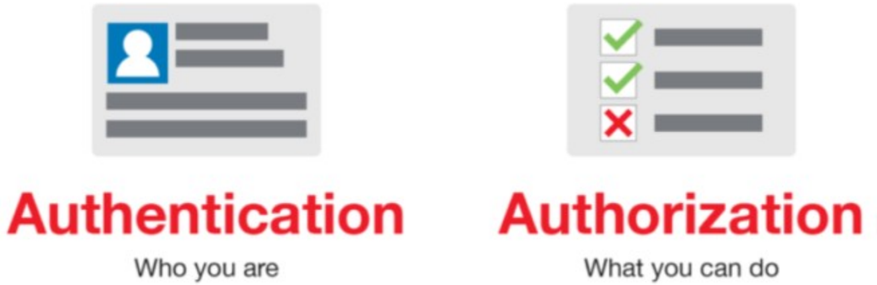

# Authentication/Authorization

## Bootcamp 4. Hafta

# Authentication/Authorization

## Authentication (Kimlik Doğrulama)
## Authorization (Yetkilendirme)




ituacm

# Authentication (Kimlik Doğrulama)

**Kullanıcı adı, eposta, ID gibi kullanıcıya özgü *public* bir bilgiyle password, passkey, passphrase gibi sadece onun bilebileceği *private* bilgiyi eşleştirerek hesaba girişte kullanılır.**

# Authorization (Yetkilendirme)

**Giriş sonrasında hesap oturumunun neleri yapabileceği.**

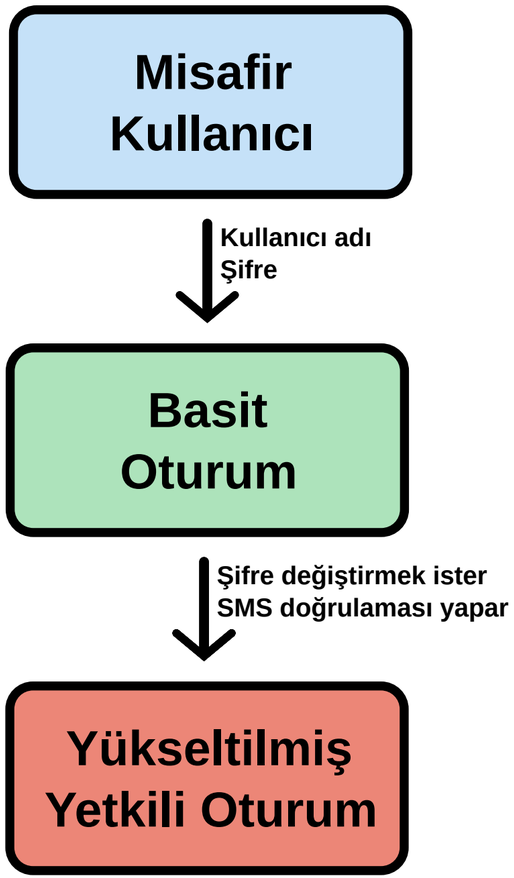
**Foreshadowing:**
**– oturum**
**– oturum yetkileri**
**– yetki yükseltme**

ituacm

# Foreshadowing: Oturum Yetkileri

**Kullanıcı giriş yaptığı anda hesabındaki her bilgiyi düzenleyebilecek bir oturum açılmaz. Mesela eposta değiştirmesi gerektiğinde “basit oturum”da ona verilmemiş “eposta değiştirme” yetkisini edinmelidir.**

# Foreshadowing: Yetki Yükseltme




**Bunları yapmak için oturumları (authentication) ve yetkilerini (authorization) bir yerde saklamamız, gerektiğinde güncelleyebilmemiz gerekiyor...**

ituacm

# Oturum: Siteler Kullanıcıyı Nasıl Hatırlar?

**Oturumların saklanmasında kullanılan yöntemleri iki çeşide ayırmak mümkündür:**

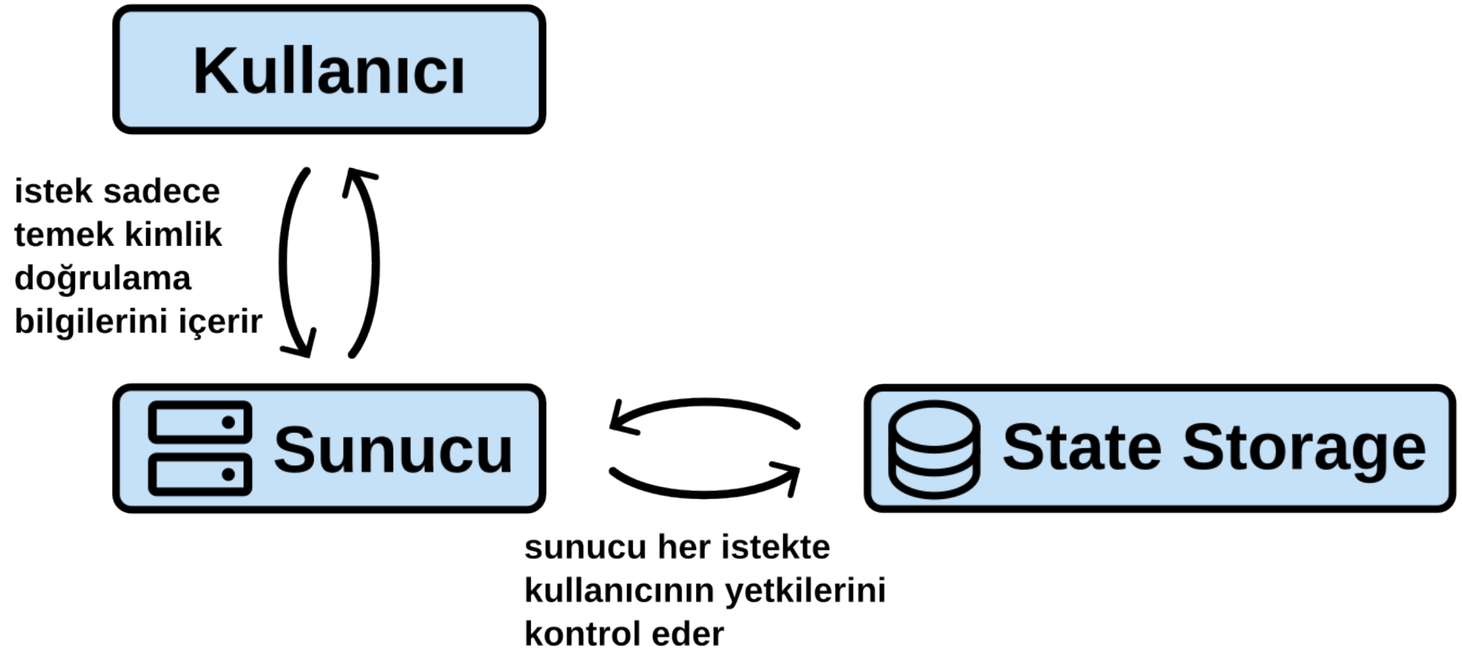
**- stateful**

**- stateless**

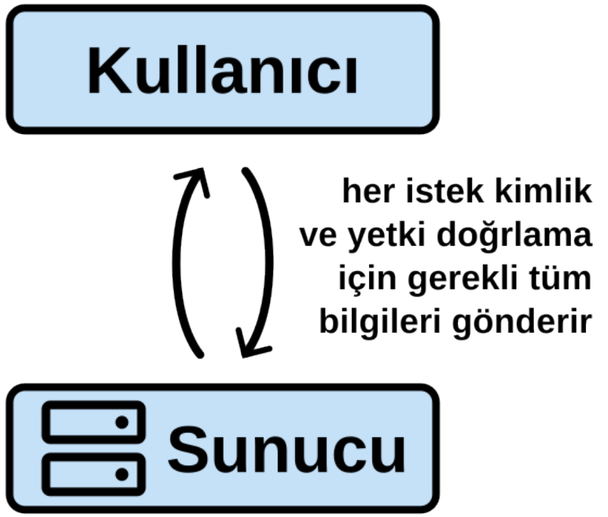
ituacm

# Oturum

## Stateful Application




## Stateless Application



# Stateful Architecture

Sunucu session bilgisini kendi tarafında tutar.

Kullanıcı giriş yaptığında sessionId oluşturulur, cookie ile client’a gönderilir.

Her istekte bu cookie sunucuya geri gönderilir.

Sunucu memory veya DB’de session verisini kontrol eder.

ituacm

# Stateless Architecture

**Mesela kimlik doğrulamada site şifre ve kullanıcı adı kullansın. Bunda herhangi bir sahtekârlık yapılamaz, şifre doğruysa doğrudur, değilse değildir.**

**Peki yetkilerini kullanıcı gönderiyorsa yalan söylemediğini nasıl bileceğiz?**

ituacm

# Dijital İmzalar

**Token içindeki bilgilerin değiştirilmediğini garanti eder.**

**Sunucu, token’ı gizli anahtarla imzalar.**

**Client sahte veri yazsa bile imza tutmaz.**

# Basitçe Hash

**Hash = Veriden tek yönlü özet çıkarma işlemi.**

**Basit düşünelim:**
**basit-hash(x, secret) = (x + 37) % secret**

**Girdi (x) değişince sonuç değişir.**
**Tek yönlüdür → Hash değerinden x'i bulmak kolay değildir.**

ituacm

# HMAC

**Mesaj + Gizli Anahtar → Hash**

**Amaç: Mesajın bütünlüğünü ve kaynağını doğrulamak.**

**basit-hash(mesaj, secret) → İmza**

**Bu imza ve mesajı kullanıcıya gönderip yetki doğruluyoruz. Secret sunucuda kaldığından sadece sunucu imza üretebiliyor**

ituacm

# JWT

**Token 3 parçadan oluşur:**

- **Header → algoritma, tip (HS256, RS256)**
- **Payload → kullanıcı bilgisi (ör: id, email, role)**
- **Signature → imza**

**Token client’a verilir, client her istekte sunucuya gönderir. Sunucu imzayı kontrol ederek token’ın değişmediğini anlar.**

ituacm