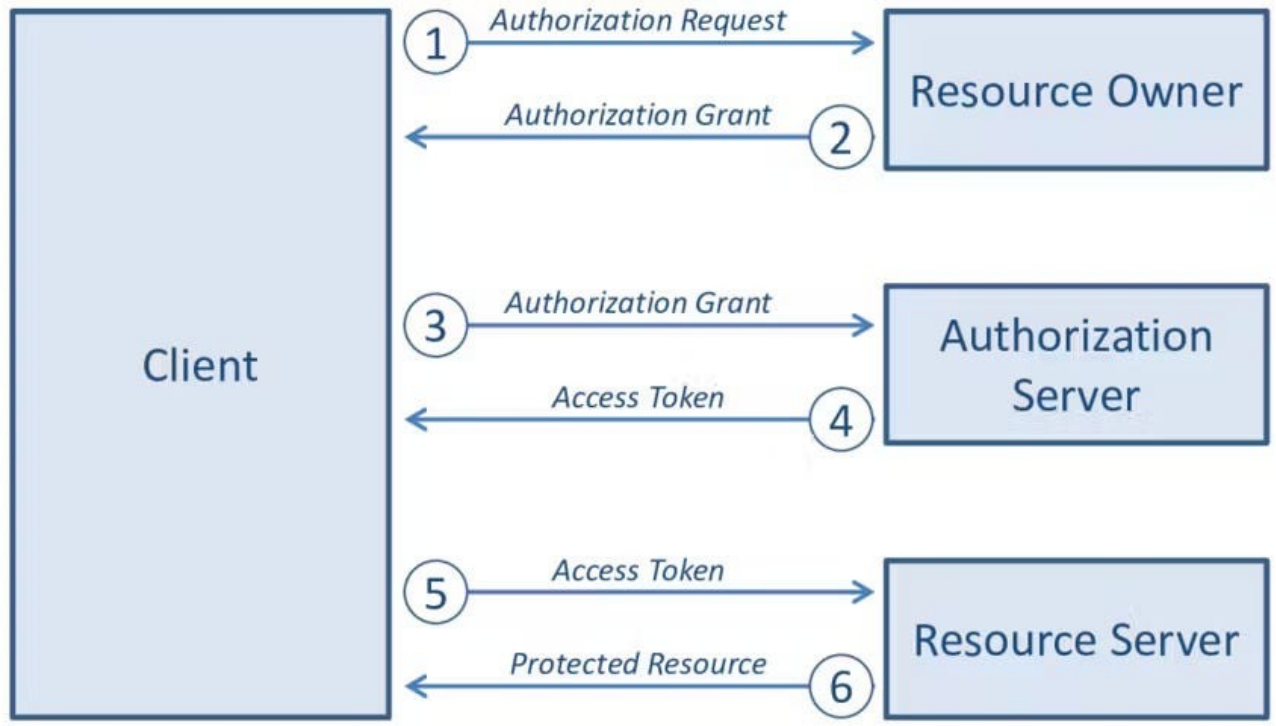
Şema (Abstract Protocol Flow)

Şemada dört ana rolün (Client, Resource Owner, Authorization Server, Resource Server) nasıl etkileşimde bulunduğu ve OAuth 2.0 sürecinin adım adım nasıl ilerlediği gösterilmiştir:



Bu şema, protokolün akışını altı adımda açıklıyor. Her bir adımı detaylı olarak ele alalım:

Adımlar

(1) Authorization Request (Yetkilendirme İsteği)

- **Ne Olur?**
    - **Client (İstemci)**, **Resource Owner (Kaynak Sahibi)**'nden yetkilendirme ister.
    - Bu istek, doğrudan kaynak sahibine yapılabilir, ancak genellikle **Authorization Server (Yetkilendirme Sunucusu)** bir aracı olarak kullanılır.
- **Örnek:**
    - Bir kullanıcı, bir mobil uygulamanın sosyal medya hesaplarına erişmesine izin vermek için giriş yapar.

---

(2) Authorization Grant (Yetkilendirme Belgesi)

- **Ne Olur?**
    - **Resource Owner**, istemciye bir "authorization grant" (yetkilendirme belgesi) verir.
    - Bu belge, istemcinin korunan kaynaklara erişim iznini temsil eder.
    - Bu belge dört türden biri olabilir:

        - Authorization Code
        - Implicit Grant
        - Resource Owner Password Credentials
        - Client Credentials
- **Örnek:**
    - Kullanıcı, sosyal medya platformu üzerinden uygulamaya erişim izni verdiğinde, uygulama bir "authorization code" alır.

(3) Access Token Request (Erişim Token'ı İsteği)

- **Ne Olur?**
    - **Client**, **Authorization Server** ile iletişime geçer.
    - Daha önce aldığı "authorization grant" belgesini sunarak bir "access token" talep eder.
- **Örnek:**
    - Mobil uygulama, sosyal medya platformuna, aldığı "authorization code" ile birlikte bir "access token" talebi gönderir.

---

(4) Access Token Issuance (Erişim Token'ının Verilmesi)

- **Ne Olur?**
    - **Authorization Server**, istemciyi ve "authorization grant" belgesini doğrular.
    - Her şey uygunsa istemciye bir "access token" sağlar.
- **Örnek:**
    - Sosyal medya platformu, doğrulama işlemlerini başarıyla tamamladıktan sonra uygulamaya bir "access token" verir.

---

(5) Protected Resource Request (Korunan Kaynak İsteği)

- **Ne Olur?**
    - **Client**, **Resource Server**'a erişmek istediği korunan kaynak için bir istek gönderir.
    - Bu isteği "access token" ile birlikte sunar.
- **Örnek:**
    - Mobil uygulama, kullanıcıya ait fotoğrafları almak için sosyal medya platformunun API'sine "access token" ile bir istek gönderir.

---

(6) Resource Access (Kaynağa Erişim)

- **Ne Olur?**
    - **Resource Server**, sunulan "access token"ı doğrular.
    - Token geçerliyse korunan kaynağı istemciye sağlar.

- **Örnek:**
    - Sosyal medya platformu, "access token"ın geçerli olduğunu doğrular ve kullanıcının fotoğraflarını uygulamaya iletir.

Notlar

1. **Authorization Server'ın Aracı Rolü:**
    - **(1)** ve **(2)** adımlarında, **Authorization Server** genellikle aracı olarak kullanılır.
    - Bu, kullanıcı deneyimini kolaylaştırır ve güvenliği artırır.
2. **Access Token:**
    - "Access token", istemcinin kaynak sunucusuna erişim için kullanacağı geçici bir kimlik belgesidir.
    - Token, belirli bir kapsam (scope) ve süreyle sınırlıdır.
3. **Bağımsız Roller:**
    - **Authorization Server** ve **Resource Server**, aynı sunucuda çalışabilir veya farklı sunucular olarak yapılandırılabilir.
    - Tek bir yetkilendirme sunucusu, birden fazla kaynak sunucusuna erişim sağlayabilir.