# SSO 单点登陆设计

## 概念

SSO： Single Sign On，官方的概念：web系统由单系统发展成多系统组成的应用群，复杂性应该由系统内部承担，而不是用户。无论web系统内部多么复杂，对用户而言，都是一个统一的整体，也就是说，用户访问web系统的整个应用群与访问单个系统一样，登录/注销只要一次就够了。

简言之，系统内部通过某种技术实现用户统一登录和注销，所以单点登录技术一定要包括两部分：登录、注销。

## 设计原理

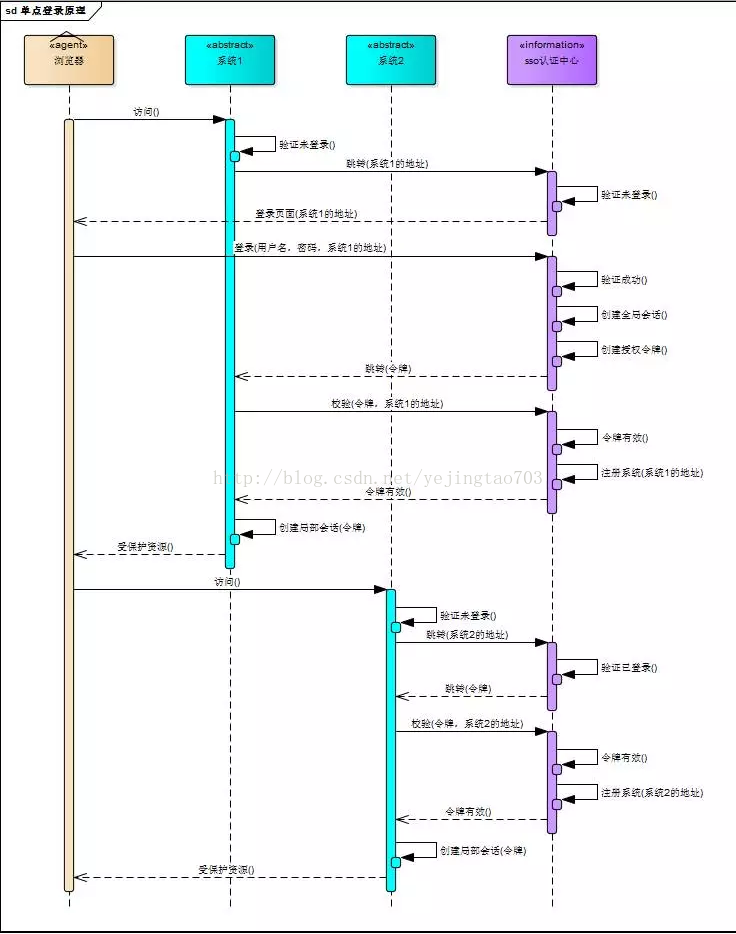
1为什么要用单点登录？因为Cookie不能跨域。

2如何实现单点登录？

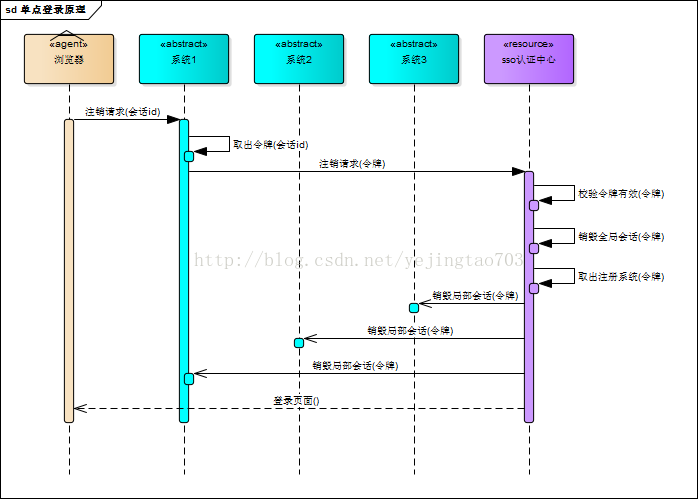
建立权限认证中心来处理登录和注销的问题，真正提供服务的应用服务端通过Filter将鉴权任务重定向给认证中心。

原理图：

登录：



注销：



首先看得出分为2个角色，一个是应用服务端，也就是认证客户端；另一个是认证服务端。根据上图来分析2个角色应该具备的功能。

认证客户端应该具备的能力：

1必须以Filter或者插件等形式提供，方便系统接入SSO。

2未登陆的用户重定向到SSO认证中心

3接收SSO发来的令牌并将该令牌发回给SSO做令牌认证

4处理令牌认证结果并创建局部会话

5拦截用户注销请求并重定向到SSO

6处理SSO发来的注销会话请求

认证服务端应该具备的能力：

0独立的web服务

1提供登陆页面，和对用户的校验

2创建全局会话、提供token

3校验token有效性并维护client端地址

4处理注销请求、销毁全局会话、并通知维护的client端地址

5判断用户有没有登录

## 关键技术

解决cookie不能跨域问题。

建立认证服务中心，使客户端和服务端进行通讯，在客户端传入服务端重定向的地址，认证中心认证成功后返回。

使用验证token

验证 token 时可以考虑使用账户中心提供高性能的验证接口，子站进行调用。

## 测试

启动测试：

app1，植入sso-client插件，域名使用test1.com,服务地址test1.com:9001

app3，植入sso-client插件，域名使用test3.com,服务地址test3.com:9003

Sso-Server，独立部署，域名使用peer3,服务地址127.0.0.1:8080

都启动之后，开始测试：

步骤一：浏览器访问app1网址：http:test1.com:9001/hello被重定向http://127.0.0.1:8080/loginPage?originalUrl=http://test1.com:9001/hello地址进行登录

步骤二： SSO-Server的登陆页面上用admin账户登陆，登陆成功后3秒钟后页面自动跳转到app1的指定的请求页面。

步骤三：在app3登陆主页 <http://test3.com:9003/hello,会发现直接跳转到app3> 指定跳转的页面。

步骤四：在SSO-Server端注销该账号

步骤五：注销后app1和app3又要重新登录了。

整个过程中可以配合redisclient观察redis数据帮助自己理解。