ASIMOV
App Modernization

Our Agentic Framework for Legacy Migration

What We Offer

Up to 70% **reduction in migration time**

Up to 60% **reduction in migration cost**

- **Highly maintainable** future ready code
- Zero **cloud dependency** on migrated code
- Source code, test cases, **documentation**
- Excellent code **quality**, **optimized** code

Note: The savings are benchmarked for 1 million lines of standalone code. The exact saving can be determined post the "Discovery & Analysis" Phase.

Our Approach Comparison

Method	Code Quality / Maintainability	Conversion Time	Conversion Costs	Maintenance Cost (TCO)
Manual re-write	😊	☹️	☹️	😊
Re-engineering & refactoring only	🙂	🙁	🙁	🙂
Cloud native migration	🙂	🙂	☹️	😐
Conventional automation tools	😐	😐	😐	😐
Mainframe emulation	🙁	🙂	🙂	🙁
Our Approach	😊	😊	😊	😊

ASIMOV’s Capability

1

Legacy to Modern Codebase

End-to-end transformation **of Legacy codebases into modern**, reliable stacks—

for example, COBOL to Java

Delphi to C#

2

Re-architecting Legacy

Re-architecting Legacy systems for modular scalable design –

Monolith to Microservice,

Tightly coupled to Modular architecture

3

Technology Version Upgrades

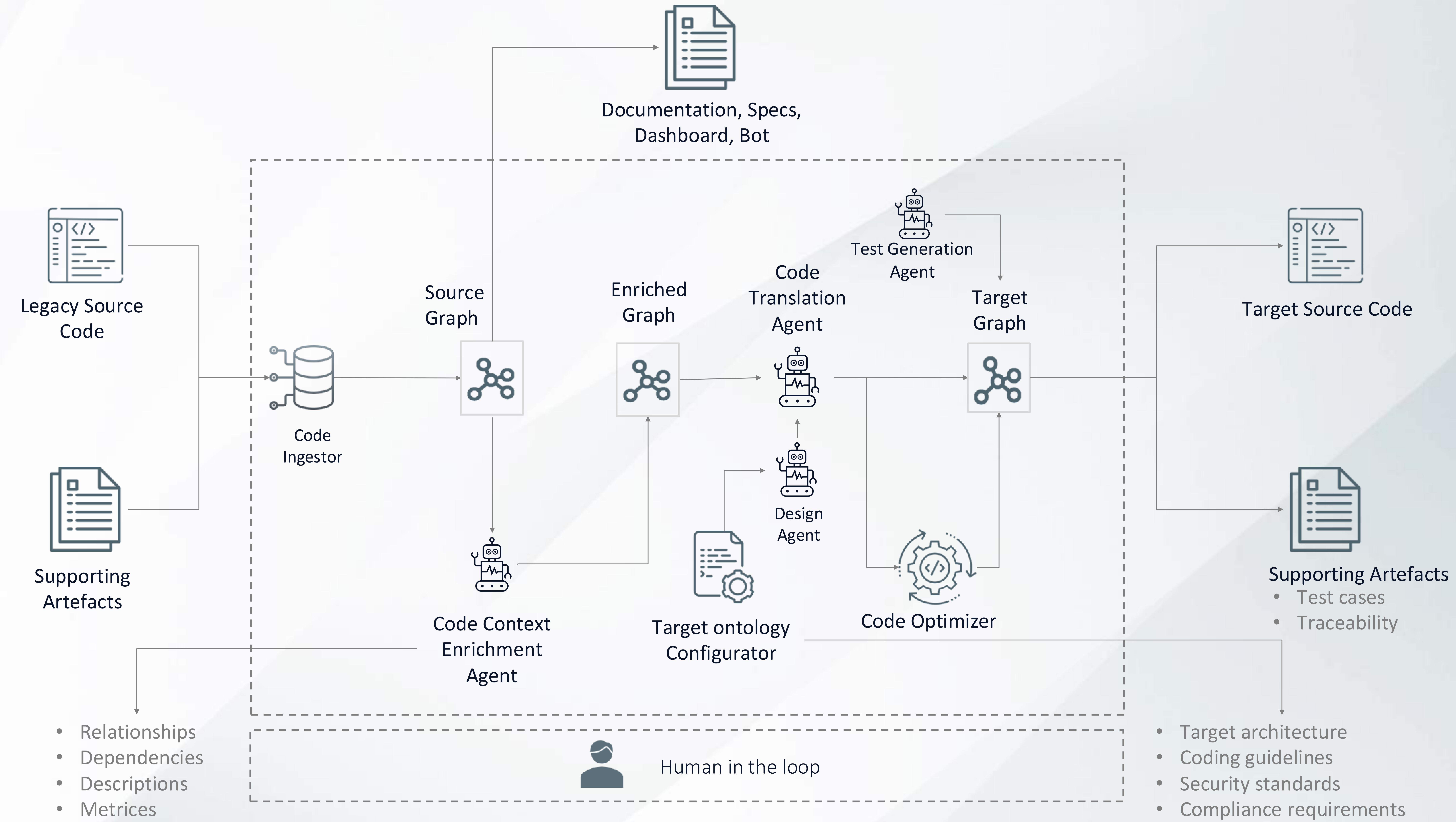
Version upgrades across technology stacks —

Angular 0 to Angular 16.

Java 8 to Java 21

In all scenarios the Approach support inclusion of Architectural upgrades based on target technology and well as enforcing best practices in the target system.

ASIMOV’s Agentic Framework



Documentation, Specs, Dashboard, Bot

Legacy Source Code

Supporting Artefacts

Code Ingestor

Source Graph

Enriched Graph

Code Translation Agent

Test Generation Agent

Target Graph

Design Agent

Code Context Enrichment Agent

Target ontology Configurator

Code Optimizer

Target Source Code

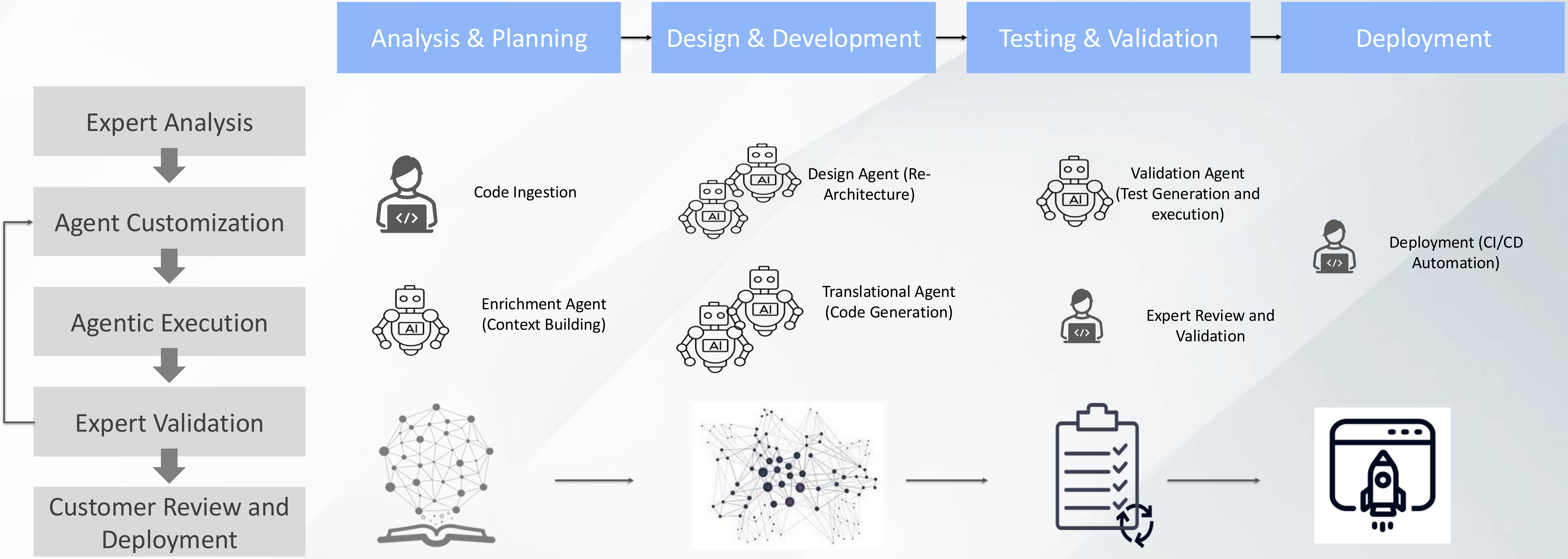
Supporting Artefacts
- Test cases
- Traceability

- Relationships
- Dependencies
- Descriptions
- Metrices

Human in the loop

- Target architecture
- Coding guidelines
- Security standards
- Compliance requirements

Migration & Modernization Approach & Accelerators



Expert Analysis

Agent Customization

Agentic Execution

Expert Validation

Customer Review and Deployment

Analysis & Planning

- Code Ingestion
- Enrichment Agent (Context Building)

Design & Development

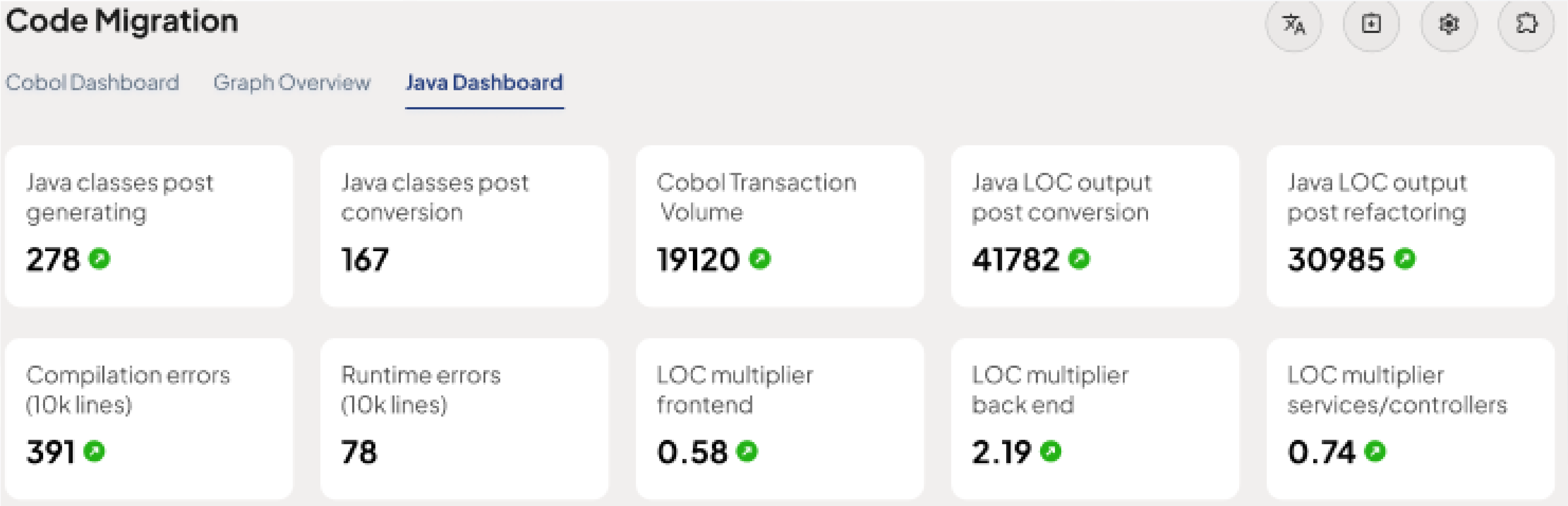
- Design Agent (Re-Architecture)
- Translational Agent (Code Generation)

Testing & Validation

- Validation Agent (Test Generation and execution)
- Expert Review and Validation

Deployment

- Deployment (CI/CD Automation)

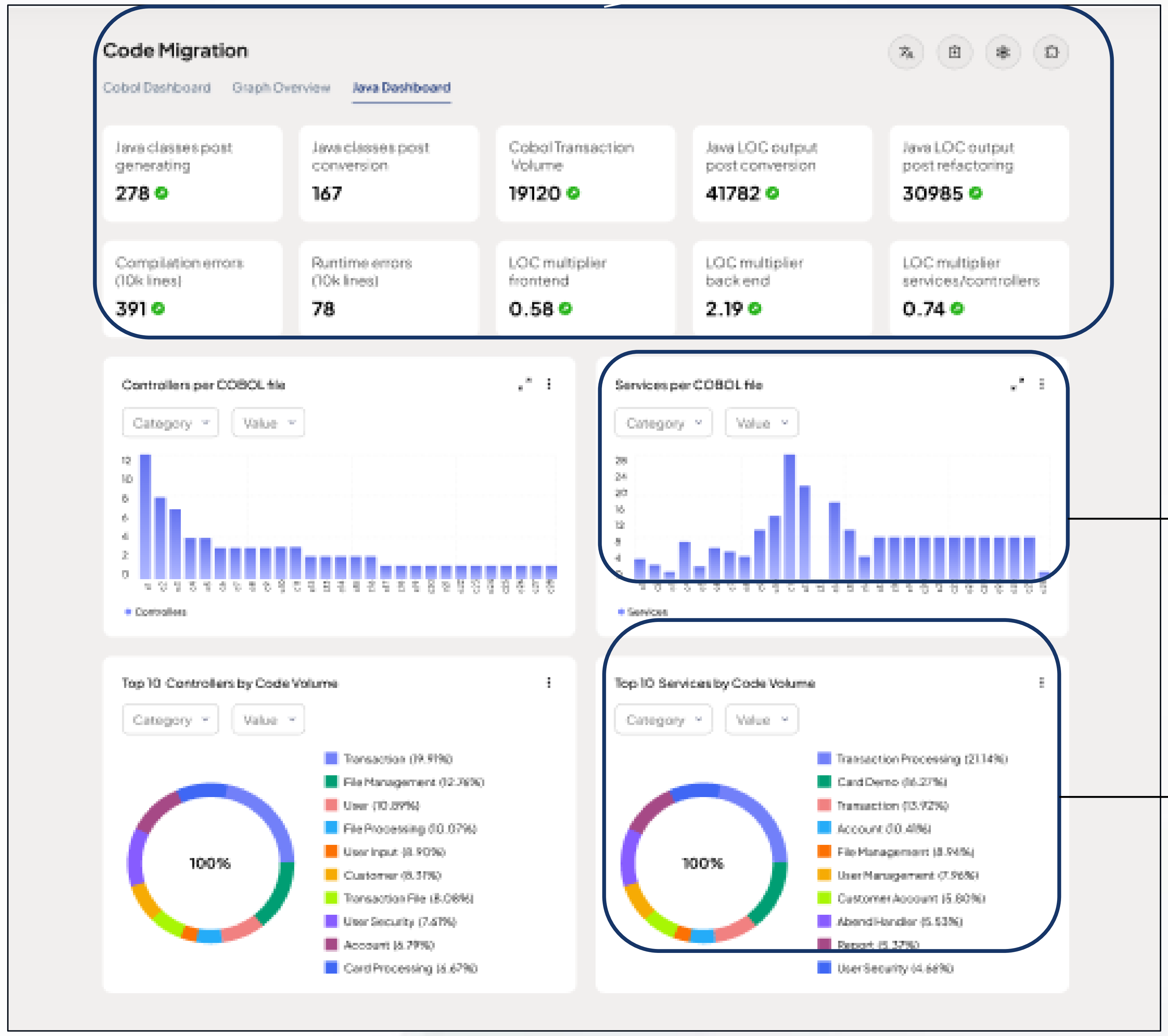


Code Migration

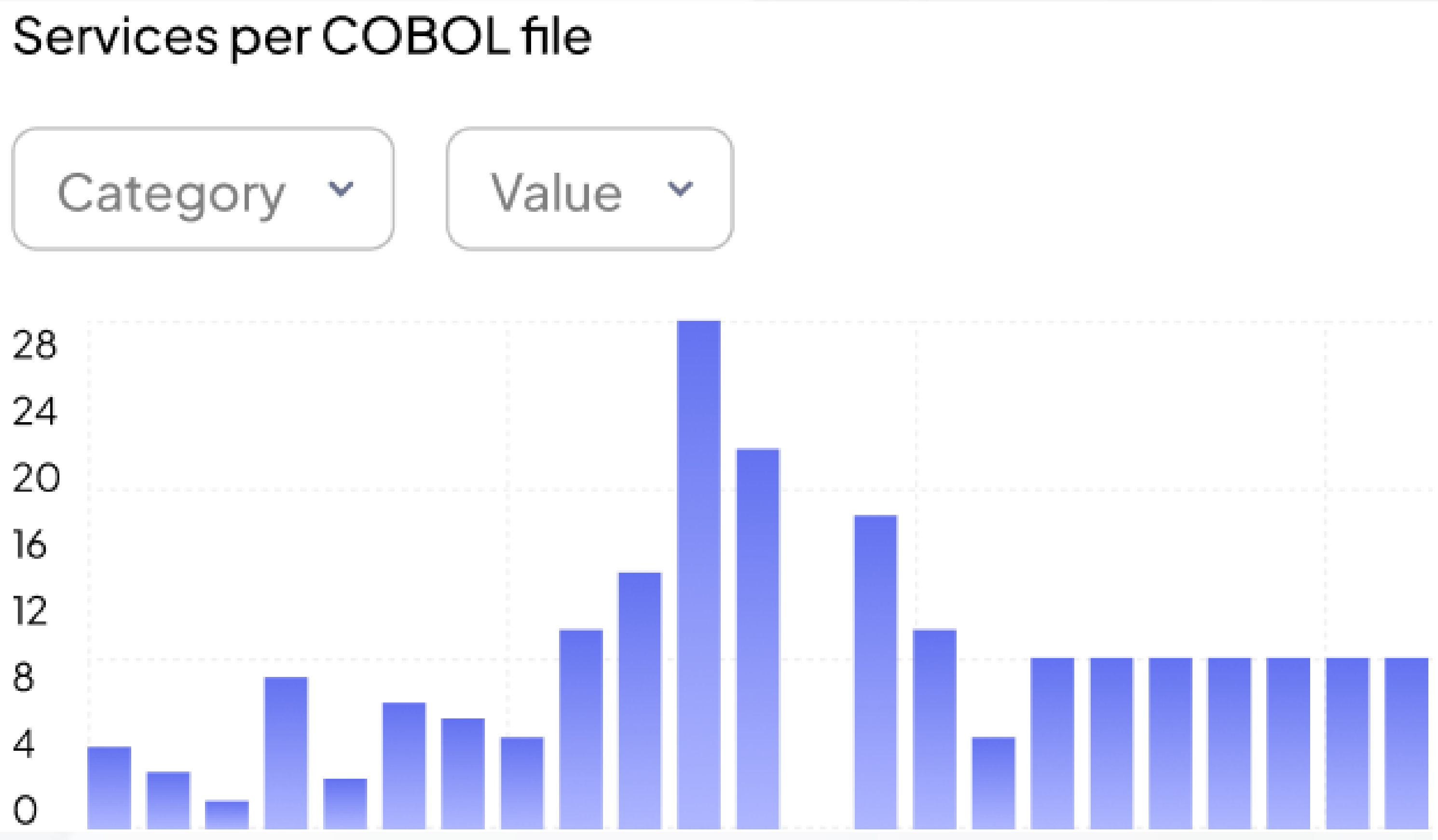
Cobol Dashboard | Graph Overview | Java Dashboard

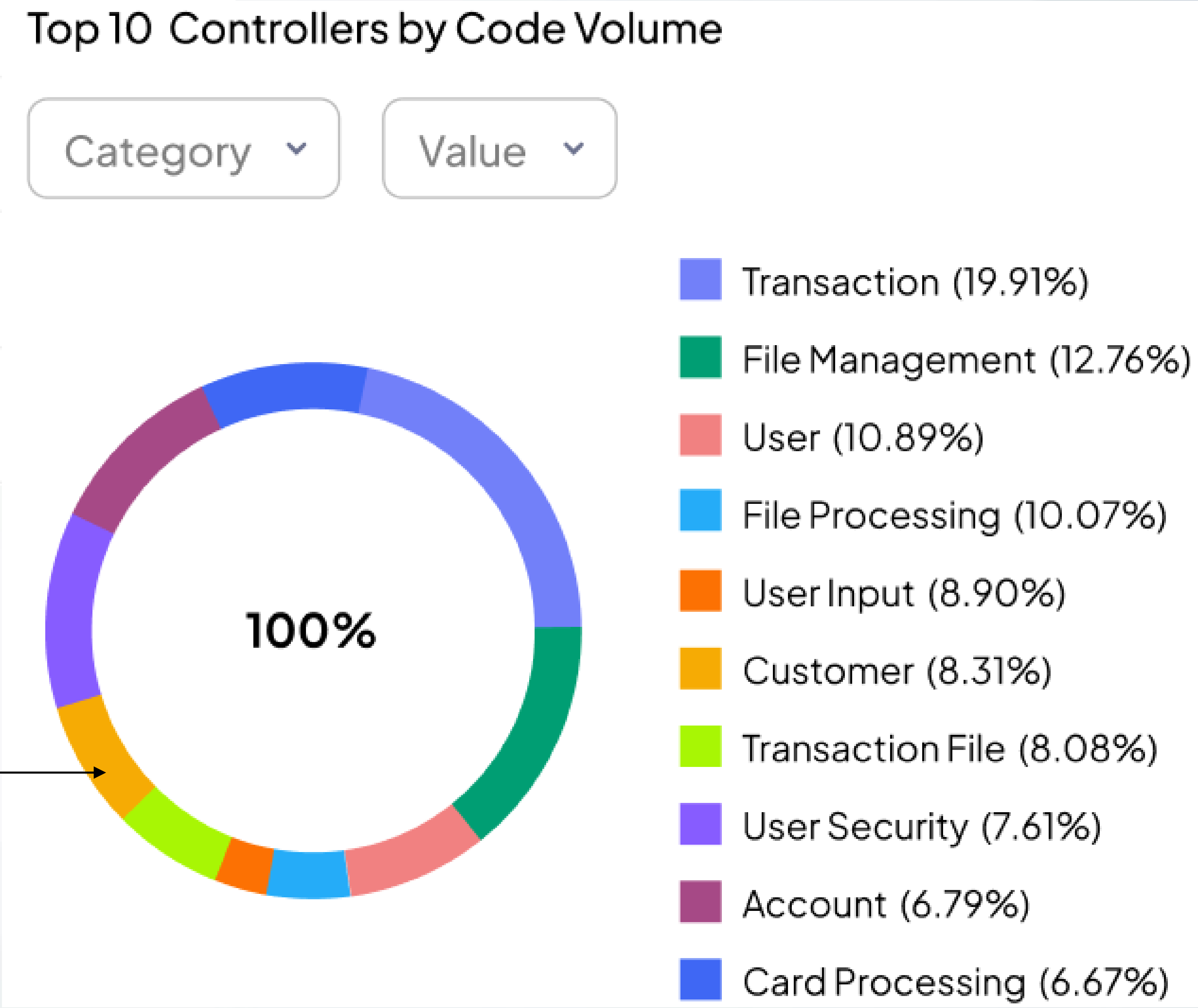
Java classes post generating 278 ✓	Java classes post conversion 167	Cobol Transaction Volume 19120 ✓	Java LOC output post conversion 41782 ✓	Java LOC output post refactoring 30985 ✓
Compilation errors (10k lines) 391 ✓	Runtime errors (10k lines) 78	LOC multiplier frontend 0.58 ✓	LOC multiplier back end 2.19 ✓	LOC multiplier services/controllers 0.74 ✓

Legacy Migration Insights

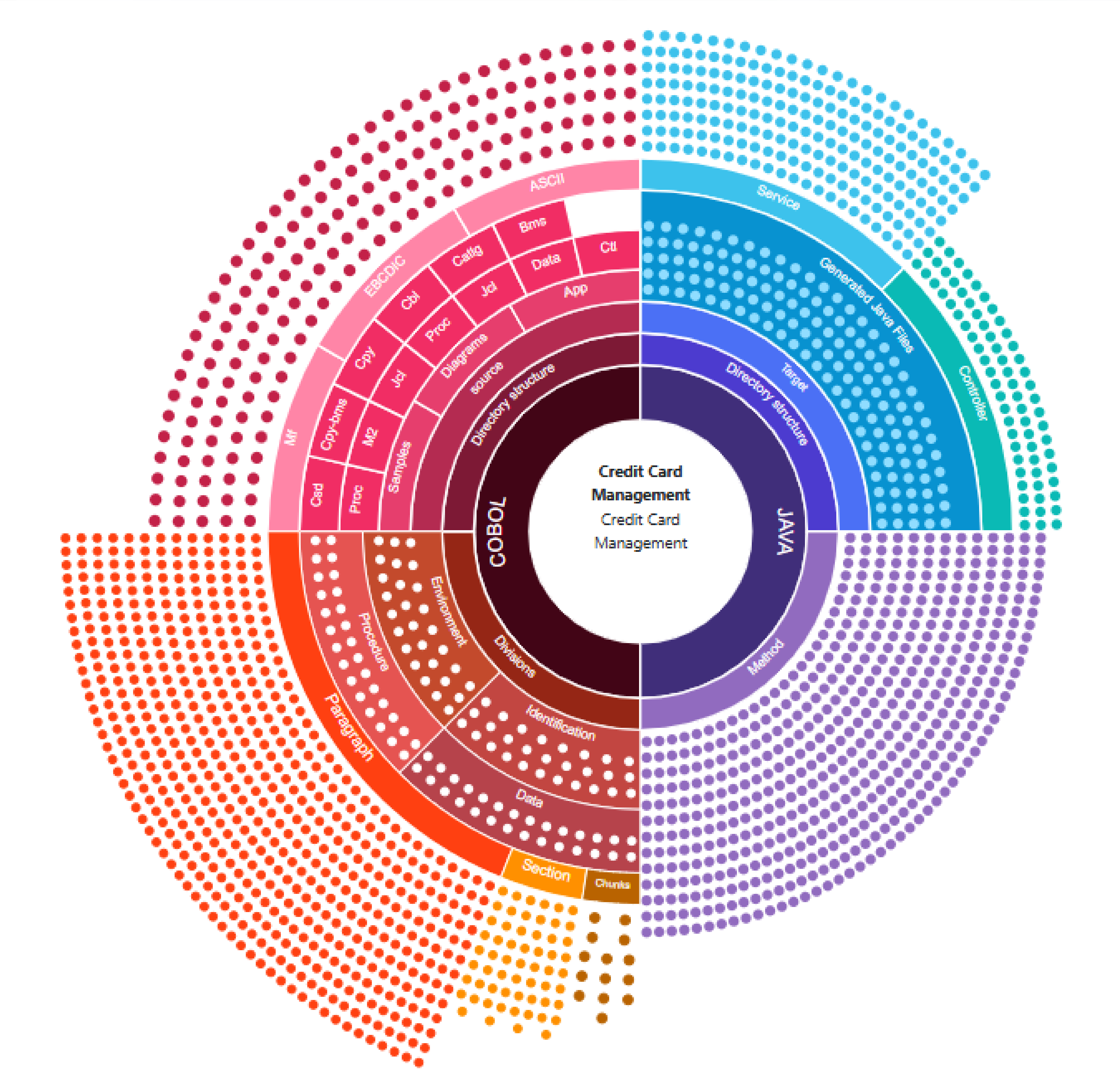


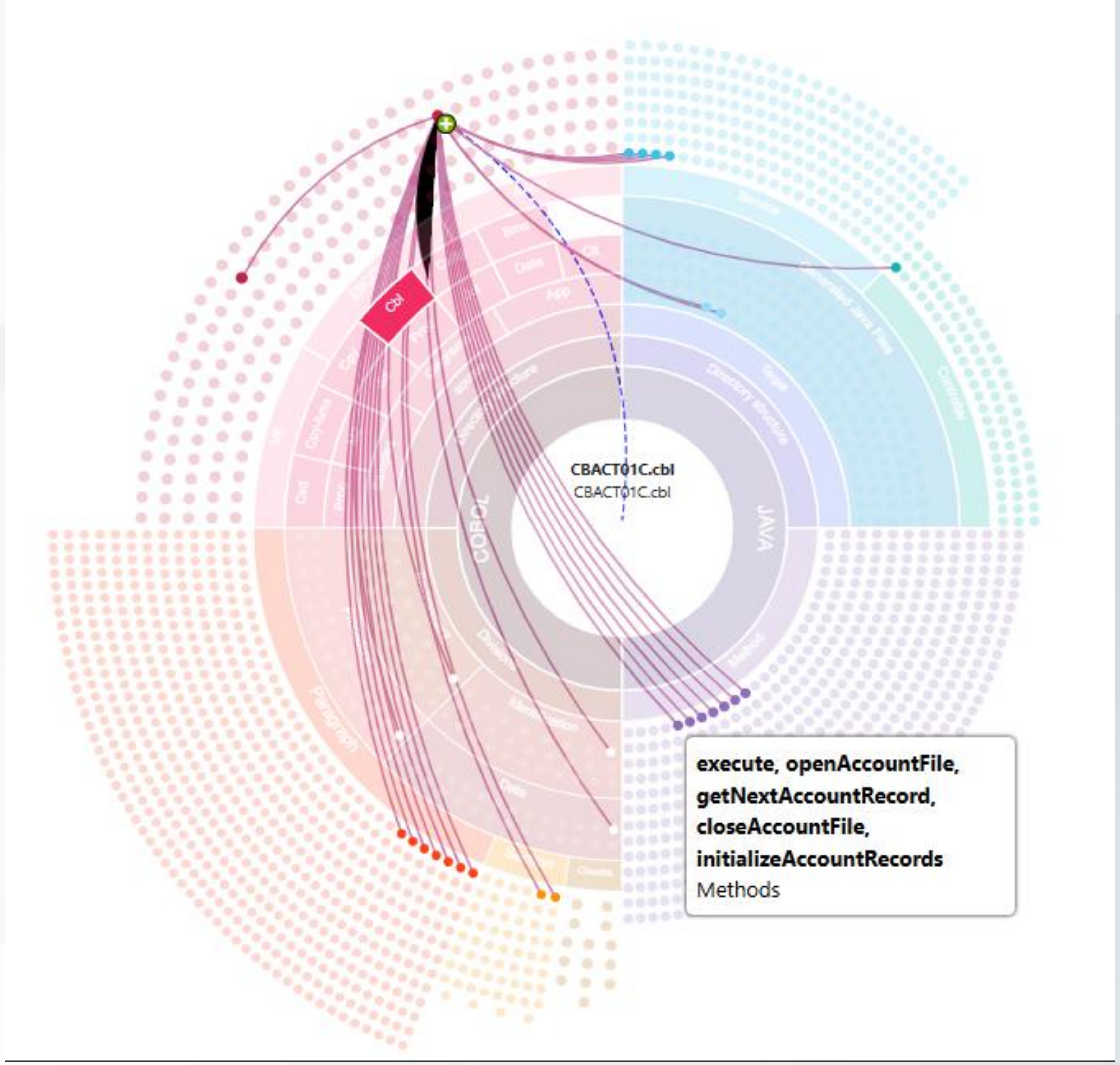
Migration Dashboard





CodeMap Explorer UI





execute, openAccountFile, getNextAccountRecord, closeAccountFile, initializeAccountRecords
Methods

Success Stories

Legacy to Modern Technology (Delphi to .NET 8 & React)

Client Overview

A **leading European education-technology provider** that processes payments, data and communications for **11,000+ schools and universities**.

Business Challenge

- **Nearly 3 million LOCs (Delphi)** could not scale to the cloud or deliver the rich browser experience
- Scarce Delphi talent slowed defect resolution,
- **Manual modernization failed** and program terminated after 1 year

Solution Overview

- Our agentic - driven migration framework **ASIMOV auto-generated clean C#/. NET Core APIs and a React application.**
- Relationship-heavy data was re-modelled in Neo4j.
- **Dedicated privacy agents** enforced field-level **encryption and anonymization** throughout the extract-transform-load process.

Key Deliverables

- **Cloud-native C#/.Net application**
- AI generated **automated Unit test with 85% coverage**
- **Accessible, responsive portal** for students, staff and administrators

Business Impact

- Modernization completed **in 40% cost** of Manual approach
- Created a foundation **for improved User experience** with latest technology

Technology Version Upgrade (Java 8 to Java 21)

Client Profile

A **widely-adopted inventory and warehouse-management platform** serving manufacturers and distributors worldwide, processing millions of transactions daily.

Business Challenge

- Critical services still ran on, out-of-support Java 7.
- Security patches had ceased
- Performance lagged as volumes grew
- Lack of automated unit test coverage

Solution Overview

- An LLM-assisted **upgrade pipeline automatically scanned one million lines of code, pinpointed deprecated APIs and produced modern Java 21 equivalents.**
- **Refactoring and container-ready builds** were rolled out in staged waves
- **Automated unit test generated** in jUnit5

Key Deliverables

- Java 21 codebase with **automated API tests**
- Updated **code documentation** for complete code base

Business Impact

- **Security-vulnerability** exposure window **cut by 75%**
- Significant **reduction in the size** of code base
- Inventory-sync **throughput was doubled**

Re-architecting Legacy (VB.Net to .Net Core)

Client Profile

A global logistics provider offering services in multimodal transport, container freight stations, and contract logistics.

Business Challenge

- Legacy application with monolithic, tightly coupled UI and business logic.
- Difficult to scale, maintain, or enhance due to outdated architecture

Solution Overview

- Applied a GenAI-agentic approach to analyse and modularize legacy VB.Net code
- Migrated UI to React for a modern web experience
- Backend restructured into .NET Core modules with clear separation of concerns

Key Deliverables

- Modular .NET Core backend and React-based frontend
- Automated toolchain for code analysis and transformation
- Technical documentation

Business Impact

- **Reduced feature development time by 50%**
- Improved system maintainability and developer onboarding
- **Future-ready, scalable and cloud agnostic architecture** ready for continuous modernization

Accionlabs

Thank You

www.accionlabs.com