# 统一认证鉴权设计概要

## 背景

在微服务架构下，一个应用会被拆分成若干个微应用，每个微应用都需要对访问进行鉴权，每个微应用都需要明确当前访问用户以及其权限。尤其当访问来源不只是浏览器，还包括其他服务的调用时，单体应用架构下的鉴权方式就不是特别合适了。在微服务架构下，要考虑外部应用接入的场景、用户–服务的鉴权、服务–服务的鉴权等多种鉴权场景。

比如用户A访问账户系统Account Service，A如果未登录，则首先需要登录，请求获取授权token。获取token之后，A将携带着token去请求访问某个文件，这样就需要对A的身份进行校验，确定A是否可以访问该文件。

为了适应架构的变化、需求的变化，统一认证鉴权模块被单独出来作为一个基础的微服务系统，为其他业务service提供服务。

## 技术方案

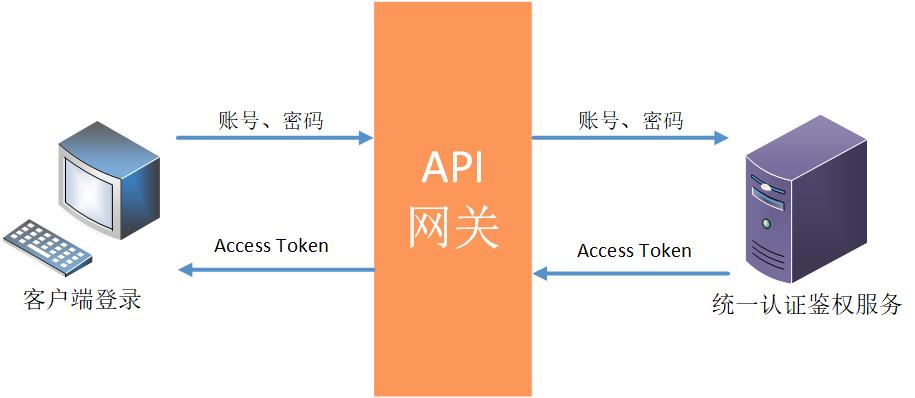
用户发出的请求主要有两种：

1. 用户尚未登录。客户端发起登录请求，网关对于登录请求直接转发到auth服务，auth服务对用户身份信息进行校验。
2. 用户已登录，请求其他服务。这种情况，客户端的请求到达网关，网关会将请求转发到对应的微服务中。由这个微服务确定用户能否调用该接口。

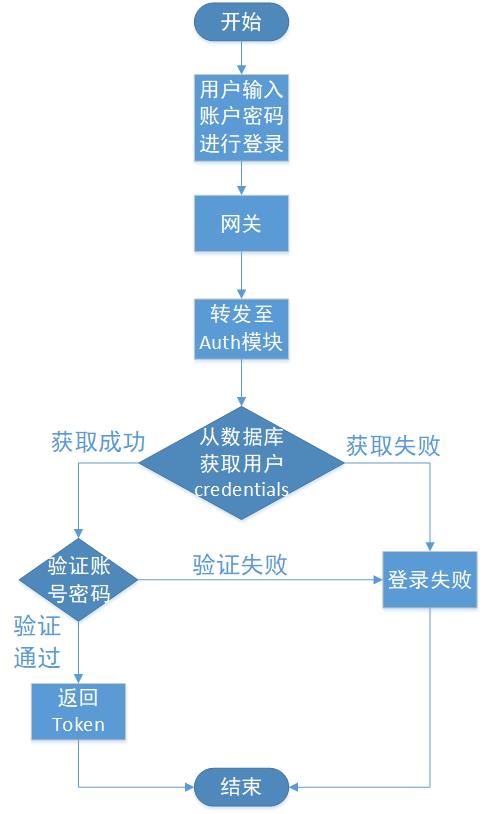
### 用户认证

用户认证部分，主要采用基于OAuth2 Token认证的方式。随着 Restful API、微服务的兴起，基于Token的认证现在已经越来越普遍。Token并非只是一个 key。Token 一般会包含用户的相关信息，通过验证 Token 就可以完成身份校验。

用户输入登录信息，发送到统一认证鉴权服务进行认证。统一认证鉴权服务验证登录信息是否正确，正确则返回包含用户基础信息、权限、有效时间等信息的Access Token。



登录流程图如下所示：



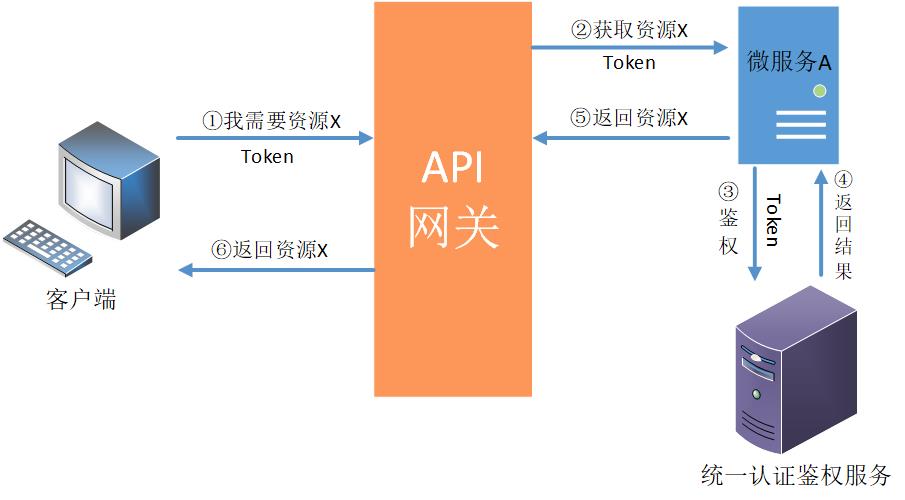
OAuth2根据使用场景不同，分成了4种模式：授权码模式（authorization code）、简化模式（implicit）、密码模式（resource owner password credentials）和客户端模式（client credentials）。在三化物联网平台的场景下，我们选择密码模式。

### API级别的权限控制

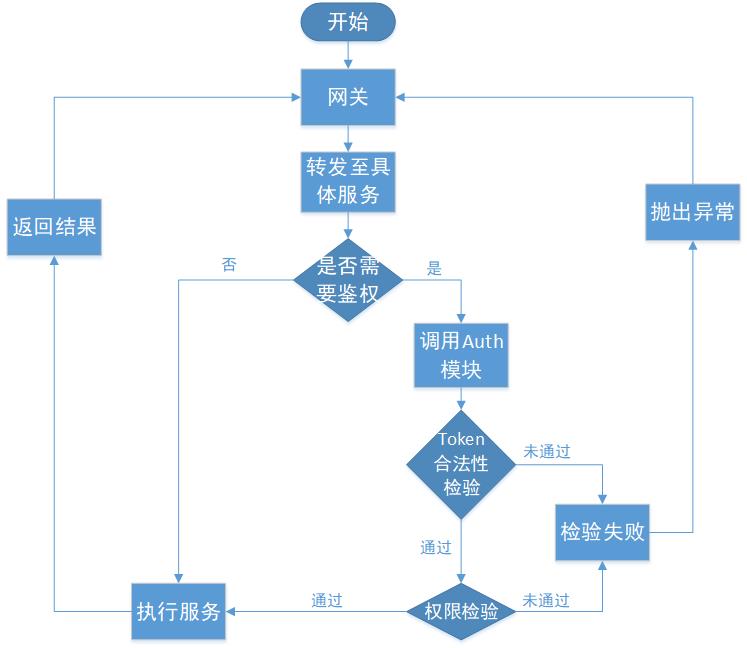
对于操作权限的控制，主要是基于Spring Security框架进行实现。

当鉴定完用户身份合法之后，对于该认证用户，我们还需要对他所进行的操作进行鉴权，以确定该用户是否拥有调用相关API的权限。

每当用户发出请求时，请求头部都会携带一个Token，该Token会被转发到统一认证鉴权模块进行合法性验证（确定该Token是系统签发）以及鉴权（确定该用户的权限），并将结果返回。各个微服务会根据返回结果来确定该请求的权限是否符合要求。符合要求则执行该服务，否则拒绝执行。



对于部分不需要权限就可以调用的接口（例如用户认证中所调用的登录接口），不需要鉴权就可以直接调用。以下是鉴权的流程图：



当服务间进行接口调用时，也需要进行鉴权。调用服务需要向鉴权中心请求token，并且携带这个token向被调用服务进行请求。被调用服务会进一步向鉴权中心请求鉴权，判定该请求是否应该被响应。

