



软件体系结构原理、方法与实践

严禁复制

❁ RIA的概念 – 背景

例：我们需要修改用户的详细信息，提交后显示修改后的信息，修改页面和显示页面是同一个页面。

问：如果不同的内容需要提交到不同的服务端地址有哪几种实现方法？

❁ RIA的概念 – 传统WEB程序的缺点

➤ 操作复杂性

由于受传统Web应用程序的局限性，当进行一个多步骤或多选项的事务时，用户要么会看到一份很长的、笨拙的页面，要么就得通过反复翻转若干网页、执行多步操作。

➤ 数据复杂性

高效率地表达复杂的数据，是现有Web应用程序所面临的巨大挑战。理想的图形工具应该能够既操作简便，又能生动明了地展示各种错综复杂的数据信息。

➤ 交互复杂性

互动性需求的应用程序使得交互的问题变得日益突出，用户的耐心变得越来越少，他们的要求是要向桌面应用程序的速度看齐。

关于传统WEB程序存在的缺点描述正确的是？

A

当进行一个多步骤或多选项操作时，用户会看到一个很长的笨拙的页面，得通过反复翻转多个网页或复杂的操作步骤来完成

B

很难高效率的展示错综复杂数据，并进行简单有效的操作。

C

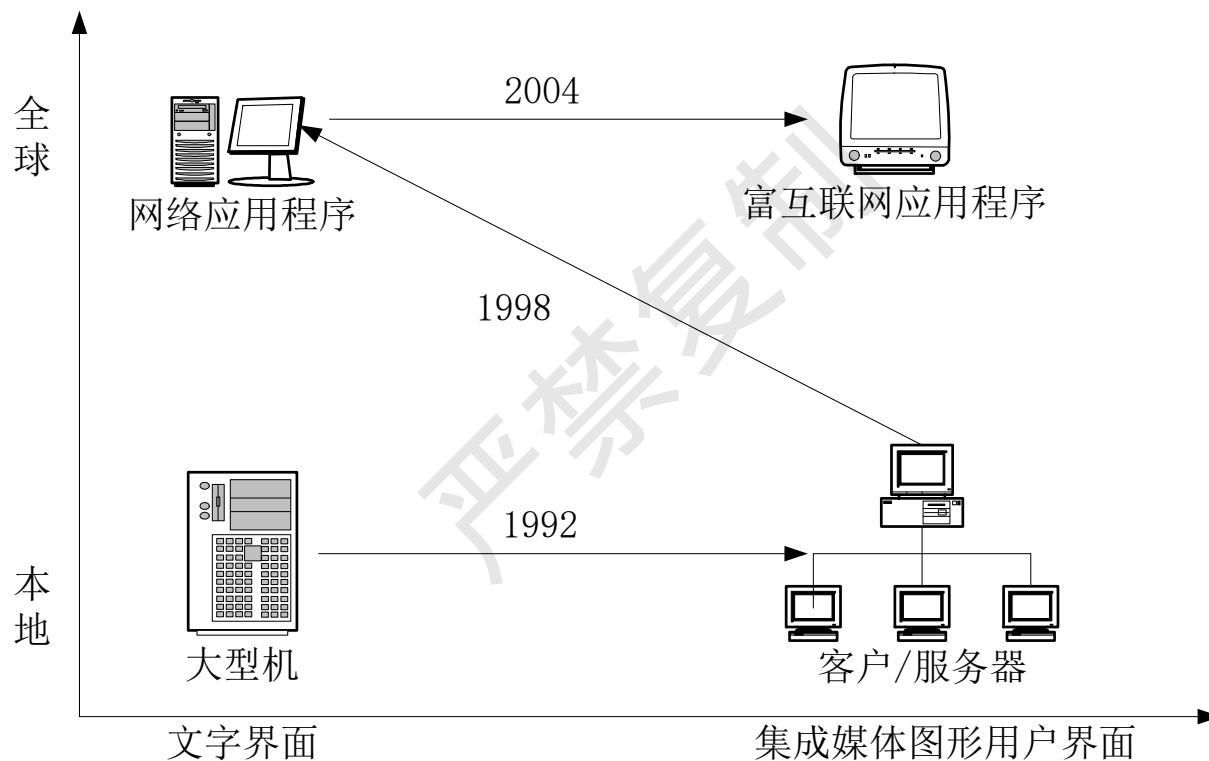
交互性复杂，很难满足日益突出的交互性需求。

D

服务器和网络的技术支持能力有限

提交

RIA的概念 – 发展过程



❁ RIA的概念 – “富” 的含义

➤ 丰富的数据模型

丰富的数据意味着客户端的用户界面能表现和应对更多更复杂的数据模式，这样才能处理客户端的运算以及异步发送、接收数据。为了达到高度复杂的数据模式,客户端允许用户构建一个高响应、交互式的应用程序。

➤ 丰富的用户界面

HTML只能为用户的界面控制提供有限的功能，反之，RIA允许一些富有创造性的界面控制，巧妙地与数据模式相合。伴随着丰富的用户界面，用户可以从早期的服务器响应影响整个界面的运作模式，迁移到只对发出请求的特定区域进行改变的模式上来。本质上，意味着界面将会被分解为由单独个体组成，来适应局部改变、服务器交互，以及客户端内部构件的通信。

❁ RIA的概念 – RIA的优点

- RIA 使用了相对健壮的客户端描述引擎,这个引擎能够提供内容密集、响应速度快和图形丰富的用户界面。
- RIA 在客户端实现数据缓存,从而可以实现一个比基于 HTML的响应速度更快且数据往返于服务器的次数更少的用户界面。
- RIA提高丰富了客户机的功能,客户机在RIA中的作用不仅是展示页面,它可以在幕后与用户请求异步地进行计算、传送和检索数据、显示集成的用户界面和综合使用声音和图像,这一切都可以在不依靠客户机连接的服务器或后端的情况下进行。
- 对企业来说无需大规模替换现有WEB应用程序,通过 RIA 技术,可以轻松构建更为直观易于使用、反应更迅速并且可以脱机使用的应用程序。
- 为企业带来多元化的效益,包括提高产品销量提高品牌忠诚度延长网站逗留时间、较频繁的重复访问、减少带宽成本、减少支持求助以及增强客户关系等。

关于富互联网体系结构的说法正确的是

A

RIA是WEB开发和部署的一种演变，所谓“富”包括丰富的数据模型和丰富的用户界面

B

客户机在RIA中实现了异步计算，传输和检索数据,声音和图像集成等功能

C

RIA利用客户端缓存技术降低界面与服务器端的交互频次。

D

RIA使用客户端引擎实现内容密集，快速响应，图形丰富的用户界面。

E

RIA可帮助企业提供多元化的重要业务效益，无须对现有程序进行大规模替换

提交

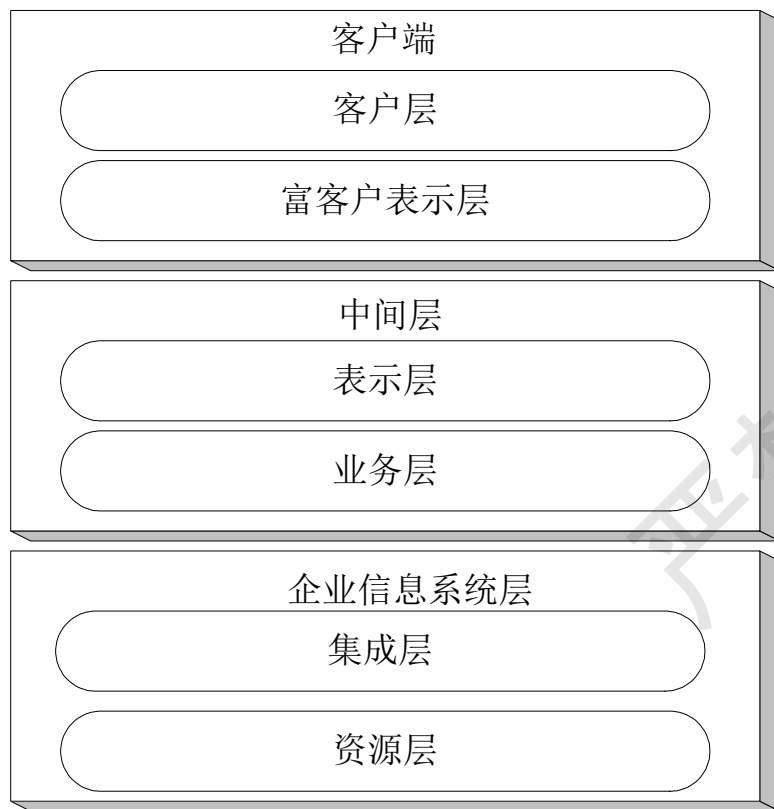
✿ RIA模型



- 客户层包含用户界面和程序接口的容器；
- 表示层包含表述内容的逻辑、处理用户会话、状态管理；
- 业务层包含系统的业务逻辑；
- 集成层包含访问远端程序和数据源的连接器和适配器；
- 资源层包含数据库资料以及像ERP这样的企业信息资源和XML文件。

典型网络部署系统的结构如图

❁ RIA模型



富客户端模型

富客户可以支持更小的单元或构件，这些构件从小到一个投票问题，大到一个完整的视图或界面，富客户模型将界面分解成许多的既可以和用户直接交互又可以和服务器进行通信的小单元模块。

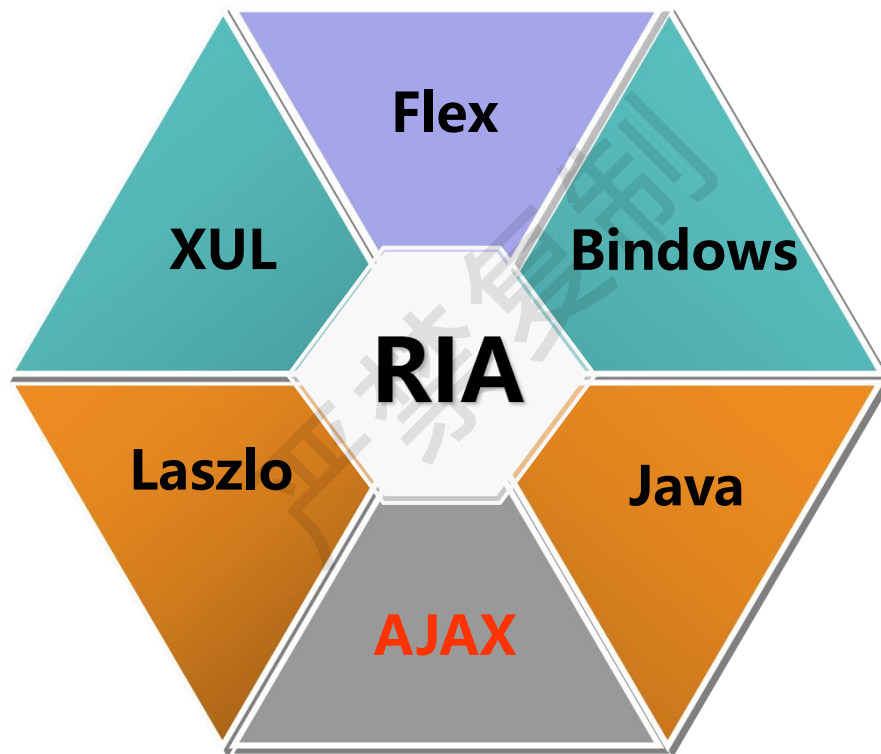
这样用户就可以控制构件创建信息发送给服务器和处理服务器的响应，可以为更零散的控制去耦合和分离程序功能并且建立面向服务的程序结构。

关于RIA模型说法正确的是:

- ☒ A 传统web应用程序是一个基于客户浏览器的请求响应模式，界面传递需要在中间层进行以HTML的形式的格式化和编译
- ☐ B 在RIA中客户端和服务端的交互是同步的。
- ☒ C 富客户端模型将界面分解成许多小单元模块，这些小单元模块既可以直接和用户交互又可以和服务器交互。
- ☐ D 富客户端模型是一个事件模型，用户需要预测事件的顺序
- ☒ E 传统模型中对任何表示层的修改都需要向服务器发送请求，做出的响应是整个页面的而不仅仅是只改动部分的。

提交

❁ RIA客户端开发技术



❁ RIA客户端开发技术 – AJAX

Asynchronous JavaScript And XML

基于XHTML和CSS标准的表示

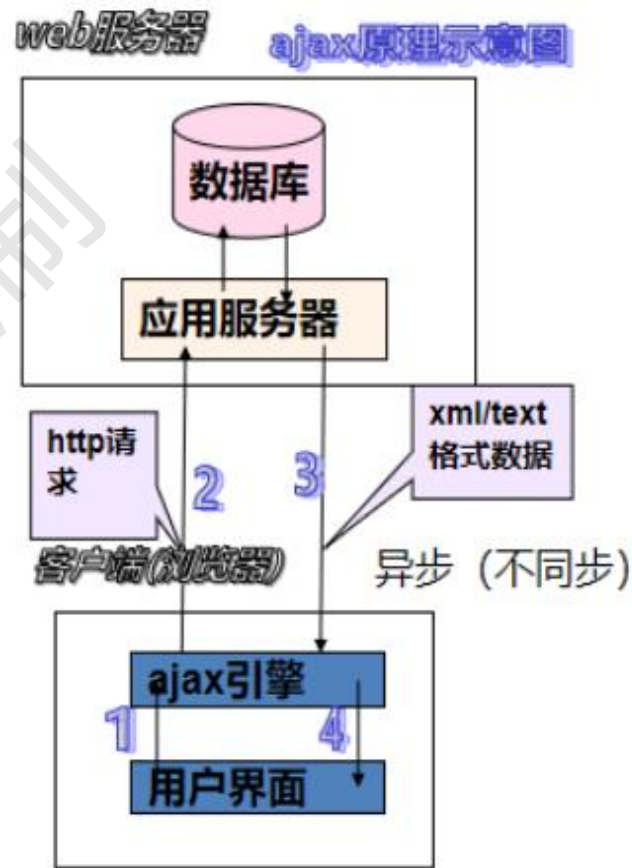
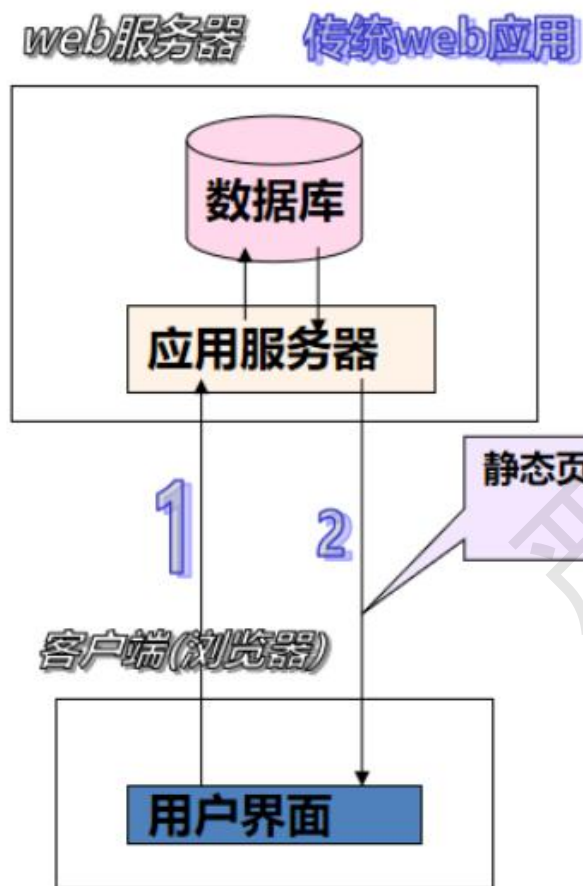
使用DOM进行动态显示和交互

使用 XML 和 XSLT 进行数据交换及相关操作

使用XMLHttpRequest与服务器进行异步通信

使用JavaScript绑定一切

RIA客户端开发技术 – AJAX



❁ RIA客户端开发技术 – AJAX

优点:

异步交互，局部更新。

改进的传统web程序数据复杂性，交互复杂性，操作复杂性的缺点

缺点:

破坏浏览器“后退”按钮的正常行为；

使用动态页面更新时，用户难以将某个特定的状态保存到收藏夹中。

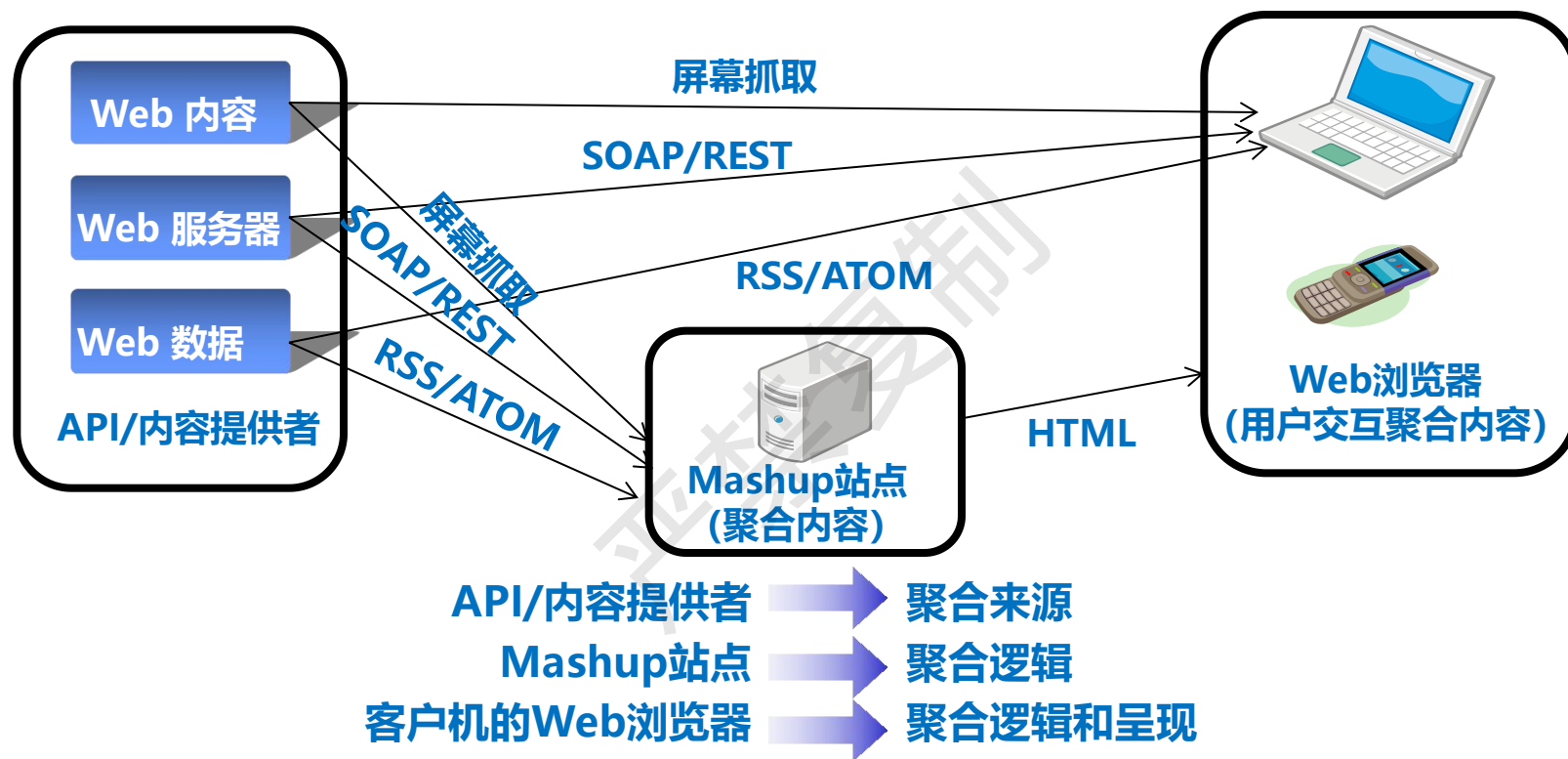
使用AJAX开发时要慎重考虑网络延迟

关于AJAX技术说法正确的是：

- ☒ A 可以使用JavaScript进行绑定
- ☐ B AJAX不能读取XML格式的数据
- ☒ C AJAX基于XHTML和CSS标准因此能够对CSS样式进行操作
- ☐ D AJAX使用HttpRequest与服务器进行通信
- ☒ E AJAX是由XHTML, CSS, DOM, XMLHttpRequest等几种技术以新的方式组合而成的。

提交

Mashup技术 – 体系结构



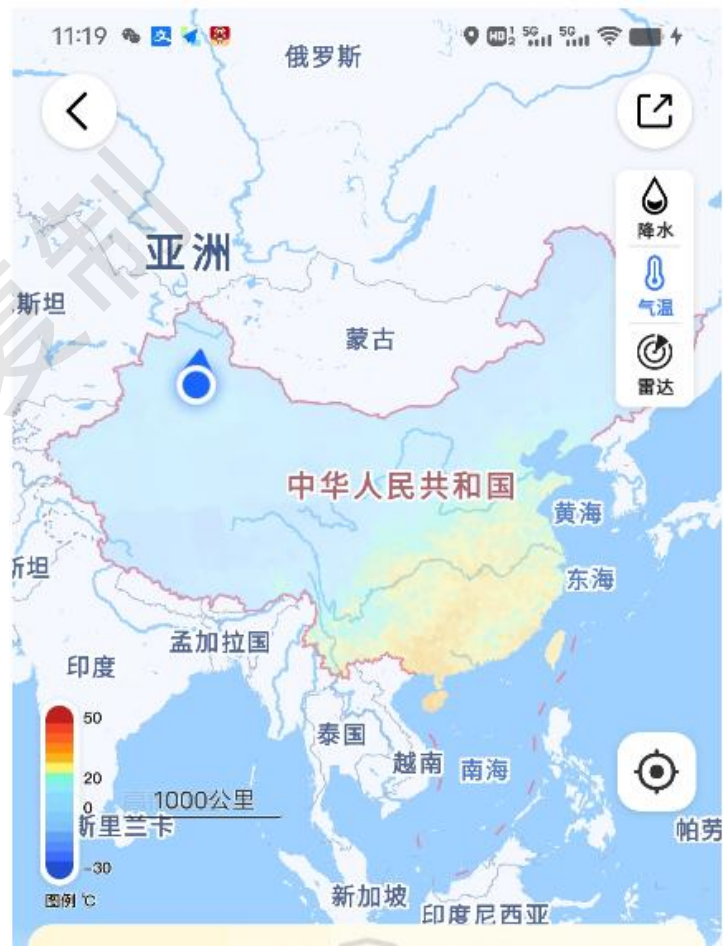
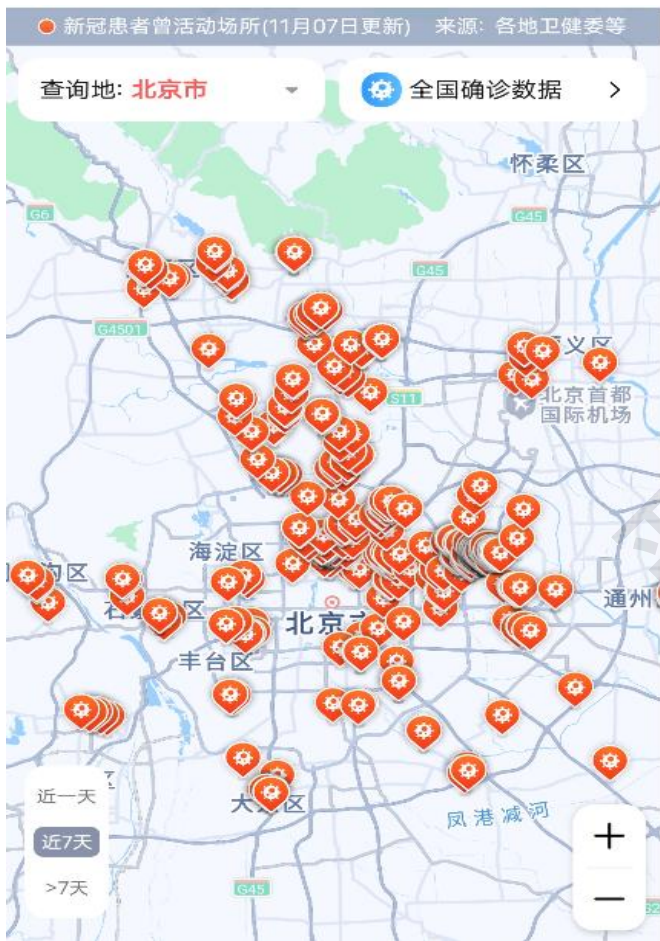
RSS：一种用于对网站内容进行描述和同步的格式，是目前最广泛的Web资源发布方式

ATOM：一种基于XML的文档格式和基于HTTP的协议，用来聚合网络内容

REST：从资源的角度看待整个网络，各处的资源由URI确定，客户端的应用通过URI获取资源的表示

SOAP：一种基于XML的数据格式定义，用来进行Web服务调用过程中的参数调用和返回

Mashup技术 - 实现技术



新的开始，新的起点，让我们一起为梦想而努力。

❁ Mashup技术 – 实现技术

- Mashup聚合的内容可概括为服务和数据。如果聚合的是服务，则通过调用API来获取各个源的功能，如果聚合的是数据，则使用RSS或Atom来获取数据。
- 在Mashup聚合时，语义Web和资源描述框架(RDF)可以帮助实现高质量的XML数据聚合和RSS摘要内容聚合。使用语义技术和资源描述框架(RDF)可以让Mashup用户更好地控制服务、信息和表示，高效高质地创建Mashup应用程序。
- Mashup常以Widget形式封装功能和数据源，这种可视化的小部件可供用户使用拖动来实现功能和数据的聚合，使得用户可以真正参与到终端编程中来。

Mashup的体系结构组成包括：

- ☒ A API/内容提供者
- ☒ B Mashup站点
- ☐ C 端口和角色
- ☒ D 客户机的WEB浏览器

提交