

# Microservices

*MICROSHOP*

Présentation des nouvelles features / ajouts

Arnaud Marquant  
Anas Shekh Omar  
Kévin Nandacoumar

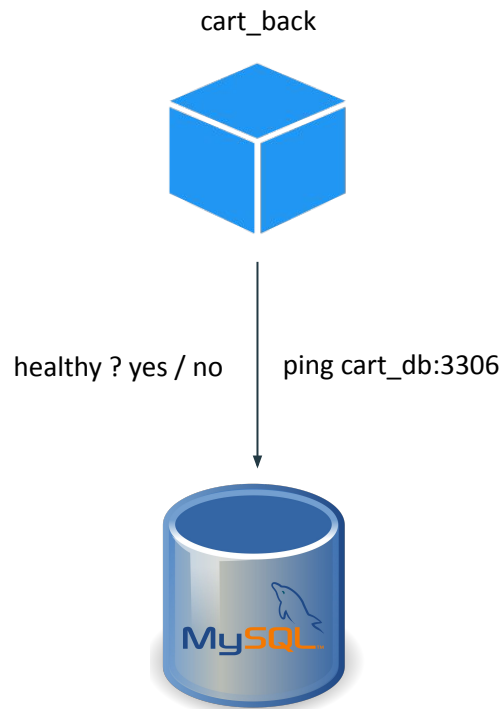
## Objectif initial et résultats

Notre objectif était de produire un site e-commerce réparti en 5 microservices avec une partie front-office pour les clients et une partie back-office pour les administrateurs.

Nous avons bien réussi à implémenter ce qui a été prévu en termes d'architecture. Elle a d'ailleurs évoluée depuis notre passage à l'oral.

Cependant nous avons pas pu trop avancer sur la partie front (visuelle) du site par manque de temps.

## Ajouts / Modifications côté back

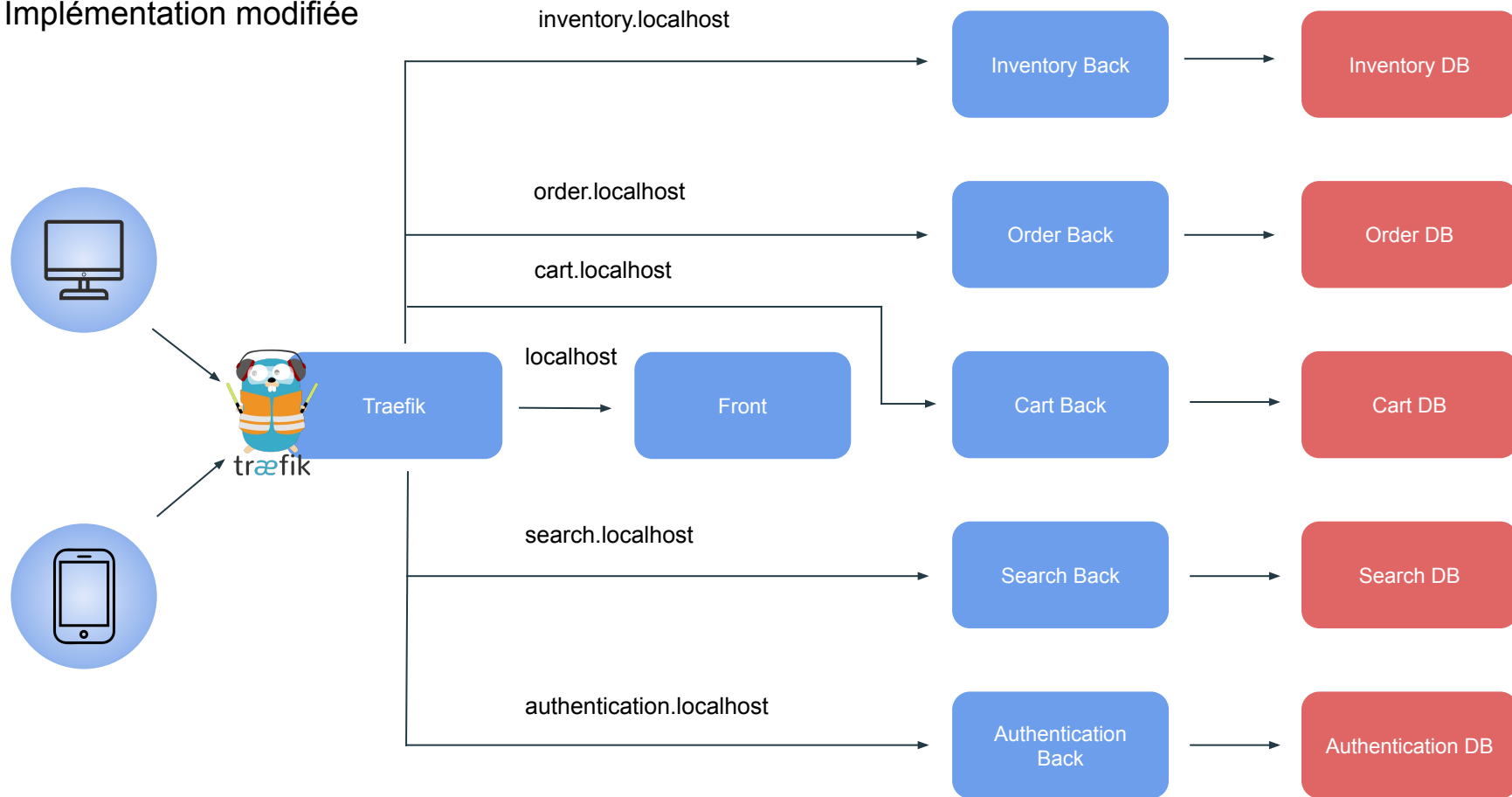


Afin de faciliter la gestion des API liées à nos microservices, nous avons choisi d'utiliser Traefik à la place de Nginx, pour le travail de reverse proxy.

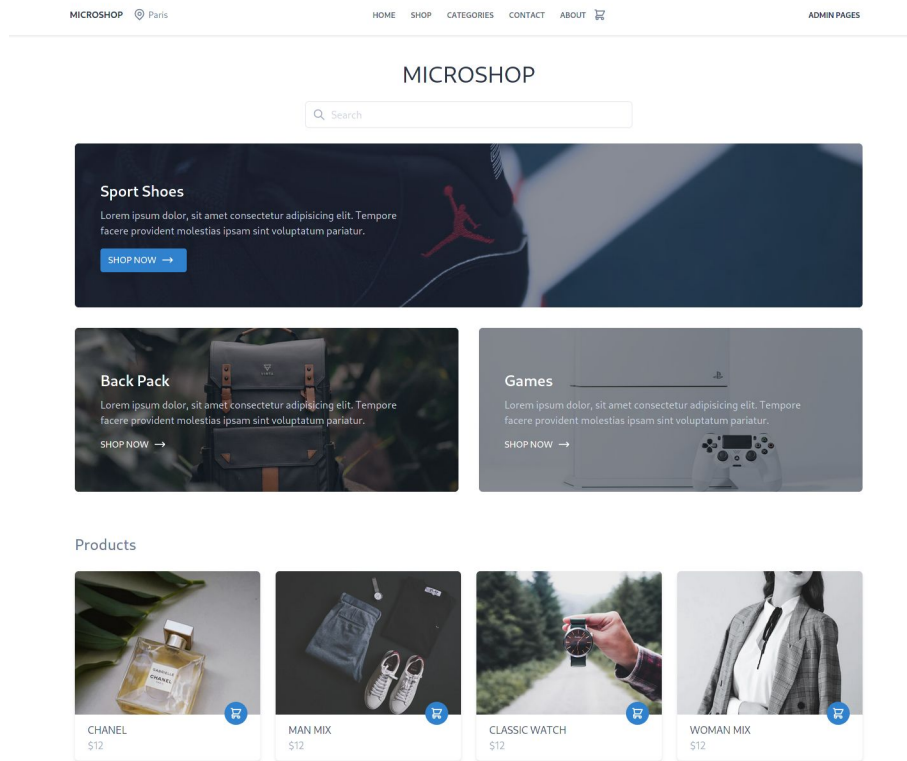
De plus, nous avons ajouté l'écoute du port de la base de donnée pour chaque container au niveau du back, afin de lancer nos back tournant sous Flask, que lorsque nos bases de données sont initialisées.

Cela permet donc de lancer notre architecture complète en n'entrant seulement que : "docker-compose up -d --build"

## Implémentation modifiée



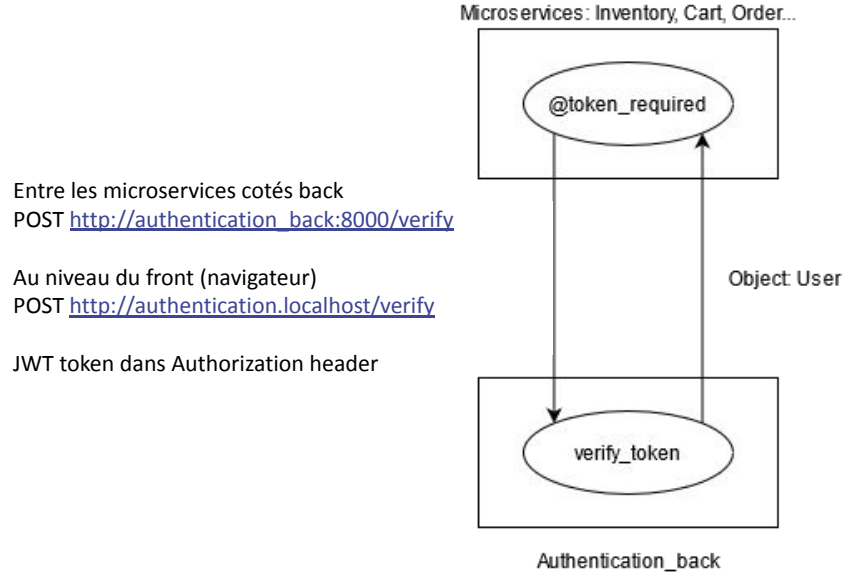
# Ajouts front



Nous avons travaillé sur la partie front office que les clients verraient en arrivant sur le site. Cela a été réalisé avec Angular.

Une approche par composant a été privilégiée pour tenter d'isoler chaque éléments du front (cart, produits...) en fonction du microservice correspondant au niveau du back.

# Schéma authentication JWT



Nous avons réussi à ajouter l'usage de token JWT, qui oblige le client de posséder un token d'accès valide pour atteindre certaines routes. Il y a également un "refresh\_token" qui est utilisé pour renouveler le token d'accès, lorsqu'il expire.

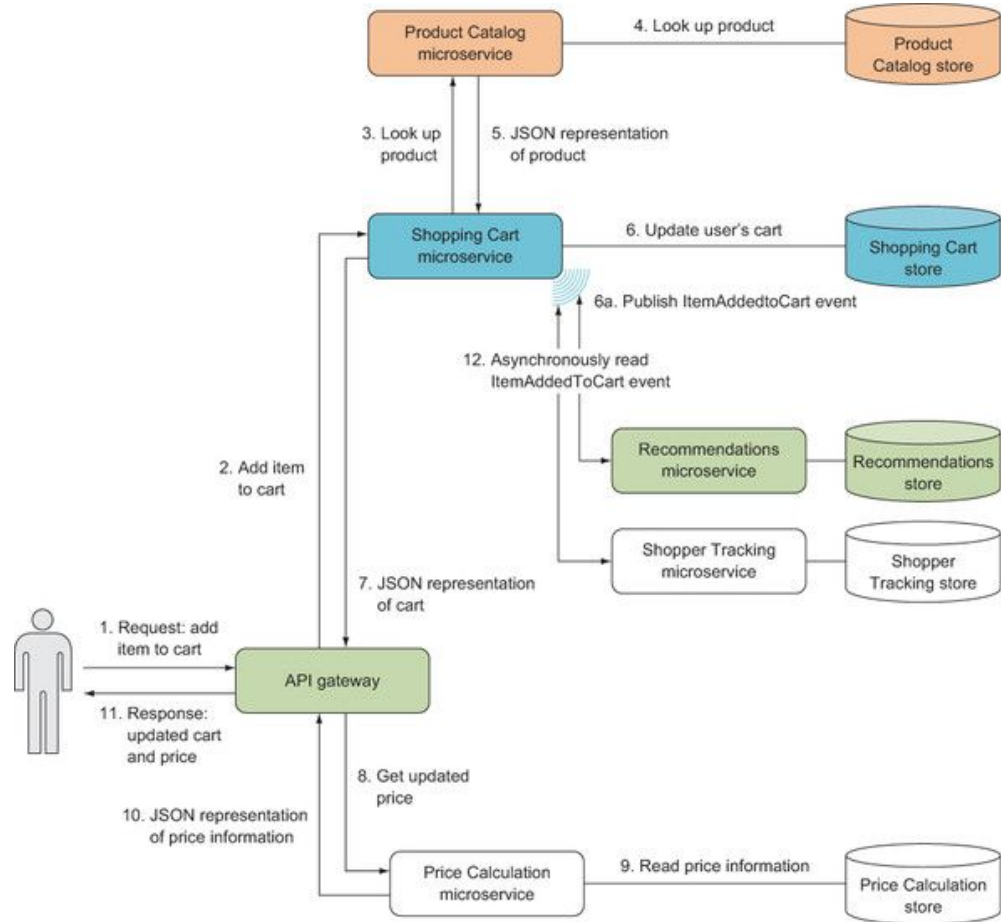
Support de présentation de  
l'oral du 09/02/2021

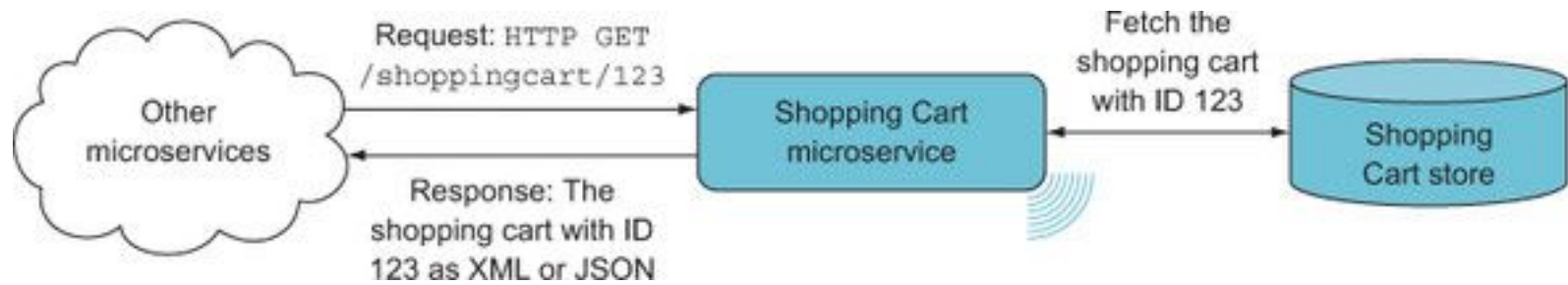
# Microservices

*MICROSHOP*

Arnaud Marquant  
Anas Shekh Omar  
Kévin Nandacoumar

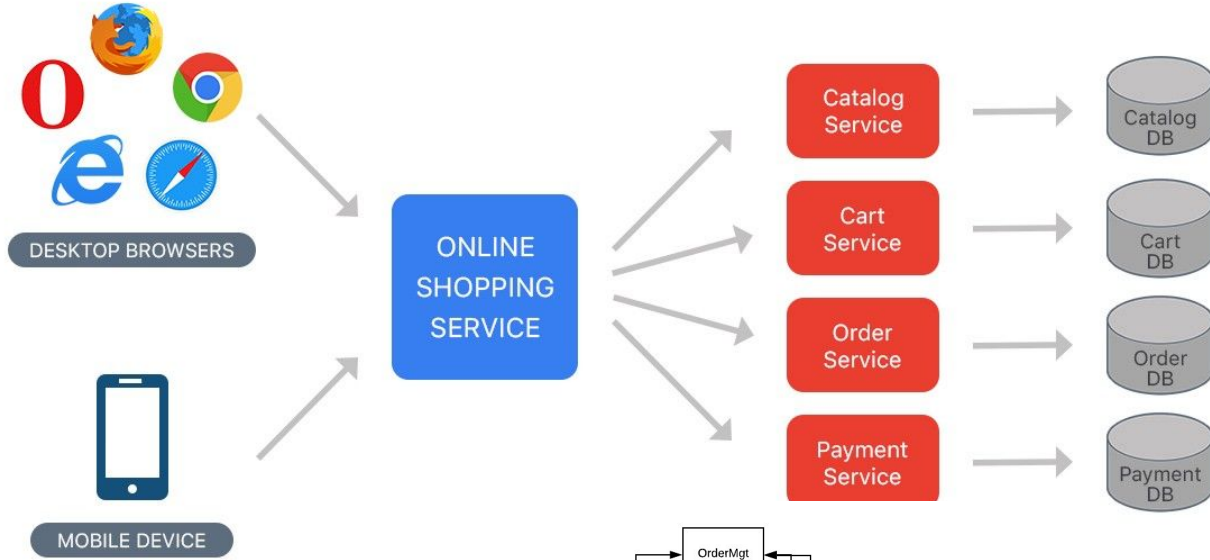




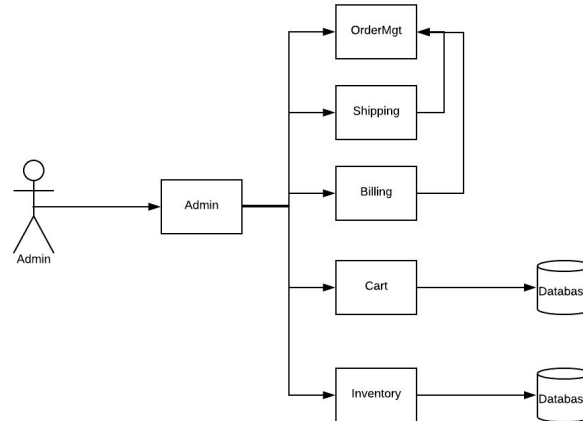


# Principe

Front-office



Back-office



# UX Design

655 Results

Sort By: Most Popular



## Lamadrid Cabernet Sauvignon Reserva 2011

Cabernet Sauvignon from Argentina, South America



WW 95

★★★★★(1)

~~\$20.00~~ **\$14.99**

Save \$5.01 (25%)

1

Add to Cart

Ships Thu, Oct 8



## Dona Paula Black Edition Red Blend 2012

Bordeaux Red Blends from Argentina, South America



JS 94

★★★★☆(27)

~~\$17.00~~ **\$14.99**

Save \$2.01 (12%)

1

Add to Cart

Ships Mon, Oct 5



## Meiomi Pinot Noir 2013

Pinot Noir from California



WS 92

★★★★☆(24)

~~\$22.00~~ **\$19.99**

Save \$2.01 (9%)

1

Add to Cart

Ships Mon, Oct 5



## Substance Cabernet Sauvignon 2013

Cabernet Sauvignon from Columbia Valley, Washington



RP 92

★★★★☆(2)

**\$16.99**

1

Add to Cart

Ships Mon, Oct 5

# UX Design

ORDER PLACED  
June 18, 2015

TOTAL

SHIP TO  
Peter Dushenski ▾

ORDER # [REDACTED]  
[Order Details](#) | [Invoice](#)

## Delivered

Your package was delivered. Please look in your mailbox or wherever you typically get mail.

Delivered on: Friday, July 10, 2015



2 of Bathroom Tissue, 2 ply, 12 pack, 300 sheets/roll (Pack of 4)

Sold by: Amazon.com.ca, Inc.

CDN\$ 39.99

Buy it Again

Track package

Return or replace items

Leave package/delivery feedback

Write a product review

## Preparing for Shipment

Delivery estimate: **Wednesday, July 15, 2015 - Tuesday, July 21, 2015 by 8:00pm**



Seventh Generation, 100% Recycled Paper Towels - Unbleached, 140 sheets/roll (Pack of 4)

Sold by: Amazon.com.ca, Inc.

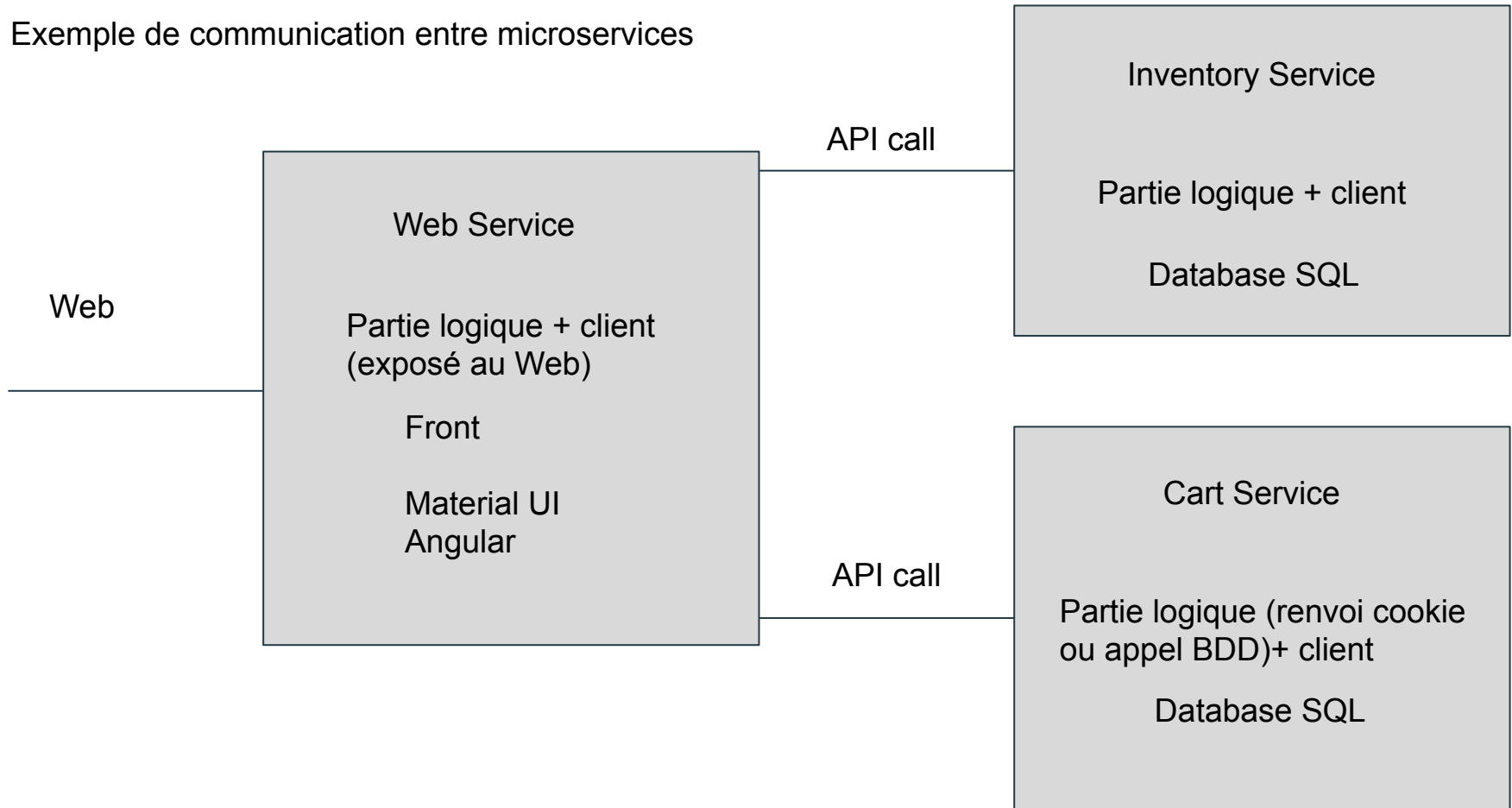
CDN\$ 55.96

Buy it Again

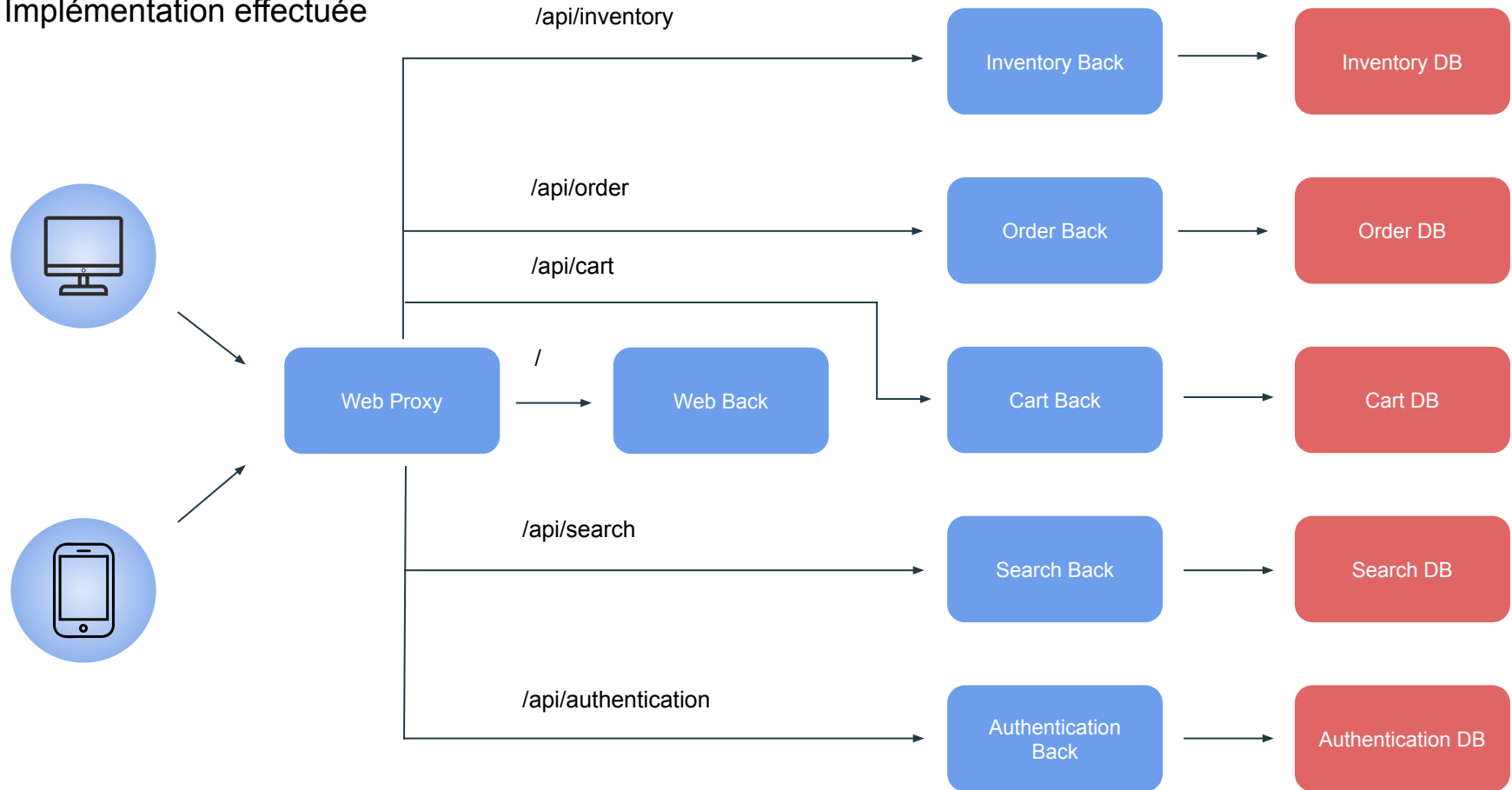
Track package

Request Cancellation

## Exemple de communication entre microservices



## Implémentation effectuée



endpoint : /

## Web Service

### Frontend

#### ProductService

- Product[] : getProducts()

#### OrderService

- Order[] : getOrders()

#### CartService

- Cart[] : getCarts()

#### AuthenticationService

- User[] : getUsers()

### Web proxy

localhost	→	web_back:5000
localhost/api/inventory	→	inventory:8000/api
localhost/api/authentication	→	authentication:8000/api
localhost/api/order	→	order:8000/api
localhost/api/search	→	search:8000/api



endpoint : /api/authentication

## Authentication Service

### Authentication backend

/login : userLogin (POST)  
/logout : userLogout (POST)  
/users : allUsers (GET)  
/registration : UserRegistration (POST)

### Authentication DB

#### Authentication

- id
- username
- email
- address
- password
- isAdmin

endpoint : /api/inventory

## Inventory Service

### Inventory backend

Product : getProduct(id)

list : getProducts([ids])

Stock : getProductStock(idProduct)

list : getProductsStock([ids])

addProduct(name, price, [images], qty)

updateProduct(id, price, [images], qty)

deleteProduct(id)

float : getPrice(idProduct)

float : getAmount([idProducts])

### Inventory DB

Product

- id
- name
- price
- images

Stock

- id
- idProduct (clé étrangère)
- quantity

endpoint : /api/order

## Order Service

### Order backend

Order : getOrder(id)

list : getOrders([ids])

addOrder(idUser, products, amount)

updateOrder(id, products, amount)

deleteOrder(id)

float : getAmount()

### Order DB

Order

- id

- idUser

- products

- amount

endpoint : /api/cart

## Cart Service

### Cart backend

Cart: getCart(id)  
Cart: getCartByUser(userId)  
list : getCarts()

createCart(userId, products)  
updateCart(id, products)  
deleteCart(id)

### Cart DB

Cart  
- id  
- userId  
- products