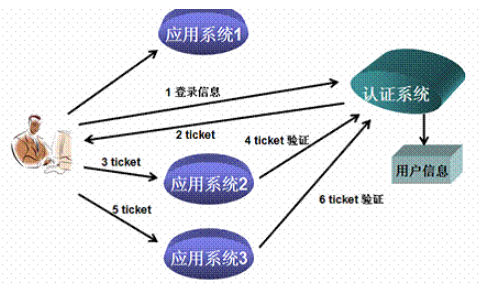
# 一：原理

## 1.客户端代码倾入式

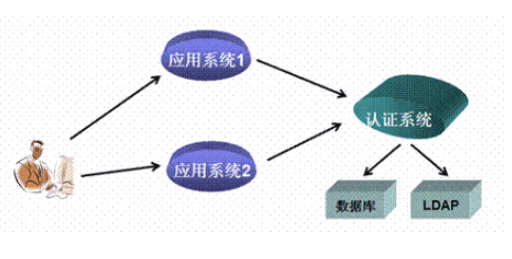


当用户第一次访问应用系统1的时候，因为还没有登录，会被引导到认证系统中进行登录（1）；

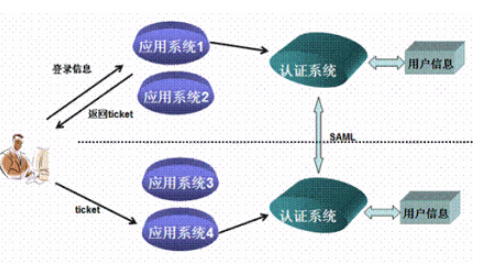
根据用户提供的登录信息，认证系统进行身份效验，如果通过效验，应该返回给用户一个认证的凭据－－ticket（2）；

用户再访问别的应用的时候（3，5）就会将这个ticket带上，作为自己认证的凭据，应用系统接受到请求之后会把ticket送到认证系统进行效验，检查ticket的合法性（4，6）。

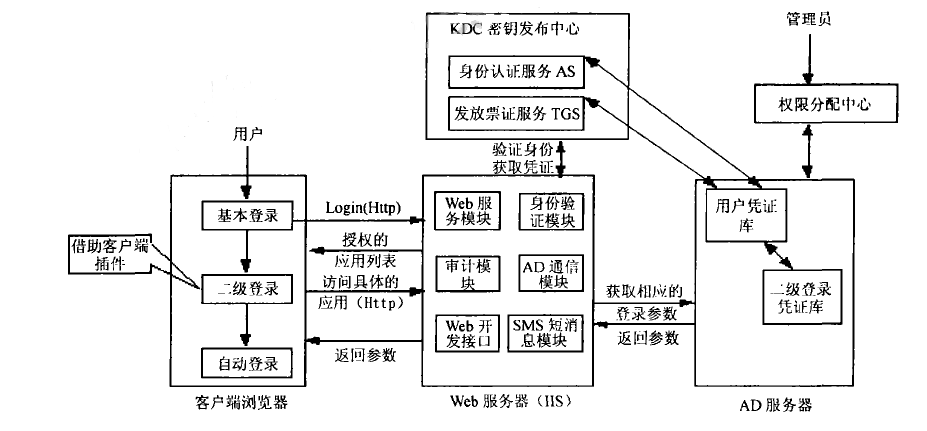
如果通过效验，用户就可以在不用再次登录的情况下访问应用系统2和应用系统3了。



单一的用户信息数据库并不是必须的，有许多系统不能将所有的用户信息都集中存储，应该允许用户信息放置在不同的存储中。事实上，**只要统一认证系统，统一ticket的产生和校验，无论用户信息存储在什么地方，都能实现单点登录**。



统一的认证系统并不是说只有单个的认证服务器，如下图所示，整个系统可以存在两个以上的认证服务器，这些服务器甚至可以是不同的产品。认证服务器之间要通过标准的通讯协议，互相交换认证信息，就能完成更高级别的单点登录。如下图，当用户在访问应用系统1时，由第一个认证服务器进行认证后，得到由此服务器产生的ticket。当他访问应用系统4的时候，认证服务器2能够识别此ticket是由第一个服务器产生的，通过**认证服务器之间标准的通讯协议（例如SAML）来交换认证信息，仍然能够完成SSO的功能**

****

## 2.客户端代码非倾入式

类似一个管理不同应用的用户名密码的中间应用

# 二：方案

1. CAS
2. Windows Azure Active Directory
3. Okta
4. 对于我们的应用场景应该是一个“密码维护工具”
5. <https://github.com/zhoufeihong/SimpleSSO>
6. <https://github.com/pjhu/keycloak>
7. <https://gitee.com/xuxueli0323/xxl-sso>

# 三：okta的研究

有三种应用验证机制：

### Secure Web Authentication (SWA)

Uses credentials to sign in. This integration works with most apps.

### SAML 2.0

Uses the SAML protocol to log users into the app. This is a better option than SWA, if the app supports it.

### OpenID Connect（OAuth 2.0）

Uses the OpenID Connect protocol to log users into an app you've built

这里我选择了第一种验证方式：SWA，将登陆url，主页面url配置好，完成其他基本配置后，通过点击okta中我们已配置成功的应用，后第一次我们只需要我们输入用户名密码即可登陆，下次即可登陆成功。

使用[Okta](https://www.okta.com/)和我的客户端的Active Directory服务器进行SAML身份验证参考：<https://developer.okta.com/blog/2017/03/16/spring-boot-saml>

先后访问链接：

1. <https://qqjiping.okta.com/home/qqorg520301_empi_1/0oai2hcokQrIXe98a356/alni2nb39zBMxr1Is356?fromPlugin=true>
2. <https://qqjiping.okta.com/plugin/bookmark?fromURI=%2Fapp%2Fqqorg520301_empi_1%2F0oai2hcokQrIXe98a356%2Fqqorg520301_empi_1_link&appInstanceId=0oai2hcokQrIXe98a356>
3. <https://qqjiping.okta.com/app/qqorg520301_empi_1/0oai2hcokQrIXe98a356/qqorg520301_empi_1_link>

测试到这里结合我们的项目，适合模仿okta的密码管理然后自动登录（验证机制SWA）

# 四：设计方案

整体流程：

在认证中心平台配置好待管理登录应用，之后在终端用户的浏览器插件中找到配置好的应用，点击后，第一次会让你输入这个应用的登录名密码，平台会存贮记住这个账号密码，之后，跳转到这个应用配置的登录页面自动填充用户名密码。

1. 开发浏览器插件或者统一登陆页面平台（客户端）

①登录注册单点登录

②显示服务端已被注册的应用（用应用ID来代表）

③点击其中一个应用，带着应用ID和sessionId和token去服务端验证这个会话是否非法，合法的话就带着应用ID对应的信息（用户名，密码，登录url）没有用户名密码这个信息就会跳转到输入此应用用户名密码的页面，输入完成后就会带着这些信息到登录url的页面，自动填充用户名密码再自动登录（通过chrome插件可以完成自动登录的效果）

④通过其他方式进入我们的登录页面，插件能自动提示是否用服务端提供的用户名密码（这个功能只有在插件之前获得过页面登录名密码信息才能用这个功能）

1. 搭建一个认证中心平台（服务端）

①应用注册于移除与分配用户

②用户管理

③应用用户密码管理

④提供一个客户端单点登录页面，提供安全完善的认证服务

参考：<https://juejin.im/entry/59d4af1bf265da06687622fb>；<http://www.cnblogs.com/oshine/p/5379349.html>；<https://blog.csdn.net/qustdong/article/details/46046553>