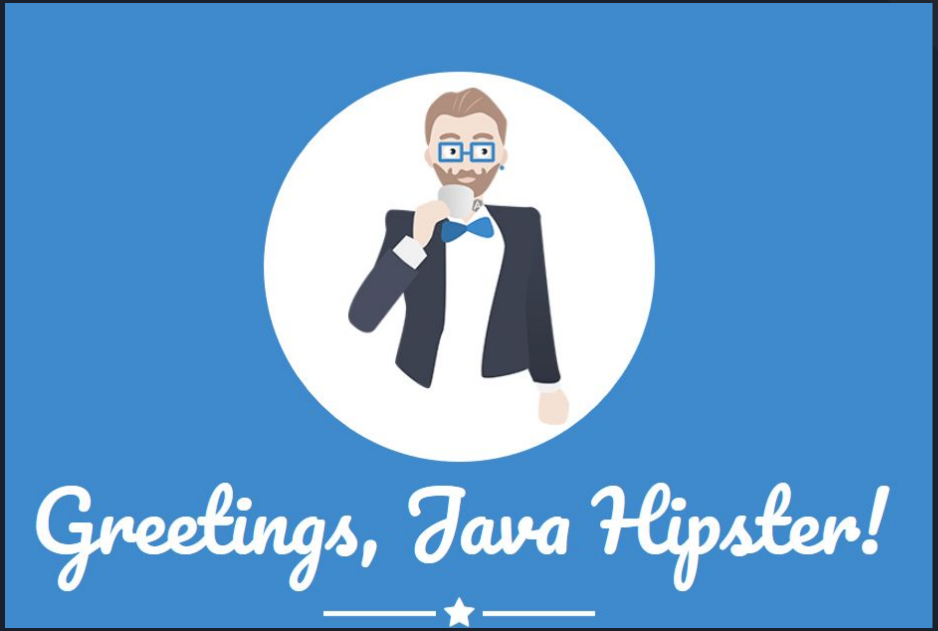
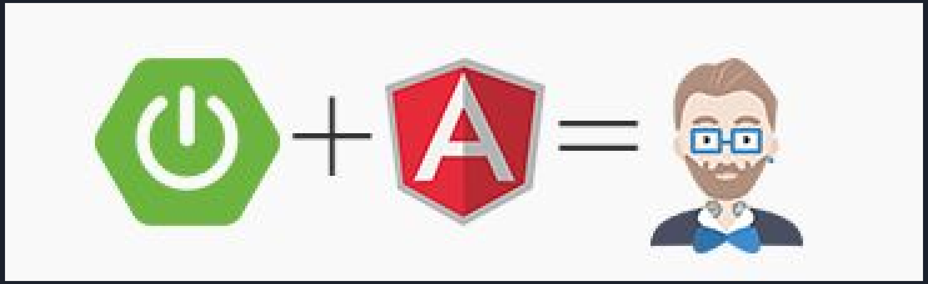


Workshop JHipster

Joey de Vlieger

Niels Killaars

What is JHipster?



Generate a complete and modern Web app or microservice architecture, unifying:

- A high-performance and robust Java stack on the server side with **Spring Boot**
- A sleek, modern, mobile-first front-end with **Angular** and **Bootstrap**
- A powerful workflow to build your application with Yeoman, Webpack/Gulp and Maven/Gradle

The advantage of JHipster

Write less code

Server side

- Spring Boot for easy application configuration
- Spring Security
- Spring MVC REST + Jackson
- Spring Websocket support
- Spring Data JPA + Bean Validation
- Spring Test Context Framework

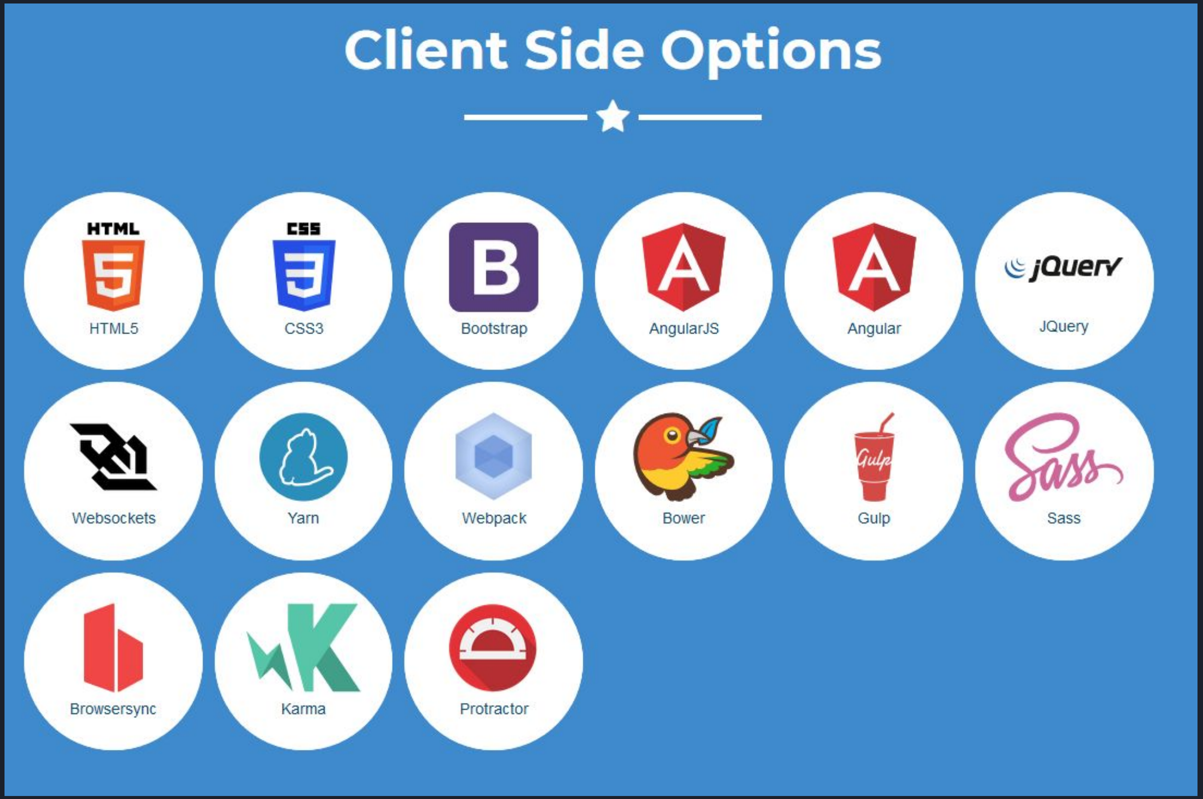
- Maven or Gradle
- Database updates with Liquibase
- Elastic Stack
- MongoDB document-oriented NoSQL database
- Cassandra column-oriented NoSQL database
- Kafka publish-subscribe messaging system

Server Side Options
Spring Boot
Spring Security
Netflix OSS
Consul
Gradle
Maven
Hibernate
Liquibase
MySQL
MariaDB
PostgreSQL
Oracle
MS SQL
MongoDB
Cassandra
EhCache
Hazelcast
Infinispan
ElasticSearch
Kafka
Swagger
ELK Stack
Prometheus
Thymeleaf
Gatling
Cucumber

Client side

- Single web page application with Angular 4
- Responsive Web Design with Twitter Bootstrap
- HTML5 Boilerplate
- Full internationalization support
- Optional Sass support for CSS design

- JavaScript libraries with Yarn or Bower
- Build, optimization and live reload with Browsersync and Webpack or Gulp.js
- Testing with Karma, Headless Chrome and Protractor
- Support for the Thymeleaf template engine, to generate web pages on the server side



Going into production

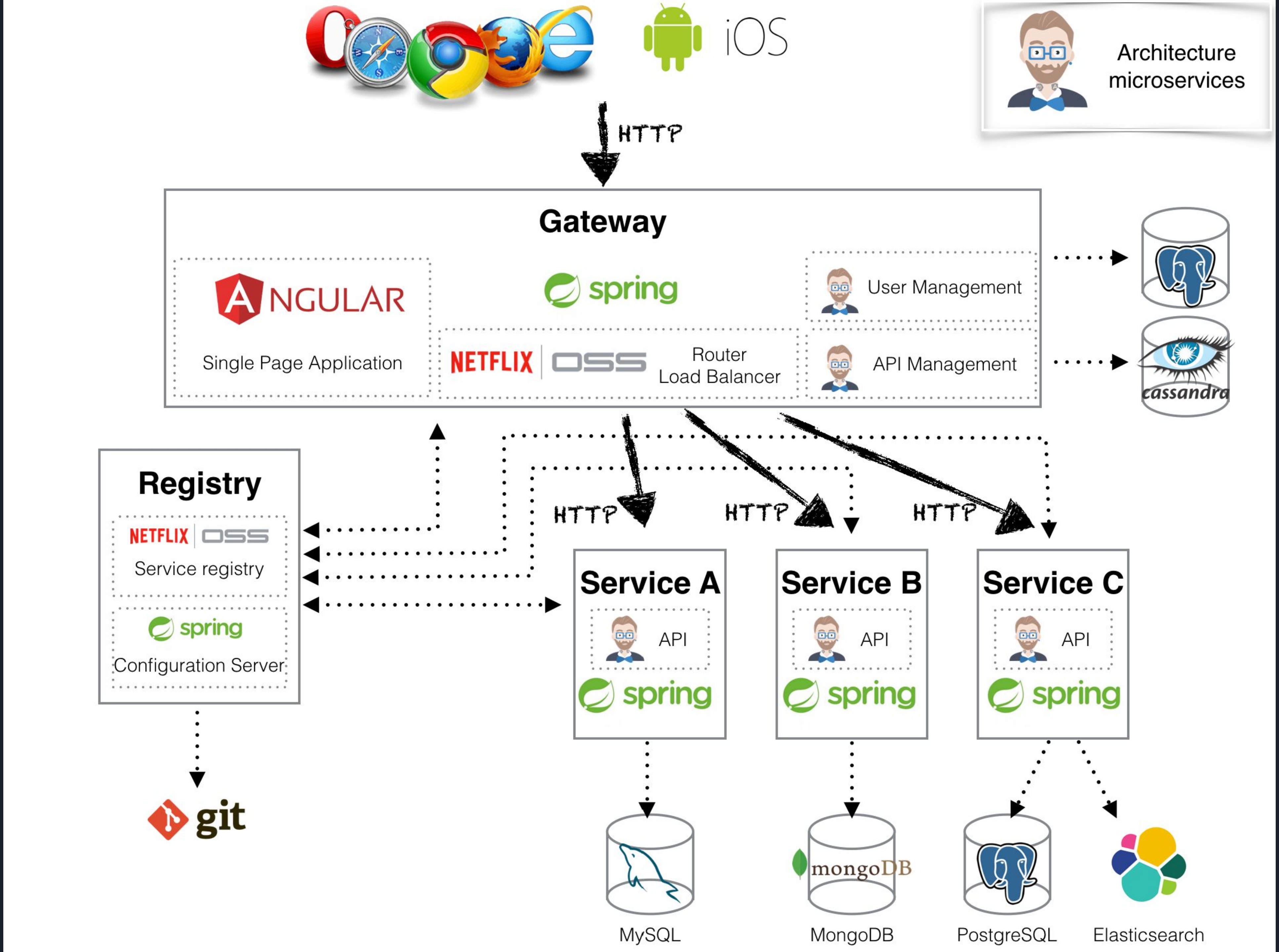
- Monitoring with Metrics
- Caching with ehcache (local cache), hazelcast or Infinispan
- Log management
- Database Connection pooling
- Builds a standard WAR file or an executable JAR file
- Full Docker and Docker Compose support
- Support for all major cloud providers: AWS, Cloud Foundry, Heroku, Kubernetes, OpenShift, Docker…

Deployment Options
Docker
Kubernetes
Heroku
CloudFoundry
AWS
Boxfuse
Rancher
OpenShift

Microservices

- Scaling
- Gateway (Handles Web Traffic)
- JHipster Registry (Configuration management)
- Service API Requests (REST)
- Separate Databases per service (Optional)

- HTTP routing using Netflix Zuul or Traefik (Load balancer)
- Service discovery using Netflix Eureka or HashiCorp Consul


iOS
Architecture microservices
HTTP
Gateway
ANGULAR
Single Page Application
spring
NETFLIX OSS
Router
Load Balancer
User Management
API Management
cassandra
Registry
NETFLIX OSS
Service registry
spring
Configuration Server
HTTP
HTTP
HTTP
Service A
API
spring
Service B
API
spring
Service C
API
spring
git
mongoDB
MySQL
MongoDB
PostgreSQL
Elasticsearch

Entities

Entity Properties

- A database table
- A JPA Entity
- A Spring Data JPA Repository
- A Spring MVC REST Controller, which has the basic CRUD operations
- An Angular router, a component and a service
- An HTML view
- Integration and Performance tests

Entity generation

- JDL Studio
  - Create UML-Shaped diagrams ready for import as entities
- JHipster Sub-generator
  - Create individual Entities

Practical Assignment:
Creating a Beverage Collection application

https://github.com/sebivenlo/jhipster

The end result

A generated web application

- Account Management
- User Management
- Built-in security
- Monitoring metrics

Example pages that present the data from your created beverage entities.

- Beverage Charts per user
- Optional styling

Any questions?