

Ecommerce Store

By Following the Domain Driven Design, we can break the monolithic application into number of small services, including:

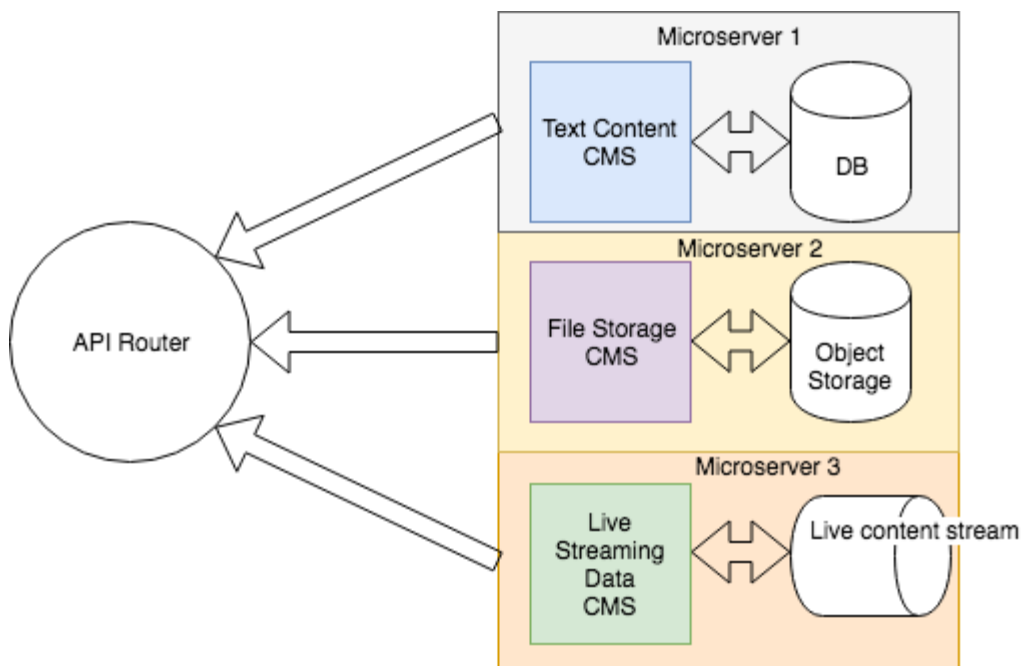
- An auth-service is serving the operations of signin, signup and signout.
- Simple CMS Service to manage the products
- Product Catalog Service
- Cart Service
- Order Service
- Payment Service
- Config Service – Provides Configuration for all other services
- Service Registration Service
- An API Gateway which is just responsible for routing the incoming requests to downstream services.

The databases are also aligned to Microservice architecture, each of the service may have their own databases

CMS – Micro Service

The aim is to create small, flexible and extendable content management system based pure on Spring Boot and MySQL or Mongo .

- It will expose REST APIs that will give access to products, related images, and documents.
- Client application will access by using [API Gateway](#) pattern



| Method | Path | Description |
|--------|--------------------------------------|--------------------------------------|
| GET | <code>/ {content-type}</code> | Get a list of {content-type} entries |
| GET | <code>/ {content-type} /:id</code> | Get a specific {content-type} entry |
| GET | <code>/ {content-type} /count</code> | Count {content-type} entries |
| POST | <code>/ {content-type}</code> | Create a {content-type} entry |
| DELETE | <code>/ {content-type} /:id</code> | Delete a {content-type} entry |
| PUT | <code>/ {content-type} /:id</code> | Update a {content-type} entry |

Discovery Service

It allows automatic detection of network locations for service instances, which could have dynamically assigned addresses because of auto-scaling, failures and upgrades

Gateway Service

Provide a proxy (routing) and client side load balancing via ribbon You can deploy multiple services for the same service and gateway will load balancing between them (Simple scalability) Details: [Spring Cloud Gateway](#)

Online Payment

Support Paypal, Stripe integration

Search

Flexible domain search capabilities in Modern-Ecommerce are provided through integration with elasticsearch

Product Service

| Method | Path | Description | User authenticated | Role |
|--------|-----------------------------------------------------|-------------------------------------------|--------------------|----------------|
| GET | /list?page={page}&pageSize={pageSize} | Get all products | x | Any |
| GET | /list/merchant?page={page}&pageSize={pageSize} | Get products created by merchant | x | merchant,admin |
| POST | /save | Create new product | x | Merchant |
| PUT | /save | Update existing product | x | Merchant |
| GET | /list/available?page={page}&pageSize={pageSize} | Get available products with inventory > 0 | x | Any |
| GET | /list/not-available?page={page}&pageSize={pageSize} | Get available products with | x | Any |

inventory
= 0

Order Service

| Method | Path | Description | User authenticated | Role |
|--------|--------------------------------------------|-----------------------|--------------------|--------|
| GET | /list/user?page={page}&pageSize={pageSize} | Get login user orders | x | Any |
| GET | /orderId | Get order by id | x | Any |
| POST | /save | Create new order | x | client |
| PUT | /save | Update existing order | x | client |

Product Catalog Service

| Method | URI | Description | Parameters | Request JSON | Response JSON |
|--------|---------------------------|---------------------------------------|------------|--------------|---------------|
| GET | /products/recommendations | List of recommended products | None | [TODO] | |
| GET | /products/{id} | Fetch product information based on id | None | [TODO] | |
| PUT | /products | Adds new product | [TODO] | [TODO] | |
| POST | /products/{id} | Updates existing product | [TODO] | [TODO] | |

Cart Microservice

REST API

Cart REST API supports following operations,

| Method | URI | Description | Parameters | Request JSON | Response JSON |
|--------|------------|-------------------------|------------|--------------|---------------|
| GET | /cart/{id} | Fetches cart by id | None | [TODO] | |
| POST | /cart/{id} | Creates or updates cart | [TODO] | [TODO] | |

