**Self-Partner Onboarding Portal - Technical Design Document**

Contents

[1. Introduction 3](#_Toc197702817)

[2. System Architecture 4](#_Toc197702818)

[3. Sequence Diagrams 5](#_Toc197702819)

[4. Technology Stack 6](#_Toc197702820)

[5. Database Schema 7](#_Toc197702821)

[6. Core Functionality 7](#_Toc197702822)

[7. Technical Components and Constructs 9](#_Toc197702823)

[8. Implementation Plan 11](#_Toc197702824)

[9. Summary 12](#_Toc197702825)

**Document Revision History:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Reviewer | Description of Changes |
| 1 | 9-May-25 | Sasank | Raghuram |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 1. Introduction

**1.1 Purpose**

This technical design document outlines the architecture, workflow, and technical specifications for implementing the Self Partner Onboarding Portal. The portal is designed to streamline the onboarding process for both B2B EDI Partners (through Anypoint Partner Manager) and Generic Partners, providing a user-friendly interface for partners to submit their information and configure their integration requirements.

**1.2 Scope**

The portal will provide:

* Partner self-registration capabilities
* Document upload functionality for security certificates
* Interface configuration for various communication protocols
* Approval workflow for partner registration
* Dashboard for administrators to manage partner onboarding requests
* Document repository for secure storage of certificates and credentials

# 2. System Architecture

**2.1 High-Level Architecture Diagram**

A diagram of a company

AI-generated content may be incorrect.

**2.2 Component Description**

**2.2.1 Client Tier**

* **Partner Portal**: Web interface for partners to register, upload documents, and configure integration settings
* **Admin Portal**: Web interface for administrators to view and manage partner onboarding requests
* **Approval Portal**: Interface for authorized personnel to review and approve partner submissions

**2.2.2 API Tier**

* **Partner API**: Handles partner registration, document uploads, and interface configuration
* **Admin API**: Provides endpoints for dashboard data and administrative functions
* **Approval API**: Manages the approval workflow process

**2.2.3 Service Tier**

* **Partner Service**: Business logic for partner management
* **Admin Service**: Business logic for administrative operations
* **Approval Service**: Business logic for approval workflows

**2.2.4 Data Tier**

* **PostgreSQL Database**: Stores partner data, configurations, and approval status
* **Azure Key Vault**: Securely stores certificates, keys, and credentials
* **Azure AD**: Handles authentication and authorization

# 3. Sequence Diagrams

**A diagram of a partner registration and onboarding sequence diagram

AI-generated content may be incorrect.**

# 4. Technology Stack

**4.1 Frontend**

* **Framework**: React.js with TypeScript
* **UI Components**: Shad cn
* **State Management**: Redux or Context API
* **Form Handling**: Formik with Yup validation
* **HTTP Client**: Axios

**4.2 Backend**

* **API Framework**: FastAPI (Python)
* **Business Logic**: Python 3.9+
* **Authentication**: Azure AD integration with JWT tokens
* **Database ORM**: SQLAlchemy
* **API Documentation**: Swagger/OpenAPI

**4.3 Database**

* **Primary Database**: PostgreSQL 14+
* **Schema Management**: Alembic migrations
* **Connection Pooling**: PgBouncer

**4.4 Security and Infrastructure**

* **Secret Management**: Azure Key Vault
* **Identity Provider**: Azure AD
* **Containerization**: Docker
* **Orchestration**: Kubernetes or Azure App Service
* **CI/CD**: Azure DevOps or GitHub Actions
* **Monitoring**: Azure Application Insights
* **Logging**: ELK Stack or Azure Log Analytics

# 5. Database Schema

**5.1 Core Tables**

****

# 6. Core Functionality

**6.1 User Authentication**

1. Users access the portal through a web browser
2. Authentication is handled via Azure AD SSO
3. Two primary roles are defined:
   * Partner: External partners who register and provide configuration details
   * Admin: Internal users who approve partner registrations and manage configurations

**6.2 Partner Onboarding Process**

**6.2.1 Anypoint Partner Manager (B2B EDI Partners) Flow**

1. Partner logs in using Azure AD credentials
2. Partner selects "New Partner Onboarding" from the navigation menu
3. Partner chooses "Anypoint Partner Manager Partner" type
4. Partner completes the 4-step process:
   * Step 1: Enter company and contact information
   * Step 2: Upload security certificates
   * Step 3: Configure interface settings (protocols, authentication, etc.)
   * Step 4: Review and submit for approval
5. System creates an approval request in the database
6. Admin receives notification of new partner registration
7. Admin reviews the submission from the Dashboard or My Approvals section
8. Admin approves or rejects the submission with comments
9. Partner receives notification of approval status
10. If approved, the system provisions the partner configuration in backend systems

**6.2.2 Generic Partner Flow**

The Generic Partner flow follows a similar pattern but with different information collected:

1. Welcome and Introduction
2. Training and Education materials
3. Tools and Resources Setup
4. Goal Setting and Performance Metrics
5. Legal and Compliance Briefing
6. Support and Communication Channels Establishment

**6.3 Security Implementation**

1. **Authentication**: Azure AD SSO enforces identity verification and access control
2. **Authorization**: Role-based access control limits functionality based on user roles
3. **Certificate Management**: Partner certificates are securely stored in Azure Key Vault
4. **Credential Security**: Sensitive connection credentials are encrypted and stored in Azure Key Vault
5. **Data Protection**: All data transmissions use TLS encryption
6. **Audit Trail**: System logs all activities for compliance and troubleshooting

**6.4 Admin Dashboard**

1. Provides overview of partner onboarding status
2. Displays metrics on pending approvals, recent submissions, and completion rates
3. Offers filtering and sorting capabilities for efficient management
4. Enables drill-down into individual partner details
5. Provides approval workflow functionality directly from the dashboard

# 7. Technical Components and Constructs

**7.1 Frontend Constructs**

1. **Component Architecture**
   * Reusable UI components for forms, tables, and cards
   * HOCs for authentication and authorization
   * Custom hooks for common functionality
2. **Routing**
   * Protected routes for authenticated users
   * Role-based route restrictions
   * Dynamic navigation based on user permissions
3. **State Management**
   * Global application state for user session
   * Form state management for multi-step processes
   * Optimistic UI updates for better user experience
4. **Form Handling**
   * Multi-step form wizard with state persistence
   * Dynamic form fields based on selected options
   * Validation at field, page, and submission levels

**7.2 Backend Constructs**

1. **API Design**
   * RESTful endpoints for CRUD operations
   * Request validation using Pydantic models
   * Pagination, filtering, and sorting support
   * Comprehensive error handling
2. **Service Layer**
   * Business logic separation from API controllers
   * Transaction management for database operations
   * Event-driven architecture for notifications
3. **Data Access Layer**
   * Repository pattern for database operations
   * Optimized query patterns
   * Connection pooling for performance
4. **Security Middleware**
   * JWT token validation
   * Role-based permission checks
   * Rate limiting for API abuse prevention
   * Input sanitization

**7.3 Infrastructure Constructs**

1. **Containerization**
   * Dockerfiles for service packaging
   * Docker Compose for local development
   * Container orchestration for production
2. **CI/CD Pipeline**
   * Automated testing (unit, integration, e2e)
   * Static code analysis
   * Infrastructure as Code (IaC) for environment provisioning
   * Blue/green deployment strategy
3. **Monitoring and Observability**
   * Application logs with structured logging
   * Performance metrics
   * Distributed tracing
   * Alerting rules

# 8. Implementation Plan

**8.1 Phase 1: Core Infrastructure Setup**

* Set up Azure AD integration
* Create PostgreSQL database schema
* Configure Azure Key Vault
* Set up CI/CD pipeline

**8.2 Phase 2: Backend Development**

* Develop core API endpoints
* Implement business logic services
* Create database access layer
* Set up security controls

**8.3 Phase 3: Frontend Development**

* Build authentication flow
* Develop partner onboarding wizard
* Create admin dashboard
* Implement approval workflows

**8.4 Phase 4: Testing and QA**

* Unit testing
* Integration testing
* User acceptance testing
* Security testing

**8.5 Phase 5: Deployment and Launch**

* Production environment setup
* Data migration (if needed)
* User training
* Go-live support

# 9. Summary

The Self Partner Onboarding Portal is designed as a comprehensive solution for automating and streamlining the partner onboarding process. Built on a modern technology stack with Python and PostgreSQL at its core, the system provides a secure, user-friendly interface for partners to submit their information and configure their integration requirements.

The architecture follows a multi-tier approach, separating concerns between presentation, business logic, and data access layers. Security is maintained through Azure AD integration, role-based access control, and secure storage of sensitive information in Azure Key Vault.

The implementation plan outlines a phased approach to development, ensuring that core functionality is built first, followed by additional features and thorough testing before deployment.