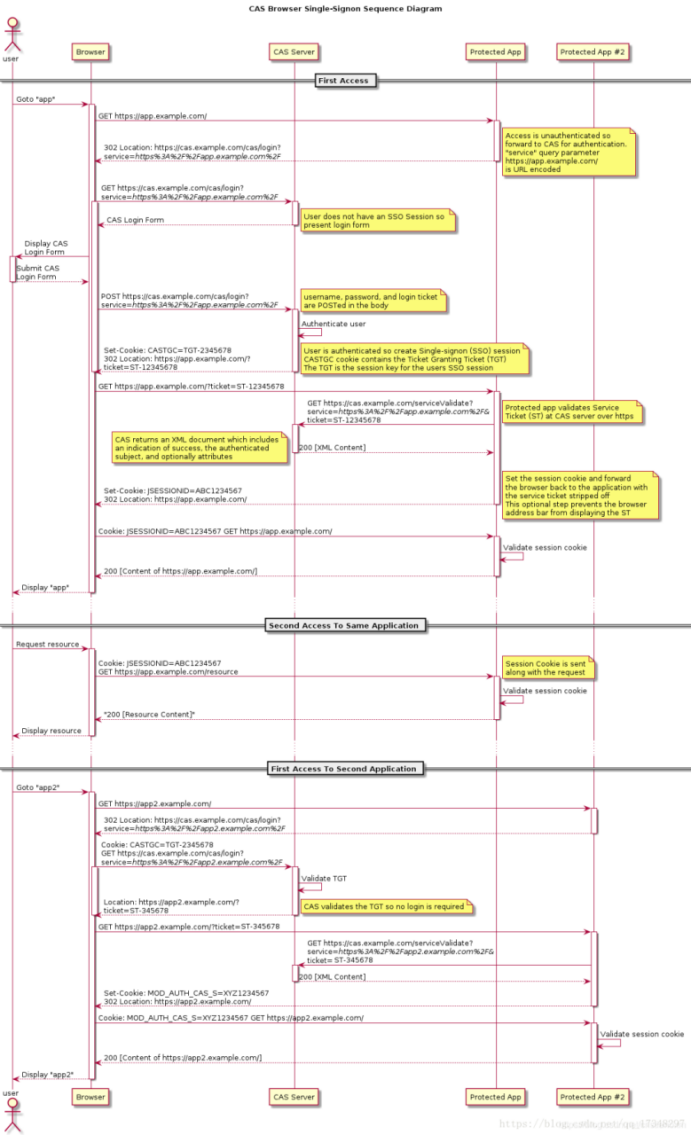
**SSO单点登录和OAuth2.0的区别和理解**

**——201 范晓蝶**

SSO是Single Sign On的缩写，OAuth是Open Authority的缩写，这两者都是使用令牌的方式来代替用户密码访问应用。流程上来说他们非常相似，但概念上又十分不同。SSO将登陆认证和业务系统分离，使用独立的登陆中心，实现了在登陆中心登陆后，所有相关的业务系统都能免登录访问资源。OAuth2。0的原理可能比较陌生，但是平时使用会很多，比如访问某网站想留言又不想注册时使用了微信授权。以上两者的账号密码是存放在登陆中心或微信服务器中的，这就是所谓的使用令牌代替账号密码访问应用。

SSO的实现有很多框架，比如CAS框架，以下是CAS框架的官方流程图。

（图片在下一页）



上面的流程大概为：

* 用户输入网址进入业务系统Protected App，系统发现用户未登录，将用户重定向到单点登录系统CAS Server，并带上自身地址service参数
* 用户浏览器重定向到单点登录系统，系统检查该用户是否登录，这是SSO(这里是CAS)系统的第一个窗口，该接口如果用户未登录，则将用户重定向到登录界面，如果已登录，则设置全局session，并重定向到业务系统
* 用户填写密码后提交登录，此时的登录界面是SSO系统提供的，只有SSO系统保存了用户的密码
* SSO系统验证密码是否正确，若正确则重定向到业务系统，并带上SSO系统的签发的ticket
* 浏览器重定向到业务系统的登录接口，这个登录接口是不需要密码的，而是带上SSO的ticket，业务系统拿着ticket请求SSO系统，获取用户信息，并设置局部session，表示登录成功返回给浏览器sessionId(tomcat中叫JSESSIONID)
* 之后所有的交互用sessionId与业务系统交互即可

最常见的例子：我们打开淘宝APP，首页就会有天猫、聚划算等服务的链接，当你点击以后就直接跳过去了，并没有让你再登录一次

OAuth2.0的流程跟SSO差不多，在OAuth2中有授权服务器、资源服务器、客户端这样几个角色，当我们用它来实现SSO的时候是不需要资源服务器这个角色的，有授权服务器和客户端就够了。授权服务器当然是用来做认证的，客户端就是各个应用系统，我们只需要登录成功后拿到用户信息以及用户所拥有的权限即可

* 用户在某网站上点击使用微信授权，这里的某网站就类似业务系统，微信授权服务器就类似单点登录系统
* 之后微信授权服务器返回一个确认授权页面，类似登录页面，这个页面当然是微信的而不是业务系统的
* 用户确认授权，类似填写了账号和密码，提交后微信授权并返回一个ticket，并重定向业务系统
* 业务系统带上ticket访问微信服务器，微信服务器返回正式的token，业务系统就可以使用token获取用户信息了