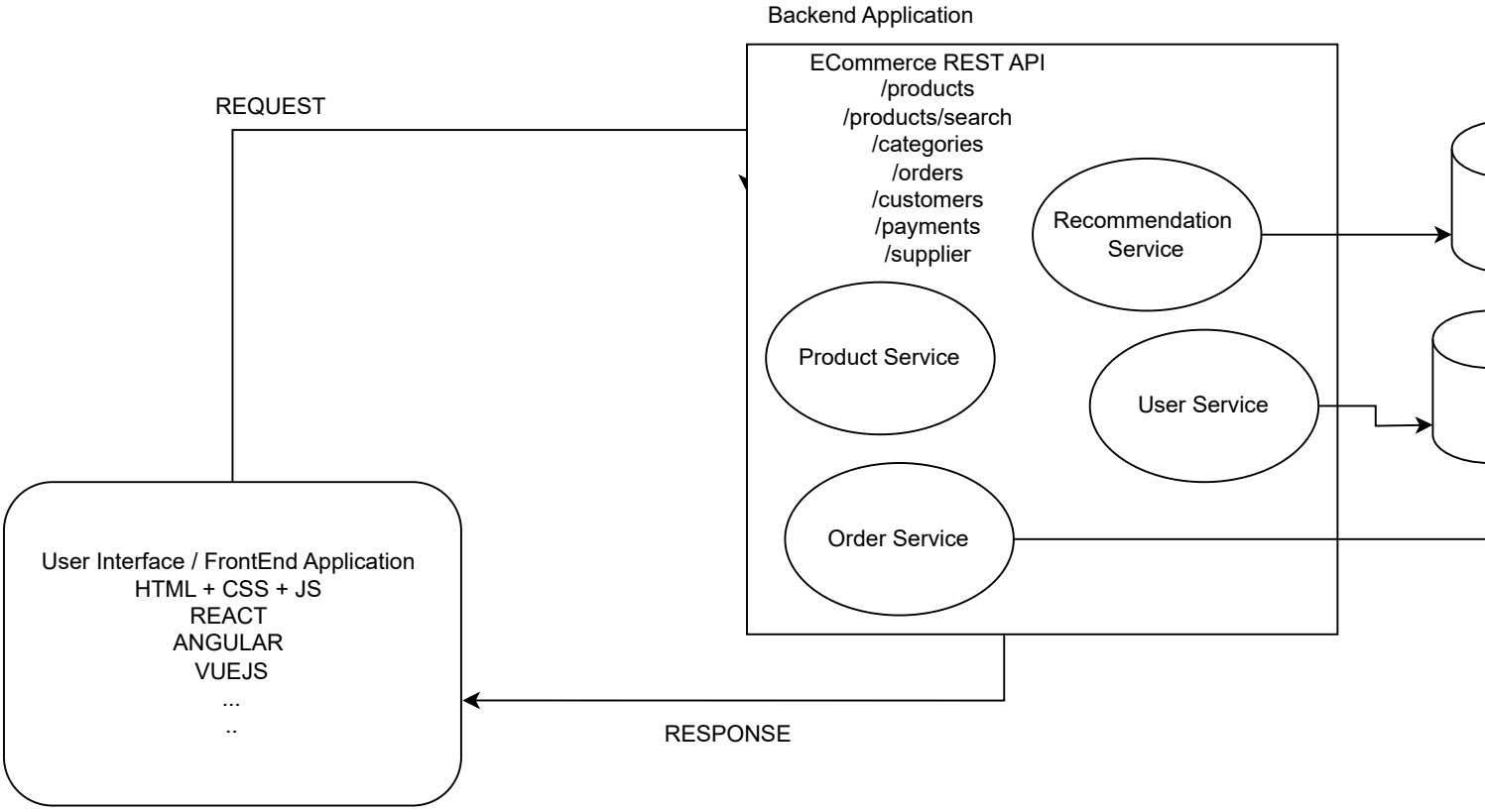
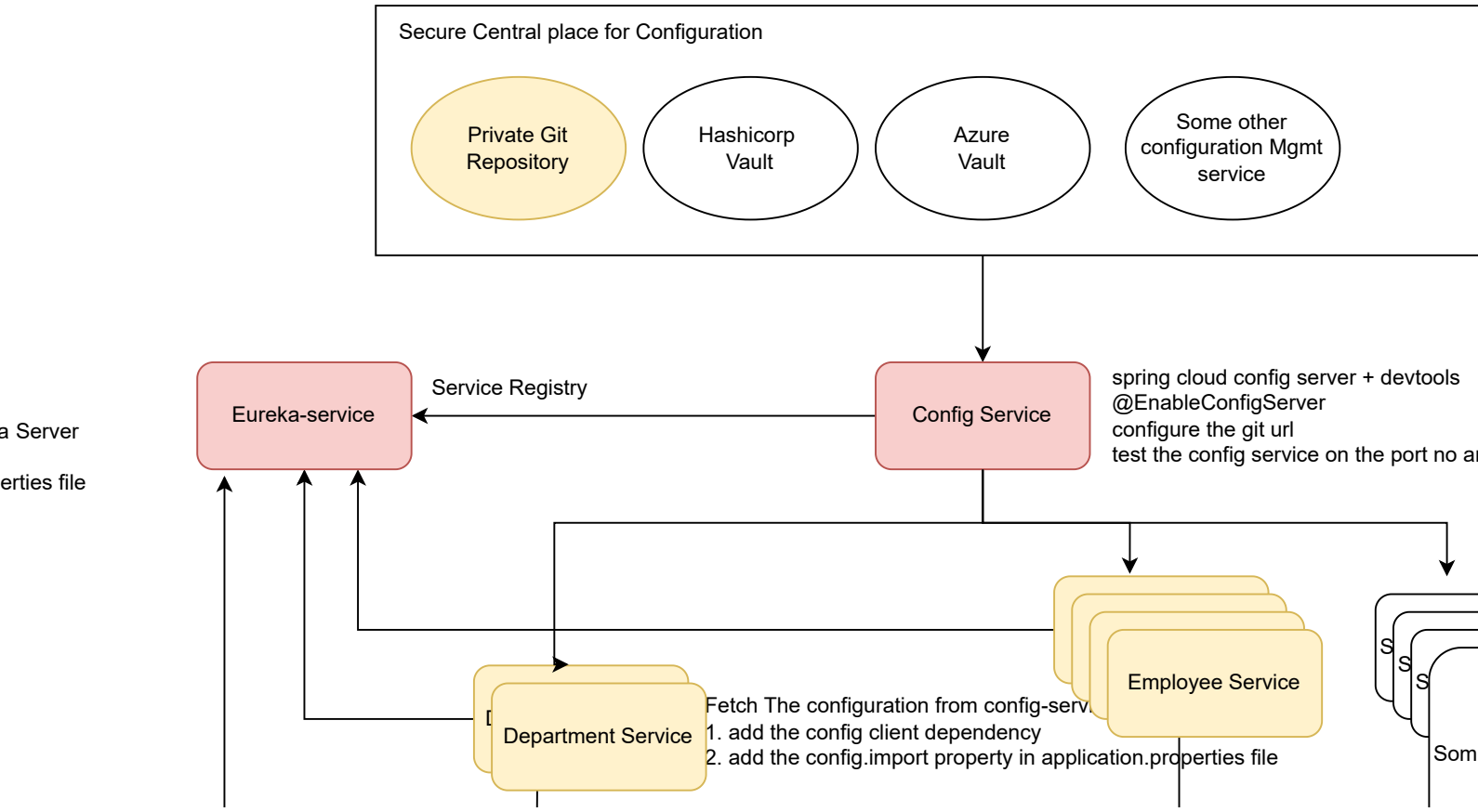
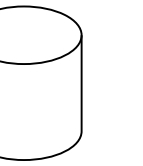
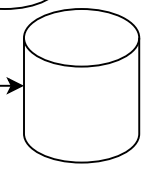
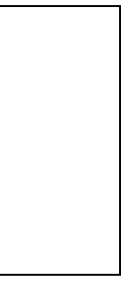
1. create a new application + devtools + Eureka
2. @EnableEurekaServer
3. add some properties to the application. prop


Backend Application
ECommerce REST API
/products
/products/search
/categories
/orders
/customers
/payments
/supplier
REQUEST
Recommendation Service
Product Service
User Service
Order Service
User Interface / FrontEnd Application
HTML + CSS + JS
REACT
ANGULAR
VUEJS
...
..
RESPONSE

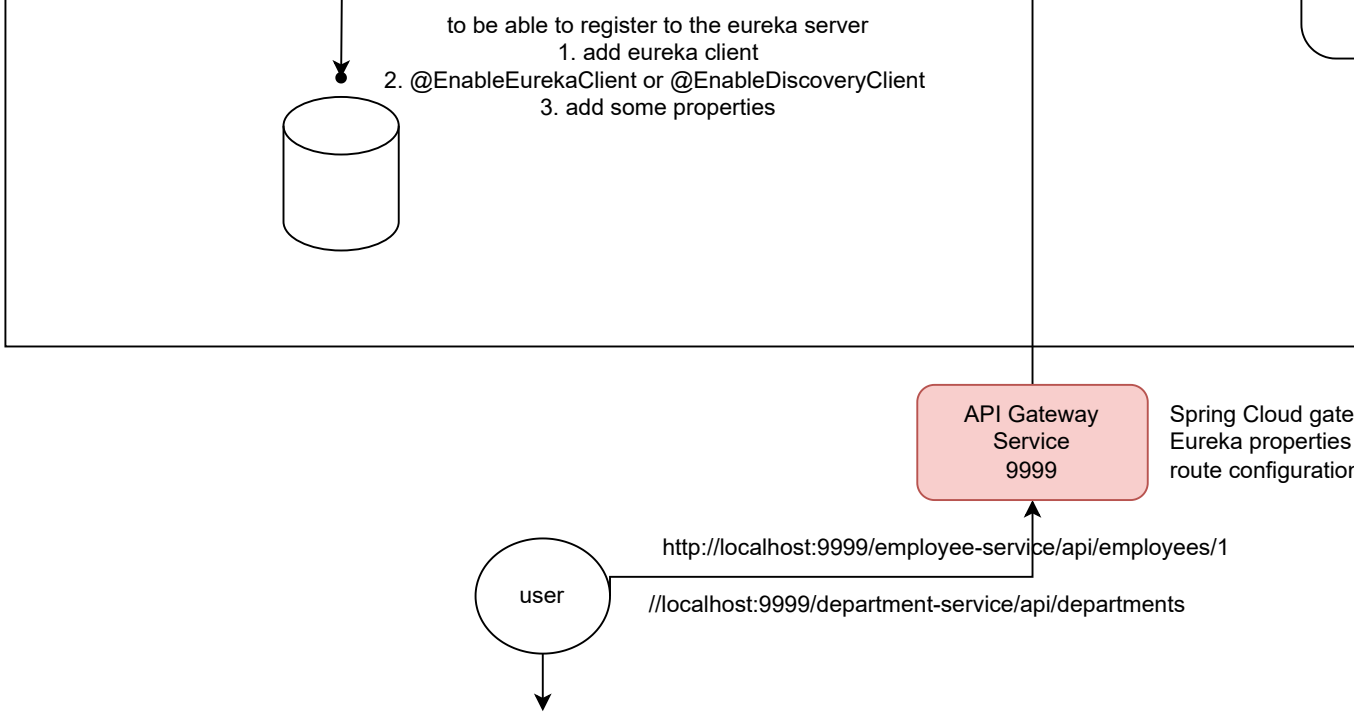

Secure Central place for Configuration
Private Git Repository
Hashicorp Vault
Azure Vault
Some other configuration Mgmt service
Eureka-service
Service Registry
Config Service
spring cloud config server + devtools
@EnableConfigServer
configure the git url
Employee Service
Department Service
Fetch The configuration from config-serv
1. add the config client dependency
2. add the config.import property in application.properties file





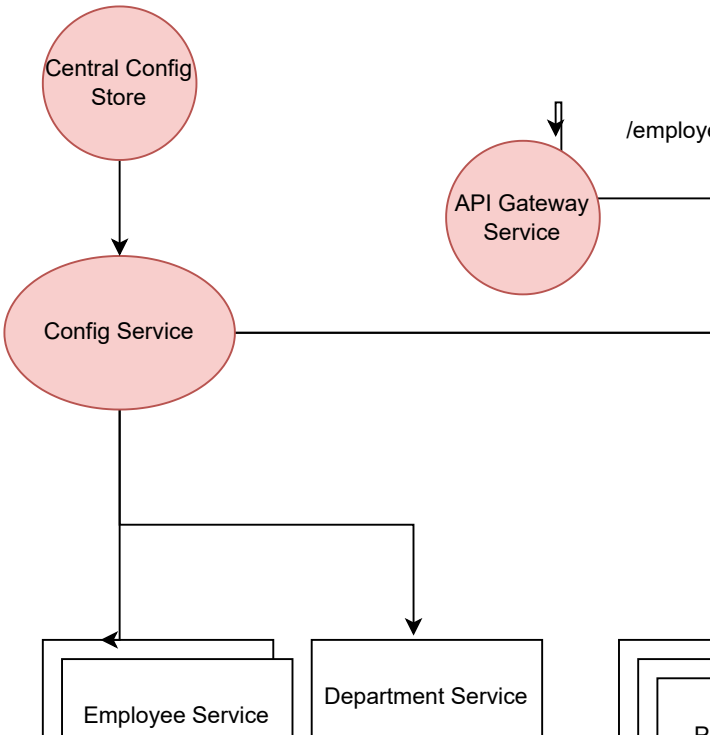


nd verify it pulls the configuration from the source


to be able to register to the eureka server
1. add eureka client
2. @EnableEurekaClient or @EnableDiscoveryClient
3. add some properties

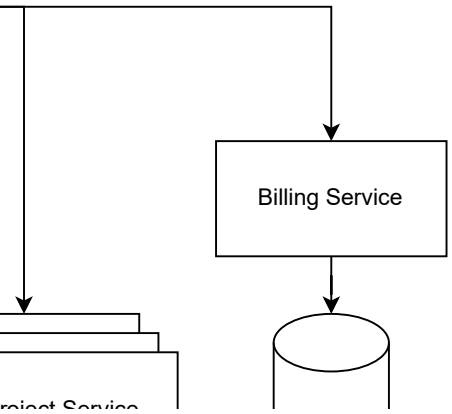
API Gateway
Service
9999
Spring Cloud gate
Eureka properties
route configuration

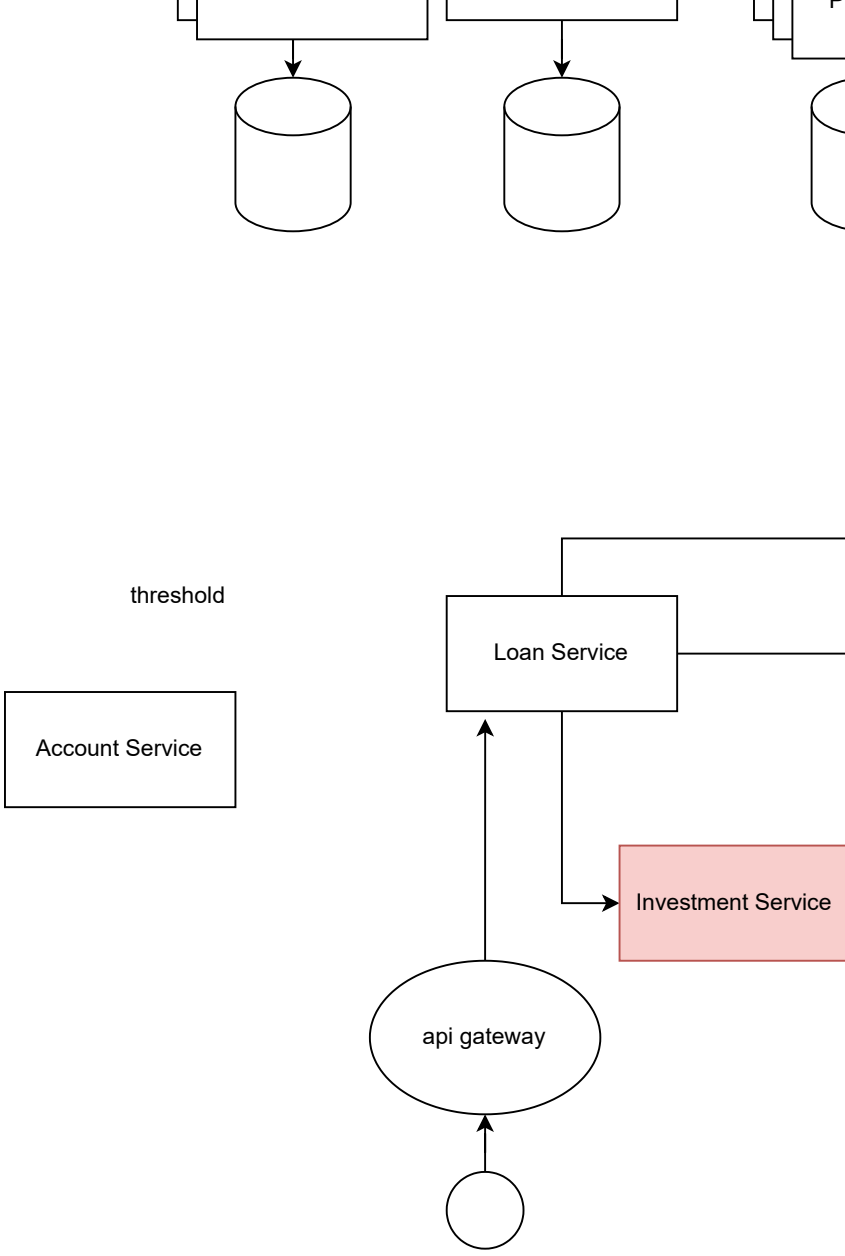
user
http://localhost:9999/employee-service/api/employees/1
//localhost:9999/department-service/api/departments

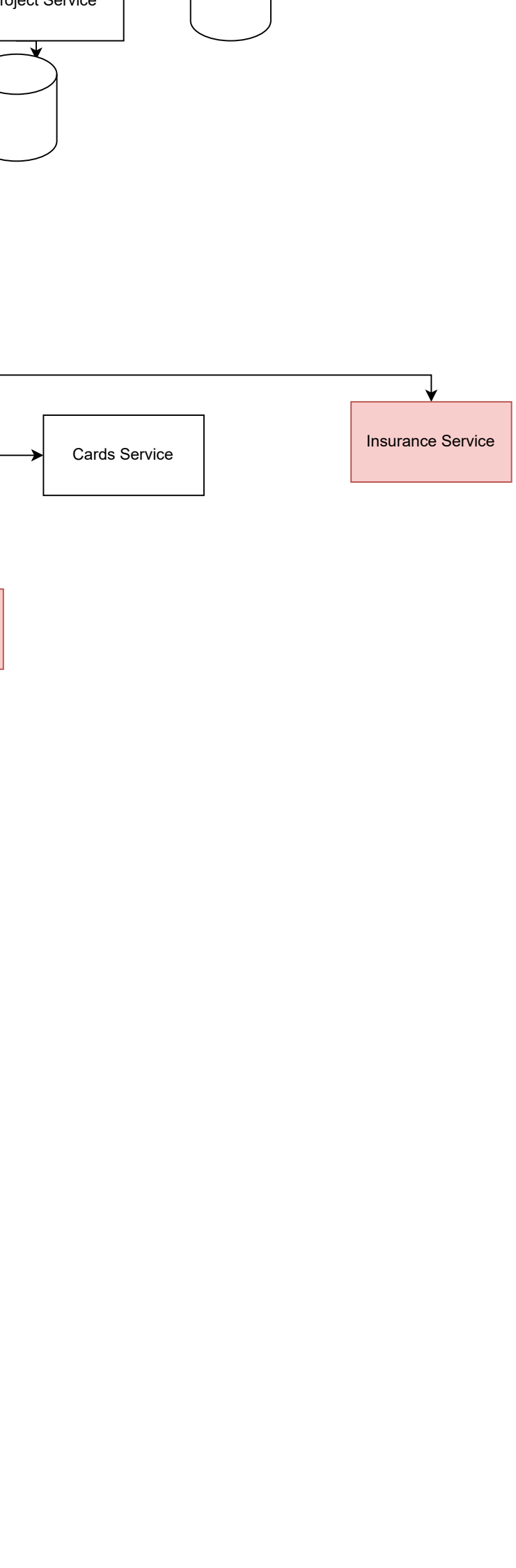

Central Config
Store
/employ
API Gateway
Service
Config Service
Employee Service
Department Service

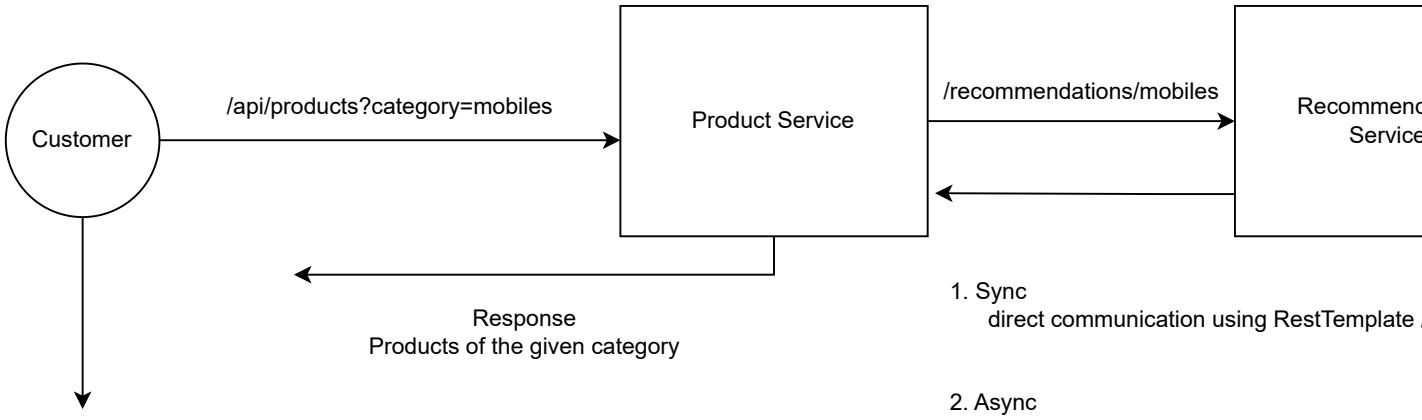
way + eureka client + config client + actuator + devtools
to get this service registered
n

ee/create

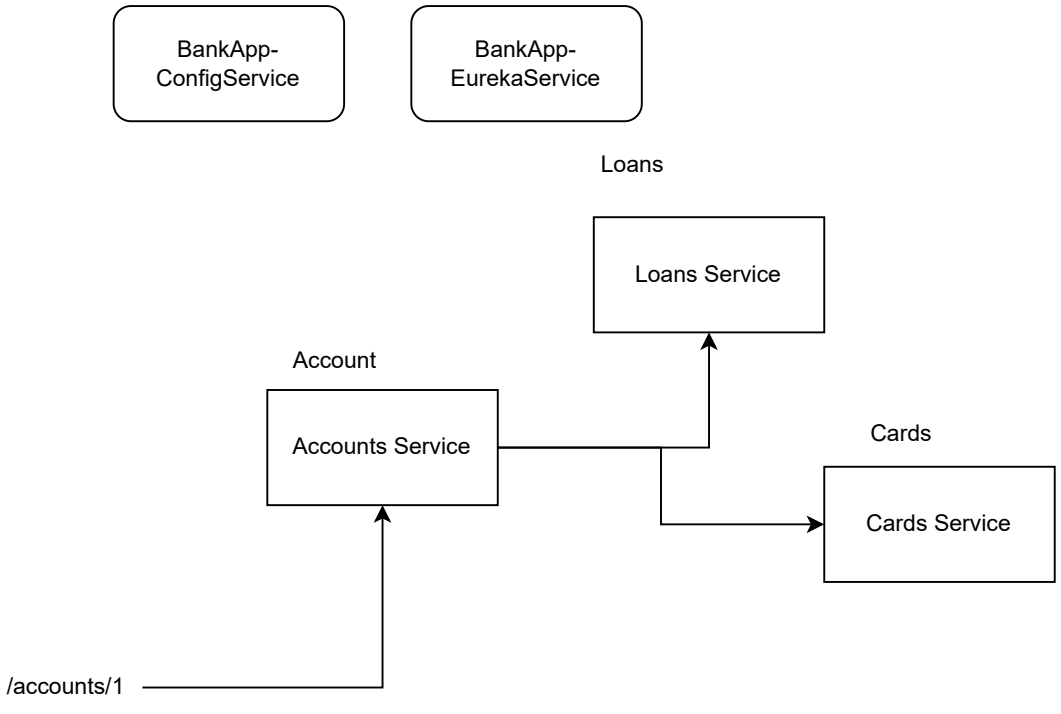



threshold
Loan Service
Account Service
Investment Service
api gateway


roject Service
Cards Service
Insurance Service


Customer
/api/products?category=mobiles
Product Service
/recommendations/mobiles
Recommend
Service
Response
Products of the given category
1. Sync
direct communication using RestTemplate
2. Async
- using messaging services like RabittMQ

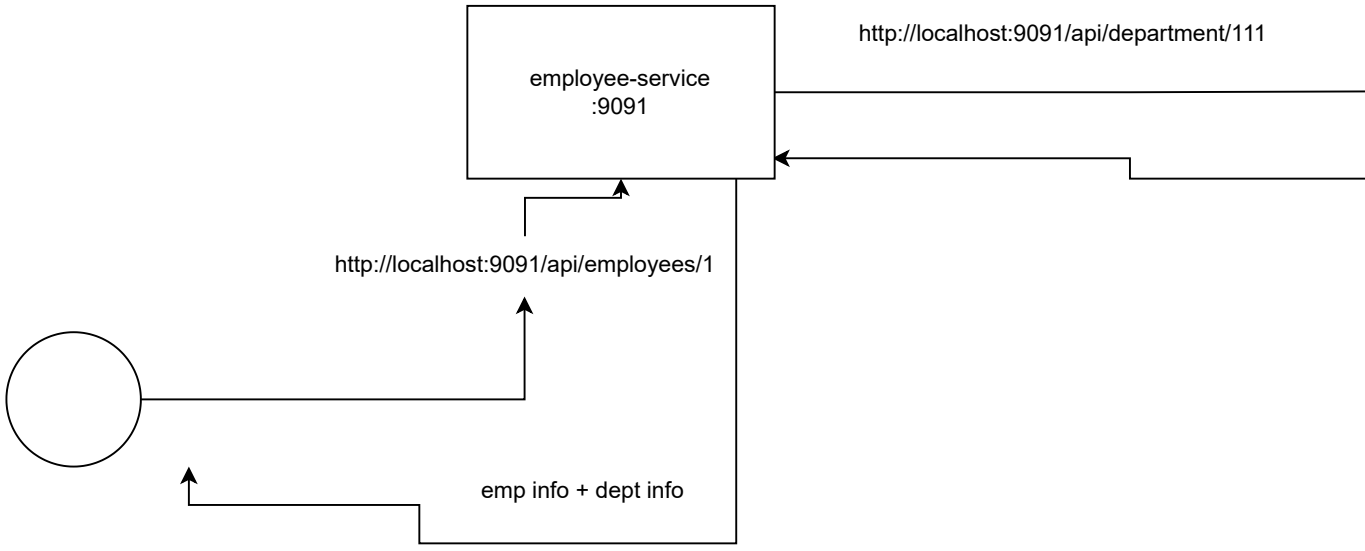
Payment Service

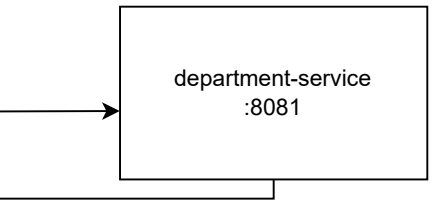

BankApp-
ConfigService
BankApp-
EurekaService
Loans
Loans Service
Account
Accounts Service
Cards
Cards Service
/accounts/1

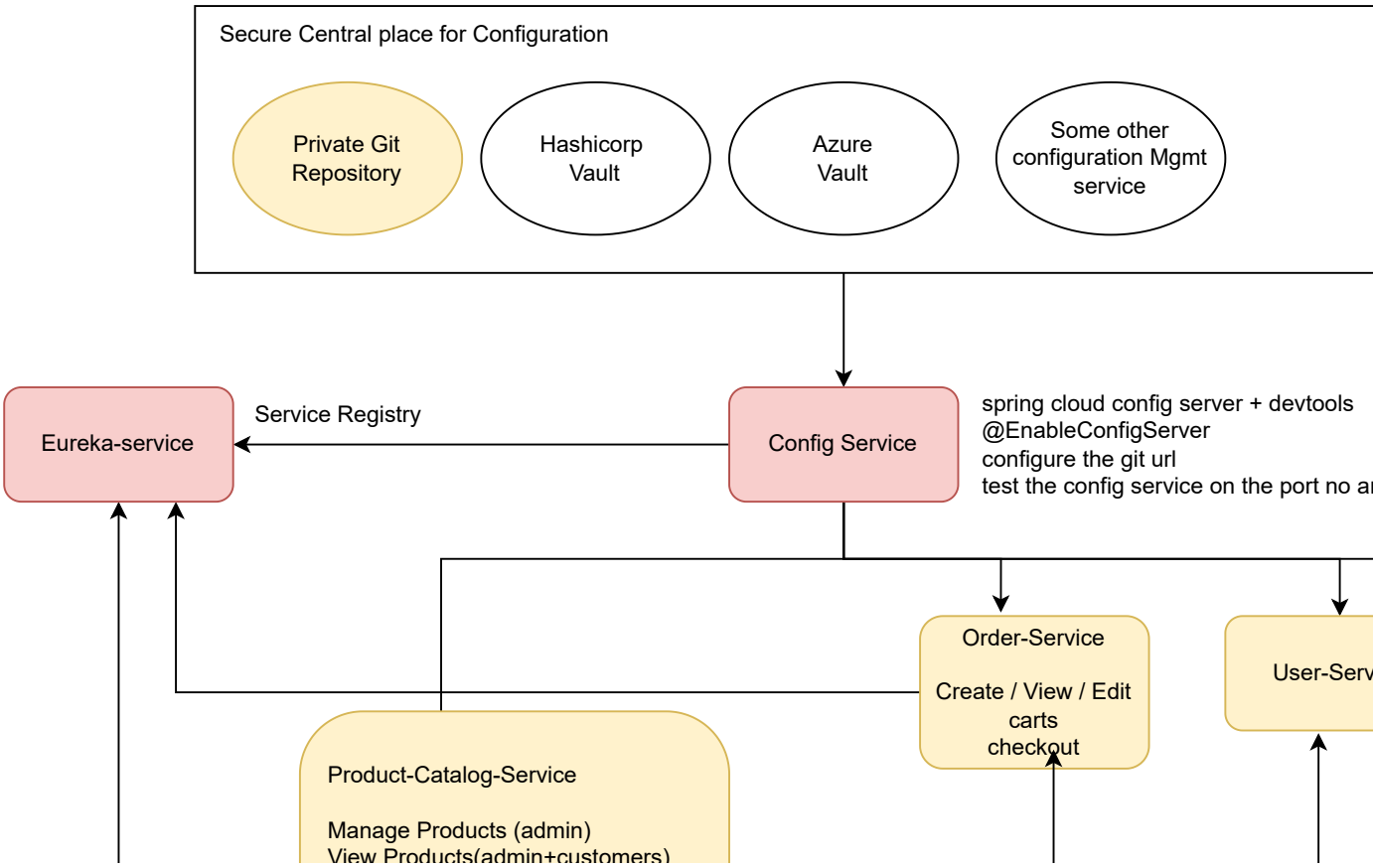
dation
e

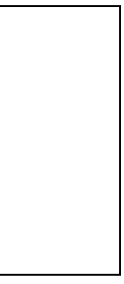
/ WebClient / OpenFeign / or use any HttpClient library

/ Apache Kafka / AWS SQS / Google PubSub / Azure Service Bus

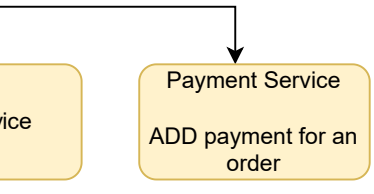

employee-service
:9091
http://localhost:9091/api/department/111
http://localhost:9091/api/employees/1
emp info + dept info


department-service
:8081

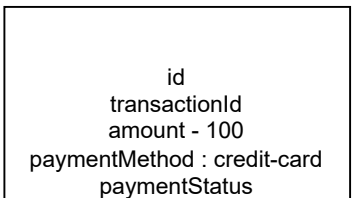

Secure Central place for Configuration
Private Git Repository
Hashicorp Vault
Azure Vault
Some other configuration Mgmt service
Eureka-service
Service Registry
Config Service
spring cloud config server + devtools
@EnableConfigServer
configure the git url
test the config service on the port no a
Order-Service
Create / View / Edit carts
checkout
User-Serv
Product-Catalog-Service
Manage Products (admin)
View Products(admin+customers)

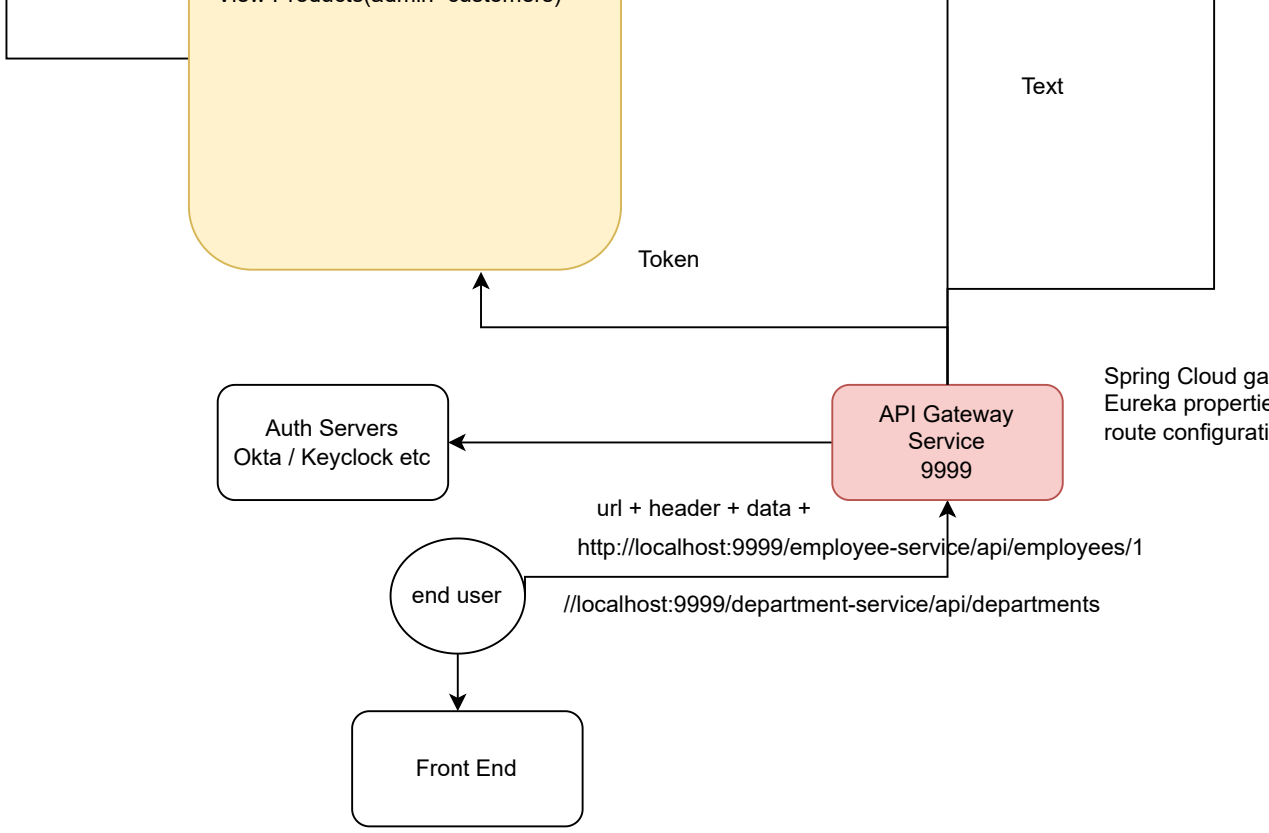


nd verify it pulls the configuration from the source



Payment




Text
Token
Auth Servers
Okta / Keyclock etc
API Gateway
Service
9999
Spring Cloud ga
Eureka propertie
route configurati
url + header + data +
http://localhost:9999/employee-service/api/employees/1
end user
//localhost:9999/department-service/api/departments
Front End

teway + eureka client + config client + actuator + devtools
es to get this service registered
on

Ca

Price

id

art

key - unique
customerId
date
Set<LineItem> lineItems;
totalprice - ?
currency
cartStatus ?

cartStatus - open / ordered / cancelled

Order

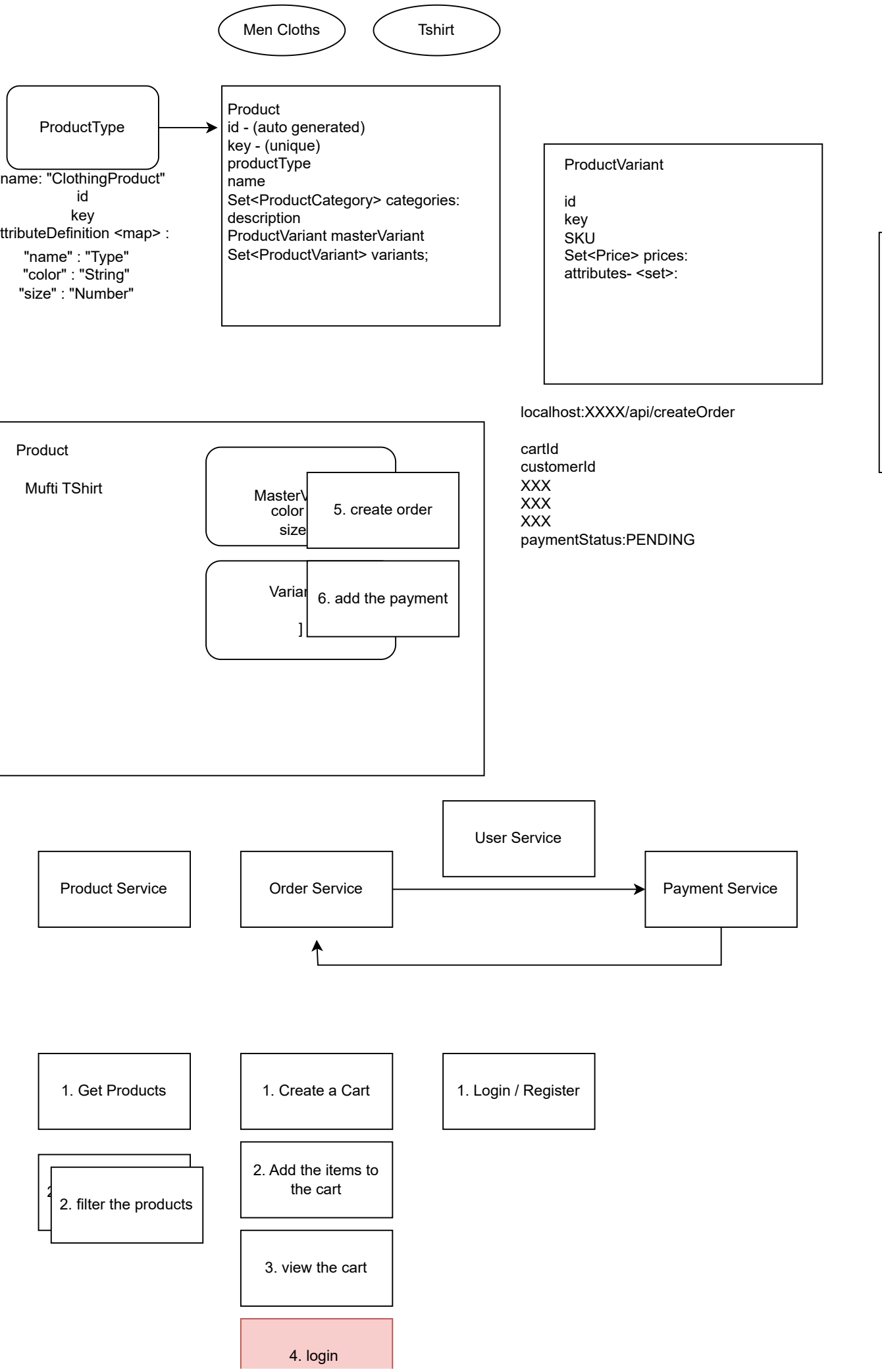
id - auto
cartId
totalPrice
paymentStatus
orderStatus

orderStatus : Created / Confirmed / Delivered / Cancelled

LineItem

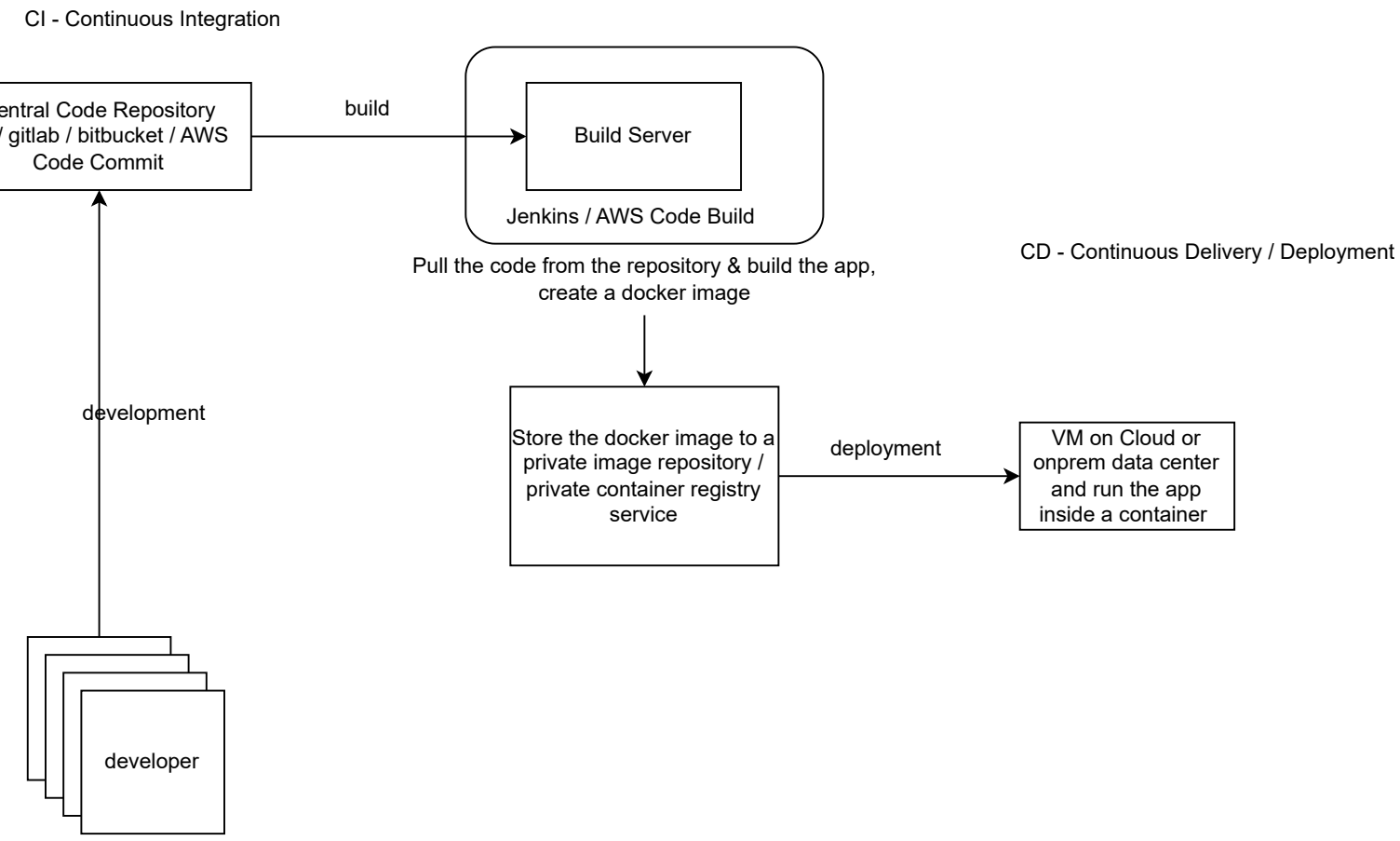
id - auto
productKey
variantKey
price
quantity
totalprice

a


Men Cloths
Tshirt
ProductType
name: "ClothingProduct"
id
key
ttributeDefinition <map> :
"name" : "Type"
"color" : "String"
"size" : "Number"
Product
id - (auto generated)
key - (unique)
productType
name
Set<ProductCategory> categories:
description
ProductVariant masterVariant
Set<ProductVariant> variants;
ProductVariant
id
key
SKU
Set<Price> prices:
attributes- <set>:
localhost:XXXX/api/createOrder
cartId
customerId
XXX
XXX
XXX
paymentStatus:PENDING
Product
Mufti TShirt
MasterV
color
size
5. create order
Varian
]
6. add the payment
User Service
Product Service
Order Service
Payment Service
1. Get Products
2. filter the products
1. Create a Cart
2. Add the items to the cart
3. view the cart
4. login
1. Login / Register

key
sku
centAmount
currency

Ce
git


CI - Continuous Integration
entral Code Repository
/ gitlab / bitbucket / AWS
Code Commit
build
Build Server
Jenkins / AWS Code Build
CD - Continuous Delivery / Deployment
Pull the code from the repository & build the app,
create a docker image
development
Store the docker image to a
private image repository /
private container registry
service
deployment
VM on Cloud or
onprem data center
and run the app
inside a container
developer