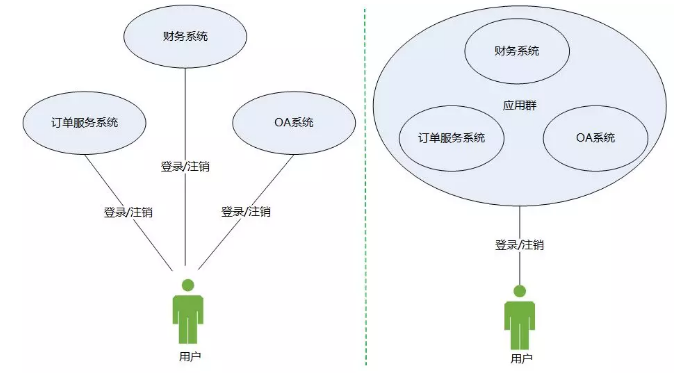
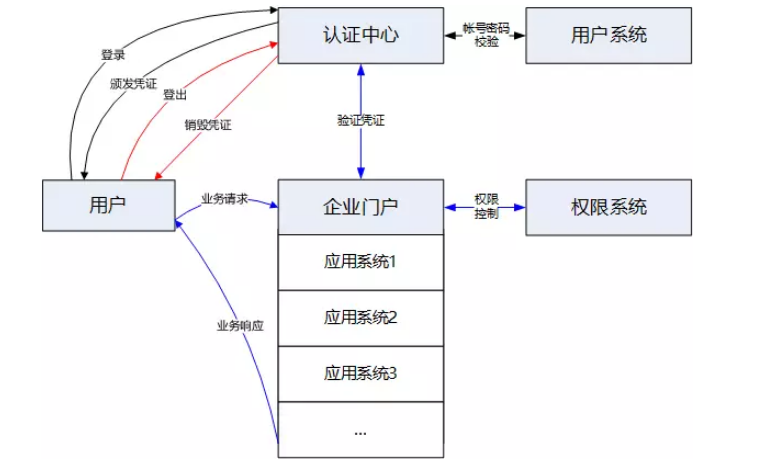
一、单点登录简介

假设一个场景：公司内部有财务、OA、订单服务等各类相互独立的应用系统，员工张三对这些系统有操作权限，如果张三想要登录某个系统进行业务操作，那么他需要输入相应的账号与密码。想象一下，当公司内部有 100 个应用系统，张三是不是要输入 100 次用户名和密码进行登录，然后分别才能进行业务操作呢？显然这是很不好的体验，因此我们需要引入一个这样的机制：张三只要输入一次用户名和密码登录，成功登录后，他就可以访问财务系统、OA 系统、订单服务等系统。这就是单点登录。



单点登录的英文全称是 Single Sign On，简称是 SSO。它的意思是说用户只需要登录一次，就可以在个人权限范围内，访问所有相互信任应用的功能模块，不管整个应用群的内部有多么复杂，对用户而言，都是一个统一的整体。用户访问 Web 系统的整个应用群与访问单个系统一样，登录和注销分别只要一次就够了。举个简单的例子，你登录了百度网页之后，点击跳转到百度贴吧，这时可以发现你已经自动登录了百度贴吧。

二、我们的技术实现



SSO 的技术实现要想做好并不容易，我们认为需求优先级应该先是单点登录和单点注销功能，然后是应用接入的门槛，最后是数据安全性，安全性对于 SSO 也非常重要。SSO 的核心是认证中心，但要实现用户一次登录，到处访问的效果，技术实现需要建立在用户系统、认证中心、权限系统、企业门户的基础上，各职责如下：

* **用户系统**：负责用户名、密码等帐户信息管理，包括增加、修改、启用、停用用户帐号，同时为认证中心提供对用户名和密码的校验。
* **认证中心**：负责凭证 token 的生成、加密、颁发、验证、销毁、登入 Login、登出 Logout。用户只有拥有凭证并验证通过才能访问企业门户。
* **权限系统**：负责角色管理、资源设置、授权设置、鉴定权限，具体实现可参考 RBAC。权限系统可为企业门户提供用户权限范围内的导航。
* **企业门户**：作为应用系统的集成门户 (Portal)，集成了多个应用系统的功能，为用户提供链接导航、用户信息和登出功能等。

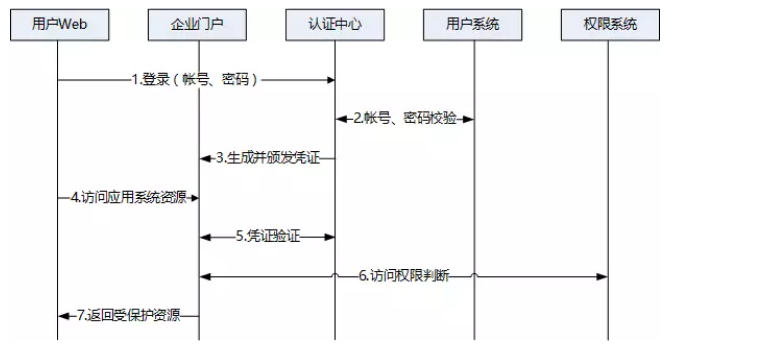
 2.1、服务端功能实现

* 登录认证：接收登录帐号信息，让用户系统验证用户的登录信息。
* 凭证生成：创建授权凭证 token，生成的凭证一般包含用户帐号信息、过期时间等信息，它是一串加密的字符串，加密算法如 AES｛凭证明文 +MD5 加信息｝，可采用 JWT 标准。
* 凭证颁发：与 SSO 客户端通信，发送凭证给 SSO 客户端。
* 凭证验证：接收并校验来自 SSO 客户端的凭证有效性，凭证验证包括算法验证和数据验证。
* 凭证销毁与登出：接收来自 SSO 客户端的登出请求，记录并销毁凭证，跳转至登录页面。

 2.2、客户端功能实现

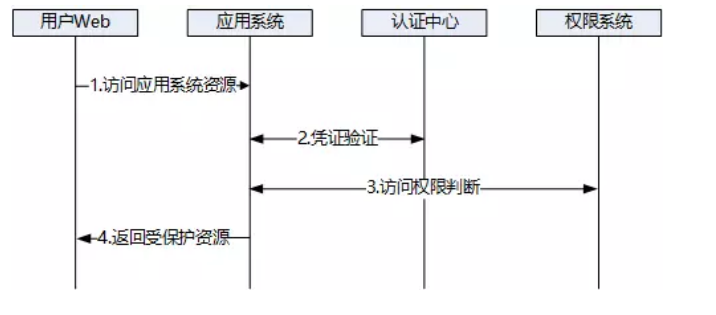
* 1、请求拦截：拦截应用未登录请求，跳转至登录页面。
* 2、获取凭证：接收并存储由 SSO 服务端发来的凭证，凭证存储的方式有 Cookie、Session、网址传参、Header 等。
* 3、提交凭证验证：与 SSO 服务端通信，发出校验凭证有效性的请求。
* 4、获取用户权限：获取该凭证的用户权限，并返回受保护资源给用户。
* 5、凭证销毁与登出：销毁本地会话，然后跳转至登出页面。

 2.3、用户单点登录流程



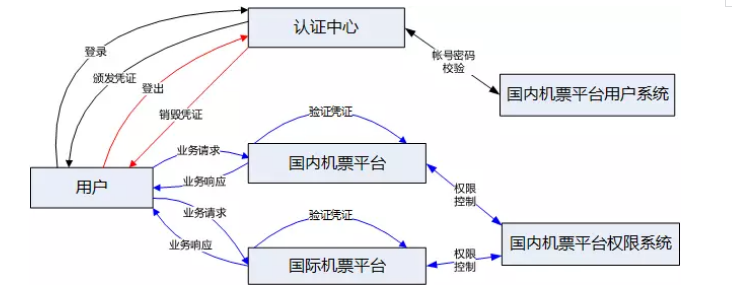
* 登录：将用户输入的用户名和密码发送至认证中心，然后认证中心调用用户系统来验证登录信息。
* 生成并颁发凭证：通过登录信息的验证后，认证中心创建授权凭证 token，然后把这个授权凭证 token 返回给 SSO 客户端。SSO 客户端拿到这个 token，进行存储。在后续请求中，在 HTTP 请求数据中都得加上这个 token。
* 凭证验证：SSO 客户端发送凭证 token 给认证中心，认证中心校验这个 token 的有效性。凭证验证有算法验证和数据验证，算法验证可在 SSO 客户端完成。

 2.4、用户访问流程和单点注销



以上是用户的访问流程，如果用户没有有效的凭证，认证中心将强制用户进入登录流程。对于单点注销，用户如果注销了应用群内的其中一个应用，那么全局 token 也会被销毁，应用群内的所有应用将不能再被访问。

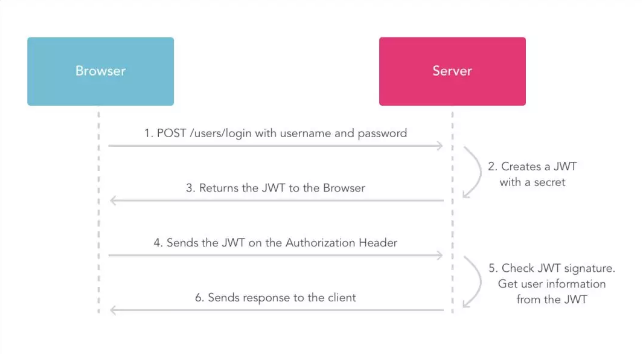
 2.5、具体接入与集成



我们的应用接入与集成具体如下：

* 1、用户系统：接入国内机票平台的用户系统，负责登录认证。
* 2、权限系统：接入国内机票平台的权限系统。
* 3、认证中心：负责生成并颁发凭证、销毁凭证，改造国内机票平台的登入、登出。
* 4、凭证验证：在国内机票、国际机票应用系统中调用 SSO 客户端组件实现凭证的验证。
* 5、企业门户：由国内机票平台、国际机票平台承担。

三、JWT 标准



JSON Web Token (JWT) 是目前应用最为广泛的 token 格式，是为了在网络应用环境间传递声明而执行的一种基于 JSON 的开放标准（RFC 7519)。该 token 设计紧凑且安全，特别适用于分布式站点的单点登录、API 网关等场景。JWT 的声明一般被用来在身份提供者和服务提供者间传递被认证的用户身份信息，以便于从资源服务器获取资源，也可以增加一些额外的其它业务逻辑所必须的声明信息。该 token 也可直接被用于认证，也可被加密。JWT 信息体由 3 部分构成：头 Header+ 载荷 Payload+ 签名 Signature，具体优点如下：

* JWT 支持多种语言，C#、Java、JavaScript、Node.js、PHP 等很多语言都可以使用。
* JWT 可以自身存储一些和业务逻辑有关的所必要的非敏感信息，因为有了 Payload 部分。
* 利于传输，因为 JWT 的构成非常简单，字节占用很小。
* 不需要在服务端保存会话信息，不仅省去服务端资源开销，而且使得应用易于扩展。