PREGRADO

UNIDAD 3 | HIGH-LEVEL SOFTWARE DESIGN & PATTERNS

REST ARCHITECTURAL STYLE

SI720 | Diseño y Patrones de Software

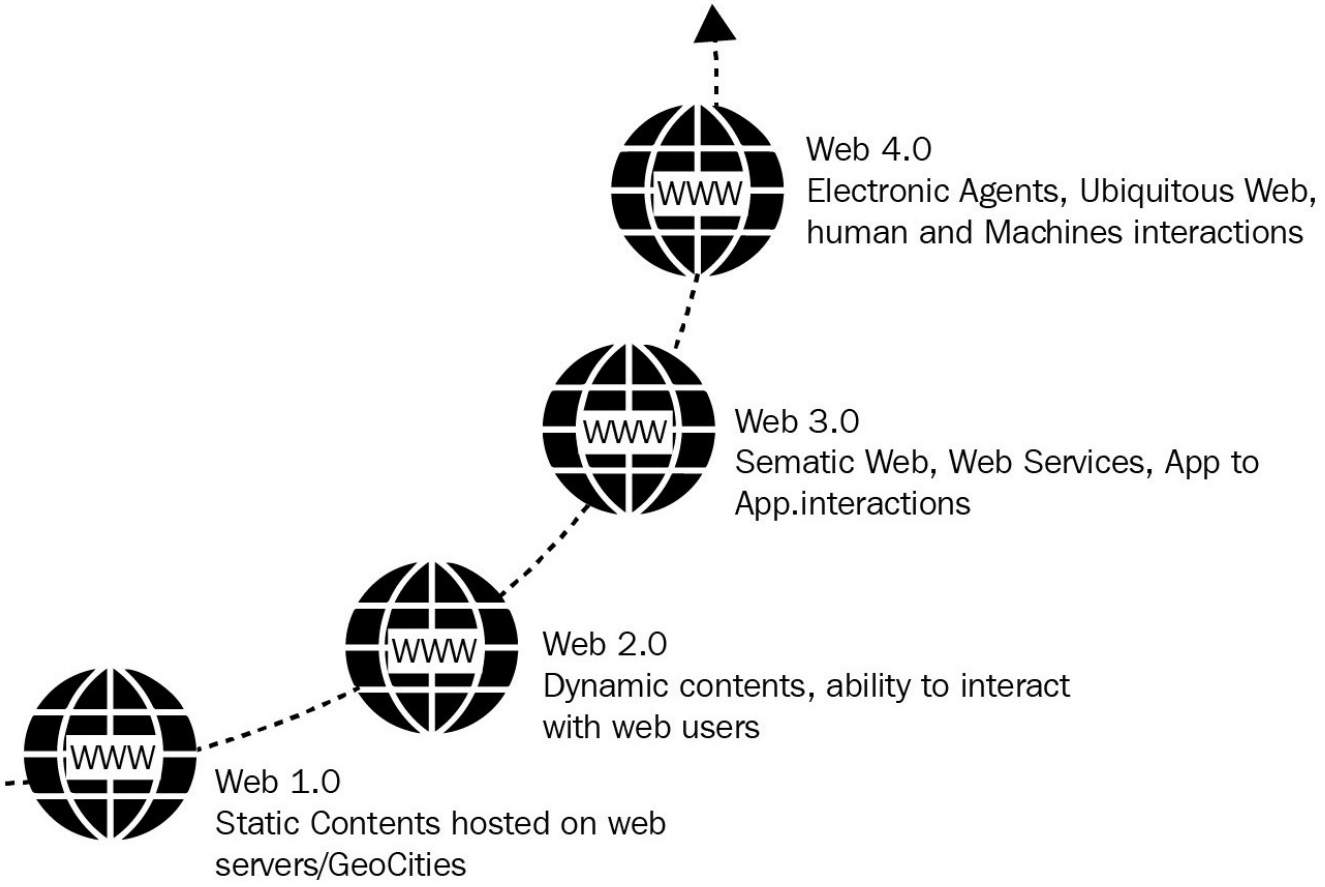
Al finalizar la unidad de aprendizaje, el estudiante elabora artefactos de diseño de software aplicando principios básicos y high-level design patterns, bajo un enfoque Domain-Driven, para un dominio y contexto determinados, utilizando lenguajes de modelado, verificando y validando la solución, considerando aspectos de code reuse y documentando los elementos de la solución.

AGENDA

INTRO
CONSTRAINTS
RESTFUL

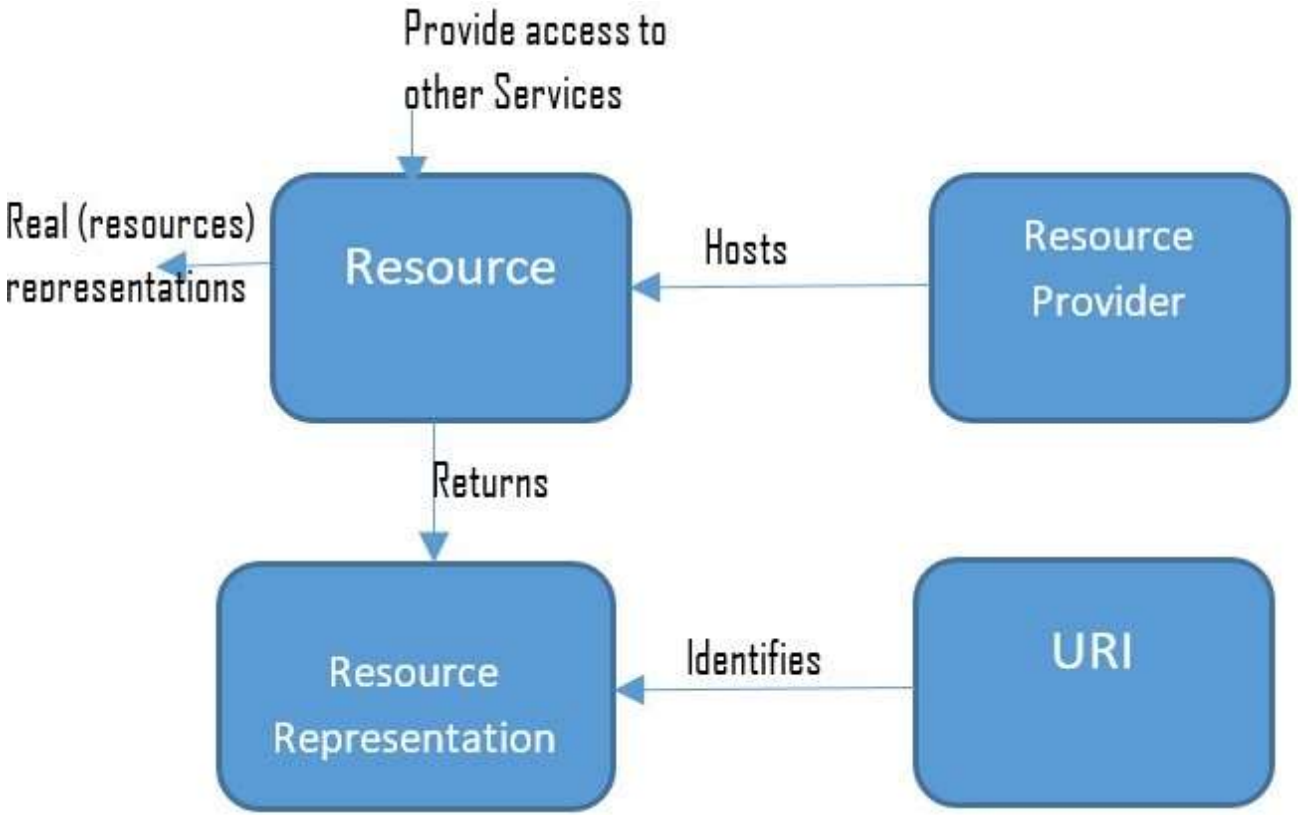
Evolution of web technologies

Resource-Oriented Architecture (ROA)



Resource-oriented architecture

La arquitectura orientada a los recursos es una base de la web semántica. La idea de ROA es utilizar tecnologías web básicas, bien entendidas y conocidas (HTTP, URI y XffiL) junto con los principios básicos de diseño.



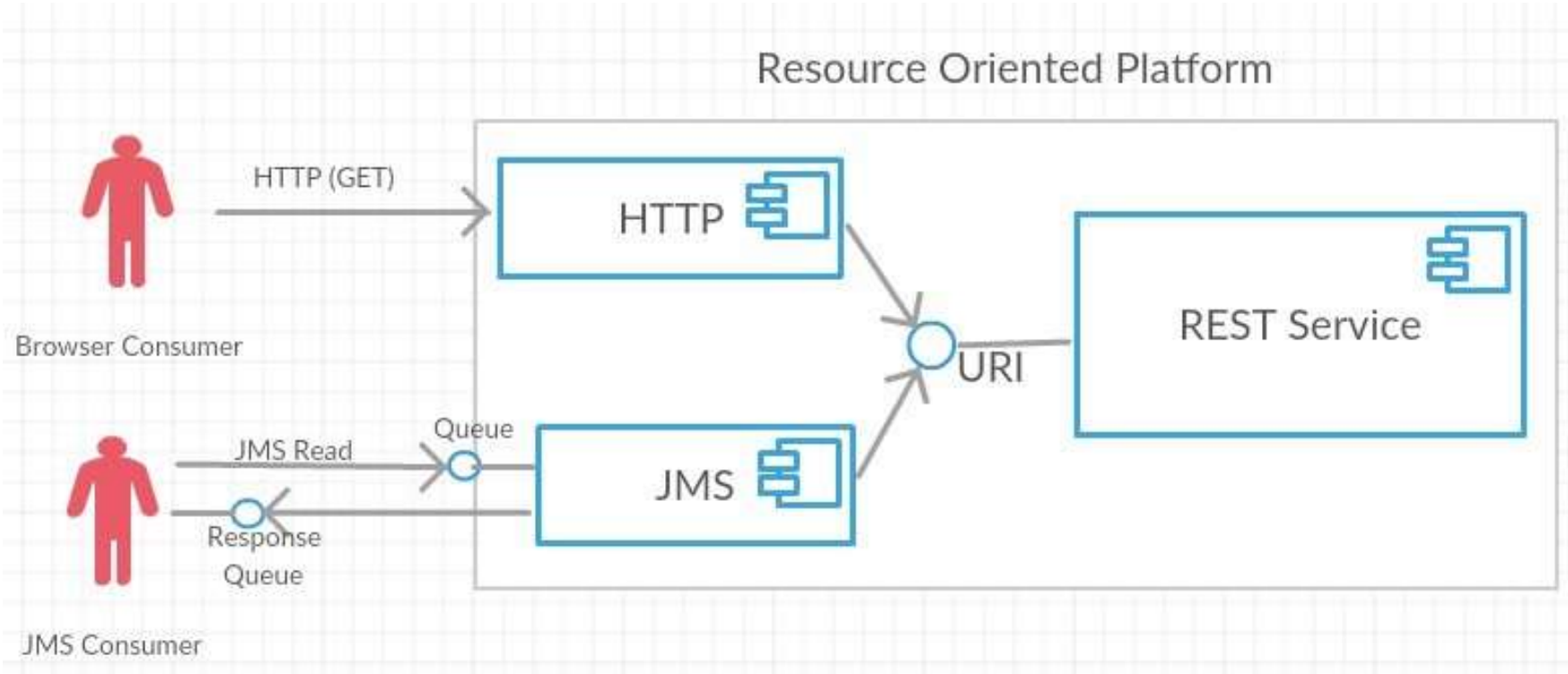
Resource-oriented architecture

La siguiente tabla resume las operaciones HTTP que se pueden usar para implementar un servicio web basado en ROA:

HTTP operation	Description
GET	Read the resource representations
PUT	Create a new resource
DELETE	Delete the resource (optionally linked resource as well)
POST	ffiodify the resource
HEAD	ffieta information of the resource

Resource-oriented design - REST

ROA/REST service:

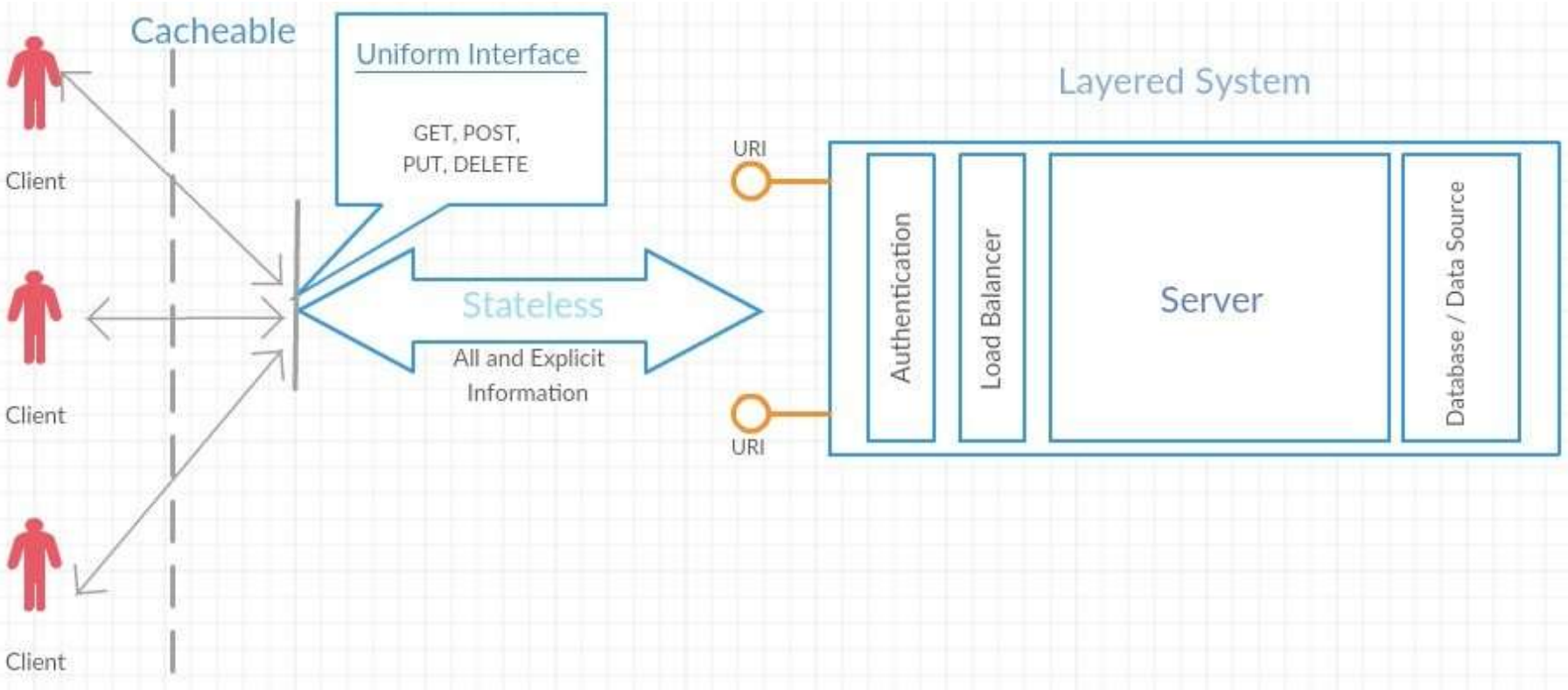


AGENDA

INTRO
CONSTRAINTS
RESTFUL

REST architecture style constraints

Características del estilo arquitectónico REST, restricciones REST:



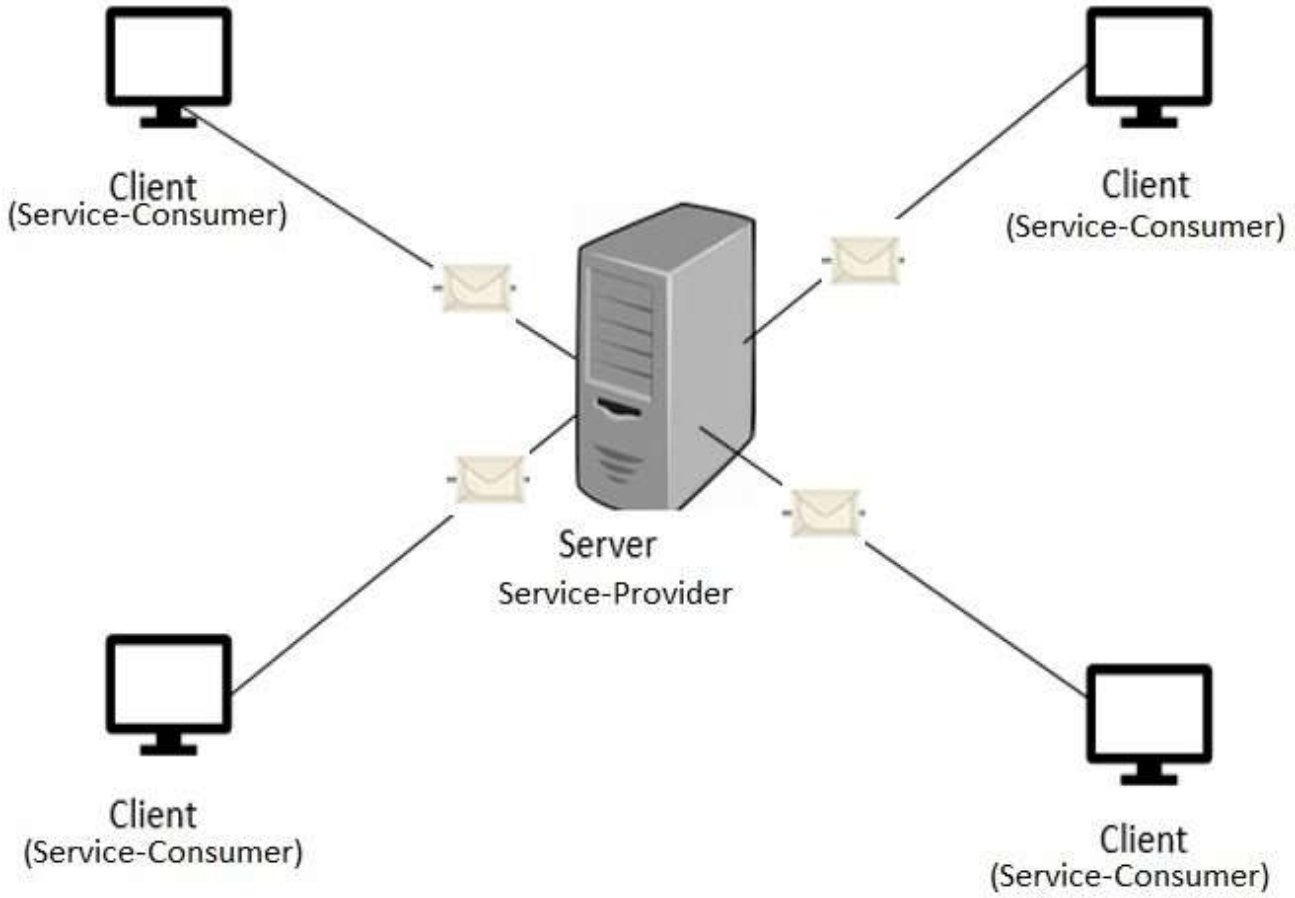
REST architectural style constraints

Las siguientes son las restricciones REST:

- Client-server
- Statelessness
- Cacheable
- Uniform interface
- Layered systems
- Code on demand

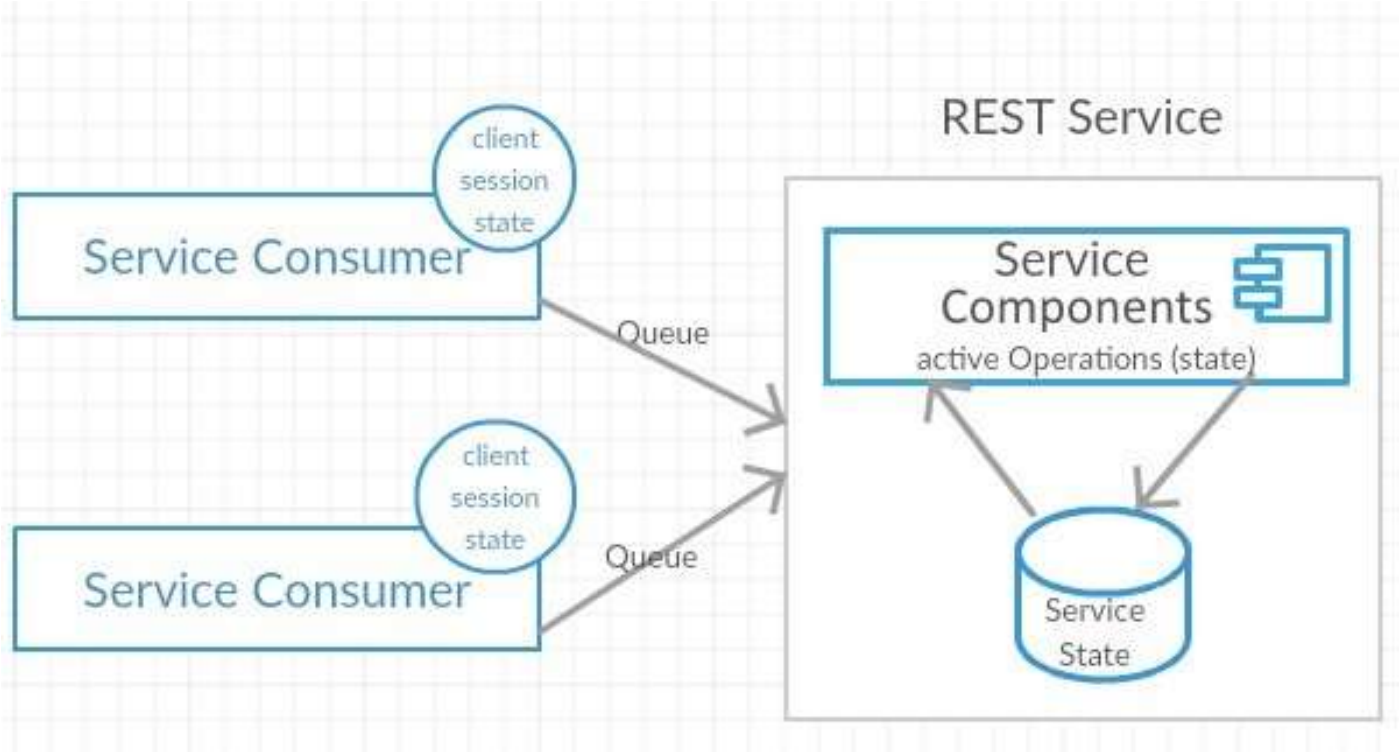
Client-server

La arquitectura o modelo cliente-servidor ayuda a separar las preocupaciones entre la interfaz de usuario y el almacenamiento de datos:



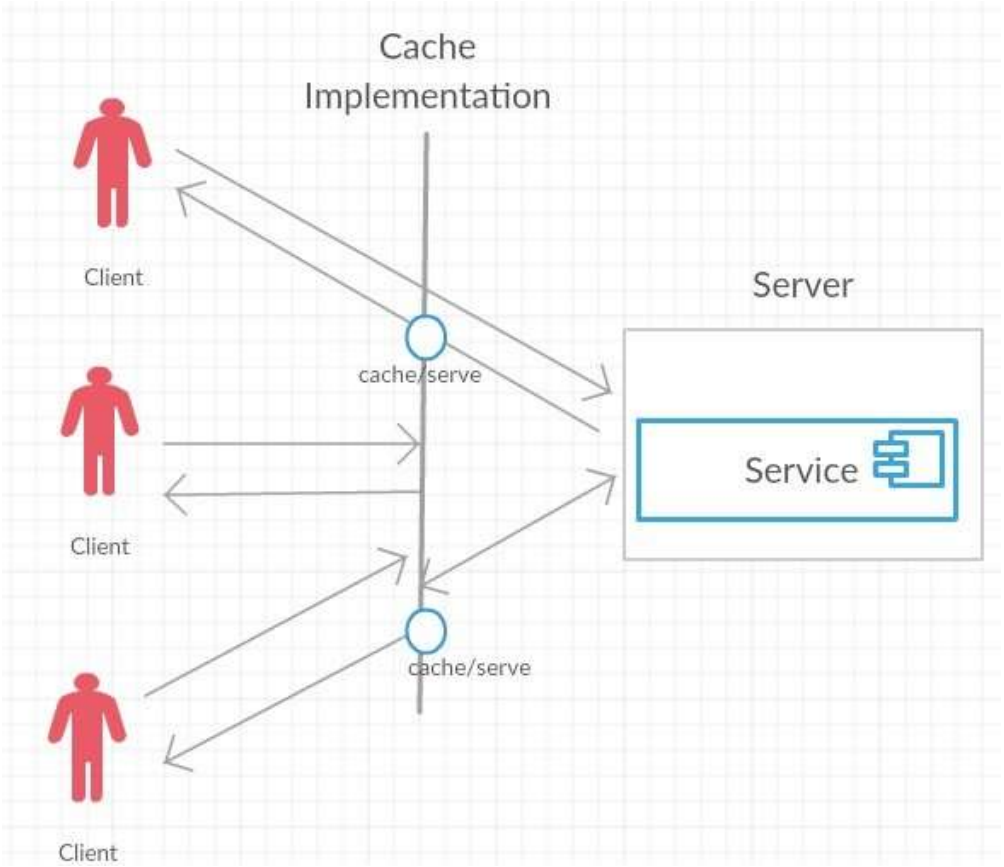
Statelessness

Ayuda a que los servicios sean más escalables y confiables.
Significa que todas las solicitudes del cliente al servidor llevan toda la información como explícita (stated), de modo que el servidor comprende las solicitudes, las trata como independientes y esas solicitudes del cliente mantienen al servidor independiente de cualquier información almacenada. contextos



Caching

El almacenamiento en caché es la capacidad de almacenar datos a los que se accede con frecuencia (una respuesta en este contexto) para atender las solicitudes del cliente, y nunca tener que generar la misma respuesta más de una vez hasta que sea necesario.



Caching

Existen diferentes estrategias o mecanismos de almacenamiento en caché disponibles, como cachés de navegador, cachés de proxy y cachés de puerta de enlace (proxy inverso), y hay varias formas de controlar el comportamiento de la caché, como pragma, etiquetas de caducidad, etc. .

Headers	Description
Expires	Atributo de encabezado para representar la fecha /hora después de la cual la respuesta se considera obsoleta
Cache-control	Un encabezado que define varias directivas (tanto para solicitudes como para respuestas) seguidas de mecanismos de almacenamiento en caché
E-Tag	Identificador único para estados de recursos del servidor
Last-modified	El encabezado de respuesta ayuda a identificar la hora en que se generó la respuesta

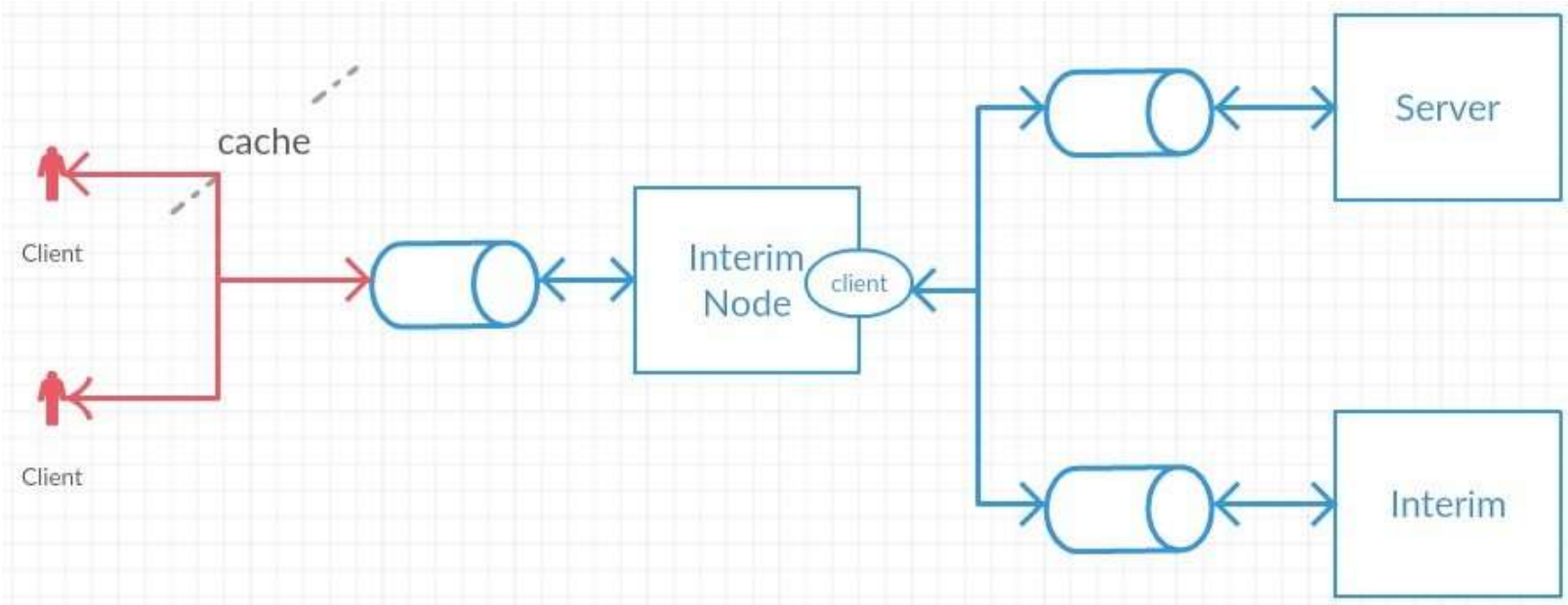
Uniform Interface

Los servicios basados en REST pueden usar la interfaz HTTP, como GET, POST, PUT, DELETE, etc., para mantener la uniformidad en la web. La intención de una interfaz uniforme es retener un vocabulario común en Internet.

El siguiente diagrama representa la combinación de HTTP ffiethods y los Resource Names para Uniform Interfaces:

Layered systems

Un sistema en capas consta de capas con diferentes unidades de funcionalidad. Las características esenciales de los sistemas en capas son que una capa se comunica por medio de interfaces predefinidas y se comunica solo con la capa superior o la capa inferior, y las capas superiores dependen de las capas inferiores para realizar sus funciones.



Code on demand

Es cualquier tecnología que permite al servidor enviar el código de software a los clientes para que se ejecute en la computadora del cliente a solicitud del software del cliente. Algunos ejemplos bien conocidos del paradigma COD en la web son los applets de Java, el lenguaje Adobe ActionScript para Flash Player y JavaScript.

AGENDA

INTRO
CONSTRAINTS
RESTFUL

RESTful

Web Services basados en arquitectura REST.

Implementan interacción basada en métodos HTTP

Aplican URI (Uniform Resource Identifier)

Representación de información en formato JSON

RESTful API

Application Programming Interface

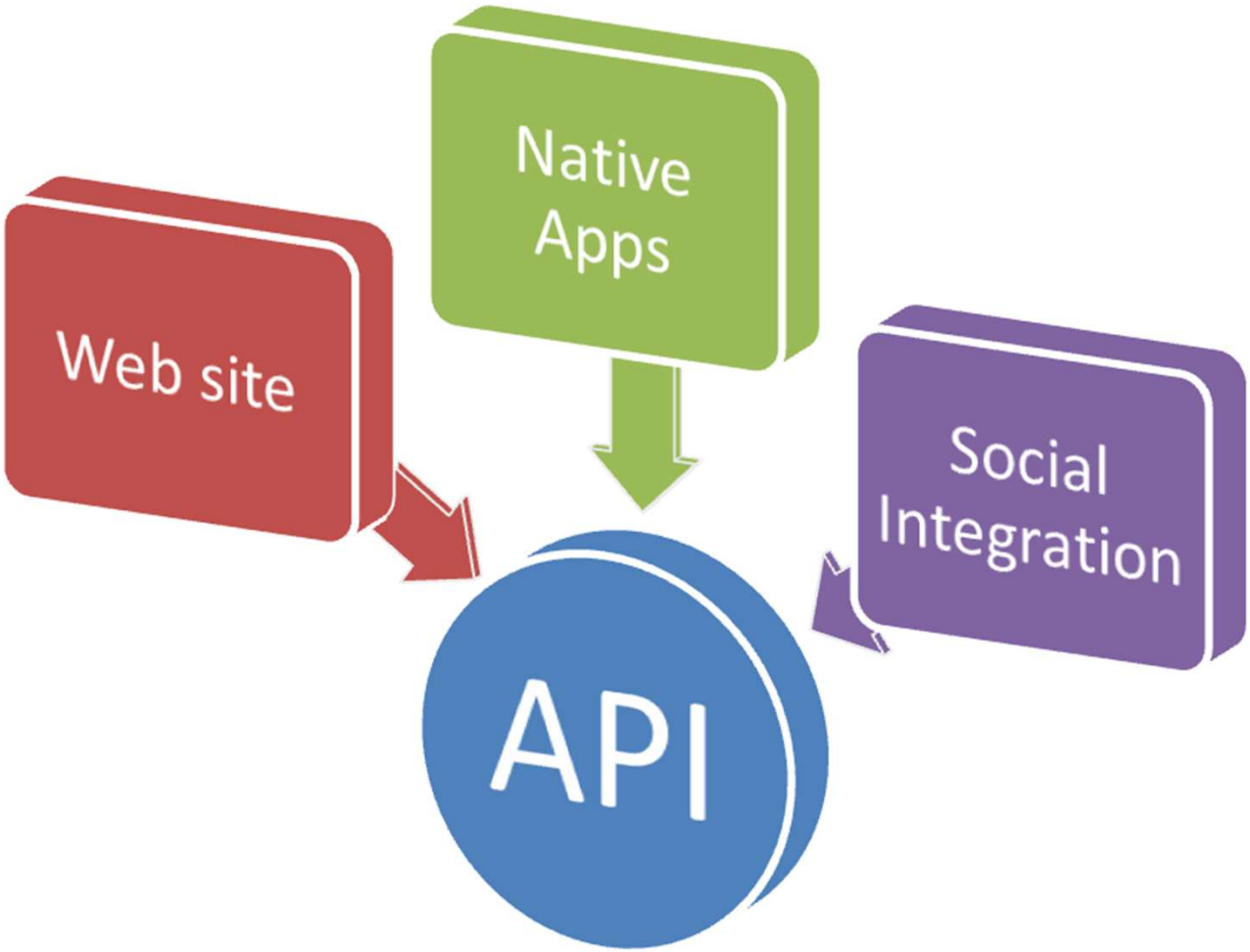
Descompone una transacción en pequeños

módulos

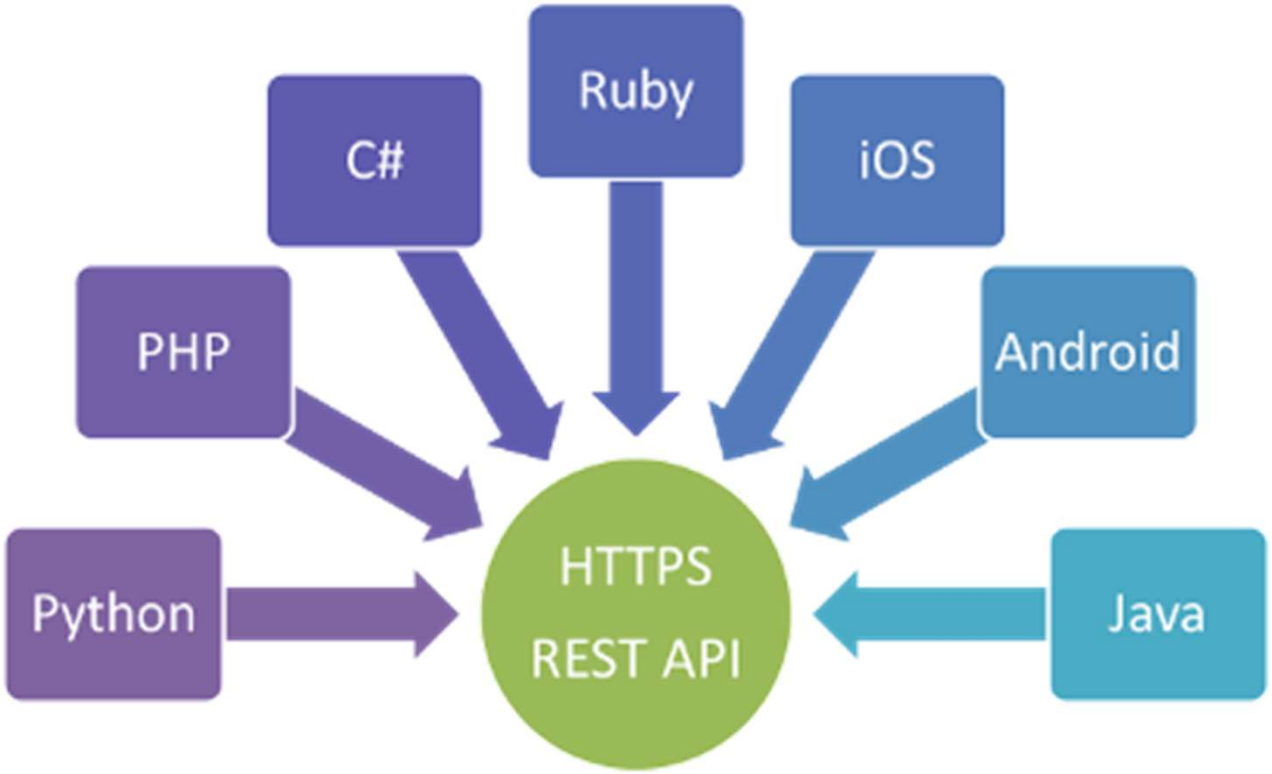
Cada módulo gestiona una parte de la transacción.

Aplica métodos basados en HTTP (RFC 2616)

RESTful API



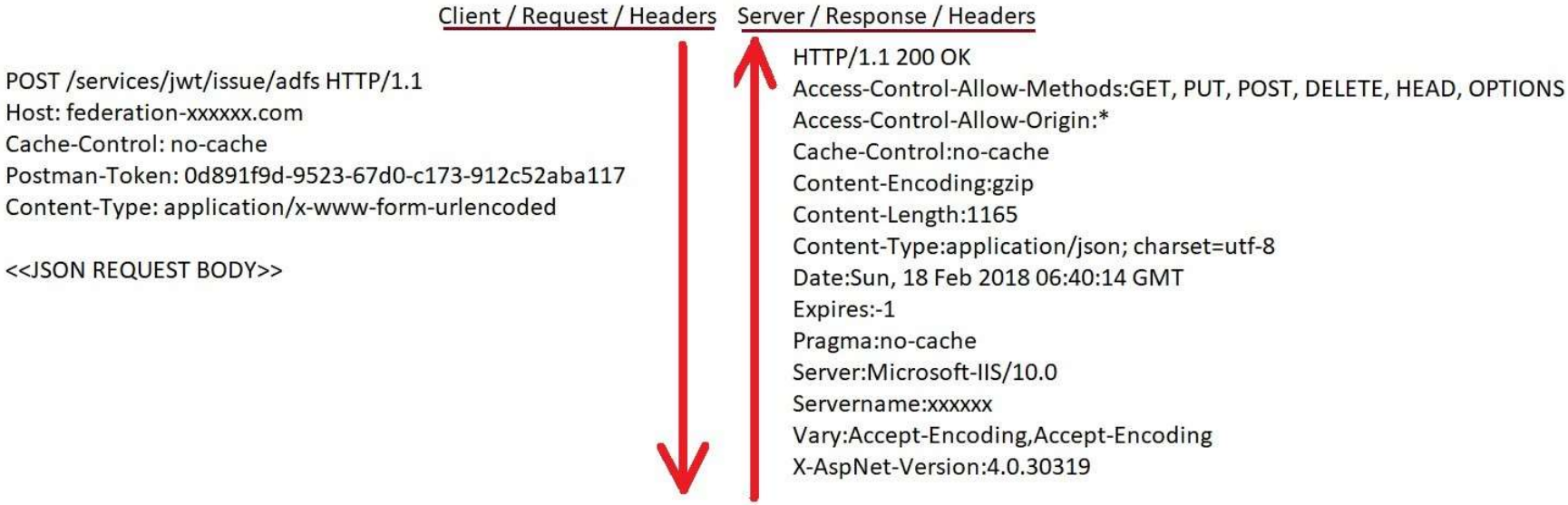
RESTful API



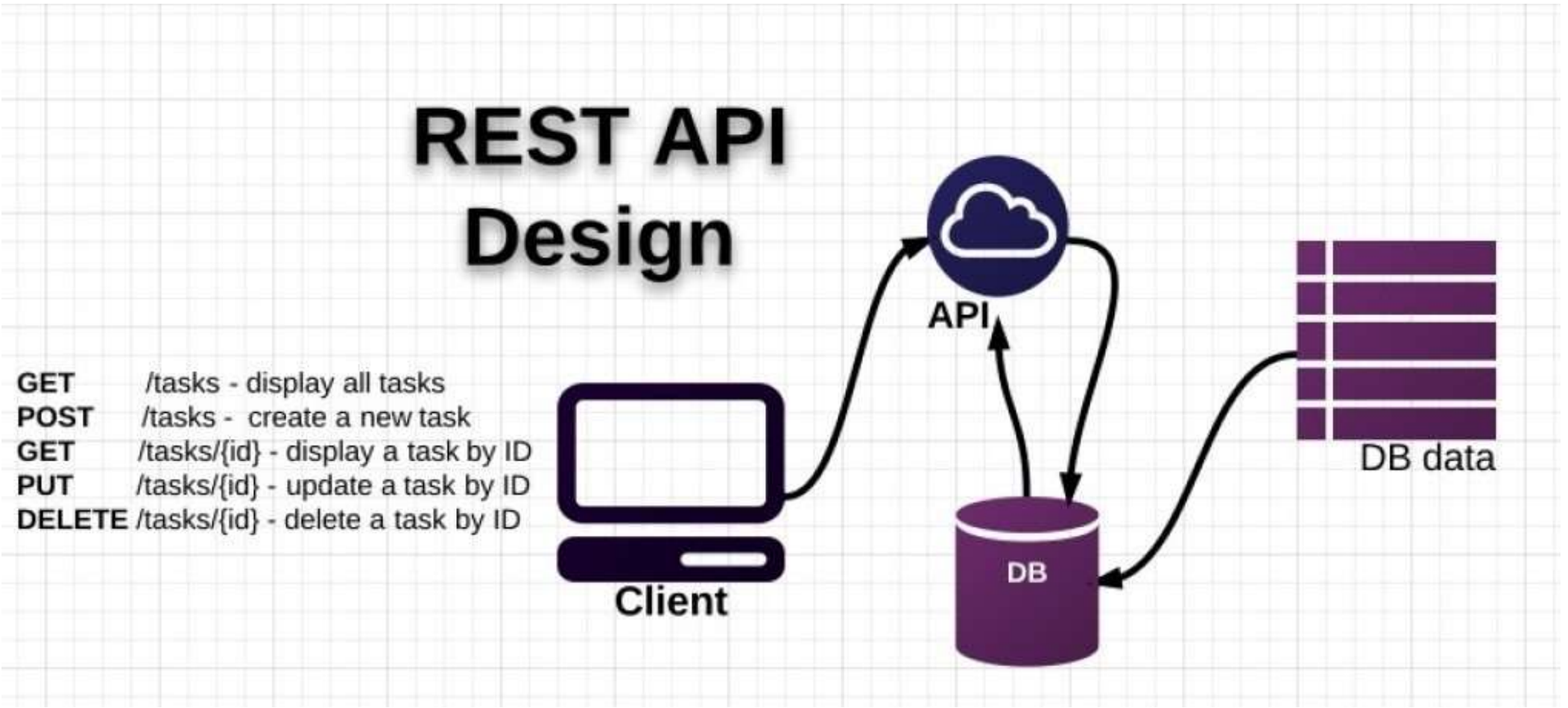
Resource-oriented design - REST

La solicitud de un cliente y la respuesta del servidor son mensajes; esos mensajes deben ser stateless y self-descriptive. Pueden tener un cuerpo y metadatos.

Las aplicaciones RESTful funcionan con la noción de tipos de mensajes restringidos (GET, HEAD, OPTIONS, PUT, POST y DELETE) y el servidor y el cliente los entienden completamente.



RESTful API


REST API
Design
GET /tasks - display all tasks
POST /tasks - create a new task
GET /tasks/{id} - display a task by ID
PUT /tasks/{id} - update a task by ID
DELETE /tasks/{id} - delete a task by ID
API
Client
DB
DB data

RESTful API

HTTP ffiethods

HTTP Verb	CRUD	Entire Collection (e.g. /customers)	Specific Item (e.g. /customers/{id})
POST	Create To create new subordinate resource	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read To retrieve resource representation/information only	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update/Replace To update existing resource	405 (ffiethod Not Allowed), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
PATCH	Update/ffiodify To make partial update on a resource	405 (ffiethod Not Allowed), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete To delete resources	405 (ffiethod Not Allowed), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

JSON

JavaScript Object Notation

JSON

Colección de pares nombre /valor

Lista ordenada de valores

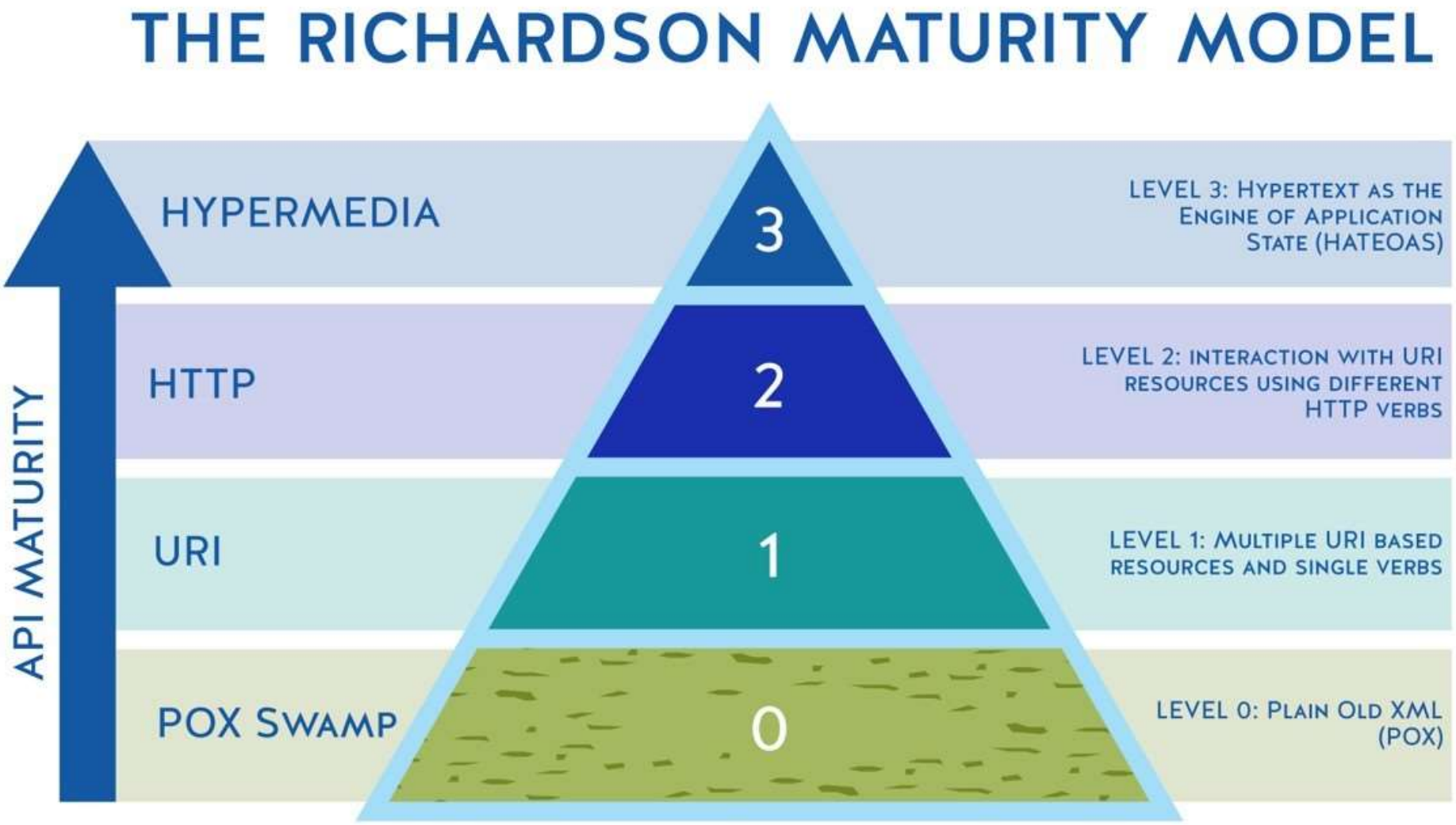
```
{"customers":[
    {"firstName":"John", "lastName":"Doe"},
    {"firstName":"Anna", "lastName":"Smith"},
    {"firstName":"Peter", "lastName":"Jones"}
]}
```

JSON Files

Extensión .JSON

ffiIffiE Type “application/json”

Richardson ffiaturity ffiodel



HATEOAS Driven REST APIs

HATEOAS (Hypermedia as the Engine of Application State)

Es una restricción de la arquitectura de la aplicación.

Hypermedia se refiere que contiene enlaces a otras formas de media como images, movies, text.

Por ejemplo: `HTTP GET` http://api.domain.com/management/departments/10

```
{
    "departmentId": 10,
    "departmentName": "Administration",
    "locationId": 1700,
    "managerId": 200,
    "links": [
        {
            "href": "10/employees",
            "rel": "employees",
            "type" : "GET"
        }
    ]
}
```

https://restfulapi.net/hateoas/

REST Resource Naming

REST Resource Naming Guide

https://restfulapi.net/resource-naming/

REST API Versioning

URI Versioning

https://example.com/api/v1
https://api.example.com/v1
https://apiv1.example.com

Otros tipos

https://restfulapi.net/versioning/

RESUMEN

Recordemos

RESTful viene de REpresentational State Transfer

A nivel de representación de información de web services, uno de las principales combinaciones es RESTful API + JSON

REFERENCIAS

Para profundizar

https://restfulapi.net/

https://martinfowler.com/articles/richardsonffiaturityffiodel.html

PREGRADO

Ingeniería de Software

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería

UPC
Universidad Peruana
de Ciencias Aplicadas

Prolongación Primavera 2390,
Monterrico, Santiago de Surco
Lima 33 - Perú
T 511 313 3333
https://www.upc.edu.pe

exígete, innova