# Shiro 身份认证授权原理

## 核心概念

Subject：主体，代表了当前操作“用户”，这个用户不一定是具体的人，与当前应用交互的任何东西都是subject，即一个抽象概念。所有Subject都绑定到SecurityManager，与Subject交互都会委托给SercurityManager；

SecurityManager：安全管理器；即所有与subject安全有关的操作都会与SecurityManager交互；且它管理着所有的Subject；它负责与里面的各个组件的交互，也可以把它理解成springmvc中的DispatcherServlet前端控制器。

Realm：域，安全数据源。shiro从Realm获取安全数据（如用户、权限、角色），就是说SecurityManager要验证用户身份，那么它需要从Realm得到用户相应的角色、权限进行验证用户是否能进行操作；可以把Realm看成DataSource，即安全数据源。

Realm

Shiro

SecurityManager

Application

Code

（the current user）

（manages all Subjects）

（access you security data）

1、应用代码通过Subject来进行认证和授权，而Subject又委托给SecurityManager；

2、SecurityManager要验证用户身份，那么它需要从Realm中获取相对应的用户、角色、权限进行比较以确定用户身份是否合法。

3、Shiro不提供维护用户、权限，而是通过Ream让开发人员自己注入。

4、Ream包含2个主要方法：

1. **doGetAuthorizationInfo**

用户认证方法，获取输入用户名和密码，根据用户名查询用户密码，判定密码是否相同。认证用户信息。

1. **doGetAuthorizationInfo**

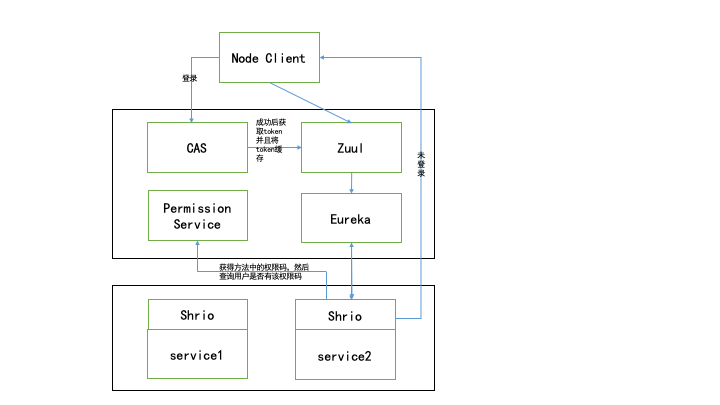
请求资源时通过资源的注解获取该资源的权限码，在认证用户信息中查询用户是否拥有该权限码。

## 目前微服务中的使用

为了保证后续服务的采用统一的登录方式，采用CAS单点登录技术做身份认证。

为了将权限控制在方法级，使用Shiro框架做权限认证。

架构示意图如下：



1、Client通过Zuul网关请求service服务，shiro判定用户未登录，返回给Client

2、Client重定向到CAS服务，并以附加参数形式通过Zuul请求Service服务。

3、CAS登录成功，Shrio获取CAS登录成功的token，放入缓存。

4、Shiro通过Permission Service获取用户的权限码列表，通过注解获取请求Service的权限码，在用户权限码列表中查询是否存在当前请求的权限码。