Table of Contents

[Introduction 2](#_Toc187523810)

[Goals and Objectives 2](#_Toc187523811)

[Technology Stack 2](#_Toc187523812)

[Architecture Overview 2](#_Toc187523813)

[Components 3](#_Toc187523814)

[Host/Container Application 3](#_Toc187523815)

[Insurance Details Micro Frontend 3](#_Toc187523816)

[Premium Calculator Micro Frontend 4](#_Toc187523817)

[Cross Cutting Use Cases 4](#_Toc187523818)

[Web Worker 4](#_Toc187523819)

[Bundler 5](#_Toc187523820)

[CSS Pre-Processor 5](#_Toc187523821)

[MFE Cross Communication 5](#_Toc187523822)

[OWASP Feature 5](#_Toc187523823)

[Live Application 5](#_Toc187523824)

# Introduction

This document outlines the high-level design for a Micro Frontend Architecture for Insurance portal intended to develop, deploy, and maintain scalable, maintainable, and independently deployable web applications. Micro Frontends follow the same principles as microservices, where the frontend is broken down into small, independently deliverable pieces.

# Goals and Objectives

An application for an insurance company needs to be developed where users can check their insurance details and pay premiums.

Adopting MFE architecture provides clear advantages for the insurance application by:

1. Independence 🡪 Allow teams to work independently on different frontend components with their own technology stack.
2. Scalability 🡪 Enable scaling for different parts of the application based on user demand.
3. Maintainability 🡪 Enhance maintainability by allowing for modular codebases.
4. User Experience 🡪 Ensure a seamless and cohesive user experience across different modules.

# Technology Stack

|  |  |
| --- | --- |
| **Facet** | **Used** |
| Framework | .NET Core 8 |
| UI (Web) | React JS |
| Module Bundler | Webpack 5 |
| Cache | Redis Cache on Azure |
| Global Load Balancer | Azure Front Door & CDN Profiles |
| API Gateway | Azure APIM |
| CDN | Azure Front Door & CDN Profiles |
| Web hostname/name resolution | Azure DNS |
| Storage (Static content with CSS/HTML/JS) | Azure Storage Account |
| Image Repository | Azure Container Registry (ACR) |
| Orchestration and Containerization | Azure Kubernetes Service (AKS) |

# Architecture Overview

Insurance MFE application architecture consists of a host/container application that loads and renders multiple micro frontend modules. Each micro frontend represents a distinct feature set of the insurance application and self-contained developed by a different team or for a specific feature set.

The diagram below provides an overview on which the assignment is based.A diagram of a application entry point

Description automatically generated

# Components

### Host/Container Application

It acts as an entry point to the application. It dynamically loads and displays below micro frontends.

* insurance-details
* premium-calculator

Hosting details of insurance-container:

|  |  |
| --- | --- |
| Application Name | insurance-container |
| Local Port | 8003 |
| Live App URL | <https://mfeapp.z13.web.core.windows.net/insurance-container/index.html> |

### Insurance Details Micro Frontend

It shows the details of the individual and family details who are insurance covered along with premium. It covers below features:

* Default sum insured value used is 300000.
* User details are static in nature saved in local storage when application load.
* User can select the amount of insurance premium and based on it the premium will be calculated.
* When a user submit the updated premium then its monthly and yearly premium updated in premium-calculator MFE.
* Communication between insurance-detail MFE and premium-calculator MFE is implemented using Webpack module federation where:
  + InsuranceDetails component from insurance-details MFE is exposed through module federation.
  + PremiumDetails component from premium-calculator MFE consume InsuranceDetails component from insurance-details MFE through a remote entry with the help of module federation functionality.

|  |  |
| --- | --- |
| Application Name | insurance-details |
| Local Port | 8006 |
| Live App URL | <https://mfeapp.z13.web.core.windows.net/insurance-detail/index.html> |

### Premium Calculator Micro Frontend

It shows the details of the monthly and yearly premium for the sum insured by an individual. It covers below features:

* Using webpack federation module, it consumes components/function exposed by insurance-detail MFE.
* When the sum insured changes then a listener is added to insurance-detail MFE and that essentially consume by premium-calculator MFE to calculate monthly and yearly premium.
* The default sum insured value used is 300000 so that it loads individually.

|  |  |
| --- | --- |
| Application Name | premium-calculator |
| Local Port | 8005 |
| Live App URL | <https://mfeapp.z13.web.core.windows.net/premium-calculator/index.html> |

# Cross Cutting Use Cases

## Web Worker

1. It is added to the *premium-calculator* MFE to calculate the yearly and monthly premium based on the sum insured.
2. For demo purpose, 5% of the sum insured is chosen as the premium value of the insurance.
3. Since all MFE’s are developed as a self-contained application hosted with different domain and port. To enable cross-origin access to worker from container application, a module *crossoriginworker* is used in *premium-calculator* MFE.

## Bundler

1. Webpack 5 is used to bundle the MFE application.
2. Webpack module federation, a functionality of Webpack 5, is used to expose the MFE’s to the container application.

## CSS Pre-Processor

Saas (Syntactically Awesome Stylesheets) is used to compile the scss files to css.

## MFE Cross Communication

1. A function *subscribeToSumInsuredUpdates* is exposed from component of insurance-detail MFE through webpack federation module.
2. The above function is subscribed in *premium-calculator* MFE and trigger web worker to re-calculate premium when the insured sum changes.

## OWASP Feature

Content security policies are used in container application – *insurance-container* in *index.html* to load modules from MFE’s only.

# Live Application

All applications – MFE’s and Container are deployed to Azure storage account (static website enabled). The below are the details:

|  |  |
| --- | --- |
| **Application** | **Public URL** |
| Container App (Main Application) | <https://mfeapp.z13.web.core.windows.net/insurance-container/index.html> |
| Insurance Detail App (MFE#1) | <https://mfeapp.z13.web.core.windows.net/insurance-detail/index.html> |
| Premium Calculator App (MFE#2) | <https://mfeapp.z13.web.core.windows.net/premium-calculator/index.html> |