

## 1-Birbirinden bağımsız iki platformun birbiriyle haberleşmesi:

Örneğin:

Biri Java ile geliştirilmiş bir platform (X) ve diğeri C# ile geliştirilmiş bir platform (Y) var. Bu iki platform, birbiriyle haberleşerek request-response ilişkisi kurmalıdır. Söz konusu iletişimi de aynı zamanda güvenlikde sağlanmalıdır.

## 2- Yöntemler:

iki platformun birbiriyle iletişim kurmasını sağlamak için aşağıdaki yöntemler kullanılmaktadır.

### a- RESTful Web Servisleri (REST API):

REST (Representational State Transfer), platformlar arası iletişim için kullanılan kolay ve yaygın bir yöntemdir.

Java ile yazılmış X platformu, bir REST API sunucusu olarak çalışır.

C# ile yazılmış Y platformu, bu REST API'ye HTTP üzerinden istek gönderir (GET, POST, PUT, DELETE gibi).

JSON veya XML formatında veri paylaşımı yapılır.

Bu yöntem hızlıdır ve kolayca uygulanabilir. JSON formatı sayesinde insan tarafından okunabilir.

Örnek:

- X platformunda bir REST API tanımlanır.

GET /api/product/103

```
{  
  "id": 103,  
  "isim": "bilgisayar",  
  "fiyat": 1000  
}
```

- Y platformu bu endpoint'i çağırarak ürün bilgisine ulaşır.

### b- SOAP (Simple Object Access Protocol):

SOAP, platformlar arasında XML tabanlı mesajlaşma için kullanılan bir protokoldür.

Java platformu bir SOAP servisi sunar. C# platformu, bu servise WSDL (Web Service Description Language) dosyası ile bağlanır.

Standart bir protokoldür ve yüksek güvenlik gereksinimlerini karşılar.  
SOAP mesajları, WS-Security ile şifrelenebilir.  
REST'e kıyasla daha karmaşıktır.

Örnek:

SOAP mesajı örneği:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <getProduct>
      <id>103</id>
    </getProduct>
  </soapenv:Body>
</soapenv:Envelope>
```

c- Mesaj Kuyrukları (Message Queues):

Mesajlaşma protokolleri (ör. RabbitMQ, Kafka) kullanılarak iki platform arasında iletişim sağlanabilir.

X platformu bir mesaj kuyruğuna (queue) mesaj gönderir.

Y platformu bu mesajı kuyruğa abone olarak alır.

Asenkron iletişim sağlar (eş zamanlılık gerekmez).

Büyük hacimli veriler için uygundur.

Örnek:

X platformu, RabbitMQ'ya bir mesaj gönderir:

```
{
  "event": "OrderCreated",
  "orderId": 103
}
```

Y platformu bu mesajı alarak işleme koyar.

ç- gRPC (Google Remote Procedure Call):

gRPC, platformlar arasında hızlı ve etkili bir iletişim sağlar.

Java ve C# arasında, gRPC ile oluşturulmuş protokoller üzerinden ikili (binary) formatta

veri aktarımı yapılır.

Performansı yüksektir.

İkili format (binary serialization) sayesinde veri boyutunu azaltır.

REST'e kıyasla daha karmaşıktır.

Örnek:

Bir ürün için gRPC Protobuf dosyası:

```
syntax = "proto3";  
service ProductService {  
  rpc GetProduct(ProductRequest) returns (ProductResponse);  
}
```

### 3-"Güvenliğin Sağlanması"

#### 1- HTTPS Kullanımı:

Tüm iletişim kanallarında HTTPS protokolü kullanılarak veri aktarımı şifrelenmelidir.

Veri aktarımı sırasında SSL/TLS sertifikaları kullanılarak güvenli bir kanal oluşturulur.

Örnek: <https://api.example.com/product/103>

#### 2- API Anahtarı (API Key):

REST API'lerde, istemcilerin (client) kimliğini doğrulamak için API anahtarları kullanılabilir. Her istemciye özel bir anahtar atanır.

#### 3- OAuth 2.0

OAuth 2.0, yetkilendirme için kullanılan modern bir standarttır.

X platformu bir OAuth token üretir ve bu token Y platformunda kullanılarak erişim sağlanır.

Örnek:

HTTP isteğine bir yetkilendirme başlığı eklenir:

Authorization: Bearer <access\_token>

#### 4- Veri Şifreleme

Mesajlar, transfer sırasında şifrelenerek üçüncü tarafların erişimi engellenir.

Örnek: AES veya RSA algoritmaları kullanılabilir.

### 4-Yorum:

Birbirinden bağımsız iki platformun iletişiminde kullanılan yöntemler, gereksinimlere ve sistemlerin karmaşıklığına göre seçilmelidir. Eğer hızlı bir çözüm gerekiyorsa REST API en uygun yöntemdir. Ancak yüksek güvenlik ve standart bir yapı gerekiyorsa SOAP tercih edilebilir.

Mesajlaşma yoğun bir senaryoda RabbitMQ veya Kafka kullanılabilir. Son olarak, güvenliği sağlamak için HTTPS, OAuth 2.0 ve şifreleme yöntemleri mutlaka uygulanmalıdır.

#### 5- Kaynakça:

REST API Dokümantasyonu  
SOAP vs REST: IBM Dokümantasyonu  
RabbitMQ Official Dokümantasyonu  
gRPC Dokümantasyonu  
OAuth 2.0 Official Dokümantasyonu.