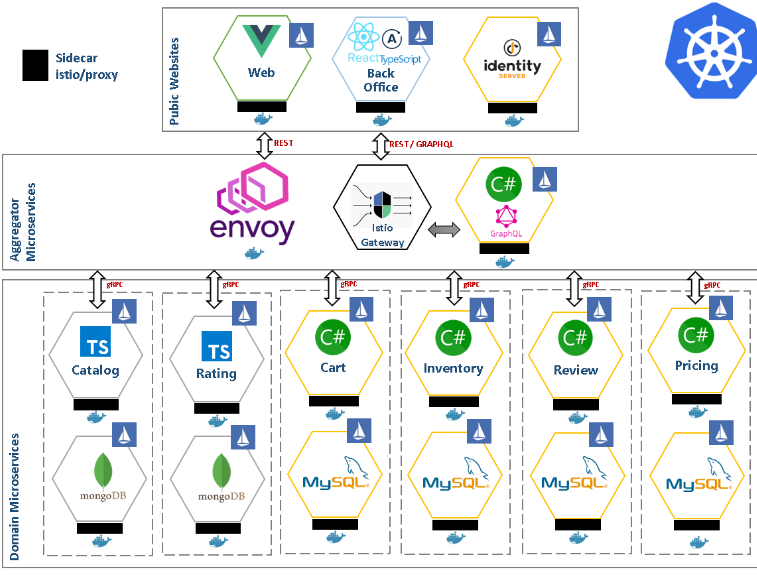
# High Level Software Architecture



Prerequisite

1. Window 10 – the OS for developing and building this demo application

2. Docker for desktop (Kubernetes enabled) – the easiest tool to run Docker, Docker Swarm and Kubernetes on Linux

3. Kubernetes /AKS – the app is designed to run on Kubernetes

4. Istio – application works on Istio services mesh

5. helm – the best package manager to find, share, and use software built for Kubernetes

6. .NET Core SDK 2.x - .NET Framework and .NET core, including ASP.NET and ASP.NET Core

7. nodejs 10.x JavaScript – Javascript superset of JavaScript that compiles to plain Javascript

8.identityserver – The Identity and Access Control solution for .NET core

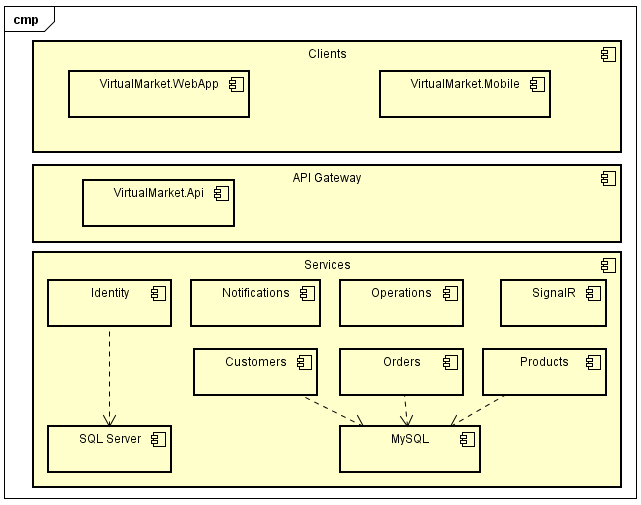
9.gRPC – a high-performance, open-source universal RPC framework

10. create – react - app: a modern web app by running one command

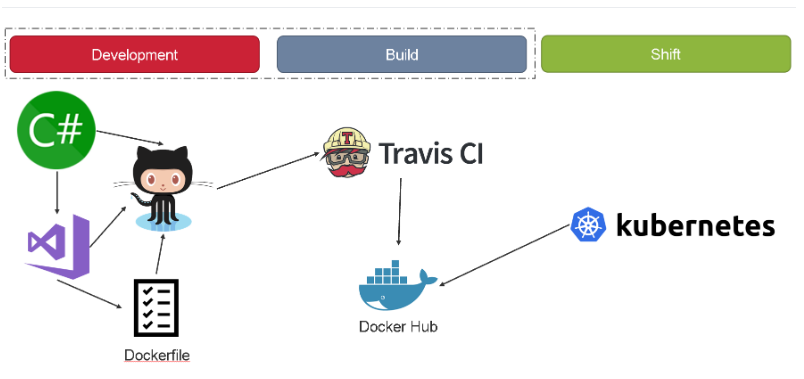
11. vue-cli – standard tooling for Vue.js development

12. Apollo-client – the best way to use GraphQL to build client applications

# Architecture



# CI/CD



**Consul** which have role a service discovery

is a distributed service mesh to connect secure, and configure service across any runtime platform and public or private cloud.

**Fabio** is an HTTP and TCP reserve proxy that configures itself with data from Consul

Traditional load balancers and reserve proxies need to be configure with a config file. The configuration contains the hostnames and paths the proxy is forwarding to upstream services. This process can be automated with tool like consul-template that generate config file and trigger a reload.

* Fabio work differently since it updates it routing table directly from the data stored in Consul as soon as there is a change and without restart and reloading.
* When you register a service in Consul all you need to add in is a tag that announces the paths the upstream service accepts, e.g: urlprefix-/user or urlprefix-/order and Fabio will do the test.

**Jaeger Tracing**

Jaeger is a CNCF distributed tracing system released by Uber that enables debugging, monitoring, and analysis of your services, and is based on OpenTracing API. It uses distributed context propagation, which essentially is the basis of distributed tracing, to assign metadata to request as they propagate through your system. List some advantages of using Jaeger:

* Performance and latency analysis.
* Service dependency analysis; you can view a DAG of your system in the UI
* Logs associated with each span
* Organization of logs into calling hierarchy
* Cost attribution
* Useful UI and libraries in Go, Node, C#, and others.
* Different levels of sampling can be configured to reduce load: constant , probabilistic, rate limiting, and remote.

**App metrics**

App Metrics is an open-source and cross-platform .NET library used to record metrics within an application. App Metrics can run on .NET Core or on the full .NET framework also supporting .NET 4.5.2 App Metrics abstract away the underlying repository of your Metrics for example InfluxDB, Prometheus, Graphite, Elasticsearch etc, by sampling and aggregating in memory and providing extensibility points to flush metrics to a repository at a specified interval. App Metrics provides various metrics types to measure thing such as the rate of request, counting the number of user login overtime, measure the time taken to execute a database query, measure the amount of free memory and so on. Metrics types supported are Apdex, Gauges, Counters, Meters, Histograms and Timers.App Metrics also provides a health checking system allowing you to monitor the health of you application through user defined checks.

**Redis**

Redis is an open source (BSD licensed, in-memory data structure store, used as a database, cache and message broker. It supports data structures such as strings, hashes, lists, sets, sorted set with range queries, bitmaps, hyperloglogs and geospatial indexes with radius queries. Redis has built-in replication, Lua scripting, transaction and different level of on-disk persistence, and provide high availability via redis Sentinel and automatic partitioning with Redis Cluster.

**RestEase**

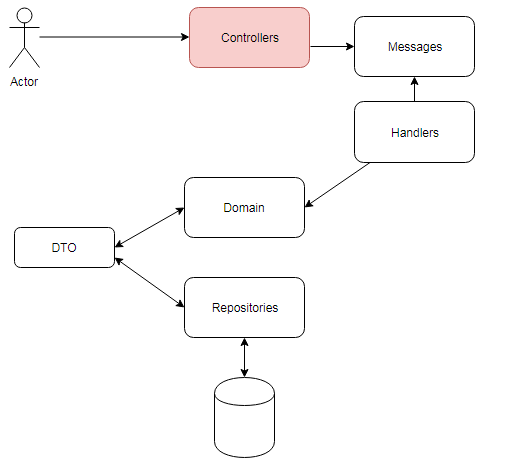
RestEase is type-safe REST clients for .NET framework and .NET core which aims to make interacting with remote REST endpoints easy, without adding unnecessary complexity.

**KeyVault**

KeyVault allow you to separate your configuration data into sensitive and non-sensitive information. For example, you could use Application Setting to store default setting but use KeyVault to store your API keys, passwords, and any Client Id’s for external services that your application depends on.

* VaultSharp is a .NET Standard 2.0 and NET 4.6 based cross-platform C# library that can be used in any .NET application to interact with Hashicorp’s Vault
* The Vault system is a secrete management system built as an Http Service by Hashicorps.

Flow control to design a specific Services



**SignalR Hub Protocol**

The signalR Protocol is a protocol for two-way RPC over any Message-based transport. Either party in the connection may invoke procedures on the other party, and procedures can return zero or more result or an error.

Caller – The node that is issuing an Invocation, StreamInvocation, CancalInvocation, Ping messges and receiving Completion, StreamItem and Ping message

Callee – The node that is receiving an Invocation, StreamInvocation, CancelInvocation, Ping messages and issuing Completion, StreamItem and Ping messages( a node can be both Callee and Caceler for different invocation simultaneously)