**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Notes** |
| 0.1 | 2024-05-17 | Lawkant Kumar | Initial draft |

Table of Contents

[System Architecture and Design Document 1](#_Toc167092201)

[1. Introduction 1](#_Toc167092202)

[1.1 Purpose 2](#_Toc167092203)

[1.2 Scope 2](#_Toc167092204)

[2. System Overview 2](#_Toc167092205)

[3. Architectural Design 2](#_Toc167092206)

[3.1 High-Level Architecture 2](#_Toc167092207)

[3.2 Components and Responsibilities 2](#_Toc167092208)

[4. Technical Stack 3](#_Toc167092209)

[5. Infrastructure Design 3](#_Toc167092210)

[7. Security Design 3](#_Toc167092211)

[8. Performance and Scalability 3](#_Toc167092212)

[12. Maintenance and Support 3](#_Toc167092213)

[15. Technical diagram. 4](#_Toc167092214)

# System Architecture and Design Document

## 1. Introduction

This document serves as a comprehensive guide outlining the architecture and design of the Messaging System. It provides a detailed overview of the system's structure, components, and interactions.

### 1.1 Purpose

The purpose of this document is to:

* Define the system architecture.
* Describe the design decisions and rationale.
* Serve as a reference for developers, designers, and stakeholders.

### 1.2 Scope

The scope of this document covers the architecture and design of the Messaging System, including its core functionalities, infrastructure, and data management.

## 2. System Overview

The Messaging System is an on-prem (without internet access) system that allows dependent components (example TCP clients) to interact with Central Component (example TCP server) and exchange messages among them. This includes additional functionality like logging and storage provider.

## 3. Architectural Design

### 3.1 High-Level Architecture

The system follows a three-tier architecture:

* **Presentation Layer**: A console interface for users. But this can be setup/configured to be used as a service without presentation layer.
* **Application Layer**: Business logic and application component to process the message.
* **Data Layer**: Option to plug in component to save the message in any data source.

### 3.2 Components and Responsibilities

* **Presentation Layer**:
  + Console interface: Provides an intuitive interface for users to start/stop the Client/Server.
* **Application Layer**:
  + Provides functionality to interact with central component (TCP Server).
  + Provides functionality to interact with dependent component (TCP Client).
  + Helps to store the data.
  + Helps to log the information.
  + Provides functionality to switch on/off the server.
* **Data Layer**:
  + Component to store the data in database.

## 4. Technical Stack

* **Programming Languages**: C# 8
* **Frameworks/Libraries**: Dot Net Express.js (Backend), React.js (Frontend)
* **Database**: Configurable and any database can be plugged-In

## 5. Infrastructure Design

The platform is deployed on a single system utilizing the TCP to communicate. We can plug in any data store like file system, SQL etc. to store the data.

## 7. Security Design

* **Authentication**: Users authenticate using email/password or OAuth2.
* **Authorization**: Role-based access control restricts access to sensitive functionalities.
* **Data Encryption**: Sensitive data is encrypted at rest and during transmission.
* **OWASP Top 10**: Mitigation strategies are implemented to address common security vulnerabilities.

## 8. Performance and Scalability

To enhance the performance TPL library has been used which helps to enhance the performance in threaded environment. Also, in case of increased traffic it will help scall to use other core available in the system to handle it.

## 12. Maintenance and Support

Loging is in place to detect and address system issues proactively.

## 15. Technical diagram.

This sample provides a structured overview of the system architecture and design, including key components, technologies used, and design considerations. Actual documents may vary in depth and detail based on project requirements and organizational standards.

A diagram of a computer component

Description automatically generated