HIPAA EDI File Processor — Functional Design & AI Roadmap

Prepared by: Himanshu Sharma

Date: [Insert Date]

# Table of Contents

This will be auto-generated in Word (References → Table of Contents).

# 1. Executive Summary

This document outlines the design of a HIPAA EDI File Processor that automates the handling, validation, and reporting of EDI 834 enrollment files (and other supported formats). The system leverages configuration-driven rules, robust error handling, and modular design to ensure compliance, scalability, and readiness for future AI/ML-based insights.

# 2. Objectives

- Automate HIPAA file validation and reporting across multiple states.  
- Ensure frequency-based processing (Daily/Monthly).  
- Maintain compliance with HIPAA standards.  
- Prepare a foundation for AI/ML integration.

# 3. Scope

## In Scope:

- Processing .EDI, .P, .TXT (configurable) files.  
- API-based HIPAA validation.  
- Report generation in Excel format.  
- Archival of processed files.  
- Frequency control based on config file.

## Out of Scope:

- Manual validation processes.  
- UI/dashboard in Phase 1 (considered in Phase 2 AI/ML).

# 4. Key Functional Requirements

## 4.1 Configuration-driven Processing

A config.json defines states, directories, and frequency.  
Example:

{  
 "States": [  
 { "Name": "California", "Path": "\\\\NAS\\CA", "Frequency": "Daily" },  
 { "Name": "Texas", "Path": "\\\\NAS\\TX", "Frequency": "Monthly" }  
 ]  
}

## 4.2 File Discovery & Local Copy

Scan directories for supported file extensions. Copy files to a local working folder before validation.

## 4.3 HIPAA Validation

Files are submitted to the existing HIPAA Validation API. The response is converted into an Excel report with:  
- File name  
- Timestamp  
- Validation results

## 4.4 Archiving

Move processed files to Archive/yyyy-MM-dd/ inside the source directory.

## 4.5 Frequency Handling

Daily → Process every run.  
Monthly → Process on first run of the month.

## 4.6 Error Handling & Logging

Log skipped files and API errors to a Logs directory.

# 5. High-Level Architecture

The following diagram explains the architecture flow:

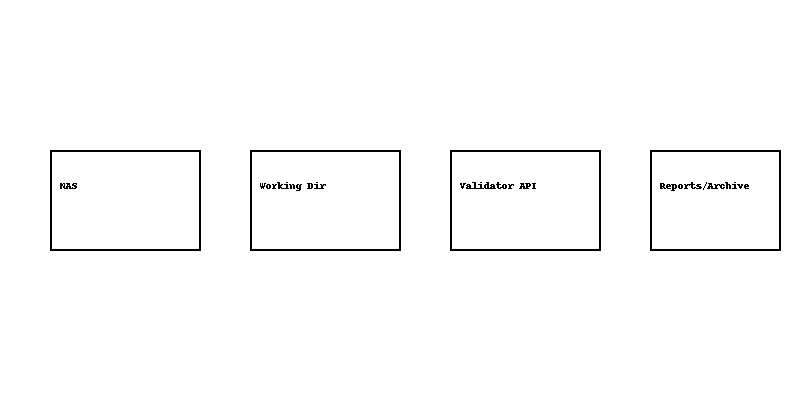


Figure 1: Architecture overview — shows flow from NAS → Working → Validator → Report → Archive and ML sidecar.

# 6. Future AI/ML Enhancements (Phase 2)

- Anomaly Detection: Identify unusual claim/error patterns.  
- Predictive Analytics: Estimate claim rejection probability.  
- Auto-classification: Group errors for faster resolution.  
- Dashboard Integration: Power BI or ML.NET-driven monitoring.

# 7. Benefits

- Fully automated HIPAA validation.  
- Configurable & scalable across states.  
- Ensures compliance & audit readiness.  
- Prepares the client for AI/ML-driven intelligence.

# 8. Assumptions & Constraints

- System runs as a .NET Framework 4.7.2 console app.  
- API endpoints for validation are already available.  
- Scheduler/Windows Task Scheduler will be used for execution.

# 9. Next Steps

1. Finalize configuration file (config.json).  
2. Deploy console app in test environment.  
3. Validate end-to-end with sample EDI files.  
4. Phase 2 planning for AI/ML integration.