System Architecture Document: Healthcare EDI Integration System

1. Overview

System Name: Healthcare EDI Integration System

Objective:

- Enable seamless and secure exchange of healthcare-related EDI transactions (e.g., 837, 835, 270/271, 834) between healthcare providers, payers, and other stakeholders.
- Ensure compliance with HIPAA, SNIP validation levels, and secure data exchange protocols.

Scope:

- Support real-time and batch processing of EDI transactions.
- Provide APIs for healthcare providers and payers to interact with the system.
- Integrate EDI translation tools (e.g., ITX/ITXA) and secure file transfer protocols (e.g., AS2, SFTP).

2. Key Components

2.1 API Gateway

Purpose:

- Act as the entry point for external systems to interact with the EDI system.
- Secure APIs with OAuth2 and TLS/SSL encryption.

Responsibilities:

- Route API requests to the appropriate backend services.
- Perform authentication and authorization for API consumers.
- Throttle and monitor API usage.

APIs:

1. **Claim Submission API**:
    - **Endpoint**: `POST /api/claims`
    - **Purpose**: Accepts healthcare claims in JSON format and converts them to X12 837 for transmission to payers.
    - **Transaction Type**: 837.
2. **Eligibility Inquiry API**:
    - **Endpoint**: `POST /api/eligibility`
    - **Purpose**: Handles eligibility inquiries and converts them to X12 270 format; returns responses in JSON based on X12 271.
    - **Transaction Type**: 270/271.

3. **Benefit Enrollment API**:
    - **Endpoint**: `POST /api/enrollment`
    - **Purpose**: Accepts enrollment details, validates the data, converts it to X12 834, and transmits it to payers.
    - **Transaction Type**: 834.
4. **Premium Payment API**:
    - **Endpoint**: `POST /api/premiums`
    - **Purpose**: Handles premium payment submissions and converts them to X12 820 format for secure transmission to payers.
    - **Transaction Type**: 820.
5. **Personal Health Record Transfer API**:
    - **Endpoint**: `POST /api/phr-transfer`
    - **Purpose**: Manages requests for personal health record transfers, validates data, converts it to X12 275 format, and sends it securely.
    - **Transaction Type**: 275.
6. **Fee Schedule API**:
    - **Endpoint**: `POST /api/fees`
    - **Purpose**: Submits healthcare fee schedules, validates the data, converts it to X12 832 format, and transmits it to providers.
    - **Transaction Type**: 832.

7. **Claim Status Inquiry API**:

    - **Endpoint**: `POST /api/claim-status`
    - **Purpose**: Accepts claim status inquiries, converts them to X12 276, and returns the response based on X12 277.
    - **Transaction Type**: 276/277.

2.2 EDI Translation Layer

Tool: ITX/ITXA (IBM Transformation Extender).

Purpose:

- Translate internal formats (e.g., JSON, XML) to EDI X12 formats (e.g., 837, 835).
- Map EDI X12 formats back to internal formats for internal system consumption.

Responsibilities:

- Ensure EDI compliance with HIPAA implementation guidelines.
- Perform SNIP validation levels 1–7.
- Provide reusable mapping templates for different transaction types.

2.3 ESB (Enterprise Service Bus)

Tool: MuleSoft or Apache Camel.

Purpose:

- Orchestrate workflows between system components.
- Route and transform data between APIs, databases, and external systems.

Responsibilities:

- Manage message routing (e.g., API to EDI translator).
- Handle asynchronous messaging for batch file processing.
- Provide error handling and retry mechanisms.

2.4 Secure Communication Layer

Protocols: AS2, SFTP, HTTPS.

Purpose:

- Facilitate secure data exchange between the EDI system and external trading partners.
- Ensure compliance with HIPAA security requirements.

Responsibilities:

- Encrypt data in transit using TLS/SSL.
- Provide audit logs for all file transfers.
- Handle acknowledgments (e.g., 997, 999) from trading partners.

2.5 Database Layer

Tool: PostgreSQL.

Purpose:

- Store transactional data, logs, and system configurations.
- Ensure data integrity and traceability.

Responsibilities:

- Log EDI transactions and API requests.
- Maintain mappings and configurations for trading partners.
- Store acknowledgment statuses and error reports.

3. High-Level Architecture Diagram

Components:

1. **API Gateway**: Handles incoming requests and routes them to appropriate backend services.
2. **EDI Translator**: Maps and transforms data between internal and EDI X12 formats.
3. **ESB**: Orchestrates workflows and manages message routing.
4. **Secure Communication Layer**: Facilitates secure file transfers with trading partners.
5. **Database**: Stores transaction logs, mappings, and configurations.

5. Security Considerations

5.1 Data in Transit

- Encrypt all API communications using TLS/SSL.
- Use AS2 for secure file transfers with trading partners.
- Implement OAuth2 for API authentication.

5.2 Data at Rest

- Encrypt sensitive data stored in the database (e.g., PHI).
- Maintain access controls to restrict database access to authorized users.

5.3 Audit and Monitoring

- Log all API requests and EDI transactions.
- Monitor system performance and detect anomalies using tools like CloudWatch or Grafana.

6. Key Deliverables

1. **High-Level Architecture Diagram**: Visual representation of system components and workflows.
2. **Component Documentation**: Detailed descriptions of API Gateway, EDI Translator, ESB, Secure Communication Layer, and Database.
3. **Security Policies**: Guidelines for ensuring data security and HIPAA compliance.
4. **Workflow Specifications**: Detailed steps for each supported transaction type (e.g., 837, 835, 270/271).