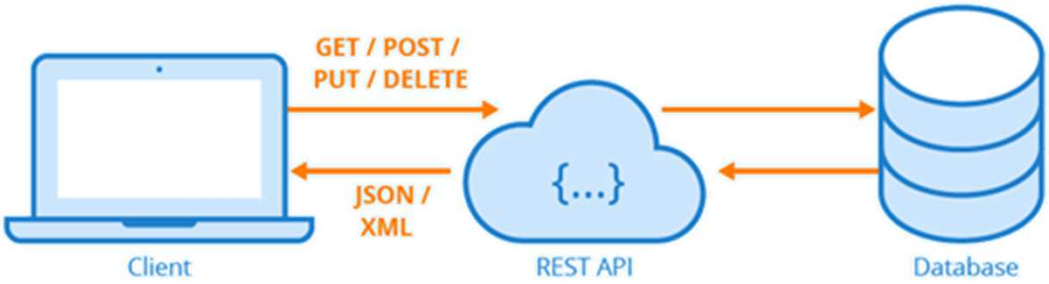
SISTEMAS WEB
CURSO 2021/2022

APIs Web

Delegación de autenticación y autorización: OAuth

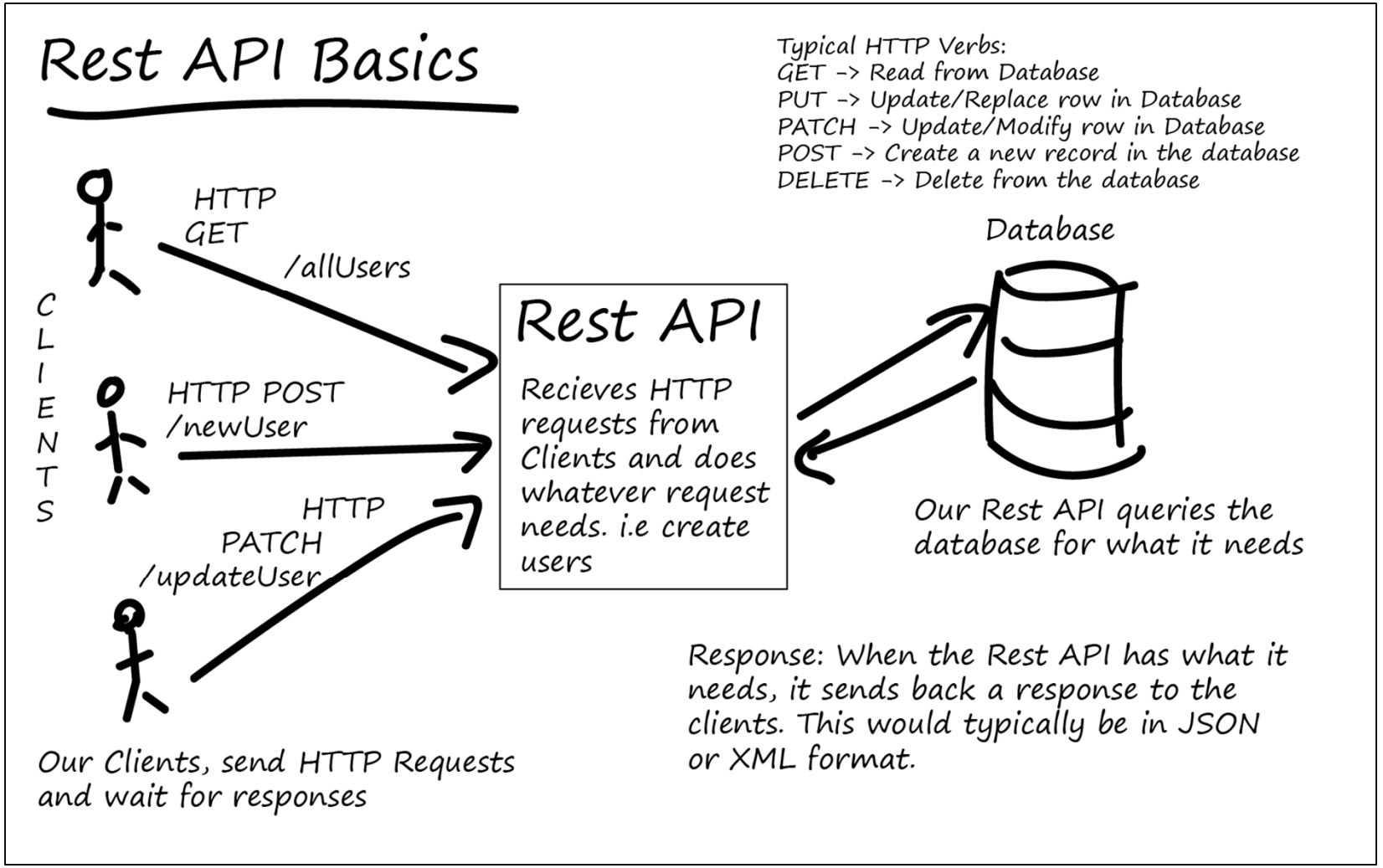
Oskar Casquero y María Luz Álvarez
Dpto. Ingeniería de Sistemas y Automática

APIs Web



HTTP method	Action	Example
GET	Obtain info about a resource	http://example.com/api/orders (retrieve order list)
GET	Obtain info about a resource	http://example.com/api/orders/123 (retrieve order #123)
POST	Create a new resource	http://example.com/api/orders (créate a new order from data provided with the request)
PUT/PATCH	Update a resource	http://example.com/api/orders/123 (update order #123 from data provided with the request)
DELETE	Delete a resource	http://example.com/api/orders/123 (delete order #123)

APIs Web

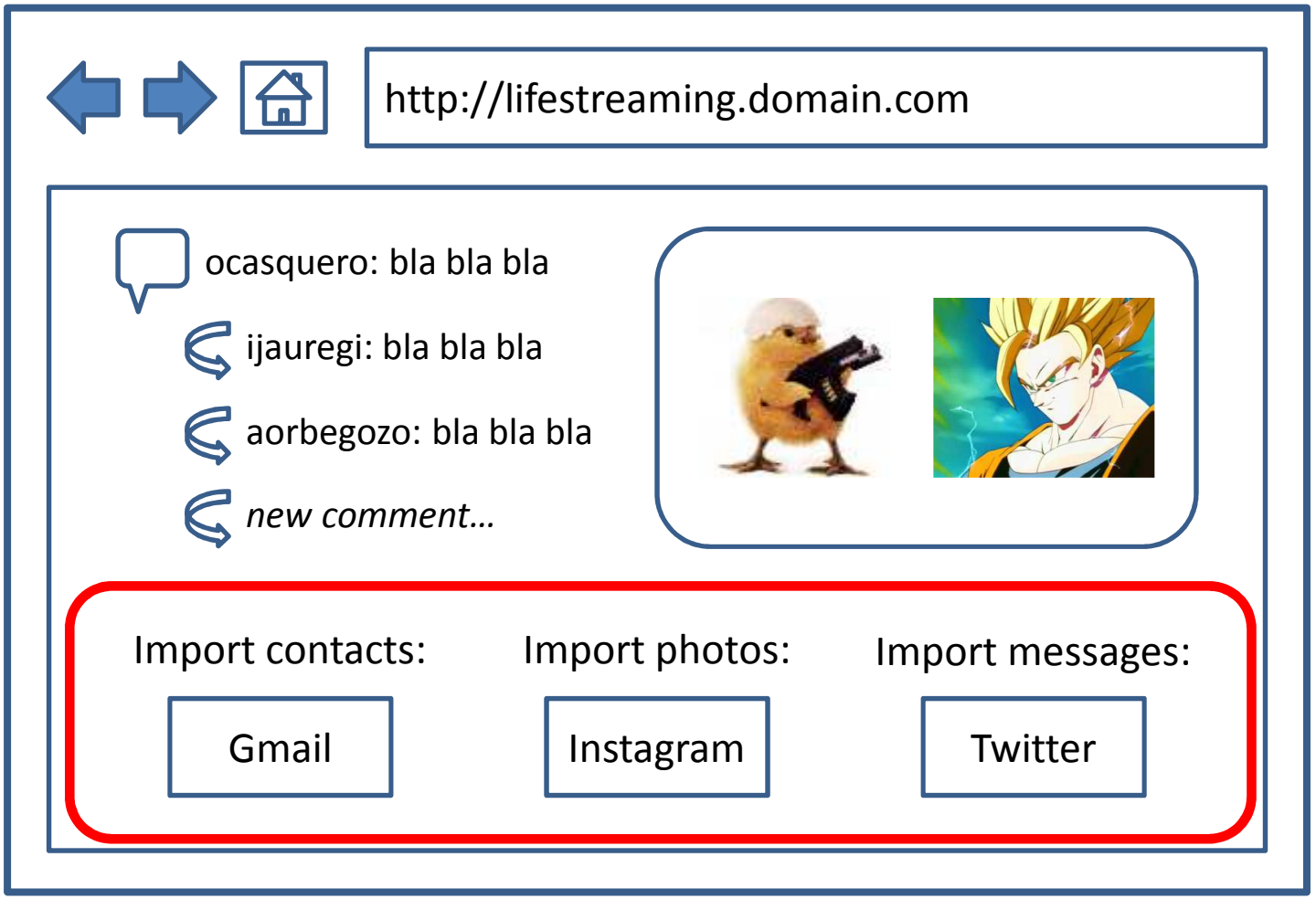

Rest API Basics
Typical HTTP Verbs:
GET -> Read from Database
PUT -> Update/Replace row in Database
PATCH -> Update/Modify row in Database
POST -> Create a new record in the database
DELETE -> Delete from the database
HTTP
GET
/allUsers
CLIENTS
HTTP POST
/newUser
HTTP
PATCH
/updateUser
Rest API
Recieves HTTP requests from Clients and does whatever request needs. i.e create users
Database
Our Rest API queries the database for what it needs
Response: When the Rest API has what it needs, it sends back a response to the clients. This would typically be in JSON or XML format.
Our Clients, send HTTP Requests and wait for responses

DELEGACIÓN DE AUTENTICACIÓN Y AUTORIZACIÓN
SUPONGAMOS…

- que estamos desarrollando una aplicación.
- Los usuarios de nuestra aplicación son usuarios de otras aplicaciones web: google calendar, google drive, dropbox, twitter, …

- **Objetivo**: que los usuarios de nuestra aplicación puedan utilizar los datos que tienen en las otra aplicaciones :
  - Google Calendar: calendarios, citas
  - Google Drive y Dropbox: ficheros
  - Twitter: tuits

DELEGACIÓN DE AUTENTICACIÓN Y AUTORIZACIÓN
PROBLEMA 1



DELEGACIÓN DE AUTENTICACIÓN Y AUTORIZACIÓN
PROBLEMA 1

DELEGACIÓN DE AUTENTICACIÓN Y AUTORIZACIÓN
PROBLEMA 2

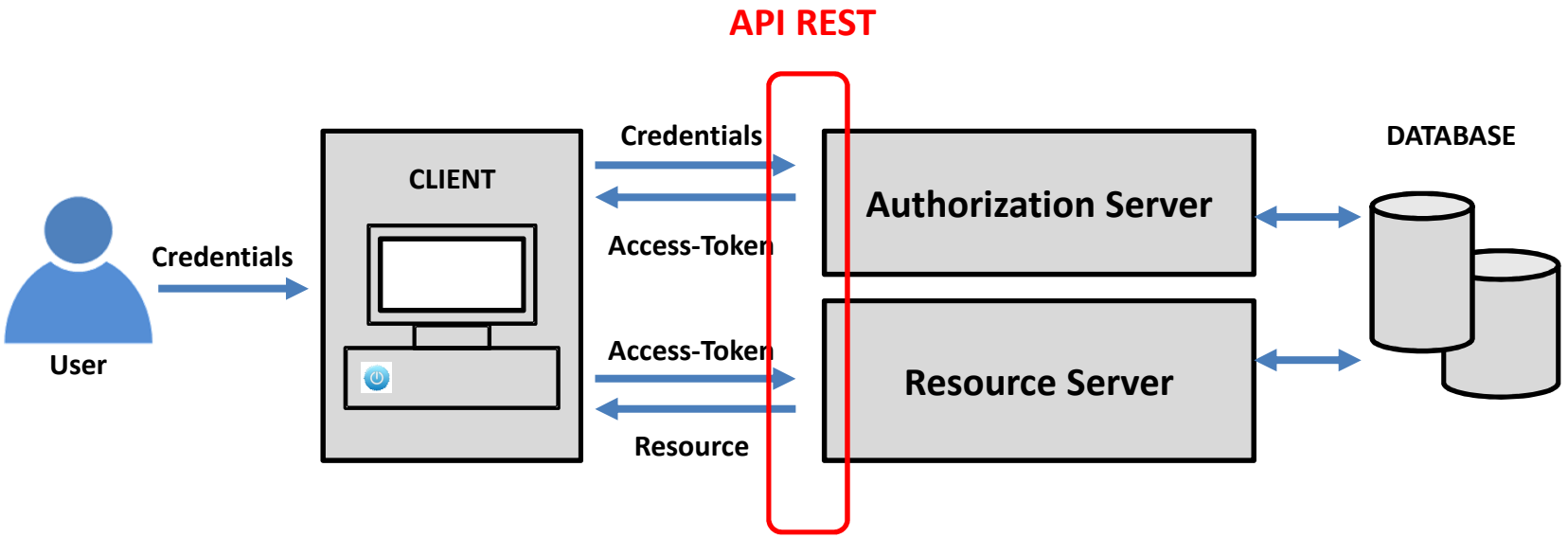
- **Autenticación** vs. **Permiso**
- La cuenta de Google nos da la posibilidad de entrar y utilizar un gran número de servicios: contacts, mail, calendar, docs, …
- Una vez autenticada, la aplicación del ejemplo anterior, lifestreaming.domain.com
  - ¿En que servicios puede entrar?
    - contactos
  - ¿Que puede hacer en este servicio?
    - Leer contactos pero no puede generar nuevos contactos.

SOLUCIÓN: OAUTH

Llave *Valet* : Esta llave especial sólo permite conducir el coche una corta distancia, bloquea el acceso al maletero o al teléfono móvil de a bordo. La idea es **darle a alguien el acceso limitado a el coche.**
Credenciales. Pero, la llave no se la das a cualquiera, en un restaurante se la das al aparcacoches si acredita que trabaja en el restaurante

- El protocolo OAuth permite a un usuario de una web A acceder a los datos que este usuario tiene en una web B.
- En este proceso, se utiliza un mecanismo indirecto para que la web A no vea el nombre de usuario y contraseña de la web B.

SOLUCIÓN: OAUTH



VERSIONES OAUTH : 2.0 Y 1.0

- OAuth 2.0: RFC 6749
  - Usuarios: Google, Amazon, Dropbox, GitHub, Facebook, Instagram, Strava, LinkedIn, …
- OAuth 1.0: RFC 5849
  - Usuarios: Twitter, Flickr, …

- Diferencia principal entre las dos versiones:
  - En OAuth 1.0 la petición HTTP va firmada

- Sitio web oficial: http://oauth.net

EJEMPLO: CALENDARIO DE GOOGLE

- Programamos un cliente Python.
- Este cliente conseguirá la lista calendarios Google de un usuario utilizando OAuth for mobile & desktop apps.