



**SOAP'tan
REST'e, REST'ten
gRPC'ye**

Ben Kimim?

- Türksat Uydu Haberleşme Kablo TV ve İşletme A.Ş - 2017 -
 - Senior Software Developer
- Ankara Üniversitesi - Yüksek Lisans - 2018 - 2022
- Fırat Üniversitesi - Lisans - 2012 - 2016



umiitkose



umiitkose



umiitkose

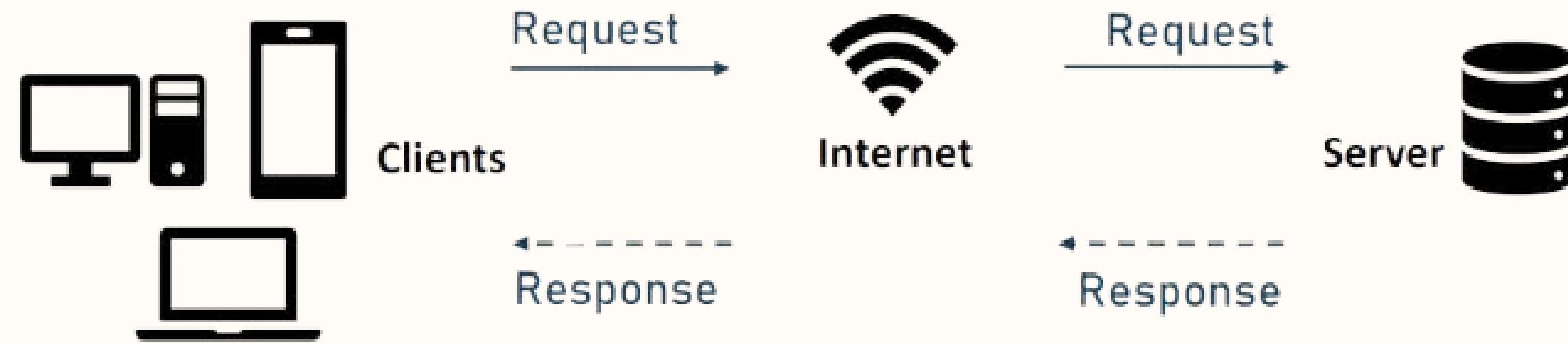


<https://umiitkose.com>

SOAP'tan REST'e, REST'ten gRPC'ye

01	SOAP	05	REST	09	gRPC
02	XML	06	HTTP Detayları	10	Protocol Buffers
03	WSDL	07	Örnek REST	11	HTTP 2.0
04	Örnek SOAP	08	SOAP vs REST	12	Örnek gRPC

SOAP vs REST vs gRPC



Web Servis Nedir?

Platform bağımsız, uygulama veya cihazların http protokolu üzerinden iletişim kurmalarını sağlayan yapılardır.

SOAP (Simple Object Access Protocol)

XML mesajları web uygulamaları üzerinden birbirleriyle haberleşirmeye yarayan mesajlaşma protokolüdür.

1

SOAP web servisler SOAP ve WSDL ihtiyaç duyar.

2

HTTP POST metotunu kullanır.

3

SOAP, mesaj göndermek için bir formattır.

4

SOAP, Internet üzerinden iletişim kurmak amacıyla tasarlanmıştır.

5

SOAP, platformdan ve programlama dilinden bağımsızdır.

6

SOAP W3C standartıdır.

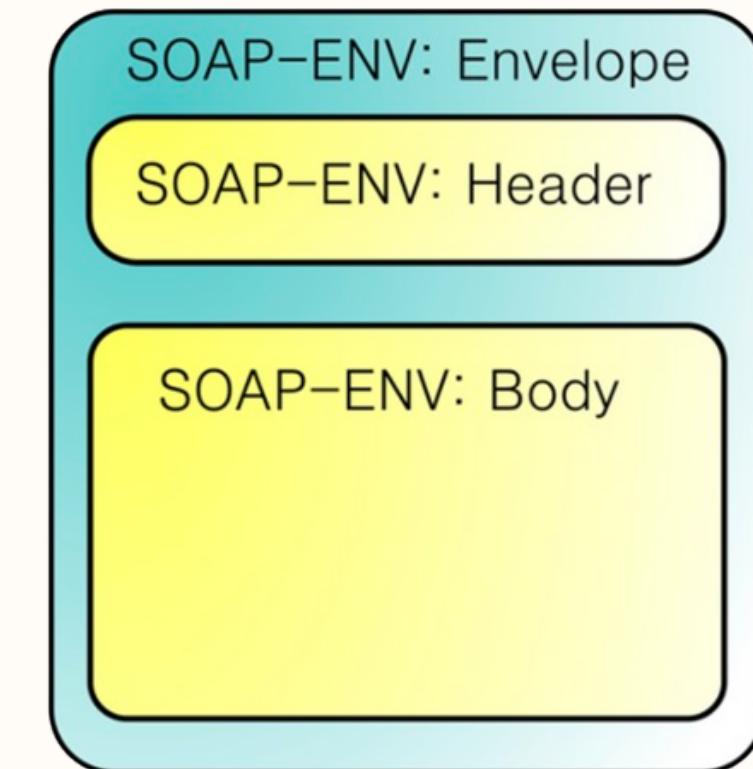
Soap 1.1 <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/>

Soap 1.2 <https://www.w3.org/TR/soap12/>

XML ve SOAP

```
| <note>
|   <to>Herkese</to>
|   <from>Türkiye Java Community</from>
|   <mesaj>Hoşgeldiniz</mesaj>
| </note>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:cal="https://umiitkose.com/calculator">
<!-- <soap:Header/> -->
<soap:Body>
  <cal:topla1>
    <firstNumber>gero et</firstNumber>
    <secondNumber>sonoras imperio</secondNumber>
  </cal:topla1>
</soap:Body>
</soap:Envelope>
```



WSDL (Web Services Description Language)

1

WSDL, soap web servisleri kullanmak için tanımlanmış dildir.

2

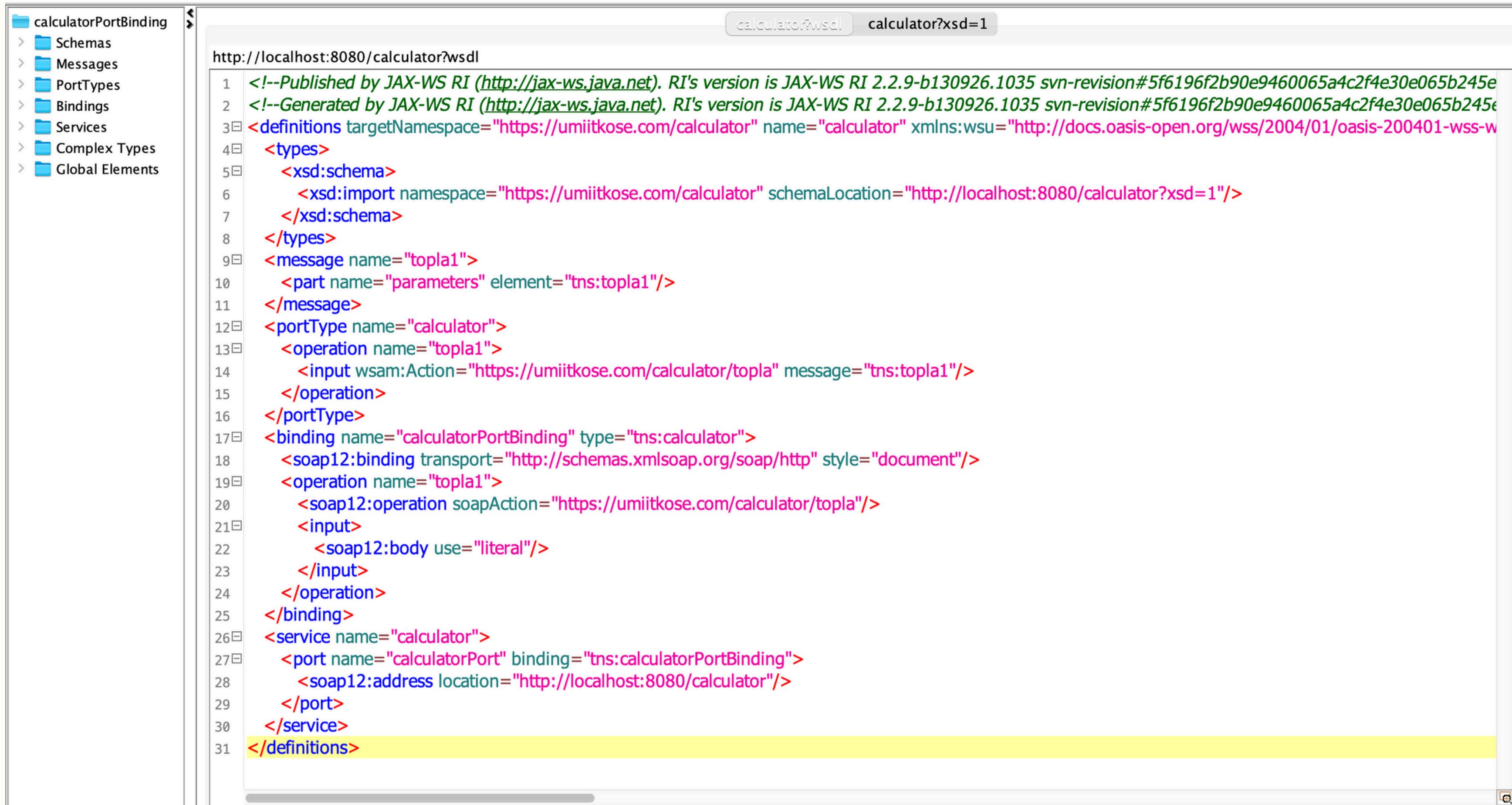
Web servisin dökümanıdır.
WSDL, XML ile yazılmıştır.

3

İçerisinde SOAP mesajın nasıl olacağı, ağı ve port adresleri gibi bilgileri tanımlar.

<https://www.w3.org/TR/2001/NOTE-wsdl-20010315>

WSDL (Web Services Description Language)



The screenshot shows a WSDL editor interface. On the left is a tree view of the service structure:

- calculatorPortBinding
 - Schemas
 - Messages
 - PortTypes
 - Bindings
 - Services
 - Complex Types
 - Global Elements

The main area displays the WSDL XML code. The URL in the address bar is `http://localhost:8080/calculator?wsdl`. The XML code defines a service named "calculator" with a port type "calculator" and a binding "calculatorPortBinding". The binding uses the "document" style and maps to the "topla1" operation, which takes a "parameters" part. The service has a single port "calculatorPort" with the specified binding.

```
<!--Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e-->
<!--Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e-->
<definitions targetNamespace="https://umiitkose.com/calculator" name="calculator" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsam="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:https://umiitkose.com/calculator" xmlns:wsse="http://schemas.xmlsoap.org/ws/2004/09/ws-security-enhiments" xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy" xmlns:wsaw="http://schemas.xmlsoap.org/ws/2004/08/addressing/wsaction" xmlns:wsinf="http://schemas.xmlsoap.org/ws/2004/09/information" xmlns:wsdl11="http://schemas.xmlsoap.org/wsdl/1.1/" xmlns:wsdl11p="http://schemas.xmlsoap.org/wsdl/1.1/part">1 <!--Published by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e-->
2 <!--Generated by JAX-WS RI (http://jax-ws.java.net). RI's version is JAX-WS RI 2.2.9-b130926.1035 svn-revision#5f6196f2b90e9460065a4c2f4e30e065b245e-->
3 <definitions targetNamespace="https://umiitkose.com/calculator" name="calculator" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsam="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="urn:https://umiitkose.com/calculator" xmlns:wsse="http://schemas.xmlsoap.org/ws/2004/09/ws-security-enhiments" xmlns:wsap="http://schemas.xmlsoap.org/ws/2004/08/addressing/policy" xmlns:wsaw="http://schemas.xmlsoap.org/ws/2004/08/addressing/wsaction" xmlns:wsinf="http://schemas.xmlsoap.org/ws/2004/09/information" xmlns:wsdl11="http://schemas.xmlsoap.org/wsdl/1.1/" xmlns:wsdl11p="http://schemas.xmlsoap.org/wsdl/1.1/part">
4   <types>
5     <xsd:schema>
6       <xsd:import namespace="https://umiitkose.com/calculator" schemaLocation="http://localhost:8080/calculator?xsd=1"/>
7     </xsd:schema>
8   </types>
9   <message name="topla1">
10    <part name="parameters" element="tns:topla1"/>
11  </message>
12 <portType name="calculator">
13   <operation name="topla1">
14     <input wsam:Action="https://umiitkose.com/calculator/topla" message="tns:topla1"/>
15   </operation>
16 </portType>
17 <binding name="calculatorPortBinding" type="tns:calculator">
18   <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
19   <operation name="topla1">
20     <soap12:operation soapAction="https://umiitkose.com/calculator/topla" />
21   <input>
22     <soap12:body use="literal"/>
23   </input>
24   </operation>
25 </binding>
26 <service name="calculator">
27   <port name="calculatorPort" binding="tns:calculatorPortBinding">
28     <soap12:address location="http://localhost:8080/calculator"/>
29   </port>
30 </service>
31 </definitions>
```

WSDL

1. Definitions

WSDL Dökümanındaki root elementtir.

2.Types

Web service kullanılabacak veriyi tanımlar

```
<xs:complexType name="person">
  <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="name" type="xs:string" minOccurs="0"/>
    <xs:element name="surname" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

4. Port type

Message elementi ile eşleşen, web servisin işlevini anlatır.

3. Message

Web serviste tanımlı metotları ifade eder.

5. Binding

Web servise ait protokol ve çıktı birimlerini söyler.

6. Service

web servis adresini gösterir.

JAX-WS

Annotation

- @WebService = Web servis tanımı
- @BindingType = SOAP versiyon için
- @OneWay = İletişim tek yönlü mü olsun
- @WebMethod = metot tanımı
- @WebParam = parametre tanımı

Kod Generate (wsimport)

wsimport -keep -verbose –s src wsdl-adresi

- -keep java kodları oluşturmak için
- -s yol belirtmek için
- -verbose ile compiler mesajı görülür.
- -p ile dosyalarımızın ait olduğu paket bilgisi verilebilir.

DEMO

SOAPUI

SOAP ve REST web servisleri test etmemiz için geliştirilmiş JAVA tabanlı açık kaynak bir uygulamadır.

Örnek

Sizde SOAPUI kullanarak SOAP web servisleri test etmek isterseniz, internette yayınlanmış aşağıdaki WSDL 'ı kullanabilirsiniz.

<http://www.dneonline.com/calculator.asmx?wsdl>

REST (Representational state transfer)

Client-server haberleşmesinde HTTP protokolüne ait metod ve kodları kullanarak Json, xml, txt vb formatlarda veri alışverişiini sağlayan mimaridir.

1

2000 yılında Roy Fielding tarafından doktora teziyle başlamıştır.

2

HTTP tüm metodlarını kullanabilir.

3

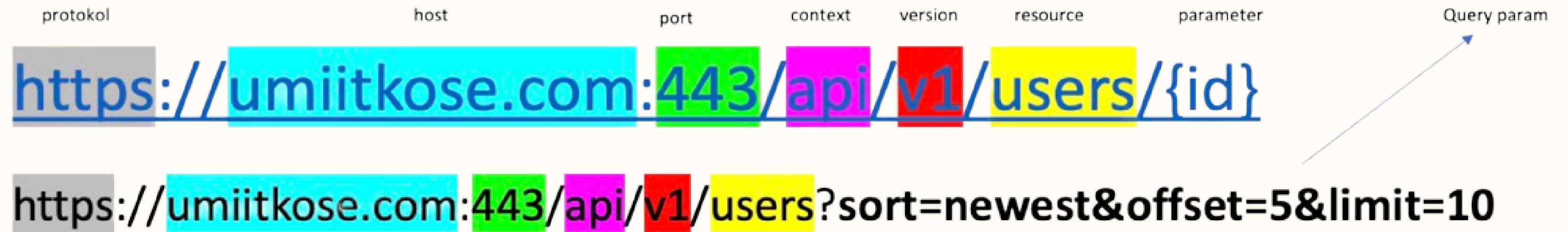
Platform ve programlama dillerinden bağımsızdır.

4

Farklı formatlarda veri tiplerini destekler.

Tez Çalışması :

https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf



protokol: Cihazların birbirleriyle iletişimini sağlayan kurallar dizisi

host: Hizmet sağlayan makine

domain: Web sitesinin internetteki adı, adresidir.

port: Bağlantı noktalarını belirtmek için kullanılan numaralardır.

XML

```
<kisiler>
  <kisi>
    <isim>Ümit</isim>
    <soyisim>KÖSE</soyisim>
    <yas>31</yas>
  </kisi>
</kisiler>
```

JSON

```
{
  "kisiler": {
    "kisi": {
      "isim": "Ümit",
      "soyisim": "KÖSE",
      "yas": 31
    }
  }
}
```

HTTP

İnternet üzerinde bilgi iletmeyi sağlayan bir protokoldür.

- <https://datatracker.ietf.org/doc/html/rfc1945>
- <https://datatracker.ietf.org/doc/html/rfc2616>
- <https://datatracker.ietf.org/doc/html/rfc9113>
- <https://datatracker.ietf.org/doc/rfc9114/>



HTTP METOTLARI

Metot	Örnek URL	Açıklama
GET	<code>http://localhost:8080/users</code>	Verileri getirmek için kullanılır.
POST	<code>http://localhost:8080/users</code>	Veri oluşturmak için kullanılır.
PUT	<code>http://localhost:8080/users/{id}</code>	Verileri güncellemek için kullanılır.
DELETE	<code>http://localhost:8080/users/{id}</code>	Verileri silmek için kullanılır.
Patch	<code>http://localhost:8080/users/{id}</code>	Belli olan veriyi güncellemek istediğimizde kullanılır.
Options	<code>http://localhost:8080/users</code>	İstek atılan endpoint için hangi metotların kullanılacağını gösterir.

HTTP STATUS KODLARI

Kod	Açıklama
100~200	Bilgi
200~300	Başarı
300~400	Yönlendirme
400~500	İstemci Hataları
500~	Sunucu Hataları

HTTP 200 & 300

Kod	Url	Açıklama
200 OK	GET /users	İşlem Sorunsuz gerçekleşti.
201 CREATED	POST /users	Başarıyla yeni resource oluşturuldu.
202 ACCEPTED	POST /roles	Sunucu İsteği Kabul etti, işleme alacak.
204 NO CONTENT	DELETE /users/{id}	Resource silindi.

Kod	Açıklama
301 Moved Permanently	Kalıcı Taşındı.
302 Moved Temporarily	Geçici Taşındı

HTTP 400

Kod	Url	Açıklama
400 Bad Request	POST /users	Eksik istek.
401 Unauthorized	POST /users	Yetki hatası.
403 Forbidden	DELETE /users	Yasaklandı.
404 Not Found	GET /users	Resource Bulunamadı.
405 Method Not Allowed	POST /users	İzin verilmeyen Metod
415 Unsupported Media Type	POST /users	Desteklenmeyen Media Türü

HTTP 500

Kod	Açıklama
500 Internal Server Error	Internal Error.
502 Bad Gateway	Geçersiz Ağ Geçidi
503 Service Unavailable	Servis kapalı
504 Gateway Timeout	Timeout hatası.

HTTP REQUEST HEADER

User-agent:

Burada verilen bilgilere göre uygun içeriği göndereceği istemci özelliklerini belirtir. Sunucular, farklı tarayıcılar ve cihazlar için içerik sunma veya uygun bir yanıt döndürme amacıyla bu bilgiyi kullanır.

Accept

Tarayıcının kabul edebileceği içerik türlerini belirtir. Örneğin, "text/html" HTML içerik türünü kabul ettiğini gösterir.

Accept: text/html

Accept-Language

İstemcinin dil tercihini belirtmek için kullanılır. Sunucu, içeriği bu tercihlere göre sunar. Bu HTTP header sayesinde birden fazla dil tercihiniz varsa virgülle ayırarak belirtebilirsiniz.

tr-TR,tr;q=0.9,en-US;q=0.8,en;q=0.7

Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.107 Safari/537.36

HTTP REQUEST HEADER

Content-type

İçeriğin türünü belirtir.

Content-type:
application/json

Authorization

Kimlik doğrulama için
kullanılır ve yetkilendirme
bilgilerini içerir. Genellikle,
kullanıcı adı ve şifre gibi
kimlik doğrulama
mekanizmalarında
kullanılır.
Bearer x.x.x

ve Dahası

Content-Length
Location
Cookie
Cache-Control

HTTP RESPONSE HEADER

Content-Type

Yanıtın içeriğinin türünü belirtir.

- text/html: HTML içeriği
- application/json: JSON
- image/jpeg: JPEG

Location

Yönlendirme durumunda (örneğin 301 veya 302 durum kodlarıyla) kullanılması gereken yeni kaynağın URLini belirtir.

ve Dahası

Content-Length
Cache Control
Set-Cookie
Date
Content-Encoding

Location:
<https://www.example.com/new-page>

DOKUMANTASYON?



Swagger Rest web servisler için dökümantasyon hazırlamanın bir yoludur.

1

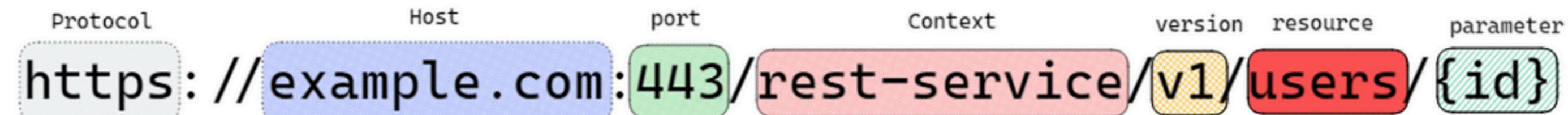
Resource fiil değil isim olmalıdır. (users |
getUser) Çokul isimler kullanılmalı.

2

URI bittiğinde / olarak sonu bırakılmamalıdır.
İsimlendirmede küçük harfler kullanılmalıdır.

3

İsimlendirme birden fazla kelimedен
oluşuyorsa – ile ayrılmalıdır. home-adress



GÜVENLİK

API için belirli güvenlik önlemleri sağlanması isteyebilirsiniz. Örneğin dakika da ne kadar istek atılabilir, hangi kullanıcılar istek atabilir, hangi yetkiye sahip kullanıcılar servisi kullanabilir gibi.

1

Rate Limit :

Servise belirli bir süre içerisinde atılacak istek sayısını belirleyen yapıdır.
“Her 1 dakika 60 istek”

2

Yetki:

API KEY, JWT, Basic Authorization gibi belirli güvenlik önlemleri alınarak servise istek yapılmaktadır.

3

HTTPS:

HTTP protokolünün güvenli halidir, TLS katmanıyla şifreleme ve güvenlik sağlar.

4

Validasyon:

Veriyi validate ederek servisi daha güvenli hale getirebiliriz.

DEMO - SOAP & REST

REST Web Service	Soap
REST mimarı'dır.	SOAP protokoldür.
REST SOAP gibi çok fazla standart tanımlamaz.	SOAP standartlarına uymaktadır
REST, SOAP'tan daha az bant genişliği ve kaynak gerektirir.	SOAP, REST'den daha fazla bant genişliği ve kaynak gerektirir.
Http'nin güvenliğini kullanır.	Kendi Güvenlik önlemleri vardır.
REST birden fazla veri formatına izin verir. Plain text, HTML, XML, JSON etc.	SOAP XML'e olanak sağlar
HTTP tüm metotlarıyla çalışır.	Http POST metotıyla çalışır.
WSDL'a ihtiyaç duymaz	WSDL'a ihtiyaç duyar

JSON ve RESTFUL SIKINTISI

1. Veri Tipleri

Farklı diller arasında json uyumsuzluğu olabiliyor.

Örneğin: Number

4. Yüksek boyutlu veri

Örneğin false 1 değerinin 1 olması gerekikten, 5 byte iletilmesi

2. Strong Field Typing

Field typelarının bilinmemesinin, değişmesi gibi problemler

5. Standart

Keskin bir dille ayrimının olmaması

3. Streaming olmaması

Rest API için streaming mantığının bulunmaması

6. Dökümantasyon

İyi dökümantasyonun edilmemiş REST API'lar sıkıntı olacaktır

gRPC

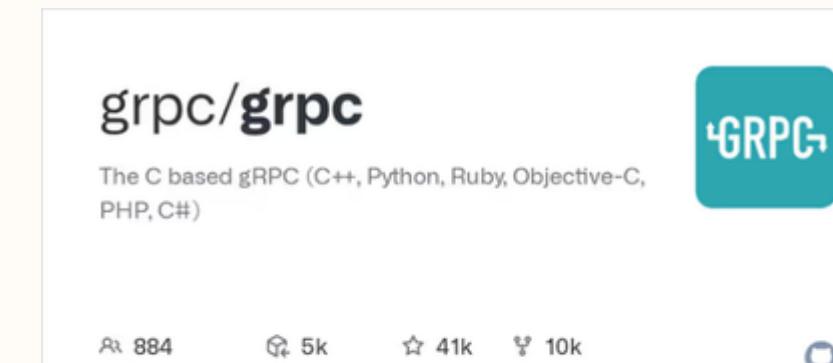
gRPC, herhangi bir ortamda çalışabilen modern, açık kaynaklı, yüksek performanslı (RPC) framework 'üdür.

1. Google geliştirdi

Google tarafından Ağustos 2016 açık kaynak olarak yayınladı. CNCF (Cloud Native Computing Foundation) 'e bağışlandı.

4. HTTP 2.0

gRPC HTTP 2.0 versiyonu ile uyumludur. HTTP 2 özelliklerini kullanır.

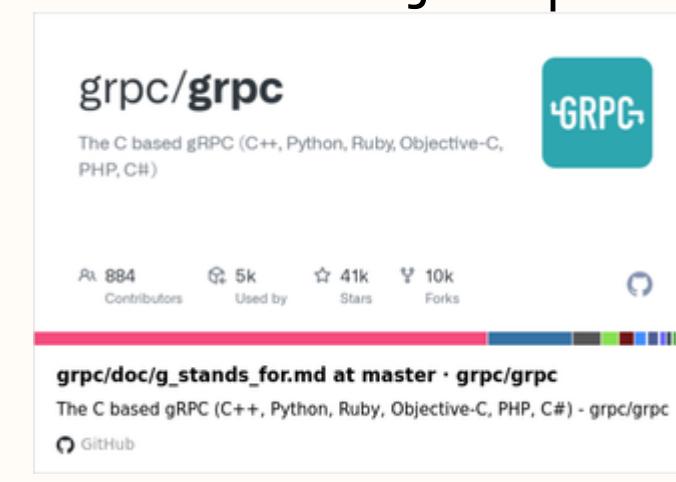


2. Protocol Buffers

Güçlü bir binary serialization dili protocol buffers kullanır.

5. Kolay Dokümantasyon

<https://grpc.io/docs>
<https://protobuf.dev/>



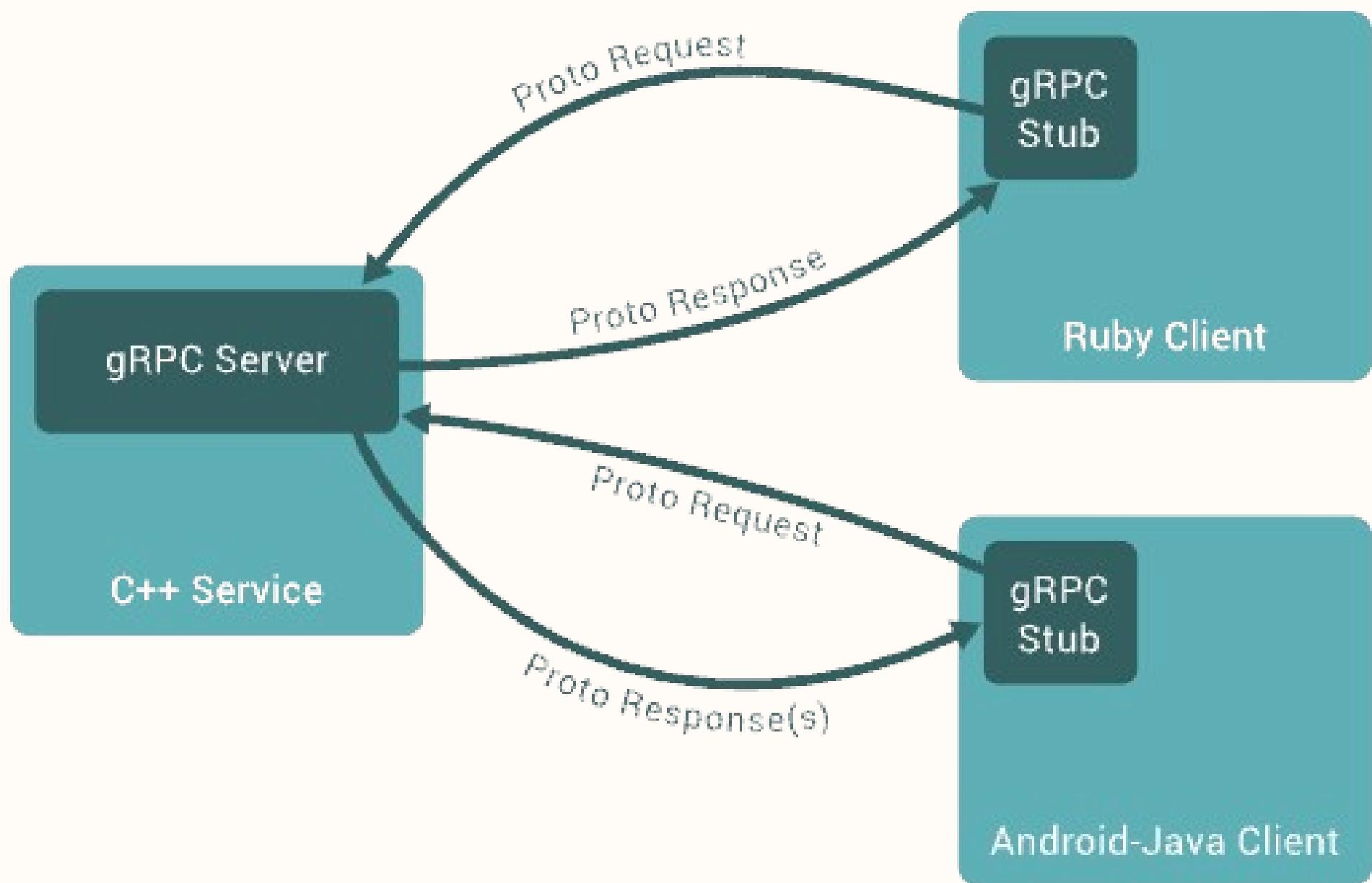
3. Ücretsiz ve Açık Kaynak

Temel özelliklerinin hepsi herkesin kullanımına açıktır.

6. Sadece google değil

Square, Netflix, CoreOS, Docker, Cisco, CockroachDB, Juniper Networks

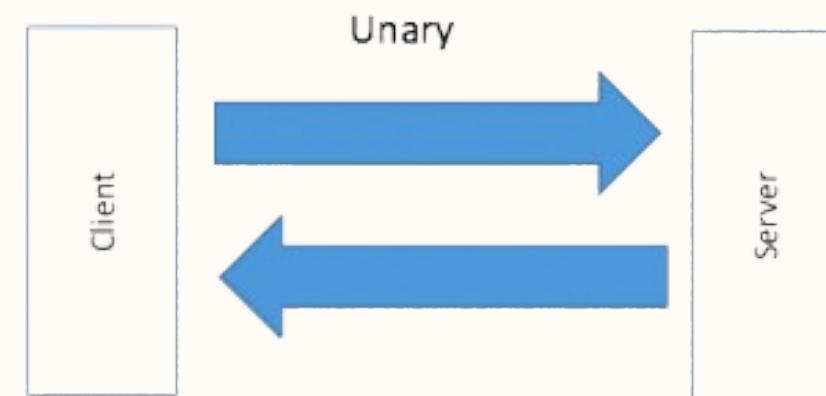
gRPC



Language	OS	Compilers / SDK
C/C++	Linux, Mac	GCC 7.3.1+, Clang 6+
C/C++	Windows 10+	Visual Studio 2019+
C#	Linux, Mac	.NET Core, Mono 4+
C#	Windows 10+	.NET Core, .NET 4.5+
Dart	Windows, Linux, Mac	Dart 2.12+
Go	Windows, Linux, Mac	Go 1.13+
Java	Windows, Linux, Mac	Java 8+ (KitKat+ for Android)
Kotlin	Windows, Linux, Mac	Kotlin 1.3+
Node.js	Windows, Linux, Mac	Node v8+
Objective-C	macOS 10.10+, iOS 9.0+	Xcode 12+
PHP	Linux, Mac	PHP 7.0+
Python	Windows, Linux, Mac	Python 3.7+
Ruby	Windows, Linux, Mac	Ruby 2.3+

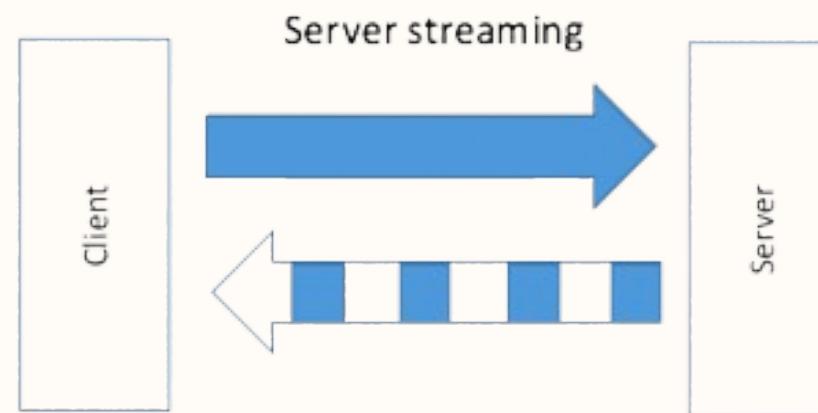
1. Unary

Client'ın servera 1 istek yollayıp 1 cevap aldığı RPC türüdür.



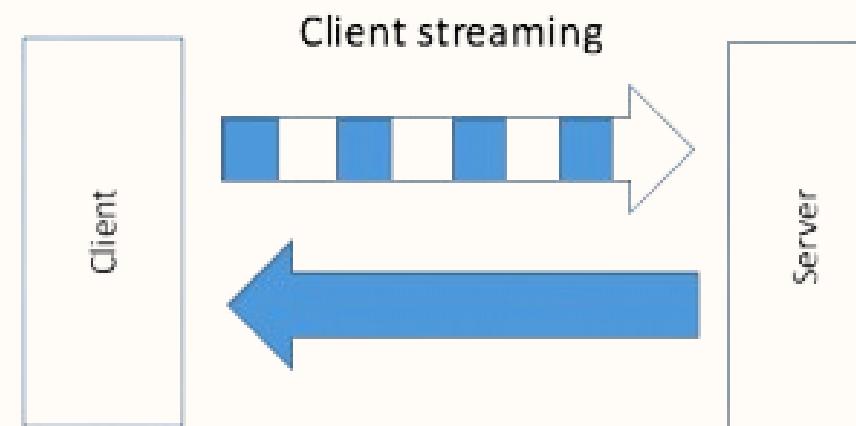
2. Server'dan Client'a Streaming

Client'ın istek attıktan sonra serverin yanıtı stream olarak verdiği türdür.



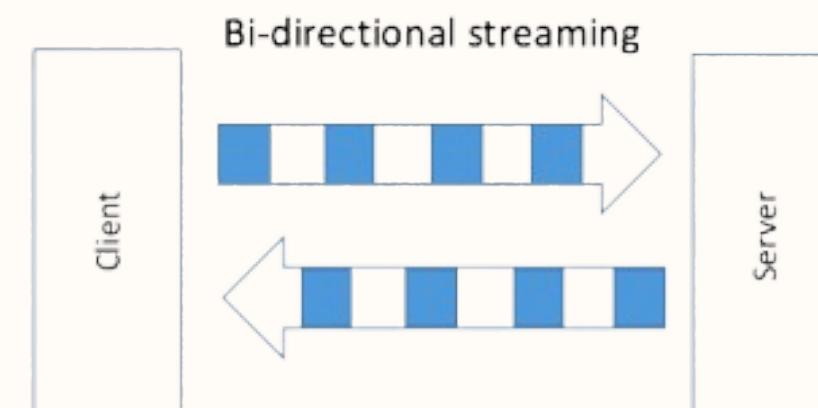
3. Client'tan Server'a Streaming

Client'tın stream başlatarak mesajları servera yollayıp, serverin yanıt verdiği türdür.



4. Bi-directional Streaming

Client ve server haberleşmesinin streaming olarak yapılması durumudur. İki stream birbirinden bağımsız işler ve client ile server mesajları herhangi bir sırayla okuyup, yazabilir.



Protocol Buffers

gRPC haberleşmesinde verileri binary formatında haberleşmeye sağlayan yapıdır. .

1. Veri Tipleri

gRPC bir çok veri tipini destekler.

2. Strong Field Typing

Field typeler bellidir.

3. Basitlik

Yazması basit ve keyiflidir.

4. IDL

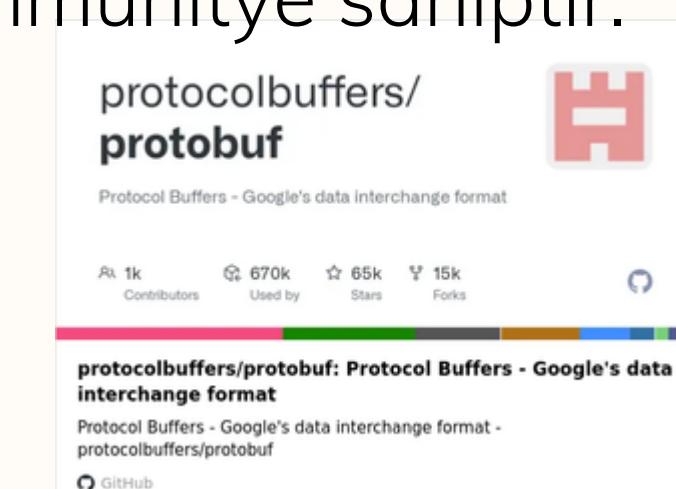
Google Protocol Buffers
Apache Thrift
Apache Avro
OpenAPI/Swagger
WSDL

5. Community

Hem gRPC hemde Protocol buffers güçlü bir communitye sahiptir.

6. Dökümantasyon

kolay okunabilir, anlaşılır bir dökümantasyona sahiptir.



PROTOCOL BUFFERS ÖZELLİKLERİ



```
public class Person{  
    private String name;  
    private int age;  
    private boolean isRetired;  
    //getter  
    //setter  
}
```



```
{  
    "name": "Ümit",  
    "age": 31,  
    "isRetired": false  
}
```



```
message Person{  
    string name=1;  
    int32 age=2;  
    bool isRetired=3;  
}
```



```
1=Ümit  
2=31  
3=1
```

Java Type	Proto Type
int / long	int32 & sint32 / int64 & sint64
float / double	float / double
boolean	bool
String	string
byte[]	bytes

PROTOCOL BUFFERS ÖZELLİKLERİ



```
public class Person{  
    private String name;  
    private int age;  
    private boolean isRetired;  
    //getter  
    //setter  
}
```



```
{  
    "name": "Ümit",  
    "age": 31,  
    "isRetired": false  
}
```



```
message Person{  
    string name=1;  
    int32 age=2;  
    bool isRetired=3;  
}
```



```
1=Ümit  
2=31  
3=false
```

Proto Type	Default
int32 / any number type	0
bool	false
string	empty string
enum	first value
repeated / map	empty list / empty map

PROTOCOL BUFFERS ÖZELLİKLERİ

1. Unique field number

Her değişkenin karşısındaki değer eşsiz olmalıdır.

2. Byte Kullanımı

1-15 için kullanılan fieldlar 1 byte, 16-2047 arası 2 byte kullanır.

3. Reserved

19000 - 19999 arası sayılar reserved edilmiştir.

4. Eğer kullanılmışsa

Eğer field daha önce kullanıldı ve kaldırıracaksanız, reserved ile gelecekte yeniden kullanabilirsiniz.

5. Empty, Timestamp vs

google içerisinde bir çok farklı iş için kütüphanede sınıf bulunduruyor, onu kullanabilirsiniz.

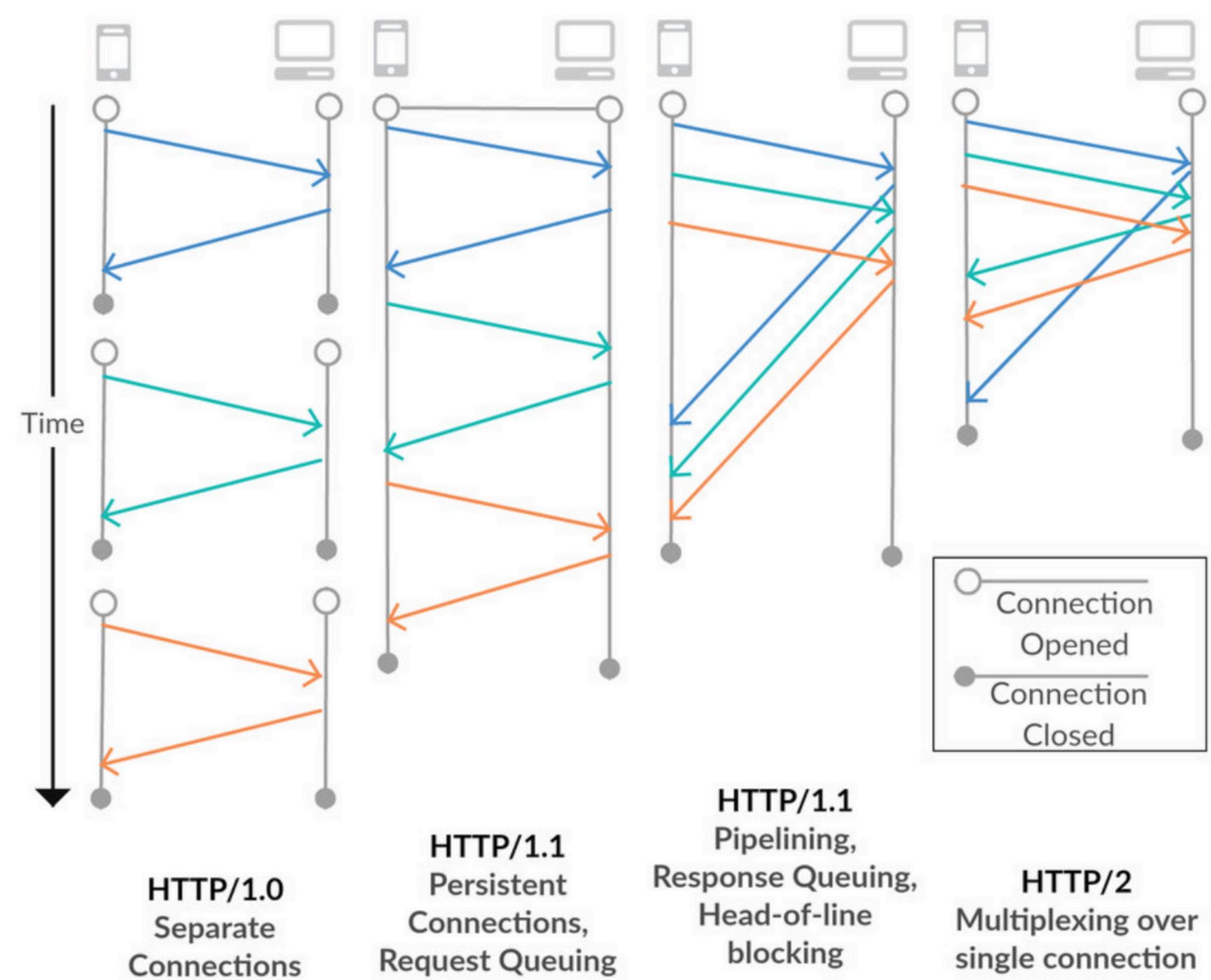
6. En büyük ve en küçük sayı

1'den küçük ve 536.870.911 sayısından büyük tanımlama yapılamaz.

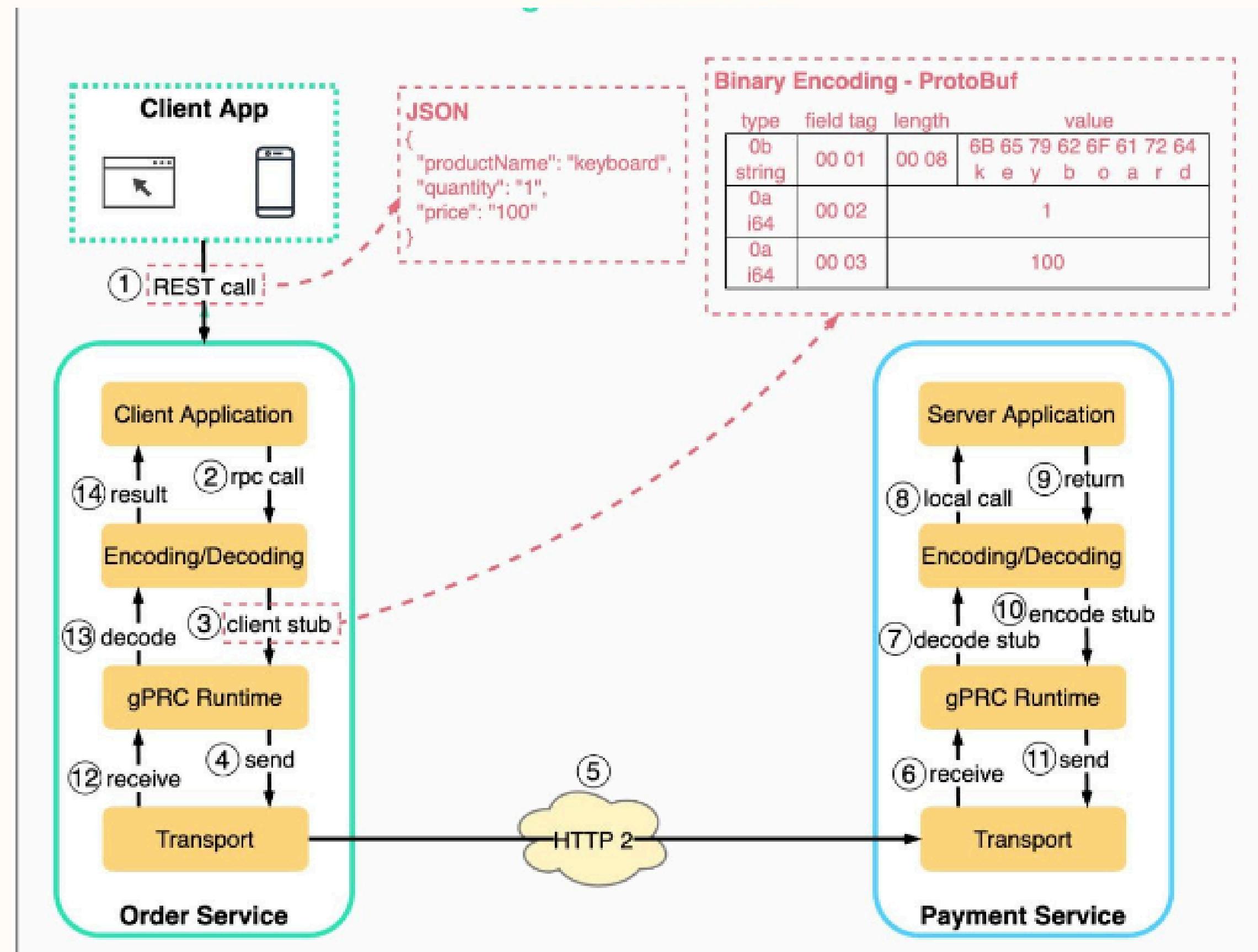
HTTP 1.0 'DAN HTTP 2.0

Örnek bir uygulama:

<http://www.http2demo.io/>



ÖZET OLARAK NE OLUYOR?



Görsel [Bytebytego](#) sitesinden alınmıştır.

SOAP & REST & gRPC

REST Web Service	Soap	gRPC
REST mimari'dır.	SOAP protokoldür.	Framework 'tür.
REST SOAP gibi çok fazla standart tanımlamaz.	SOAP standartlarına uymaktadır	Protocol Buffers kullanılmalı, gRPC ile iletişim sağlanmalı.
REST, SOAP'tan daha az bant genişliği ve kaynak gerektirir.	SOAP, REST'den daha fazla bant genişliği ve kaynak gerektirir.	Binary serialization dolayısı diğer web servislere göre daha az bant genişliği gerektirir.
Http'nin güvenliğini kullanır.	Kendi Güvenlik önlemleri vardır.	Kendi güvenlik önlemleri de vardır.
REST birden fazla veri formatına izin verir. Plain text, HTML, XML, JSON etc.	SOAP XML'e olanak sağlar	Protocol buffers ile serialization olarak veri taşınır.
HTTP tüm metotlarıyla çalışır.	Http POST metotuyla çalışır.	?
WSDL'a ihtiyaç duymaz	WSDL'a ihtiyaç duyar	Proto ile kullanır.



TEŞEKKÜRLER

Ümit KÖSE
umiitkose