Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

Aspectos Finales

IIC2513 - Tecnologías y Aplicaciones Web

Sebastián Vicencio R.
2do Semestre 2020

CORS

Cross-Origin Resource Sharing

CORS

Contexto

Partamos con un ejemplo práctico:

React + API

¿Por qué pasó esto?

Same origin policy

Seguridad en browsers

Modelo de seguridad fundamental de browsers

Origin: tupla "protocol + host + port"

Browser tiene un mecanismo de seguridad que permite acceder sólo a recursos que son del mismo origin

Same origin policy

Seguridad en browsers

Excepciones

- <img>
- <script>
- <link>
- <video> <audio>

Cross-Origin Resource Sharing

¿Qué es?

Mecanismo que permite acceder a recursos cross-origin

Está basado en la utilización de headers HTTP, que permiten este acceso

Cross-Origin Resource Sharing

¿Qué es?

Dos escenarios

- **Request simples**: GET, POST (con ciertos content-types)
- **Requests preflight**: todos los demás métodos que podrían causar side effects
  - Involucra un request previo extra

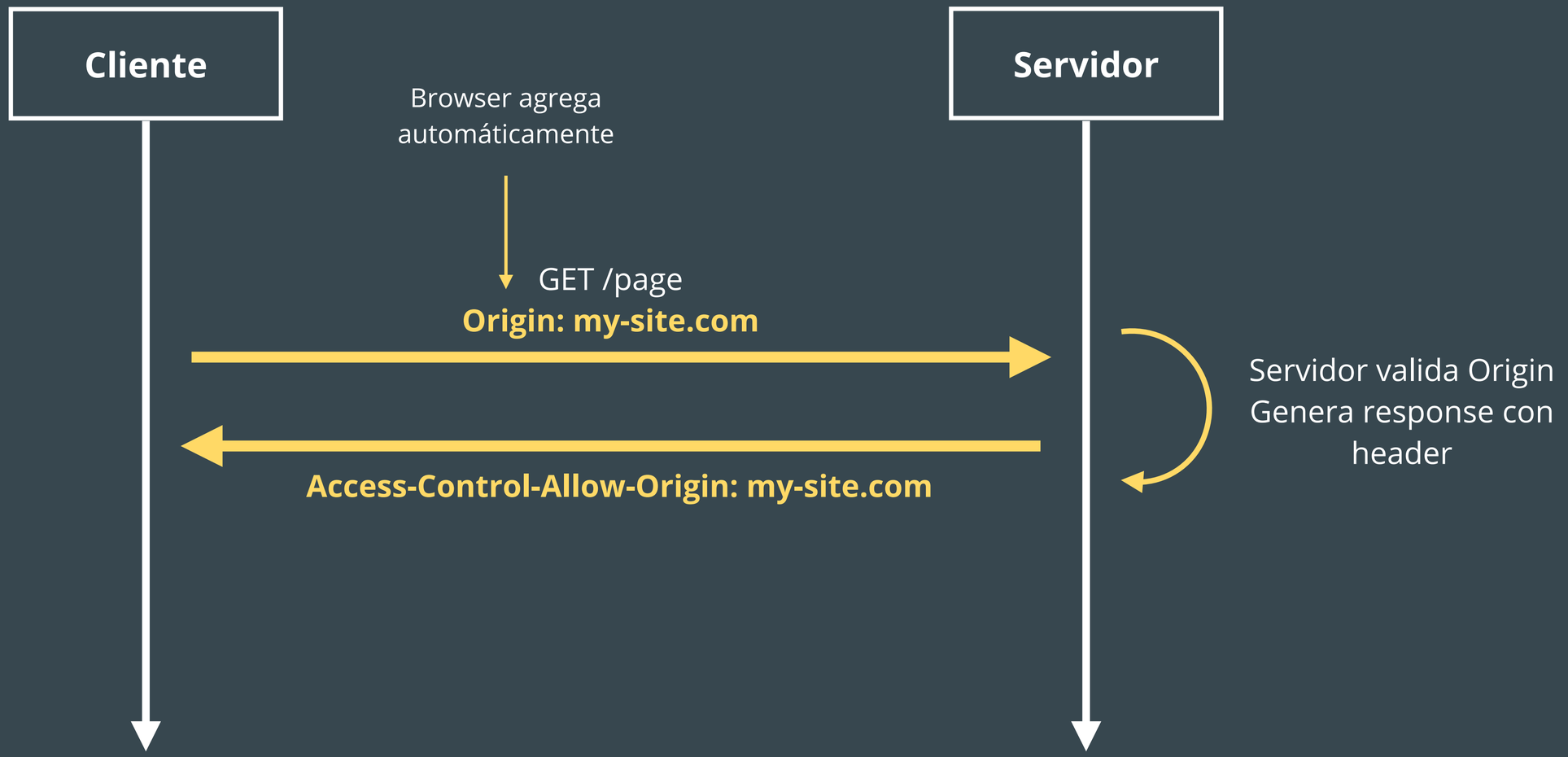
Request simple CORS

¿Qué es?

Request que cumple con las siguientes condiciones:

- **Métodos**: GET, POST, HEAD
- **Headers**: un grupo reducido dentro de los cuales están:
  - Content-Type
  - Accept
- **Content-Type** sólo puede tener los valores:
  - application/x-www-form-urlencoded
  - multipart/form-data
  - text/plain

Request simple CORS



Request preflight CORS

¿Qué es?

Request que ocurre cuando no se cumple alguna condición de request simple

Browser envía primero un request HTTP con método OPTIONS para determinar si es posible enviar el request original

Request preflight CORS

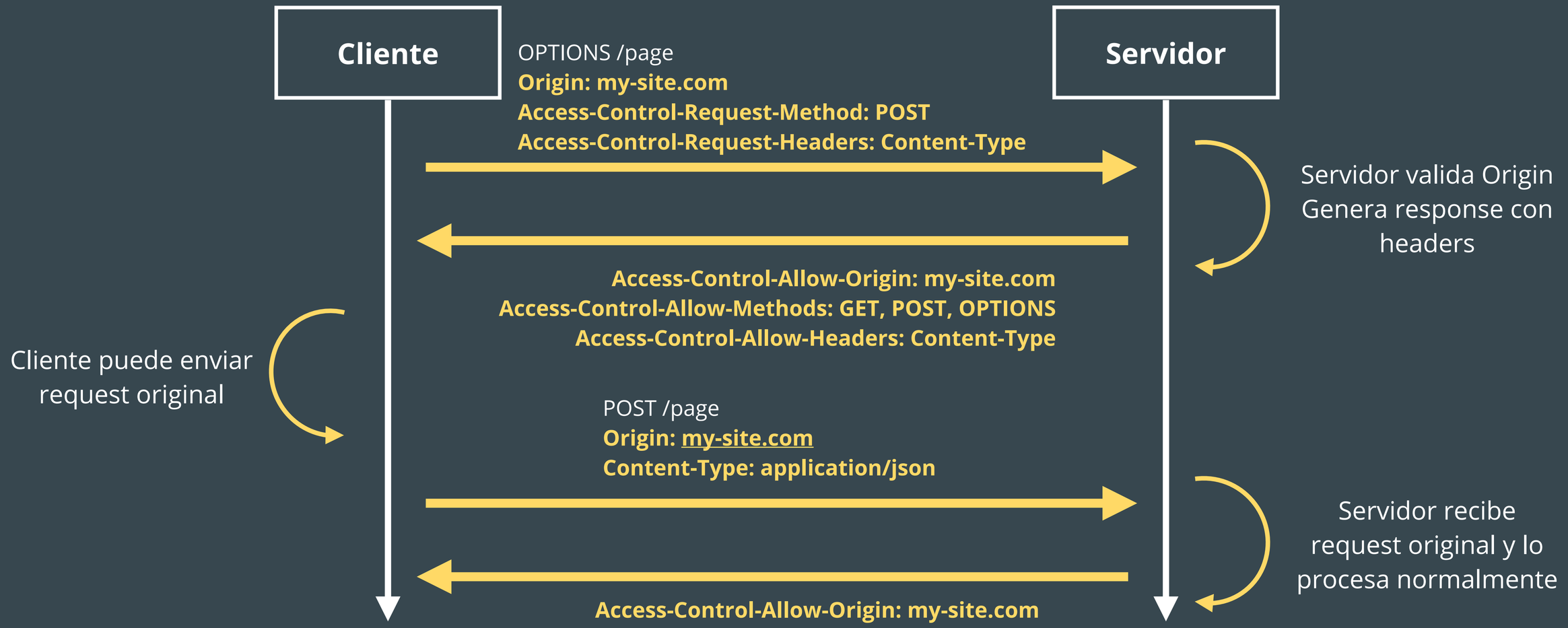
¿Qué incluye?

Request con información en headers del request original

- `Access-Control-Request-Method`
- `Access-Control-Request-Headers`

Browser determina automáticamente qué valores enviar en headers basado en el request original

Request preflight CORS



Request preflight CORS

Response

Servidor debe determinar si responde o no a un request bajo esas circunstancias

Incluye headers que indicarán qué tipo de request está permitido por el servidor

- `Access-Control-Allow-Origin`
- `Access-Control-Allow-Methods`
- `Access-Control-Allow-Headers`

Configuración CORS

Por el lado del backend

Configurar 3 cosas

- Procesar **requests** que vengan con **header Origin**
- Responder **requests simples** con headers adecuados
- Responder **requests preflight** (método OPTIONS) con headers adecuados

Configuración CORS

Por el lado del backend

En koa existe @koa/cors

```
const koaCors = require('@koa/cors');

app.use(koaCors());
```

Dominios

my-domain.com

Dominio

¿Qué es?

Es un identificador único y human-friendly de un sitio web

Apunta a una dirección IP

`http://my.example.org/general/specific`

subdomain second-level domain top-level domain

DNS

¿Qué es?

¿Cómo el browser sabe a qué IP corresponde un dominio?

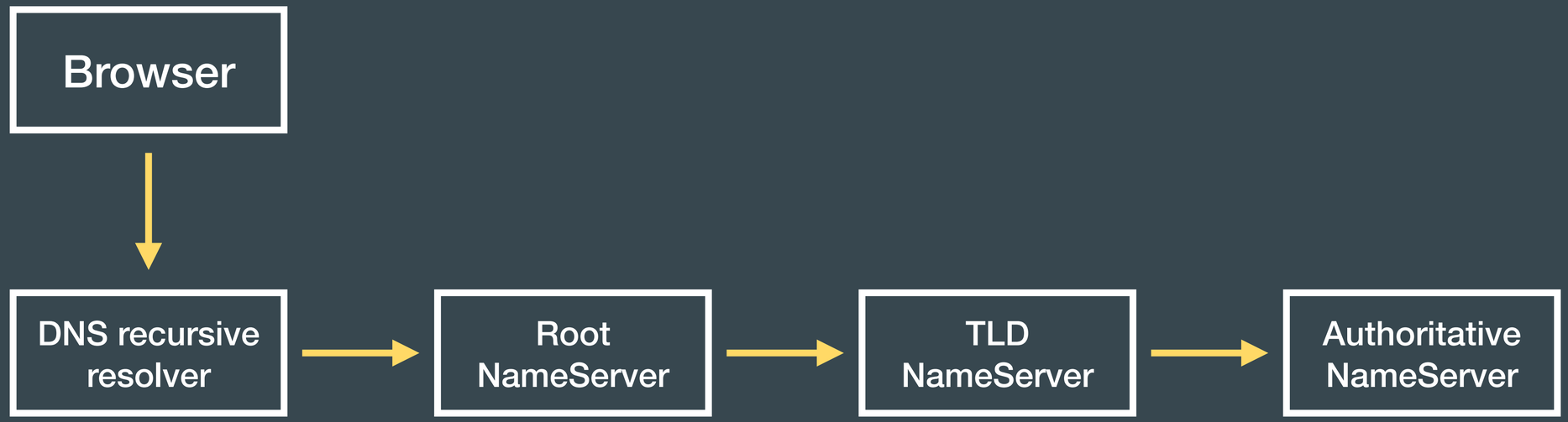
Utilizando el Domain Name System (DNS)

Proceso llamado DNS resolution o lookup

DNS lookup

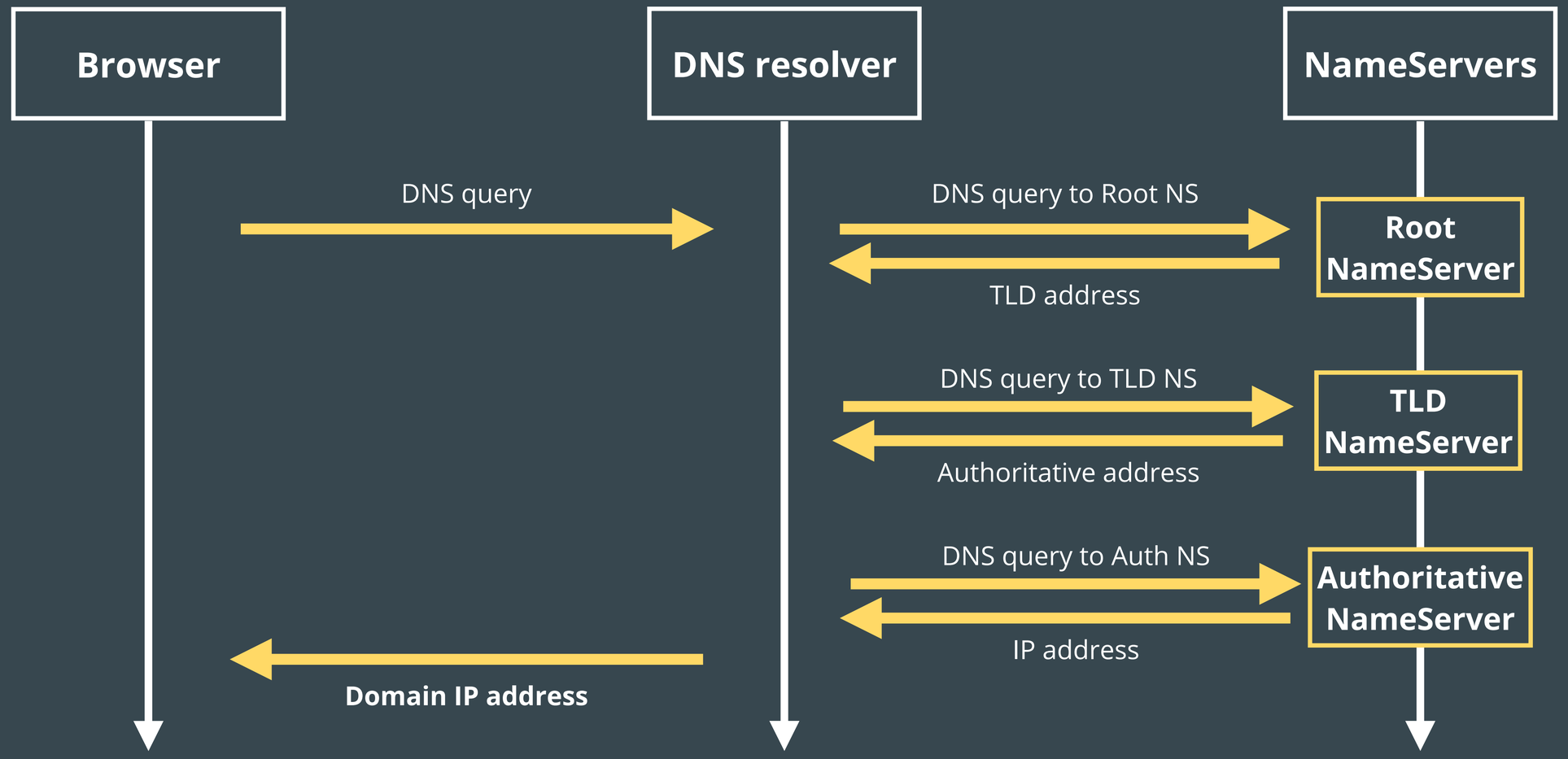
¿Cómo funciona?

Browser realiza una DNS query que desencadena el proceso de lookup



DNS lookup

¿Cómo funciona? - Flujo completo



Dominios en la práctica

¿Dónde comprar?

Extranjero

- GoDaddy
- Namecheap

Chile

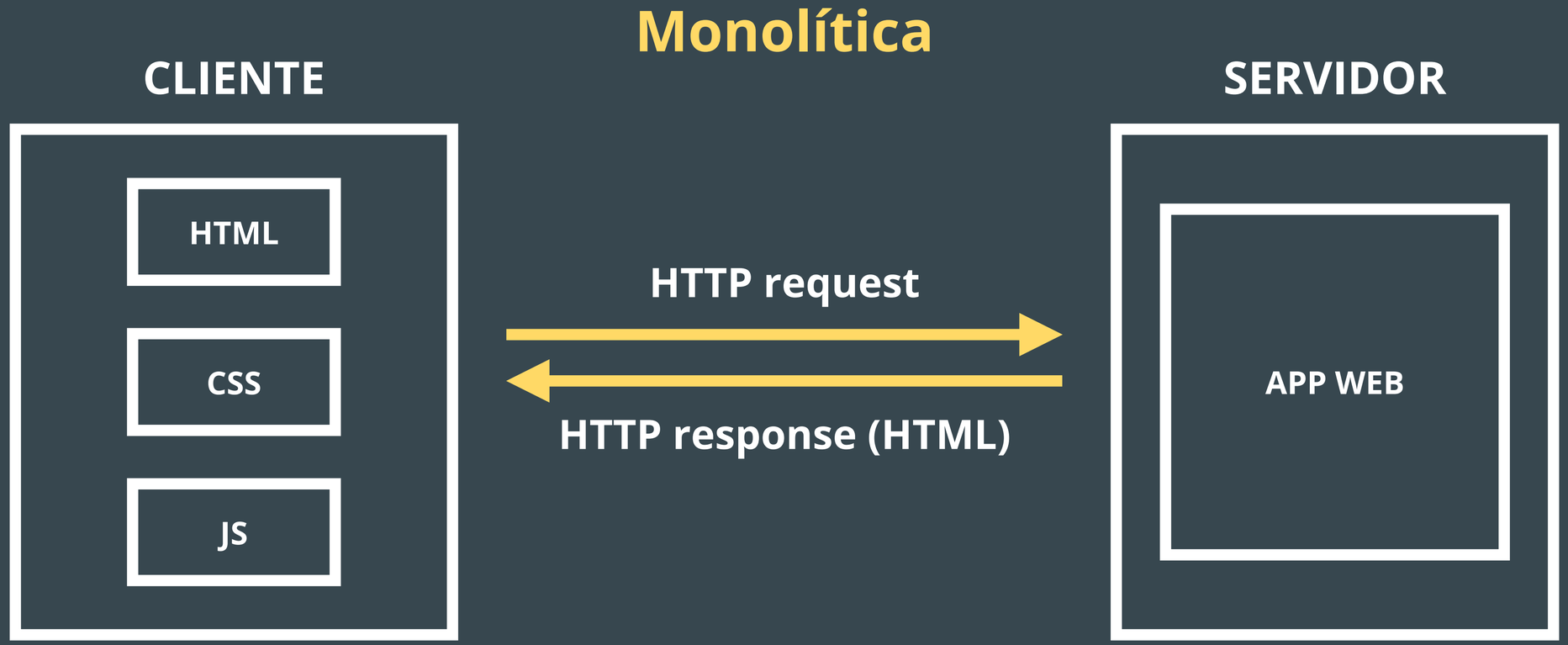
- NIC Chile

Desarrollo web hoy en día

Herramientas y prácticas

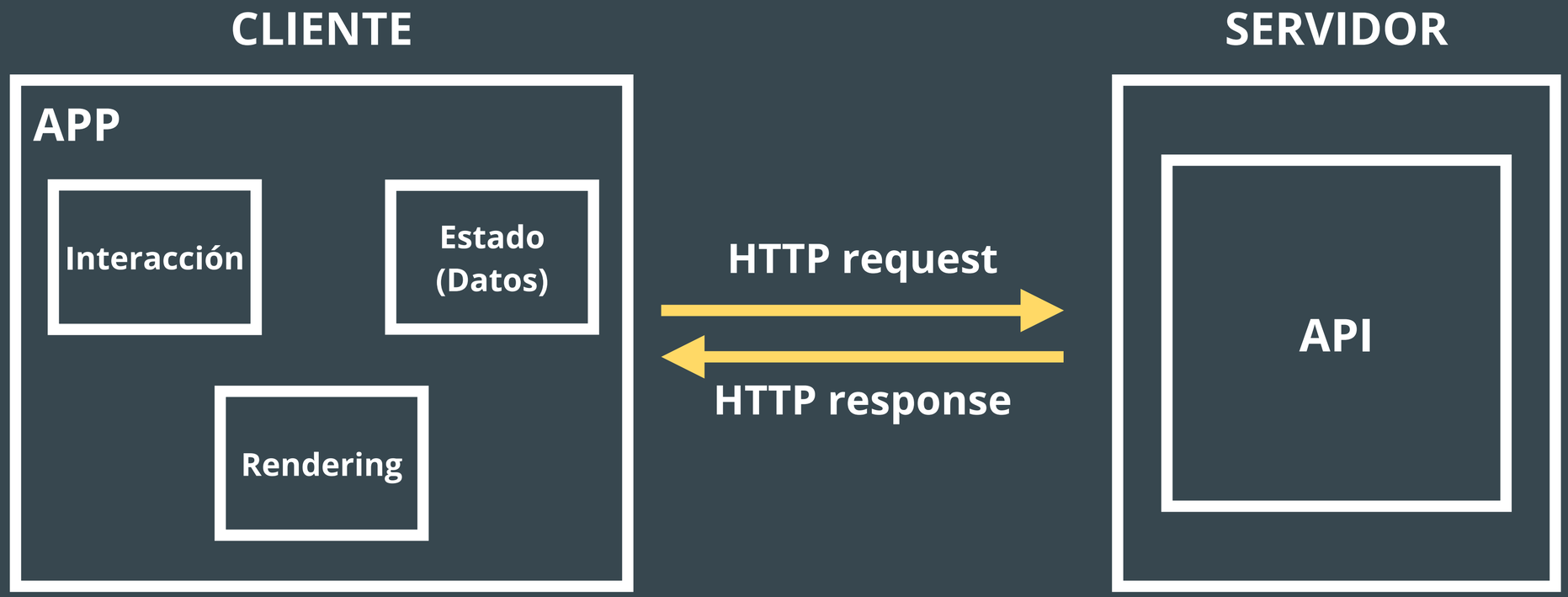
Desarrollo web hoy en día

Contexto: App web tradicional



Desarrollo web hoy en día

Contexto: SPA + API



React en la práctica

Herramientas y prácticas

Create React App

Creando una SPA

Herramienta que nos permite crear un esqueleto (boilerplate) de una single-page application con React instalado

- **Correr una versión en desarrollo**
- **Empaquetar una versión para producción**
- **Output es HTML + CSS + JS**

```
npx create-react-app app-name
cd app-name
npm start # Or yarn start if installed
```

Application state

Contexto

Hasta ahora: componentes tienen su propio estado

¿Y si queremos compartir estado entre diferentes componentes?

- Podríamos definir estado en un componente más arriba en el árbol y pasarlo como prop
- Pasar props de componente en componente puede ser tedioso

Application state

Definición

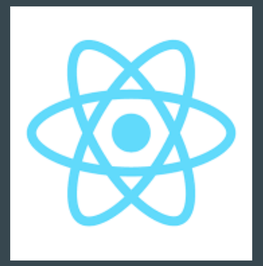
`state` que puede ser utilizado por distintos componentes de la aplicación

Ejemplo: un header con el nombre de usuario

Application state

¿Cómo lograrlo?

Context API



Redux

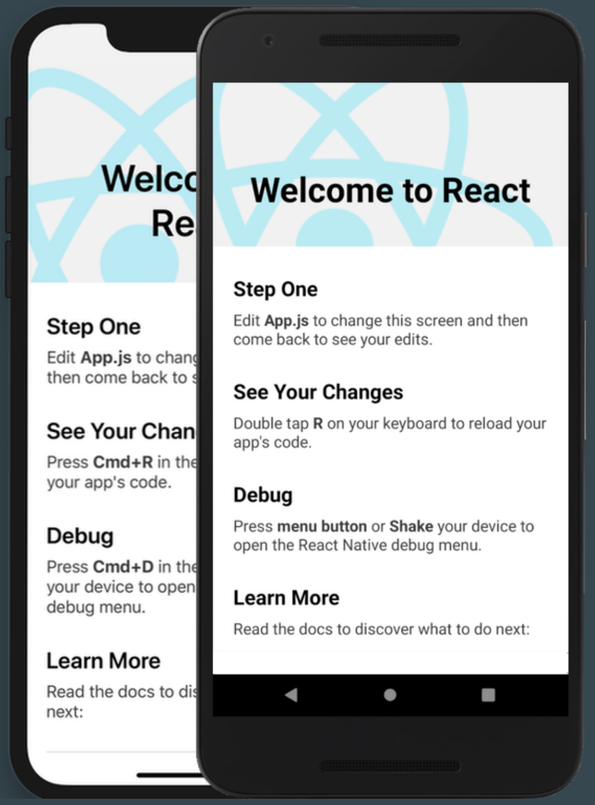
React Native

React para móviles

Librería que permite desarrollar aplicaciones móviles utilizando React

```
import React from 'react';
import {Text, View} from 'react-native';

const WelcomeScreen = () =>
  <View>
    <Text>
      Edit App.js to change this screen and turn it
      into your app.
    </Text>
    <Text>
      Read the docs to discover what to do next:
    </Text>
   </View>
```

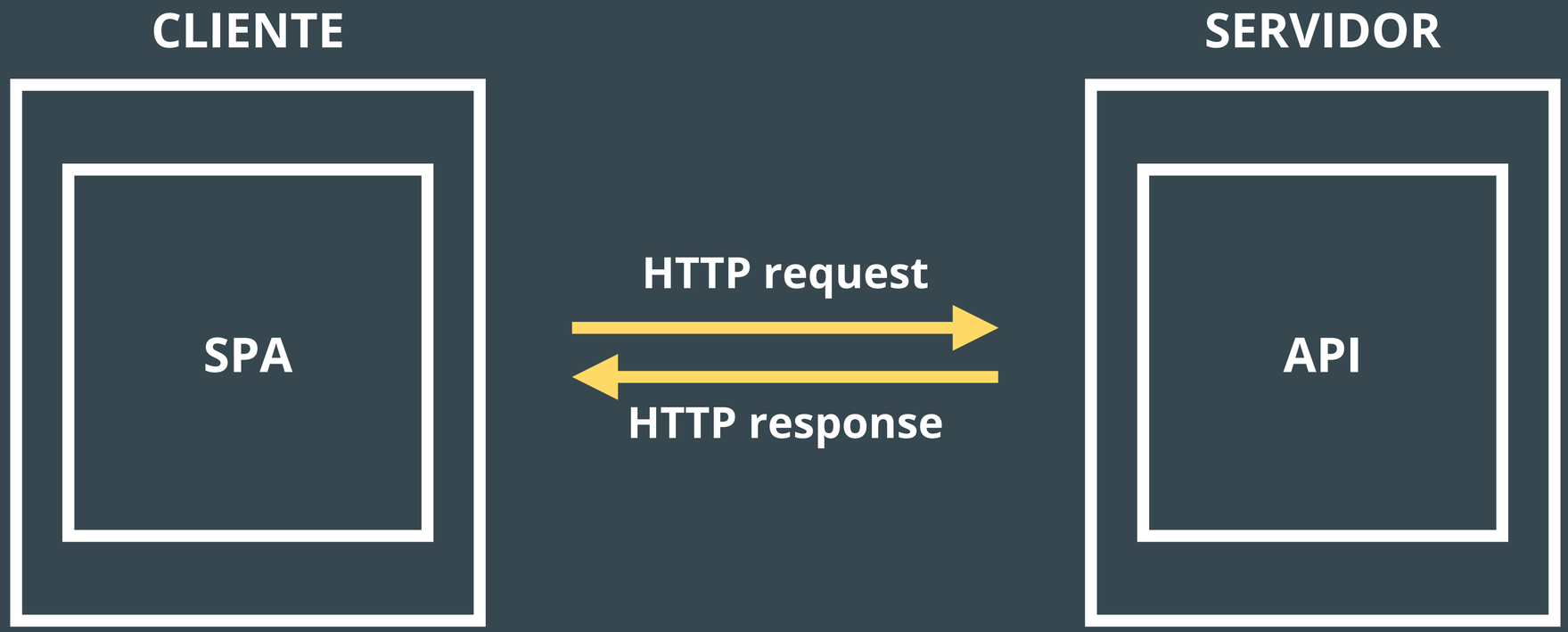


Arquitecturas modernas

Interacción de aplicaciones

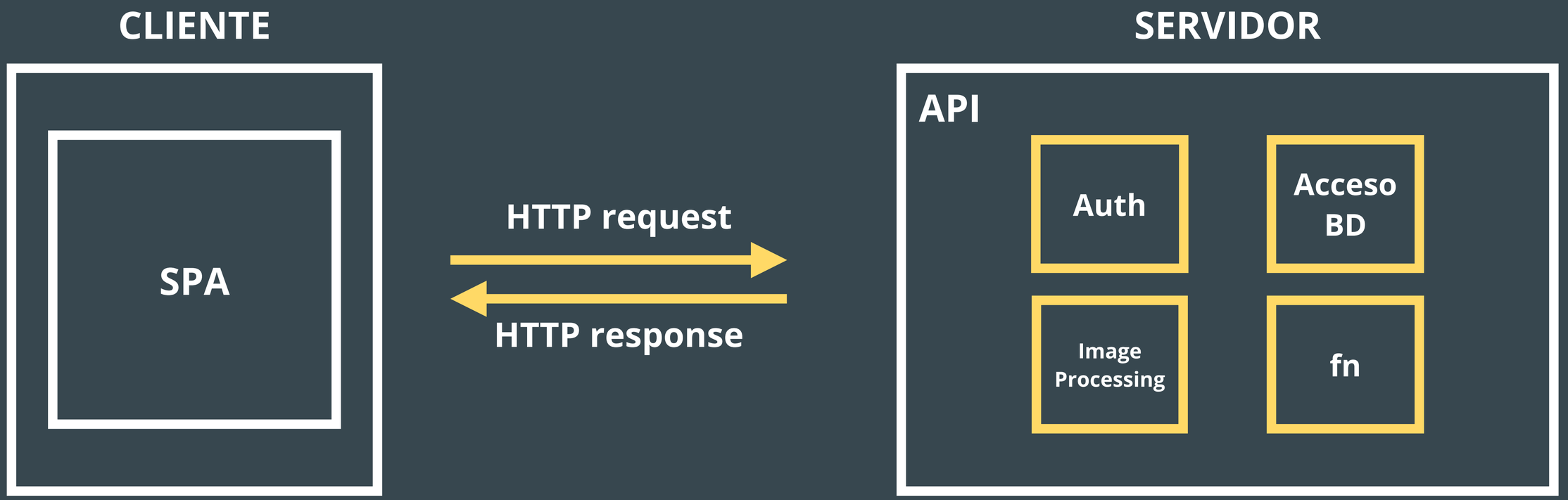
Arquitecturas modernas

Contexto: SPA + API



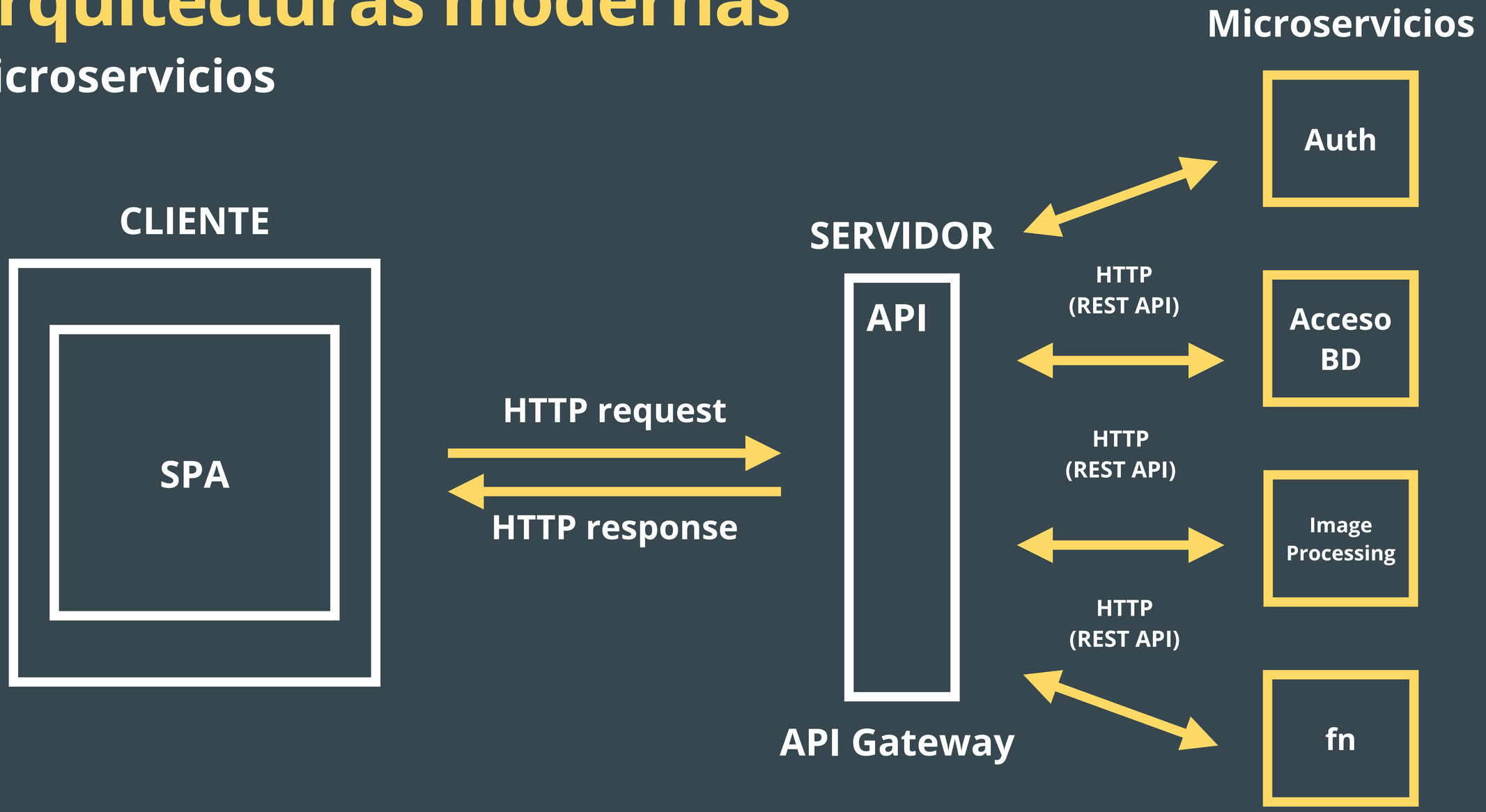
Arquitecturas modernas

Módulos o servicios



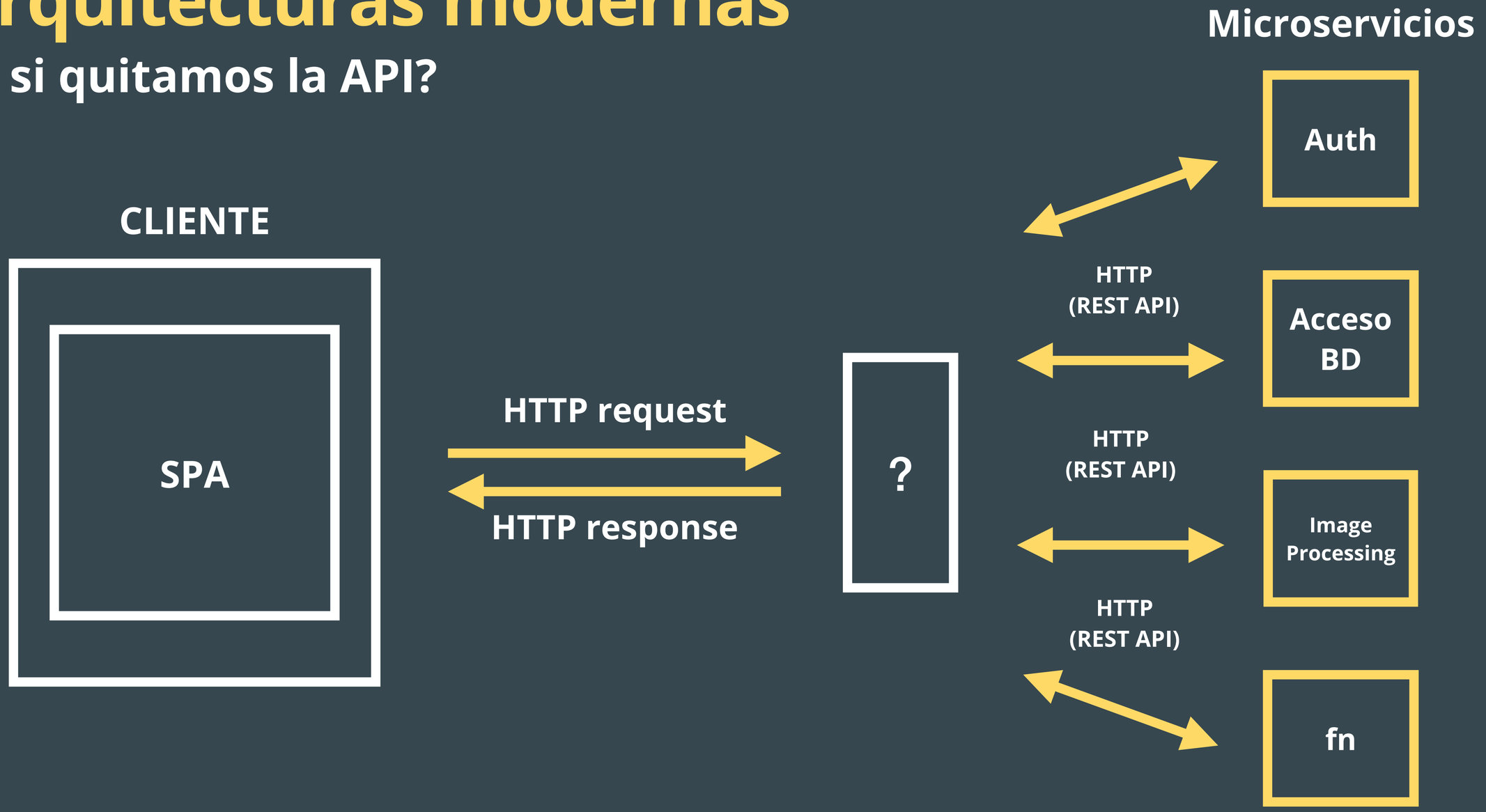
Arquitecturas modernas

Microservicios



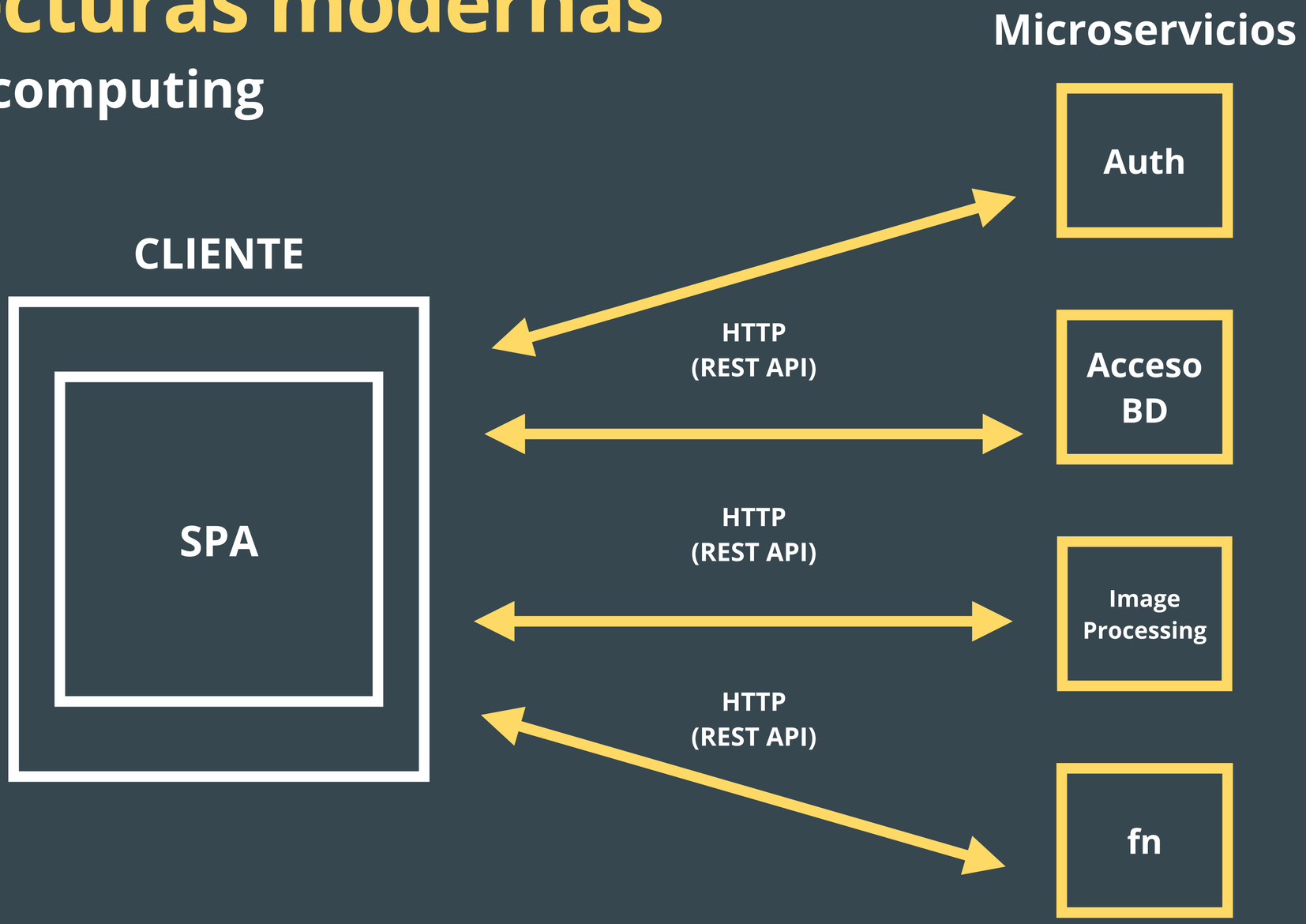
Arquitecturas modernas

¿Y si quitamos la API?



Arquitecturas modernas

Serverless computing



Proveedores serverless

Los más conocidos

Google

- Google Clould Platform
  - Cloud Functions
- Firebase
  - Firestore (DB)
  - Authentication

Amazon

- Amazon Lambda
- Amazon API Gateway
- Amazon DynamoDB
- Amazon Cognito (auth)

Azure

- Azure functions
- App service

Framework: Serverless

Otros temas interesantes

Para profundizar

GraphQL

Jamstack

Service workers

Consejos finales

Para seguir aprendiendo Web

Consejos finales

¿Cómo seguir?

Developer Roadmap

Frontend, Backend, DevOps

Consejos finales

¿A quién seguir?

- Dan Abramov
- Jared Palmer
- Emma Bostian
- Addy Osmani
- Paul Irish
- Cassidy Williams
- TJ Holowaychuk
- Ryan Florence
- Chris Coyier
- Feross Aboukhadijeh
- Majo Ledes

Consejos finales

Reflexión de cierre

La Web es un universo
La Web la hacemos todos

Este es sólo el comienzo...

¡Gracias!

Éxito como desarrolladores

Referencias

- MDN - "Same-origin policy"
- web.dev - "Same-origin policy"
- MDN - "Cross-Origin Resource Sharing (CORS)"
- Cloudflare - "What Is DNS? | How DNS Works"
- microservices.io - "What are microservices?"
- Cloudflare - "What is serverless computing? | Serverless definition"