Shopizer commerce

Reference Architecture document

v 2.1

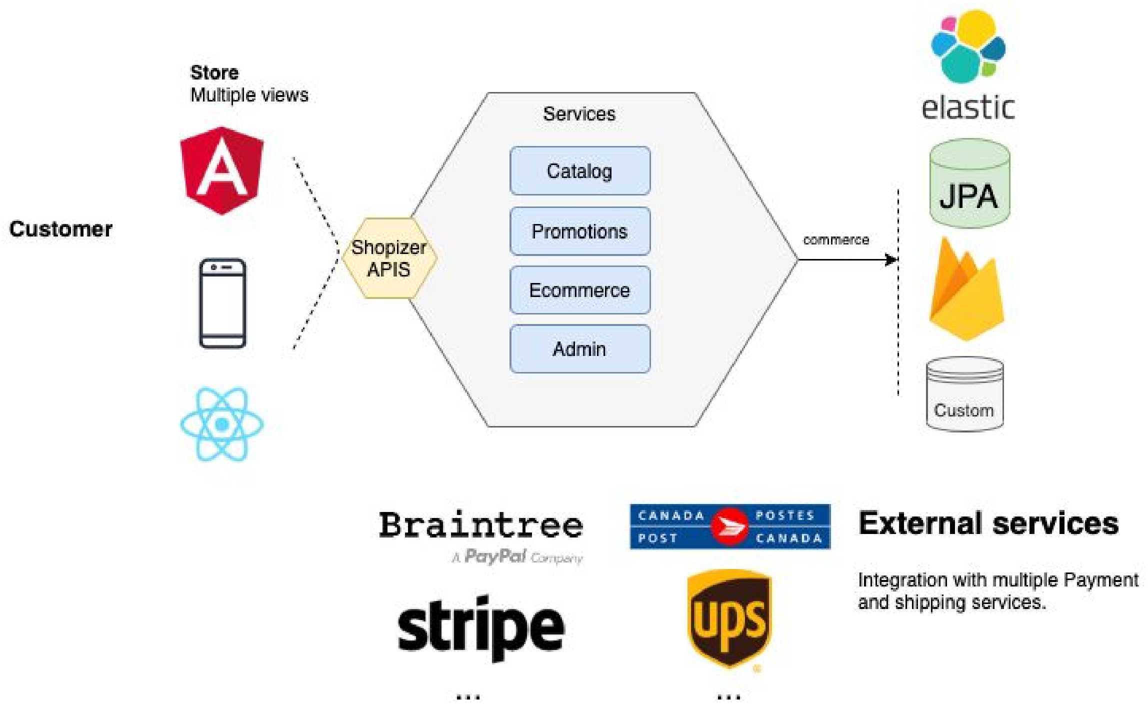
Project description

Shopizer is enterprise open source e-commerce software for retailers who want flexibility, speed and control of their commerce platform. Shopizer is a software solution that gives organizations the ultimate flexibility to take an experience-first approach to commerce, with simple powerful APIs and built in stores models.

Key technical benefits

- Headless commerce
- Services built with Spring framework (Spring boot, Spring Security)
- Front end Angular - React
- Vulnerability checks
- Open source and open standards
- Cloud Ready: Deploy Shopizer in the public
  or private cloud on Amazon Webservices (AWS), Microsoft
  Azure or Google Cloud Platform (GCP)
- Run on premise servers
- Run from images

Shopizer Conceptual



Shopizer is a platform aiming to provide services, tools and connectors for building your own private commerce cloud or on premise commerce application. Services are built using Spring Framework providing packages for building enterprise applications. Shopizer persistence uses JPA and has extensions for Google Firebase NoSQL database and a set of interfaces for connecting to other external data sources.

Shopizer supports integration with external payment and shipping modules such as Stripe, Fedex, Braintree, USPS and more. A set of modules extensions allows using various content management storage such as JBoss Infinispan, AWS S3 and external web servers such as NGINX or Apache server. Elastic tools provide searching functionality.

A REST api exposes all commerce functionality (B2C, B2B, C2C, Multi-Stores) as well as complete system administration. Spring Security configured of the box with JWT Bearer token authentication provides application interface authentication and authorization.

Components view (JSP)

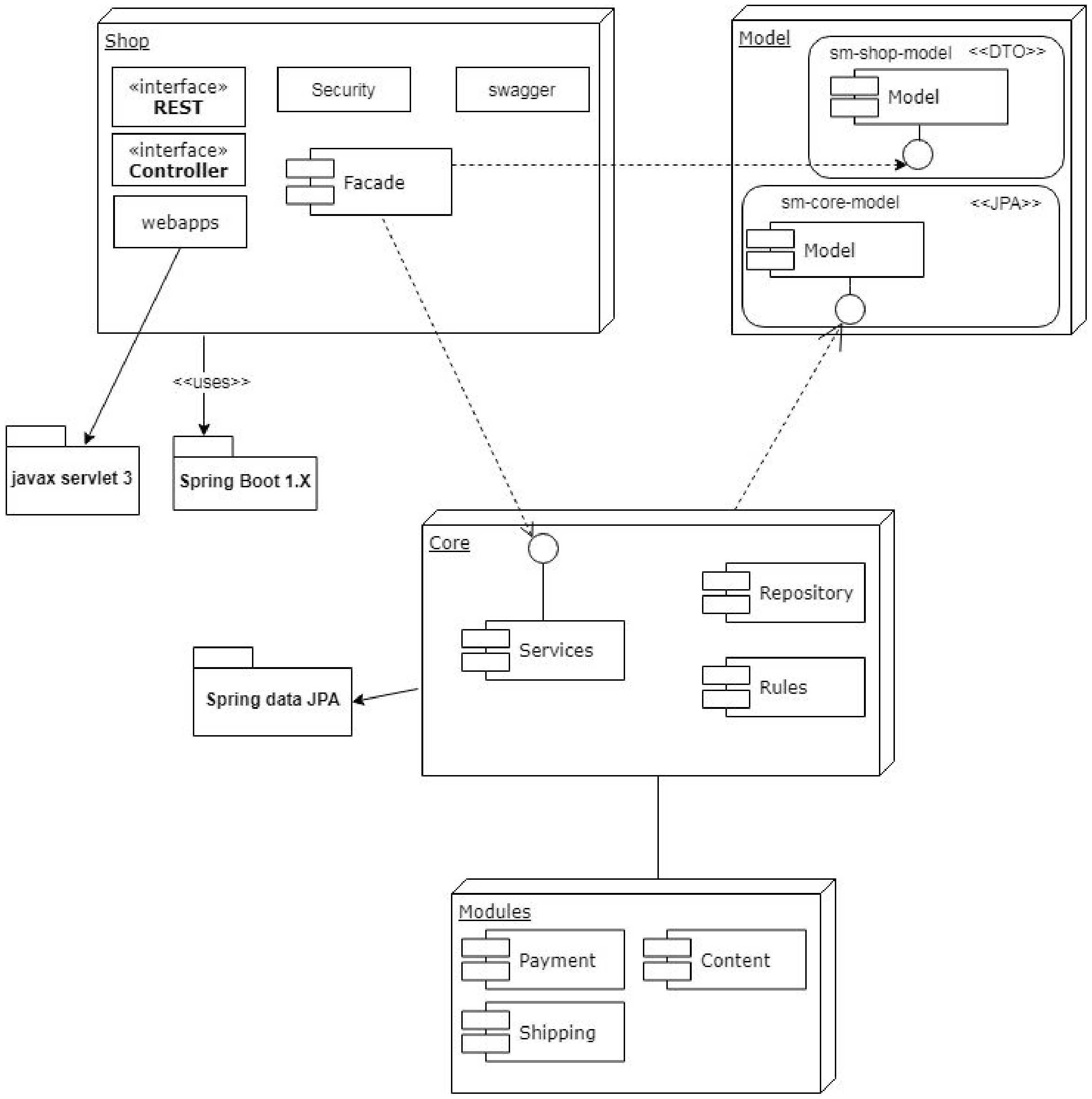
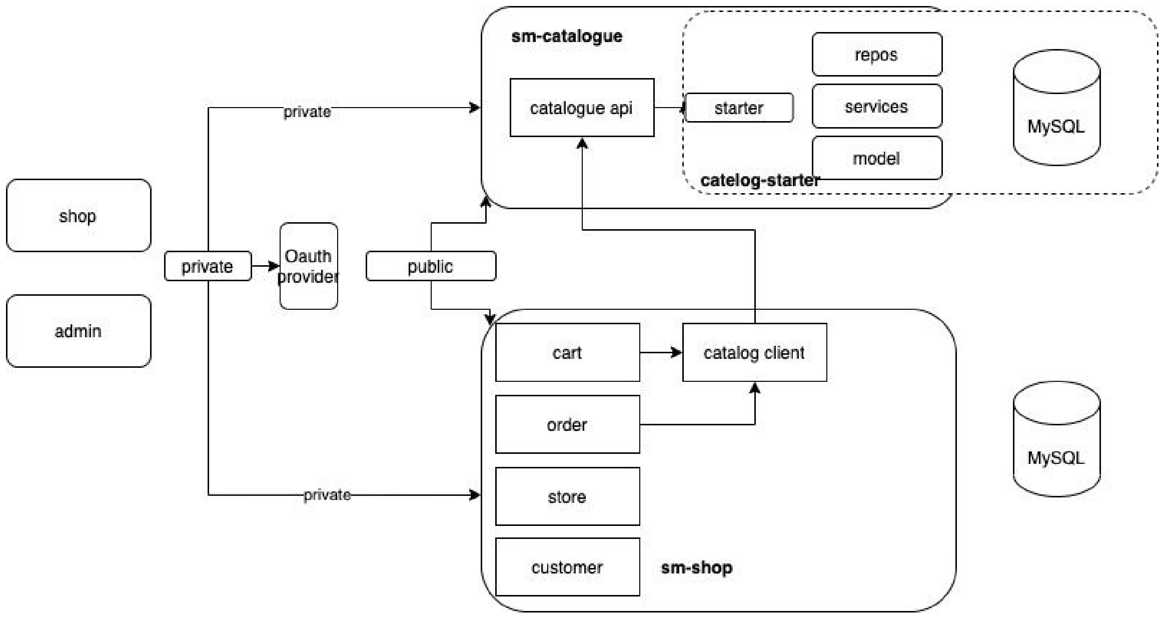


Figure 2 - Shopizer components

The Component Model describes the entire hierarchy of components in terms of their responsibilities, their interfaces, their (static) relationships, and the way they collaborate to deliver required functionality.

Components view (Target)



Target software architecture (ongoing) includes the creation of microservices starting with catalogue view and administration. Authentication provider connectors to GCP (Firebase), Azure, and AWS services such as Cognito.

Components view

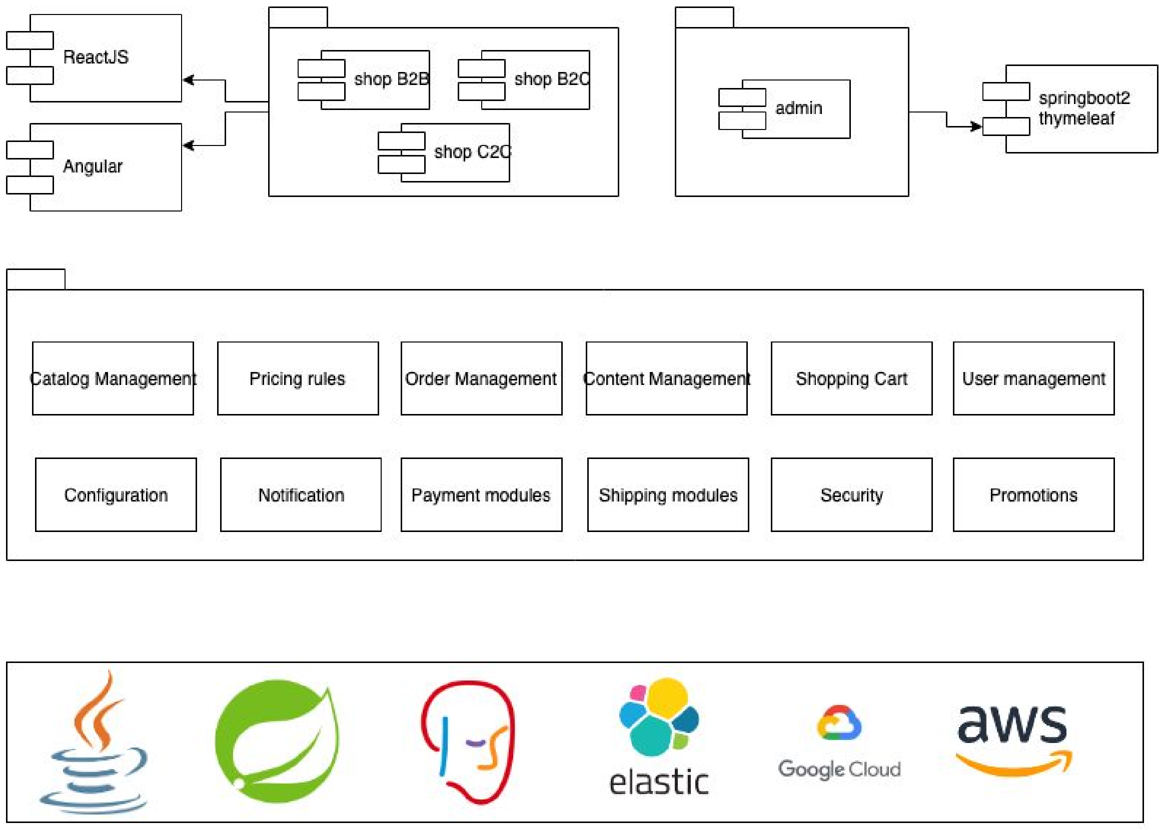


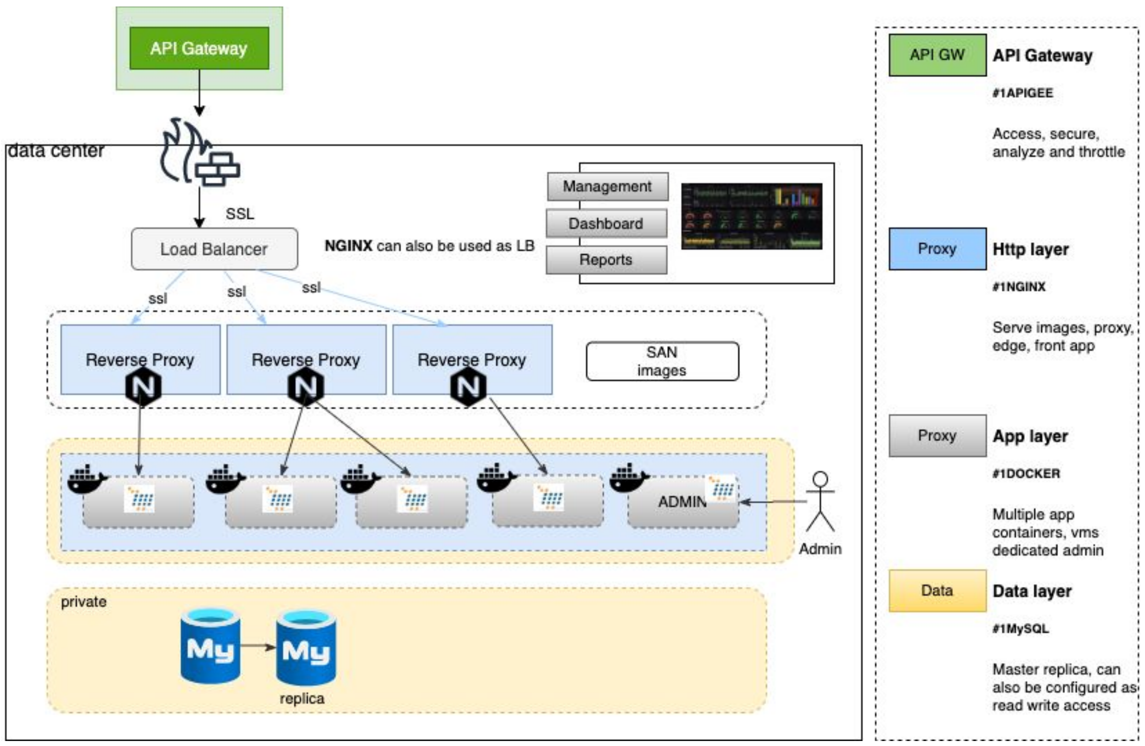
Figure 3 - Logical view

B2B, B2C and C2C functionality offering the following services

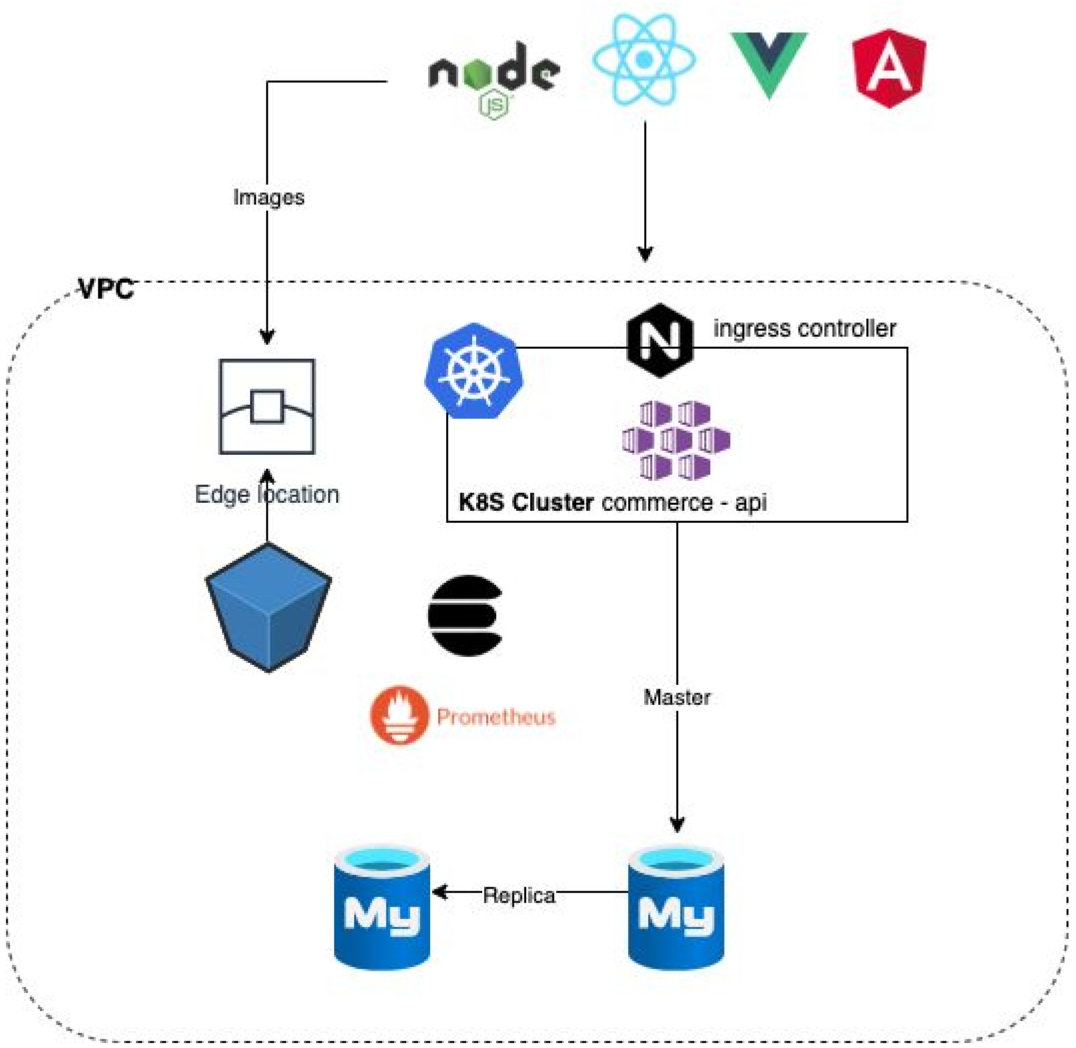
- Catalog management
- User management
- Customer management
- Content management
- Order management
- Pricing management
- Shopping cart management
- Configuration management
- Shipping
- Payment
- Promotions
- Search

Technology stack is built on Java (requires java 8), Spring Boot (Core, Security, DATA JPA), JBoss Drools rules engine, Elastic Search and cloud services offered by GCP and AWS for infrastructure, storage and security.

Reference architecture and infrastructure - On premise



Reference architecture - infrastructure - Cloud


Images
VPC
ingress controller
Edge location
K8S Cluster commerce - api
Prometheus
Master
Replica