DESENVOLUPAMENT I IMPLEMENTACIÓ D'UNA BOTIGA ONLINE

Projecte Final de Grau

Jordi Alvaro Arqués

12 de juny de 2019

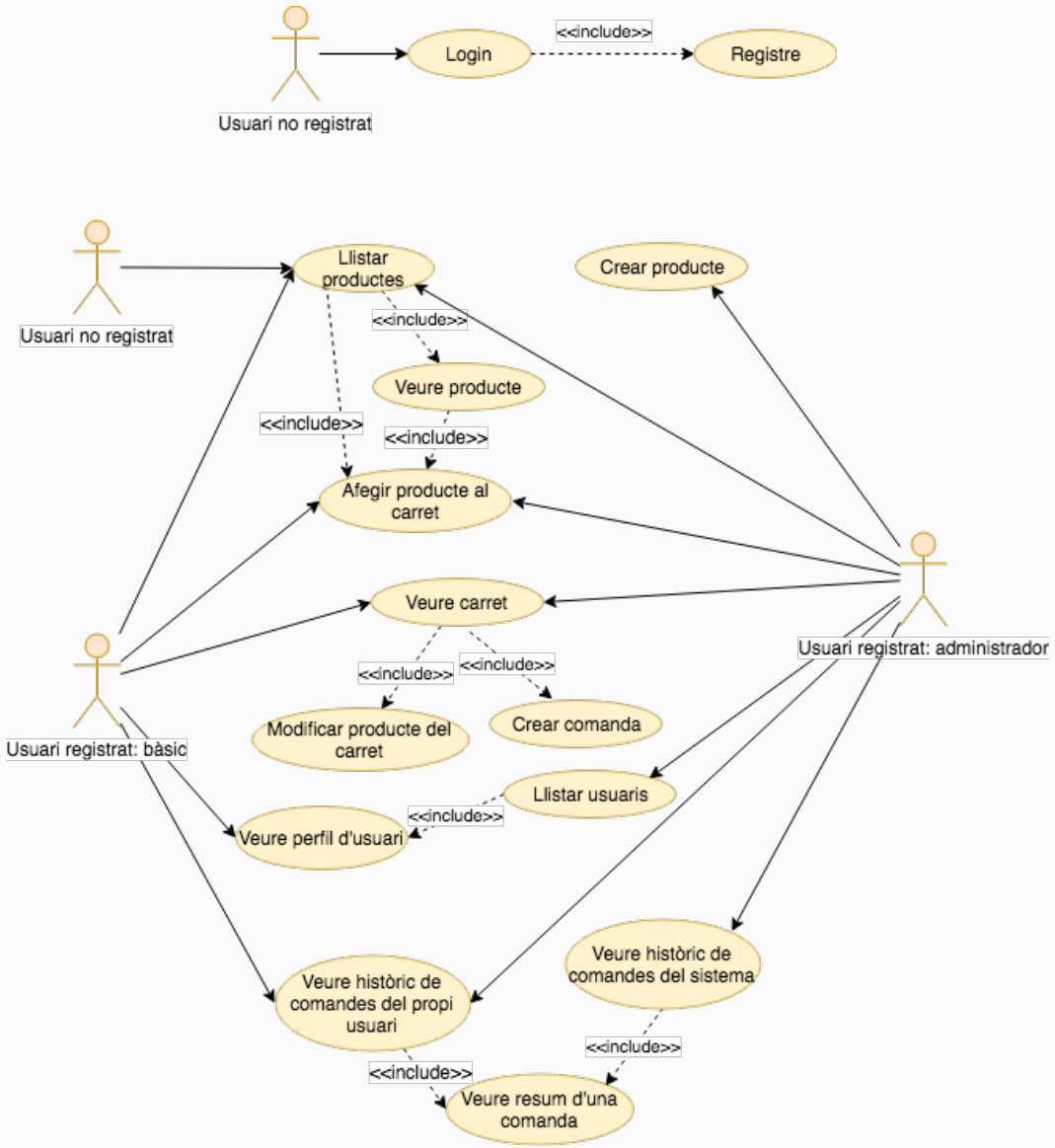
Universitat Oberta de Catalunya

Índex

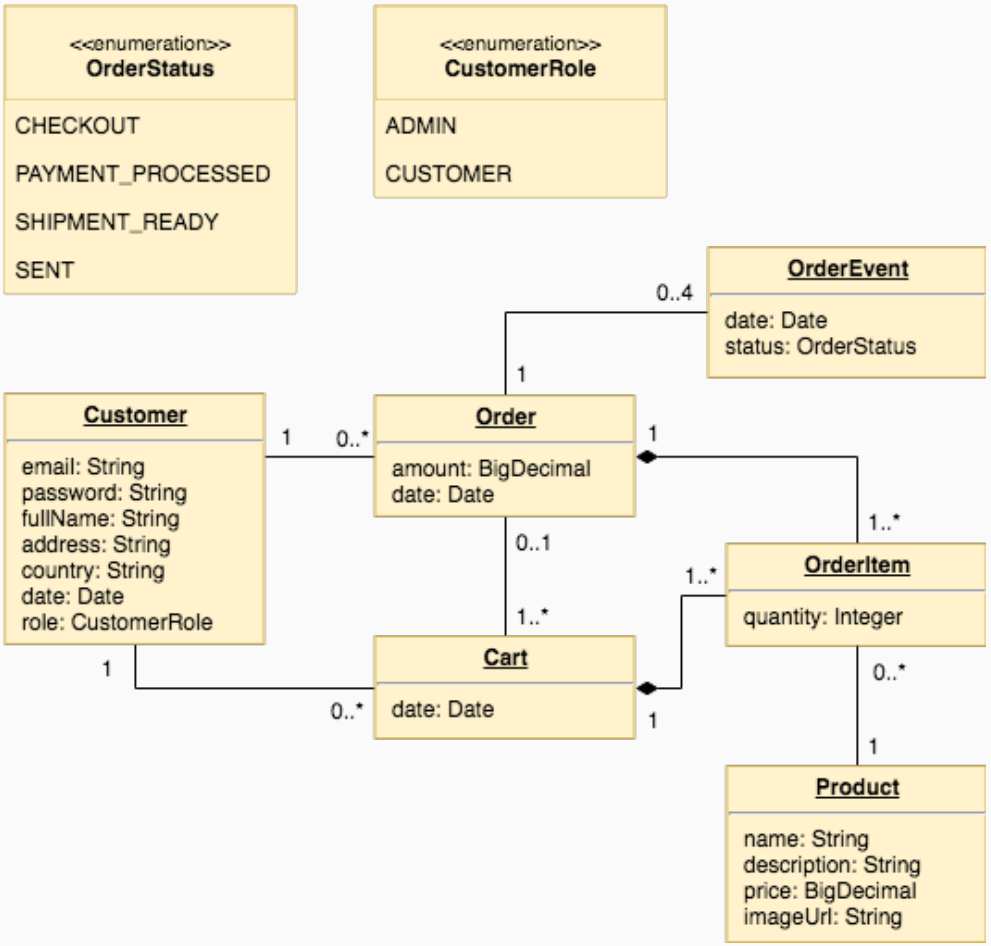
Objectius

- Dissenyar una arquitectura en microserveis.
- Incloure serveis addicionals:
  - API Gateway
  - Service Discovery
- Aprofundir en les llibreries i *frameworks*:
  - Spring
  - ReactJS
- Configurar un procés d'integració contínua
- Investigar:
  - Event sourcing
  - NoSQL
  - Llibreries reactives (Reactor)

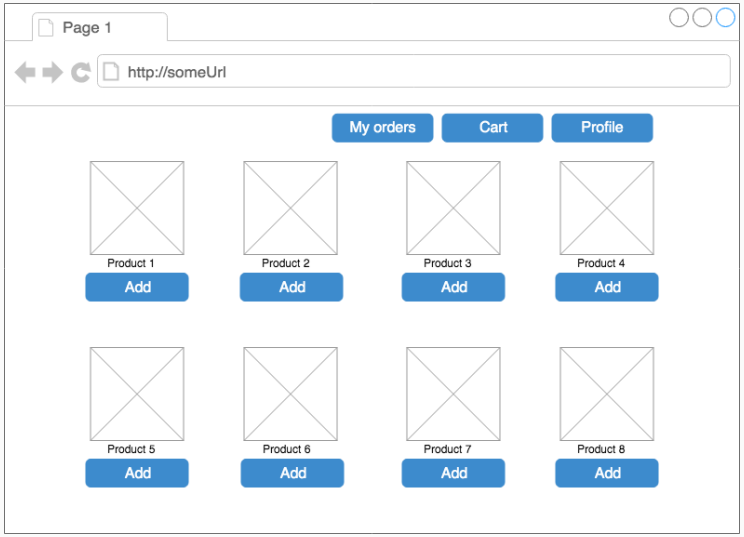
Disseny i anàlisi previ: Casos d'ús

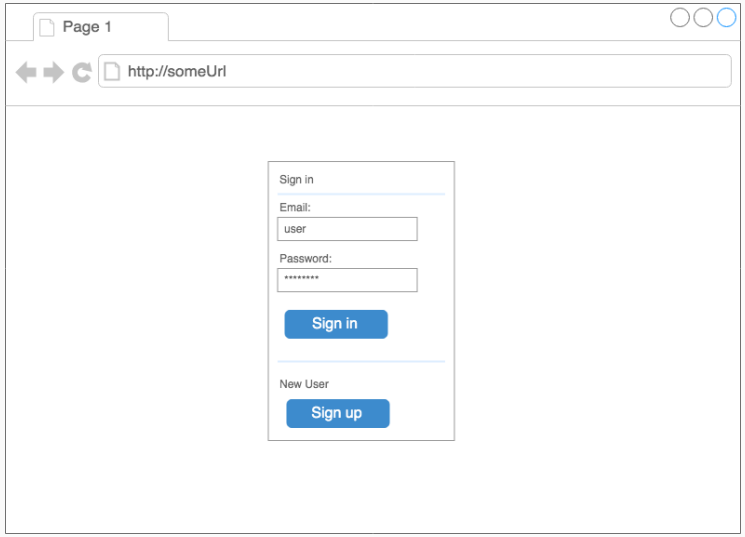


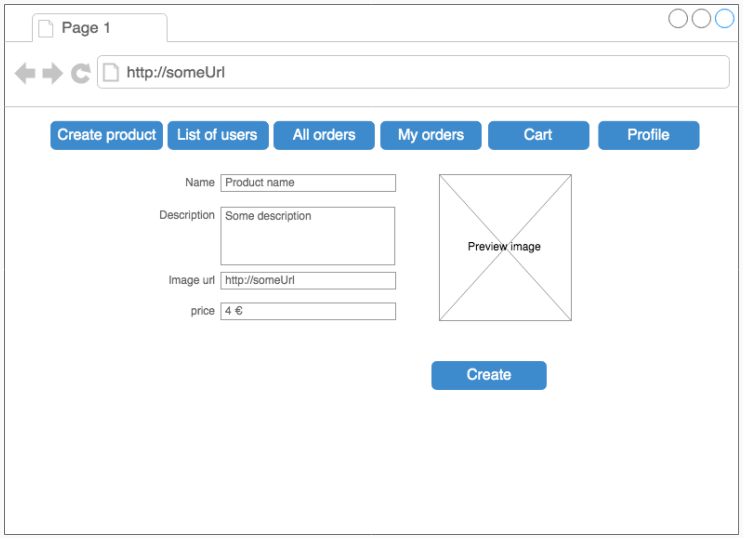
Disseny i anàlisi previ: Diagrama de classes

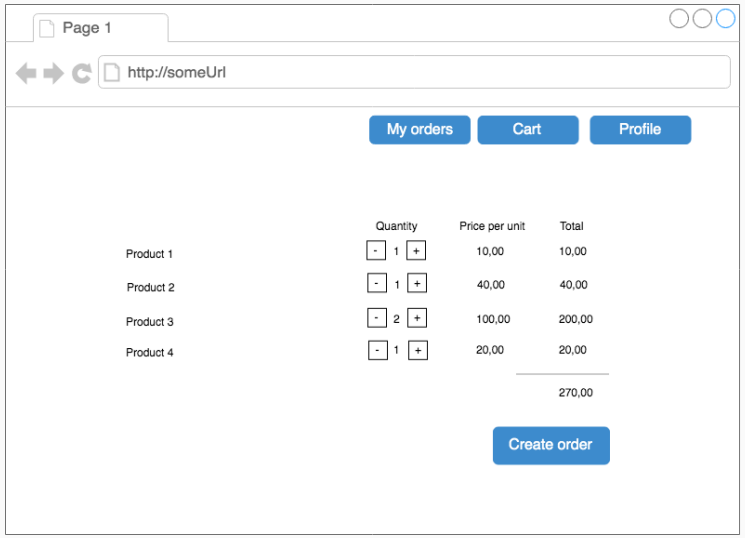


Disseny i anàlisi previ: Model de pantalles









Tecnologies utilitzades: Backend

- *Framework* complet i modern per a aplicacions Java: **Spring**
- Llibreries incloses a Spring:
  - Informació operacional del sistema: **Spring Boot Actuator**
  - Integració amb MongoDB: **Spring Data MongoDB**
  - Eines de gestió de microserveis: **Spring Cloud Netflix: Eureka, Zuul, Hystrix, Ribbon**
  - Interacció amb *REST endpoints*: **Feign Client**
- Altres llibreries:
  - Llibreria per aplicacions reactives: **Project Reactor**
  - Reducció de codi: **Lombok**
  - Mapatge entre classes: **MapStruct**

Tecnologies utilitzades

Frontend:

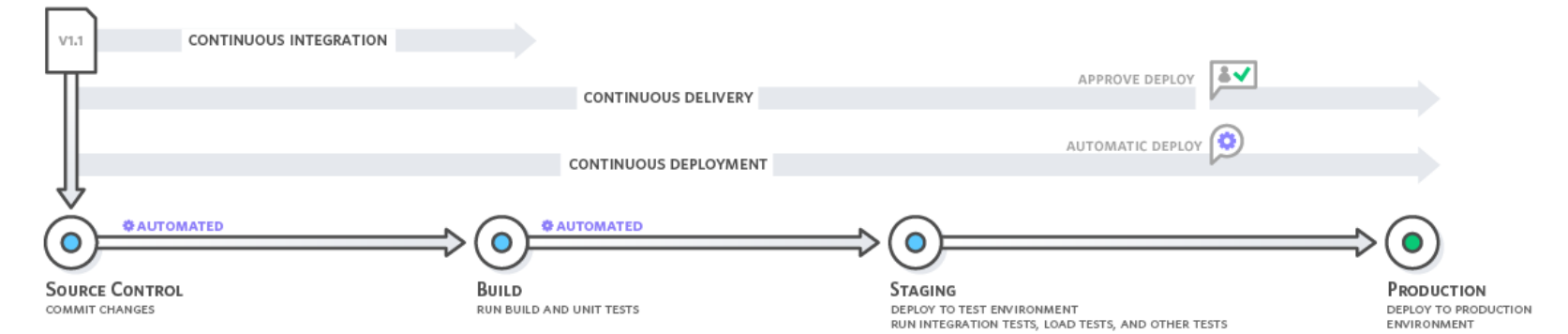
- Llibreria CSS: **Bootstrap 4**
- Llibreria JS: **ReactJS**
- Llibreria de gestió d'estat global a JS: **Redux**

Base de dades:

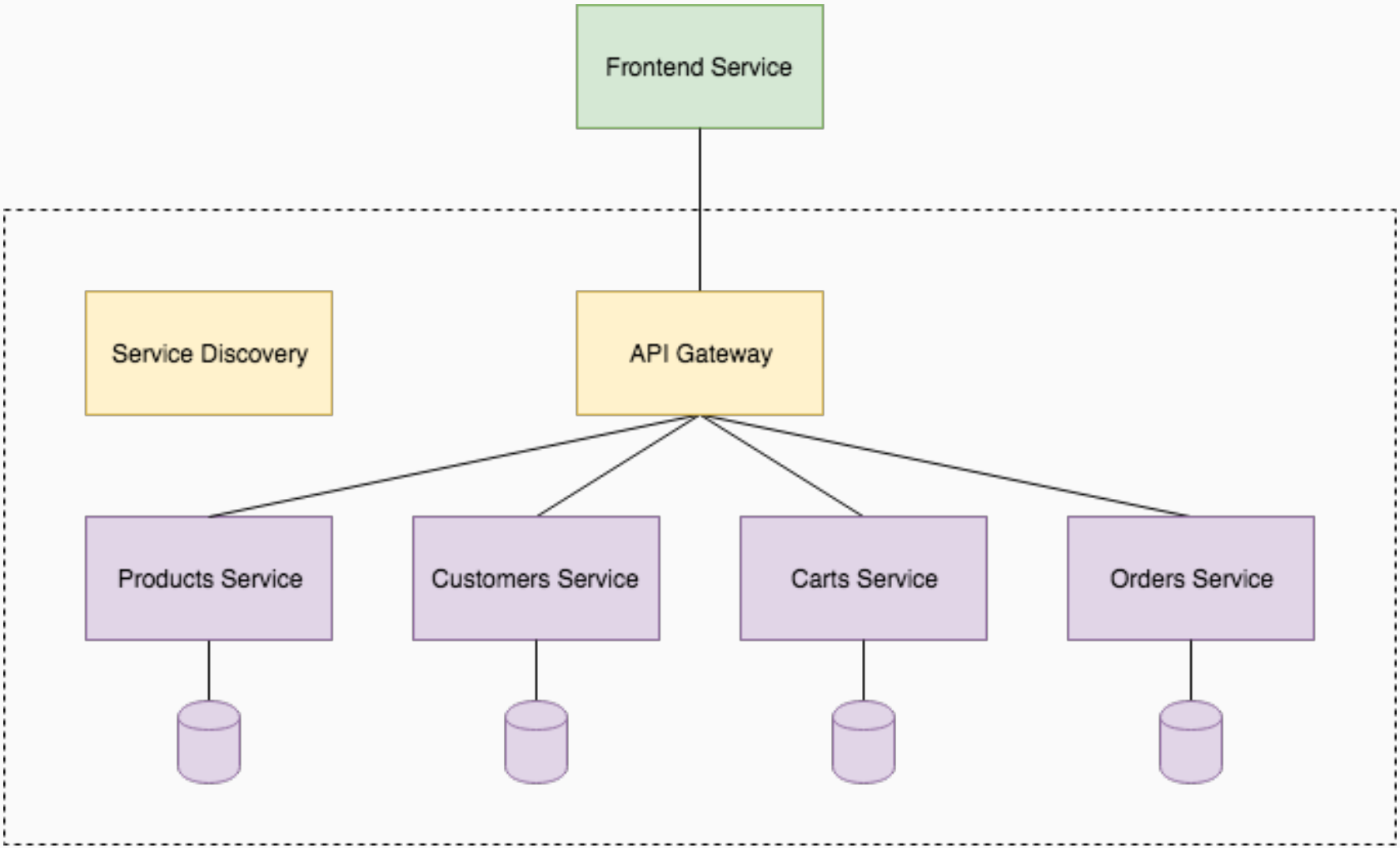
- No relacional (NoSQL): **MongoDB**

Tecnologies utilitzades: Integració contínua

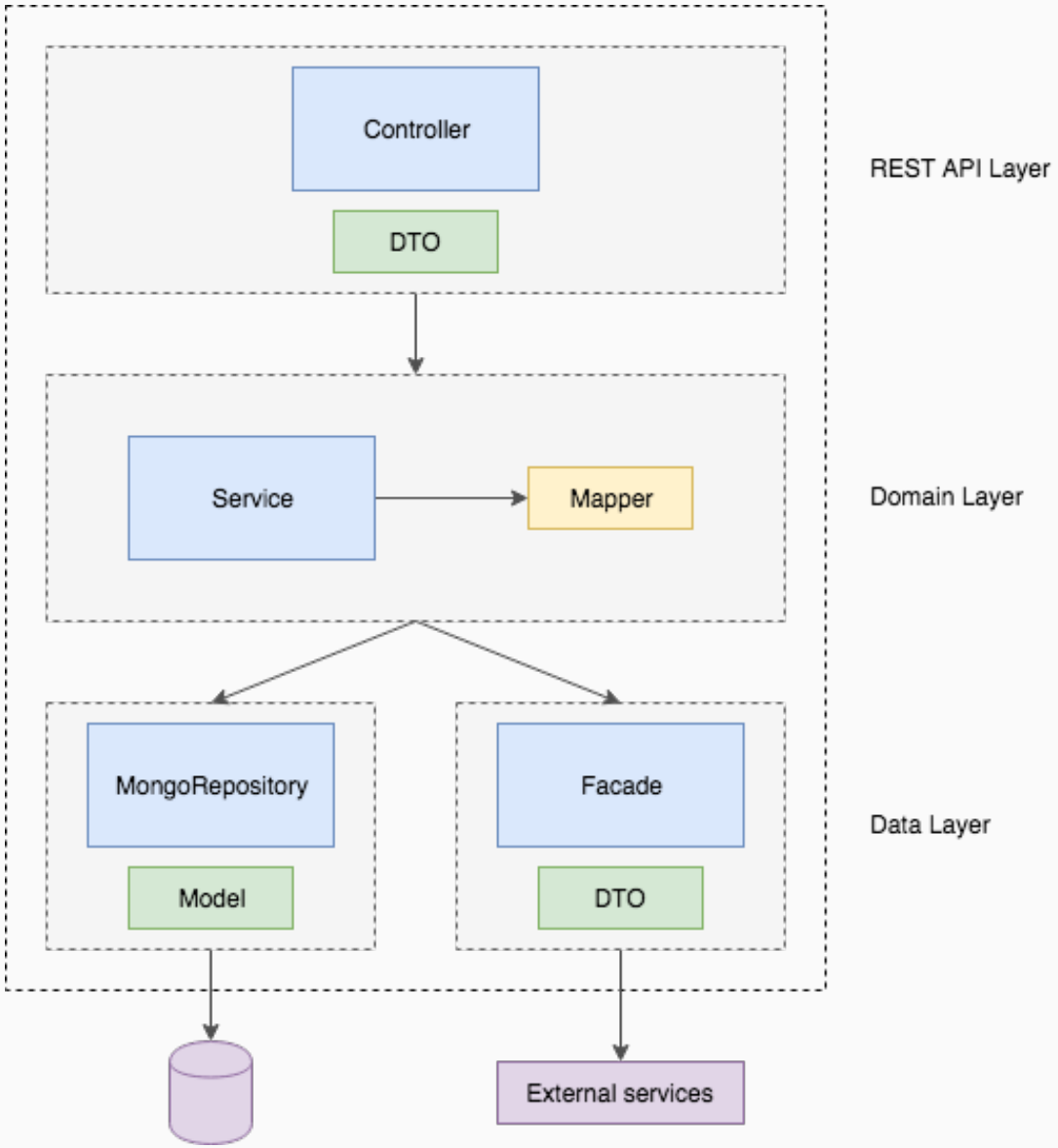
- Control de versions: **Git** (GitHub com a proveïdor)
- Gestor de contenidors d'aplicacions: **Docker**
- Eina d'integració contínua: **TravisCI**
- *Platform as a Service* (*PaaS*): **Heroku**
- Servei d'allotjament de webs estàtiques: **GitHub Pages**



Arquitectura del sistema

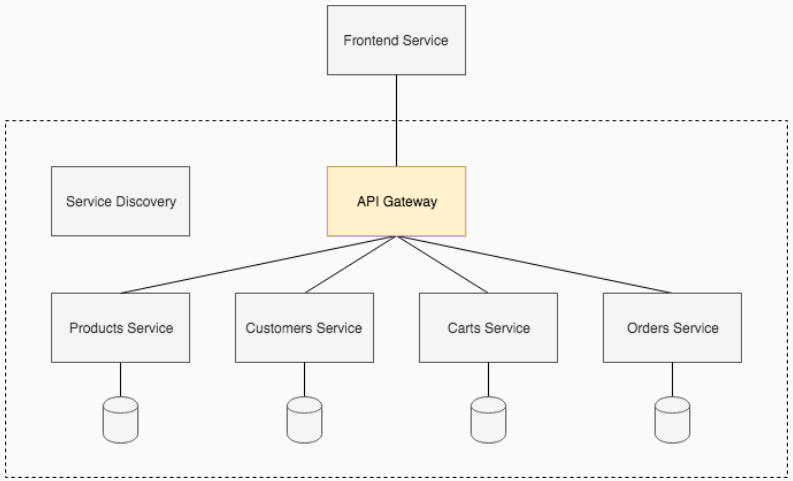


Arquitectura del sistema: Parts comunes al *backend*



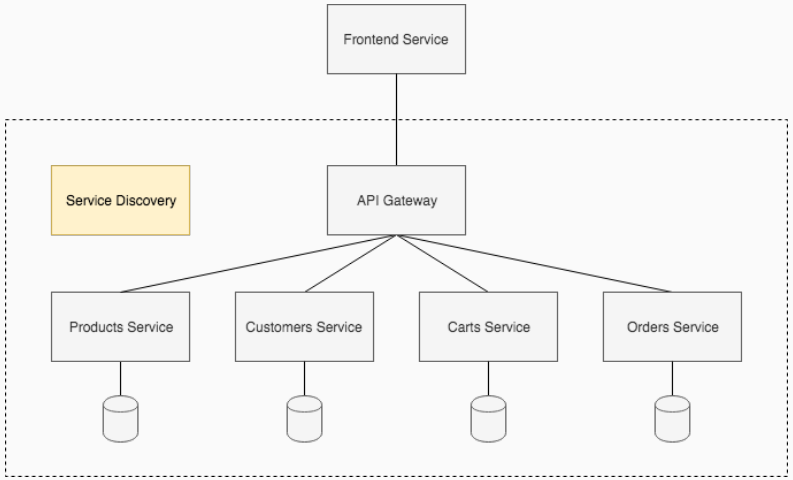
Arquitectura del sistema: *API Gateway Service*

- Punt d'entrada al sistema des de l'internet públic.
- Enrutament i composició de crides *REST*.
- Protecció i filtratge al backend.
- Implementació feta amb Zuul de Spring Cloud Netflix.
- Accepta *CORS* des del frontend.

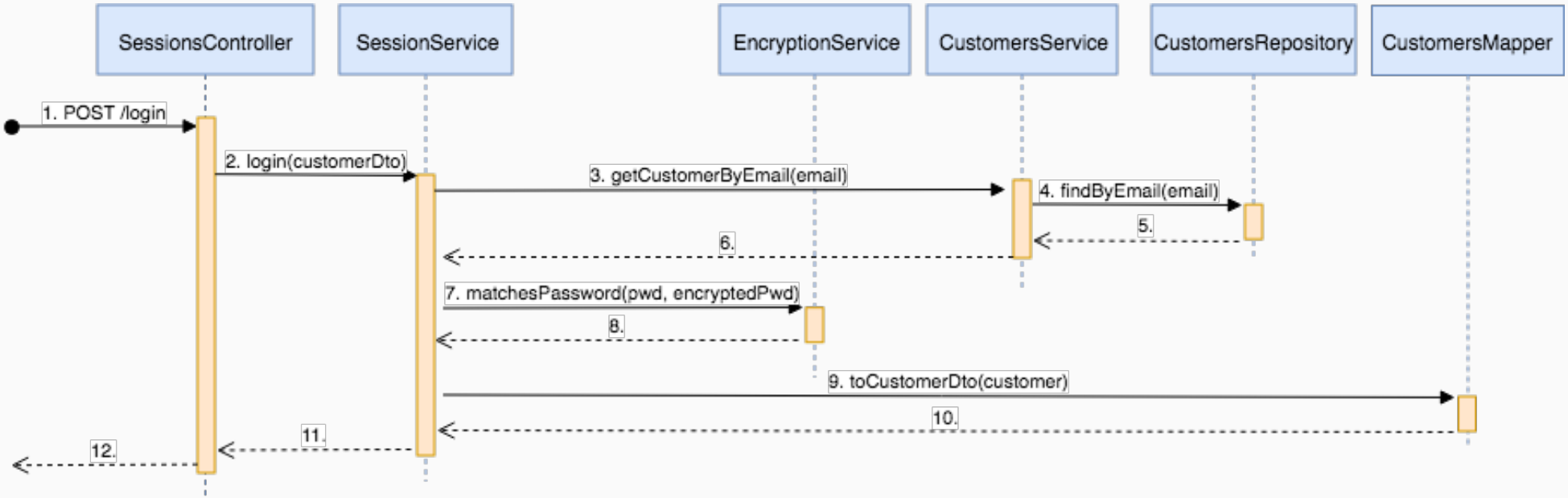


Arquitectura del sistema: *Service Discovery*

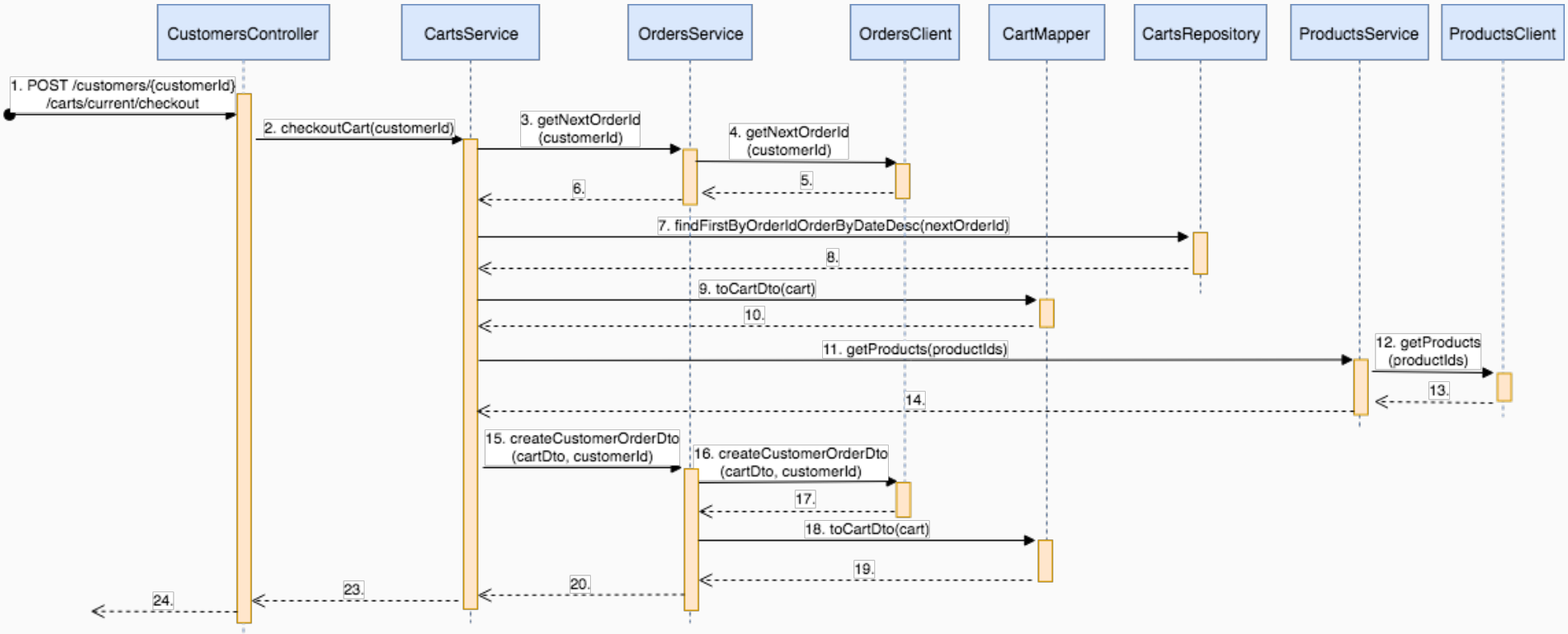
- Servei de descobriment i registre d'aplicacions.
- Funcionament:
  - A mesura que s'inicien les aplicacions s'hi registren.
  - En un moment, l'aplicació **A** vol contactar amb **B**:
    - **A** contacta amb el *Service Discovery* i pregunta per **B**
    - El *Service Discovery* li respon amb la *URL* correcta de **B**
    - **A** contacta directament a **B**



Arquitectura del sistema: *Customers Service*



Arquitectura del sistema: *Carts Service*



Treball futur

- Completar amb altres microserveis: inventari, pagament, enviament i marketing.
- Millorar la capa de seguretat: OAuth2, SSO, paginació als *endpoints*.
- Utilitzar Amazon Web Services.
- Moure els *DTOs* a una llibreria externa.
- Afegir tests i millorar la cobertura del codi.
- Habilitar internacionalització: idioma i moneda.
- Documentar els *endpoints* a Swagger.

Conclusions

Metes aconseguides:

- Creació d'un sistema de microserveis seguint principis SOLID.
- Implementació de serveis addicionals: *API Gateway* i *Discovery Service*.
- Aprofundiment i estudi dels *Frameworks*: Spring i ReactJS.
- Configuració d'un procés d'Integració Contínua.
- Desplegament del sistema a Heroku i GitHub Pages.
- Investigació de Reactor i NoSQL.

Demo!

Gràcies!