性能优化专题

OAuth2

主讲人：ROBERT: 2831742582

目录 /CONTENTS

01 应用场景

02 认证授权模式

03 简单的系统

04 手机客户端

01

应用场景

APPLY

案例一，停车事件

OAuth2解决：资源授权问题

去酒店

服务生使用泊车钥匙代泊车

酒店停车事件

1、开壕车到酒店
2、酒店服务生代泊车
3、泊车钥匙（只能开两公里，不能打开车内酒柜）
4、泊车
5、取车（自己的钥匙，全功能）

定义

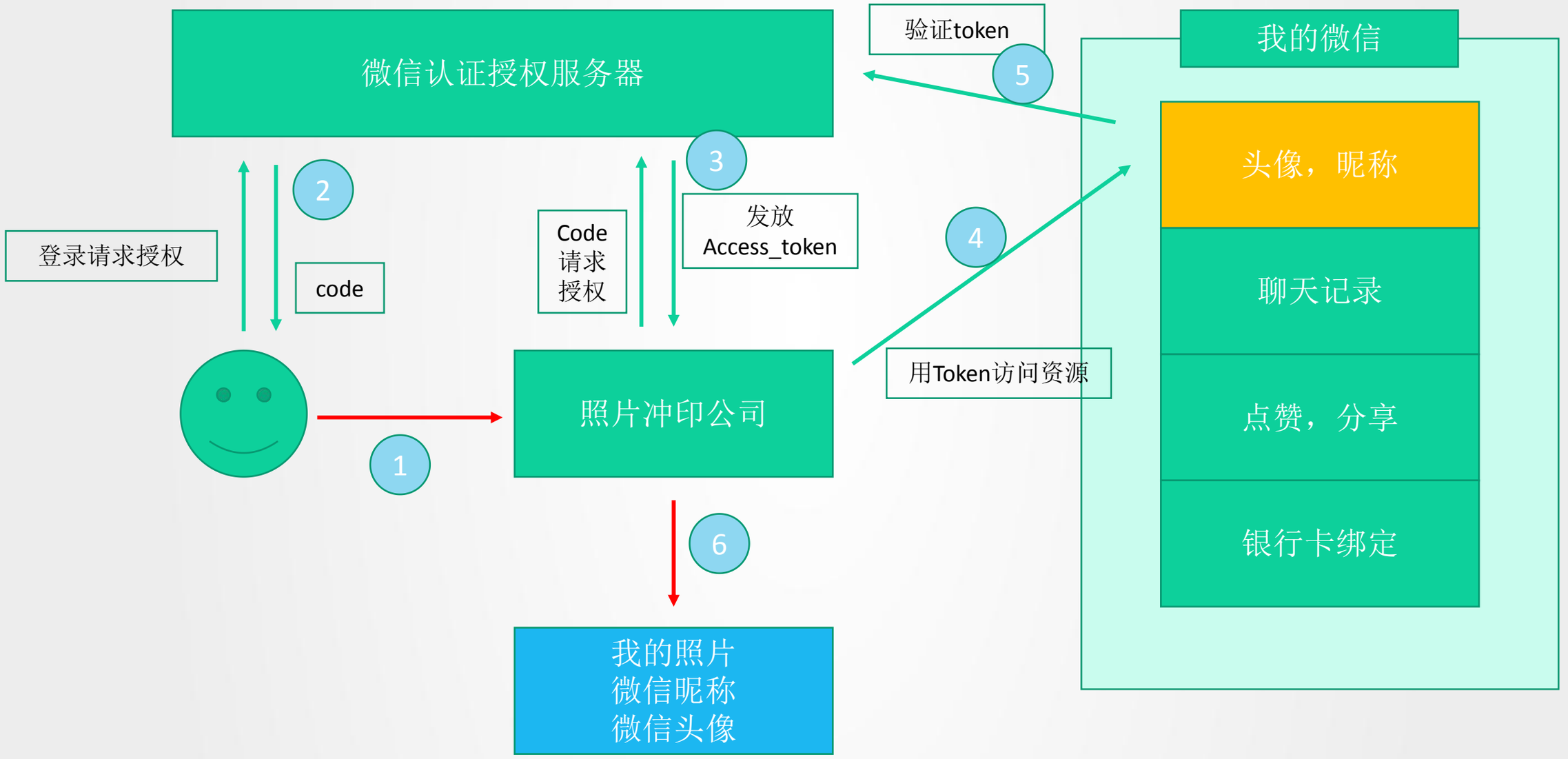
OAuth（Open Authorization，开放授权）是为用户资源的授权定义了一个安全、开放及简单的标准，第三方无需知道用户的账号及密码，就可获取到用户的授权信息，并且这是安全的。

专用名词：

（1）**Third-party application**：第三方应用程序，又称"客户端"（client）。
（2）**Resource Owner**：资源所有者，又称"用户"（user）。
（3）**User Agent**：用户代理，指浏览器。
（4）**Authorization server**：认证服务器，即服务提供商专门用来处理认证的服务器。
（5）**Resource server**：资源服务器，即服务提供商存放用户生成的资源的服务器。它与认证服务器，可以是同一台服务器，也可以是不同的服务器。

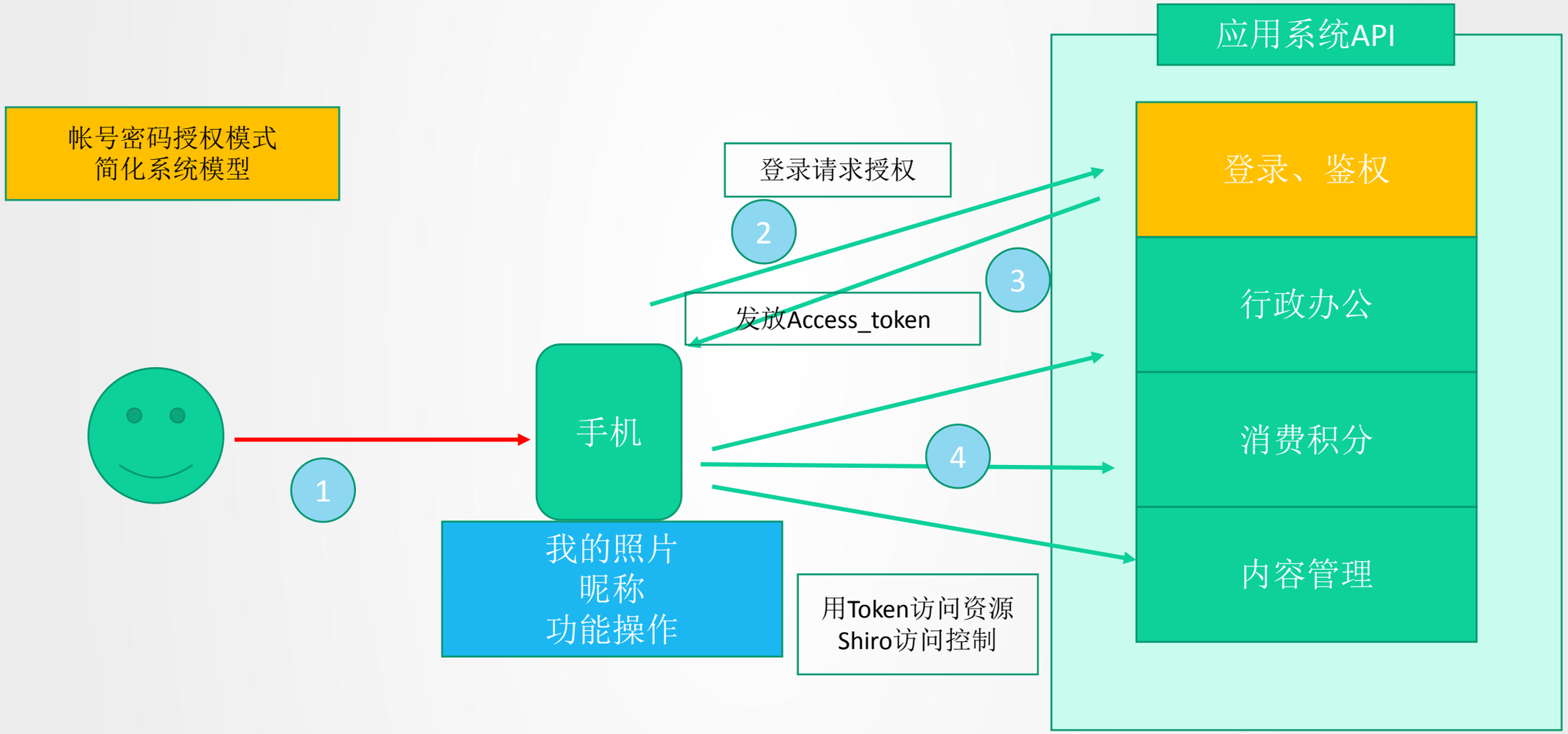
场景一、微信登录

OAuth2解决：资源授权问题



场景二、移动端登录

OAuth2解决：资源授权问题



其他应用场景

- 扫码登录
- 优惠码发放
- 有时限的邀请码
- 短信，邮箱验证
- 多平台共用的api及授权

02

认证授权模式

AUTHRIZATION

四种模式

授权码模式(Authorization Code)

先登录，获取code
客户端使用code取得access_token，服务器销毁code
使用access_token访问资源

帐号密码模式(Resource Owner Password Credentials)

直接使用帐号密码获取access_token
客户端使用access_token访问资源

客户端模式((Client Credentials Grant)

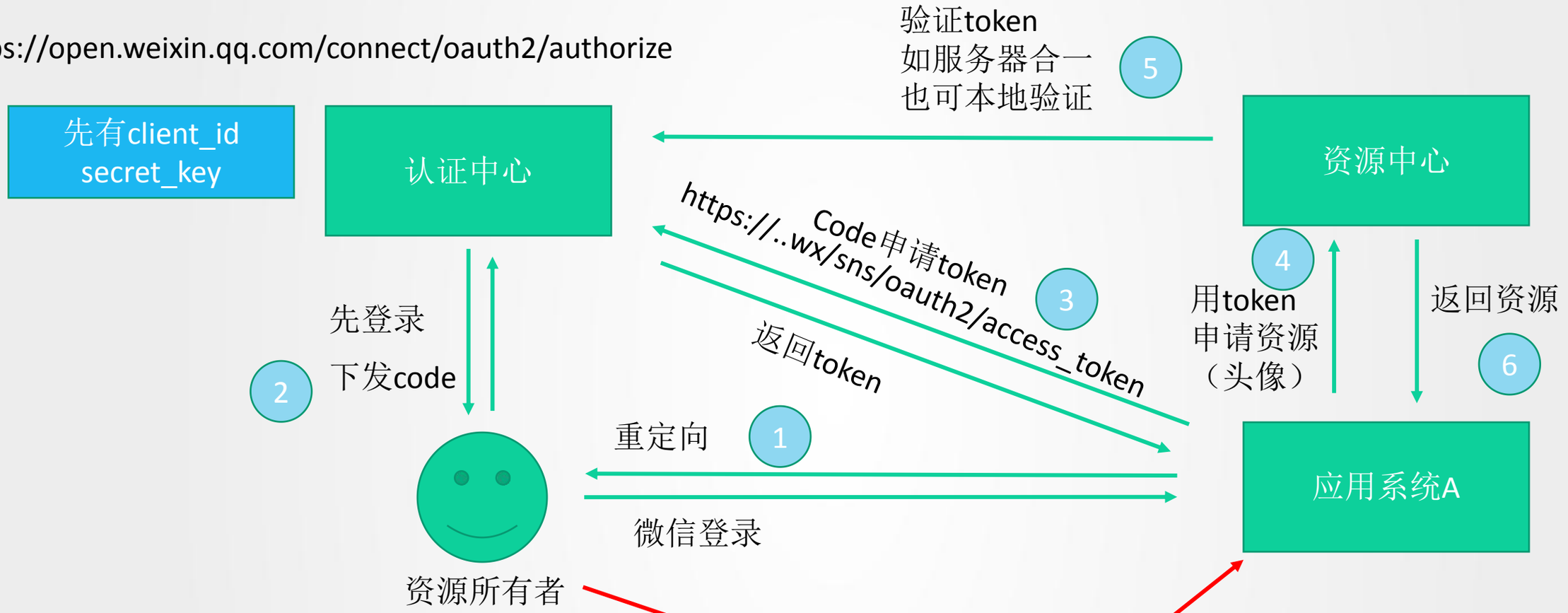
客户端就是用户，自身申请认证。

紧凑模式(Implicit)

直接在浏览器中向认证服务器申请令牌，无需经过client端的服务器，
所有步骤在浏览器中完成，直接在回调url中传递令牌。

授权码模式

https://open.weixin.qq.com/connect/oauth2/authorize



Authorize接口参数

- response_type：表示授权类型，必选项，此处的值固定为"code"
- client_id：表示客户端的ID，必选项
- redirect_uri：表示重定向URI，可选项
- scope：表示申请的权限范围，可选项
- state：表示客户端的当前状态，可以指定任意值，认证服务器会原封不动地返回这个值。

token接口参数

grant_type：表示使用的授权模式，必选项，此处的值固定为"authorization_code"。

code：表示上一步获得的授权码，必选项。

redirect_uri：表示重定向URI，必选项，且必须与A步骤中的该参数值保持一致。

client_id：表示客户端ID，必选项。

授权码模式

授权码模式是最复杂的，也是最安全的

1、客户端请求验证，由用户获取code

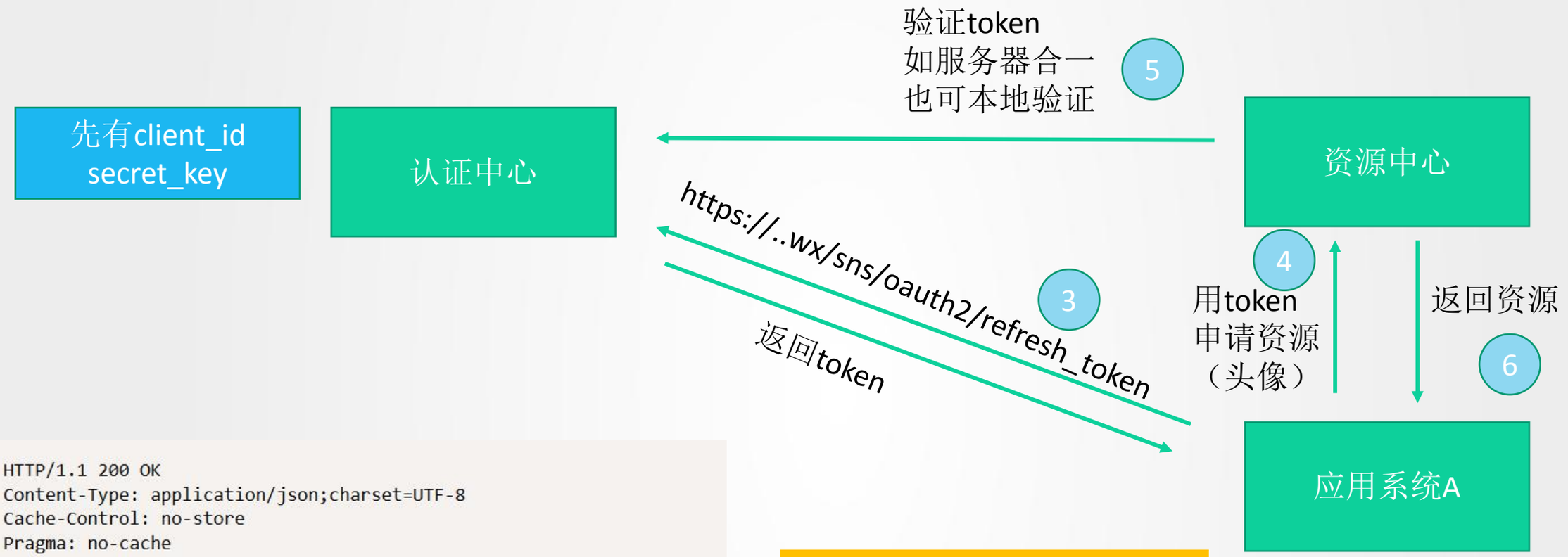
2、客户端拿到code，请求token

3、销毁code，下发token，而用户拿不到token，客户端保存

4、客户端使用token访问资源

5、过期后使用refresh_token刷新token再次使用

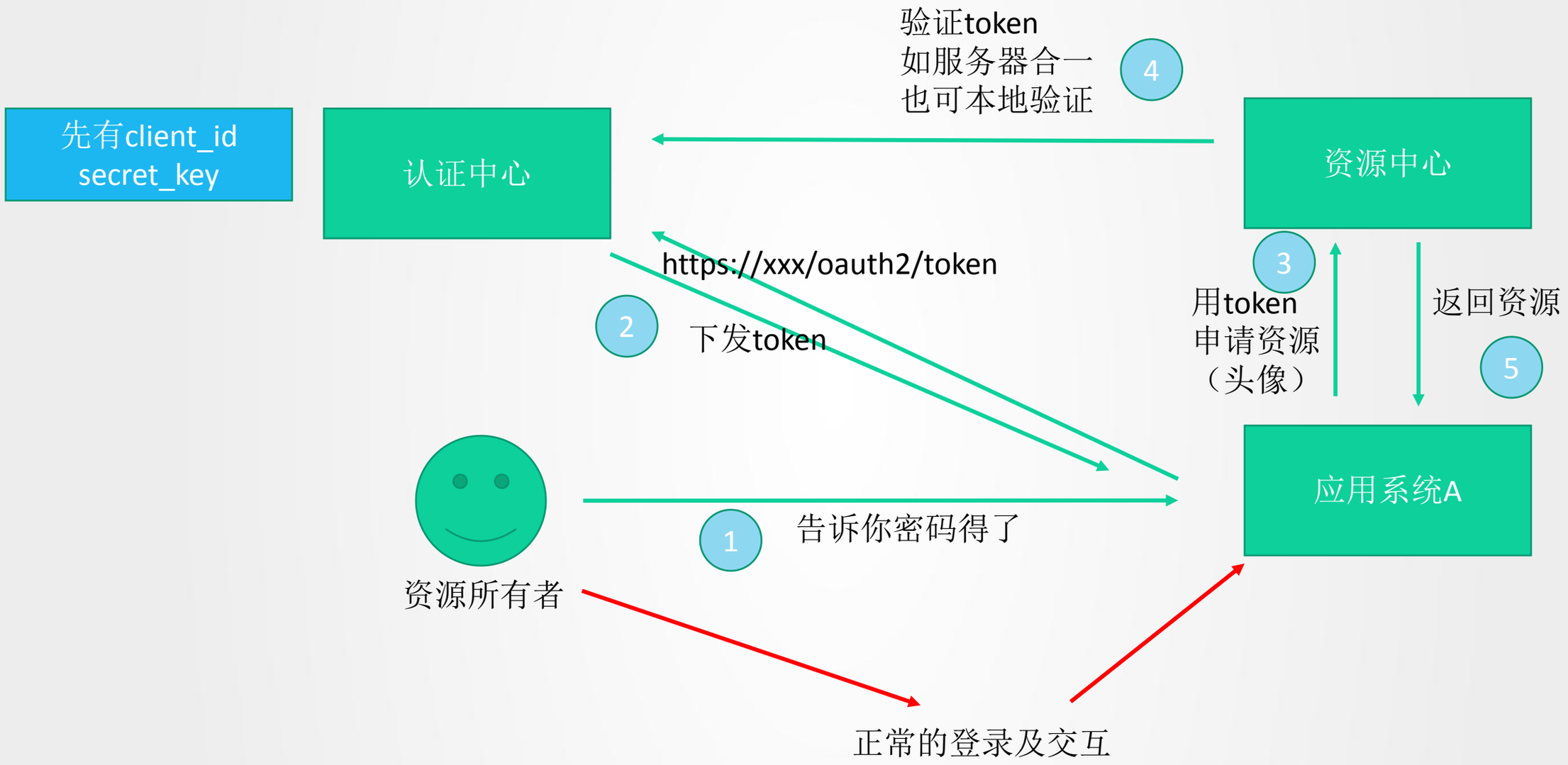
授权码模式-token过期



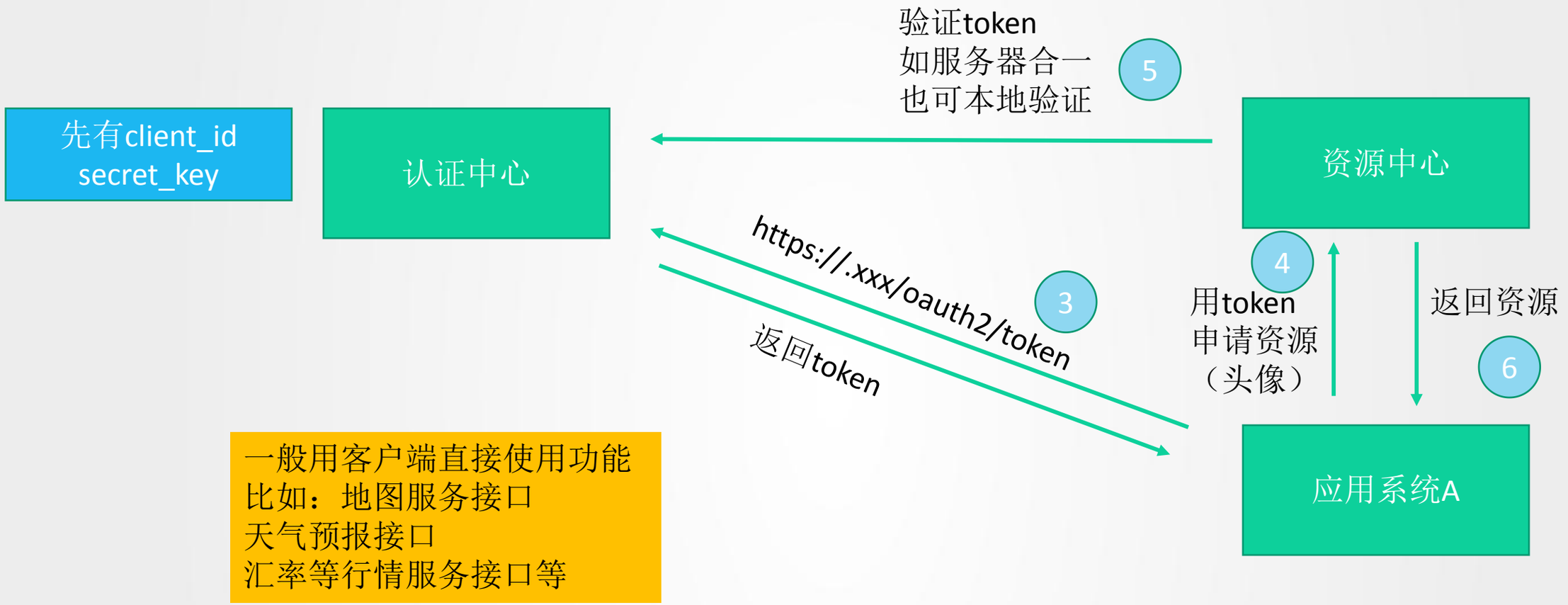

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token":"2YotnFZFEjr1zCsicMWpAA",
  "token_type":"example",
  "expires_in":3600,
  "refresh_token":"tGzv3JOkF0XG5Qx2TlKWIA",
  "example_parameter":"example_value"
}
```

用refresh_toke直接申请token

帐号密码模式


先有client_id
secret_key
认证中心
验证token
如服务器合一
也可本地验证
4
资源中心
https://xxx/oauth2/token
2
下发token
3
用token
申请资源
（头像）
返回资源
5
应用系统A
1
告诉你密码得了
资源所有者
正常的登录及交互

客户端模式



一般用客户端直接使用功能
比如：地图服务接口
天气预报接口
汇率等行情服务接口等

客户端模式

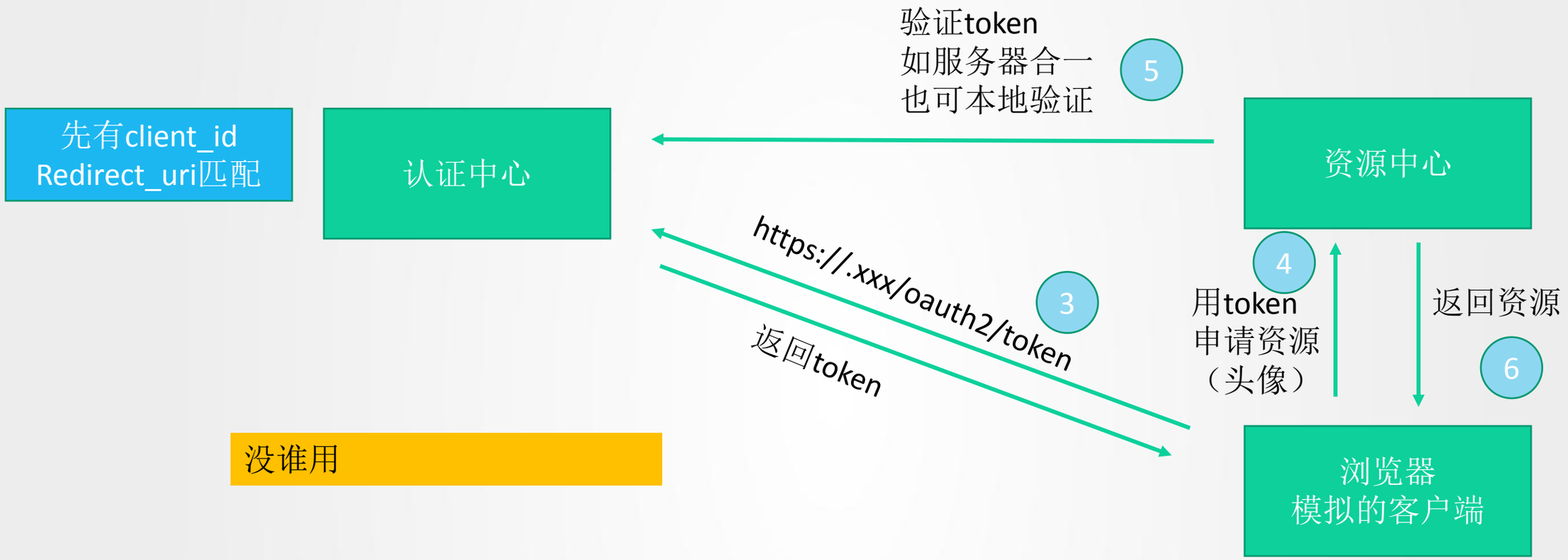
公开服务调用的一种类型

1、与用户无关的应用

2、服务器之间通信

3、对用户透明，增强站点功能的一类

简化模式（紧凑）


先有client_id
Redirect_uri匹配
认证中心
验证token
如服务器合一
也可本地验证
5
资源中心
https://.xxx/oauth2/token
3
返回token
4
用token
申请资源
（头像）
返回资源
6
浏览器
模拟的客户端

没谁用

简化模式（紧凑）

不要用它，不要用它，不要用它

1、不支持refresh_token

2、浏览器即客户端

3、用户拿到token，可能安全性有问题

03

简单的系统

SIMPLE SYSTEM

Code模式

认证中心
AuthorizeController
AccessTokenController
Token管理
客户端管理

资源中心
UserinfoController
Token校验

Apache Oltu组件

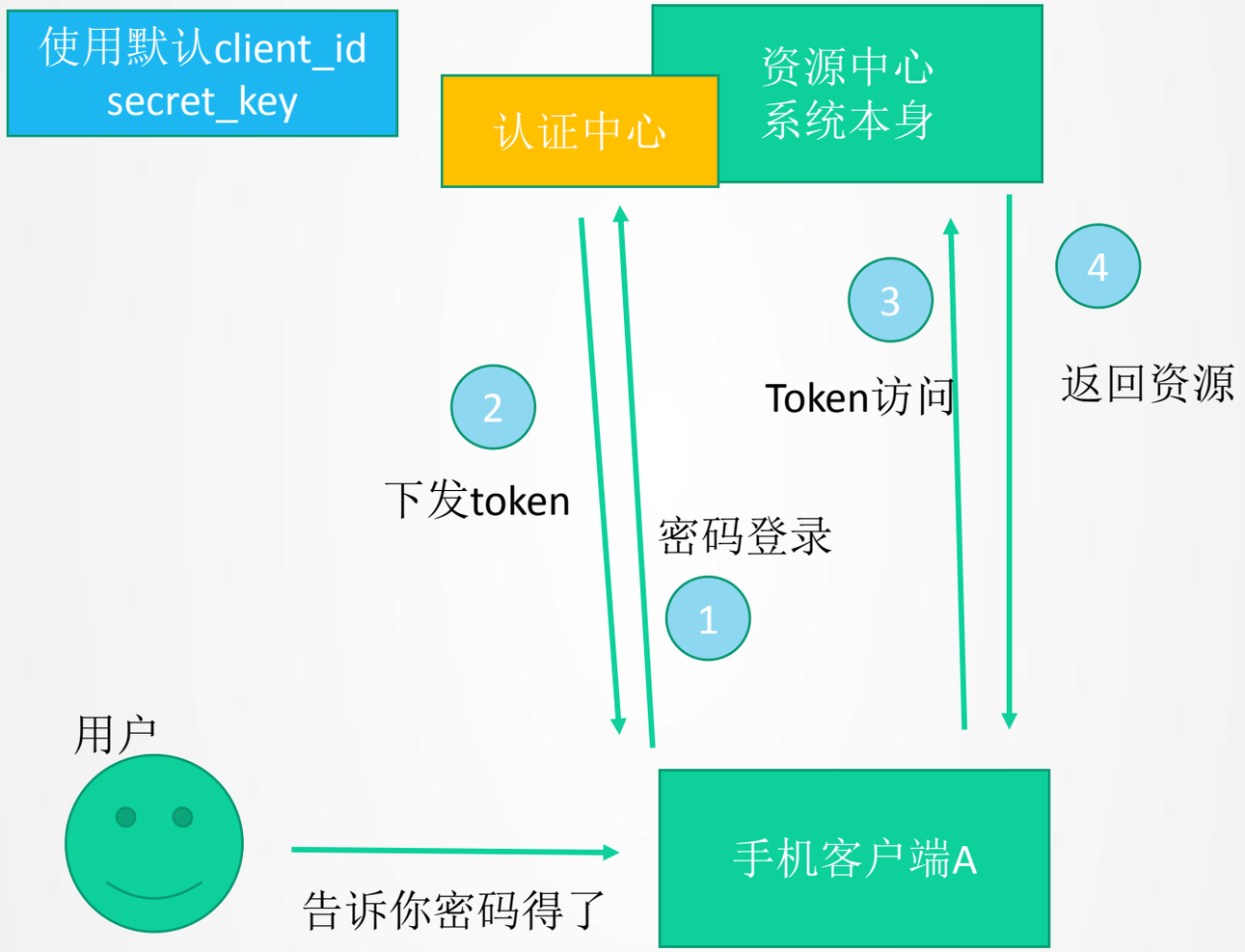
应用系统A
LoginController
Token暂存
RefreshToken暂存

04

手机客户端

MOBILE CLIENT

告诉你密码得了系列


使用默认client_id
secret_key
认证中心
资源中心
系统本身
4
3
2
返回资源
Token访问
下发token
密码登录
1
用户
手机客户端A
告诉你密码得了