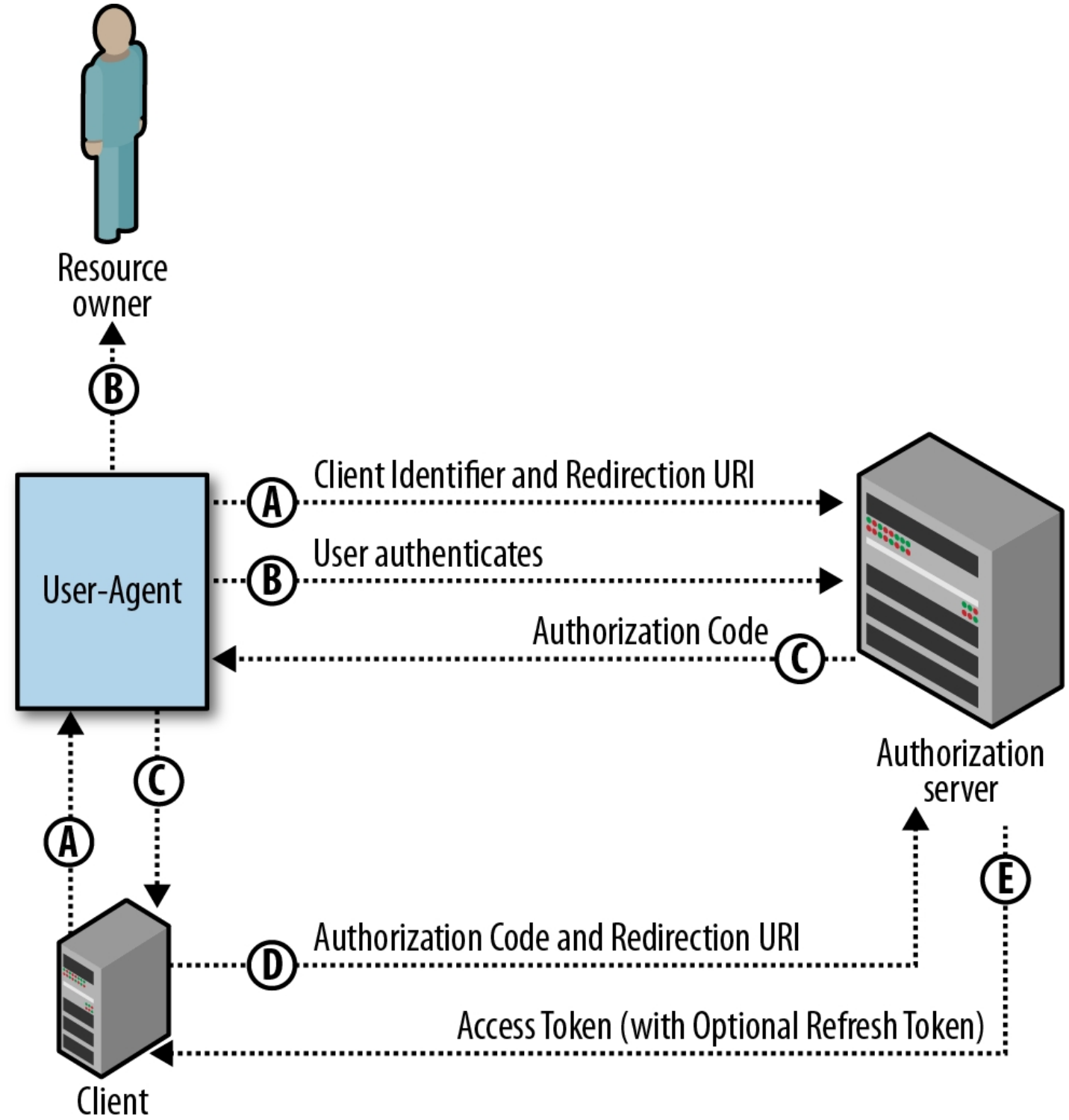
适用范围

需要得到长期授权

OAuth客户端是Web应用服务器

OAuth访问令牌不宜泄露给用户的环境

流程剖析



1. 让用户明白所做的操作并请求认证

当牵涉到OAuth认证时，首先应最好能更进一步的让用户知道该操作到底会发生什么。在用户确认之后，这时应用应将用户引导至OAuth认证页面。在该页面中，API提供者会向用户说明应用会授权访问用户数据。

该授权接口的URL会在开发者文档中给出，以谷歌为例：

https://accounts.google.com/o/oauth2/auth

在请求该页面时还需附带几个参数：

client_id

在应用注册时提供

redirect_uri

授权认证后的重定向地址

scope

应用所请求访问的数据，一般由空格分隔的多个字符串组成

response_type

对于此授权类型来说为“code”，即在授权成功后返回一个认证code

state

一个随机字符串，用于防止跨站攻击（CSRF）

这是一般的情况，对于不同的开放平台来说其所需的参数可能有所增减。

对于正常授权的情况，用户将会随后被重定向到redirect_uri指定的URL下。反之，则会返回错误信息。“access_denied”可能是最常见的错误，当然也有一些其他情况，例如：

invalid_request

这种情况一般说来是参数传递有误

unauthorized_client

client_id未授权

unsupported_response_type

不支持此授权类型

invalid_scope

所请求的scope不正确

server_error

授权服务器出错

temporarily_unavailable

授权服务器不可用

2. 交换authorization code用于获取access_token

如果授权一切顺利，授权服务器会将用户重定向到redirect_uri指定的URL上，并附带几个参数：

code

即所需要的authorization code

state

与请求时的state参数值相同，用于确定来源

然后就是利用该code去获取access_token，这是一个HTTP POST请求并需要携带如下参数：

code

授权得到的authorization code

redirect_uri

和之前的相同

grant_type

这里应是authorization_code

该请求还需附带应用注册时得到的client_id以及client_secret。

如果一切顺利，授权服务器将会以JSON格式返回希望的数据：

access_token

API的访问令牌

token_type

下发的访问令牌类型，通常是“bearer”

access token可能具有时效性，因此还可能含有一下一些内容：

expires_in

access token的有效期，以秒为单位

refresh_token

用于在access token过期后再次获取新的access token

以下是一个JSON格式返回的示例：

```
{
    "access_token" : "ya29.AHES6ZSzX",
    "token_type" : "Bearer",
    "expires_in" : 3600,
    "refresh_token" : "1/iQI98wWFfJNFWIzs5EDDrSiYewe3dFqt5vIV-9ibT9k"
}
```

3. 访问API

这就不用具体阐述了。

access tokens和refresh tokens同时存在的用意

简单来说，这有助于增加安全性并在某些环境下提升性能。