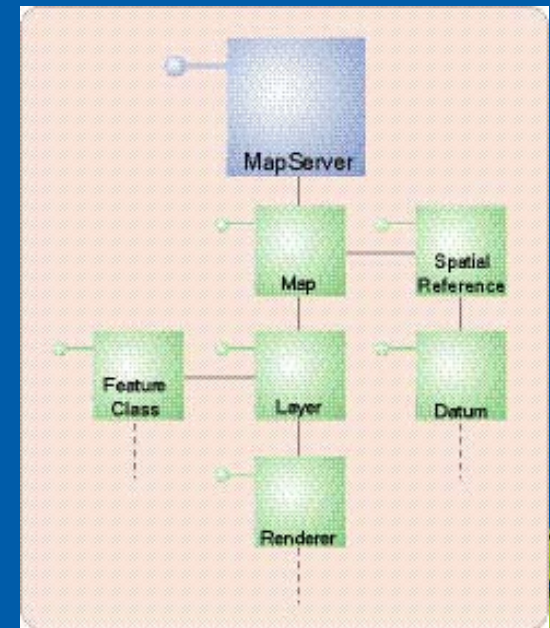
开发人员参考资料

- 安装 Java ADF
- 资料
 - Server_Developers_Guide.pdf
 - Engine_Developers_Guide.pdf
- Javadoc
- 示例
- Internet
 - <http://arcgisdeveloperonline.esri.com>



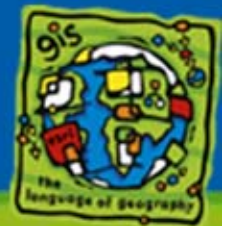
Server Objects

- 粗粒的 ArcObject
 - `com.esri.arcgis.carto.MapServer` – 展示一个 mxd 文档
 - `com.esri.arcgis.location.GeocodeServer` – 展示一个定位器
- 展示一套无状态方法
 - `MapServer.exportMapImage`
 - `GeocodeServer.geocodeAddresses`
- 提供到其它相关 objects 的连接
 - `MapServer -> Map -> Layer`



MapServer Object

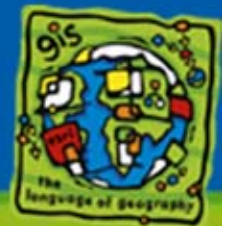
- 描述和展示图及图层
- 支持查询, 点击查看和特性分析
- 提供通向数据库的入口
- 描述和展示地图的布局
- 支持向多种栅格和矢量格式的高分辨率的图形转化



Geocode Server Object

支持：

- 地址标准化
- 地址确认
- 单一地址的定位
- 批量的地址定位



编辑 Server Objects

- Impersonate

```
new ServerInitializer().initializeServer(domain, user, password);
```

- 连接到一个 Server

```
IServerConnection connection = new ServerConnection();  
connection.connect(host);
```

- 得到一个 Server object

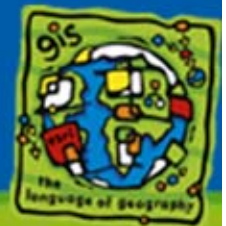
```
IServerObjectManager som = connection.getServerObjectManager();  
IServerContext sc = som.createServerContext("world", "MapServer");  
IMapServer ms = new IMapServerProxy(sc.getServerObject());
```

- 新建一个 object

```
IPoint pt = new IPointProxy(sc.createObject(Point.getClsid()));  
pt.putCoords(0.0, 0.0);
```

- 释放 Server context

```
sc.releaseContext();
```



Simple Example – MapServer

```
//Impersonate
new ServerInitializer().initializeServer(domain, user, password);

//连接
IServerConnection connection = new ServerConnection();
connection.connect(host);

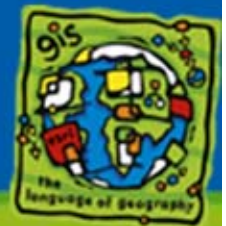
//得到 Server Object
IServerObjectManager som = connection.getServerObjectManager();
IServerContext sc = som.createServerContext("RedlandsMap",
    "MapServer");
IMapServer mapSvr = new IMapServerProxy(sc.getServerObject());

//用 Server Object
IMapServerObjects mapSvrObjects = new IMapServerObjectsProxy mapSvr);
IMap map = new
    IMapProxy(mapSvrObjects.getMap(mapSvr.getDefaultMapName()));

IFeatureLayer fLayer = new IFeatureLayerProxy(map.getLayer(0));
IFeatureClass fClass = new
    IFeatureClassProxy(fLayer.getFeatureClass());

//展示结果
System.out.println(fClass.featureCount(null));

//释放 context
sc.releaseContext();
```



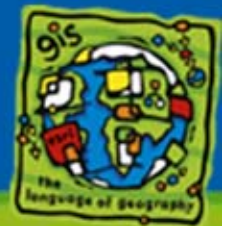
使用“空”的 Context

- 不需要现有的 Server Object
- 运行过程中创造 ArcObjects

//产生到一个 workspace 的连接

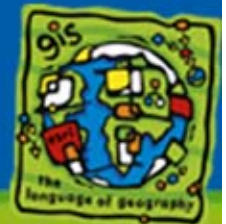
```
IServerContext sc = som.createServerContext("", "");  
IWorkspaceFactory factory = new  
    IWorkspaceFactoryProxy(ctx.createObject(  
        SdeWorkspaceFactory.getClsid()));
```

- 利用 GIS 功能



应用状态(State)

- 你可以 stateful 或是 stateless 地使用 server 中运行的 Server Object
- 如果你 stateless (i.e 只读) 使用 Server Object, 你可以和其它用户(或是会话期)共用这个 Server Object.
- 如果你 stateful (i.e 读写) 使用 Server Object 你不能和其它用户(或是会话期)共用.



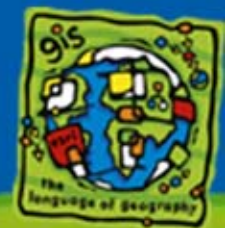
Shallowly Stateful 应用

- 需要在不同步骤中间保持 state, 进行展示 查询和分析
 - 每个用户/会话期可以有一个当前范围
 - 每个用户/会话期可以有一套可见层
- 使用 pooled Server Objects
- State 在ADF这边维护
- 使用Web服务器的“会话期管理”功能
 - 把GIS objects 系列化后输入 strings

```
mapDesc = mapServerInfo.getDefaultMapDescription();  
strMapDesc = context.saveObject(new IXMLSerializeProxy(mapDesc));
```

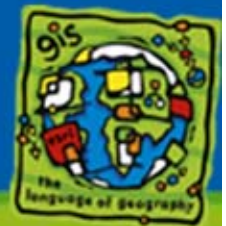
- 把 strings 存储成 session 变量
- 根据需要再获得 objects

```
mapDesc = new  
    IMapDescriptionProxy(context.loadObject(strMapDesc));
```



Deeply Stateful 应用

- 需要维护 state 来修改, 制图等等
 - 数据库进行多需求修改
 - 可多需求同时绘图
- 使用 non-pooled Server Object
- State 在 Server 这边维护



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

总结

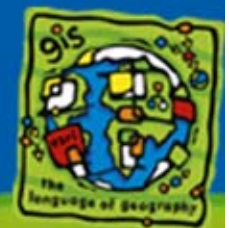
演示

问题？



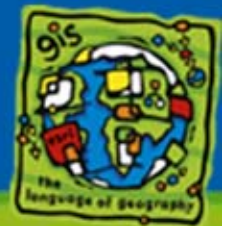
Web 控件 (Controls)

- 什么是 Web Control?
 - JavaServer Faces (JSF) 的 custom component
 - JSF 的信息:
<http://java.sun.com/j2ee/javaxserverfaces/>
- 他们提供什么?
 - 可再利用 UI 组件, 表现成 JSP tags
 - 事件模型和动作监听器
 - 能与多种 renderers 接口 (i.e. HTML, WML, DHTML, ...)
 - 在 HTTP requests 中间保持State
 - 与其它 JSF components 的接口

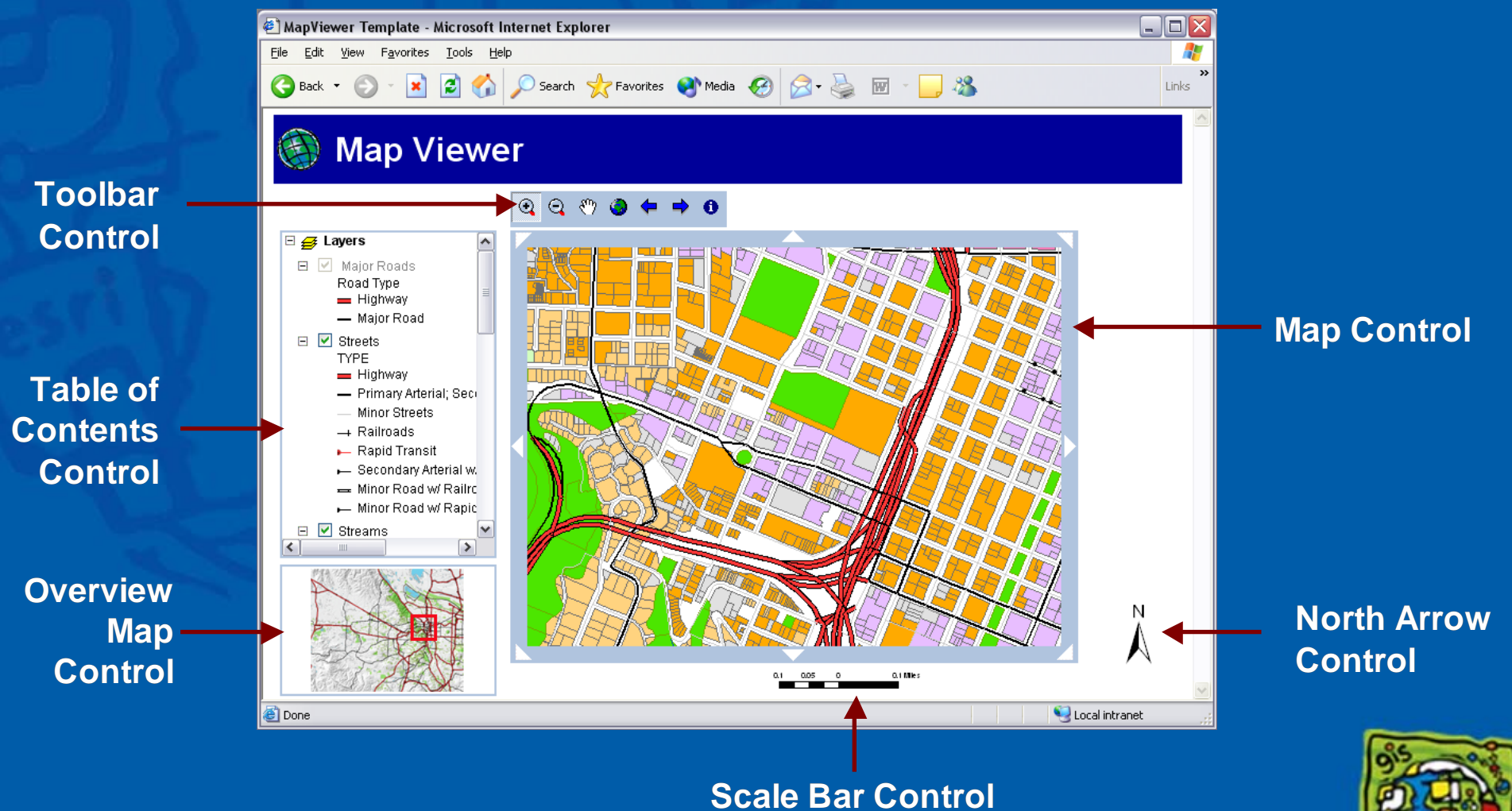


ArcGIS Web Controls

- 提供向 MapServer 和 GeocodeServer API 的入口
- 负责处理与 GIS server 工作的复杂性
- 提供额外所需的数据 objects
- 提供大量的开发框架



Web Controls



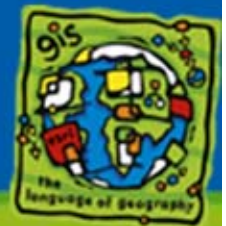
Map Control

- 展示 Map Server Object 的数据结构 (data frame)
- 嵌入式地图浏览 (map navigation) 工具 (例如, 视窗移动, 放缩)
- 已写好的用于用户方(client-side) 行为的 JavaScript (例如, 拖一个矩型窗口)
- 使用嵌入式服务器 (server-side) 方的行为或是编写你自己的程序



Page Layout Control

- 展示 Map Server Object 的布局
- 用于浏览布局和布局内各自的数据结构的嵌入式工具
- 已写好的用于用户方(client-side) 行为的 JavaScript (例如, 拖一个矩型窗口)
- 使用嵌入式服务器 (server-side) 方的行为或是编写你自己的程序



Overview Map Control

- 展示 Map Server Object 的数据结构 (data frame)
- 提供相应的 map control 的总介点
- 区域图框表现出目前的图区域
- 移动区域图框来移动相应的 map control 确区域

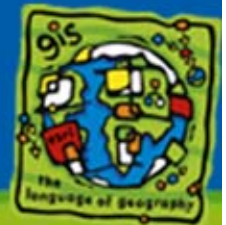
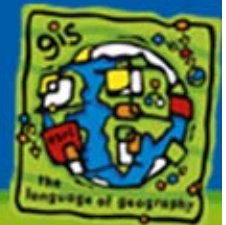


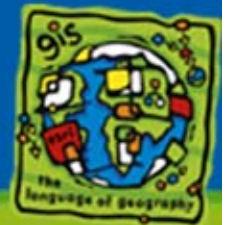
Table of Contents Control

- 列出数据结构和图层和它们的图标
- 标示出所有的数据结构或是单一数据结构
- 转换图层的可视性
- 可以将行为和外观用户化 (例如, 去掉图层框)



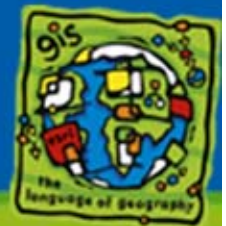
Geocode Control

- 在 Web 应用中启动地址定位
- 控制 向Geocode Server Object 的连接
- 建立连接, 渲染地址输入列, 并返回定位结果



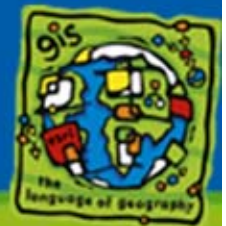
North Arrow Control

- 展示指北针
- 和 Map Control 动态连接
- 选择你想要的图标和大小



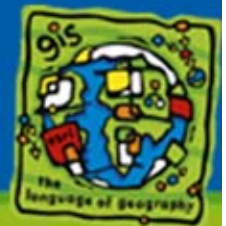
Scale Bar Control

- 展示比例尺
- 和 Map Control 动态连接
- 可定字体, 颜色, 大小, 类型, 划分数目



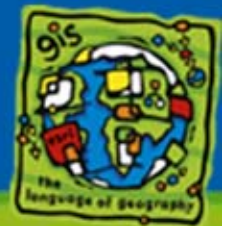
Context Control

- 建立和维护与 GIS server 和 server object 的连接
- 维护在它的范围内与所有的 controls 的关系
- 检测一个 server object 的 pooling 方式
- 释放 server context



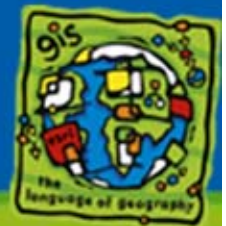
Identify Results Control

- 列出点击结果
- 提供树状结果



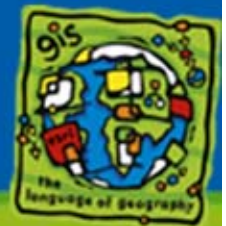
通过 JSP 使用 Web Controls

```
<% @ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<% @ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<% @ taglib uri="http://www.esri.com/arcgis/webcontrols" prefix="ags"
    %>
...
<f:view>
<h:form>
<ags:context id="mapContext" resource="Mexico@Host">
    <ags:map id="Map0" left="233" top="65" width="535"
        height="438" borderWidth="2" borderStyle="solid" />
    <ags:toc left="16" top="65" width="204" height="309"
        borderWidth="2" borderStyle="solid" />
    <ags:overview id="Overview0" resource="Mexico@Host" left="16"
        top="380" width="204" height="124"    borderWidth="2"
        borderStyle="solid"/>
</ags:context>
</h:form>
```



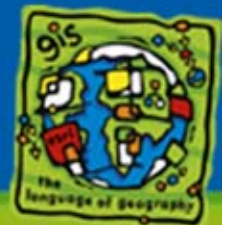
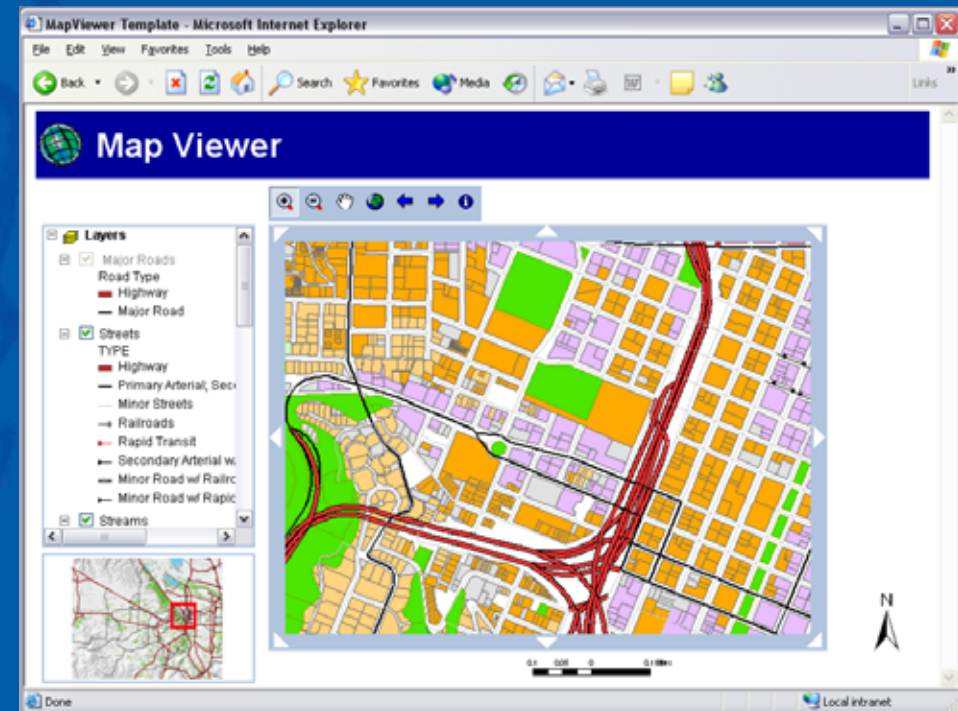
Web 模板 (Templates)

- Templates 是开发起点
 - 便于修改
 - 安装时是 Developer Kit 的一部分
- 用 ArcGIS Web Controls 来开发的



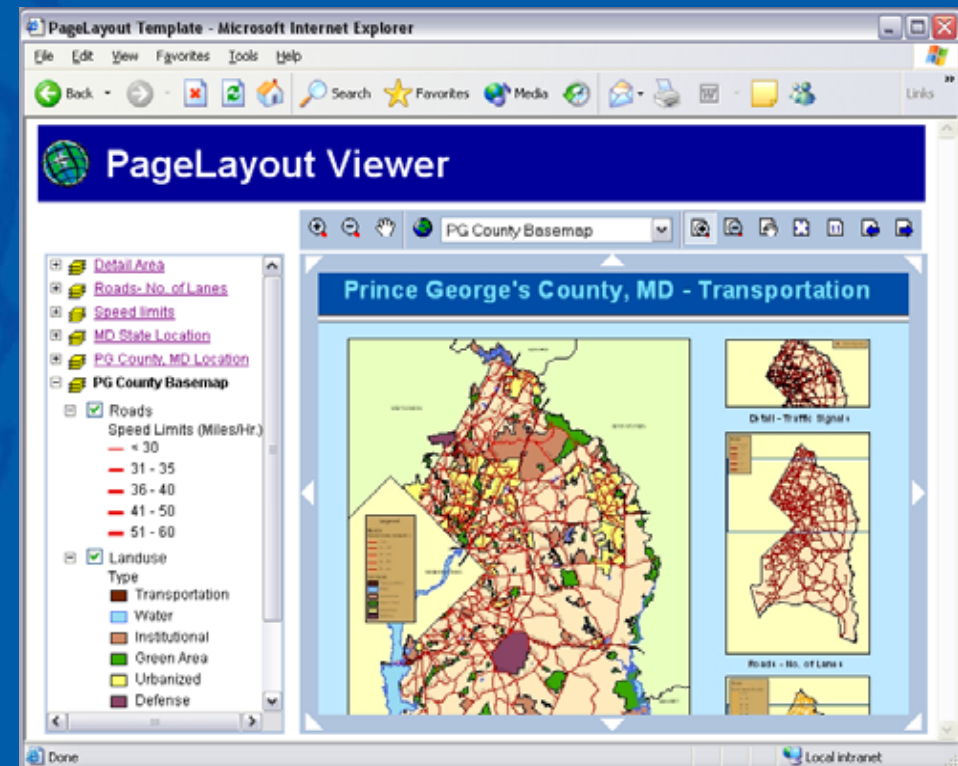
Map Viewer Template

- 提供基本的数据展示和查询
- 地图浏览
- 视图前进/后退范围
- 点击查看



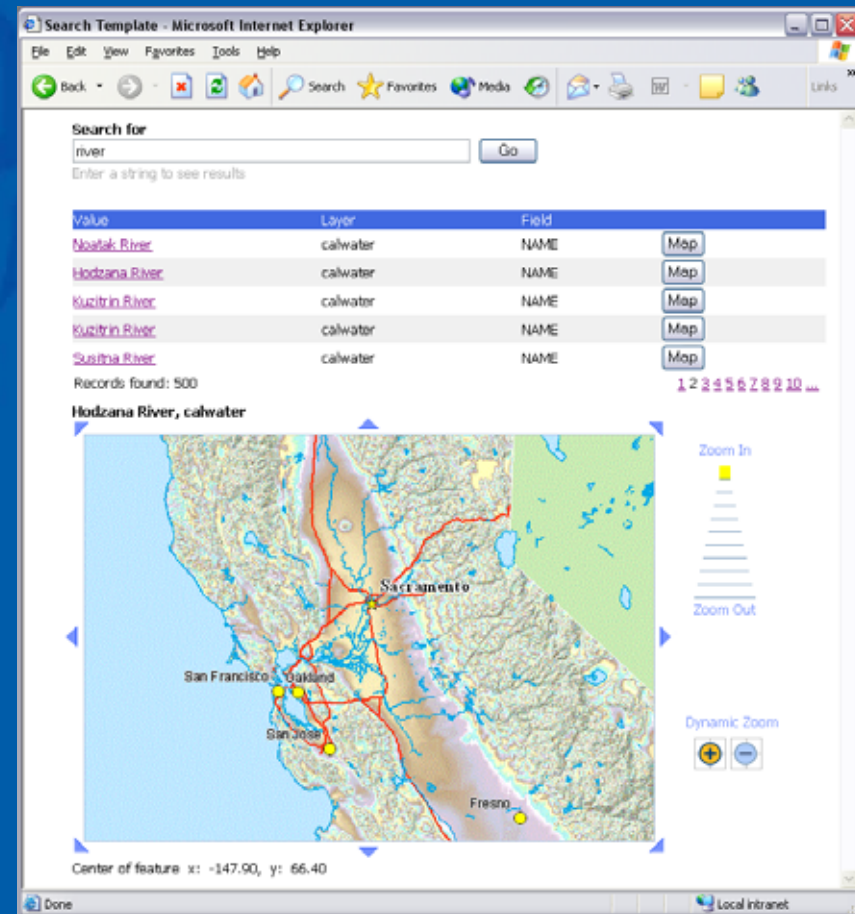
Page Layout Template

- 展示地图的布局, 提供所有的数据结构和周边地图
- 布局浏览
- 数据结构浏览



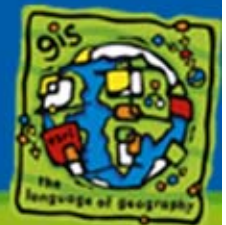
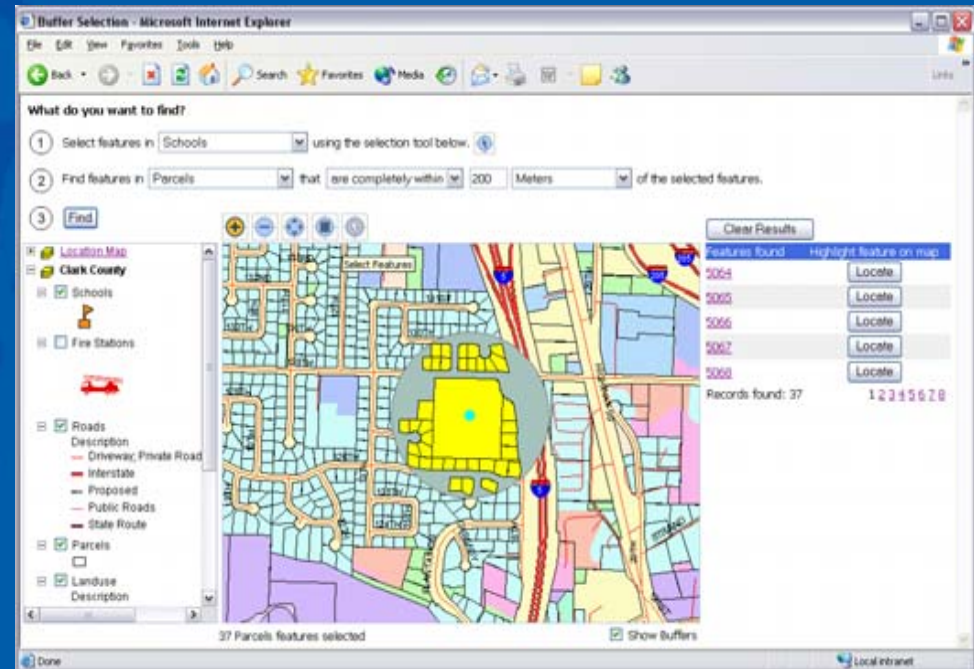
Search Template

- 根据特性查询
- Web 搜索 engine 类型
- 展示特性和搜索结果图



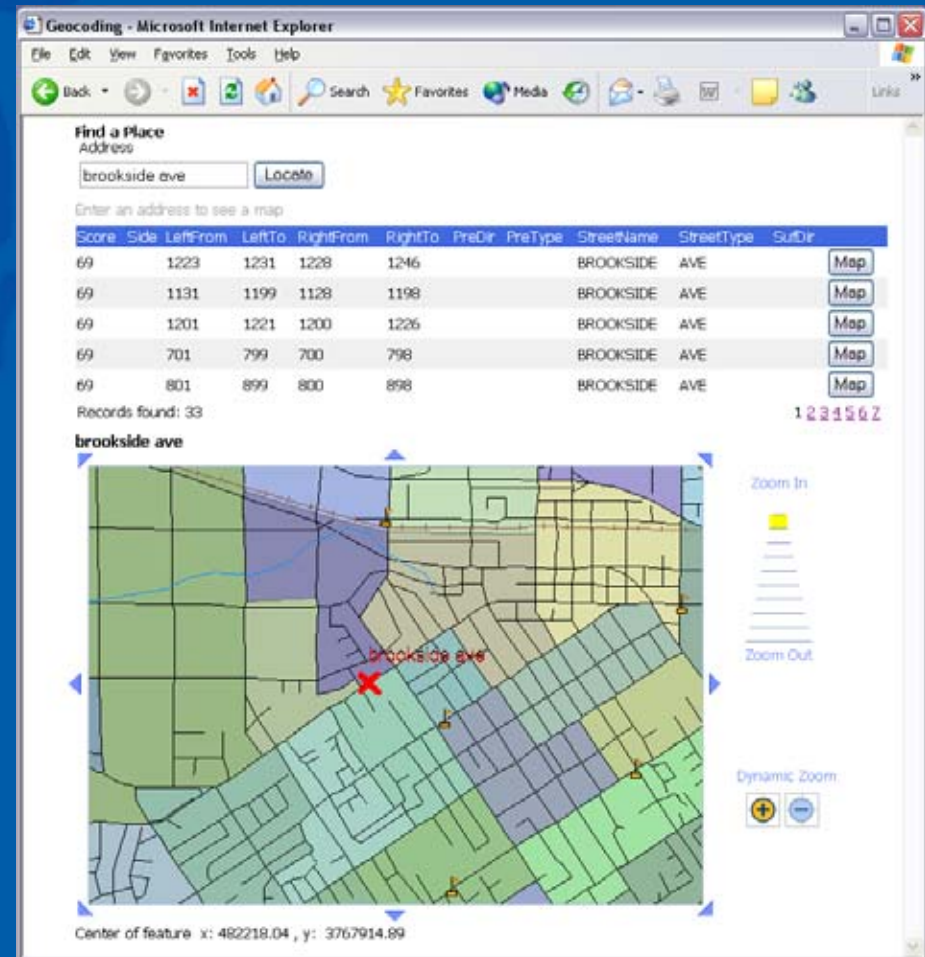
Buffer Selection Template

- 搜索靠近其它物件的物件
- 创建一个围绕所选物件的缓冲区并搜索完全包括其中或与之相交的物件



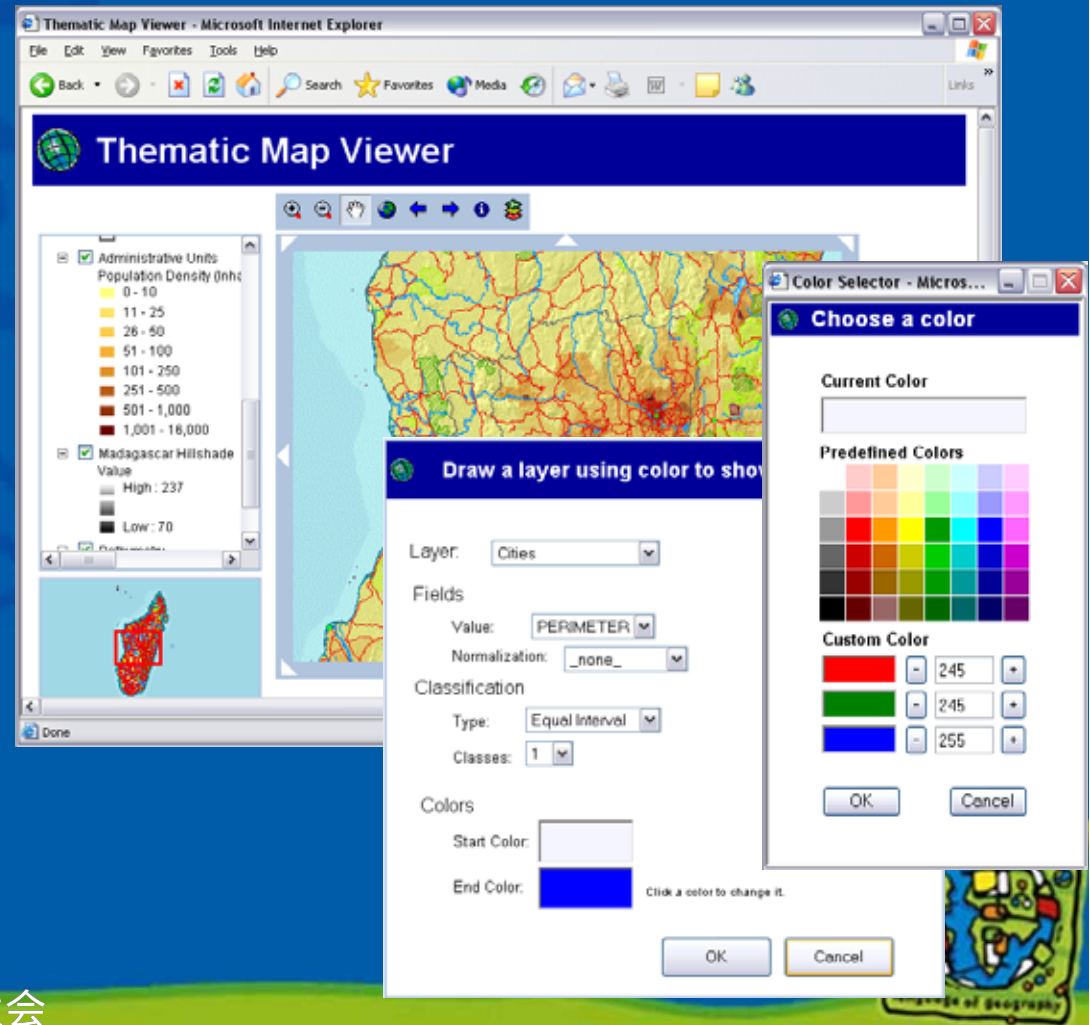
Geocode Template

- 地址定位
- 列出所定地址的地图
- 使用 Map Server 和 Geocode Server object



专题图 (Thematic) Map Template

- 动态变换图标
- 根据特性分级别
- 分级类型
 - Natural Breaks
 - Equal Interval
 - Quantile
- 要求 non-pooled server object



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

总结

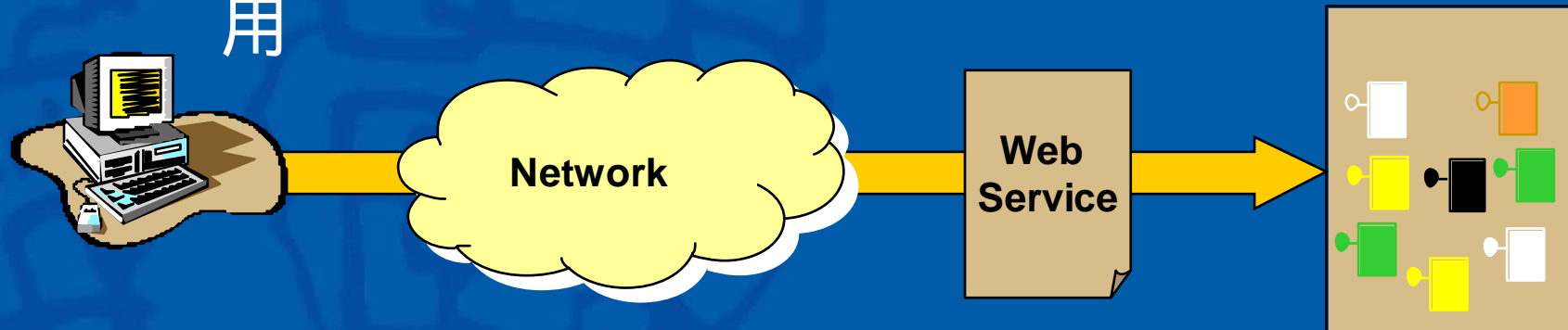
演示

问题？



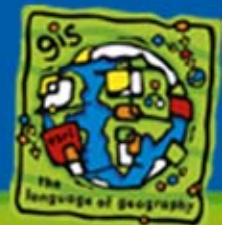
综述

- 什么是 Web service ?
 - 利用互联网络技术标准开发的网络界面或功用

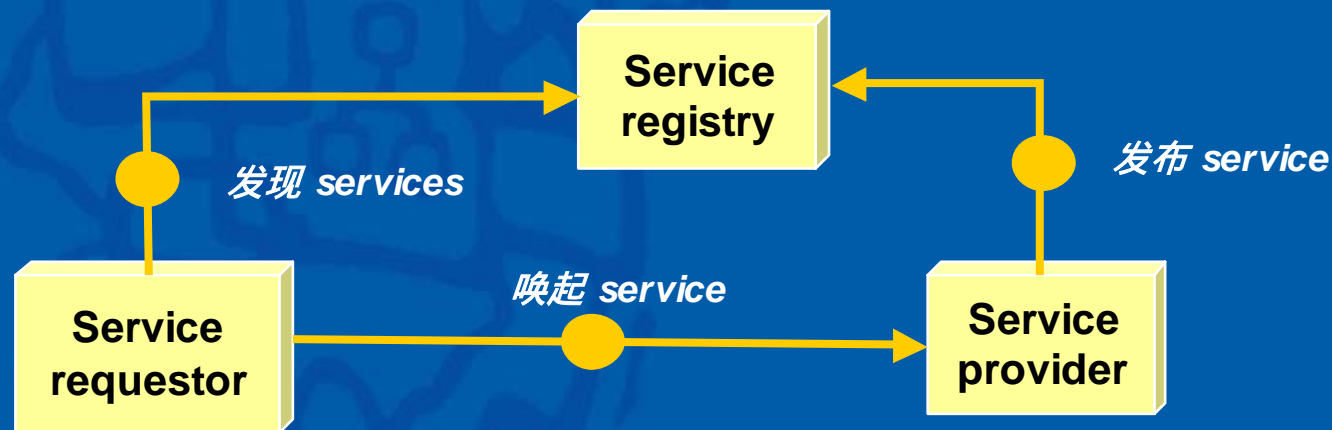


Web service 技术

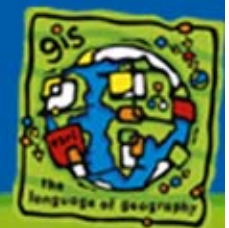
Discovery	UDDI
Description	WSDL
XML Messaging	XML-RPC, SOAP, XML
Transport	HTTP, SMTP, FTP



Web Service

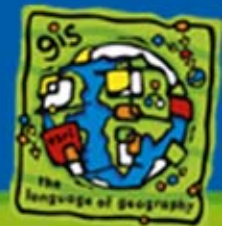


- Service provider 角度
 - ArcGIS Server Web service
 - 应用 Web service
- Service requestor 角度
 - ArcGIS Server 用户 (ArcMap, ArcCatalog)
 - 其它 clients (Applets, Web 应用, etc)



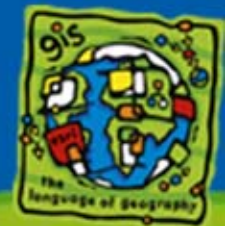
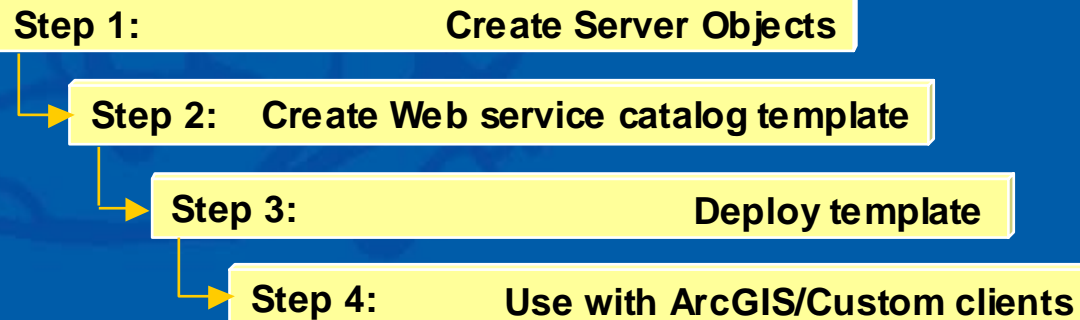
ArcGIS Server Web Services

- 提供可通过 Internet 获得的 Server Objects
 - 类型
 - Map service
 - Geocode service
 - 功能
 - Map
 - Data
 - Query
 - Geocode
 - Reverse Geocode
- 可在 Web service catalogs 里分组



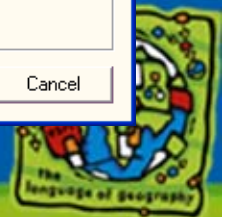
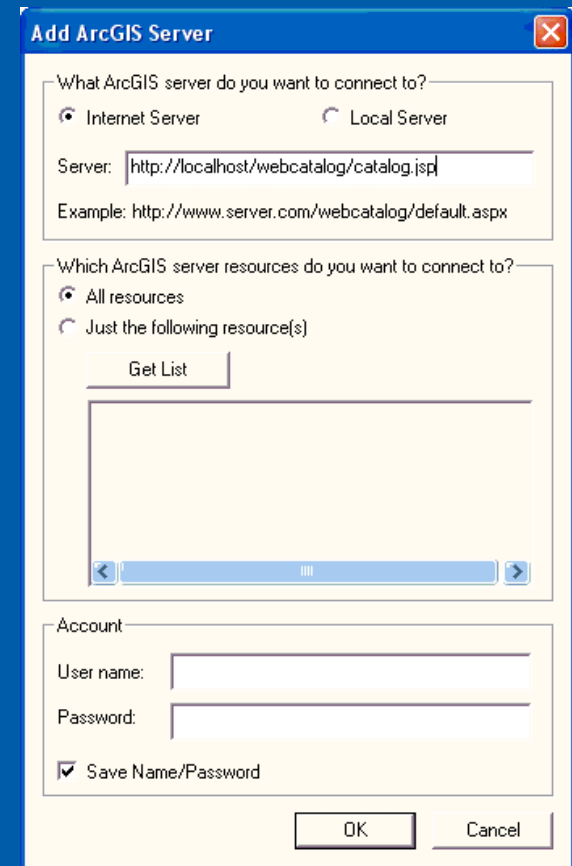
使用 ArcGIS Server Web Service

- Web service 和 Web service catalogs 的 Templates
- Development plan



Web Service Catalog Template

- 把相关的 server objects 分组
- 制造一个 web services 的浏览器
- 通过 HTTP 提供到 GIS Server的连接
- ArcGIS Desktop 用户可在 ArcCatalog 里连接



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

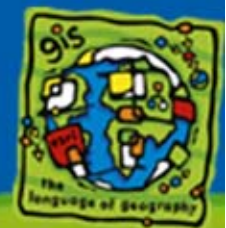
开发中的重要概念

进一步开发

总结

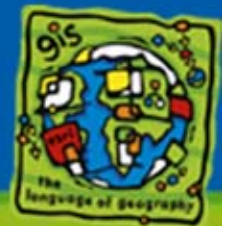
演示

问题？



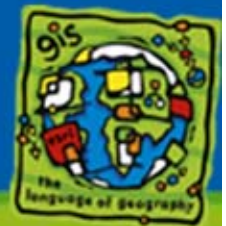
重要概念

- 使用 Web controls API
- 使用 server contexts
- 用户化控件 (controls) 的外观
- 访问 GIS server
- 部署一个应用



使用 Web Controls API

- 外观 – Web Controls 提供用户界面
- 技术逻辑 – 数据 Objects (Java) 提供常见的 GIS 功能和与 GIS Server 的连接



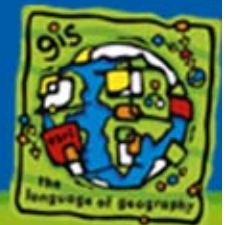
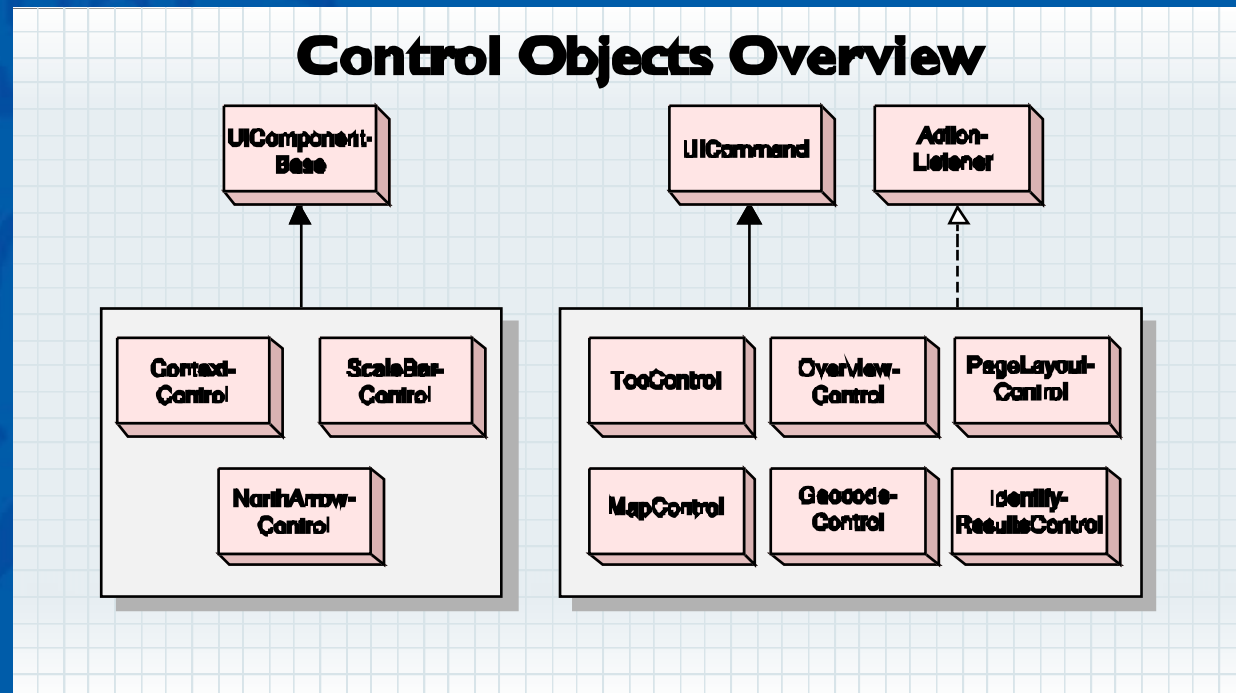
Java Web Controls 与 Data Objects

外观	商业逻辑
Context	WebContext
Map	WebMap
Overview Map	WebOverview
Page Layout	WebPageLayout
Table of Contents	WebToc
Geocode	WebGeocode
Scale Bar	WebScalebar
North Arrow	WebNorthArrow
Identify Results	WebIdentifyResults



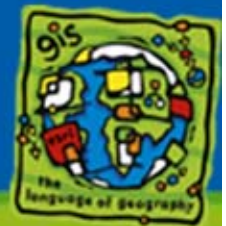
Java ADF Class Library

- 用 JavaServer Faces (JSF) 开发的
- 作为JARs 分发



Java ADF 包

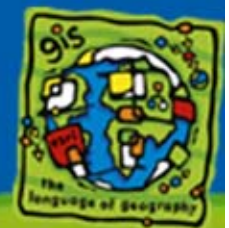
- com.esri.arcgis.webcontrols: base package
- faces.component: Web controls
- faces.event: event arguments
- faces.renderkit.xml: custom renderers
- faces.taglib: tag handlers
- data: data objects
- ags.data: ArcGIS Server data objects
- ags.faces.event: actions and listeners
- util: ArcObjects helper methods
- faces.validator: input validation



Java ADF Web Controls Tag Library

- 每一个控件都表现为一个 tag
- Tags 一个举例
 - context
 - map
 - pageLayout
- 连接到一个 GIS server object 并展示地图和内容列表

```
<h:view>  
  <h:form ... >  
    <ags:context id="context" resource="MyMap@Host" >  
      <ags:map ... />  
      <ags:toc ... />  
    </ags:context>  
  </h:form>  
</h:view>
```



在 Java 中将外观用户化

- JSP 特性

```
<ags:map id="Map0" borderColor="#B0C4D"
        borderStyle="solid" ... />
```

- XSL 变量

```
<xsl:variable
name="customBorderColor">#B0C4DE</xsl:variable>
<xsl:variable      name="customBorderStyle">solid</xsl:variable>
```

- CSS 类型

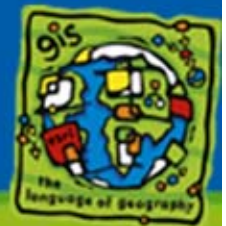
```
div.mapClass {
    border-color:#B0C4DE;
    border-style:solid;
```

- Web controls 在 HTML 渲染
 - 成为不同的 markup 语言



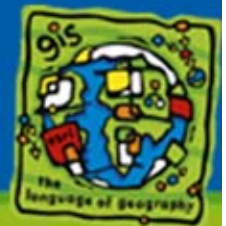
使用 Server Contexts

- Server contexts 提供与 server objects 的连接
- 用户方的应用必须适宜地使用 server objects
- 通过 Java Context Control 使用 Server Contexts
 - 检测 server object 的 pooling 方式
 - 获得 server context
 - 释放 server context



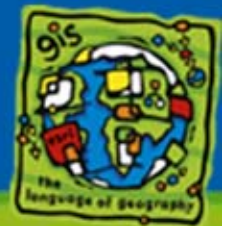
访问 GIS Server

- Impersonate 访问 GIS server 的帐号
 - 必须是 agsusers 组的成员
- Java ANT 建立命令
- ANT Build 创造应用
 - 输入用户的 domain, username, 和 password
 - 帐号信息储存在 arcgis_webapps.properties 文件里



在 Java 中部署应用

- ANT 随 Java ADF 一起安装
- 用 “*arcgisant build*” 来唤起 ANT build scripts
 - 确定所有的资源都包含在 classpath 里
 - 新建一个 <application-name>.war
- 用 “*arcgisant <servlet-container>-deploy*” 来部署这个应用
 - 在部署应用之前编辑 *common.properties* 文件



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

总结

演示

问题？



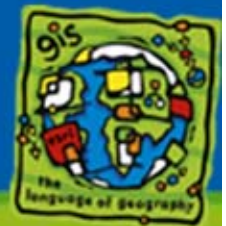
进一步开发

- 为什么?
 - 增加较已提供的即时可用的用法更多的功能
- 怎样做?
 - 新建一个新的 Web 应用或是扩展一个模板
 - 用 Server API 和 Web Controls API 和 全 ArcObjects 语句
 - 增加新的工具和命令



增加工具

- 什么是工具?
 - 用户方的行为 + server方的工具行为组
- ADF 为用户方行为提供 JavaScript library
 - MapDragRectangle, MapPoint, MapCircle, ...
 - PageDragRectangle, PagePoint, ...
- ADF 提供server方的工具行为组
 - ZoomInToolAction, ZoomOutToolAction, IdentifyToolAction, PanToolAction, SelectionToolAction, ...
 - PageLayoutZoomInToolAction, PageLayoutZoomOutToolAction, ...



增加工具 – 举例

- 用 JSP 展现工具

```
<ags:context id="mapContext" resource="Mexico@host" >  
  <ags:map id="Map0" ... />  
  <input type="button" value="Zoom"  
    onclick="MapDragRectangle('Map0', 'ZoomIn');"/>  
</ags:context>
```

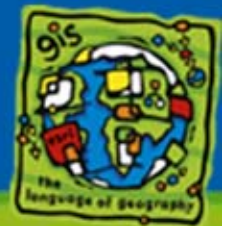
- 用 “key” 连接 server 方的行为 (tools/default.xml)

```
<tool-item>  
  <key>ZoomIn</key>  
  <action-class>  
    com.esri.arcgis.webcontrols.ags.faces.event.ZoomInToolAction  
  </action-class>  
  <client-action>MapDragRectangle</client-action>  
</tool-item>
```



增加命令

- 什么是命令?
 - 被一个事件激起的 Server 方行为
- 新建命令, 通过把 JSF 命令内容捆绑到:
 - Server 方的行为监听器
 - JavaBeans 行为方式
 - 方式后面的程序语句 (只限 IDE)
- ADF 提供的行为监听器:
 - FixedZoomListener, DirectionalPanListener, ZoomFullExtentListener, PageLayoutDirectionalPanListener, ...
- ADF 提供的行为方式:
 - doFind, doBuffer, doClassify, ...



增加命令 – 举例

- 用行为监听器来视窗地图

```
<h:commandButton ... >  
  <h:actionListener value=  
    “com.esri.arcgis.webcontrols.ags.faces.event.ZoomFullExtentListener” />  
</h:commandButton>
```

- 用行为方式来执行“寻找”

```
<h:inputText value=“{#sessionScope['context'].attributes  
  ['esriAGSFindResults'].searchString />  
<h:commandButton action=“{#sessionScope['context'].attributes  
  ['esriAGSFindResults'].doFind />
```

注意:

- AGSFindResults class 为执行“寻找”提供商业逻辑
- context 新建 AGSFindResults 并存储 object

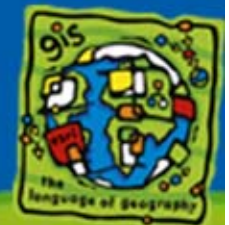
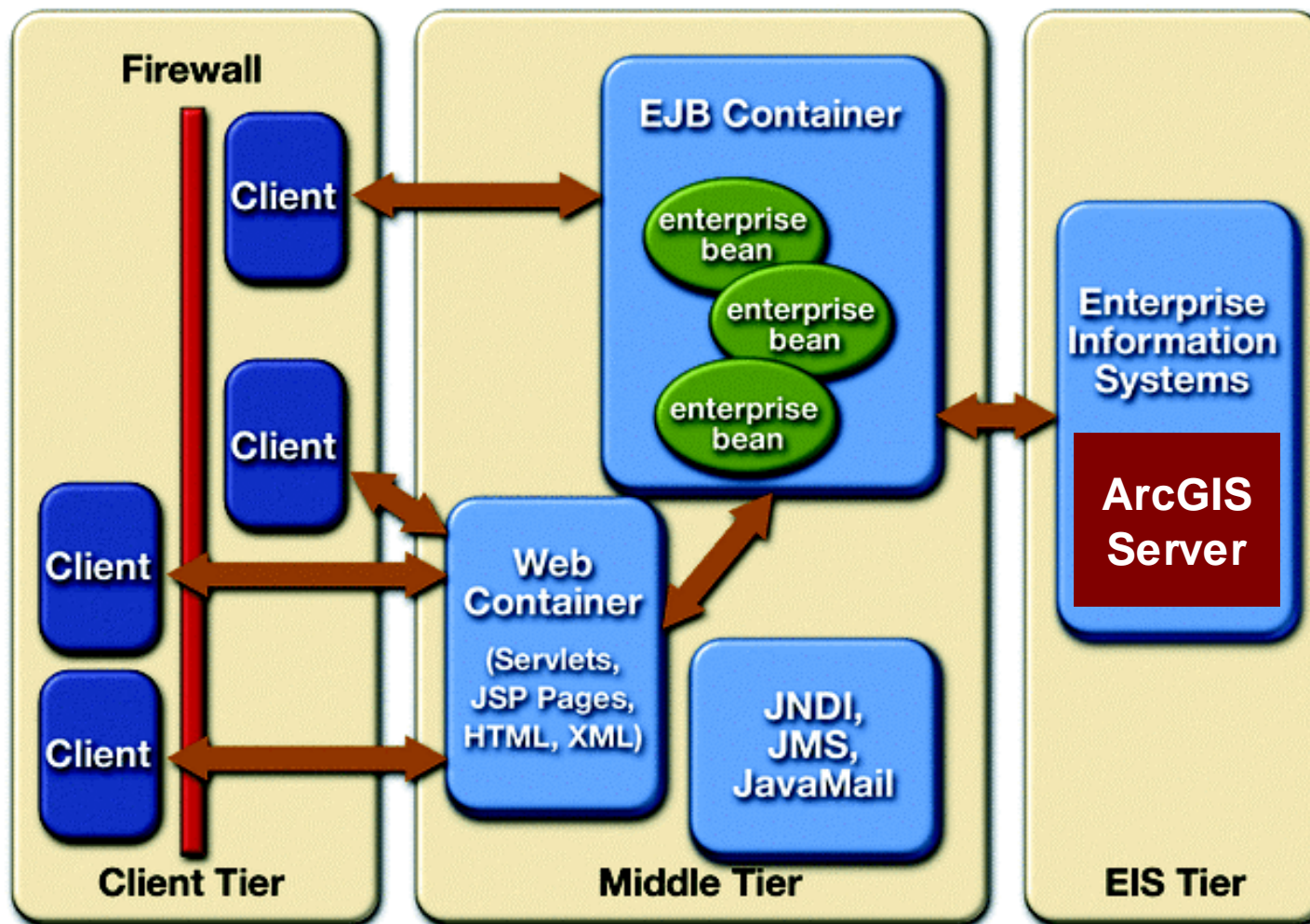


Enterprise Java Beans

- EJB 构造是
 - 可度量的
 - 即时执行的
 - 多用户的
 - 安全的
- J2EE Connector Architecture (JCA) 是
 - 用 EIS 连接 EJBs 的标准
 - 可度量的, 安全的, 和可即时执行的
 - 由 EIS 提供商提供资源调试器 (Resource Adapter - RA)
 - RA 嵌入应用服务器里
 - 这能够确保在 EIS, 应用服务器, 和大型企业应用之间的连接

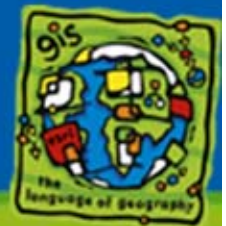


J2EE Enterprise Application

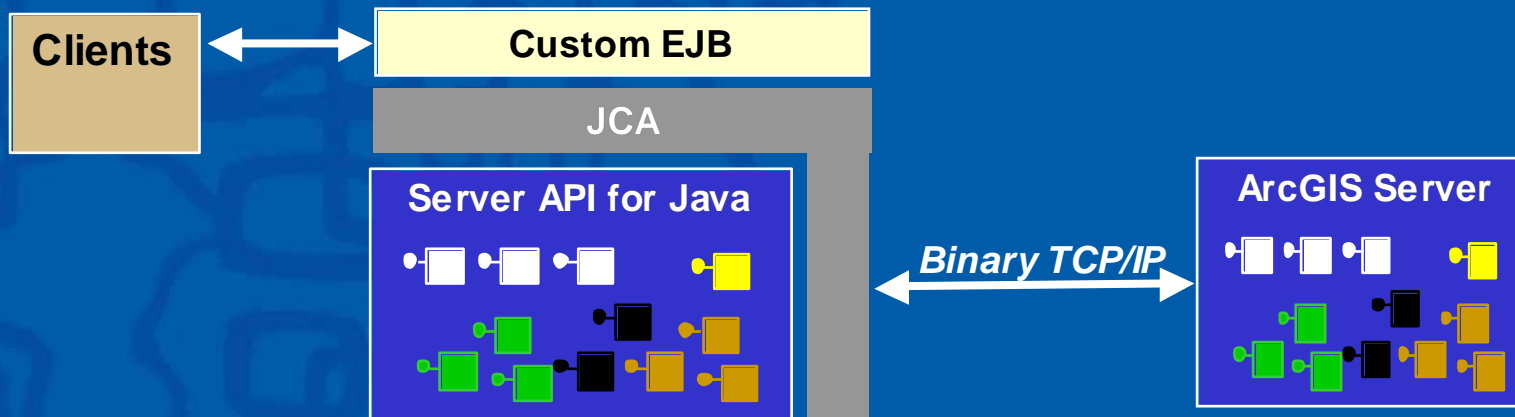


为什么用 JCA ?

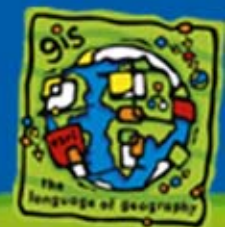
- EJB 规则规定你不能:
 - 新建非标准 class loaders
 - 读或写非动态最终 fields
 - 新建或管理 threads
 - 使用 thread 同步起始
 - 装载 native library
 - 监听网络 socket
 - 新建, 修改或是去掉文件
 - 读或是写文件描述器
 -



J2EE Connector Architecture



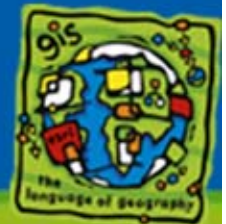
- 资源调试文件
 - \$/ArcGIS/Java/**intrinsycjca.rar**
 - \$/ArcGIS/Java/**intrinsycjca.jar**
- 资源
 - Server_Developers_Guide.pdf 的附录C
 - JavaDoc
 - 开发人员环境



IDE 支持拖拽 Web Controls

- IBM WSAD 5.1.2 含有空间调试器嵌入 (IBM 04年十月发行)
- Sun Creator (还没有确定发行日期, 在ESRI 用户会议上有展示)
- 调研和其它支持 JSF 的 IDEs 的融合

注意: JSF 定义会在不久的将来提供和IDE融合的标准



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

总结

演示

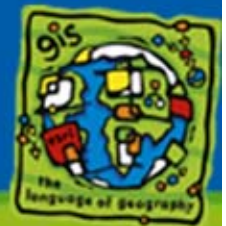
问题？



总结

你可以认为 Java ADF 是:

- 一套表现为 JSP tags 的用户化 Web controls.
- 一套可用于你 Web 应用起始点的模板.
- 一个访问GIS server 的用户应用的接口.
- 一个 J2EE Connector Architecture (JCA) 资源调和器, 允许 Enterprise JavaBeans (EJB) 来调用 ArcGIS Server objects 并和它一起工作.



总结

- ArcGIS Server API 提供用 Java 全访问 ArcObjects
- Web Controls 和 Templates 简化 Web 应用开发
- 使用 ArcGIS Server API 扩展 WebControls API 和写新的 GIS功能



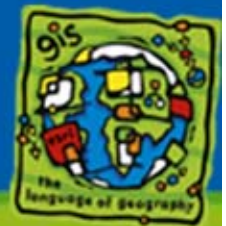
总结

- Java ADF 支持每一个应用层次的多种技术
- 开发人员, 需求, 构造和设计理念决定应做何种选择
- 独立于J2EE 供应商 vendor 和操作系统
- JCA 提供和 EIS 的紧密融合



将来计划

- Web Controls 将会在 9.0 SP2 提供 portlet 技术
- 新的 Controls
 - 工具条 Control
 - 多来源 Map Control (叠加 AGS, ArcIMS, WMS, ...)
- 新的利用 Portlets 的 Web 应用
 - Map Explorer
 - Catalog Explorer



资料

- <http://java.sun.com/j2ee/javaserverfaces>
- <http://www.jsfcentral.com>
- http://java.sun.com/blueprints/guidelines/designing_enterprise_applications/
- <http://java.sun.com/index.jsp>
- IBM Redbooks - <http://www.redbooks.ibm.com/>



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

总结

演示

问题？



内容

综述

ArcGIS Server Java API

Web 控件和模版

Web Services

开发中的重要概念

进一步开发

总结

演示

问题？

