



河北师范大学软件学院
Software College of Hebei Normal University



Android 基础开发

第十二章 Android中的网络应用



Java与移动智能设备开发



教学目标

- 掌握Android中创建网络应用的方法



目录

- 1 基于TCP协议的网络通信
- 2 使用URL访问网络资源
- 3 Webservice网络编程
- 4 Webview视图



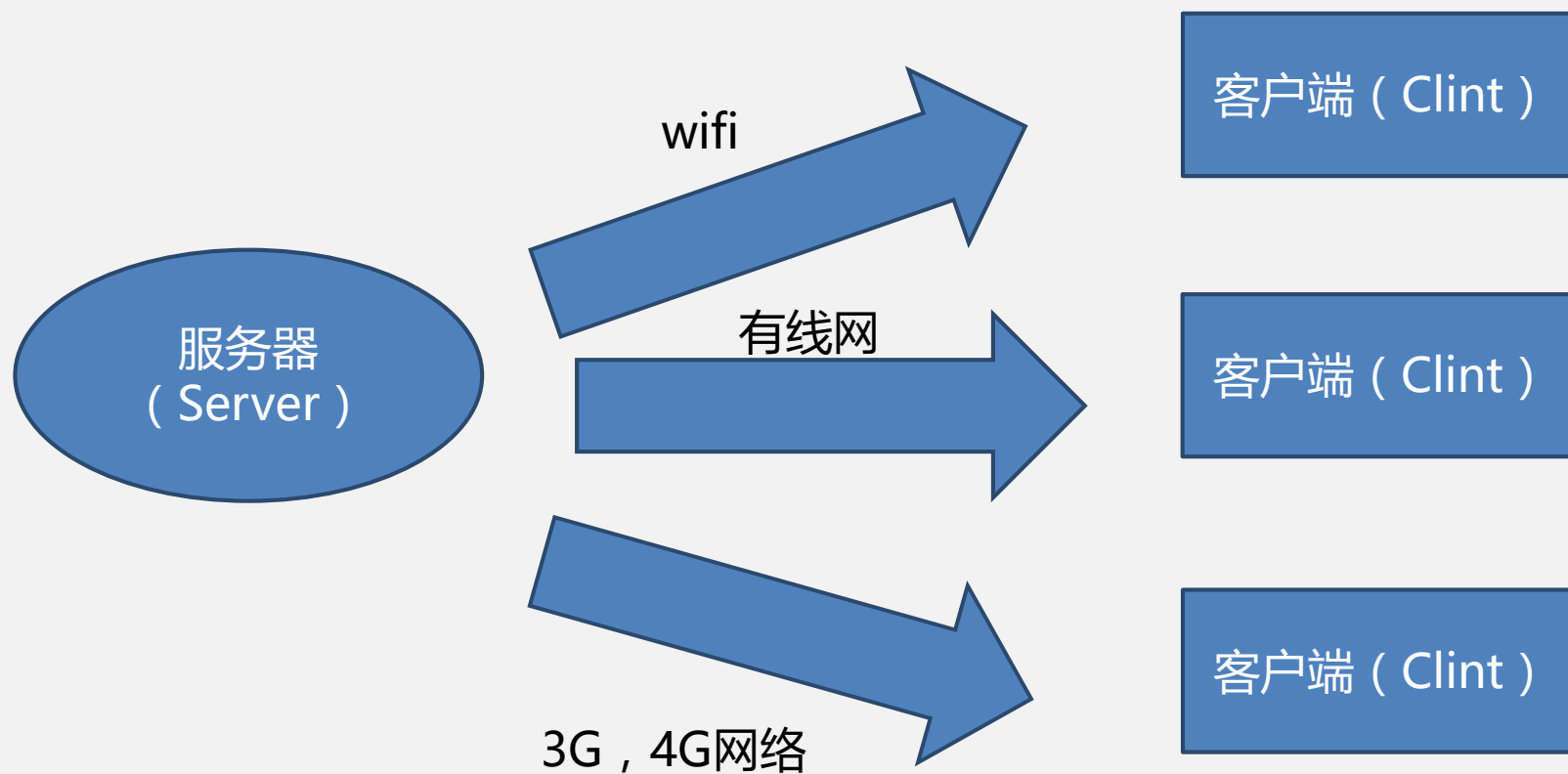
手机的使用

服务器





网络





JAVA中的网络访问形式

- TCP/IP网络通讯协议：
 - ServerSocket
 - Socket
- UDP网络通讯：
 - DatagramSocket
 - DatagramPacket
 - MulticastSocket



Android的网络访问方式

- Android中访问网络的方式：
 - HttpClient网络
 - URL访问网络
 - URLConnection访问网络

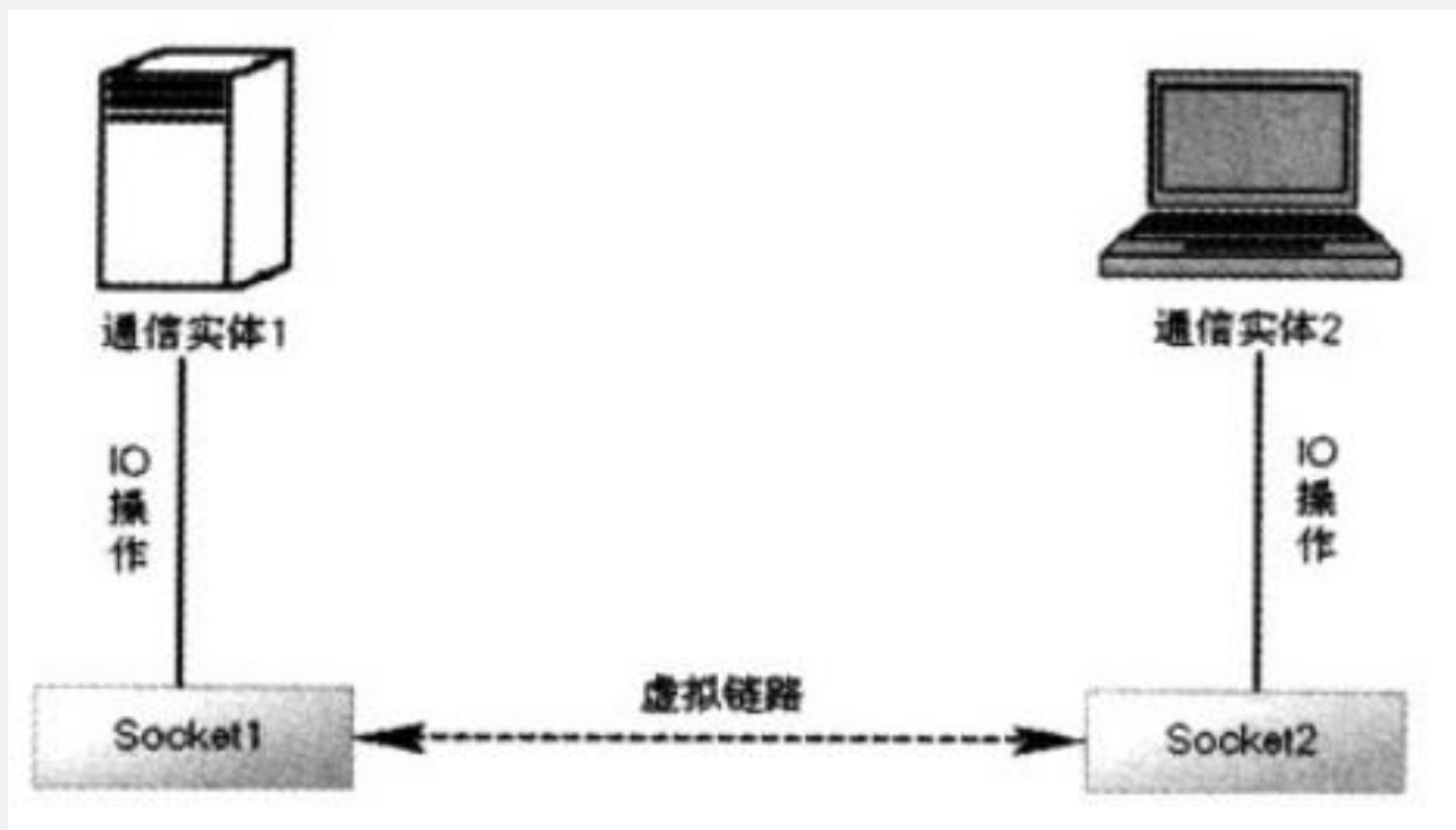


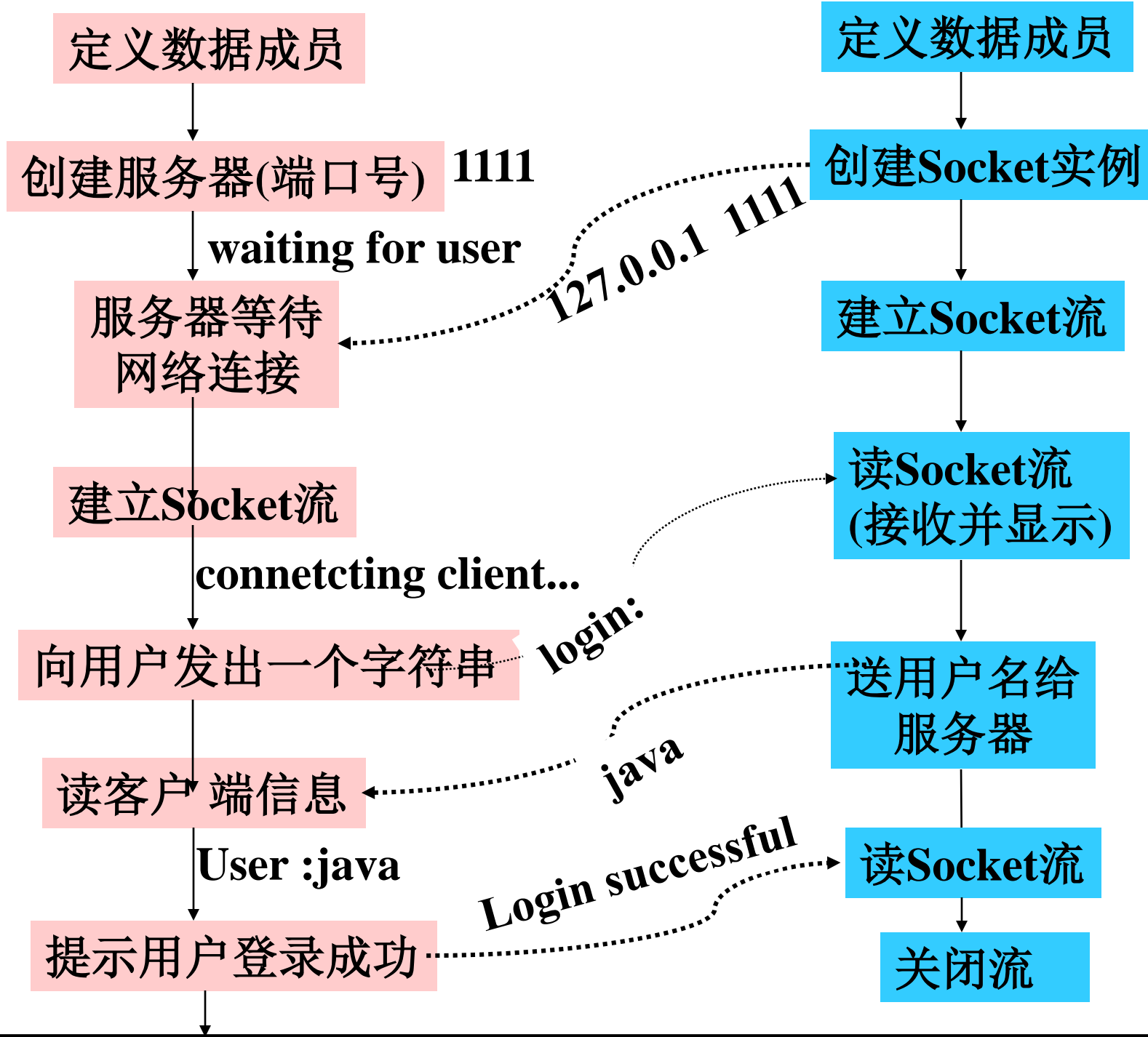
TCP/IP协议

- IP协议是两台计算机之间通讯的语言，保证计算机能发送和接受分组数据，IP协议负责将消息从一个主机传到另一个主机，以及分包处理。
- TCP协议是一种端对端协议。TCP协议在传输过程中负责收集信息包，并按照适当的顺序排列好，在接收到数据包后，将其还原。TCP保证数据包在传输过程中的准确性。
(重发机制)



TCP/IP协议







使用Socket模拟网络通讯

- Java中可以通过Socket实现网络通讯，其中使用到的最主要的类有：
 - ServerSocket类
 - Socket accept()：接受客户端的连接请求，收到连接返回连接客户端的Socket；否则一直处于等待状态
 - ServerSocket(int port)：初始化ServerSocket类
 - 通常一个服务器不应该只接受一个客户请求，所以在java中通过循环来不断调用accept方法



使用Socket模拟网络通讯

– Socket类

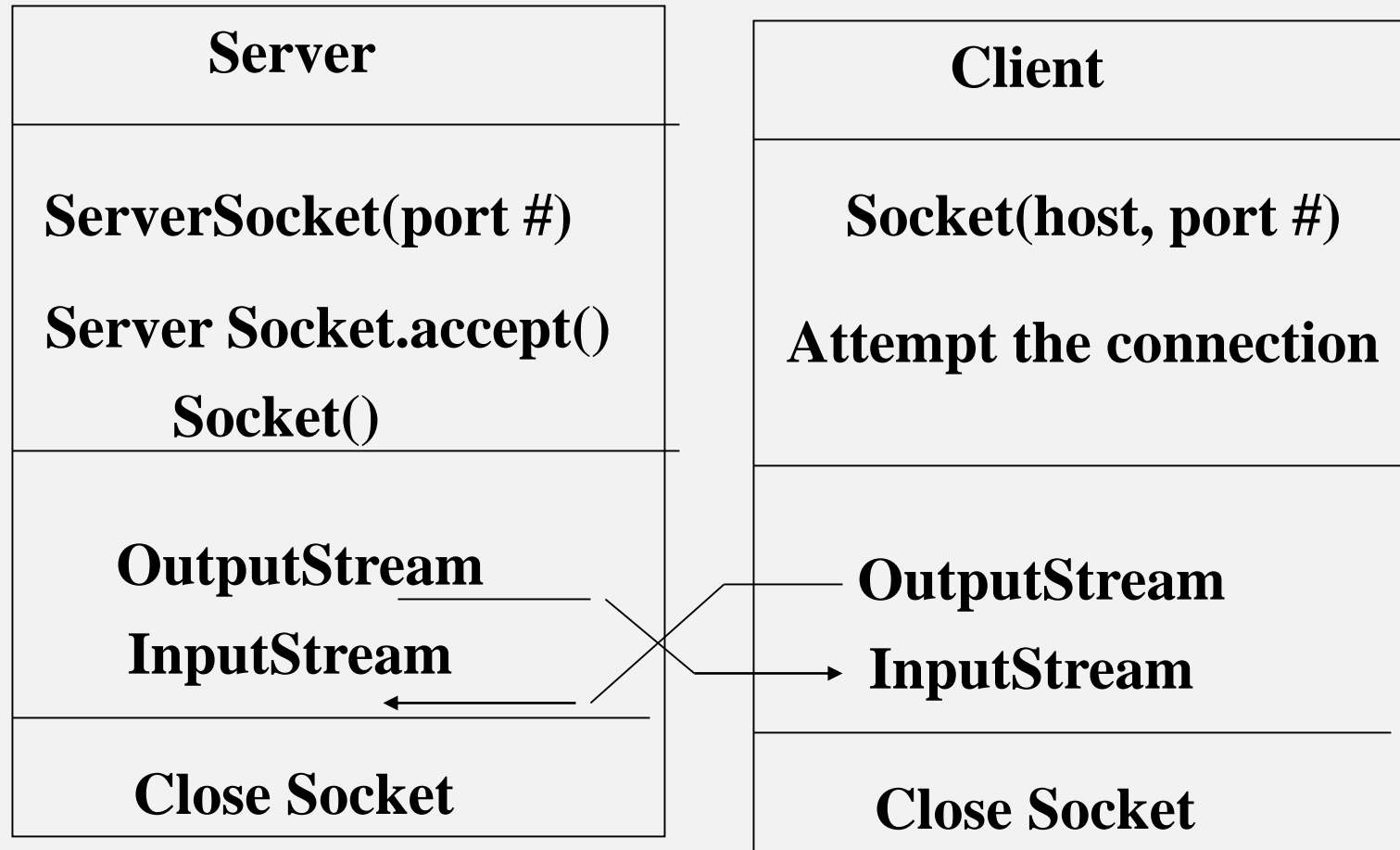
- `Socket(string remoteadd , int port)` : 初始化Socket类。
当调用这个方法时客户端连接到服务器端。

– 服务端客户端连接后的数据传

- `InputStream getInputStream` : 返回该Socket对象对应的输入流
- `OutputStream getOutputStream` : 返回对象对应的输出流



Socket通信





使用Socket模拟网络通讯

- Socket经常是长时间通讯
- Socket经常设计大量数据读取，连接很长
- 所以一般使用多线程处理Socket连接



目录

- 1 基于TCP协议的网络通信
- 2 使用URL访问网络资源
- 3 Webservice网络编程
- 4 Webview视图



URL

- URL (Uniform Resource Locator) 统一资源定位器，指向互联网“资源”的指针。
- 一般的URL组成：
 - <http://www.baidu.com/index.php>
 - protocol://host:port/resourceName
- Android中提供了URL类表示URL对象：
 - String getFile () : 获得URL的资源名
 - String getHost () : 获得URL的主机名
 - int getPort () : 获得URL的端口号



使用URL读取网络资源

- URL对象中的openStream方法可以读取URL资源的InputStream，从而获得远程资源。
- Demo
 - 获得网络图片
 - 文件下载



使用URLConnection提交请求



- URL的openConnection() 方法将返回一个URLConnection对象，表示应用程序和URL的网络连接，然后程序可发送请求。
- 创建URL的连接的步骤如下：
 1. 调用URL对象的openConnection方法得到URLConnection对象。
 2. 设置URLConnection对象的参数和请求属性。
 3. 发送get请求或post请求。
 4. 获得远程资源。



使用URLConnection访问



- 使用URLConnection访问：
 - `Int getResopnseCode ()` : 获得服务器响应代码。
 - `String getResponseMessage ()` : 获得响应消息。
 - `String getRequestMethod ()` : 获得请求方法。
 - `void setRequestMethod (method)` 设置发送方法。



目录

- 1 基于TCP协议的网络通信
- 2 使用URL访问网络资源
- 3 Webservice网络编程**
- 4 Webview视图



Web Service

- Android应用通常运行在嵌入式设备中，其计算能力，系统资源都是有限的，所以一般Android的应用的数据来源于网络中的计算机，Android充当其客户端使用。
- Android应用如何获取服务器端的数据呢？

Web Service



Web Service

- Web Service用于消除不同平台，不同语言之间的实现差异。将已有的应用发布成开放式服务，允许互联网上任何地方，任何平台，任何语言进行访问。
- 常见的Web Service有：
 - SOAP方式：比较早期
 - REST方式：最近的新行形式



Web Service-SOAP

- SOAP方式的Webservice主要涉及如下技术：
 - SOAP(Simple Object Access Protocol)
 - WSDL(Web Service描述语言)
 - UDDI(统一描述，发现和整合协议)

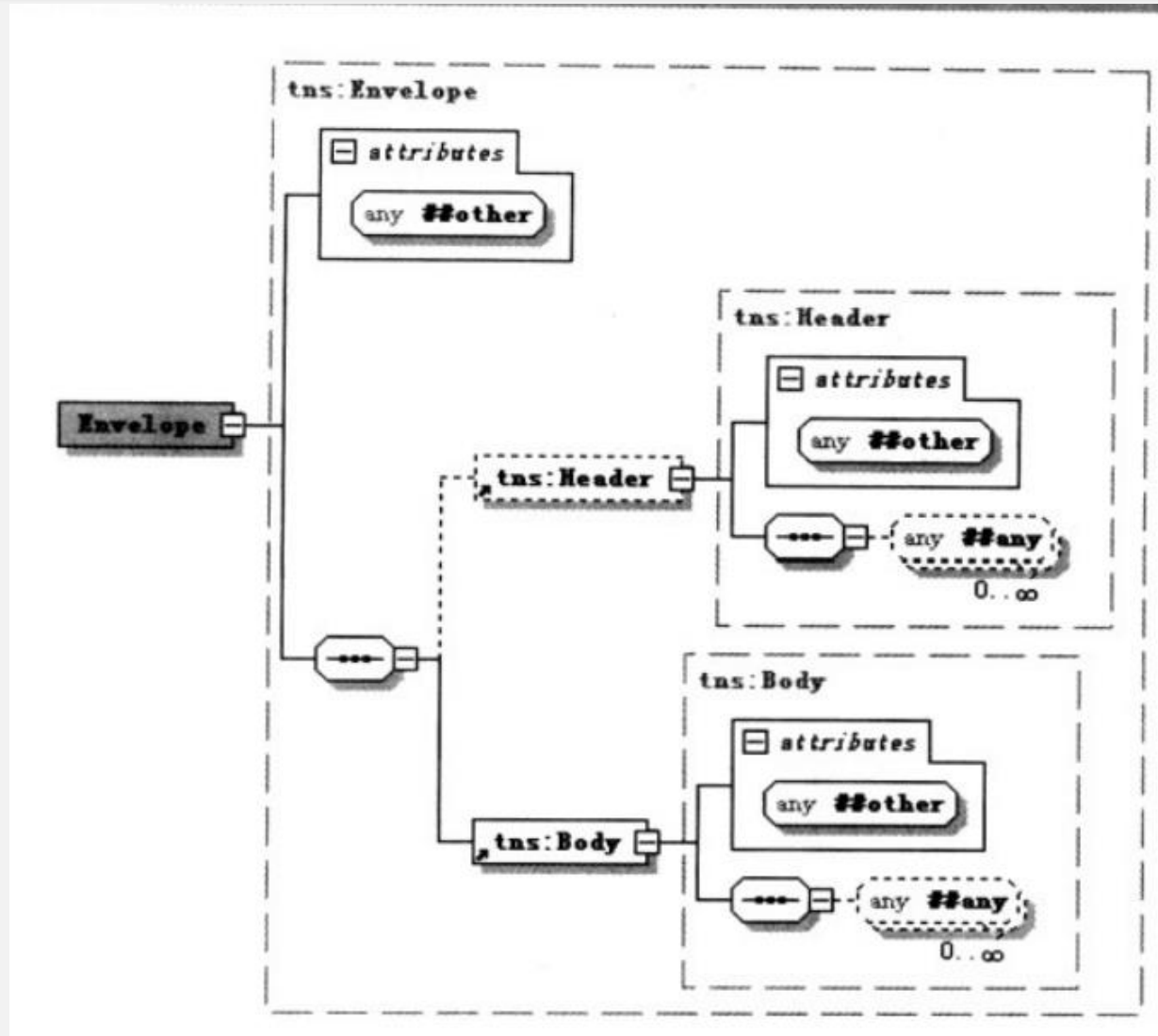


Web Service-SOAP

- SOAP(Simple Object Access Protocol , 简单对象访问协议)它允许一个应用程序向另一个应用程序发送XML消息。SOAP能通过HTTP , JMS , SMTP等协议传输。
- SOAP依赖于XML文档构建 , 主要包含如下元素 :
 - 必须的<Envelop... />根元素
 - 可选的<Header.../>包含SOAP的消息头信息
 - 必须的<Body... />包含所有的调用和相应信息



Web Service-SOAP





Web Service-SOAP

- WSDL(Web Service Description Language)使用XML描述Web Service。主要包括如下部分：
 - WHAT部分：用于定义Web Service所提供的方法，由 `<type.../>`、`<message.../>`、`<portType.../>` 元素
 - HOW部分：用于定义如何访问Web Service。定义了数据详情和访问Web Service的必要协议。
 - WHERE部分：用于定义Web Service位于何处，网络地址。`<service.../>` 元素定义

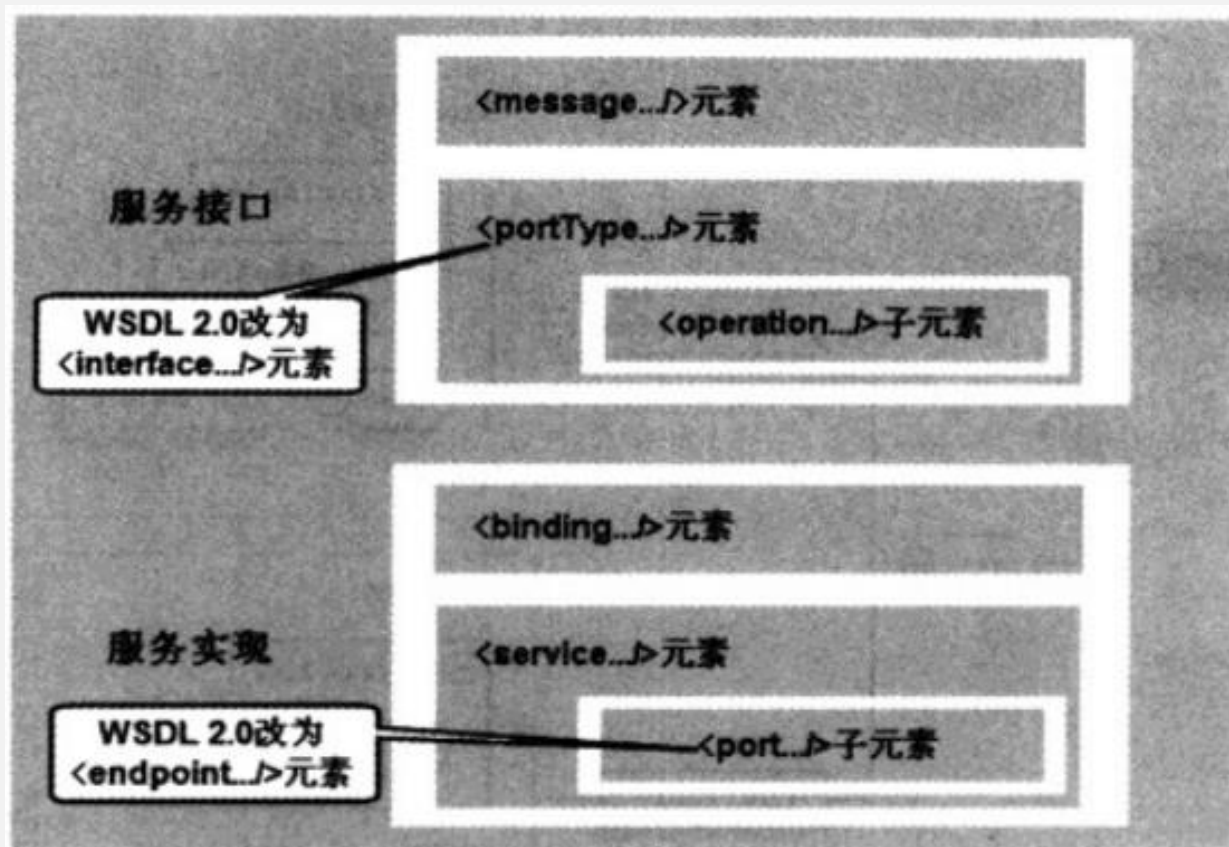


Web Service-SOAP

- 一般WSDL分为两部分呢：
 - 第一部分定义服务接口：
 - `<message.../>` 定义操作方式
 - `<portType.../>` 中的 `<operation.../>` 节点定义远程调用操作
 - 第二部分定义服务的实现：
 - `<binding.../>` 定义特定的通信协议，数据编码
 - `<service.../>` 包含 `<prot.../>` 节点，实现绑定机制，服务访问协议，断点地址。



Web Service-SOAP





Web Service-SOAP

- UDDI(Universal Description , Description andIntegration , 统一描述 , 发现整合协议)是一套注册规范。
- 企业向外注册Web Service的时候使用的协议。



Web Service- REST

- REST(Representational State Transfer表述性状态转移)是一种针对网络应用的设计和开发方式，可以降低开发的复杂性，提高系统的可伸缩性。
- REST其实并不是什么协议也不是什么标准，而是将**Http协议**的设计初衷作了诠释，在Http协议被广泛利用的今天，越来越多的是将其作为传输协议，而非原先设计者所考虑的应用协议。
- 很多国外的网站基本都提供了REST服务。



目录

- 1 基于TCP协议的网络通信
- 2 使用URL访问网络资源
- 3 Webservice网络编程
- 4 Webview视图**



WebView视图

- Webview 视图是一个Android中的组件，本身就是一个浏览器。
- 通过WebView可以进行一般网页浏览器的功能：
 - void goBack() : 后退
 - void goForward() : 前进
 - void loadUrl(String url) : 加载制定URL的网页
 - boolean zoomIn() : 放大网页
 - boolean zoomOut() : 缩小网页



内容回顾

- 1 基于TCP协议的网络通信
- 2 使用URL访问网络资源
- 4 Webservice网络编程
- 5 Webview视图



Thank you!

