Software Engineering and Applications 软件工程与应用, 2014, 3, 152-160
Published Online December 2014 in Hans. http://www.hanspub.org/journal/sea
http://dx.doi.org/10.12677/sea.2014.36018

Hans汉斯

The Design and Research of Single Sign-On Model Based on REST in the Cloud Computing Integrated Information Service Platform

Jun Xing, Qibo Sun, Guangwei Zhang

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing
Email: helloxingjun@gmail.com

Received: Oct. 23rd, 2014; revised: Nov. 26th, 2014; accepted: Dec. 4th, 2014

Abstract

Cloud computing technology makes the Internet applications development easier. Cloud computing combining information service platform becomes modern information services infrastructure. Application services based on cloud computing integrated information service platform become more diverse. These applications require the unified identity authentication and access control management mechanisms to improve the ease and the security of the application development. Single sign-on system is an indispensable function module to implement this mechanism. This paper describes an efficient and scalable single sign-on model based on REST protocol. This paper also focuses on solving the issues of token distribution and maintenance in the system. In the addition, it discusses the main directions of security and performance optimization in centralized single sign-on system.

Keywords

Authentication, Single Sign-On (SSO), Token, REST Protocol, Optimization

云计算综合信息服务平台下基于REST的单点登录模型设计与研究

邢　军，孙其博，张光卫

北京邮电大学网络与交换国家重点实验室，北京
Email: helloxingjun@gmail.com

收稿日期：2014年10月23日；修回日期：2014年11月26日；录用日期：2014年12月4日

摘　要

云计算技术的发展，使得互联网应用系统的开发过程日趋平台化。云计算结合信息服务平台组成了现代信息服务基础设施。基于云计算综合信息服务平台的应用服务更加丰富多样，这些应用都需要统一的身份认证和访问控制管理机制，以提高应用开发的便捷性和安全性。单点登录系统就是实现这一机制不可缺少的功能模块。本文通过介绍一种基于REST协议并且具有高效性、扩展性的单点登录模型，着重解决系统设计与研究过程中的票据分发和维护等关键问题，说明系统实现过程中的流程规范，并进一步讨论集中式单点登录系统中的安全及性能优化的主要方向。

关键词

身份认证，单点登录，认证票据，REST协议，优化

1. 引言

身份认证是网络通信中建立可信安全通道的重要过程，是安全信息系统的“门禁”模块[1]。大多数互联网应用的软件结构中离不开身份认证模块的设计和实现。云计算是一种模型，它可以实现随时随地，便捷地，随需应变地从可配置计算资源共享池中获取所需的资源(例如，网络、服务器、存储、应用、及服务)，资源能够快速供应并释放，使管理资源的工作量和与服务提供商的交互减小到最低限度[2]。云计算是继分布式计算、网格计算对等计算之后的一种新型计算模式，它以资源租用、应用托管、服务外包为核心[3]。云计算结合已有的综合信息服务资源(北斗导航系统，车联网服务，道路信息监控服务等等)共同构成现代化云计算综合信息服务平台。基于该平台的软件开发过程进一步被简化。大量依托于平台服务的软件被开发出来。这些软件几乎都需要实现身份认证与访问控制等基本安全控制模块的功能，重复地开发这些模块将导致大量冗余的工作，并且使得安全质量的控制更加困难。单点登录(Single Sign-On SSO)提供一种机制，让不同的应用系统迅速获得统一的认证功能，实现全局、安全的软件环境[4]。该机制使得用户在完成一次登录后，能够在多个应用系统之间共享身份认证结果，同样在一次登出操作后，即完成对所有相应的应用系统的登出功能。云计算平台下应用开发更加依赖于平台服务，单点登录系统可以作为统一的安全认证模块，整合于云计算平台基础服务组件中。

单点登录系统已经在传统的互联网应用以及企业应用中有一定的应用。目前也有一些相对规范的解决方案，如 Microsoft 公司的 Passport，IBM 的 WebSphere Portal Server 等成熟的商业解决方案，以及如 CAS 协议这样的开源的 SSO 系统实现方案。这些实现方案所针对的侧重点方面有所不同，并且所完成的功能早已超出了单纯的单点登录功能范畴。并且系统的模块实现复杂，没有更多地考虑到云计算平台开发的互联网特性，而是更加适合传统的企业级应用，更加关注安全性，稳定性等。在实际互联网应用中往往存在复杂度高，不够灵活，扩展新差，分布式并发性能低等缺点。本文将讨论实现一种相对简单，灵活的基于 REST 协议的 SSO 实现方案，并且讨论在云计算互联网环境下性能和安全方面的优化策略。着重介绍系统实现的主要结构，流程规范，以及 cookie 跨域等核心问题的解决方案。

2. 基本概念

2.1. 系统角色

单点登录系统主要包含三种不同的角色，分别是用户、应用服务和认证服务。三种角色构成了单点登录过程中需要控制的实际行为对象。单点登录的过程实际就是用户在访问一组应用服务过程中，需要由认证服务校验其身份真实性，并且建立用户与应用服务互信的过程。理解三种角色各自的主要功能是设计并实现单点登录系统的前提。三种角色主要的关系与如图 1。

2.1.1. 用户

用户是信息资源或者应用服务的拥有者，是应用访问的主体。对于 Web 应用，用户的角色通常由浏览器代表。用户的目的是成功进入已经注册的应用系统，并且成功获得个人在应用系统的应有的信息资源或者服务。为了成功证明自身的身份信息合法、真实，用户需要提供其真实、有效的身份标识。常用的用户身份认证标识包括：用户口令、智能卡标识、用户生物特征等。其中后两项认证标识需要终端设备辅助完成，通常用于企业级应用、金融业务应用等安全性要求较高的场合，目前互联网应用的认证身份标识主要还是用户口令/密码。

2.1.2. 应用服务

应用服务为用户提供实际有用的信息资源或服务，是应用访问过程中的客体。在身份认证过程中，应用服务在为用户提供信息资源或服务之前，必须验证用户的身份信息的真实性，以便确定用户对哪些资源或者服务具有操作权限。可以说身份认证不但是完成用户对应用的使用身份的验证，也是在为用户的授权控制做前提准备。在单点登录适用的模型中，应用服务可以是多种多样的，当用户完成对其中一个应用的验证工作或者在同一的入口完成验证工作后，他就能对所有注册应用进行正常的访问。

2.1.3. 认证服务

认证服务是专门为用户访问一系列应用进行身份认证的服务模块，可以作为独立的认证服务器部署，也可以和应用服务一起部署。认证服务应当完成对用户身份信息的维护，用户身份的验证，票据的生成和维护等工作。云计算平台下，更加强调认证系统的服务功能，应当给应用开发者提供良好的业务流程、服务接口以及数据结构规范，确保服务认证过程的正确使用和安全可靠。

2.2. 认证票据

Token 即认证票据，是应用系统用户在登录系统后，由认证服务模块统一分配的，代表用户身份的唯一标识。维护该票据是为了在 SSO 系统中表示用户在系统全局的登录状态，票据的状态应当在各个应用系统中同步维护，作为用户是否为登录活跃状态的标识。票据的维护主要包括票据生成，票据分发，票据删除三个主要过程。

2.2.1. 票据生成

全局 Token 的生成是在认证服务器完成用户身份验证之后，验证工作集中由认证服务器完成，全局 Token 可以采用单向哈希的方法生成唯一的票据字符串。本系统为了字典攻击等针对单向 Hash 等攻击方法，在 Hash Key 中除了用户名之外，加入当前时间戳和随机字符，确保安全性。Token 生成规则可表述如下：

$$\mathrm{Key} = \mathrm{username} + \mathrm{timestamp} + \mathrm{random}() \tag{1}$$

$$\mathrm{Token} = \mathrm{Hash}(\mathrm{Key}) \tag{2}$$

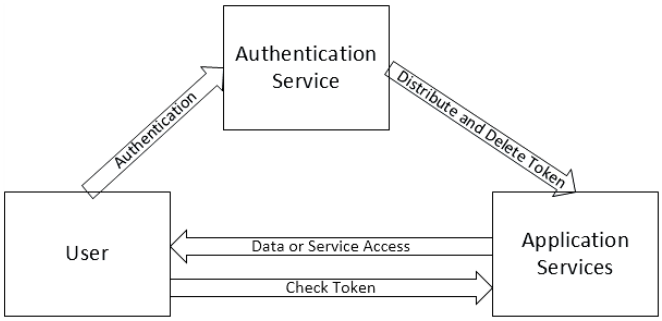

Figure 1. Role of Single Sign-On system
图 1. 单点登录系统角色

2.2.2. 票据共享

认证服务生成票据后的重要一步是如何与所有的应用服务共享票据，实现票据的共享是单点登录应用端各自验证的前提。目前票据在客户端存储的主要方式是对应域名的 cookie，因此需要解决的问题即在不同的域名下共享 cookie 内容。由于一个 cookie 只能由一个域名访问的特性，如果要为每一个应用对应的登录用户保持登录票据信息，必须为每一个应用服务对应的域初始化含有相同票据的 cookie。这个 cookie 的写入过程交由认证服务完成，在用户成功验证后，认证服务生成票据信息，并通过异步数据传输，初始化用户所有可访问应用的 cookie。但由于不同域之间不能访问对方 cookie 信息，因此在异步方式传输 cookie 时可以采用 jsonp 数据格式。jsonp 是使用 javascript 脚本允许跨域的特性，利用 js 函数回调的方式传送 cookie 数据。

票据内容共享如图 2 所示。其中，认证服务部分需要植入异步 cookie 数据发送代码，代替人为的浏览器访问去初始化每一个用户可访问应用的 cookie 内容信息，各个应用部分需要开放 cookie 初始化接口服务。为了能够成功设置不同域应用的 cookie 信息，服务端脚本采用 Ajax 方式传动 jsonp 数据格式，设置各应用域 cookie 信息。

2.2.3. 票据删除

单点登录的退出部分需要完成所有票据的删除工作。当用户从一个应用退出时，调用认证服务模块删除票据接口服务，由认证服务接口模块调用每一个应用的票据删除服务，删除所有应用后台保存的用户票据信息，确保用户 cookie 中的票据被删除或者自动失效。

2.3. REST 协议

表述性状态转移(Representational State Transfer [5]，简称 REST)是 Roy Fielding 博士在 2000 年他的博士论文中提出来的一种软件架构风格。它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。表述性状态转移是一组架构约束条件和原则。满足这些约束条件和原则的应用程序或设计就是 RESTFUL。与 RPC 风格的 Web 服务架构相比，采用 REST 的 Web 服务架构在扩展性、安全性、数据耦合性等方面存在着优势[6]。以 REST 协议定义单点登录系统中的服务接口，可以保证相应服务的流程规范并且安全可靠。如票据的分享和删除，可以定义的 REST 服务接口列表如表 1。

一组规范定义的 REST 服务接口，包括接口地址，接口参数，以及服务接口使用的 HTTP 协议方法。

3. 解决方案

云计算综合信息服务平台，基于多种综合信息基础服务，包括北斗导航信息服务，车联网信息服务等等。在云计算环境下，为众多的应用服务提供可用的基础信息资源和服务，目前有诸如中石油运油车

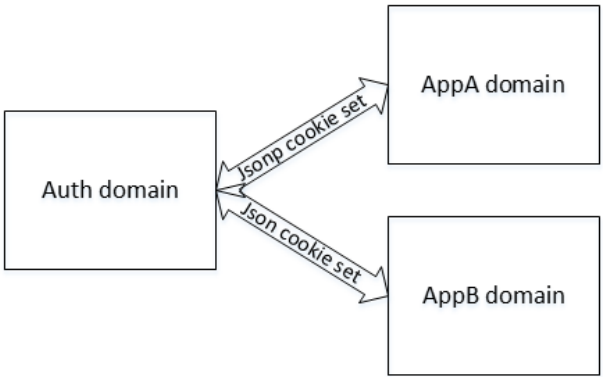

Figure 2. Sharing token in Single Sign-On system
图 2. 单点登录模型 Token 共享

Table 1. RESTFUL interface mode
表 1. RESTFUL 的服务接口方式

RESTFUL 服务接口	参数	接口方法
URLA/setToken/	{TokenInfo}	POST
URLA/delToken/	{TokenID}	DELETE
URLA/putToken/	{TokenID, TokenInfo }	PUT

监控，森林防火等多种应用基于该平台开发。统一认证和访问控制子系统作为重要的安全服务模块，集成于综合信息服务平台提供认证和授权服务。其中统一认证部分主要采用的就是 SSO 系统结构，为分布在各地，各企业的企业应用提供企业内部应用单点登录的功能。

统一认证模块主要包含了认证服务端和应用服务端认证组件两个独立的部分。应用服务端认证组件，是在应用开发过程中放入各个应用软件模块中的验证组件，主要提供每个应用独立的登入状态的检测，配合服务端设置登录票据等等功能。认证服务主要提供用户身份信息的维护，应用信息的维护，单点登录入口，登录票据的维护等功能。统一认证系统部署后的系统结构如图 3。

3.1. 认证流程

实现 Web 环境下单点登录，主要包括以下几个步骤，如图 4 所示为用户初次登录系统，成功验证的时序图。1) 用户浏览器访问应用服务，此时校验用户的 cookie 是否存在以及 cookie 中的登录票据是否有效。如果有效进入正常访问阶段。否则重定向到统一认证服务模块登录页面。2) 用户在登录页面输入用户名和密码等身份信息，由认证服务验证其有效性，确认正确则生成 Token，并将 Token 异步分享到用户对应的应用，最后重定向到用户访问的应用。3) 用户应用再次校验用户浏览器的 cookie 中的 Token 有效性，正确识别则允许用户正常访问。

3.2. 系统模块

3.2.1. 应用服务端

首先，应用服务端是应用的“门户”，要过滤用户的访问，判断用户登录票据是否存在或者有效。如果发现用户票据不存在或者已经失效，则将用户重定向到认证服务器的登录页面，否则允许用户的当次的访问行为。

其次，应用服务端需要在验证成功后初始化自身的票据，与认证服务器及其他应用保持一致，并且

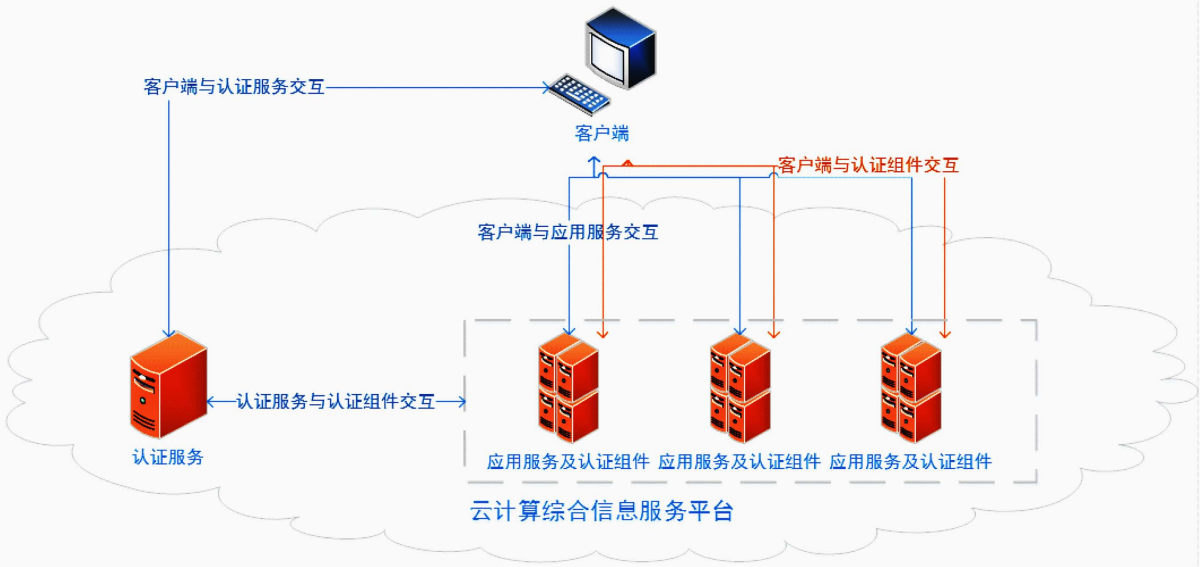

Figure 3. Deployment of Single Sign-On system
图 3. 单点登录系统部署结构

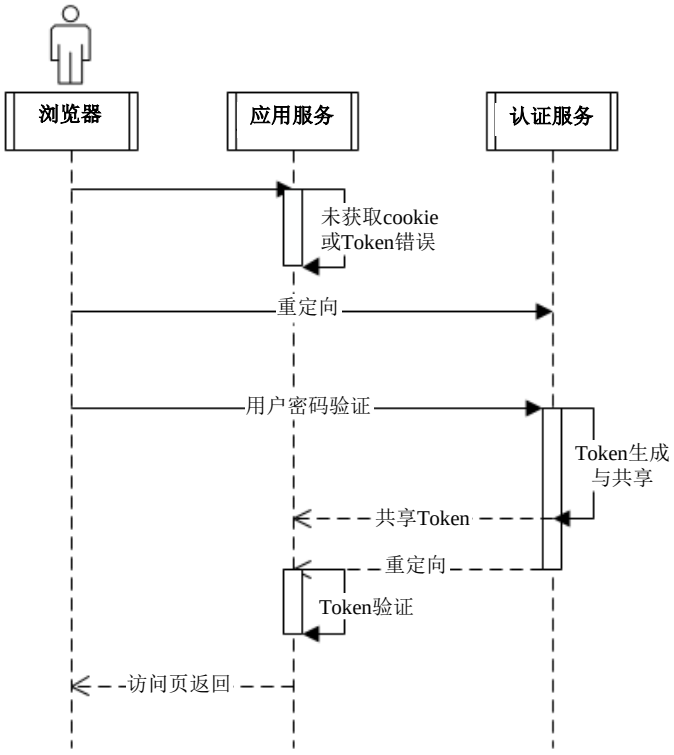

Figure 4. Sequence of Single Sign-On system
图 4. 单点登录模型时序图

维护该票据在本应用的状态，如存放在自身的会话 session 中，有效提高访问的效率，而不是每次都需要在认证服务器中获取。

最后，应用服务端应当提供单点退出的功能，当用户在当前应用发出退出请求，当前应用应当通知认证服务器，删除全局维护的票据，或者使当前使用的票据失效。

除此之外，应用服务端模块应当具有较好的配置性。在简单的配置认证服务器的地址，域名，特定的认证服务地址后，应当可以嵌入应用端的开发环境直接使用。

3.2.2. 认证服务端

认证服务端提供用户登录的入口，应用端发现用户处于未登录状态，首先将用户重定向到该登录页面，并携带登录应用的返回地址，以便用户成功登录后返回应用正常使用。在用户成功验证用户名，密码等信息后，认证服务端按照规则生成用户的票据，并分发到每个应用共享用户票据。由于每个用户可以访问的应用个数不太一致，因此每个用户都需要生成其应用的可访问应用列表，并提供其 Token 分发地址。具体格式如表 2。

当应用单点退出时，认证服务在得到用户的单点退出请求时，同样需要根据列表的信息，删除所有应用的 Token 信息。除此之外，认证服务器需要提供一定的服务提供该应用管理和用户注册的功能，以便生成用户与应用服务之间的对应关系，确定哪些用户可以正常访问哪些应用服务。这样，所有的认证操作皆集中于认证服务器进行，采用这种模型，有利于保护用户的密码，减少密码泄露的可能性[7]。

4. 对比 REST 与 RPC 风格应用于单点登录系统

与传统 RPC 风格的 Web 服务架构相比，单点登录服务接口采用 REST 的 Web 服务架构在扩展性、安全性、数据耦合性、交互性能等方面存在着优势。云计算综合信息服务平台下应用呈现多样性，数量大，用户数量大等特点，REST 服务接口的实现方案，恰好可以适应这些新的需求。高扩展性的特点可以使得使用认证服务应用可以基于多种不同的环境平台进行开发(如 JAVA, .NET 等)，只需要关注通信数据本身即可。安全性方面确保认证服务和应用服务开放的服务接口是按需提供的，可以屏蔽未开放的资源与服务，确保这部分资源的安全性。较低的数据耦合性使得单点登录服务端与应用端的数据交互更加关注数据本身。而不需要考虑具体的服务接口实现方式。同时能够较好地克服登录服务在使用分布式解决方按时数据耦合的问题。性能优势可以确保在使用集中式的单点登录方案时，认证服务端等响应性能良好。

4.1. 扩展性

在 RPC 风格的架构中，如 SOA，服务由细粒度的自定义操作组成不同的服务具有不同的专有接口。每个接口具有自己的语义和操作参数，服务的接口契约对服务的定义非常关键。客户端要想与 SOA 服务正确的互操作，它必须理解每个服务接口契约的语义。由于操作的数量是没有限制的，在 Web 这个开放的分布式环境中会产生紧密耦合和接口复杂等问题，难以达到 Web 级规模的可扩展性要求。在接口问题上 REST 采取了与 RPC 不同的方式，REST 架构为所有资源提供统一的操作接口。REST 主张使用 HTTP 标准的 GET、POST、PUT 以及 DELETE 来进行请求和响应。GET 用于获取资源的表示，POST 创建一个资源，PUT 修改资源，DELETE 删除一个资源，每个操作都具有明晰的语义。它使得客户端和服务器端的实现解耦，可以独立扩展。

4.2. 安全性

SOAP 是目前 RPC 风格的 Web 服务中使用最广泛的协议之一。客户端与服务器通过交换 SOAP 数据包来实现交互。 SOAP 数据包通常被放入 HTTP 文档中，利用 HTTP 的 POST 方法来传交换 SOAP 数据包来实现递，借助 HTTP 协议的 80 端口可以很方便的穿越防火墙。但这种将 HTTP 当作管道的方式存在安全隐患，因为每个请求的真实意图包装在 SOAP 文档中，当 SOAP 文档解析时才被服务器了解，如果 SOAP 携带的是一些危险的请求(如非法删除、恶意修改)，那么这些请求将不能被防火墙所阻挡。为了消

Table 2. User service interface list
表 2. 用户服务接口列表

应用名称	Token 分发	Token 删除	Token 更改
AppA	URLA/setToken	URLA/delToken	URLA/putToken
AppB	URLB/setToken	URLB/delToken	URLA/putToken

除这些潜在的危险，防火墙不得不执行额外的协议过滤。相对于 RPC，REST 的安全模型更加简单有效。REST 中所有事物都被抽象成了资源，每个资源都有唯一的 URI，有了这两个约束，如果要隐藏某个资源，不发布它的 URI 即可。REST 使用 HTTP 的统一接口，每个接口的语义清晰，且不含有 SOAP 式的嵌套。对资源的 4 种操作分别设置权限，就可以形成不同的安全策略。

4.3. 数据耦合性

RPC 风格的架构中，在服务契约中定义的不仅仅是接口，还包括交互所涉及的数据格式。如 SOA 架构在 WSDL 中普遍使 XML Schema 来定义数据，使客户端与服务器间的耦合度大大增加。而 REST 要求消息是自描述的，表示中的数据格式是可以变化的，它取决于客户端与服务器内容协商的结果。不同的消息可以通过 HTTP 的 content-type 和 accept 头中指定不同的格式-前者，后者在请求中指明调用者期望在响应中接受的数据格式。REST 对数指明消息的数据负载格式据格式的处理方式减轻了客户端与服务器间的数据耦合。

4.4. 交互性能

首先，REST 性能优势来自于它与生俱来的简单性。REST 建立在已经广泛使用的 Web 标准之上，不需要额外的附加标准，从而避免了对大型专用平台的依赖，减少对系统资源的占用。比如在数据传输方面，REST 的交互直接使用 HTTP 协议，客户端和服务器都免除了解析和封装 SOAP 数据包的性能消，也降低了传输的负载。其次，REST 主张客户端或媒介缓存那些服务器标识为允许缓存的应答来消除一些不必要的交互，以便提高性能。

5. 安全及性能优化方向

在实现基本的单点登录特定功能的前提下，需要着重考虑单点登录过程中的安全问题和性能问题。安全性方面，由于用户认证状态的维护主要依赖登录票据信息，因此需要考虑登录票据的保密性。除此之外，由于采用集中式的单点登录结构，所有票据的产生和维护工作需要交由认证服务器进行。因此，认证服务器中数据的安全性，认证服务器的可靠性都需要加强。集中式的单点登录结构，需要考虑系统的性能问题，在一次登录后，应该尽可能地将验证功能交由应用端认证组件完成，除此之外，对于频繁访问的用户信息的存放也需要优化。

5.1. 数据加密

数据加密主要应用与账号、密码、认证票据等信息，防止用户密码，登录票据等重要数据泄露。用户密码从一开始的注册生成，到用户验证阶段均采用单向加密策略，如 MD5 加密。在加密的过程中适当加入一定的冗余信息，提高加密的随机性，可以有效防止字典攻击等。登录票据在认证服务端和应用端认证组件之间传递，可以采用对称加密。确保登录票据的安全性。

5.2. 数据备份

由于用户身份信息，应用信息，用户应用访问信息等集中存放在认证服务端。认证服务端的数据服

务需要加入数据备份的功能，防止用户身份信息，应用信息等丢失损毁。必要时这些备份信息需要异地存放，防止同地物理伤害。

5.3. 日志审计与流量检测

认证服务向所有应用，用户开放，为了防止恶意的认证服务访问，需要记录用户的登录信息，检测用户等登录频次，访问流量信息等，防止恶意的 DDOS 攻击等。控制用户等登录频次和应用的请求频次在合理的范围内。

5.4. 内存数据库应用

由于采用集中式的认证管理，常用的用户信息可以存放在 cache 中，合理地运用内存数据库，如 Redis，Memcache 等将活跃用户等身份信息，应用访问列表等信息，使用内存数据库存放，可以有效提高认证服务器存取数据的速度。

5.5. 维护本地 Token

应用服务在共享 Token 后，应当可以将各自的 Token 信息维护在每个应用服务的会话 session 中。即确保用户在登录后，每个应用维护本地的 Token，而不必每次用户访问都需要去认证服务获取验证。简言之，应用服务只有在用户登录和用户退出时，才和认证服务交换信息，在用户正常访问时，用户的 Token 校验工作全部在应用服务端进行。

6. 结语

本文讨论的单点登录系统，是基于 J2EE 的 Web 实现，主要针对 Web App 的开发设计。系统主要分为客户端和服务端两个模块，具有较好的嵌入性，在正确配置的情况下，客户端可以直接植入应用使用，只需要配置简单的参数。服务端的构建需要一定的参数和数据库服务配置支持。系统的核心基于 Token 的维护和共享，主要采用 cookie 跨域的方式实现。客户端和认证服务的主要服务接口，采用 REST 协议实现。最后从安全和性能两个方面讨论了系统的优化方法和思路。

参考文献 (References)

[1] 郭代飞, 杨义先 (2003) 数字身份认证技术的现状与发展. *计算机安全*. **7**, 1-4

[2] Mell, P. and Grance, T., 美国国家标准与技术研究所(NIST) (2011) NIST 关于云计算的定义.

[3] 林闯, 苏文博 (2013) 云计算安全：架构、机制与模型评价. *计算机学报*, **9**, 1765-1784

[4] 刘润达, 诸云强 (2007) 一种简单跨域单点登录系统的实现. *计算机应用*, **2**, 288-291

[5] Fielding, R.T. (2000) Architectural styles and the design of network-based software architectures. PhD Thesis,

[6] 冯新扬, 沈建京 (2010) REST 和 RPC：两种 Web 服务架构风格比较分析. *小型微型计算机系统*, **7**, 1393-1395

[7] 钟林栖 (2006) 基于 CAS 协议的单点登录系统的研究. 硕士论文, 四川大学, 四川.