WHITE PAPER

# The AI-Driven Applications *Modernization* Platform.

A Pega and AWS Co-Branded Solution for GSI's to accelerate Cloud Migration & Modernization on legacy applications.

PEGA | aws

WP / 2026 ▸ V1.0

# Table of Contents

# Introduction to AI-Driven Modernization

For decades, legacy applications have been the operational bedrock of global enterprises. Yet, these very systems—often built on monolithic, outdated technology stacks—now represent a significant barrier to growth, agility, and innovation. The high cost of maintenance, immense technical debt, scarcity of skilled talent, and the inherent risk of "big bang" migration projects have left many organizations in a state of operational paralysis.

This whitepaper introduces an AI-Driven App Modernization platform from Pega and AWS, designed not just to replatform legacy systems, but to fundamentally reimagine the business processes they support. This platform represents a paradigm shift, providing a low-risk, AI-accelerated methodology that enables organizations to shed technical debt and move away from high-risk, manual transformations. By uniting the global scale and advanced AI/ML infrastructure of AWS with Pega's industry-leading platform for AI-powered decisioning and workflow automation, this solution transforms complex legacy architectures into a portfolio of intelligent, agentic workflows. The result is a purpose-built, cloud-native system designed to increase automation, dramatically improve customer experiences, and create lasting business value.

This document's purpose is to provide a comprehensive technical overview of the platform's composable architecture and a proposed, three-phase methodology that powers these transformations, enabling organizations to de-risk their modernization journey, time-to-value, and unlock new avenues for innovation.

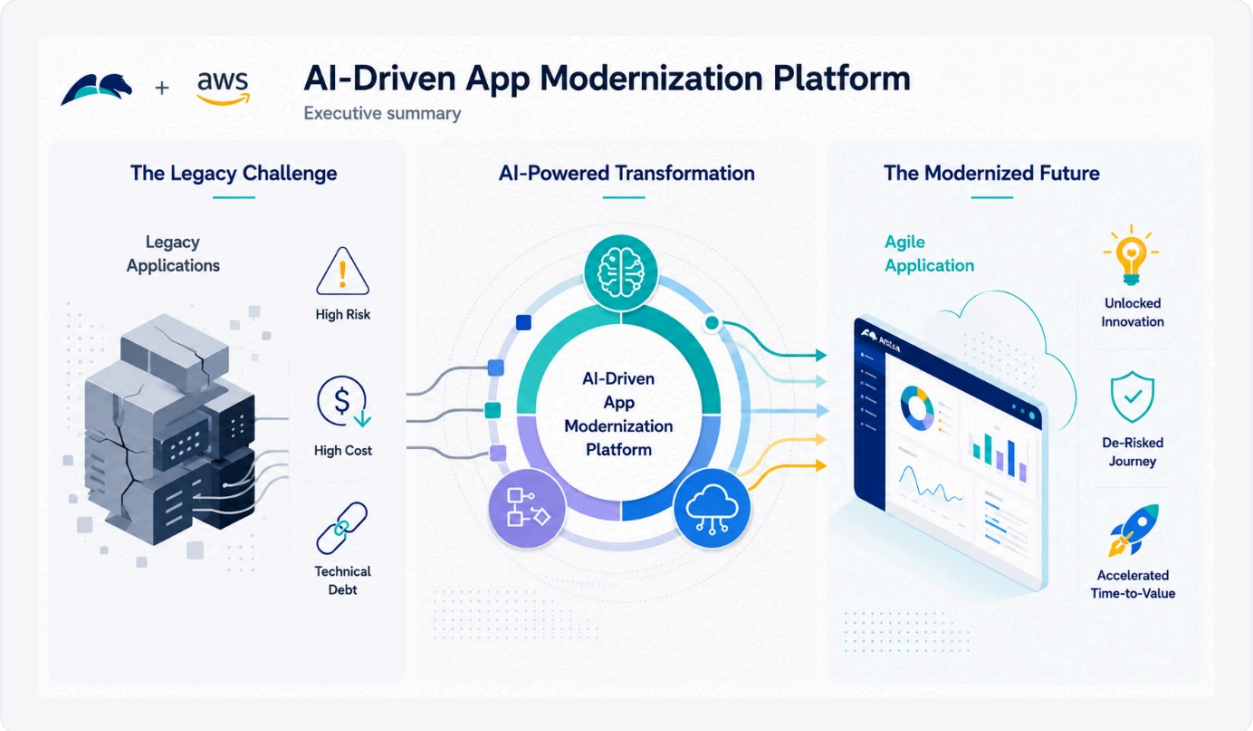



**Figure 1:** Shifting the Modernization Paradigm. The AI-Driven App Modernization Platform acts as a transformative engine, deconstructing high-risk, monolithic legacy systems into agile, low-risk, and cost-effective composable applications running on the cloud.

# Target Use Cases and Scenarios: From Modernization to True Transformation

The AI-Driven App Modernization platform is engineered to address a wide range of legacy environments. However, our approach goes beyond simple 1-to-1 replacement. The core value lies in the opportunity to reimagine outdated processes onto an agentic platform. This strategic shift transforms core operations into intelligent, automated workflows, unlocking dramatic improvements in efficiency, automation, and customer experience.

Our methodology provides a tailored, AI-accelerated path for the following complex legacy estates:

### Mainframe Systems

For core business applications running on mainframes—often the systems of record for banking, insurance, and logistics—our platform provides a low-risk, phased approach to extract business logic and data, transforming monolithic COBOL or PL/I applications into agile, cloud-native Pega workflows.

### Legacy BPM Platforms

Organizations looking to move off aging Business Process Management (BPM) suites (e.g., IBM BPM, Oracle BPM) can leverage a unique accelerator. The methodology is predicated on a direct-to-Blueprint import of BPMN diagrams and other key assets, enabling the rapid reimagining of existing processes into intelligent, automated Pega workflows while shedding significant technical debt.

### Legacy Java & .NET Applications

Custom-built applications on older Java or .NET frameworks often suffer from complex interdependencies and monolithic architectures. Our platform analyzes this code to identify and untangle business logic, enabling a strategic migration to a modern, microservices-based architecture on the Pega platform.

### Lotus Notes / Domino Applications

For the thousands of business-critical applications locked in Lotus Notes, we provide a specialized Notes-to-Blueprint modernization path. This dedicated capability, a result of targeted acquisitions, allows for the rapid analysis and transformation of Notes forms and business logic directly into the Pega platform, preserving critical business functions while moving to a modern, consolidated environment.

### SAP & Legacy ERP Transformation

For enterprises seeking a "clean core" ERP strategy, our platform acts as the agile innovation and automation layer. By building new applications and workflows on Pega, organizations can ringfence their legacy ERP (like SAP), reduce expensive customizations, and deliver agile, modern experiences without disrupting the core system of record.

## The Capstone: Application Consolidation and Rationalization

Ultimately, the goal is not to replace old silos with new ones. The true power of this platform-based approach is its ability to drive application consolidation and rationalization.

Instead of treating each legacy system as a separate, 1-to-1 modernization project, our platform provides a unified foundation to converge functionality. As you transform applications from Mainframe, Notes, and legacy Java, you can identify and eliminate redundant processes, creating a single, cohesive set of agentic workflows. This strategy fundamentally breaks down business and technology silos, resulting in a streamlined architecture that radically reduces complexity and lowers the Total Cost of Ownership (TCO) for your enterprise.

# Composable Platform Architecture

The proposed AI-Driven App Modernization Platform is built upon a unique composable architecture that combines two vertical pillars with a central engine of four horizontal processing layers. This structure provides a complete, end-to-end framework for the entire modernization journey, from foundational infrastructure to final business value delivery.


**Figure 2:** Anatomy of the AI-Driven App Modernization Platform: The platform's architecture integrates Pega and AWS pillars with four horizontal processing layers. The engine flows from the AWS foundation upward, sequentially analyzing legacy assets, designing the new application in Pega Blueprint, and generating it on the Pega Platform, all while supported by a universal data and integration layer and is crowned by the GSI Partner Services layer, which ensures expert delivery and business value realization.

> *"A complete, end-to-end framework for the entire modernization journey — from foundational infrastructure to final business value delivery.*

# The Core Modernization Engine

As shown in Figure 2, stacked on top of the AWS foundation is the central processing engine. It is comprised of four distinct horizontal layers that work in sequence to transform legacy assets into modern applications.

### Discovery, Analysis & Pre-processing

The first horizontal layer acts as the discovery engine. It uses specialized AWS tools like AWS Transform and Kiro to perform deep analysis of legacy source code, producing crucial Legacy Insights. A specialized Pega Blueprint agent then processes these insights, leveraging its native integration capabilities through Amazon Bedrock AgentCore (such as A2A and MCP) to create enriched artifacts for a clean, structured input for the design layer.

### AI-Powered Application Design and Generation (Pega Blueprint)

This is the "core design intelligence" of the platform. Pega Blueprint ingests the Enriched Artifacts from the analysis layer and uses generative AI to accelerate the design of the new Pega application. It translates legacy logic and business rules into a modern context through AI-Assisted Design, maps business processes with Workflow & Persona Modeling, and produces an Executable Architecture for the new application.

### AI-Powered Application Generation (Pega Platform)

This layer transforms the design into a tangible, working application. Leveraging Pega GenAI capabilities, it takes the Executable Architecture from Pega Blueprint and generates a functional Modernized application. Developers can then use the Low-code App Studio to enhance the application, which runs on Pega's enterprise-grade engine for Decisioning & Workflow Automation.

#### DATA & INTEGRATION SERVICES

This crucial horizontal layer underpins the entire engine, providing universal connectivity and data flow to ensure the modernized application operates seamlessly within a broader enterprise ecosystem. Its capabilities include:

- **Cloud-Native & Microservices Integration:** A robust framework for enabling communication through modern architectural patterns like microservices, APIs, and event-driven messaging, allowing the new Pega application to connect with other cloud-native services.
- **Legacy & External Platform Connectivity:** Provides customizable cloud-native integration solutions to connect with a wide range of external systems, including iPaaS solutions, legacy modernization platforms such as OpenLegacy, and other essential enterprise software.
- **Data Modernization Paths:** Offers prescriptive strategies and tools for migrating data from legacy databases to modern, cloud-native data stores.
- **Identity & Security:** Implements a unified security model to manage authentication, authorization, and fine-grained access control across all integrated systems.

# Foundation: Cloud Infrastructure & Managed Platform

The bedrock of this solution is a dual foundation that combines the global scale of AWS infrastructure with the robust, managed capabilities of the Pega Cloud platform.

## Layer 1: The Cloud & AI Infrastructure (AWS)

This foundational layer provides the underlying power, global reach, and elasticity for the entire solution. It leverages Amazon Web Services for core infrastructure, including:

- **Global Scale & Resiliency:** Access to a worldwide network of data centers, providing the foundation for high availability and disaster recovery.
- **Scalable Compute & Networking:** Elastic compute capabilities and robust networking to handle demanding enterprise workloads.
- **A Rich AI/ML Portfolio:** Direct, secure access to a powerful suite of advanced AI services, including Amazon Bedrock and Amazon SageMaker, which power the intelligence throughout the modernization journey.

## Layer 2: The Pega Cloud Platform

Running on this infrastructure is the Pega Cloud platform, a fully managed, automated, and secured Platform-as-a-Service (PaaS) specifically engineered to run enterprise-grade Pega applications. It is far more than just servers; it is a comprehensive service that offloads operational burdens and accelerates time-to-value. Key components of this managed service include:

- **An "Always-On" Architecture:** The platform is designed from the ground up for mission-critical reliability, providing an "always-on" architecture with built-in redundancy and proactive monitoring to ensure business continuity.
- **Enterprise-Grade Security & Governance:** Pega Cloud comes with a dedicated Security Operations Center (SOC) providing 24/7 threat monitoring. It enforces governance through features like role-based access, audit trails, and adherence to rigorous compliance standards.
- **Integrated DevOps & Automation:** The platform includes a standardized, best-practice CI/CD pipeline powered by Pega Deployment Manager, providing a fully automated workflow for deploying and promoting application changes, enforcing quality gates, and dramatically increasing release velocity.
- **Proactive Support & Telemetry:** Your environment is supported by deep Pega expertise, featuring 24x7x365 global support, predictive diagnostics, and continuous performance monitoring to ensure your applications are always running at peak efficiency.

ARCHITECTURE

# Delivery & Value Realization: GSI Partner Services

This layer is focused on business value and expert delivery. Global System Integrators (GSIs) with deep industry and Pega expertise leverage the entire underlying platform to provide strategic Consulting & Delivery services. They manage End-to-End Transformation Programs and offer pre-built, accelerator Industry Solutions that enrich the platform's output and reduce clients' time to market, ensuring the modernized application delivers its full business potential.

METHODOLOGY

# The 3-Phase Modernization Methodology

Now that we have outlined the overall AI-Driven App Modernization Platform, now let's understand how this platform can be leveraged using a three-phase methodology.

As summarized in Figure 3, the platform is underpinned by a versatile and prescriptive three-phase execution methodology. This framework is designed to be technology-agnostic, providing a universal ABC "Anything-to-Blueprint-to-Cloud" path for modernization. It systematically guides an organization from its unique source of legacy complexity—whether rooted in mainframe applications, distributed Java/.NET systems, or older packaged software—to a modernized, agile future on Pega Cloud.

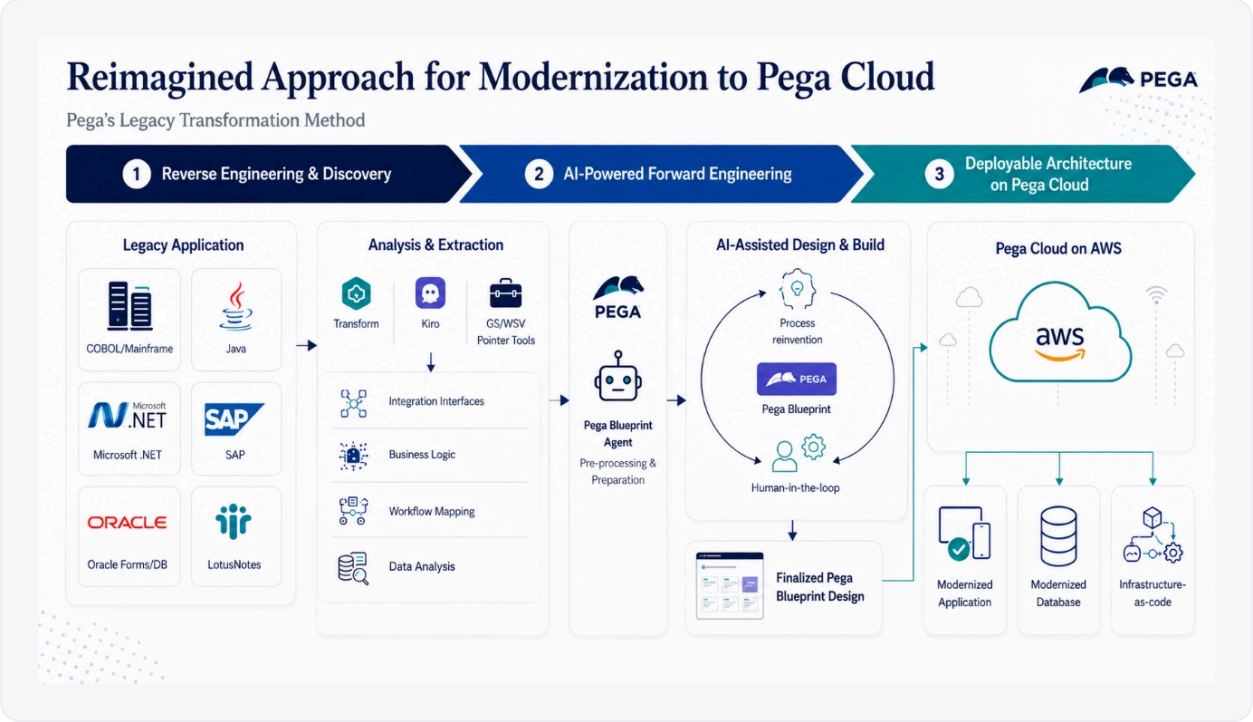



**Figure 3:** The 3-Phase Modernization Methodology: The methodology guides the transformation across three phases: first, reverse engineering legacy assets into structured artifacts (Phase 1); second, using those artifacts for AI-powered design in Pega Blueprint (Phase 2); and finally, deploying the resulting design as a modernized application on Pega Cloud (Phase 3).

# A universal path from legacy to cloud

01

## Phase 1: Reverse Engineering & Discovery

The primary goal of this initial phase is discovery and automated extraction. It begins by taking a wide range of legacy application assets as input, including COBOL/Mainframe, Java, .NET, SAP, Oracle Forms/DB, and Lotus Notes. These assets are processed using the powerful analysis engines of AWS Transform, Kiro, and other specialized partner tools. This automated analysis deconstructs the legacy system to extract its core components. The key output is a set of well-defined artifacts—Integration Interfaces, Business Logic, Workflow Mappings, and Data Analysis—which provides the essential, fact-based foundation for Phase 2.

02

## Phase 2: AI-Powered Forward Engineering

This phase uses agentic AI to accelerate application design and reimagine outdated processes. The legacy artifacts extracted during Phase 1 are fed into the Pega Blueprint Agent, the core AI engine of this methodology. This agent analyzes the legacy inputs, infuses them with Pega's embedded industry best practices, and generates a comprehensive starter application design. This output then facilitates a collaborative "AI-Assisted Design & Build" cycle of process innovation and human-in-the-loop review, resulting in a High-Fidelity Blueprint—a validated, best-practice-aligned application design ready for deployment. See Appendix [E]: The Pega Blueprint Agent for a detailed overview.

03

## Phase 3: Deployable Architecture on Pega Cloud

This final phase materializes the architectural blueprint as a production-grade application on Pega Cloud. The core strategy is an incremental rollout using the Strangler Fig pattern, which allows the new Modernized Application to gradually and safely take over functionality from the legacy system with zero business disruption. This phase delivers three key components: the Modernized Application itself; a secure Coexistence Architecture that uses an Anti-Corruption Layer to ensure seamless integration; and the Infrastructure as Code (IaC) templates that automate the deployment of all supporting AWS services. This IaC provisions the critical integration backbone—such as the Amazon API Gateway acting as the Strangler Façade and the secure hybrid networking to your data center—required for the Pega Cloud application to operate with the legacy environment (see Appendix D for a detailed list of deployed services).

*To illustrate the power and precision of this three-phase methodology in a real-world context, the section below examines how this applies to one of the most prevalent and complex modernization challenges enterprises face today: the modernization of a legacy mainframe application.*

# Deep Dive: Phase 1 — Reverse Engineering & Discovery

As shown in Figure 4 below, this foundational phase tackles the single greatest challenge in mainframe modernization: the "black box" of undocumented, decades-old functionality. The primary goal is to perform a deep, AI-driven analysis that transforms this opaque legacy environment into a structured, queryable knowledge base.

We will explore this phase through its three core components:

- **The Input:** The specific mainframe source code from different technologies that the platform ingests.
- **The Process:** The powerful analysis engine that deconstructs the code.
- **The Output:** The critical, technology-agnostic intelligence that is extracted to fuel the modernization in Phase 2.

Let's examine each of these in detail.

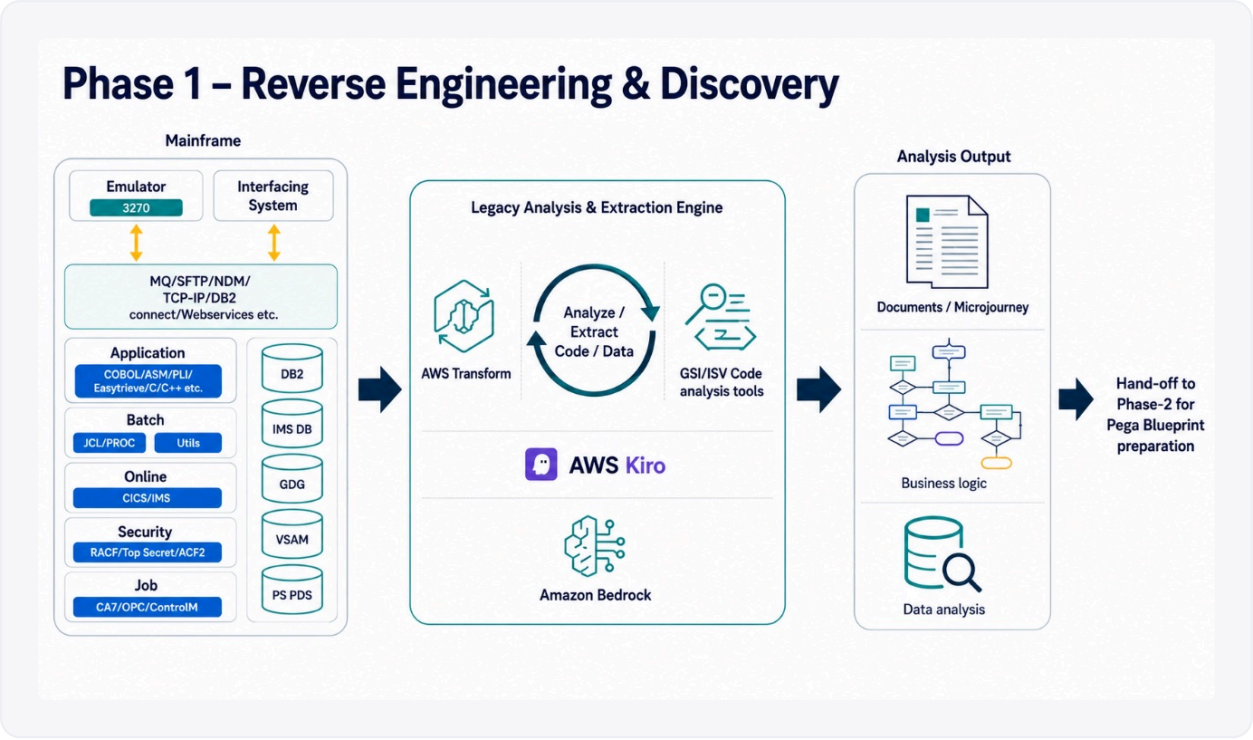



**Figure 4:** The Discovery Process: From Legacy Input to Extracted Output: This diagram details the core Process of analysis and extraction. It shows how the legacy application Input is deconstructed through a business hierarchy—from Line of Business down to the Component Level—to produce the final Output: a set of structured, technology-agnostic artifacts including Business Logic, Integration Interfaces, Workflow Mappings, and Data Analysis.

## Input (Legacy Application & Data Sources)

The platform is designed to ingest a wide array of core mainframe technologies. This includes code written in COBOL, Job Control Language (JCL) that orchestrates batch processing, data files like VSAM, messaging queues such as MQ, foundational databases including DB2 and IMS, and more.

## Process (Legacy Analysis & Extraction Engine)

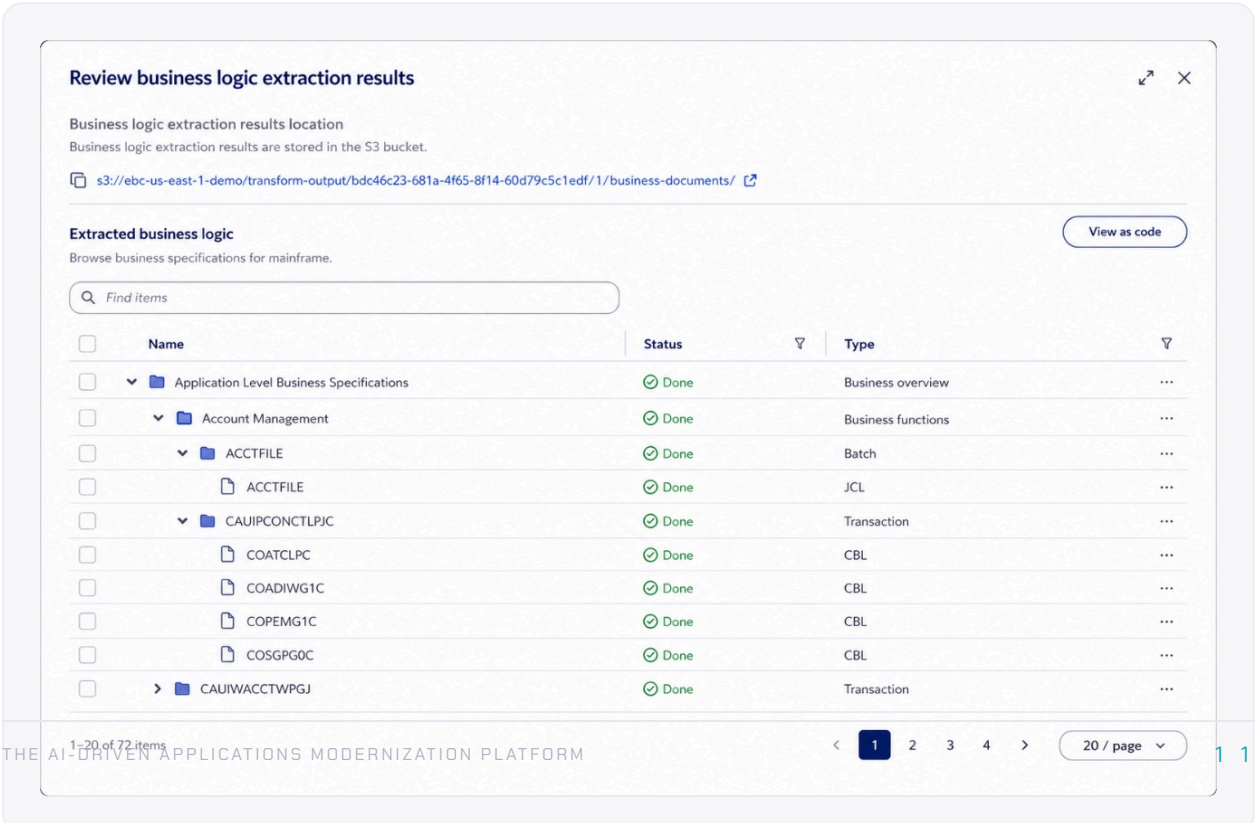
At the heart of this phase is a powerful analysis engine. AWS Transform is the primary platform, performing deep code analysis to parse syntax, identify dependencies, and map complex relationships. The engine's core purpose is to translate legacy complexity into a structured, universal format that the Pega Blueprint Agent can understand and reason in Phase 2. While opinionated, the architecture is adaptable; for unique environments, AWS Kiro can be used with specialized partner solutions. This ensures the process can meet customers where they are, turning opaque legacy systems into a queryable knowledge base.

## Output (Extracted Artifacts — The Foundational Intelligence for the Pega Blueprint)

This process does not produce running code. Instead, it generates the Raw Extracted Artifacts—the technology-agnostic intelligence required to build the Phase 2 modernization blueprint. This output decouples the business value from the legacy code.

### Decoupled Business Logic Extraction (BLE)

To separate business value from legacy code, AWS Transform's Business Logic Extraction feature uses AI agents to categorize the application's functionality into a clear business hierarchy. This multi-level analysis starts from a high-level Line of Business (e.g., Credit Cards), drills down into Business Functions (e.g., Rewards), then Business Features (transaction/batch jobs logic), and finally extracts specific rules at the Component Level from COBOL and JCL files (as shown in Figure 5 below). By translating complex, embedded logic into plain language, the platform provides clear visibility that helps business stakeholders validate existing rules and allows technical teams to group common functions into domains for effective migration planning. Ultimately, this captures all business rules and decisioning logic in a technology-agnostic format, ready to be reimplemented as modern Pega processes.



## Workflow Mapping

Building directly on the Business Logic Extraction, this output provides a visual representation (see Figure 6) of how legacy components are interconnected within their business domains. The tool generates dependency maps that show how different programs, transactions, and batch jobs work together to fulfill a business function/workflow. This visualization is critical for strategic planning, as it allows architects and business leaders to identify loosely coupled functionalities that are ideal candidates for initial modernization.

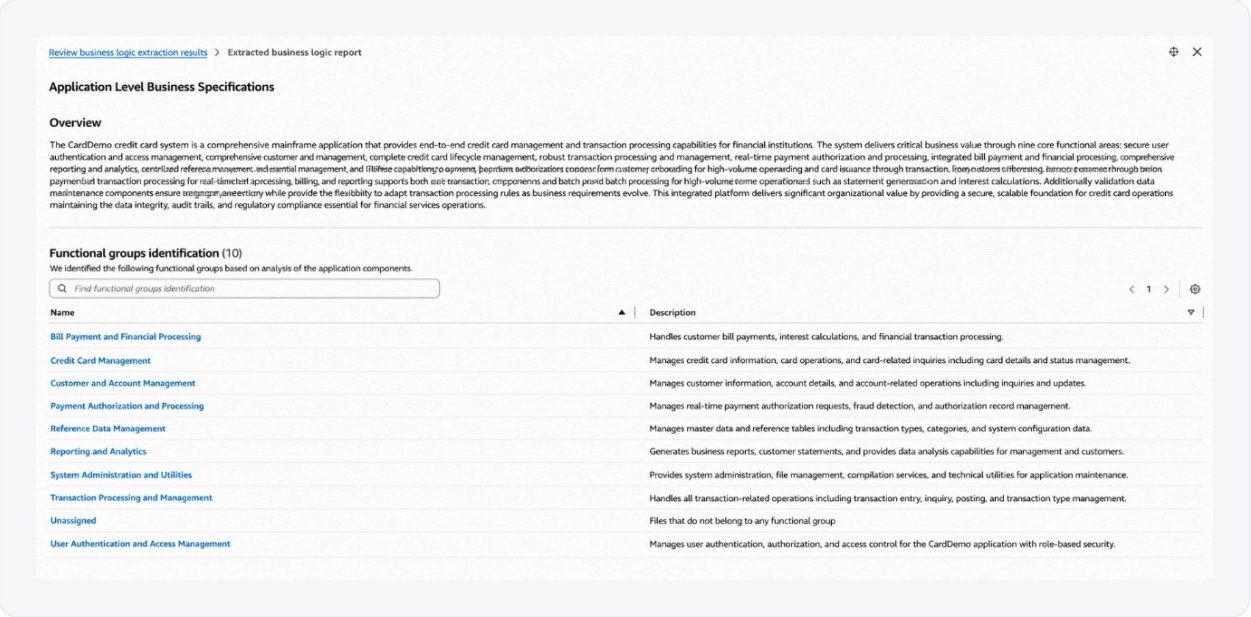



**Figure 6:** Workflow and Application-Level Business Specifications.

### Comprehensive Integration Inventory

A clear and complete inventory of all external and internal APIs, batch jobs, and communication points. This is the foundational data needed to plan the Strangler Fig pattern and identify where tools like OpenLegacy will be required for hybrid coexistence architecture.

### Data Analysis

To fully deconstruct legacy data, AWS Transform generates two critical outputs that serve as the direct input for the Pega Blueprint Agent: A Data Lineage map and a comprehensive Data Dictionary.

### Data Lineage

This artifact provides complete visibility into data relationships by mapping exactly how programs and JCL jobs interact with data sources (as shown in Figure 7 below). It reveals the "where" of the data by distinguishing between read, write, update, and delete operations across all files and databases, making it essential for identifying dependencies and planning a safe migration.

## Data Dictionary

This artifact provides the "what" of the data by acting as a comprehensive, auto-generated catalog for complex and often poorly documented mainframe data structures, as shown in Figure 8 below. It provides field-level analysis of copybooks, including their PICTURE clauses, REDEFINES, and modern data type equivalents, accelerating data transformation design.

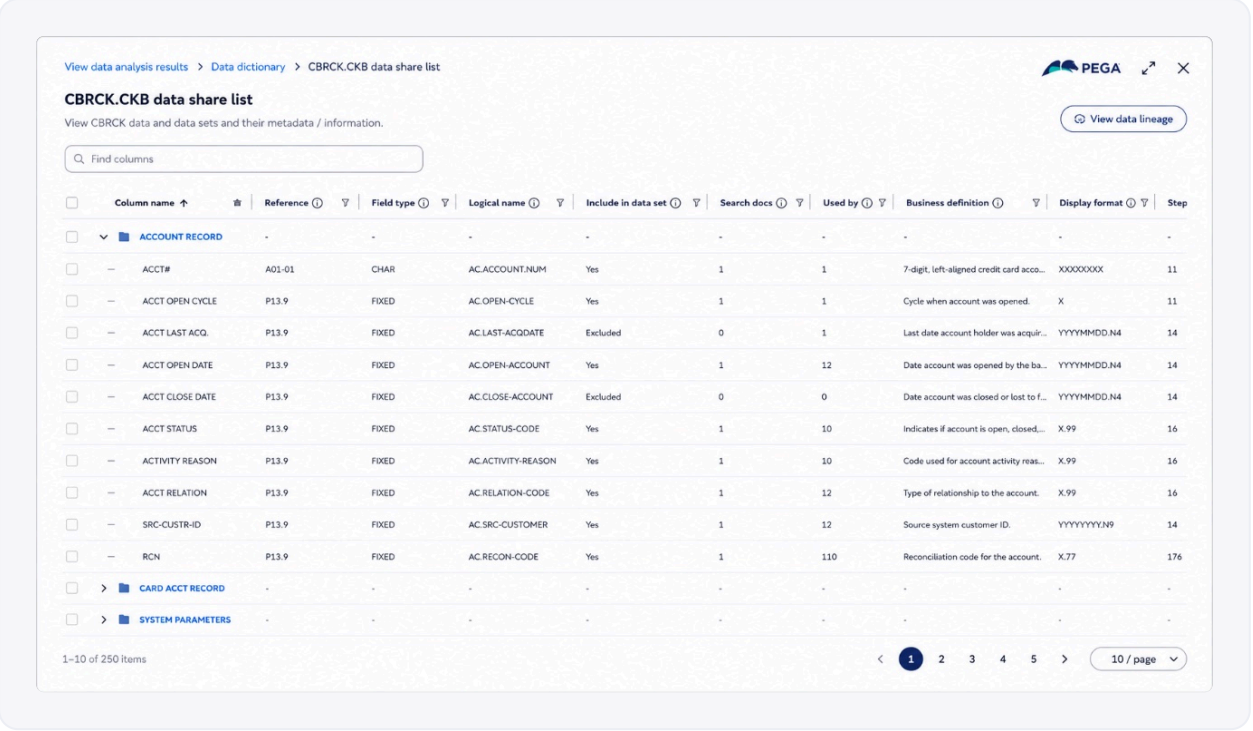



**Figure 8:** Data Dictionary.

> *A comprehensive, auto-generated catalog for complex and often poorly documented mainframe data structures — accelerating data transformation design.*

# Deep Dive: Phase 2 — AI-Powered Forward Engineering

Following the deep discovery in Phase 1, as shown in Figure 9 below, Phase 2 shifts from analysis to tangible creation. This phase is driven by the Pega Blueprint platform, which ingests the detailed artifacts from AWS Transform into the Pega Blueprint platform, which in turn, translates them into a high-fidelity, forward-looking, and modern Pega application design. The entire process is transparent, collaborative, and adaptable to different modernization strategies.

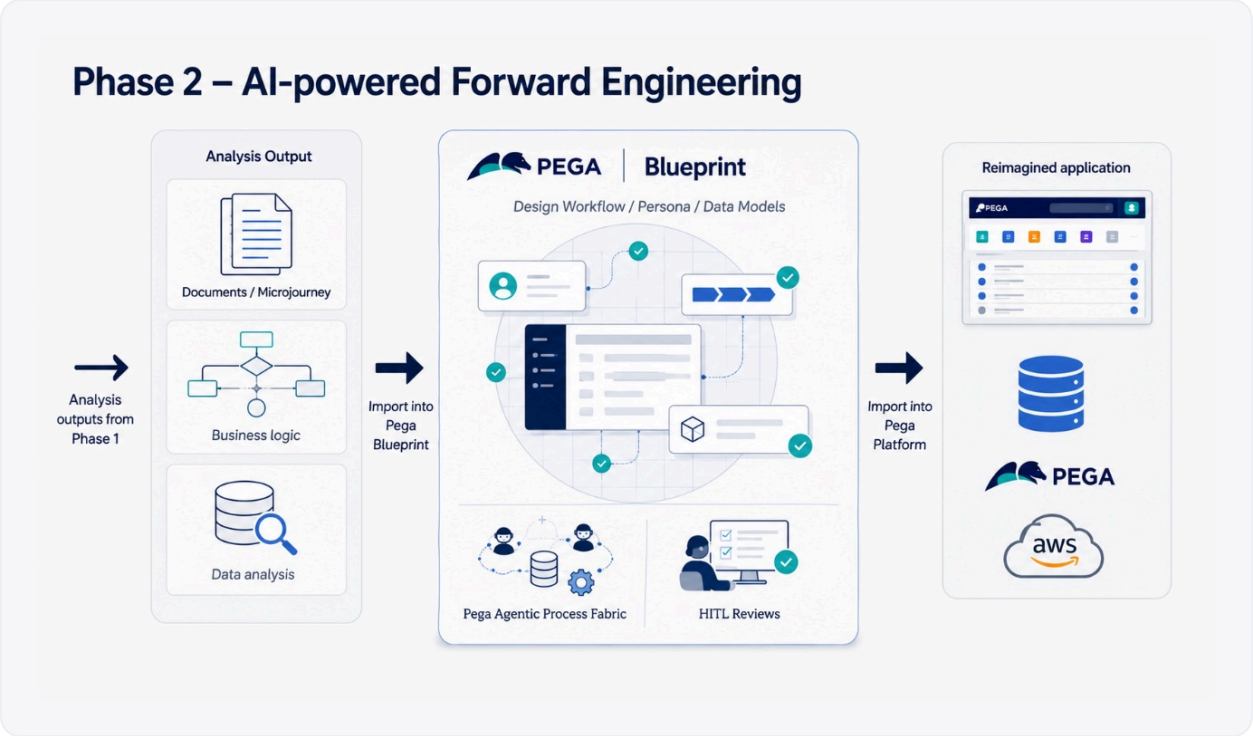



**Figure 9:** From Legacy Insights to Modern Blueprint: Phase 2 ingests the legacy insights and artifacts from Phase 1 and uses the Pega Blueprint platform's collaborative cycle of AI-assisted design and human-in-the-loop validation to translate them into a high-fidelity, validated blueprint for the new Modernized Application.

## Input: Pre-processing the AWS Transform Artifacts

The second phase begins as the Pega Blueprint Agent ingests the rich, structured intelligence generated by AWS Transform. The agent's primary role here is not just to pass data along, but to act as an intelligent interpreter, pre-processing these artifacts and translating them into a format that the Pega Blueprint design engine can use to build a high-quality, enterprise-grade application foundation. The agent thoroughly prepares each of the four key artifacts from Phase 1:

### Workflow Mapping & Decoupled Business Logic

The agent first analyzes the Workflow Mapping to understand the end-to-end sequence of programs and transactions that constitute a business process. It then correlates this macro view with the granular rules from the Decoupled Business Logic. By combining the "how" (workflow) with the "what" (rules), the agent prepares a preliminary structure of Pega case types, stages, and steps, ensuring the core intent of the legacy process is preserved and ready for modern design.

### Comprehensive Integration Inventory

The agent parses the complete list of APIs, message queues, and batch jobs. It pre-maps standard interfaces (e.g., REST, SOAP) to their corresponding Pega connector rules and identifies the data being exchanged. This provides an immediate, actionable inventory for the design engine and flags any proprietary or complex integrations that may require specialized solutions, giving architects early visibility.

### Data Analysis (Data Dictionary & Data Lineage)

This is where the agent performs its most critical data transformation. It intelligently combines the two data outputs from Phase 1:

- It uses the Data Dictionary to translate mainframe data definitions (like COBOL PICTURE clauses, REDEFINES, and OCCURS) into modern, standard field types (e.g., Text, Number, Date).
- It then consults the Data Lineage map to understand the relationships and CRUD (Create, Read, Update, Delete) patterns between data sources.

By synthesizing the "what" (dictionary) with the "where" (lineage), the agent proposes a complete set of modern Pega data objects, complete with their fields and relationships, ensuring the new application's data structure is robust and maintains integrity. With this thorough pre-processing complete, the Pega Blueprint Agent has transformed the raw intelligence into a highly structured, context-rich format. These prepared artifacts are now primed for ingestion by the AI-Powered Design & Generation Engine, which will build the final, executable application blueprint.

## Process: The AI-Powered Design & Generation Engine

With the legacy artifacts meticulously prepared by the Pega Blueprint Agent, the core of Phase 2 begins. The Pega Blueprint platform ingests this structured data not as a static file, but as the contextual foundation for an interactive design process. This engine bridges the old and new worlds by first using AI to generate a complete application design, and then engaging human experts to validate and refine it.

### Step 1: AI-Powered Design Generation

The Pega Blueprint Platform, now acting as a design architect, intelligently interprets the prepared artifacts to propose a complete, modern application structure. It doesn't just map fields; it translates legacy concepts into a high-fidelity Pega design.

### From Workflow and Logic to Case Lifecycles

The agent takes the prepared Workflow Mapping and Business Logic to construct the backbone of the new application: a modern Case Type structure. For example, a legacy JCL dependency graph for "Monthly Customer Invoicing" is automatically proposed as a "Customer Invoicing" Case Type, with the sequential program calls from the old workflow becoming distinct stages and steps in the new, visualized business process workflow.

### From Data Structures to Modern Data Models

Using the pre-processed Data Analysis artifacts, the agent proposes a complete, object-oriented data model. It transforms the cryptic hierarchies from old VSAM copybooks and DB2 tables into relational Pega Data Objects. For instance, a customer copybook is re-engineered into a Customer data object, with its relationships to other new objects like Policy and Claims automatically defined.

### From Legacy Calls to Modern Integrations

The agent reviews the Integration Inventory and earmarks every external program call (e.g., CICS LINK or MQ calls) as a placeholder for a modern Pega Connector. It defines the inputs and outputs for each future API, providing an instant, actionable integration plan that architects can use to build out the new service-oriented architecture.

## Step 2: AI-Assisted Reimagination & Design

The AI-generated starter blueprint is not the endpoint; it is the launchpad for reimagining your application at speed. This phase moves beyond simple validation into active, collaborative design within the Pega Blueprint platform, which serves as a shared palette for business and IT to co-create the future state. Instead of just correcting the past, stakeholders are empowered to:

- **Rearchitect Customer Journeys:** Go beyond the constraints of the legacy system to design streamlined, end-to-end customer experiences.
- **Introduce Intelligent Automation:** Embed AI and decisioning directly into the new workflow to automate tasks, predict needs, and guide users to optimal outcomes.
- **Simplify the Data Model:** Radically streamline and rationalize decades of complex data structures into a clean, business-centric model.
- **Design Digital-First Experiences:** Craft modern, AI-first user interfaces that are intuitive and effective on any channel.

## Inherit Enterprise-Grade Capabilities

Crucially, modernizing onto this platform means the reimagined application automatically inherits the powerful, out-of-the-box capabilities of the Pega Platform. As stakeholders design the new workflows, they are building on a foundation that already includes:

- A Rich Case Management Framework for handling long-running, complex work.
- Predictive AI and Decisioning Agents that can be seamlessly integrated.
- A true Center-out™ Architecture that separates business logic from channels.
- Built-in Reporting and Process Optimization tools.

This continuous feedback loop between human expertise and AI, built on a platform of powerful capabilities, is what elevates the output from a simple interpretation to a high-fidelity, transformational blueprint—a validated, optimized, and future-proof application design.

> *"This elevates the output from a simple interpretation to a high-fidelity, transformational blueprint — a validated, optimized, and future-proof application design.*

## Output: Assets Ready for Build

The result of this collaborative phase is a validated blueprint that provides a set of concrete, deployable assets for the final generation phase:

- **The Modernized Pega Blueprint design,** including case types, data models, integrations, and user personas.
- **The Modernized Data Transformation Strategy,** detailing how legacy data will map to the new Pega data schema.

# Deep Dive: Phase 3 — Deployable Architecture on Pega Cloud

This is the realization phase, where the high-fidelity blueprint from Phase 2 is transformed from a strategic design into a living, production-grade application. As illustrated at a high level in Figure 10, this phase is about executing a risk-mitigated transition from legacy to modern by deploying a sophisticated, incremental architecture.

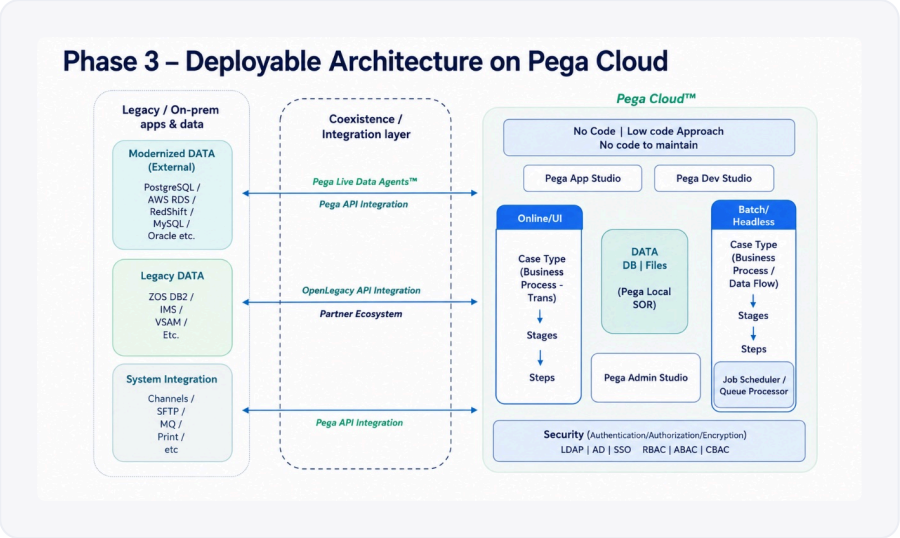



**Figure 10:** Architectural Pattern for Hybrid Coexistence: This diagram shows how the new Pega application and the legacy mainframe can run in parallel. A managed API layer allows the new application to safely retrieve data from the legacy system without being directly coupled to it.

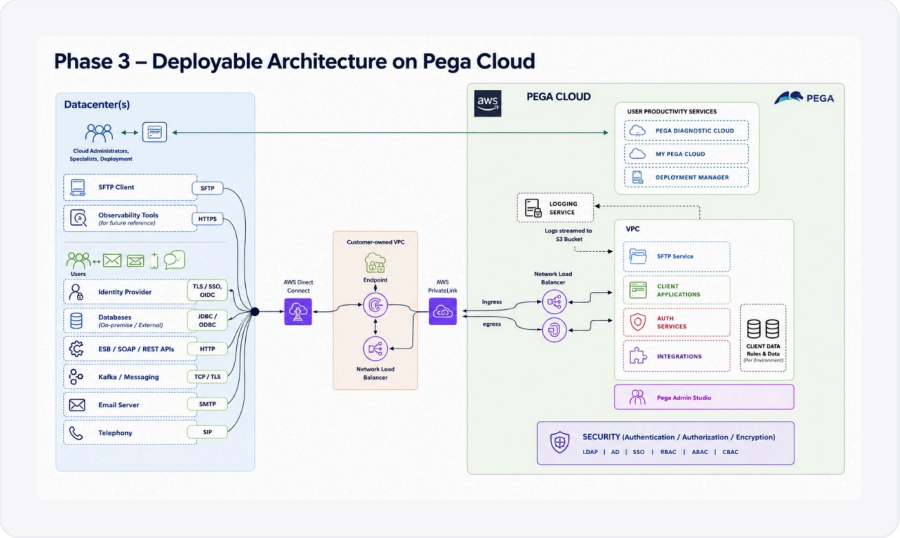



**Figure 11:** Phase 3 – Deployable Architecture on Pega Cloud: This diagram illustrates the final deployment phase where the high-fidelity blueprint becomes a production-ready application. It shows how the Modernized Application is deployed on Pega Cloud, utilizing the Strangler Fig pattern for incremental rollout and Infrastructure as Code (IaC) for automated provisioning of supporting AWS integration services, ensuring secure coexistence with legacy systems.

# The Core Strategy: Incremental Modernization with the Strangler Fig Pattern

Modernization is rarely a "big bang" event. The most successful approach is incremental, using the Strangler Fig pattern to gradually replace a legacy system. The Host Tree is the legacy mainframe, and the Fig Vine is the new Pega application that grows around it.

This pattern relies on two critical components, as illustrated in Figure 11 below:

### The Strangler Façade (API Gateway)

This is the central nervous system of modernization, acting as a single, intelligent entry point for all application traffic. It provides dynamic, policy-based routing while handling cross-cutting concerns like security and observability (see Appendix A for detailed capabilities).

### The Anti-Corruption Layer (ACL)

Implemented using OpenLegacy, this is a sophisticated translator that protects the new Pega application from legacy complexity. It allows Pega to communicate with the mainframe using modern APIs, without needing to understand legacy protocols or data formats (see Appendix B for detailed functions).

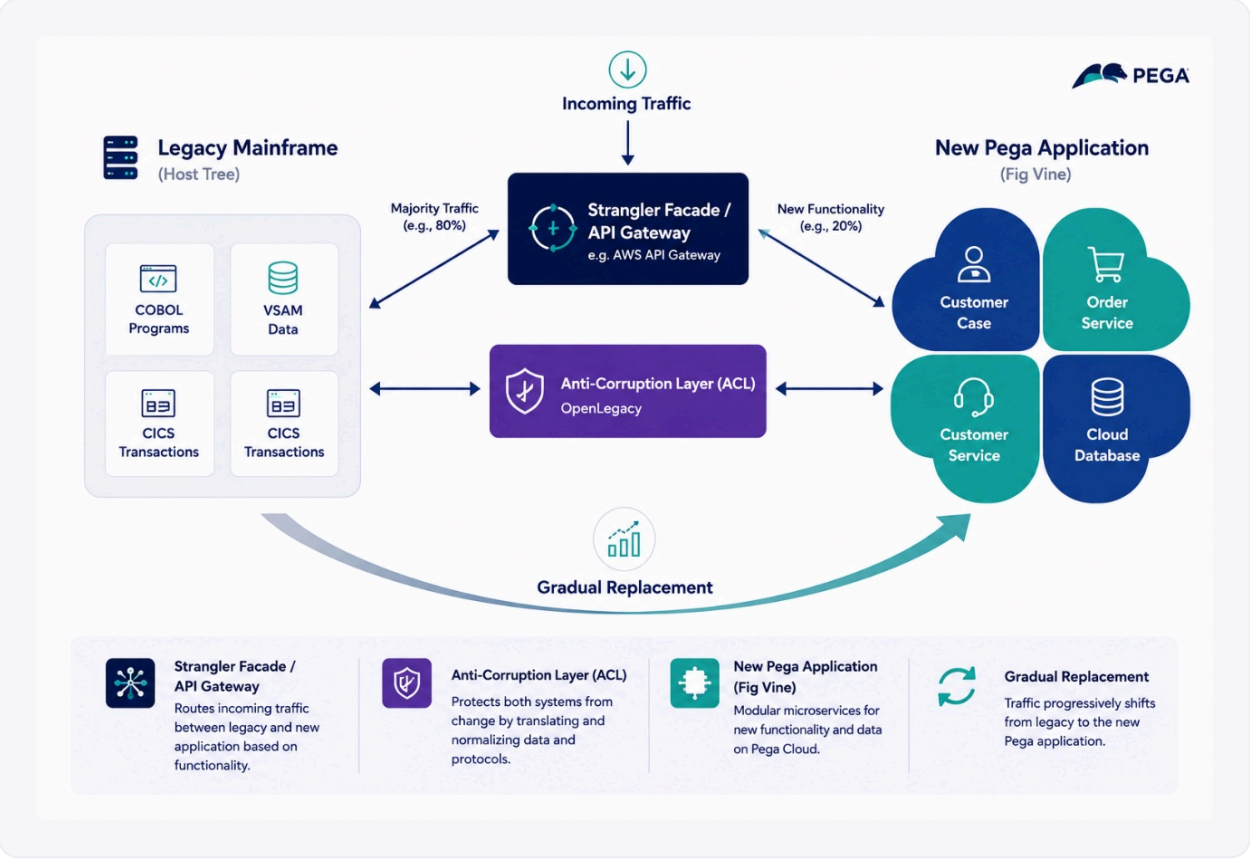



**Figure 11:** The Strangler Fig Pattern in Action: The architecture uses a Strangler Façade (API Gateway) to intelligently route traffic between the legacy mainframe and the new Pega application. The Anti-Corruption Layer, implemented by OpenLegacy, is the critical component that protects the modern application by creating clean, modern REST APIs for any legacy data or logic that is still needed.

# Putting the Strategy into Practice: Two Paths to Modernization

The legacy modernization strategy supports two primary paths. The recommended approach is an incremental one that delivers value quickly while mitigating risk, as shown in Figure 12 below.

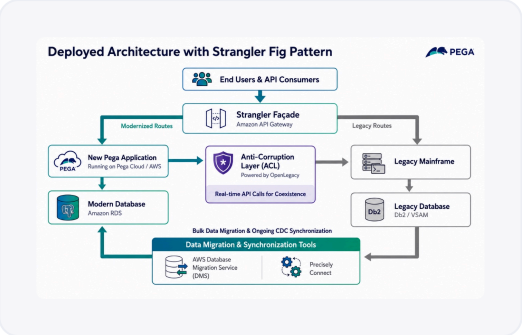



**Figure 12:** The Deployed Strangler Fig Architecture: A deployed architecture using the Strangler Fig Pattern to incrementally modernize a legacy mainframe system. A façade routes traffic, while an Anti-Corruption Layer and data migration tools ensure seamless coexistence and data synchronization between the new and old systems.

### Path 1: Coexistence and Hybrid Agility (The Recommended Path)

This is the heart of the incremental strategy. The application is modernized in logical "migration waves," where each wave delivers a complete business domain and its associated data.

- **Initial State:** The facade routes 100% of traffic to the legacy mainframe.
- **Migration Wave 1: Modernize the "Account Management" Domain:** The first full business domain, "Account Management," is built as a Case Type in Pega, encompassing all related functions (e.g., viewing, updating, and creating accounts). As a core part of this wave, the corresponding account data is migrated from the legacy system to the new Pega database using the dedicated data migration strategy. The Strangler Façade is then reconfigured to route all account-related API traffic (e.g., GET /api/accounts/{id}, POST /api/accounts, PUT /api/accounts/{id}) to the new Pega application, which now works against its own modern datastore.
- **Ongoing Coexistence:** While this is happening, the new "Account Management" feature may still need data from a domain that has not been migrated yet (e.g., checking a customer's billing history). For this, it makes a clean API call to the Anti-Corruption Layer (Appendix B), which retrieves the data from the mainframe without Pega ever touching the legacy system directly.
- **Eventual State:** The process continues, with each wave migrating into another business domain (like "Billing" or "Order Management") and its associated data, until the mainframe is fully "strangled" and can be safely decommissioned.

### Path 2: Full Migration

In scenarios where a "big bang" is feasible, the blueprint can map every legacy artifact to a modern Pega or AWS equivalent. While less common, this approach results in a single, comprehensive deployment and involves a one-time, large-scale data migration event.

### The Data Strategy: A Hybrid Approach to Migration & Synchronization

A successful modernization is underpinned by a robust data strategy. The methodology employs a specialized, hybrid tooling strategy using AWS DMS for relational data and partner tools (like Precisely and Connect) for complex, non-relational data. This handles both bulk migration and ongoing synchronization using Change Data Capture (CDC) (see Appendix C for a detailed breakdown of data tooling).

# The Technical Foundation: Automated Build and Deployment

This entire strategy is underpinned by a powerful, automated toolchain that brings the modernized application to life.

### Automated Application Build

The validated case types, data objects, and UI designs from the blueprint are used as direct inputs to automatically generate the core Pega application structure, minimizing manual effort.

### Automated Provisioning with Infrastructure as Code (IaC)

Pre-built IaC templates are used to deploy the entire supporting AWS architecture, including all necessary networking, security, and data services needed to connect Pega Cloud to the legacy environment (see Appendix D for a detailed list of deployed services).

A HOLISTIC, LAYERED SECURITY MODEL

The architecture is secured at every level, from the underlying infrastructure to the end-user application:

- **Infrastructure-Level Security:** Is established using the IAM Roles and Policies deployed via the IaC templates, enforcing least-privilege access between all AWS services and systems.
- **Application-Level Security:** Is managed within Pega itself, using its robust Role-Based Access Control (RBAC) and Single Sign-On (SSO) integration to control fine-grained user authentication and authorization.

## Outcome: Secure, Agile, and Incremental Modernization

The final outcome is not just a deployed application, but a strategic platform for continuous improvement. This three-phase approach delivers:

- **A Secure and Scalable Pega Application:** Running on a fully managed, resilient cloud platform.
- **Seamless Legacy Coexistence:** Enabling business continuity and mitigating the risks associated with a "big bang" migration.
- **Incremental Value Delivery:** Allowing the business to realize the benefits of modernization from the very first component deployed, rather than waiting for the entire project to be completed.

# Key Benefits for Business Stakeholders

While the technology is powerful, the ultimate measure of success is business impact. This section translates the platform's technical advantages into the tangible business outcomes that matter to stakeholders.

## 01

### Accelerated Time-to-Value

By dramatically reducing manual effort in discovery (Phase 1) and design (Phase 2), the AI-driven approach cuts down project timelines from years to months. This means the business starts realizing the benefits of modernization—whether cost savings, new revenue streams, or improved efficiency—significantly faster.

## 02

### Reduced Modernization Risk

The platform directly mitigates the primary risks that cause modernization projects to fail. The automated extraction of business logic prevents the inadvertent loss of critical functionality. The iterative "human-in-the-loop" process ensures the new application is validated against business requirements at every step, eliminating costly rework and ensuring the final product meets expectations.

## 03

### Lower Total Cost of Ownership (TCO)

Moving from expensive, high-maintenance legacy systems to an efficient, managed Pega Cloud on AWS architecture delivers substantial financial benefits. It slashes licensing and MIPS costs, eliminates technical debt, and frees up valuable IT resources to focus on innovation instead of "keeping the lights on."

## 04

### Increased Business Agility and Innovation

The modernized Pega application is not merely a lift-and-shift of old functionality. It is a transformation to a low-code platform that empowers the business. This newfound agility allows the organization to rapidly adapt to changing market demands, launch new products, and continuously improve processes without getting stuck in long IT development cycles.

## 05

### Enhanced Customer and Employee Experience

Modernization directly impacts the bottom line by improving user-facing interactions. A modern, responsive, and intelligent application built on Pega leads to higher customer satisfaction, increased loyalty, and improved productivity and morale for employees who no longer have to fight with outdated tools.

NEXT STEPS

# Next Steps & Call to Action

The journey from legacy paralysis to modern agility is now clearer and more achievable than ever. The AI-Driven App Modernization Platform from Pega and AWS provides the technology, methodology, and expert guidance to ensure your transformation is a success.

We invite you to take the next step in your modernization journey.

## Schedule a Personalized Demo

See the platform in action. Contact us to schedule a live demo tailored to your specific legacy systems and business goals. Our experts will show you how we can turn your most complex legacy challenges into opportunities for growth.

## Book a Discovery Workshop

Ready for a deeper dive? Engage with our modernization specialists in a complimentary discovery workshop. In this session, we will help you analyze your application portfolio and map out a high-level transformation strategy using our proven methodology.


CONTRIBUTING AUTHORS

**Surender Kumar**
Mainframe Modernization Architect
Pega

**Andre Boaventura**
AWS Solution Architect
Pega

**Sourav Sarkar**
Senior Specialist Solutions Architect
AWS

**Ujwal Bukka**
Senior Partner Solutions Architect
AWS

REFERENCE

# Appendix

# A. Detailed Capabilities of the Strangler Façade

The Strangler Façade is more than just a simple router. It is the central nervous system of the modernization strategy, typically implemented using a robust API Gateway like Amazon API Gateway. Its primary function is to act as a single, unified entry point for all application traffic, intelligently deciding where to direct each incoming request based on a configurable set of rules. This routing is not static; it is a dynamic, policy-based mechanism that allows for precise control over the modernization journey. Beyond simple routing, the Strangler Façade provides several critical capabilities:

### Unified Security Enforcement

It becomes the single point of enforcement for security policies like authentication (AuthN) and authorization (AuthZ), ensuring consistent security across both modern and legacy endpoints.

### Observability and Centralized Logging

Since all requests pass through the gateway, it provides a single, invaluable point for logging, monitoring, and tracing. This makes it easier to debug issues and understand traffic patterns during the transition.

### Client Decoupling

The façade abstracts the backend complexity. Client applications (e.g., a mobile app or a web portal) only need to know the gateway's address, not the location or technology of the systems behind it. This means backends can be swapped from mainframe to Pega with zero changes to the client-side code.

### Performance and Resilience

The gateway can handle cross-cutting concerns like response caching and rate-limiting to protect backend services from being overwhelmed.

# B. The Anti-Corruption Layer (ACL)

The Anti-Corruption Layer is a critical defensive pattern that acts as a sophisticated, bidirectional translator between the modern Pega application and the legacy mainframe. Implemented using OpenLegacy, its sole purpose is to protect the new application from the "corruption" of legacy complexity. The Pega application should not have to understand mainframe concepts like COBOL data structures, EBCDIC character sets, or CICS transactions. The ACL handles this translation work seamlessly. The Anti-Corruption Layer's key functions include:

### Automated API Generation

OpenLegacy automatically inspects legacy artifacts (like COBOL copybooks and program definitions) and generates modern, OpenAPI-compliant REST API contracts. This turns a cryptic mainframe program into a resource that looks and feels like any other modern microservice.

### Protocol Mediation

The new Pega application communicates using standard, stateless HTTP/REST protocols. The ACL receives these calls and translates them into the stateful, specialized protocols required to interact with mainframe systems, completely hiding this complexity from the modern developer.

### Data Transformation

This is one of its most critical roles. The ACL automatically handles the transformation between modern and legacy data formats. This includes converting data types (e.g., a JSON string to a COBOL PIC X(20)), translating character sets (e.g., UTF-8 to EBCDIC), and correctly handling complex mainframe data structures like REDEFINES and OCCURS.

### Decoupling and Abstraction

By providing this clean API boundary, the ACL ensures the new Pega application is completely decoupled from the mainframe's implementation details. Pega developers can work faster using familiar tools (REST, JSON) without needing to be mainframe experts. This abstraction allows the underlying mainframe logic to be changed or replaced later with zero impact on the Pega application, as long as the API contract is honored.

# C. Detailed Breakdown of the Data Strategy

A successful modernization is underpinned by a robust and flexible data strategy. The methodology employs a specialized, hybrid tooling strategy to handle both one-time migrations and ongoing synchronization using Change Data Capture (CDC).

### FOR RELATIONAL DATA (E.G., DB2)

**Tool:** AWS Database Migration Service (DMS).

**Capabilities:** DMS is highly efficient for bulk schema and data migration from mainframe relational databases to Amazon RDS. It also provides native Change Data Capture (CDC) capabilities to handle ongoing, real-time replication, ensuring data stays in sync during a long coexistence period.

### FOR NON-RELATIONAL & COMPLEX FILE DATA (E.G., VSAM, IMS)

**Tools:** Specialized partner solutions such as Precisely Connect and others.

**Capabilities:** These platforms have deep knowledge of complex COBOL copybook layouts to reliably extract, transform, and load this data into a modern relational format. Crucially, they also offer robust Change Data Capture (CDC) features specifically designed for these non-relational mainframe data sources, enabling real-time synchronization.

# D. Detailed List of Deployed Services (IaC)

A key activity in Phase 3 is the automated deployment of the supporting AWS architecture. This is achieved by leveraging a library of pre-built Infrastructure as Code (IaC) templates (using tools like AWS CloudFormation or Terraform). These templates provision all the necessary AWS services required to connect the Pega Cloud environment to legacy systems and enable the modernization strategy, including:

| | |
|---|---|
| **Amazon API Gateway** | To serve as the intelligent Strangler Façade, routing traffic between modern and legacy endpoints. |
| **AWS Database Migration Service (DMS)** | To manage relational data migration and provide Change Data Capture (CDC) for data synchronization. |
| **Secure Hybrid Networking** | Deploys the core cloud network infrastructure, including the Virtual Private Cloud (VPC), subnets, and security groups, which is then securely extended to the on-premise data center via AWS Direct Connect or a VPN Gateway to create a private, hybrid communication backbone. |
| **IAM Roles and Policies** | To enforce secure, least-privilege access between all AWS services, Pega Cloud, and the legacy systems. |
| **Observability and Monitoring** | Deploying Amazon CloudWatch for centralized logging, metrics, and alarms gathered from the API Gateway and other services. |
| **Compute for Partner Tools** | Provisioning Amazon EC2 instances or container services (ECS/EKS) required to run specialized partner data migration tools such as Precisely and Connect. |

# E. The Pega Blueprint Agent

The Pega Blueprint Agent is a generative AI engine deeply integrated within the AWS Transform. It serves as the intelligent design component of the methodology, translating the raw technical analysis from the discovery phase into a modern, best-practice Pega application design.

## Core Functionality: From Analysis to Blueprint

### Embedded Analysis & Ingestion

The agent operates directly on the output generated by the AWS Transform discovery tools. As AWS Transform analyzes the legacy application codebase (e.g., Mainframe COBOL, legacy Java), it extracts a detailed repository of technical artifacts. The Pega Blueprint Agent is designed to consume this structured output directly.

### Best Practice Enrichment

This is the core value added. The agent uses Pega's vast library of industry-specific best practices, cross-references the technical artifacts from AWS Transform against proven application models. It identifies the underlying business logic within the legacy code and determines the optimal way to represent it as a modern Pega workflow.

### Generative Design

The agent then generates a comprehensive starter Blueprint, translating the logic of the legacy system into a clear, editable, and business-friendly application design.

## The Input: AWS Transform Artifacts

The Pega Blueprint Agent is engineered to understand the specific, structured output produced by the AWS Transform analysis. This includes, but is not limited to:

- **Code Component Analysis:** Deconstructed COBOL programs, Java classes, or .NET assemblies.
- **Data Model Extraction:** Analysis of database schemas, file layouts (like COBOL copybooks), and in-code data structures.
- **Transaction & Logic Mapping:** A repository of identified business transactions and the code blocks that execute them.
- **UI Component Analysis:** Deconstruction of legacy screens (like Mainframe BMS maps) into their constituent fields and actions.

By working directly with the output of the AWS Transform, the agent ensures a seamless and highly accurate transition from technical analysis to business-focused design.

## The Blueprint Deliverable: A Rich Application Design

The generated Pega Blueprint is a multi-faceted design asset that provides a running start for development. Key components include:

| | |
|---|---|
| **Case Type Structure** | A full hierarchy of case types and sub-cases that model the core business process. |
| **Data Models** | A complete, proposed data structure with fields, data types, and relationships. |