CEU | Universidad San Pablo

Introducción a los Servicios Web

Sistemas Web II

Grado en Ingeniería de Sistemas de Información

Álvaro Sánchez Picot
alvaro.sanchezpicot@ceu.es

v20250213

Basado en el trabajo de:

- David González Márquez

¿Qué es una API?

¿Qué es un servicio web?

¿Qué es una API?

- Application Programming Interface
- Interacción entre dos piezas de software
- Lista de:
  - Métodos para invocar (requests)
  - Sus parámetros
  - Valores de retorno
  - Formato de datos

¿Qué es una API?

- API local:
  - Interacción en la propia máquina. ej:
    - API del Sistema Operativo
    - Biblioteca de C
- API remota (servicio web):
  - Interacción en máquinas diferentes
  - Comunicación a través de una red
  - Ej.: API de Twitter

¿Qué es una API?

- Proveedor (provider):
  - La entidad que ofrece sus servicios a través de la API
- Consumidor (consumer):
  - La entidad que solicita los servicios

¿Qué es un servicio web?

Techopedia.com

A Web service is a software service used to communicate between two devices on a network.

More specifically, a Web service is a software application with a standardized way of providing interoperability between disparate applications. It does so over HTTP using technologies such as XML, SOAP, WSDL, and UDDI.

¿Qué es un servicio web?

Webopedia.com

The term Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall.

¿Qué es un servicio web?

Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. The applications interface, not the users. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users.

Web services allow different applications from different sources to communicate with each other without time-consuming custom coding, and because all communication is in XML, Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl, Windows applications can talk with UNIX applications.

Web services do not require the use of browsers or HTML.

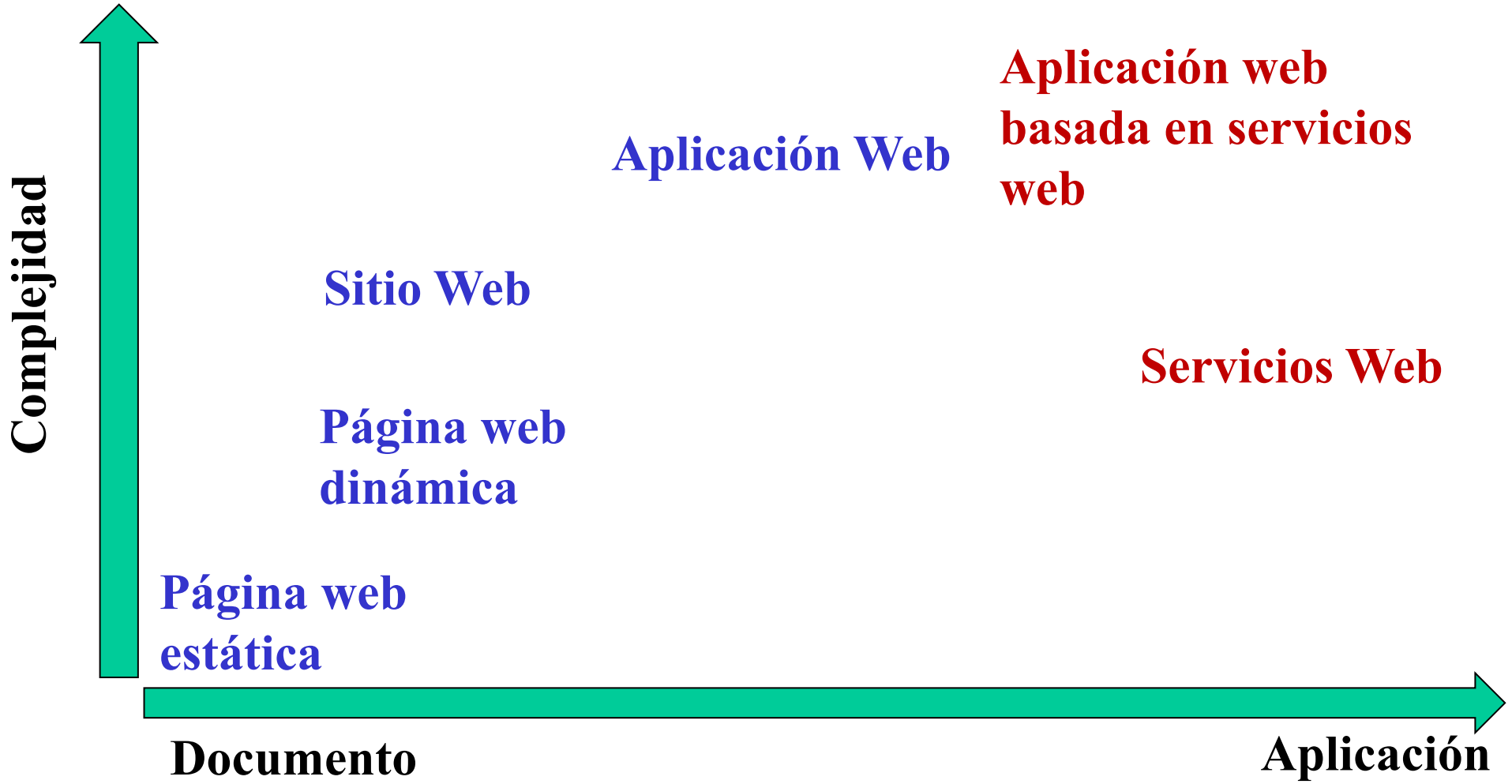
Web services are sometimes called application services.

¿Qué es un servicio web?

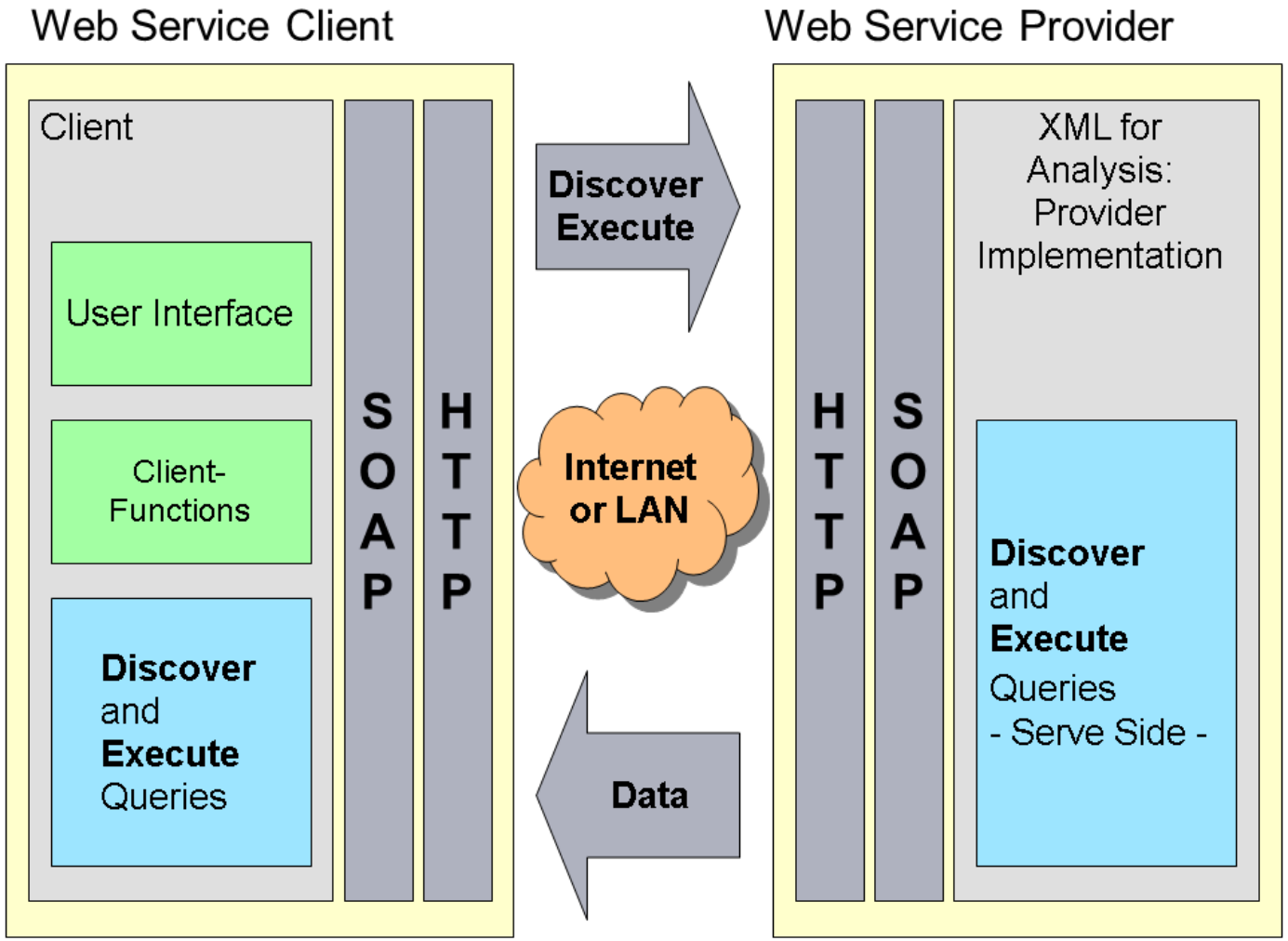
W3C

- A Web service is a software system designed to support interoperable machine-to-machine interaction over a network.
- (Para que sea una implementación estandarizada)
- It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

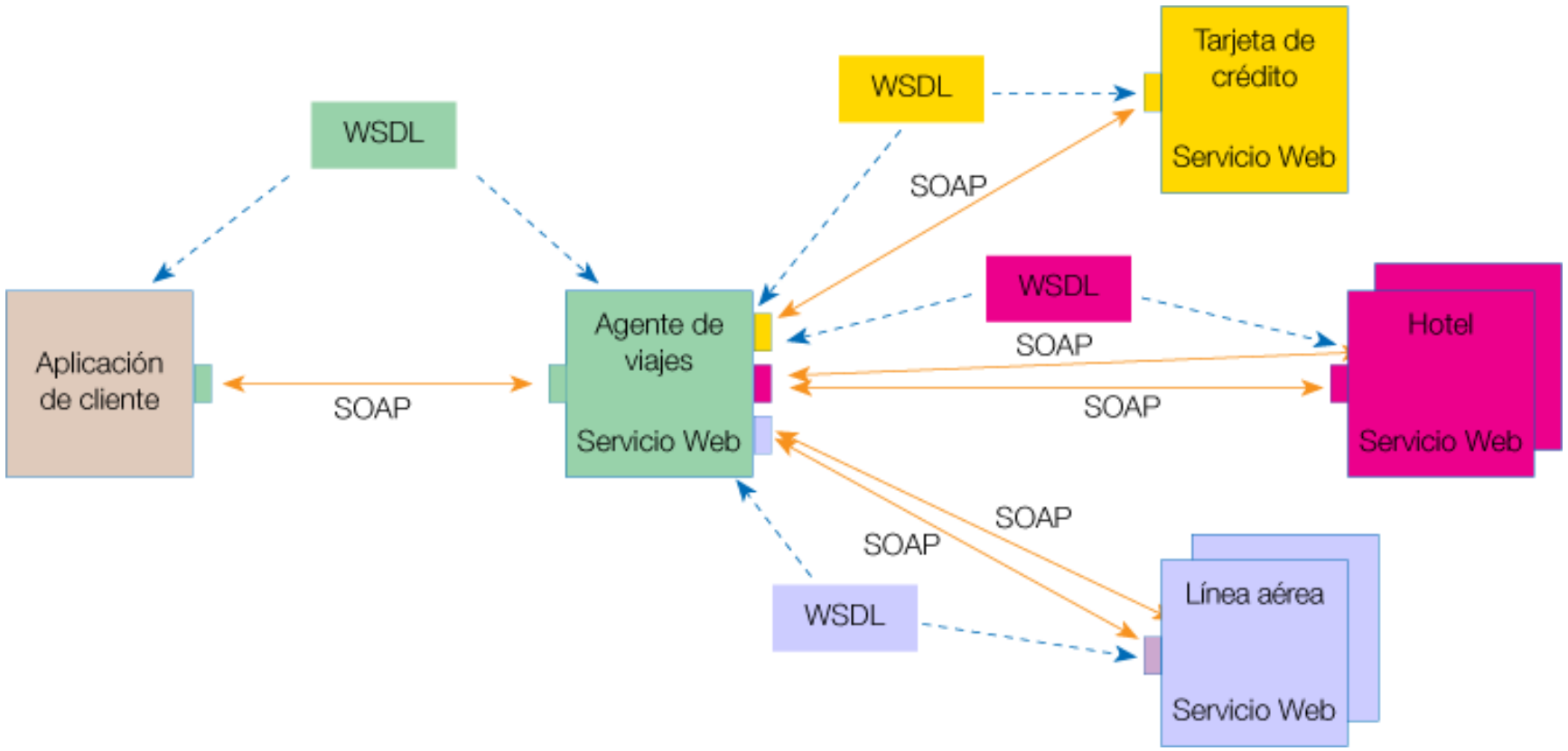
Mapa de términos



Estructura


Web Service Client
Web Service Provider
Client
User Interface
Client-
Functions
Discover
and
Execute
Queries
SOAP
HTTP
Discover
Execute
Internet
or LAN
Data
HTTP
SOAP
XML for
Analysis:
Provider
Implementation
Discover
and
Execute
Queries
- Serve Side -

Estructura


WSDL
WSDL
Tarjeta de crédito
Servicio Web
SOAP
WSDL
Hotel
Servicio Web
SOAP
SOAP
Aplicación de cliente
SOAP
Agente de viajes
Servicio Web
SOAP
SOAP
WSDL
Línea aérea
Servicio Web

¿Ventajas?

Ventajas

- Interoperabilidad
- Estandarización
- Ubicuidad
- Encapsulamiento (information hiding)
- Integración y composición de servicios

Ventajas

- Interoperabilidad
  - Independientes de la plataforma
  - Independientes del lenguaje
- Estandarización
  - Las APIs son contratos que no deberían romperse
  - El proveedor promete no cambiar la API
  - Protocolos basados en texto, que hacen que sea más fácil acceder a su contenido y entender su funcionamiento

Ventajas

- Ubicuidad
  - Los servicios web se comunican utilizando HTTP, XML y JSON, por lo tanto, cualquier dispositivo que soporte estas tecnologías puede implementar o acceder a estos servicios
- Encapsulamiento
  - Information hiding
  - Ninguna de las partes conoce la implementación de la otra
- Integración y composición de servicios
  - Reutilización de servicios
  - Creación de servicios más complejos

Ventajas (Aplicaciones web)

- Ahorran tiempo de instalación y despliegue
- “No hay problemas de compatibilidad”
- No hay necesidad de instalarlas y no ocupan espacio
- Actualizaciones inmediatas
- Consumo de recursos bajo
- Multiplataforma
- Portables (PC, Móvil,…)
- Alta disponibilidad
- Difícil de atacar mediante virus
- Ideales para colaboración

¿Desventajas?

Desventajas

- Soporte de transacciones
  - Por detrás de CORBA u otros estándares
- Rendimiento
  - Por su uso de texto y XML, por detrás de RMI, DCOM o CORBA
- Problemas derivados del uso de HTTP
- Interfaces “inmutables”
- Respuesta no garantizada

CORBA (Common Object Request Broker Architecture): Estandar que facilita el desarrollo de aplicaciones distribuidas
RMI (Remote Method Invocation): API de Java para invocación remota de métodos
DCOM (Distributed Component Object Model): Software de Microsoft para facilitar la comunicación entre objetos COM

Desventajas (Aplicaciones web)

- Menor funcionalidad que las aplicaciones de escritorio
- ~~Limitaciones del navegador~~
- Disponibilidad supeditada al proveedor de red
- Alta dependencia del servidor
- Limitación por el protocolo HTTP utilizado
- Cuello de botella en el ancho de banda para grandes cantidades de datos
- Necesidad de cifrar los datos
- ~~Necesidad de software adicional o versiones específicas de navegadores~~

70 – 80: RPCs

- Remote Procedure Calls
- Exponer funcionalidades
- Invocar bibliotecas remotas como si fueran locales
- Aparición de las redes

90: Intento de estandarización

- Aparición de varios estándares:
  - Common Object Request Broker Architecture (CORBA)
  - Component Object Model (COM)
  - Distributed Component Object Model (DCOM)
- Complicados de manejar
- Requerían uso del mismo lenguaje
- Requerían instalación en local de parte del servicio remoto (_stub_)

1998: XML-RPC

- Evolución de las RPCs
- Usa HTTP para el transporte
- Codifica la información en XML
- Pretende ser sencillo (7 páginas de especificación)

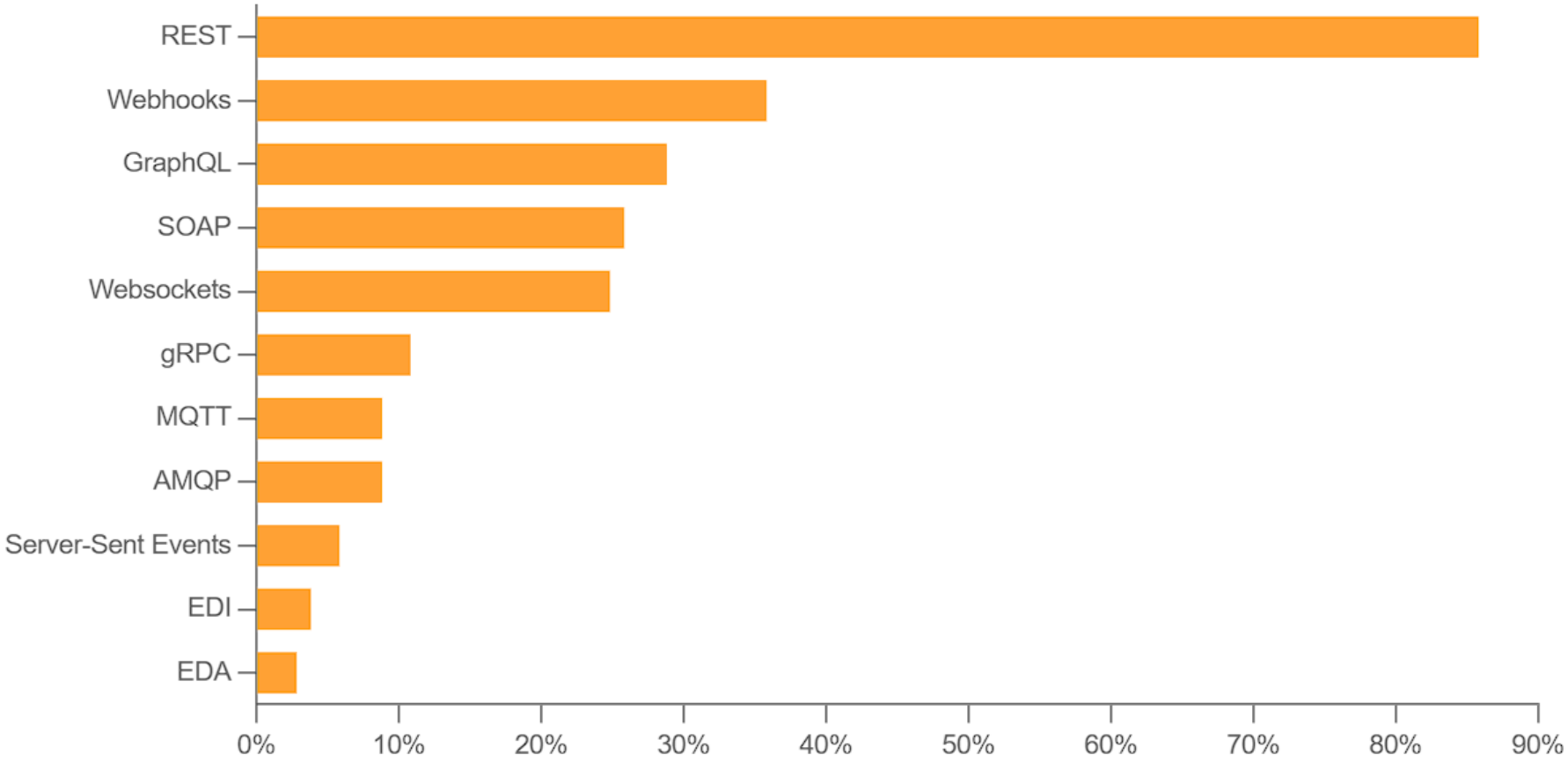
1999: SOAP

- Originalmente Simple Object Access Protocol
- Evolución de XML-RPC
- Especificación en el IETF
- Versión 1.2 recomendación en el W3C

2000: REST

- Representational state transfer
- Estilo de arquitectura software
- Aplica una serie de restricciones
- Stateless

Tecnologías



Multiple choices allowed.

Fuente: https://www.postman.com/state-of-api/2023/api-technologies/#api-technologies

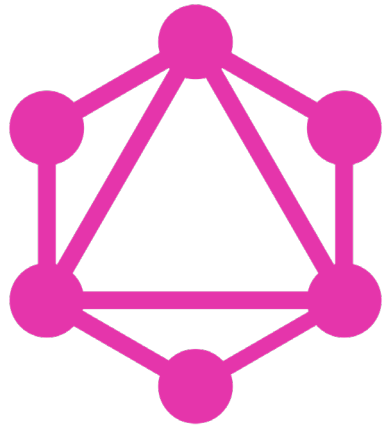
Webhooks

- user-defined callbacks made with HTTP
- Mensajes automatizados resultado de un evento concreto
- Mensajes específicos
- Para comunicación uno a uno entre dos apps
  - Usados por ejemplo en CI/CD
- Más información

```
https://yourapp.com/data/12345?Customer=bob&value=10.00&item=paper
```

```
      To:  yourapp.com/data/12345
Message:   Customer:  Bob
             Value:  10.00
              Item:  Paper
```

GraphQL



- Desarrollado por Facebook desde 2012
- Primera versión pública en 2015
- Estable desde 2018
- Gestionado en la actualidad por la Linux Foundation
- Permite solicitar exactamente los recursos necesarios
- Desventajas:
  - Complica el cacheado
  - Complejo para servicios sencillos
- Más información

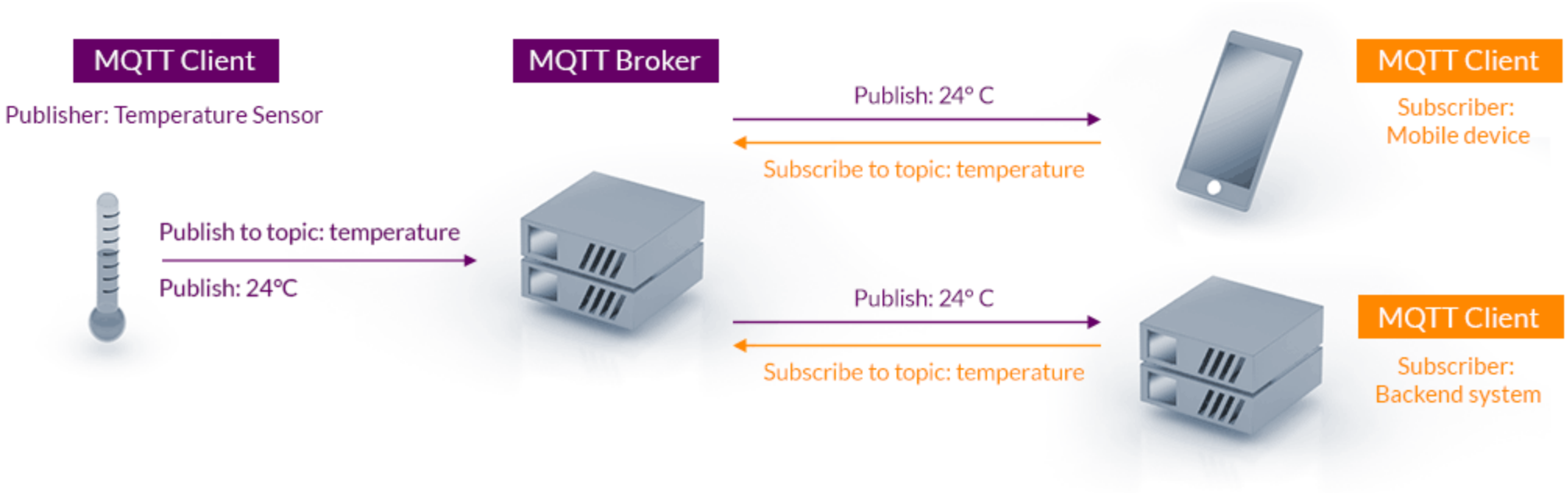
gRPC

- Google Remote Procedure Call
- Open source
- HTTP/2 para transporte
  - Imposible crear un cliente para el navegador
- Más información

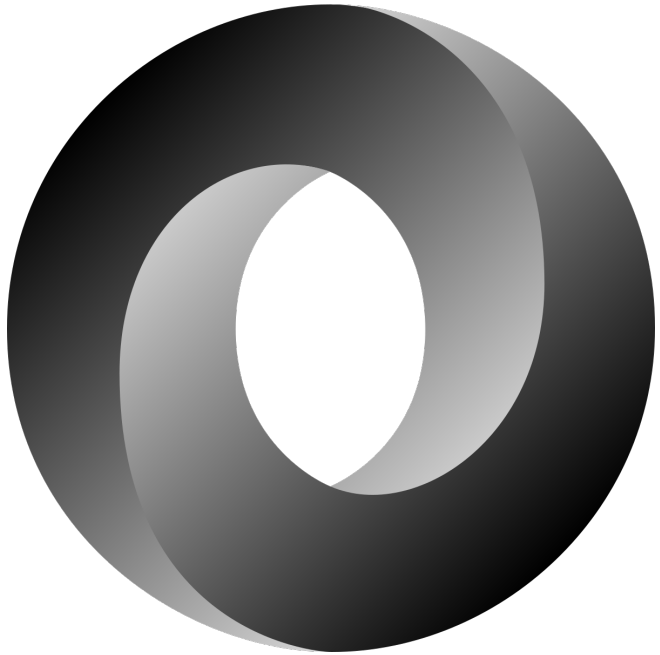
MQTT

- publicación/suscripción de mensajes
- Ligero
- Para dispositivos
- Estándar para IoT
- Más información

MQTT


MQTT Client
Publisher: Temperature Sensor
MQTT Broker
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber:
Mobile device
Publish to topic: temperature
Publish: 24°C
Publish: 24° C
Subscribe to topic: temperature
MQTT Client
Subscriber:
Backend system

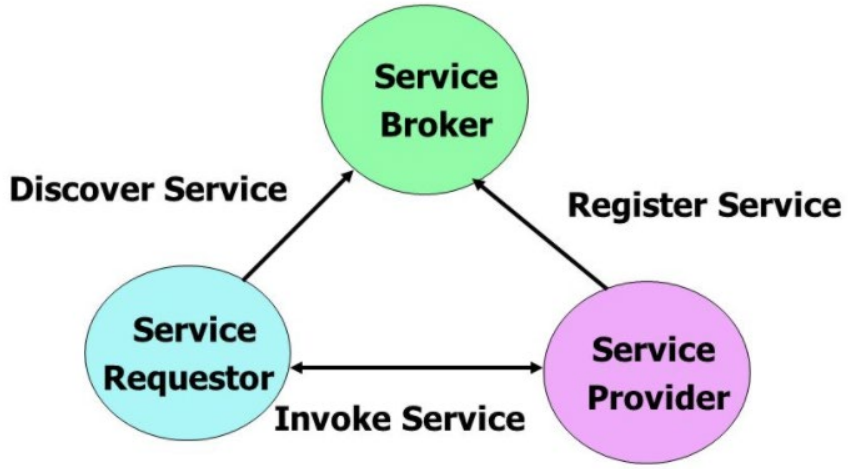
Formato de los datos



JSON

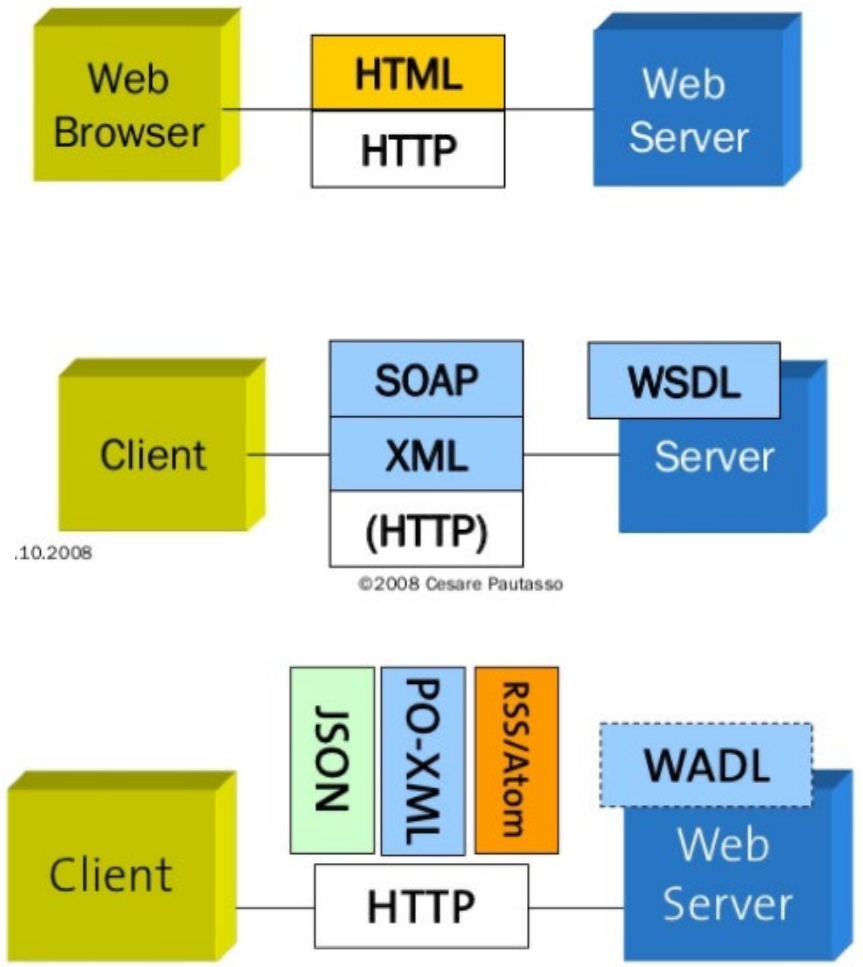
Los servicios web pueden ser

- Descritos…
- Publicados…
- Localizados…
- Utilizados…



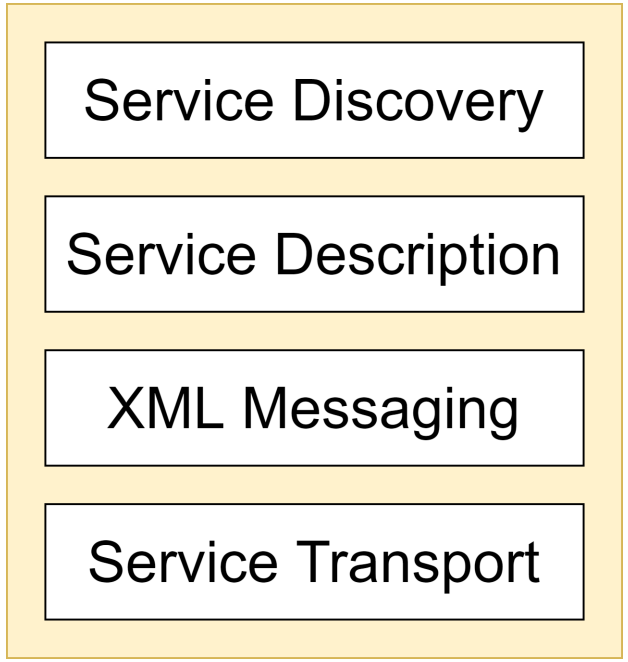
… a través de una red, generalmente Internet.

Arquitectura

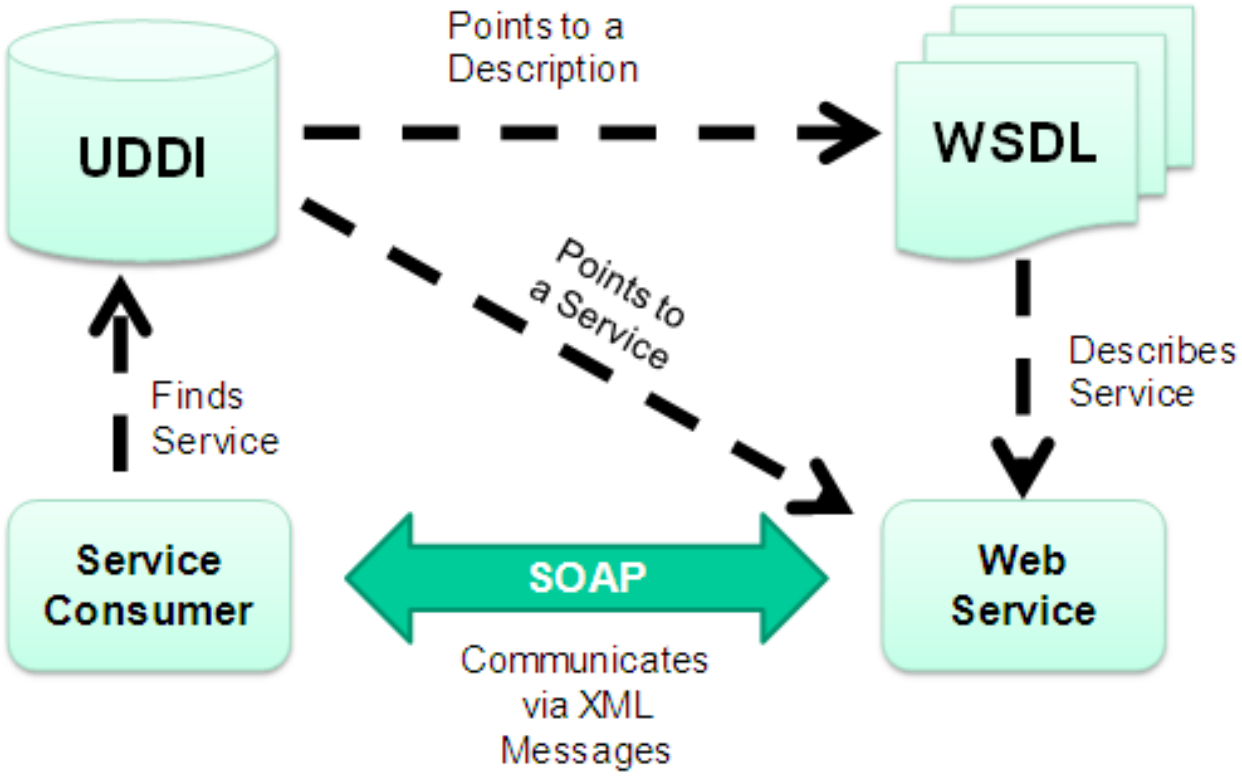


Arquitectura

- El stack del protocolo:
  - Descubrimiento de servicios
    - UDDI
  - Descripción de servicios
    - WSDL
  - Mensajes XML
    - XML-RPC y SOAP
  - Capa de transporte
    - HTTP, SMTP, FTP, etc.



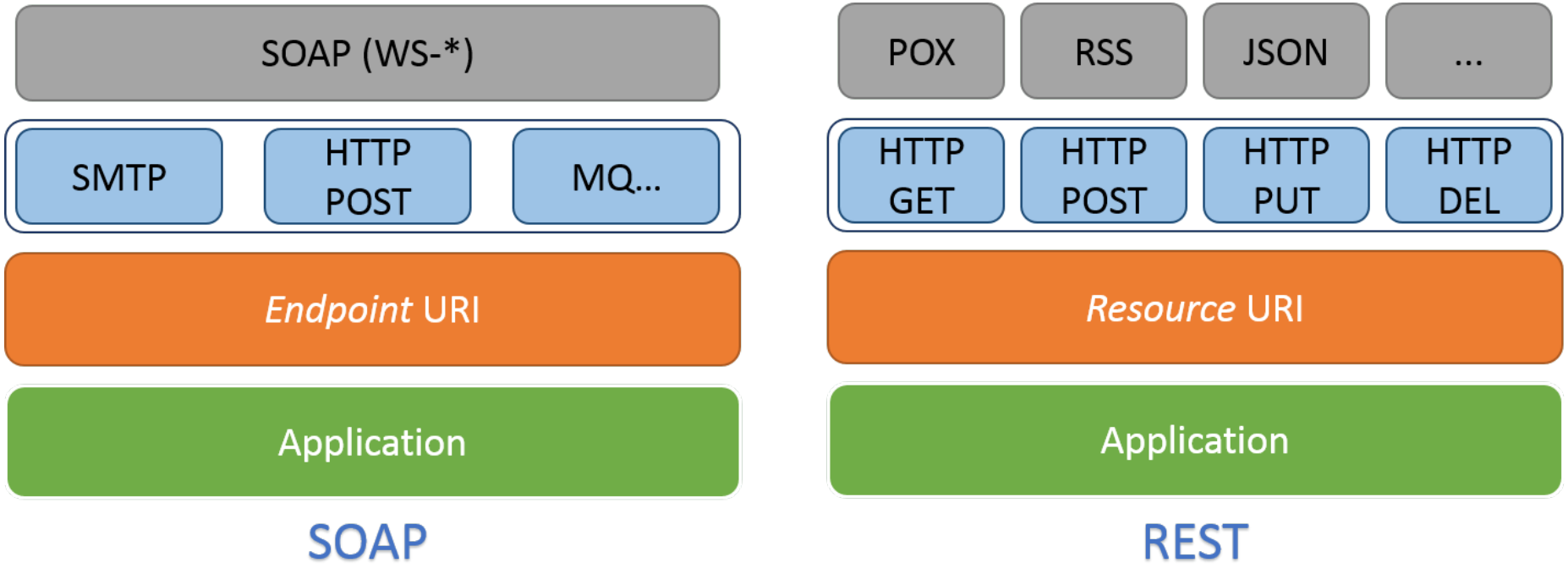
Arquitectura



Tipos de servicios web

- SOAP
  - Simple Object Access Protocol
  - Es un protocolo
  - Expone operaciones que representan lógica
  - Las operaciones se muestran con WSDL
- REST
  - REpresentational State Transfer
  - Es una arquitectura de software
  - Se desarrolló paralelamente a HTTP/1.1
  - Cada URL representa un objeto

Tipos de servicios web



Tipos de servicios web

SOAP	REST
An XML-based message protocol	An architectural style protocol
Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
Invokes services by calling RPC method	Simply calls services via URL path
Does not return human readable result	Result is readable, which is just plan XML or JSON
Transfer is over HTTP or other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
JavaScript can call SOAP, but it is difficult to implement	Easy to call from JavaScript
Performance is not great compared to REST	Performance is much better compared to SOAP. Less CPU intensive, leaner code, etc.

Tipos de servicios web

SOAP

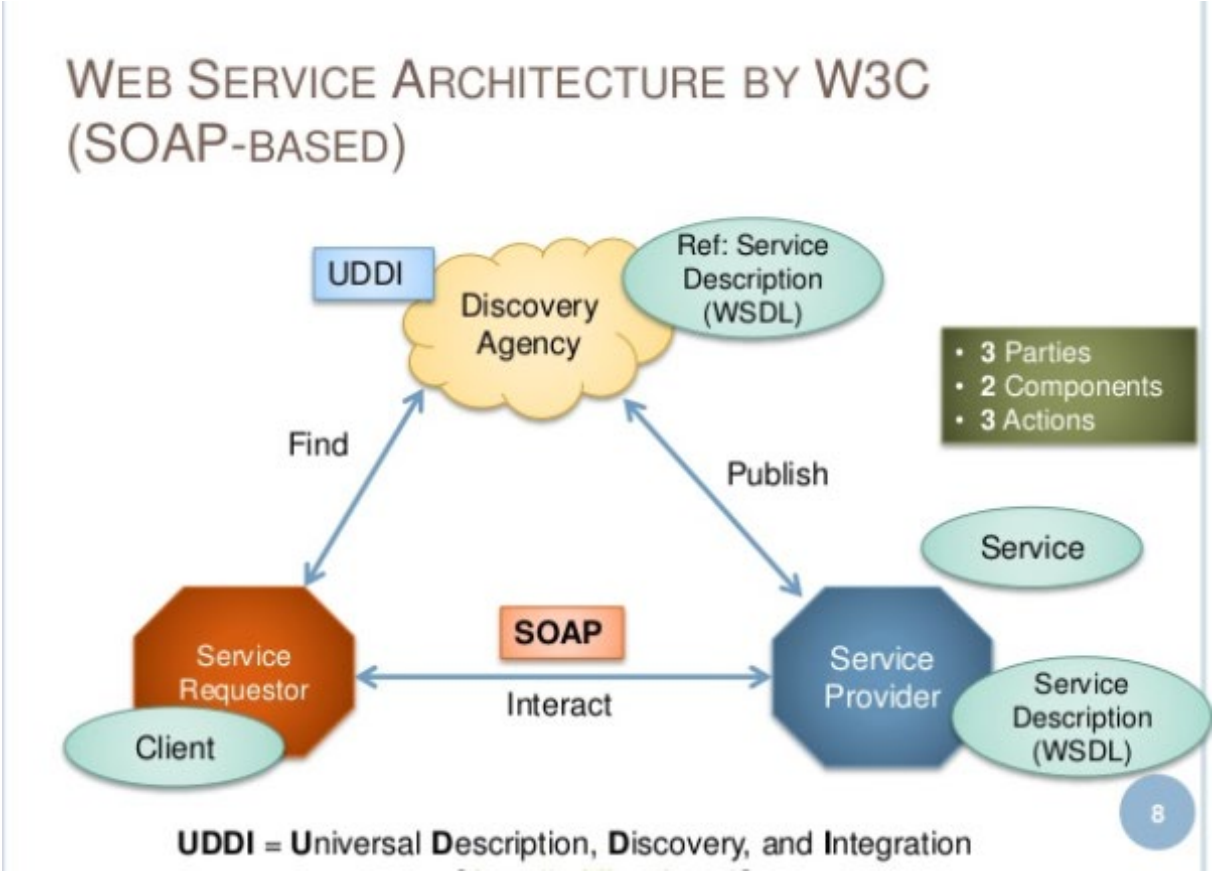
- Expone operaciones que representan lógica
- Codifica todo en XML con adjuntos
- Soporta operaciones con estado (Stateful)
- Muy estandarizado
- Define su propia seguridad

- No puede usar REST

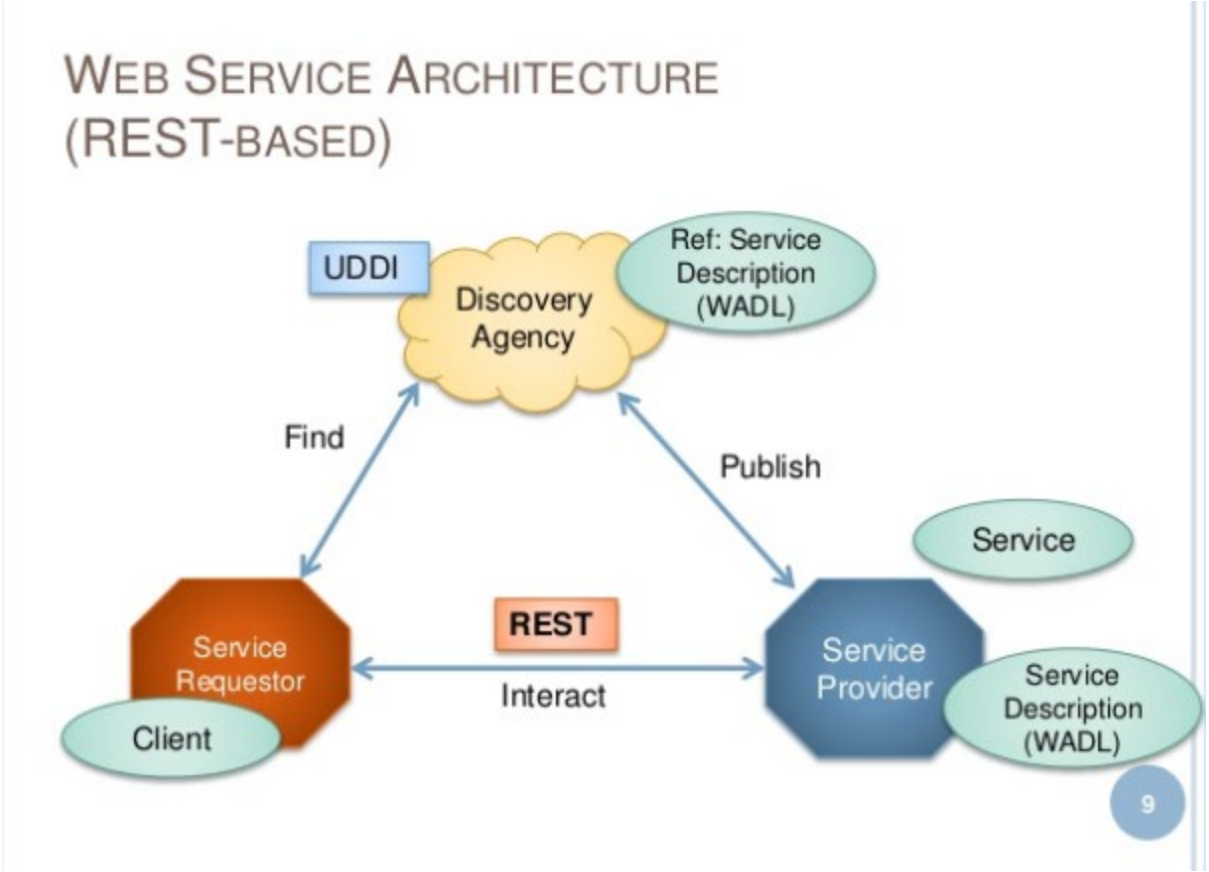
REST

- Expone recursos que representan datos
- Utiliza varios formatos, pero basa todo en texto plano
- Diseñado para operaciones sin estado (Stateless)
- Evita estándares
- Hereda la seguridad del transporte
- Puede usar SOAP

Tipos de servicios web

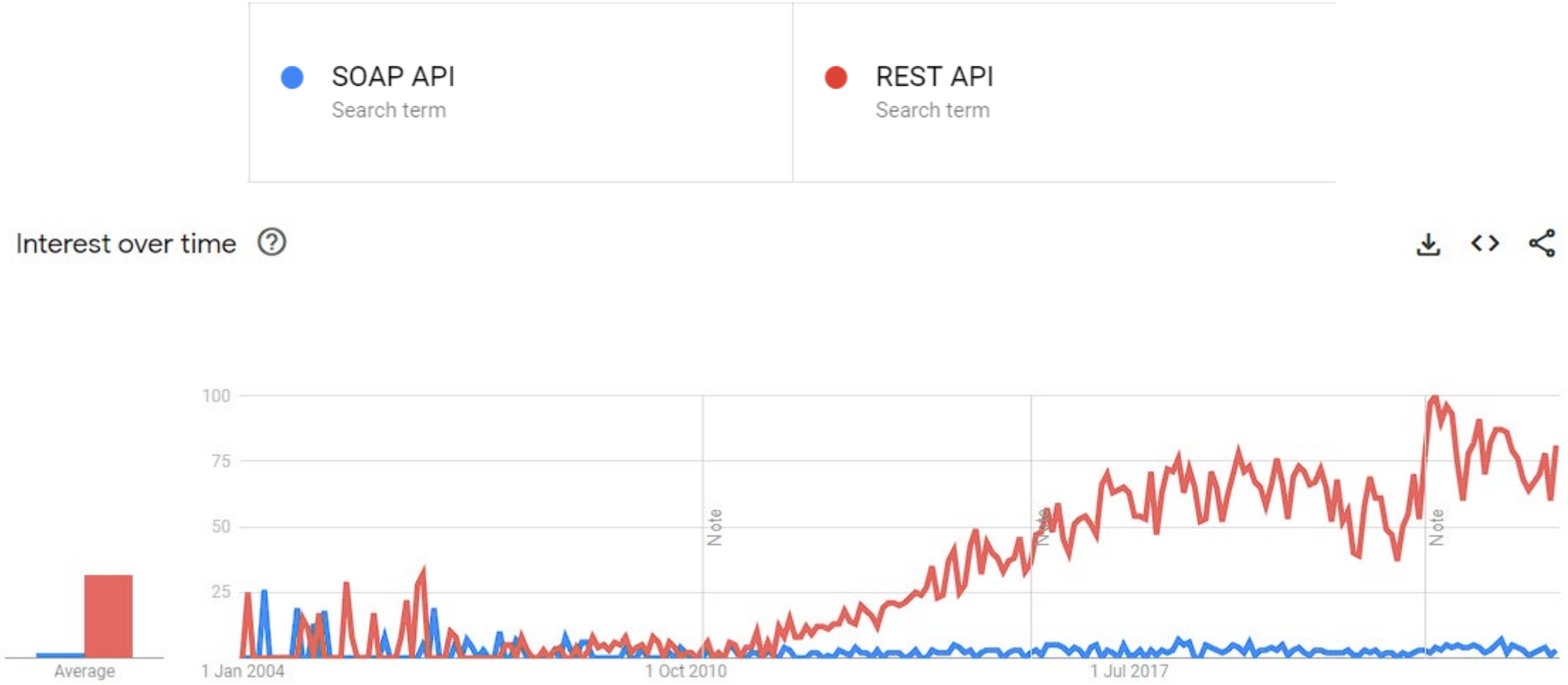


Tipos de servicios web



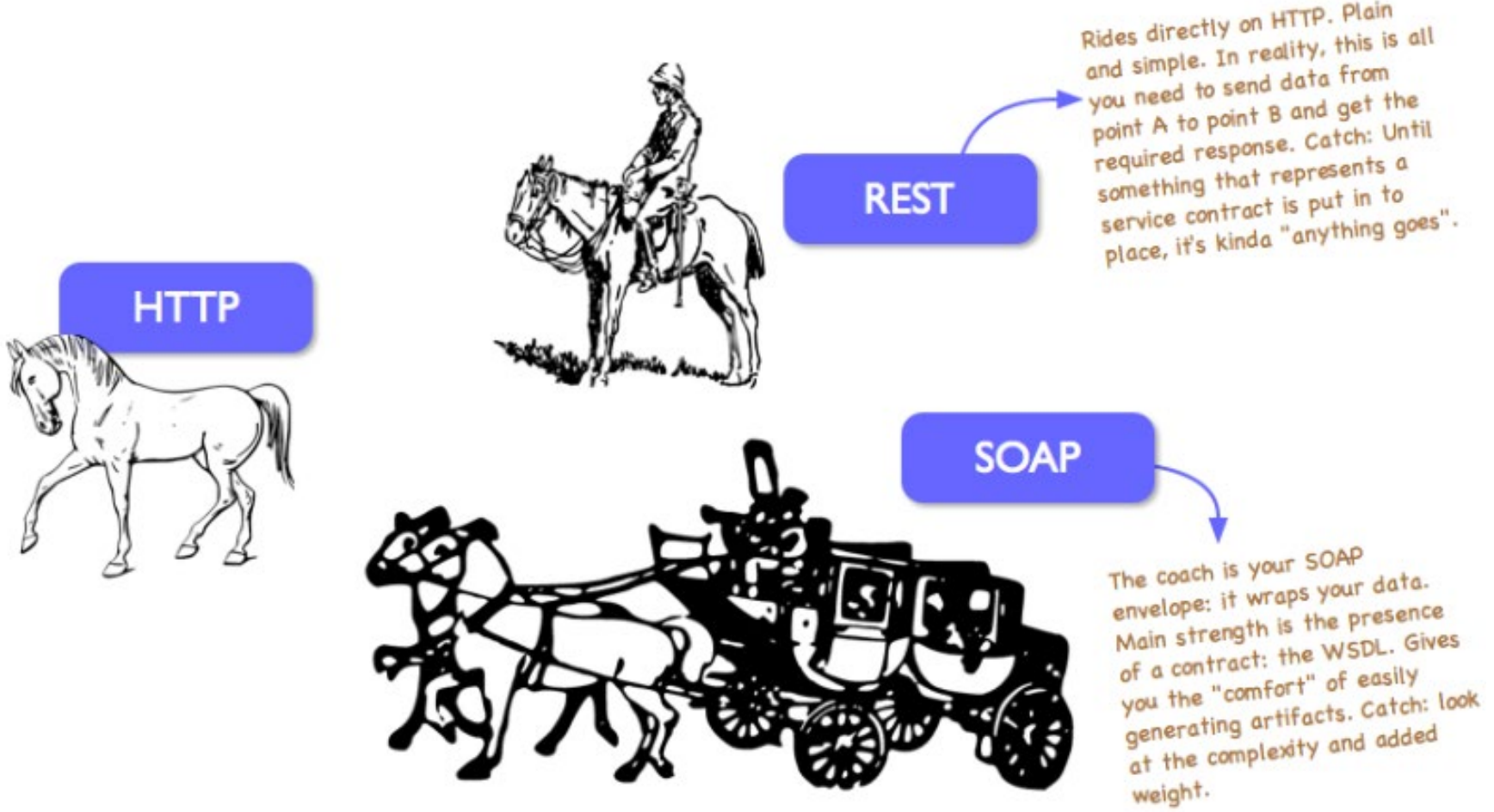
Tipos de servicios web

Tipos de servicios web



https://trends.google.com/trends

Tipos de servicios web



Tipos de servicios web

Enlace

Tipos de servicios web

REST

REST APIs should be independent of the protocol

- Stateless
- Cacheable
- Uniform Interface
- Idempotent

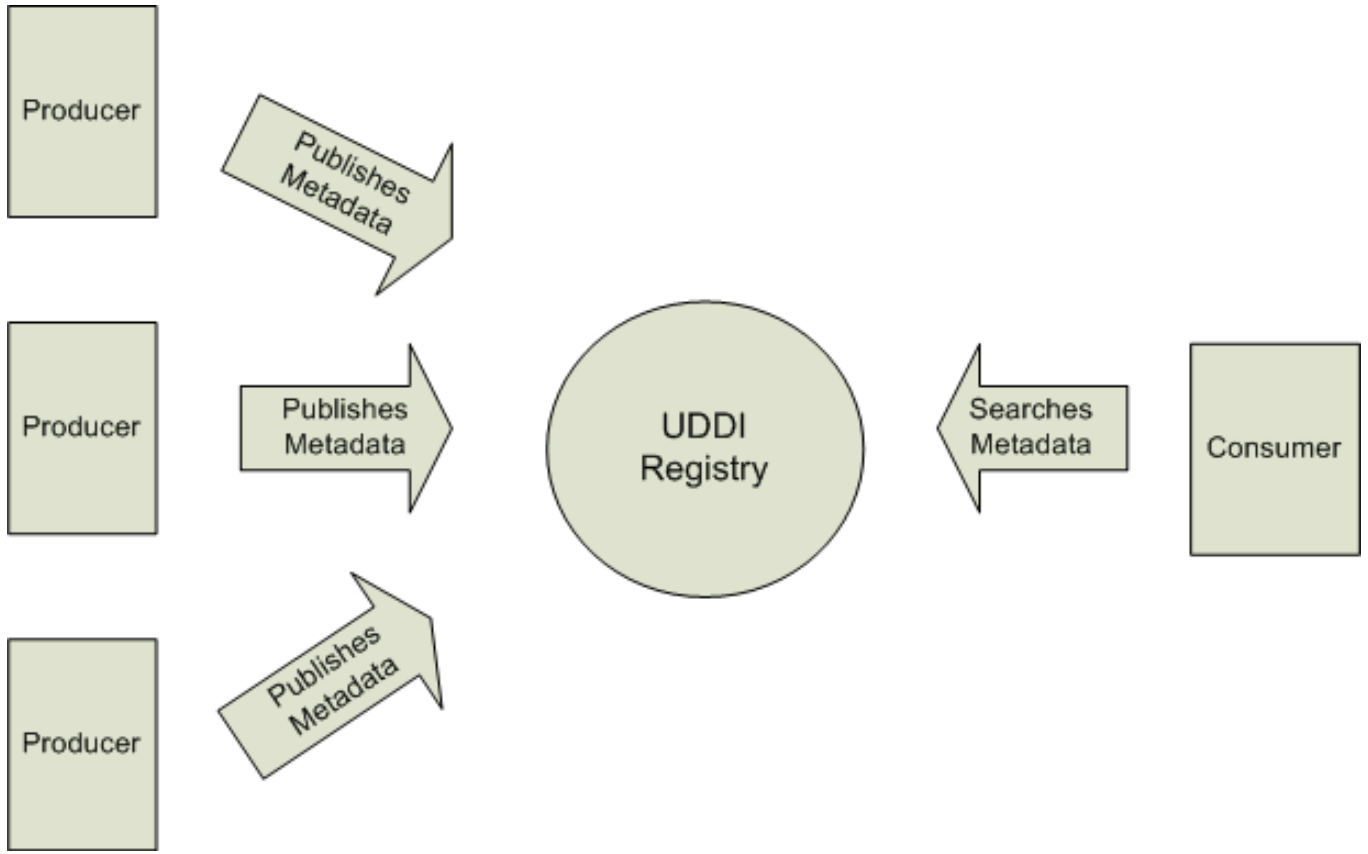
Tipos de servicios web

- RESTful
  - Orientadas al recurso
  - La función va en el método HTTP
- RPC-Style
  - Orientadas al proceso
  - Toda la información va en el “paquete”, el encapsulamiento no cambia
- REST-RPC Hybrid
  - Mix

UDDI

- Universal Description, Discovery and Integration
- Basado en XML
- Pretendía ser un repositorio central de servicios web
- Impulsado por la industria, pero prácticamente abandonado en su idea original
- Se sigue utilizando, pero de forma privada

UDDI


Producer
Publishes Metadata
Producer
Publishes Metadata
UDDI Registry
Searches Metadata
Consumer
Producer
Publishes Metadata

WSDL

- Web Services Description Language
- Basado en XML
- Describe la interfaz pública de los servicios web
- Un cliente puede conectarse a un servicio web para leer el WSDL y determinar qué funciones están disponibles en el servidor

WSDL

Web service interface definition	Port type	Operation signatures
	Messages	Parameter definitions
	Types	Complex type definitions
	Bindings	Transport protocol and payload format
Web service implementation	Service	Service definition element
	Ports	Supported interface bindings

WSDL

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
 xmlns:tns="https://graphical.weather.gov/xml/DWMLgen/wsdl/ndfdXML.wsdl" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
 xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns="http://schemas.xmlsoap.org/wsdl/"
 targetNamespace="https://graphical.weather.gov/xml/DWMLgen/wsdl/ndfdXML.wsdl">
  ▼<types>
    ▼<xsd:schema targetNamespace="https://graphical.weather.gov/xml/DWMLgen/wsdl/ndfdXML.wsdl">
       <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/"/>
       <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/"/>
      ▼<xsd:complexType name="weatherParametersType">
        ▼<xsd:all>
           <xsd:element name="maxt" type="xsd:boolean"/>
           <xsd:element name="mint" type="xsd:boolean"/>
           <xsd:element name="temp" type="xsd:boolean"/>
           <xsd:element name="dew" type="xsd:boolean"/>
           <xsd:element name="pop12" type="xsd:boolean"/>
           <xsd:element name="qpf" type="xsd:boolean"/>
           <xsd:element name="sky" type="xsd:boolean"/>
           <xsd:element name="snow" type="xsd:boolean"/>
           <xsd:element name="wspd" type="xsd:boolean"/>
           <xsd:element name="wdir" type="xsd:boolean"/>
           <xsd:element name="wx" type="xsd:boolean"/>
           <xsd:element name="waveh" type="xsd:boolean"/>
           <xsd:element name="icons" type="xsd:boolean"/>
           <xsd:element name="rh" type="xsd:boolean"/>
           <xsd:element name="appt" type="xsd:boolean"/>
           <xsd:element name="incw34" type="xsd:boolean"/>
           <xsd:element name="incw50" type="xsd:boolean"/>
           <xsd:element name="incw64" type="xsd:boolean"/>
           <xsd:element name="cumw34" type="xsd:boolean"/>
           <xsd:element name="cumw50" type="xsd:boolean"/>
           <xsd:element name="cumw64" type="xsd:boolean"/>
           <xsd:element name="critfireo" type="xsd:boolean"/>
           <xsd:element name="dryfireo" type="xsd:boolean"/>
```

WSDL

```
▼<message name="GmlTimeSeriesRequest">
    <part name="listLatLon" type="xsd:string"/>
    <part name="startTime" type="xsd:dateTime"/>
    <part name="endTime" type="xsd:dateTime"/>
    <part name="compType" type="xsd:string"/>
    <part name="featureType" type="xsd:string"/>
    <part name="propertyName" type="xsd:string"/>
  </message>
▼<message name="GmlTimeSeriesResponse">
    <part name="dwGmlOut" type="xsd:string"/>
  </message>
▼<message name="NDFDgenByDayRequest">
    <part name="latitude" type="xsd:decimal"/>
    <part name="longitude" type="xsd:decimal"/>
    <part name="startDate" type="xsd:date"/>
    <part name="numDays" type="xsd:integer"/>
    <part name="Unit" type="xsd:string"/>
    <part name="format" type="xsd:string"/>
  </message>
▼<message name="NDFDgenByDayResponse">
    <part name="dwmlByDayOut" type="xsd:string"/>
  </message>
▼<message name="NDFDgenByDayLatLonListRequest">
    <part name="listLatLon" type="xsd:string"/>
    <part name="startDate" type="xsd:date"/>
    <part name="numDays" type="xsd:integer"/>
    <part name="Unit" type="xsd:string"/>
    <part name="format" type="xsd:string"/>
  </message>
▼<message name="NDFDgenByDayLatLonListResponse">
    <part name="dwmlByDayOut" type="xsd:string"/>
  </message>
▶<portType name="ndfdXMLPortType">
  ...
  </portType>
▶<binding name="ndfdXMLBinding" type="tns:ndfdXMLPortType">
  ...
  </binding>
▼<service name="ndfdXML">
  ▼<port name="ndfdXMLPort" binding="tns:ndfdXMLBinding">
      <soap:address location="https://graphical.weather.gov:443/xml/SOAP_server/ndfdXMLserver.php"/>
    </port>
  </service>
</definitions>
```

Enlace

WADL

- Web Application Description Language
- Basado en XML
- Describe la interfaz pública de los servicios web
- WSDL en principio no estaba diseñado para REST al no admitir distintos métodos HTTP que POST, por ello surgió WADL
- Menos flexible que WSDL, pero más ligero y fácil de usar
- Sin embargo, para servicios REST no es necesaria una descripción por lo que su uso no está muy extendido

WADL

```
<?xml version="1.0"?>
<application xmlns:xsd="http://www.w3.org/2001/XMLSchema"
             xmlns:html="http://www.w3.org/1999/xhtml"
             xmlns="http://research.sun.com/wadl/2006/10">
  <doc xmlns:smartbear="http://www.smartbear.com/" />
  <resources base="http://localhost:8080/">
    <resource path="accountcreation">
              <method name="GET" id="viewAccountRegistration">
             <doc xml:lang="en" title="Register a new account">
               The account register service can be used to fill in account registration forms.
             </doc>
             <response>
                <representation mediaType="text/html"/>
             </response>
           </method>
              <method name="POST" id="createUserAccount">
              <doc xml:lang="en" title="Register a new account">
                Creating the account after having filled in the registration form.
              </doc>
                <request>
              <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="query" name="username">
                        <doc>The username</doc>
                    </param>
                    <param xmlns:xs="http://www.w3.org/2001/XMLSchema" type="xs:string" style="query" name="password">
                          <doc>The password</doc>
                    </param>
                    <representation mediaType="application/json"/>
                </request>
             <response>
                <representation mediaType="text/html"/>
             </response>
           </method>
   </resource>
  </resources>
</application>
```

Aplicación web VS servicios web

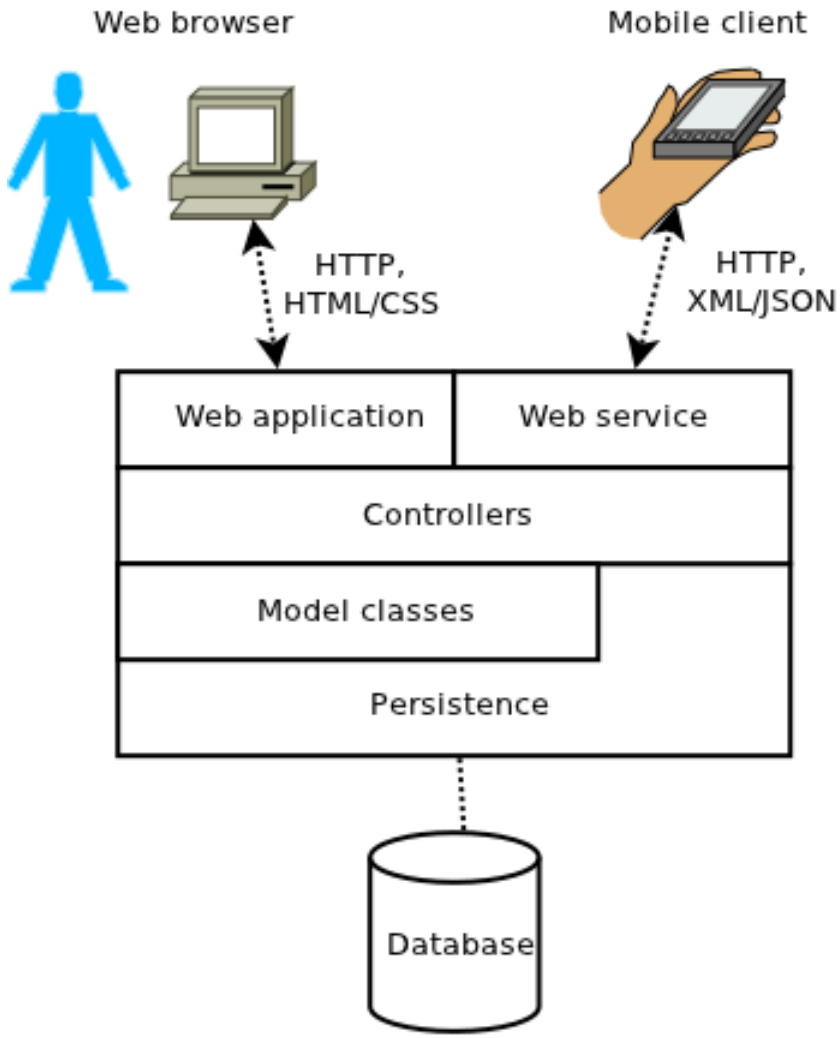
Aplicación web

- Diseñadas para humanos
- Tienen interfaces
- Human to Machine
- Tienen usuario
- Información y presentación (HTML)

Servicios web

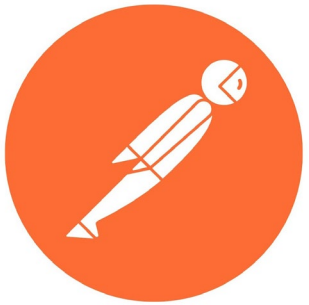
- Diseñados para máquinas
- Exponen APIs
- Machine to Machine
- Tienen aplicaciones
- Información sin presentación (XML)

Aplicación web VS servicios web



Tim Berners-Lee

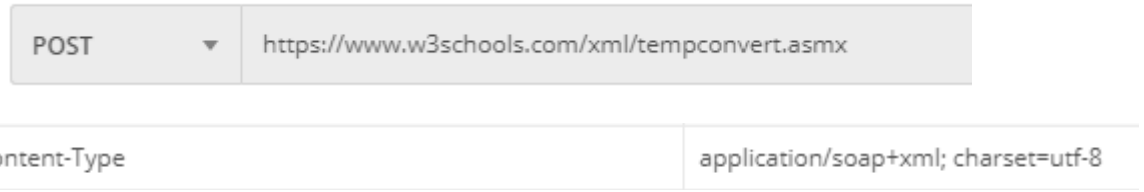
Ejemplos – SOAP



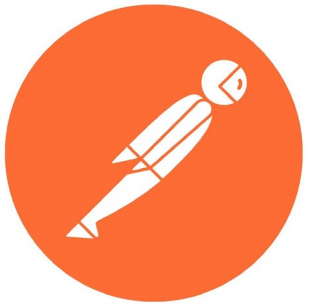
Conversor de temperatura (ºC ↔ ºF)

- Información
- Método: POST
- URL: https://www.w3schools.com/xml/tempconvert.asmx
- Header:

```
POST /xml/tempconvert.asmx HTTP/1.1
Host: www.w3schools.com
Content-Type: application/soap+xml; charset=utf-8
Content-Length: length
```



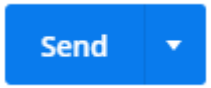
Ejemplos – SOAP



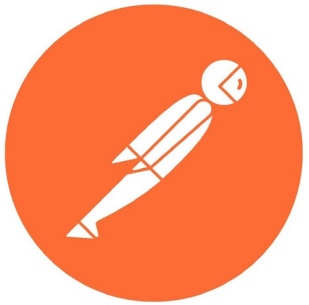
- Body (reemplazar string por el valor deseado):

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <soap12:Body>
    <CelsiusToFahrenheit xmlns="https://www.w3schools.com/xml/">
      <Celsius>string</Celsius>
    </CelsiusToFahrenheit>
  </soap12:Body>
</soap12:Envelope>
```

- Darle a



Ejemplos – SOAP



- Respuesta:

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <soap:Body>
        <CelsiusToFahrenheitResponse xmlns="https://www.w3schools.com/xml/">
            <CelsiusToFahrenheitResult>248</CelsiusToFahrenheitResult>
        </CelsiusToFahrenheitResponse>
    </soap:Body>
</soap:Envelope>
```

Ejemplos – SOAP

Más ejemplos

- Calculadora

http://www.dneonline.com/calculator.asmx

- Información del tiempo

https://graphical.weather.gov/xml/SOAP_server/ndfdXMLserver.php

- Varios

https://documenter.getpostman.com/view/8854915/Szf26WHn?version=latest

Ejemplos – REST

- Paypal REST API

https://developer.paypal.com/docs/api/overview/

- Buscar países

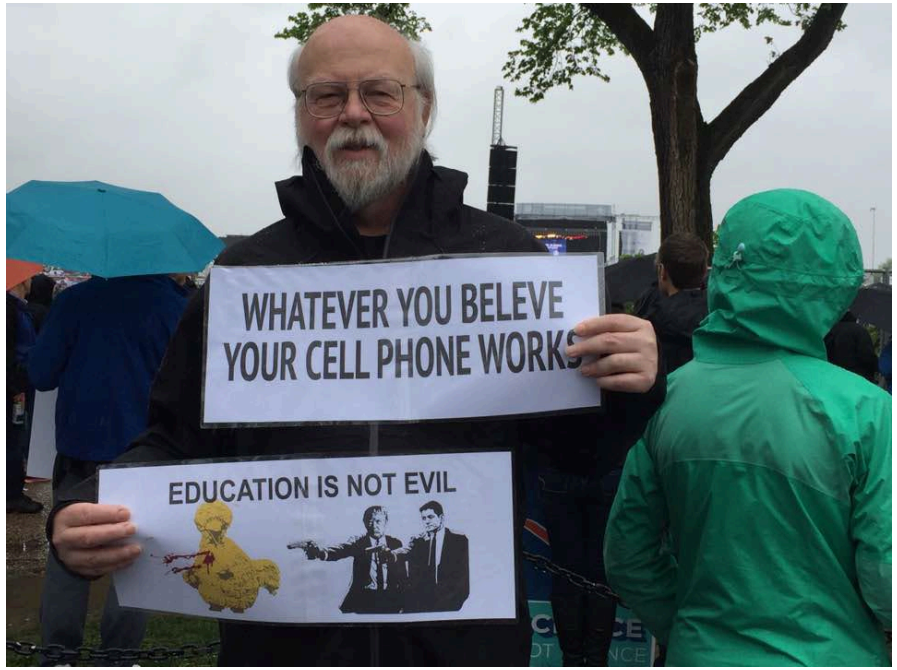
http://www.groupkt.com/post/c9b0ccb9/country-and-other-related-rest-webservices.htm

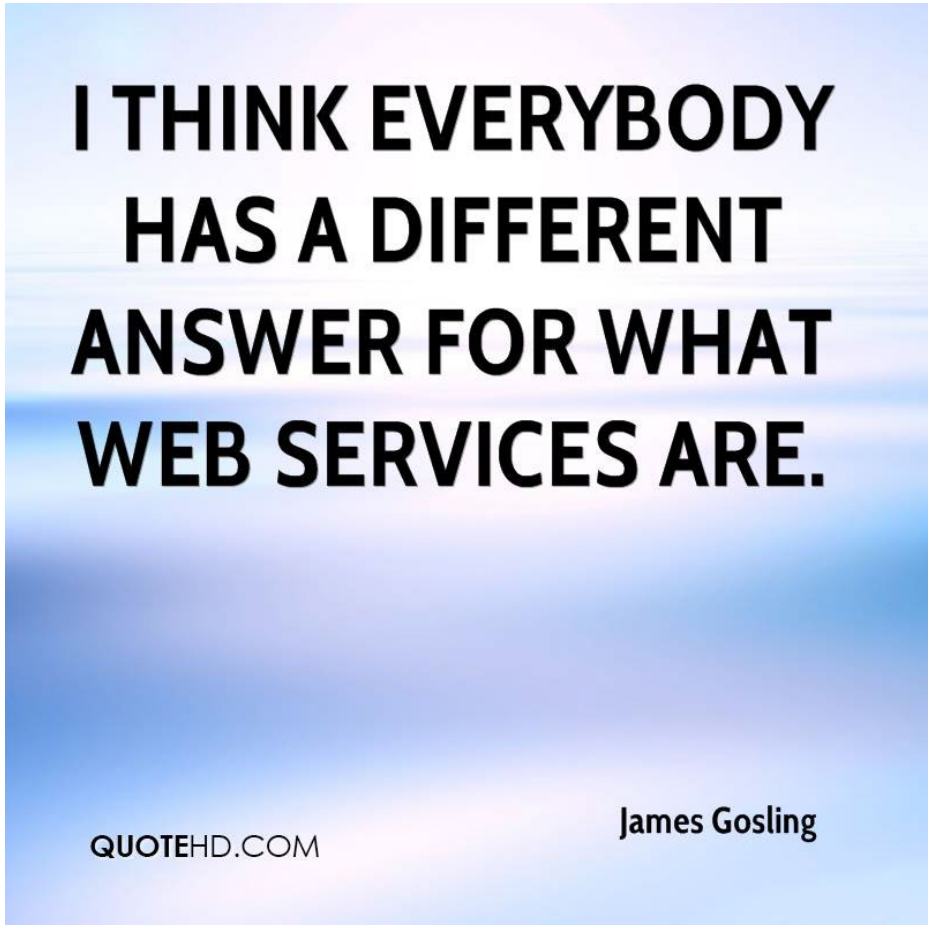
- Posts

https://jsonplaceholder.typicode.com/

- Varios

Dudas


WHATEVER YOU BELEVE
YOUR CELL PHONE WORKS
EDUCATION IS NOT EVIL


I THINK EVERYBODY HAS A DIFFERENT ANSWER FOR WHAT WEB SERVICES ARE.
QUOTEHD.COM
James Gosling

Referencias

- Postman SOAP tutorial

https://learning.postman.com/docs/sending-requests/supported-api-frameworks/making-soap-requests/

- RESTful Web Services

https://www.crummy.com/writing/RESTful-Web-Services/RESTful_Web_Services.pdf

- Definición de API

https://oai.github.io/Documentation/introduction.html