基于 OAUTH2.0 的高校统一认证中心①

俞 晓[1], 左友东[2]

[1](四川师范大学 计算机科学学院, 成都 610001)
[2](四川师范大学 网络与信息化管理处, 成都 610001)

摘 要: 给出了一种高校统一认证中心的设计原理和实现, 可以使高校用户在校内繁多的子系统间实现统一登录认证, 并保持认证数据的一致性, 同时, 由于采用先进并被广泛采用的 OAUTH2.0 协议, 不仅保证了高校内部子系统认证的规范和安全性, 也使外部开放平台账号能够接入认证.

关键词: 统一认证; OAuth2.0; 隐式授权; 数据集成

Unified Certification Center of Universities Based on OAUTH2.0

YU Xiao[1], ZUO You-Dong[2]

[1](College of Computer Scinece, Sichuan Normal University, Chengdu 610101, China)

[2](Network and Information Management Department, Sichuan Normal University, Chengdu 610101, China)

Abstract: This paper puts forward one kind of design principles and implementation of unified certification center of universities. This implementation allows users to achieve a unified login authentication across multiple subsystems and can maintain the consistency of the authentication data. Meanwhile, due to the use of advanced and widely adopted OAUTH2.0 protocol, not only the system specifications and safety certification of university subsystems is guaranteed, but also external open platform account can access authentication.

Key words: unified certification; OAuth2.0; implicit authorization; data integration

随着高校信息化建设的推进, 各部门满足各种业务需求的子系统越来越多, 其中很多子系统需要面向全校教职工和学生甚至校外人员提供个性化服务, 如果不建立统一认证中心, 各子系统只能重复建设数万的用户帐号才能满足提供个性化服务的需求, 同时造成用户维护多个帐号密码的局面, 而这些用户账号通常就是教职工和学生用户, 教职工用户的数据来源是人事部门, 学生用户的来源一般是教务部门, 因此, 只有建立一个统一的认证中心, 才能够保证账户数据的一致性, 从而使系统与系统间的安全的细粒度的业务集成成为可能, 并且通过选择合适的认证协议来接入第三方开放平台账号登录(例如 QQ、腾讯微博等), 而且还可以使各个子系统间实现安全的双向单点登录.

由于互联网技术的迅猛发展、系统之间的相互协作日益增多, 开放与共享数据的需求不断增加, 互联网服务之间的整合已经成为必然趋势, 一个通过自身开放平台来实现数据互通甚至用户共享的时代已经来临[1]. 把网站的服务封装成一系列计算机易识别的数据接口开放出去, 供第三方开发者使用, 这种行为就叫做 Open API, 提供开放 API 的平台本身就被称为开放平台. 随着 OAUTH2.0 的发布以及各大公司的支持, OAUTH 协议成为了目前最主流的开放平台授权方式. 支持 OAUTH2.0 的平台包括 Google API[2]、Windows Live[3]、腾讯[4]、新浪微博[5]、淘宝[6]、百度[7]、人人网[8]、开心网[9]等国内外主流的开放认证平台. 基于此, 本文研究的高校统一认证中心的实现也是基于 OAUTH2.0 协议的.

1 OAuth2.0认证授权技术

1.1 OAuth 2.0 协议

OAuth 2.0 是 OAuth 协议的下一版本, 但不向后兼

① 基金项目:四川省教育厅重点项目(14ZA0029)
收稿时间:2013-12-08;收到修改稿时间:2014-02-20

容 OAuth 1.0. OAuth 2.0 关注客户端开发者的简易性, 同时为 Web 应用, 桌面应用和手机, 和起居室设备提供专门的认证流程. OAuth 的 2.0 授权协议允许第三方应用程序获取机会有限的 HTTP 服务, 无论是以资源所有者的名义通过协调的批准资源所有者和 HTTP 服务互动, 或通过允许第三方应用程序, 以自己的名义获得的访问.

OAuth2.0 通过引入授权层和从资源所有者那里分离客户端的角色来解决传统认证方式的安全问题. 在 OAuth2.0 中, 客户端请求访问被资源所有者控制和资源服务器托管的资源, 并发出了跟资源所有者不同的凭据集合. 客户端获得一个访问令牌, 即一个一个字符串, 表示一个具体范围, 寿命和其他访问属性, 而不是使用资源所有者的凭据来访问受保护资源. 访问令牌被授权服务器批准资源的所有者颁发给第三方客户端. 客户端使用访问令牌访问资源服务器托管的受保护资源.

OAuth2.0 主要的授权方法包括四种:

Authorization Code: 客户端是 web 服务器程序的一部分, 通过 http 请求实现, 是 OAuth 1.0 流程的简化版本;

Implicit Grant: 客户端运行于用户代理内(通常用 JavaScript 等脚本语言在浏览器中实现);

Resource Owner Password Credentials: 这种授权许可的类型需要最终用户和客户端有很强的信任关系, 客户端可以直接使用资源拥有者的私有证书(用户名和密码)用作访问许可来获取 Access Token;

Client Credentials : 客户端使用它的私有证书去获取 Access Token.

1.2 OAuth 2.0 的抽象协议流

OAuth 为客户端提供了一种代表资源拥有者访问受保护资源的方法. 在客户端访问受保护资源之前, 它必须先从资源拥有者获取授权(访问许可), 然后用访问许可交换访问令牌(Access Token, 包含许可的作用域、持续时间和其它属性等信息). 客户端通过向资源服务器出示访问令牌来访问受保护资源.

使用 OAuth2.0 机制, 进行认证授权, 获取访问令牌, 并通过访问令牌来访问受保护资源, 如图 1 所示的抽象流描述的流程包括以下步骤:

(1)客户端向资源所有者请求授权.

(2)客户端获得了资源所有者的授权认可, 此授权认可是四种标准授权认可类型中的一种, 也可以是其中一种的扩展类型.

(3)客户端提供授权认可, 以请求一个访问令牌, 该令牌是可以被授权服务器验证的.

(4)授权服务器验证客户端, 并且核实授权认可, 如果是有效, 就发放访问令牌.

(5) 客户端提供访问令牌给资源服务器验证, 判断此客户端是否可以获取受保护的资源.

(6)资源服务器核实访问令牌, 如果有效则响应请求.

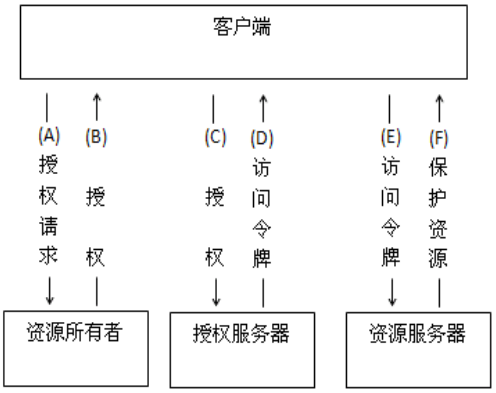

图 1　OAuth2.0 抽象工作流示意图[10]

2　统一认证中心数据集成的实现

统一认证中心需要基础的账号数据, 包括教职工编号和学生学号. 一般而言, 高校的教职工编号来源于人事部门, 本专科生学号来源于教务部门, 研究生学号则来源于研究生管理部门. 为了保证账号的一致性和实时性, 有必要从上述三个部门的管理系统中将账号实时定时同步到统一认证中心数据库中. 为了稳定准确地同步账号数据, 数据抽取/数据转换工具采用了 oracle ODI 中间件技术, ODI(Oracle data integrator)是使用 E-LT 的理念设计出来的数据抽取/数据转换工具. ODI 能够支持异构数据, 包括 SQL SERVER, ORACLE 等主流关系数据库都支持, 是基于元数据管理的, 可以通过触发器方式完成实时的账户数据同步. 统一认证中心数据集成的实现结构图如下.

如图 2 所示, 统一认证中心账号, 主要来源于人事部门、教务部门以及研究生管理部门, 通过数据同步工具 ODI 的接口同步到统一认证中心数据库, ODI 同步用来保证统一认证中心数据库与数据来源数据库的数据保持一致. 统一认证平台为业务系统提供统一

的认证功能服务.

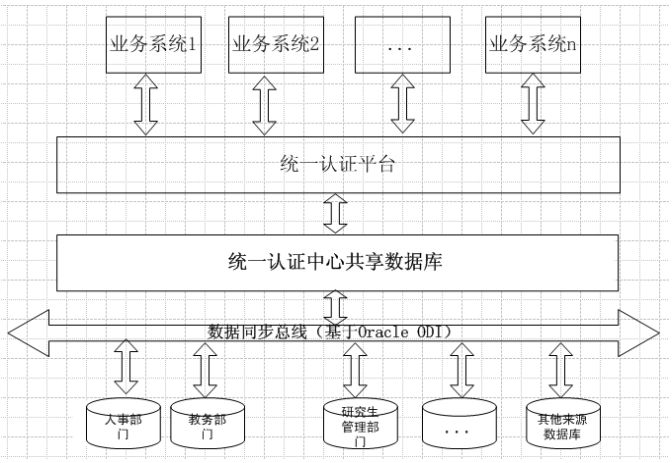

图 2 统一认证中心数据集成示意图

3 统一认证接口的实现

3.1 授权类型

OAuth2.0 定义了四种授予类型: 授权码、资源所有者密码凭据、客户端凭据、隐式授权.

授权码: 客户端是 web 服务器程序的一部分, 通过 http 请求实现, 是 OAuth 1.0 流程的简化版本

资源所有者密码凭据: 资源所有者密码凭据(即用户名和密码)可直接作为一个权限授予获得访问令牌. 凭证只有在资源所有者和客户端之间存在一个高等级的信任的时候使用(如设备的操作系统或一个高特权的应用程序), 或者当其它权限授予类型不可用时(如授权码)凭据才被使用.

客户端凭据: 客户端使用它的私有证书去获取访问令牌.

隐式授权: 是一个简化的授权码流, 为了优化客户端实现在浏览器中使用例如 JAVAScript 的脚本语言. 在隐流中, 客户端直接发出一个访问令牌(作为资源的所有者授权的结果), 而不是给客户端发出一个授权码. 授予类型是隐式的就像没有中间的凭据(如授权码)被发出(后来被用来获得一个访问令牌). 隐式授权提高了某些客户端的响应速度和效率(如实现在浏览器应用程序的客户端), 因为它降低了获得所需的往返访问令牌的数量.

高校目前的业务子系统一般都是 B/S 结构的, 因此采用脚本语言(例如 JAVAScript)注入的方式完成访问令牌的使用可以简化子系统的接入, 因此, 本文统一认证中心的实现采用了隐式授权的方式. 隐式授权原理如图 3 所示:

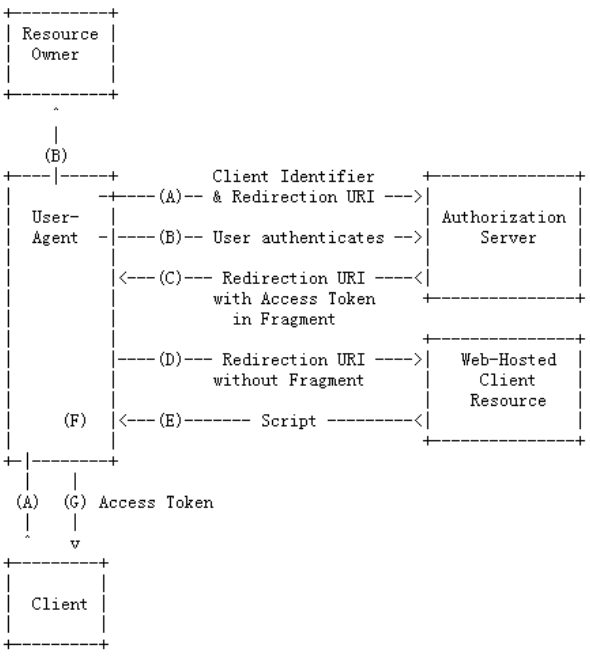

图 3 OAuth2.0 隐式授权示意图[10]

(1)客户端启动流程是从引导资源所有者的用户代理到授权端点开始的. 客户端包括客户端标示符, 请求的范围, 本地状态和一个重定向 URI, 一旦访问被授权, 授权服务器将把重定向 URI 返回用户代理.

(2)授权服务器通过用户代理验证资源所有者并建立资源所有者是否授予或拒绝客户端的访问请求.

(3)假设资源所有者授权访问, 授权服务器通过用户代理把前面提供的重定向 URI 返回给客户端. 重定向 URI 包含在访问令牌的 URI 片段里.

(4)用户代理通过 web 托管客户端发送重定向请求. 客户端在本地保留信息片段.

(5)Web 托管客户端资源返回一个 web 页面(通常是一个嵌入了脚本的 HTML 文档), 这个 web 页面能完全访问保留在用户代理信息片段中的重定向 URI, 并且提取片段中的访问令牌(和其它参数)

(6)用户代理在本地执行 web 托管客户端资源的脚本, 提取访问令牌.

(7)用户代理把访问临牌传给客户端.

3.2 数据结构设计

OAuth2.0 机制的核心工作流程应当存储在认证中心数据库服务器中. 每一个接入应用的子系统需要在数据库中注册如下结构:

字段	说明
AppId	接口编号, 子系统接入的唯一编号
AppName	接口名称, 子系统接口的名称
AppPwd	接口密钥, 子系统与认证中心约定的秘钥, 用来给账号返回信息加 MD5 摘要.
AppUri	直达入口, 是子系统申请接入时在统一认证中心保存的一个 http 请求处理模块的网址, 加上转向参数后形成多个细粒度的业务网址.

服务器账户的结构如下:

字段	说明
user_id	用户编号, 不为空, 通常是指教师的职工号、学生的学号、QQ 登录的 OpenId、新浪微博的 OpenId、或其他格式的帐号.
user_name	用户名称, 不为空, 通常是指教师的姓名、学生的姓名、QQ 昵称、新浪微博昵称等.
user_gid	用户的院系所号.
user_lbm	用户类别码
user_zwm	教职工用户的职位编码
user_zcm	教职工用户的职称编码
user_nj	学生用户的年级
user_bh	学生用户的班号
user_role_list	用户的子系统角色列表, 代表用户在本子系统的拥有的功能权限
md5	不为空, 由 user_id + user_gid + user_lbm + user_zwm + user_zcm + user_nj + user_bh + user_role_list + AppPwd 组成的原始字符串的 md5 摘要, 其中 AppPwd 是子系统密钥, 用于对关键数据进行校验以提高关键应用的安全性.

3.3 认证接口实现

(1)获取 access_token 访问令牌

认证接口参数设计如下:

response_type: 必须, 设为 token;

client_id: 必须, 申请接入时分配的 AppId;

redirect_uri: 可选, 重定向路径, 包含和以申请接入时设置的直达入口 AppUri 为开始部分.

state: 可选, 是由子系统产生的状态值(比如随机数、SessionID 等), 会原样返回, 用于防止 CSRF(cross-site request forgery)攻击.

outer_type: 可选, 用于直接启动某种类型的开放帐号登录, 如: 腾讯 QQ、新浪微博账号等. 这些第三方账号可以用在需要保存个人信息的场合, 但又可省去注册的步骤, 例如高校的人事招聘系统对社会招聘的时候, 社会用户可以通过这种方式登录并填报相关信息, 而无需在系统中注册新的账户.

如果 redirect_uri 为 http://x.y.z/test.htm, 则用户登录后会重定向结果为:

http://x.y.z/test.htm#access_token=xxx&token_type=bearer&expires_in=3600.

子系统用 javascript 脚本从 window.location.hash 提取 access_token. 而子系统需要提供相应的脚本支持. 为了方便子系统的集成, 提取访问令牌的 javascript 脚本采用脚本注入的方式, 即子系统包含如下代码片段来注入脚本:

<script type="text/javascript" src="http://注入的脚本路径/?接口编号=接口编号的值"></script>.

(2) 利用访问令牌(access_token)获取授权信息:

请求参数说明:

access_token: 访问令牌字符串;

client_id: 申请接入时分配的 AppId;

callback: javascript 回调方法.

响应数据说明:

出现任何错误时: {null}.

未出现任何错误时: 返回完整的账户数据.

如果存在 callback 参数时, 则返回并调用回调方法.

3 结语

高校信息化建设有一定复杂性, 尤其是业务系统繁多, 建设基于 OAuth2.0 统一的认证中心不仅可以实现统一认证、双向单点登录, 而且可以集成开放平台账号登录, 从而实现校内、校外账号的统一认证, 高校内部的业务子系统均可以安全接入认证中心, 从而为统一授权和细粒度业务集成提供了支撑平台. 统一授权和细粒度的业务集成可以为用户提供更好的用户体验和功能服务, 从而使用户不必关心业务办理在哪个系统中, 从用户角度来看, 就是一个完整的大系统, 这方面的设计和实现值得进一步研究.

参考文献

1 时子庆,刘金兰,谭晓华.基于 OAuth2.0 的认证授权技术. 计算机系统应用,2012,21(3):260–261.
2 Making auth easier: OAuth 2.0 for Google APIs. http://googlecode.blogspot.com/2011/03/making-auth-easier-oauth-20-for-google.html.2011.3.
3 OAuth2.0. http://msdn.microsoft.com/en-us/library/hh243647.aspx. 2013.11.
4 OAuth2.0 简介.http://wiki.open.qq.com/wiki/website/OAuth2.0 简介.2013.12.
5 授权机制说明.http://open.weibo.com/wiki/授权机制说明.2013.9
6 淘宝 OAuth2.0 服务.http://open.taobao.com/doc/detail.htm?spm=0.0.0.0.okHQ8S&id=118.2013.12
7 OAuth 介绍.http://developer.baidu.com/wiki/index.php?title=docs/oauth. 2013.12
8 人人网开放平台 OAuth2.0 上线.http://dev.renren.com/blog/193. 2011.1
9 OAuth2.0 文档.http://wiki.open.kaixin001.com/index.php?id=OAuth2.0 文档. 2013.12
10 The OAuth 2.0 Authorization Framework, http://tools.ietf.org/html/rfc674.2012.