OAuth

- four roles
  - resource owner -- a person, or the end-user
  - resource server -- hosting the protected resources accessed by tokens
  - client -- making protected resource requests
  - authorization server -- issuing access tokens to the client

- others
  - user agent -- web browsers, desktop applications, etc.

Protocol Flow

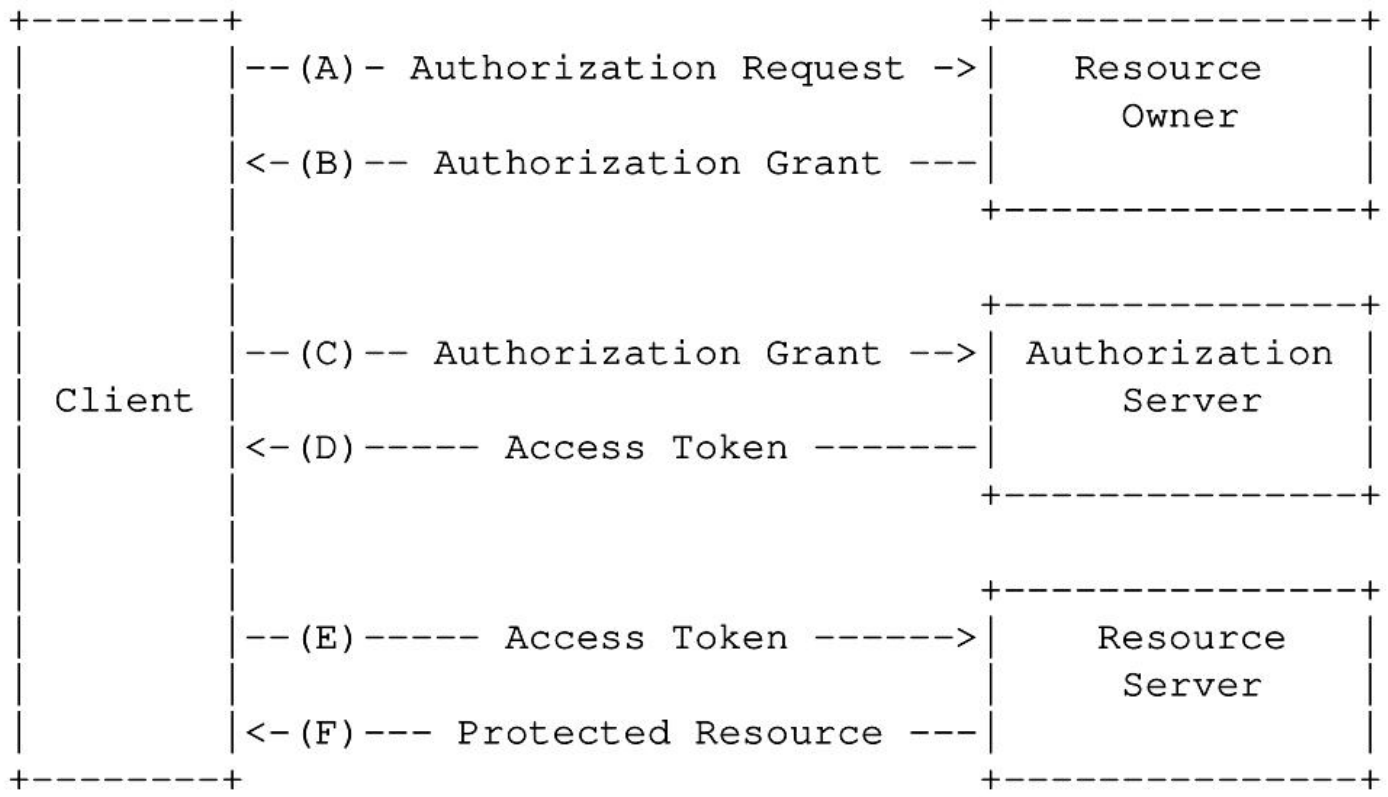


Figure 1: Abstract Protocol Flow

（A）用户打开客户端以后，客户端请求用户给予授权。

（B）用户同意给予客户端授权。

（C）客户端使用上一步获得的授权，向认证服务器申请令牌。

（D）认证服务器对客户端进行认证以后，确认无误，同意发放令牌。

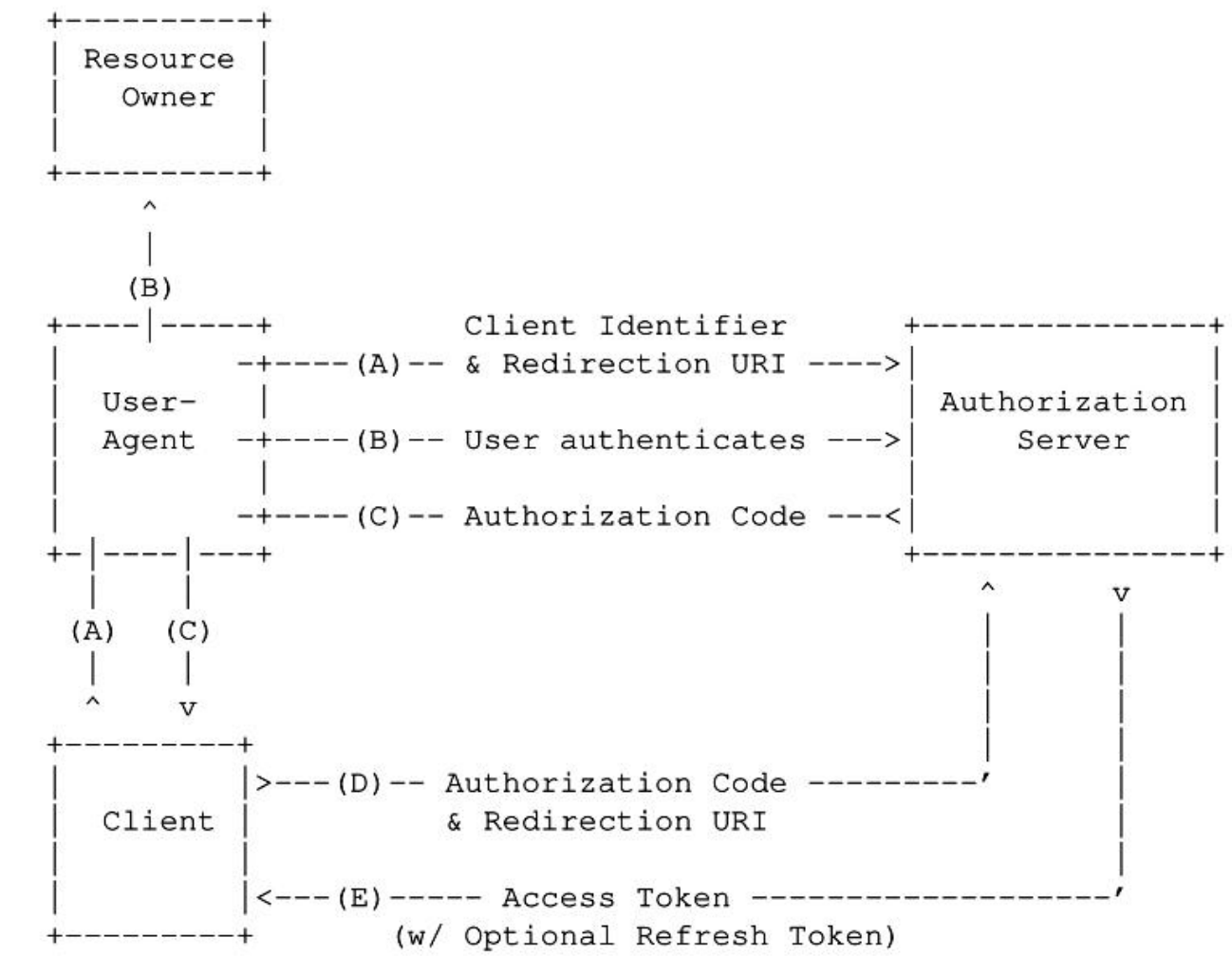
（E）客户端使用令牌，向资源服务器申请获取资源。

（F）资源服务器确认令牌无误，同意向客户端开放资源。

Authorization Grant Types

1. Authorization Code 授权码模式
2. Implicit 简化模式
3. Resource Owner Password Credentials 密码模式
4. Client Credentials 客户端模式

Authorization Code



Note: The lines illustrating steps (A), (B), and (C) are broken into two parts as they pass through the user-agent.

（A）用户访问客户端，后者将前者导向认证服务器。

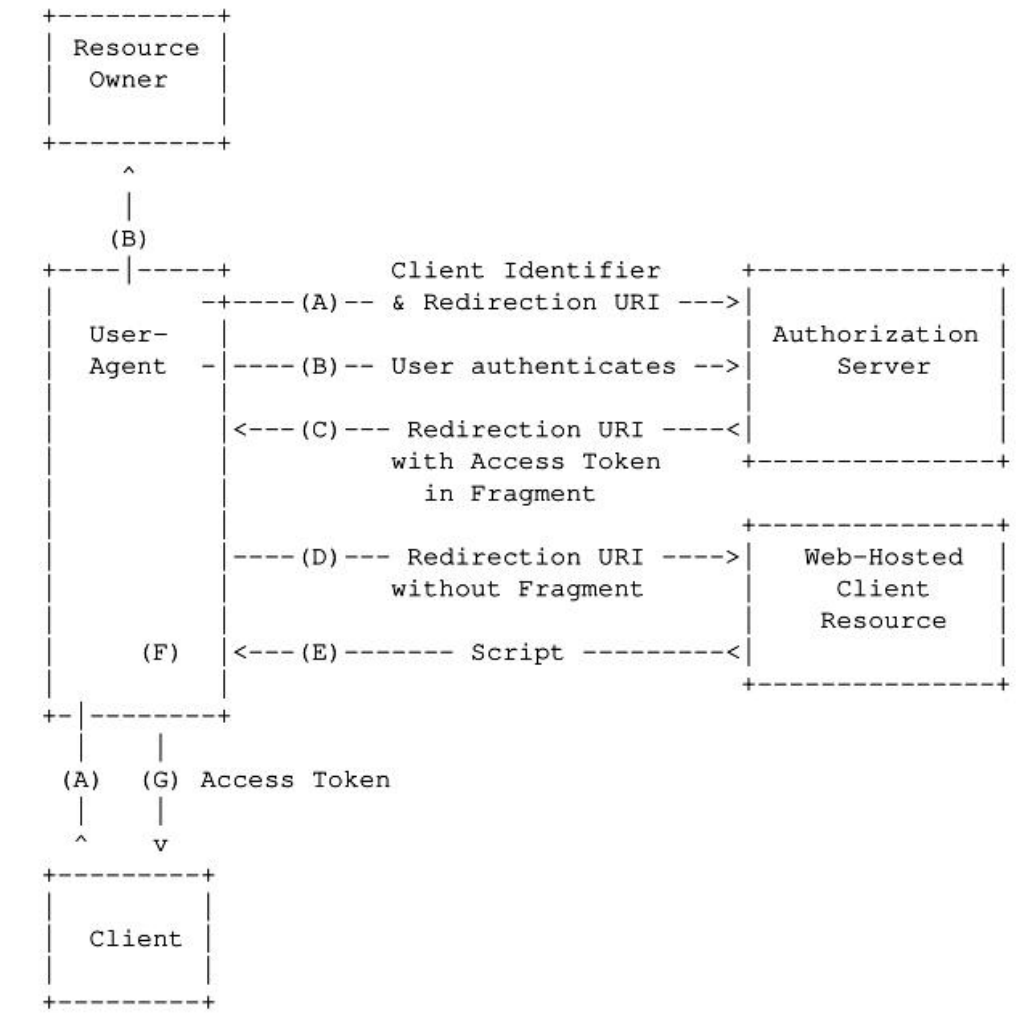
（B）用户选择是否给予客户端授权。

（C）假设用户给予授权，认证服务器将用户导向客户端事先指定的"重定向URI"（redirection URI），同时附上一个授权码。

（D）客户端收到授权码，附上早先的"重定向URI"，向认证服务器申请令牌。这一步是在客户端的后台的服务器上完成的，对用户不可见。

（E）认证服务器核对了授权码和重定向URI，确认无误后，向客户端发送访问令牌（access token）和更新令牌（refresh token）。

Implicit Grant



Note: The lines illustrating steps (A) and (B) are broken into two parts as they pass through the user-agent.

（A）客户端将用户导向认证服务器。

（B）用户决定是否给于客户端授权。

（C）假设用户给予授权，认证服务器将用户导向客户端指定的"重定向URI"，并在URI的Hash部分包含了访问令牌。

（D）浏览器向资源服务器发出请求，其中不包括上一步收到的Hash值。

（E）资源服务器返回一个网页，其中包含的代码可以获取Hash值中的令牌。

（F）浏览器执行上一步获得的脚本，提取出令牌。

（G）浏览器将令牌发给客户端。

Resource Owner Password Credentials Grant

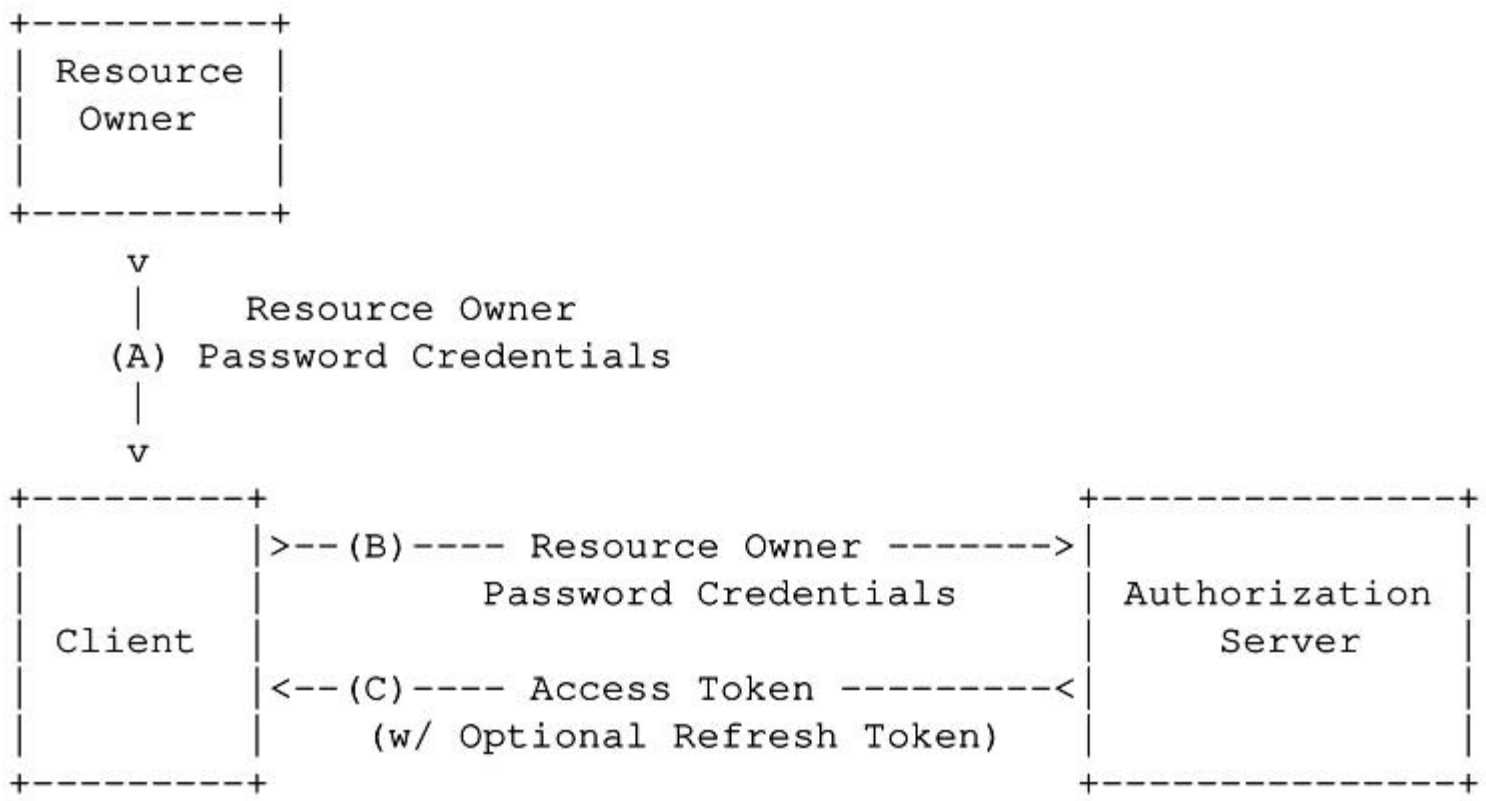


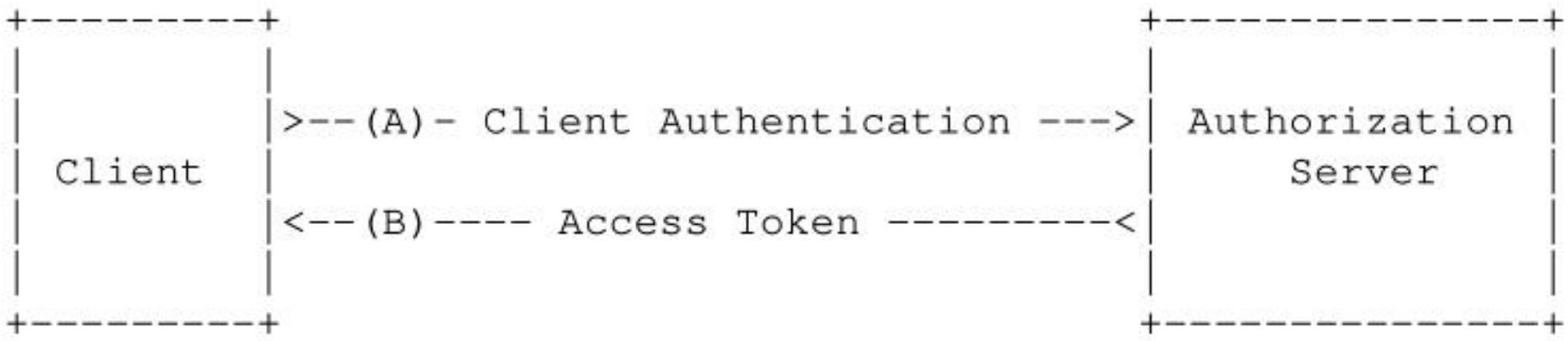
Figure 5: Resource Owner Password Credentials Flow

（A）用户向客户端提供用户名和密码。

（B）客户端将用户名和密码发给认证服务器，向后者请求令牌。

（C）认证服务器确认无误后，向客户端提供访问令牌。

Client Credentials Grant



（A）客户端向认证服务器进行身份认证，并要求一个访问令牌。

（B）认证服务器确认无误后，向客户端提供访问令牌。

不涉及用户授权

Implementation

https://github.com/DeprecatedCode/oauth2lib