

# SIWEA System Modernization: Visual Age to .NET 9 Migration Analysis

## 1. Project Overview

This document outlines the complete technical analysis and detailed planning for the modernization of the SIWEA (Claims Indemnity Payment Authorization System). The system, originally implemented in **IBM VisualAge EZEE 4.40**, is being migrated to a modern architecture based on **.NET 9, React 19, and Azure Cloud Platform**.

The goal is to replace a critical mainframe application (CICS, DB2, ESQL) that processes hundreds of payment authorization requests daily and integrates with three external systems (CNOUA, SIPUA, SIMDA).

## 2. Legacy System Analysis: IBM VisualAge EZEE 4.40

### 2.1 Architecture and Technologies

- **Development Platform:** IBM VisualAge EZEE 4.40 (last revised 2014, CAD73898).
- **Environment:** Mainframe with CICS, DB2, and ESQL.
- **Architecture Model:** Traditional 3-layer mainframe:
  - **Presentation Layer:** CICS Maps (SIWEG, SIWEGH) for 3270 terminal screens.
  - **Business Logic Layer:** ESQL (Extended SQL) for stored procedures and business rules.
  - **Data Layer:** IBM DB2 with 13 relational tables (e.g., TMESTSIN, THISTSIN).
- **Integration:** SOAP Web Services for external systems (CNOUA, SIPUA, SIMDA).
- **Transactions:** CICS Transaction Server for ACID control.
- **Authentication:** RACF / EZEUSRID for mainframe access control.

### 2.2 Key Functionalities

The SIWEA system provides 6 main functionalities for claims operators:

- **F1: Claim Search:** Search by protocol (3 parts), claim number, or leader code.
- **F2: Payment Authorization:** Creation of payment requests with specific types (1-5), validation, and beneficiary handling.
- **F3: Movement History:** Visualization of all previous authorizations.
- **F4: Consortium Validation:** CNOUA integration for product validation (6814, 7701, 7709).
- **F5: Phase Management:** Workflow control with phases (opening, analysis, approval, closing).
- **F6: Claims Dashboard:** Consolidated view of pending and authorized claims.

### 2.3 Business Rules (42 Identified, 15 Detailed)

The system implements 42 critical business rules in ESQL. Key rules include:

- **BR-001: Payment Type Validation:** TIPPAG must be 1-5.
- **BR-002: Beneficiary Obligation:** BENEF is mandatory if TPSEGU != 0.

- **BR-003: Standard Operation Code:** All authorizations use CODOPE = 1098.
- **BR-004: Monetary Correction Type:** TIPCOR is always '5'.
- **BR-005: Conversion to BTNF:** VALPRIBT = VALPRI × VLCRUZAD (from TGEUNIMO with date validation).
- **BR-006: Business Date Validation:** All operations use DTMOVABE from TSISTEMA, never system clock.
- **BR-007: Transaction Atomicity:** 3 atomic operations (INSERT THISTSIN, UPDATE TMESTSIN, UPDATE SI\_SINISTRO\_FASE). Full rollback on failure.
- **BR-008: Counter Increment:** OCORHIST in TMESTSIN increments by 1.
- **BR-009: Phase Management - Opening:** New phase recorded with DTENCFAS = 9999-12-31.
- **BR-010: Phase Management - Closing:** Phase updated with DTENCFAS = current DTMOVABE.
- **BR-011: CNOUA Validation for Consortium:** Products 6814, 7701, 7709 require SOAP call to CNOUA.
- **BR-012: SIPUA Validation for EFP:** If EF\_CONTR\_SEG\_HABIT record exists, validate contract via SIPUA.
- **BR-013: SIMDA Validation for HB:** Non-EFP products with HB indicator validate contract via SIMDA.
- **BR-014: Pending Value Calculation:** VALPEND = SDOPAG - TOTPAG.
- **BR-015: Operator Registration:** Log EZEUSRID in THISTSIN.USUCAD for audit.

## 2.4 Legacy Data Model (13 Entities)

The DB2 database contains 13 main interrelated tables. Core entities include:

- **TMESTSIN:** Claim Master (4-part composite key).
- **THISTSIN:** Claim History (5-part composite key).
- **TGERAMO:** Branch Master.
- **TGEUNIMO:** Currency Unit (rates by date range).
- **SISINISTROFASE:** Claim Phase tracking.

Additional entities: TSISTEMA, TAPOLICE, SI\_ACOMPANHA\_SINI, SI\_REL\_FASE\_EVENTO, EF\_CONTR\_SEG\_HABIT, MigrationStatus (new), ComponentMigrationTracking (new), PerformanceMetrics (new).

## 2.5 External System Integrations

- **CNOUA:** SOAP/HTTP for consortium product validation.
- **SIPUA:** SOAP/HTTP for EFP contract validation.
- **SIMDA:** SOAP/HTTP for HB contract validation.
- **Resilience Strategy:** Polly (Retry, Circuit Breaker, Timeout, Fallback, Serilog logging with correlation ID).

# 3. Target Architecture: .NET 9 + React 19 + Azure

The migration adopts a Clean Architecture pattern with clearly separated layers.

## 3.1 API Layer (Presentation)

- **Technology:** ASP.NET Core 9.0 Web API with REST controllers.
- **Legacy Integration:** SoapCore 1.1 for maintaining legacy SOAP contracts.
- **Security:** JWT Authentication + Active Directory integration.

- **Documentation:** Swagger/OpenAPI 3.0.

### 3.2 Core Layer (Domain)

- **Entities:** Claim, ClaimHistory, Payment.
- **Services:** IClaimService, IPaymentService, IValidationService.
- **Logic:** Framework-agnostic business logic in C# 12.
- **Validation:** FluentValidation.

### 3.3 Infrastructure Layer (Data)

- **ORM:** Entity Framework Core 9 with database-first approach.
- **Data Access:** Repository pattern.
- **External Integrations:** HttpClient with Polly for resilient external calls.
- **Logging:** Serilog for structured logging.

### 3.4 Frontend (React 19)

- **Type:** Single Page Application (SPA) with TypeScript.
- **Routing:** React Router DOM 7.
- **State Management:** React Query for server state.
- **Communication:** Axios for HTTP.
- **Styling:** Site.css preserved for visual consistency.

## 4. Migration Methodology: MIGRAI Framework

MIGRAI is a proprietary AI-assisted framework guiding the modernization process:

- **M - Modernization:** Complete migration to modern stack while preserving 100% business logic.
  - **Backend:** .NET 9, C# 12 (records, pattern matching, nullable reference types).
  - **Frontend:** React 19 (concurrent rendering, Server Components, hooks).
  - **Cloud:** Azure App Service (horizontal scaling), Azure SQL Database.
  - **Architecture:** Clean Architecture.
  - **Quality:** Reduced technical debt, automated tests.
- **I - Intelligence (AI):** Uses LLMs (Claude 3.5 Sonnet) for acceleration and validation.
  - **Code Generation:** ESQ to C# (95%+ accuracy).
  - **Test Generation:** Unit tests from Given-When-Then specs.
  - **Documentation:** Extraction of legacy comments.
  - **Code Review:** Automated compliance analysis.
  - **Knowledge Mining:** Identification of undocumented patterns/rules.
- **G - Gradual Migration:** Phased rollout with incremental deliveries.
  - **Prioritization:** User stories (P1-P6).
  - **Feature Toggles:** LaunchDarkly for controlled activation.
  - **Parallel Operation:** Minimum 2 weeks Visual Age + .NET side-by-side.
  - **Data Migration:** Incremental with continuous validation and rollback.
  - **Phase Gates:** Acceptance criteria for phase advancement.
- **R - Resilience:** Implementation of resilient patterns for high availability.
  - **Policies:** Polly (Retry, Circuit Breaker, Timeout, Fallbacks, Graceful Degradation).
  - **Transaction Rollback:** EF Core TransactionScope for ACID properties.
- **A - Automation:** Complete CI/CD pipeline with automated quality gates.
  - **Build:** GitHub Actions (.NET SDK 9.0, Node.js 18+).
  - **Tests:** xUnit (unit), TestServer (integration), Playwright (E2E).

- **Code Coverage:** 80% minimum.
- **Security Scan:** OWASP, CodeQL.
- **Deploy:** Azure App Service via Terraform.
- **Monitoring:** Application Insights.
- **I - Integration:** Perfect integration preserving existing contracts.
  - **SOAP Legacy:** SoapCore with exact namespaces.
  - **REST Modern:** OpenAPI 3.0 with /api/v1 versioning.
  - **Database:** Zero schema changes, EF Core Fluent API for mapping.
  - **Authentication:** Active Directory LDAP mapping EZEUSRID to UPN.
  - **External Services:** Preservation of CNOUA/SIPUA/SIMDA contracts.
  - **Error Codes:** Backward-compatible (EZERT8 -> HTTP 400).

## 5. Key Takeaways for Gemini

- **Project Goal:** Modernize SIWEA from IBM VisualAge EZEE 4.40 (mainframe, CICS, DB2, ESQL) to .NET 9, React 19, and Azure.
- **Legacy System:** Understand its 3-layer mainframe architecture, key functionalities, 42 business rules (especially the 15 detailed ones), 13 DB2 entities, and SOAP integrations.
- **Target System:** Clean Architecture, ASP.NET Core 9.0 Web API, React 19 SPA, EF Core 9, Azure.
- **Migration Strategy:** MIGRAI Framework (Modernization, Intelligence, Gradual, Resilience, Automation, Integration) is central. AI (LLMs) plays a significant role in code/test generation and knowledge mining.
- **Business Rules are Critical:** 100% of legacy business logic must be preserved and accurately migrated.
- **Testing is Extensive:** Unit, integration, E2E tests are crucial, with 80% code coverage minimum.
- **Resilience is Key:** Polly is used for external service integrations.
- **Deployment:** Azure via Terraform, CI/CD with GitHub Actions.
- **Key Documents:** docs/README\_ANALYSIS.md and this summary are primary sources of truth.