2. Client Registration bölümü, OAuth 2.0 protokolünün başlatılmasından önce bir **client** (istemci) uygulamasının yetkilendirme sunucusuna nasıl kaydolması gerektiğini açıklar. Bu kayıt süreci, istemcinin yetkilendirme sunucusuyla etkileşimde bulunmasını gerektirir. Ancak, bu etkileşimlerin ne şekilde gerçekleşeceği ve kaydın nasıl yapılacağı, OAuth 2.0 spesifikasyonunun kapsamı dışında kalır. Ancak genellikle, bu süreç bir HTML kayıt formu aracılığıyla bir kullanıcı etkileşimi gerektirir.

1. İstemci Kayıt Süreci

İstemci kaydının gerekliliği, OAuth 2.0 protokolünü başlatmadan önce istemcinin yetkilendirme sunucusuna kaydolmasını zorunlu kılar. Ancak, kayıt sürecinde istemcinin doğrudan yetkilendirme sunucusuyla etkileşimde bulunması her zaman gerekmez. Yetkilendirme sunucusu, kaydı başka yollarla da yapabilir. Örneğin, istemcinin güvenilir bir kanal aracılığıyla bir **self-issued** (kendiliğinden verilen) veya **third-party-issued** (üçüncü tarafça verilen) bir beyanname sunması, ya da istemcinin kaydını gerçekleştirmek için bir **client discovery** (istemci keşfi) süreci uygulanabilir.

Bu, istemci kayıt işleminin, bazı durumlarda doğrudan etkileşim gerektirmeden de gerçekleştirilebileceği anlamına gelir.

2. Kayıt Sürecinde İstemcinin Sağlaması Gereken Bilgiler

İstemci kaydını gerçekleştiren geliştirici, aşağıdaki bilgileri sağlamalıdır:

- **İstemci tipi**: İstemci kaydında, istemcinin tipi belirtilmelidir. Bu, OAuth 2.0 protokolü için çok önemlidir çünkü istemcinin güvenlik özellikleri, protokolde nasıl işlem yapacağı ve hangi tür verilere erişim talep edeceği, istemci tipine göre değişir.
- **Redirection URIs**: Kayıt sırasında, istemci uygulaması tarafından kullanılan **redirection URIs** (yönlendirme URI'ları) belirtilmelidir. Bu URI'lar, istemci uygulamasının yetkilendirme sunucusundan alacağı yanıtların yönlendirilmesi gereken yerlerdir. Bu bilgiler, istemci ile sunucu arasındaki iletişimde güvenliği sağlamak için kritik öneme sahiptir.
- **Diğer gerekli bilgiler**: İstemci kaydı için, yetkilendirme sunucusunun talep edebileceği diğer bilgiler de sağlanmalıdır. Örneğin, istemci uygulamasının adı, web sitesi, açıklaması, logosu ve yasal şartları kabul etme gibi bilgiler de istenebilir.

3. Güven ve Güvenlik

Kaydın güvenli bir şekilde yapılması için, istemci ve yetkilendirme sunucusu arasında güvenli bir kanal üzerinden etkileşim sağlanmalıdır. Bu, istemci bilgilerinin doğru bir şekilde alınıp kaydedilmesini ve doğru istemci tipinin ve diğer bilgilerin doğrulanmasını sağlar.

Özetle

Client Registration, istemci uygulamasının OAuth 2.0 protokolüne katılabilmesi için yetkilendirme sunucusuna kaydolmasını gerektirir. Bu kayıtta, istemcinin tipi, redirection URI'ları gibi bilgilerin yanı sıra diğer güvenlik önlemleri de dikkate alınır. Ancak, doğrudan istemci-yetkilendirme sunucusu etkileşimi her zaman gerekli değildir ve farklı güvenli yöntemlerle istemci kaydı yapılabilir.

2.1. Client Types bölümü, OAuth 2.0 protokolünde istemcilerin iki ana tipte sınıflandırıldığını açıklar. Bu sınıflandırma, istemcilerin yetkilendirme sunucusuyla güvenli bir şekilde kimlik doğrulaması yapabilme yeteneklerine dayanır. Temelde, istemcilerin kimlik bilgilerini (client credentials) gizli tutma kapasitesine göre iki ana türü belirlenmiştir:

1. Confidential Clients (Gizli İstemciler)

Confidential istemciler, kimlik bilgilerini güvenli bir şekilde saklayabilen ve yetkilendirme sunucusu ile güvenli bir şekilde kimlik doğrulaması yapabilen istemcilerdir. Bu istemciler, genellikle güvenli bir sunucu üzerinde çalışır ve istemci kimlik bilgilerine yalnızca kısıtlı erişimi olan uygulamalar tarafından erişilebilir.

- Örnek: Web uygulamaları, API istemcileri veya sunucu tarafı uygulamaları.
- Güvenlik: Bu istemciler, kimlik bilgilerini (client credentials) saklamak ve bunları güvenli bir şekilde kullanmak için gereken altyapıya sahiptir. Bu nedenle, bu tür istemciler daha güvenli kabul edilir.

2. Public Clients (Açık İstemciler)

Public istemciler, kimlik bilgilerini güvenli bir şekilde saklayamayan istemcilerdir. Bu istemciler, genellikle kullanıcının cihazında çalışan uygulamalardır (örneğin, yerel uygulamalar veya web tarayıcısında çalışan uygulamalar). Bu tür istemciler, kimlik bilgilerini güvenli bir şekilde saklayamazlar çünkü istemci kodu, kullanıcıya açık bir şekilde dağıtılır ve kullanıcıya veya cihazdaki diğer uygulamalara erişilebilir.

- Örnek: Yerel uygulamalar (native applications), web tarayıcı tabanlı uygulamalar.
- Güvenlik: Bu istemcilerde, kimlik bilgileri genellikle uygulama içinde açıkça saklanır, bu da güvenlik risklerine yol açabilir. Bu tür istemciler, genellikle kimlik doğrulaması için başka güvenlik önlemleri (örneğin, kullanıcının kimliği doğrulandıktan sonra kısa süreli erişim jetonları kullanma) gerektirir.

3. Client Type Belirlenmesi

İstemci türü, yetkilendirme sunucusunun güvenli kimlik doğrulama tanımına ve istemci kimlik bilgilerini açıklamaya karşı kabul edilebilir seviyesine dayanır. Yetkilendirme sunucusu, istemci türü konusunda varsayımlar yapmamalıdır, yani her istemci türünün güvenlik gereksinimleri farklıdır ve her birine uygun şekilde yaklaşılmalıdır.

4. Dağıtık İstemciler

Bir istemci, birden fazla bileşen içeriyor olabilir ve her bileşenin farklı bir istemci türü ve güvenlik bağlamı olabilir. Örneğin, bir dağıtık istemci hem gizli istemci bileşenlerine (sunucu tabanlı) hem de açık istemci bileşenlerine (tarayıcı tabanlı) sahip olabilir. Eğer yetkilendirme sunucusu bu tür istemcileri desteklemiyorsa, her bileşen ayrı bir istemci olarak kaydedilmelidir.

5. OAuth 2.0 İstemci Profilleri

OAuth 2.0, farklı istemci türleri için spesifik profiller tanımlar:

- **Web Application (Web Uygulaması)**: Bu, bir **confidential client** (gizli istemci) olup, web sunucusunda çalışır. Kaynak sahipleri, bu istemciye HTML kullanıcı arayüzü aracılığıyla

erişir. İstemci kimlik bilgileri ve erişim jetonları yalnızca web sunucusunda saklanır ve kaynak sahibine veya diğer kullanıcı ajanlarına (örneğin, tarayıcılar) erişilebilir değildir.

- **User-Agent-Based Application (Kullanıcı Ajansı Tabanlı Uygulama)**: Bu, bir **public client** (açık istemci) olup, istemci kodu bir web sunucusundan indirilir ve kullanıcı ajanı (örneğin, web tarayıcısı) üzerinde çalıştırılır. Bu tür istemcilerde, kimlik bilgileri ve protokol verileri kaynak sahibi tarafından kolayca erişilebilir ve genellikle görünürdür. Ancak, kullanıcı ajanının özellikleri, yetkilendirme talebinde bulunurken yardımcı olur.
- **Native Application (Yerel Uygulama)**: Bu da bir **public client** olup, kaynak sahibinin cihazına kurulur ve çalıştırılır. Bu tür istemcilerde, protokol verileri ve kimlik bilgileri erişilebilir, ancak uygulama tarafından kullanılan dinamik olarak verilen kimlik bilgileri (örneğin, erişim jetonları) kabul edilebilir seviyede korunabilir. Bu tür kimlik bilgileri, kötü niyetli sunuculardan korunduğu gibi, bazen cihazdaki diğer uygulamalardan da korunabilir.

6. Özet

OAuth 2.0, istemcileri **gizli** ve **açık** olarak iki ana kategoride sınıflandırır. **Gizli istemciler**, kimlik bilgilerini güvenli bir şekilde saklayabilirken, **açık istemciler** bunu yapamaz ve genellikle daha yüksek güvenlik önlemleri gerektirir. OAuth 2.0, farklı istemci türleri için güvenlik gereksinimlerini ve uygulama senaryolarını dikkate alarak, her tür için uygun çözümler sunar.

2.2. Client Identifier bölümü, OAuth 2.0 protokolünde istemcinin yetkilendirme sunucusundan aldığı "client identifier" (istemci kimliği) hakkında bilgi verir. Bu kimlik, istemcinin kaydının benzersiz bir temsili olarak kullanılır ve bir dizi önemli özellik taşır.

1. Client Identifier Nedir?

Client identifier, istemcinin kaydını temsil eden benzersiz bir dizidir. Yetkilendirme sunucusu, istemci kaydını doğruladıktan sonra, her istemciye bu tür bir kimlik verir. Bu kimlik, istemcinin **yetkilendirme sunucusu** ile olan ilişkisinin bir parçası olarak kullanılır ve istemcinin kaydını tanımlar.

2. Client Identifier'ın Gizli Olmaması

Client identifier, **gizli bir bilgi değildir**. Bu kimlik, **kaynak sahibine** (resource owner) açık bir şekilde gösterilebilir ve istemci ile kaynak sahibi arasında karşılaşılan herhangi bir yerde görünebilir. Bu nedenle, client identifier yalnızca istemciyi tanımlamak için kullanılır ve **client authentication (istemci kimlik doğrulama)** amacıyla tek başına kullanılamaz.

Örneğin, istemci kimliği genellikle istemci tarafından yapılan isteklerde başlıklar (headers) veya URL parametreleri gibi açık yollarla gönderilebilir, çünkü bu kimlik **gizli** değildir ve her zaman gizlilik veya güvenlik için saklanmaz.

3. Benzersizlik ve Yetkilendirme Sunucusuna Bağlılık

Client identifier, her yetkilendirme sunucusu için **benzersiz** olmalıdır. Yani, bir istemci için atanan kimlik, sadece o yetkilendirme sunucusu tarafından geçerli olacaktır. Aynı istemci, başka bir yetkilendirme sunucusuyla kaydolduğunda, o sunucu ona farklı bir client identifier atayabilir. Bu, farklı sunucuların birbirlerinin istemci kimliklerini tanımayacaklarını ve bağımsız olarak çalışacaklarını garanti eder.

4. Kimlik Boyutu

Client identifier'ın boyutu, bu spesifikasyonda açıkça tanımlanmamıştır. Bu nedenle, istemciler bu kimliğin boyutu hakkında herhangi bir varsayımda bulunmamalıdır. Ancak, **yetkilendirme sunucusu**, istemciye verdiği kimliğin boyutunu **belgelendirmelidir**. Bu, istemcilerin doğru bir şekilde işlem yapabilmesi için önemlidir.

5. Kimlik Doğrulama İçin Kullanılamaz

Client identifier, istemcinin kimliğini belirlemek için kullanılsa da, **istemci doğrulama** için yeterli bir bilgi değildir. Yani, istemci kimliği yalnızca istemcinin kaydını tanımlar ve istemcinin yetkilendirme sunucusuyla güvenli bir şekilde iletişim kurabilmesi için ek güvenlik önlemleri gereklidir (örneğin, istemci şifresi veya başka bir kimlik doğrulama yöntemi).

6. Özet

- **Client identifier**, istemcinin kaydını temsil eden benzersiz bir dizidir ve yalnızca istemcinin tanımlanmasında kullanılır.
- Bu kimlik **gizli değildir**, kaynak sahibi tarafından görülebilir ve istemci kimlik doğrulamasında tek başına kullanılmaz.

- Kimlik, her yetkilendirme sunucusu için benzersizdir ve istemci bu kimlik ile ilişkilendirilen sunucuya bağlıdır.
- Kimlik boyutu belirsizdir ancak yetkilendirme sunucusu bunu **belgelendirmelidir**.

Sonuç olarak, client identifier, istemciyi tanımlamak için kullanılan bir kimliktir ancak tek başına güvenlik için yeterli değildir ve doğrulama amacıyla kullanılmaz.

2.3. Client Authentication bölümü, istemci türü "confidential" (gizli) olan bir istemcinin, yetkilendirme sunucusuyla nasıl kimlik doğrulaması yapması gerektiğini açıklar. Bu bölümde, istemci doğrulaması için kullanılan yöntemlerin güvenlik gereksinimlerine dayalı olarak nasıl yapılandırılacağına dair önemli bilgiler bulunmaktadır.

1. Confidential Client (Gizli İstemci)

Confidential client, istemci kimlik bilgilerini güvenli bir şekilde saklayabilen bir istemci türüdür. Genellikle, istemci uygulaması, bir **sunucu** üzerinde çalışır ve istemci kimlik bilgileri (örneğin, parola, özel anahtar ve kamu anahtar çifti gibi) bu sunucuda güvenli bir şekilde saklanır.

- **Gizli İstemciler için Kimlik Doğrulama Yöntemleri**: Yetkilendirme sunucusu, gizli istemci türü için güvenlik gereksinimlerine uygun bir **kimlik doğrulama yöntemi** belirler. Bu, şifre tabanlı bir kimlik doğrulama, dijital sertifikalar veya kamu/özel anahtar çifti gibi farklı yöntemler olabilir. Yetkilendirme sunucusu, istemcinin güvenli bir şekilde kimliğini doğrulamak için bu kimlik doğrulama yöntemlerini kabul edebilir. Bu, istemcinin kimlik doğrulaması yapılırken sunucu ve istemci arasındaki güvenli iletişimi sağlar.
- **Client Credentials**: Gizli istemciler, genellikle bir **set** (takım) istemci kimlik bilgisi (client credentials) alır. Bu kimlik bilgileri, istemcinin yetkilendirme sunucusuyla güvenli bir şekilde kimliğini doğrulaması için kullanılır. Örneğin, istemci bir **parola** ya da **kamu/özel anahtar çifti** kullanabilir. Bu kimlik bilgileri istemciyle birlikte güvenli bir şekilde saklanır.

2. Public Client (Halka Açık İstemci)

Public client, istemci kimlik bilgilerini güvenli bir şekilde saklayamayan ve genellikle bir kullanıcının cihazında çalışan istemci türüdür. Örneğin, web tarayıcılarında veya mobil cihazlarda çalışan istemciler, istemci kimlik bilgilerini güvenli bir şekilde saklamakta zorluk yaşayabilirler. Bu nedenle, **public client** istemcileri için kimlik doğrulaması yapılmaz.

- Yetkilendirme sunucusu, **public client** istemcileri için kimlik doğrulama yöntemleri belirleyebilir. Ancak, public client kimlik doğrulaması **istemcinin tanımlanması** için kullanılmamalıdır. Yani, **public client** türündeki istemciler, kimlik doğrulama için **client authentication** yöntemlerine dayanmazlar.

3. Birden Fazla Kimlik Doğrulama Yöntemi Kullanılamaz

Bir istemci, aynı istek içinde birden fazla kimlik doğrulama yöntemi kullanmamalıdır. Bu, istemcinin güvenliğini ve iletişimini gereksiz yere karmaşıklaştırabilir ve sunucu için güvenlik açıklarına yol açabilir. Örneğin, istemci aynı istekte hem şifre hem de anahtar doğrulaması kullanamaz. İstemci yalnızca bir kimlik doğrulama yöntemi seçmeli ve bu yöntemi kullanmalıdır.

4. Özet

- **Confidential client** için, yetkilendirme sunucusu güvenlik gereksinimlerine uygun bir kimlik doğrulama yöntemi belirler ve istemci, bu yöntemle kimlik doğrulaması yapar.
- Gizli istemciler, genellikle bir set **client credentials** (kimlik bilgileri) alır ve bu bilgileri güvenli bir şekilde kullanır.
- **Public client** istemcileri, kimlik doğrulama için yalnızca token (erişim belirteci) gibi mekanizmalar kullanır, kimlik doğrulama amacıyla istemci bilgileri kullanılamaz.

- İstemci, her istekte yalnızca **bir** kimlik doğrulama yöntemi kullanabilir.

Bu bölüm, istemcinin, güvenli bir kimlik doğrulama süreciyle yetkilendirme sunucusuyla iletişim kurmasını ve doğru güvenlik gereksinimlerine dayalı olarak işlem yapmasını sağlar.

2.3.1. Client Password bölümü, istemcilerin kimlik doğrulamasını yapmak için **client password** (istemci parolası) kullanma yöntemini açıklar. Bu yöntem, istemcinin kimliğini doğrulamak için **HTTP Basic Authentication** şemasını kullanmayı içerir. Ayrıca, istemci kimlik bilgileriyle birlikte gönderilebilecek alternatif bir yöntem ve güvenlik önlemleri de tartışılmaktadır.

1. HTTP Basic Authentication

İstemciler, bir **client password** (istemci parolası) sahibi olduklarında, yetkilendirme sunucusuyla kimlik doğrulaması yapmak için **HTTP Basic Authentication** şemasını kullanabilirler. Bu şema, RFC 2617'de tanımlanmıştır.

- **Client Password Kullanımı**: İstemci, kendisini tanıtmak ve kimliğini doğrulamak için HTTP isteği içerisinde, **client identifier** (istemci kimlik bilgisi) ve **client password** (istemci parolası) bilgisini içerir. Bu bilgileri, "application/x-www-form-urlencoded" kodlama algoritması ile encode eder.
    - **Client Identifier**: İstemcinin benzersiz kimliğidir.
    - **Client Password**: İstemcinin gizli parolasıdır.

  Bu bilgilerin her biri, HTTP isteğinde uygun şekilde encode edilerek, **Authorization header** (Yetkilendirme başlığı) ile gönderilir. Örnek olarak:

Authorization: Basic czZCaGRSa3F0Mzo3RmpmcDBaQnIxS3REUmJuZlZkbUl3

Burada, `Basic` *anahtar kelimesi, temel kimlik doğrulamasının kullanılacağını belirtir ve ardından client identifier ile client password bilgileri encode edilerek yer alır.*

- **Yetkilendirme Sunucusunun Desteklemesi**: Yetkilendirme sunucusu, client password kullanan istemciler için **HTTP Basic Authentication** şemasını desteklemek zorundadır. Bu, istemcinin kimlik doğrulaması yapabilmesi için gereklidir.

2. Alternatif Kimlik Doğrulama Yöntemi

Yetkilendirme sunucusu, istemci kimlik bilgilerini **request body** (istek gövdesi) içinde de alabilir. Bu, client password yerine, istemci kimlik bilgilerini aşağıdaki parametrelerle göndermeyi içerir:

- **client_id**: İstemciye, kayıt sırasında verilen benzersiz kimlik bilgisi.
- **client_secret**: İstemcinin gizli parolası.

Bu yöntemin kullanımı, özellikle HTTP Basic Authentication şemasını veya başka bir parola tabanlı kimlik doğrulama şemasını doğrudan kullanamayan istemciler için uygundur. Ancak, bu yöntemin **request URI** (istek URI’sı) içinde yer almaması gerektiği belirtiliyor, sadece isteğin gövdesinde kullanılmalıdır.

Örnek kullanım:

POST /token HTTP/1.1
Host: server.example.com
Content-Type: application/x-www-form-urlencoded

grant_type=refresh_token&refresh_token=tGzv3JOkF0XG5Qx2TlKWIA
&client_id=s6BhdRkqt3&client_secret=7Fjfp0ZBr1KtDRbnfVdmIw

3. TLS (Transport Layer Security) Zorunluluğu

Yetkilendirme sunucusu, istemci kimlik doğrulaması için parola tabanlı bir yöntem kullanıldığında, **TLS** (Transport Layer Security) protokolünün kullanılmasını zorunlu kılar. TLS, verilerin güvenli bir şekilde iletilmesini sağlar ve parola gibi hassas bilgilerin güvenliğini artırır. Bu, istemci kimlik bilgileri ile yapılan her istekte **TLS** kullanılması gerektiğini belirtir.

4. Brute Force (Kaba Kuvvet) Saldırılarına Karşı Koruma

Bu kimlik doğrulama yöntemi, **client password** (istemci parolası) içerdiği için, yetkilendirme sunucusu, kaba kuvvet saldırılarına karşı koruma sağlamak zorundadır. Kaba kuvvet saldırıları, kötü niyetli bir saldırganın parolaları tahmin etmeye çalışarak istemci kimlik doğrulamasını kırmaya çalıştığı saldırılardır. Yetkilendirme sunucusu, bu tür saldırılara karşı savunma mekanizmaları eklemelidir.

Özetle:

- **Client Password** yöntemi, istemcilerin **HTTP Basic Authentication** şemasını kullanarak kimlik doğrulaması yapmasına olanak sağlar. Bu şemada, istemci kimlik bilgileri (client identifier ve client password) HTTP isteği içerisinde gönderilir.
- Alternatif olarak, istemci kimlik bilgileri isteğin gövdesinde **client_id** ve **client_secret** parametreleri olarak da gönderilebilir, ancak bu yöntem genellikle önerilmez.
- **TLS** kullanımı zorunludur; bu, istemci parolasının güvenli bir şekilde iletilmesi için gereklidir.
- Ayrıca, sunucunun kaba kuvvet saldırılarına karşı savunmasız olmaması için gerekli güvenlik önlemleri alması gerekir.

Bu bölüm, istemcilerin güvenli bir şekilde kimlik doğrulaması yapmalarını sağlamak için önemli bilgiler ve güvenlik önlemleri sunar.

2.3.2. Other Authentication Methods *bölümü, yetkilendirme sunucularının istemci kimlik doğrulaması için* ***HTTP authentication*** *şemalarını esnek bir şekilde kullanabilmelerini açıklamaktadır. Bu bölümde, yetkilendirme sunucusunun* ***HTTP Basic Authentication*** *dışında başka kimlik doğrulama yöntemlerini de destekleyebileceği ve bu yöntemlerin nasıl uygulanması gerektiği belirtiliyor.*

1. Diğer Kimlik Doğrulama Yöntemleri (Other Authentication Methods)

Yetkilendirme sunucusu, **HTTP authentication** (HTTP kimlik doğrulama) için gereksinimlerini karşılayan herhangi bir uygun kimlik doğrulama şemasını destekleyebilir. Yani, sadece **HTTP Basic Authentication** değil, başka kimlik doğrulama yöntemleri de kullanılabilir.

MAY ifadesi, yetkilendirme sunucusunun ihtiyaca göre farklı kimlik doğrulama şemalarını kullanabileceği, ancak bunun zorunlu olmadığı anlamına gelir. Örneğin:

- **OAuth 2.0 Bearer Token** gibi daha modern ve güvenli kimlik doğrulama yöntemleri,
- **OAuth 2.0 Client Assertion** (istemci iddiaları) gibi alternatifler,
- **OAuth 2.0 JWT (JSON Web Token)** gibi teknolojiler,
- Diğer özel şemalar (örneğin, şirketin iç güvenlik standartlarına uygun özel şemalar) kullanılabilir.

2. İstemci Kimliği ve Kimlik Doğrulama Yöntemi Arasındaki Eşleştirme

Eğer yetkilendirme sunucusu, **diğer kimlik doğrulama yöntemlerini** kullanıyorsa, bu durumda **client identifier** (istemci kimliği) ile belirli bir kimlik doğrulama şeması arasında açık bir eşleştirme yapılması gerekmektedir. Yani, her istemci kaydı, kullanılan kimlik doğrulama şemasıyla ilişkilendirilmelidir.

- Örneğin, bir istemci kayıtlıysa ve bu istemci özel bir kimlik doğrulama yöntemi kullanacaksa, yetkilendirme sunucusu, bu istemci kaydını o yönteme uygun şekilde eşleştirmelidir.
- Bu eşleştirme, istemci kayıtlarında belirtilmeli ve yetkilendirme sunucusu bu eşleştirmeyi kullanarak hangi kimlik doğrulama şemasının uygulanacağını belirlemelidir.

3. Kimlik Doğrulama Yöntemi Belirlenmesi

- Yetkilendirme sunucusu, desteklediği kimlik doğrulama şemalarının her birinin güvenlik gereksinimlerini yerine getirdiğinden emin olmalıdır.
- Her kimlik doğrulama şeması, istemcinin güvenliğini sağlamak için **kimlik doğrulama ve yetkilendirme** süreçlerinde gereksinimlere uygun olmalıdır. Örneğin, bazı kimlik doğrulama yöntemleri, güvenli olmayan ağlarda kullanılmaya uygun olmayabilir ve bu nedenle sadece TLS (Transport Layer Security) ile korunmuş bağlantılarla sınırlı olmalıdır.

4. Eşleştirme Örneği

Bir istemci kaydında, istemci kimliği **client_id** belirtilir. Bununla birlikte, bu istemciye uygulanacak kimlik doğrulama şeması da tanımlanmalıdır. Örneğin:

- Eğer istemci, **OAuth 2.0 Client Assertion** kullanarak kimlik doğrulaması yapacaksa, yetkilendirme sunucusu bu istemcinin kaydını **Client Assertion** ile ilişkilendirebilir.
- Eğer istemci **Basic Authentication** kullanacaksa, bu kimlik doğrulama şeması da ilgili kayıtta belirtilir.

5. Özet

- **Yetkilendirme sunucusu**, **HTTP authentication** için yalnızca temel kimlik doğrulama şemaları değil, güvenlik gereksinimlerine uygun herhangi bir şemayı da destekleyebilir.
- Bu şemaların kullanılması durumunda, **client identifier** (istemci kimliği) ile ilgili kimlik doğrulama yöntemi arasında açık bir eşleştirme yapılmalıdır.
- Yetkilendirme sunucusunun, her kimlik doğrulama yönteminin güvenlik gereksinimlerini doğru şekilde yerine getirdiğinden emin olması gerekir.

Bu bölüm, OAuth 2.0 protokolünü daha esnek ve geniş bir uygulama yelpazesinde kullanılabilir kılmak için alternatif kimlik doğrulama yöntemlerine imkan tanır.

2.4. Unregistered Clients bölümü, OAuth 2.0 protokolünde, **kayıtsız istemciler** (unregistered clients) kullanımının mümkün olduğunu belirtmektedir, ancak bunun belirli kısıtlamaları ve ek gereksinimleri olduğu açıklanmaktadır. Şimdi bu kısmı detaylı olarak inceleyelim:

1. Kayıtsız İstemciler (Unregistered Clients)

- **Kayıtsız istemciler**, OAuth 2.0 protokolüne kaydedilmemiş istemcilerdir. Normalde OAuth 2.0'da, bir istemcinin çalışabilmesi için, istemcinin **yetkilendirme sunucusuna** kaydedilmesi gerekir. Ancak bu bölümde, OAuth 2.0 spesifikasyonu, kayıtsız istemcilerin de kullanılabileceğini kabul etmektedir.
- Kayıtsız istemciler, genellikle kimlik doğrulama veya yetkilendirme sürecine dahil olan, ancak **istemci kaydı** yapılmayan uygulamalardır. Bu, belirli durumlarda uygulamalara bazı esneklikler sağlayabilir, ancak protokolde bu kullanım için herhangi bir standart önerisi veya gereksinimi yoktur.

2. Kayıtsız İstemciler Kullanımının Kısıtlamaları

- **Kayıtsız istemcilerin kullanımı**, **bu spesifikasyonun kapsamı dışındadır**. Yani OAuth 2.0 protokolü, kayıtsız istemcilerin kullanımını desteklemez veya standart hale getirmez. Kayıtsız istemcilerin kullanımı, protokole dahil edilmediği için bunun nasıl yapılacağı, güvenlik gereksinimleri ve uygulama uyumluluğu konusunda **ek güvenlik analizleri** ve **değerlendirmeler** yapılması gerekmektedir.
- Kayıtsız istemcilerin kullanımı, **güvenlik riskleri** ve **interoperabilite** (uyumluluk) sorunlarına yol açabilir. Özellikle, istemci kimlik doğrulama sürecinde kaydın yapılmaması, **kimlik doğrulama ve yetkilendirme işlemlerinin** daha az güvenli olmasına sebep olabilir. Bu nedenle, kayıtsız istemcilerin kullanımı, dikkatli bir şekilde gözden geçirilmeli ve **ek güvenlik önlemleri** alınmalıdır.

3. Ek Güvenlik İncelemesi ve Uyumluluk

- Kayıtsız istemciler, OAuth 2.0 protokolünde yer almadığı için, güvenlik açısından **ilave analizler** gerektirir. Çünkü kayıtsız istemcilerin yetkilendirme süreci, protokolle uyumlu olmayabilir ve istemci ile yetkilendirme sunucusu arasındaki güven ilişkisi zayıflayabilir.
- Ayrıca, kayıtsız istemcilerin kullanımı, OAuth 2.0 spesifikasyonunun diğer taraflarıyla uyumsuzluk yaratabilir. Bu nedenle, kayıtsız istemcilerle ilgili uygulanabilirlik ve güvenlik değerlendirmelerinin yapılması gerekmektedir.

4. Özet

- **Kayıtsız istemciler** OAuth 2.0'da **resmi olarak desteklenmemektedir**, ancak kullanımları mümkündür.
- Kayıtsız istemcilerin kullanımı **güvenlik riskleri** ve **uyumluluk sorunları** oluşturabilir, bu nedenle dikkatli bir şekilde incelenmesi ve ek güvenlik önlemlerinin alınması gerekmektedir.
- Spesifikasyon, kayıtsız istemciler için özel bir düzenleme veya standart sunmadığı için, bu tür istemcilerin kullanımı, protokolle uyumsuzluk oluşturabilir.

Bu bölüm, **OAuth 2.0 protokolü** için güvenlik açısından belirli bir **koruma sağlamak** amacıyla istemcilerin kaydını **gereklilik** olarak belirlemiştir. Ancak kayıtsız istemciler için bazı durumlar dışında, belirli kullanım senaryolarında bu tür istemcilerin kullanılabileceği belirtiliyor.