

基于 SOAP 协议的统一身份认证服务设计与实现

罗 婵,董丽丽,马宗方

(西安建筑科技大学 信息与控制工程学院,陕西 西安 710055)

摘 要:企业门户系统的目标是消除企业信息孤岛,无缝集成各种应用系统并实现统一身份认证。运用 SOAP 的消息机制和 Web Services 的核心概念及运行机制,提出了基于 SOAP 协议的统一身份认证系统架构,解决了集成新旧系统时出现的关键问题。采用 .NET 技术实现了原型系统,实现了用户统一身份认证,从而完成了在异构平台、异构数据库下的企业数据流、应用流的全面集成。

关键词:无缝集成;统一身份认证;XML;SOAP;Web 服务

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2006)10-0237-03

Design and Implementation of Uniform Identity Authentication Service Based on SOAP

LUO Chan, DONG Li-li, MA Zong-fang

(School of Information & Control, Xi'an Architecture & Technology University, Xi'an 710055, China)

Abstract: The aim of enterprise portal is to eliminate enterprise information isolated island, seamlessly integrate all kinds of application systems and realize uniform identity authentication. Using the framework of SOAP messages, the kernel concept and the running mechanism of Web services, the architecture of uniform identity authentication system was put forward to resolve the key problem in integrating different systems. At last, a prototype system was implemented using .NET to realize the uniform identity authentication. Therefore, the enterprise data flow and application flow which is under the environment of varied platforms and heterogeneous databases are integrated.

Key words: seamless integration; uniform identity authentication; XML; SOAP; Web services

0 引 言

企业门户是一个联接企业内部和外部的网站,它可以为企业提供一个单一的访问企业各种信息资源的入口,企业的员工、客户、合作伙伴和供应商等都可以通过这个门户获得所需信息和个性化服务。企业门户可以无缝集成企业的各种应用系统。

企业门户系统要集成各种应用系统,这些应用系统可能是原来的系统或者是第三方开发的系统,如协同办公系统、决策支持系统等。此时面临如下问题:首先,原来的应用系统或者新开发的应用系统如何能够在避免对现有系统进行大规模修改的基础上方便地挂接到门户平台上,尽量减少人力与财力的耗费;其次,如果每个应用系统都有其自身的用户系统和认证方式,那么当某个应用系统挂接到门户系统的时候,让用户频繁登录,对于用户和后台管理来说都会造成极大不便。解决第一个问题的方法是对子系统的开发制定一个简单易行的标准,使得不同系统、

不同平台实现无缝集成;解决第二个问题的方法是基于 SOAP 协议开发统一身份认证服务,使得新开发的应用系统与现有系统的权限统一管理、授权和认证,即系统原有的用户可以使用自己已有的账号登录已成功注册、授权的子系统,可以对该子系统中所拥有权限的模块功能进行操作。文中重点介绍第二个问题的解决方法。

1 统一身份认证系统

1.1 基本原理

基于 SOAP 协议构建统一身份认证系统的架构,采用 Web Services 实现统一身份认证、授权。在这种架构下,所有接口的数据格式均采用 XML 描述,服务的请求和响应则利用 SOAP 协议,这样将有利于解决远程认证服务中跨平台的系统互连问题。统一身份认证系统的架构如图 1 所示。

SOAP 客户端的任务与通常的 B/S 结构中 Server 端的任务类似:接收并响应 Browser^[1],接收客户的 POST 请求,加上适当的 SOAP 头,建立 SOAP 请求包,以 HTTP 形式向 SOAP 服务器发出 SOAP 请求,并接受 SOAP 服务器返回的响应结果。

SOAP 服务端负责接收远程的 SOAP 请求,解析

收稿日期:2006-02-16

基金项目:陕西省自然科学基金资助项目(2001x30)

作者简介:罗 婵(1982-),女,安徽六安人,硕士研究生,研究方向为网络与分布式系统;董丽丽,副教授,研究方向为计算机网络应用与分布式系统。

SOAP 报文,执行 SOAP 请求,生成 SOAP 应答报文,返回给 SOAP 客户端。

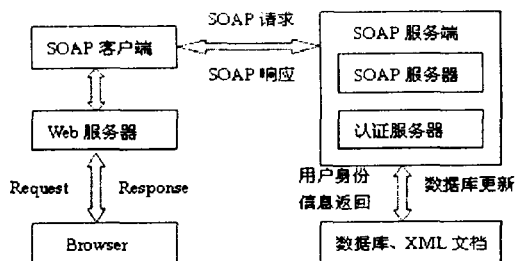


图 1 统一身份认证系统的架构

SOAP 客户端与 SOAP 服务端可以在不同机器上实现,它们之间以 HTTP 协议通讯,传输的数据是 XML 格式的纯文本形式字符串。HTTP 协议和 XML 都是平台无关的,因此 SOAP 客户端与 SOAP 服务端完全可以是异构的。另外,SOAP 服务端的功能实现层可以分布在不同的机器上实现,SOAP 服务端通过调用功能实现层的各个组件完成任务,从而达到负载均衡。用户身份信息主要存储在数据库服务器或 XML 文档中。

1.2 设计与实现

门户平台应该主要着眼于建立企业内部各系统的单点登录和统一授权机制。门户平台提供统一的入口,企业员工从这个入口登录,然后可以根据自己拥有的权限进入相应的子系统及子系统中相应的功能模块。用户在门户平台和各个子系统的权限由管理员统一分配,管理员可以授权给某个用户任何一个系统中的某些功能点,并且随着系统的动态增加,管理员可以动态地授权,而不会影响用户在其它系统中的权限。

管理员根据系统数据库中所保存的功能信息给用户授权。当用户登录门户平台时,门户平台通过调用认证服务发放令牌给用户。当用户请求登录某个子系统时,子系统使用该令牌调用认证服务来对用户身份进行统一认证,如果用户是合法用户,子系统根据用户权限加载相应的功能。图 2 显示了统一身份认证流程。

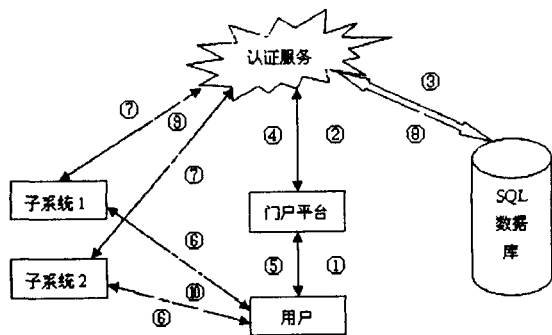


图 2 统一身份认证系统实现原理图

统一身份认证流程可描述如下^[2];

第一步:用户登录门户平台,通过安全连接提交用户名和密码。

第二步:门户平台调用认证服务。

第三步:认证服务在系统数据库中查找用户权限信息

并返回。

第四步:认证服务给门户平台返回用户权限令牌。

第五步:门户平台给用户发放权限令牌。

第六步:用户使用这个权限令牌访问某个子系统。

第七步:子系统向认证服务发送系统标识和用户权限令牌。

第八步:认证服务根据子系统标识和用户权限令牌在系统数据库中查找该子系统授权给该用户的功能点的信息,并返回。

第九步:子系统解析用户权限信息,加载功能内容。

第十步:子系统对用户返回访问结果。

2 统一身份认证系统的 SOAP 消息机制

基于 SOAP 协议的身份认证方法的请求/响应过程经过 4 个阶段:生成身份认证的 SOAP 请求、SOAP 请求的接收和处理、SOAP 响应消息的形成、SOAP 消息的用户响应。

2.1 生成身份认证的 SOAP 请求

SOAP 客户端创建 SOAP 请求并发送到 SOAP 服务器。这些请求以纯文本方式编写,每个请求都有标准的 HTTP 格式头。在创建 SOAP 消息时,需要把附加信息添加到这些标准 HTTP 格式中。

传递给 SOAP 服务器的 SOAP 请求头如下:

```
POST /portal/webservice/AuthenticationService.asmx HTTP/1.1
```

```
Host: 202.200.200.1
```

```
Content-Type: text/xml; charset=utf-8
```

```
Content-Length: length
```

```
SOAPAction: "http://tempuri.org/CheckUser"
```

SOAP 请求内容由一个必需的 SOAP Envelope 组成。SOAP Envelope 包含一个可选的 SOAP Header 和一个必需的 SOAP Body。SOAP Envelope 是表示 SOAP 消息的 XML 文档的顶级元素;SOAP Header 是为支持在松散环境下在通信方之间尚未预先达成一致的情况下为 SOAP 消息增加特性的通用机制;SOAP Body 为该消息的最终接收者所想要得到的那些强制信息提供了一个容器^[3]。调用认证服务的具体 SOAP 请求如下,其中 CheckUser 是认证服务器上被调用的方法之一,它带有 2 个参数(账号和密码)。

```
<? xml version="1.0" encoding="utf-8"? >
```

```
<soap:Envelope
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
```

```
>
```

```
<soap:Body>
```

```
<CheckUser xmlns="http://tempuri.org/">
```

```
<account>string</account>
```

```

    <password>string</password>
  </CheckUser>
</soap:Body>
</soap:Envelope>

```

2.2 SOAP 请求的接收和处理

SOAP 服务端处理 SOAP 请求的对象是 Web Services。Web Services 接收到 SOAP 客户端发来的 SOAP 请求后解析报文,从 SOAP 请求中提取出请求方法(SOAP method),根据 SOAP 服务端的绑定文件 AuthenticationService.wsdl 找到与 SOAP 请求相应的信息,将其映射为本地认证服务器上的组件调用,然后调用认证服务器上真正的应用程序,执行 SOAP 请求。SOAP 服务端的绑定文件是一个 XML 格式的文件^[4],形式如下:

```

<wsdl:binding name = "AuthenticationServiceSoap" type = "tns:
AuthenticationServiceSoap">
  <soap:binding transport = "http://schemas.xmlsoap.org/soap/
http" />
  <wsdl:operation name = "CheckUser">
    <soap:operation soapAction = "http://tempuri.org/Check
User" style = "document" />
    <wsdl:input>
      <soap:body use = "literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use = "literal" />
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation> ... </wsdl:operation>
</wsdl:binding>

```

绑定文件中的每个 < binding > 标记对应一个 SOAP 请求方法,其属性指明了 SOAP 请求方法,完成该请求的机器名、对象名(progID)及相应的方法(operation)。

2.3 SOAP 响应消息的形成

SOAP 服务端的 Web Services 对象从 SOAP 请求中解析出 SOAP 请求方法 CheckUser,然后从绑定文件中获知 SOAP 请求方法 CheckUser 应该在 URL 为“http://xxx/AuthenticationService.wsdl”的机器上,由 AuthenticationService 类对象的“CheckUser”方法完成。到此为止,SOAP 服务端上的 Web services 对象就可以创建远程对象来连接数据库,并根据参数执行 SOAP 请求。最后将执行结果加上 SOAP Envelope 信息,得到对上述 SOAP 请求的 SOAP 响应。SOAP 响应包括 SOAP 响应头和 SOAP 响应内容:

HTTP/1.1 200 OK

Content-Type: text/xml; charset = "utf-8"

Content-Length: length

<? xml version = "1.0" encoding = "utf-8"? >

<soap:Envelope

xmlns:xsi = http://www.w3.org/2001/XMLSchema-instance

xmlns:xsd = "http://www.w3.org/2001/XMLSchema"

xmlns:soap = "http://schemas.xmlsoap.org/soap/envelope/">

<soap:Body>

< CheckUserResponse xmlns = "http://tempuri.org/">

< CheckUserResult >boolean</ CheckUserResult >

</ CheckUserResponse >

</soap:Body>

</soap:Envelope>

2.4 SOAP 消息的用户响应

SOAP 响应消息被 SOAP 服务端生成后,再转给 SOAP 客户端,SOAP 客户端分析 SOAP 响应并将结果返回给 Web 服务器^[5],Web 服务器获取用户权限信息,根据权限信息加载显示模块,返回给用户。

3 结 论

Web Services 的所有协议都是基于 Web 标准的协议,客户端可以通过 Internet 调用 Web Services,获得和发送基于 XML 的串行格式的数据。即使各应用系统位于不同的位置及平台,都可以实现对 Web Services 的调用,并且可以穿越防火墙,实现身份认证。因此系统具有良好的可扩展性和可集成性,其用户管理也变得简单灵活。

系统在某企业的门户平台上已经得到了应用。通过统一身份认证服务,将各种应用系统进行了无缝集成和统一管理。应用 XML Web Services 和 SOAP,可独立于特定平台,使得用户通过门户平台,方便使用异构数据库、异构平台下的各个系统,从而实现了简单、灵活的数据集成与应用集成。

参考文献:

- [1] 施明辉,孙荣胜.用基于 XML 的 SOAP 机制构建应用系统[J].计算机应用,2002,22(4):80-83.
- [2] 柴晓路,梁宇奇.Web Services 技术、架构和应用[M].北京:电子工业出版社,2003.
- [3] 李安渝.Web Services 技术与实现[M].北京:国防工业出版社,2003:69-70.
- [4] 许 勇.基于 SOAP 异构分布式对象互操作模型设计与实现[D].成都:四川大学,2004.
- [5] Knowles C.用 SOAP Toolkit v 3.0 建立和使用 Web Services [EB/OL]. http://www.dev-club.com/mag/html/vol5/SoapToolkitv3-ASP.htm,2004.