# **SAMAS 2.0 Tech Stack Overview**

## **Backend Tech Stack**

### **.Net Core Web API**

* Exploring .Net Core vs .Net Framework: —
* .Net Core Web API Learning:  
  Official Docs:<https://learn.microsoft.com/en-us/aspnet/core/fundamentals/apis?view=aspnetcore-8.0>  
  Beginner Series:  
  [Back-end Web Development with .NET for Beginners - YouTube](https://www.youtube.com/playlist?list=PLdo4fOcmZ0oWunQnm3WnZxJrseIw2zSAk)
* Domain Driven Design (Clean Architecture):  
  Reference Store:<https://gitlab.com/arjun.khunt/cleanarchitecture>
* PostgreSQL Integration in Web API  
  Official Docs for EF Core:<https://learn.microsoft.com/en-us/ef/core/>  
  Code reference:<https://gitlab.com/arjun.khunt/as-monolith-sample/-/tree/main?ref_type=heads>
* CQRS (Command Query ResponsibilitySegregation):  
  Conceptual references:  
  [Applying CQRS and CQS approaches in a DDD microservice in eShopOnContainers - .NET | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/eshoponcontainers-cqrs-ddd-microservice)  
  [Implementing reads/queries in a CQRS microservice - .NET | Microsoft Learn](https://learn.microsoft.com/en-us/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/cqrs-microservice-reads)
* CORS (Cross Origin Resource Sharing)  
  Official Docs: [Enable Cross-Origin Requests (CORS) in ASP.NET Core | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/security/cors?view=aspnetcore-8.0)  
  Security & Identity: [ASP.NET Core security topics | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/security/?view=aspnetcore-8.0)
* Routing in .Net Core Web API  
  Official Docs: [Routing in ASP.NET Core | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/routing?view=aspnetcore-8.0)
* Authorization in Web API  
  Official Docs: [Introduction to authorization in ASP.NET Core | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/security/authorization/introduction?view=aspnetcore-8.0)  
  Types: Simple, Role-based, Policy-based, Claims-based
* Middleware Integration:  
  Official Docs: [ASP.NET Core Middleware | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/?view=aspnetcore-8.0)  
  Write our custom middleware: [Write custom ASP.NET Core middleware | Microsoft Learn](https://learn.microsoft.com/en-us/aspnet/core/fundamentals/middleware/write?view=aspnetcore-8.0)
* Mediator Design Pattern and usage of MediatR with CQRS

Introductory blog: <https://code-maze.com/cqrs-mediatr-in-aspnet-core/>

Sample Project reference ardalis clean architecture template : <https://github.com/ardalis/CleanArchitecture/tree/main/sample>

* Bootstrapping a Web API

## **Communication Framework**

* gRPC
* RabbitMQ
* Apache Kafka
* Web Socket
* SignalR

Publishing commands and events for async communication and loose temporal coupling

RabbitMQ:<https://docs.google.com/document/d/19eYLh2G7epbMZC27sCJf4p2PRUfEJdZV/edit?usp=sharing&ouid=107502776202862140039&rtpof=true&sd=true>

Kafka:<https://docs.google.com/document/d/1qWCyCNxGCVE13_NY-zu_kVSbTu_G7oqr/edit?usp=sharing&ouid=107502776202862140039&rtpof=true&sd=true>

Using dapr: official docs:<https://docs.dapr.io/developing-applications/building-blocks/pubsub/pubsub-overview/>

gRPC:

Service to service communication, synchronous communication

<https://docs.google.com/document/d/1ZXkJwAWIdm1ESDRe1L0gNWZcYdfk_YmZ/edit?usp=sharing&ouid=107502776202862140039&rtpof=true&sd=true>

Official Docs:<https://learn.microsoft.com/en-us/aspnet/core/grpc/?view=aspnetcore-8.0>

SignalR:

real time communication  
Signalr uses websockets, server-sent events (SSE), and long-polling in that order, based on client device and browser support.

SignalR official docs:<https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-8.0>  
WebSocket: refer official docs

## **PostgreSQL and MySQL**

* Comparison of PostgreSQL vs MySQL in terms of performance for Read and Write
* Complexity in integration with .Net Core Web API
* MongoDB database for unstructured requirements

Note: Create POC Web APIs that have integrations with PostgreSQL and MySQL both and demonstrate the communication between them via all different communication frameworks.

PostgreSQL, MySQL, and MongoDB all have EF Core providers available.  
PostgreSQL, MySQL have been available for long time; while MongoDB just became GA.  
EF team has kept escape hatch mechanism for MongoDB client directly if EF doesn’t support the required feature.

4) xUnit Testing Framework: Comparison report:<https://docs.google.com/document/d/1UGYWb_0GJj7D4eeqyuR3JM3tlEj_0DlI/edit?usp=sharing&ouid=107502776202862140039&rtpof=true&sd=true>

Code reference:<https://github.com/dotnet-architecture/eShopOnWeb/tree/main/tests/UnitTests>

Front-end

1) Angular or React

SPA, Page Navigations, opening one SPA from another, using Services for Backend interactions, complex control usage in Angular/React -> e.g. Tree View, Data Grid, Expander etc.

2) Front end Testing framework -> Jasmin + Karma

[Testing • Overview • Angular](https://angular.dev/guide/testing)

[Introduction – Testing Angular](https://testing-angular.com/introduction/#introduction)

Logging Framework: ELK Stack and its usage

Monitoring Framework: Prometheus or similar tool

<https://medium.com/c-sharp-progarmming/net-core-microservice-metrics-with-prometheus-and-grafana-d228bea89283>

API Gateway & Load Balancer: —

Additional Study items

.Net Core :

6) Docker based deployment

[Docker overview](https://docs.docker.com/guides/docker-overview/)

7) Kubernetes for container orchestration and horizontal scaling

[Installing kubeadm | Kubernetes](https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/)

[HorizontalPodAutoscaler Walkthrough | Kubernetes](https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/)

8) multi-tenant architecture  
Broad concept with many design, deployment and operational considerations  
Refer web resources for additional references  
Library with basic helper features: [MultiTenant Docs (finbuckle.com)](https://www.finbuckle.com/MultiTenant/Docs/GettingStarted)

VS Code for Web API and Angular

VS Code for Web API:  
[C# and .NET Development in VS Code for Beginners - YouTube](https://www.youtube.com/playlist?list=PLdo4fOcmZ0oWUrumb503vY3v6O3u3P4rW)  
VS Code for Angular:

[Angular TypeScript Tutorial in Visual Studio Code](https://code.visualstudio.com/docs/nodejs/angular-tutorial)

Go through 'KeyCloak' for Login  
Official Docs: [Keycloak](https://www.keycloak.org/)