In the traditional client-server authentication model, the client

requests an access-restricted resource (protected resource) on the

server by authenticating with the server using the resource owner's

credentials. In order to provide third-party applications access to

restricted resources, the resource owner shares its credentials with

the third party. This creates several problems and limitations:

o Third-party applications are required to store the resource

owner's credentials for future use, typically a password in

clear-text.

o Servers are required to support password authentication, despite

the security weaknesses inherent in passwords.

o Third-party applications gain overly broad access to the resource

owner's protected resources, leaving resource owners without any

ability to restrict duration or access to a limited subset of

resources.

o Resource owners cannot revoke access to an individual third party

without revoking access to all third parties, and must do so by

changing the third party's password.

### 

### Roles

OAuth defines four roles:

**resource owner**

An entity capable of granting access to a protected resource.

When the resource owner is a person, it is referred to as an

end-user.

**resource server**

The server hosting the protected resources, capable of accepting

and responding to protected resource requests using access tokens.

**client**

An application making protected resource requests on behalf of the

resource owner and with its authorization. The term "client" does

not imply any particular implementation characteristics (e.g.,

whether the application executes on a server, a desktop, or other

devices).

**authorization server**

The server issuing access tokens to the client after successfully

authenticating the resource owner and obtaining authorization.

The interaction between the authorization server and resource server

is beyond the scope of this specification. The authorization server

may be the same server as the resource server or a separate entity.

A single authorization server may issue access tokens accepted by

multiple resource servers.