
Mainframe to Java Migration: Multi-Agent Architecture Documentation

Overview

This architecture models a modular, multi-agent system designed to automate the migration of enterprise mainframe applications (typically COBOL, JCL, VSAM, Copybooks) to modern Java applications. It leverages AI and traditional parsing technologies orchestrated by an AI Orchestrator (Google ADK or LangGraph) to enable scalable, maintainable, and transparent migration.

Layers and Components

1. Parsing Layer

Responsible for ingesting and syntactically analyzing raw mainframe artefacts. It converts code and metadata into structured, machine-readable formats for downstream analysis.

- **COBOL Parser (ANTLR Grammar)**
    - Parses COBOL source code using ANTLR-based grammar.
    - Extracts Abstract Syntax Tree (AST), program structure, variables, and control flow.
- **JCL Parser**
    - Parses Job Control Language scripts.
    - Identifies job steps, resources, and execution dependencies.
- **Copybook Parser**
    - Processes COBOL copybooks defining data structures.
    - Extracts detailed data layout and field metadata.
- **VSAM Schema Parser**
    - Parses VSAM dataset definitions.
    - Extracts data organization and access methods.

---

2. Analysis Layer

Performs deeper semantic analysis on parsed artefacts to extract business logic, dependencies, data mappings, and migration insights.

- **Dependency Mapper**
    - Builds call graphs and data flow maps.
    - Identifies code module dependencies, usage patterns, and sequence of execution.
- **Business Rules Extractor**
    - Analyzes ASTs and metadata using NLP and AI techniques.
    - Extracts explicit and implicit business rules embedded in code logic.
- **Data Model Mapping**
    - Maps legacy data structures to modern relational or NoSQL schemas.
    - Aligns VSAM and DB2 structures with Java persistence models.
- **Migration Report Generator**
    - Produces detailed reports capturing complexity, migration risks, dependencies, and estimated effort.
    - Provides actionable insights for planning and risk mitigation.

---

3. Transformation Layer

Converts analyzed artefacts and rules into deployable Java code and related artifacts, ready for testing and deployment.

- **Intermediate Representation (IR) Generator**
    - Converts parsed and analyzed data into an abstract intermediate format.
    - Serves as a neutral bridge between legacy code and target Java constructs.
- **Java Code Generator**
    - Translates IR into idiomatic Java code.
    - Applies best practices, design patterns, and integrates business rules.
- **Test Harness Generator**

    - Automatically generates unit and integration tests for migrated components.
    - Ensures behavior correctness and facilitates regression testing.
- **Deployment Package Builder**
    - Prepares deployable Java artifacts, including JARs, WARs, configuration files.
    - Supports integration into CI/CD pipelines.

AI Orchestrator Agent (Google ADK / LangGraph)

- **Role**
    - Coordinates all agents across layers.
    - Manages workflows, data exchange, error handling, and iterative improvements.
    - Leverages AI capabilities for intelligent decision-making, such as prioritizing complex modules, suggesting code optimizations, or identifying migration bottlenecks.
- **Benefits**
    - Enables scalable, modular, and maintainable migration processes.
    - Supports incremental migration and continuous validation.
    - Facilitates human-in-the-loop interventions where necessary.

Inputs & Outputs

- **Inputs**
    - Enterprise mainframe artefacts including COBOL source, JCL scripts, copybooks, VSAM definitions, DB2 DDL.
- **Outputs**
    - Fully migrated, maintainable Java application code.
    - Comprehensive migration reports outlining progress, risks, and recommendations.

Summary

This multi-agent architecture enables a **holistic, AI-driven migration** from mainframe to Java, minimizing manual effort and maximizing automation, transparency, and control. It modularizes complex tasks into specialized agents coordinated by a central orchestrator, paving the way for efficient enterprise modernization.