

# Challenges

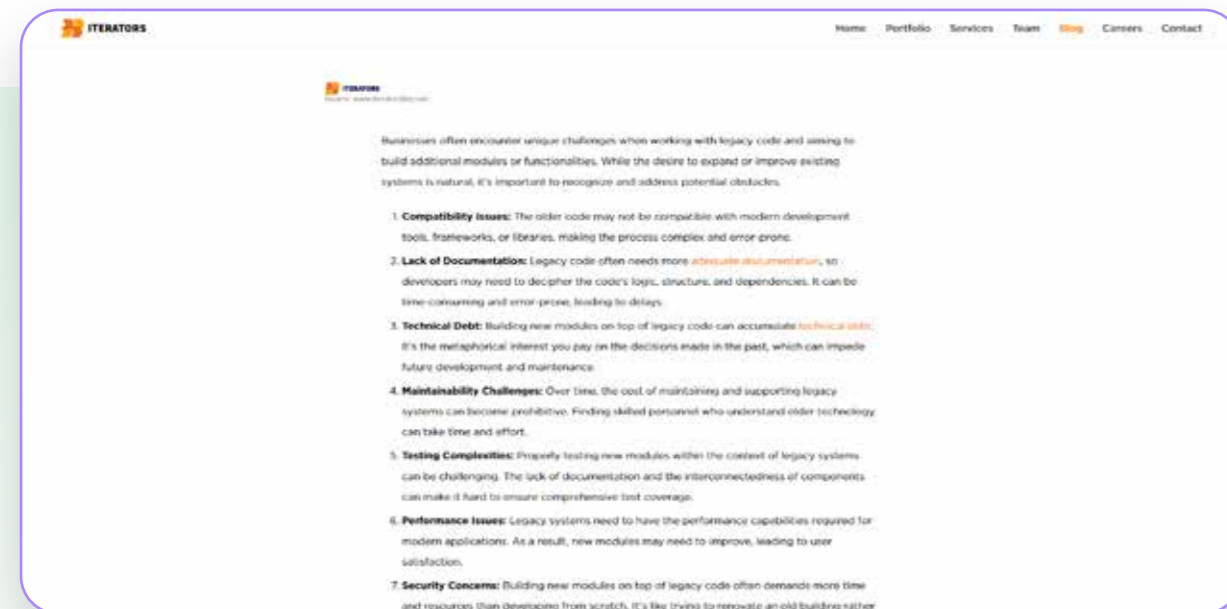
## in Legacy Conversion

### Security Vulnerabilities

40% of organizations have reported security breaches linked to outdated legacy systems.

### Escalating Maintenance Costs

Legacy systems consume up to 60% of IT budgets, limiting resources for innovation.



[Go to source](#)

### Integration Complexity

62% of IT leaders identify integration with modern applications as a top challenge.

### Insufficient Documentation

Over 70% of legacy codebases lack adequate documentation, complicating maintenance efforts.



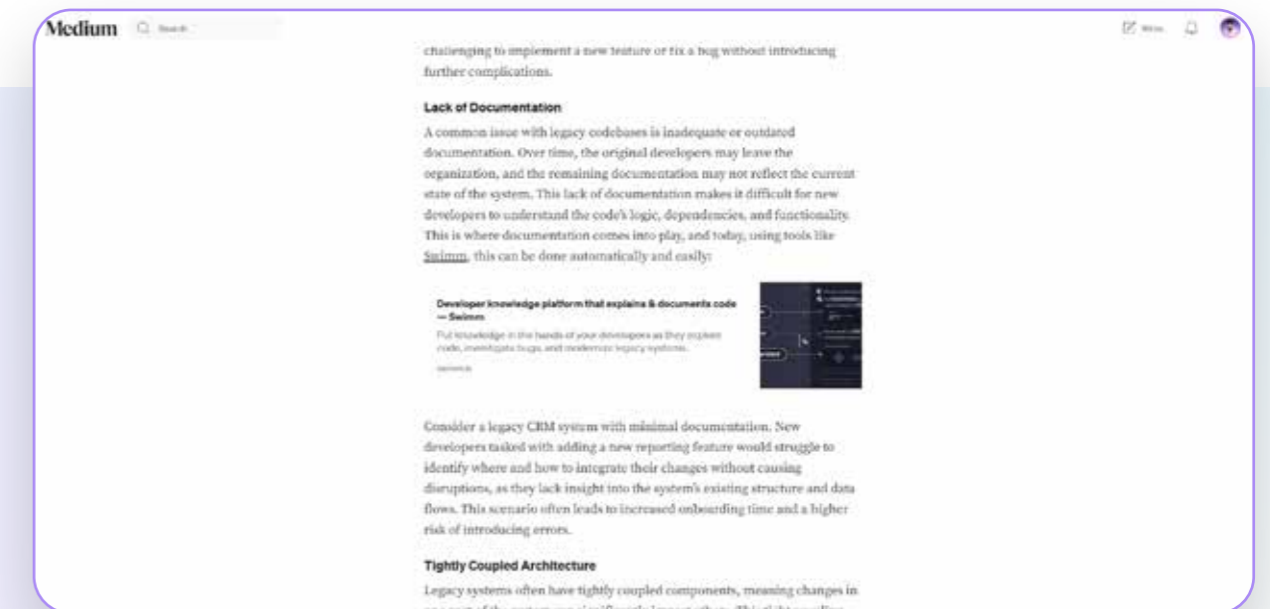
[Go to source](#)

### Performance Degradation

Organizations face performance issues with legacy applications, leading to up to 50% degradation in efficiency.

### Talent Shortage in Legacy Technologies

An estimated 40% of COBOL programmers are expected to retire in the next decade, exacerbating the talent gap.



[Go to source](#)

# Features

- **Gen AI-Driven Code Analysis**

Optimizes legacy code using advanced generative AI

- **AI-Powered Documentation**

Automatically generates accurate documentation

- **Flow Diagram Generation**

Creates visual flow diagrams for system processes

- **Automated Test Case Creation**

Develops deployable test cases for functionality validation

- **Cloud-Native Code Conversion**

Transforms legacy code to modern cloud-native architecture

- **Secure AI Models**

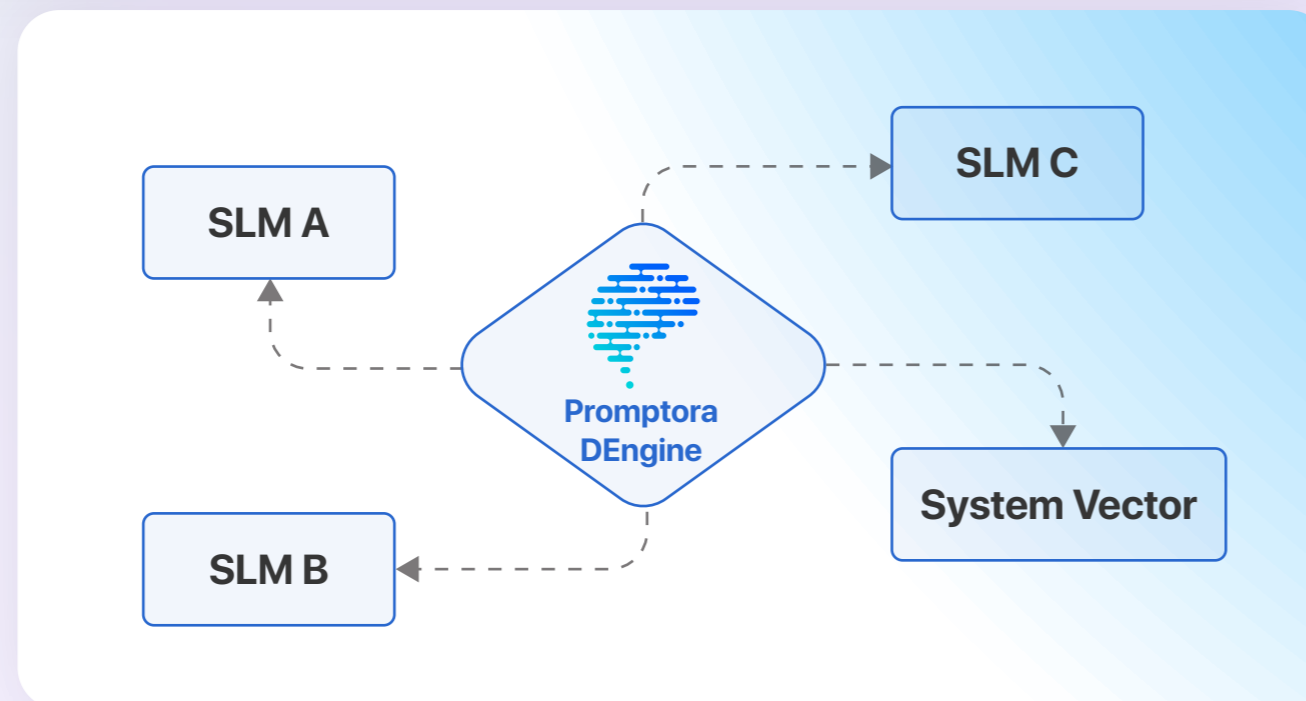
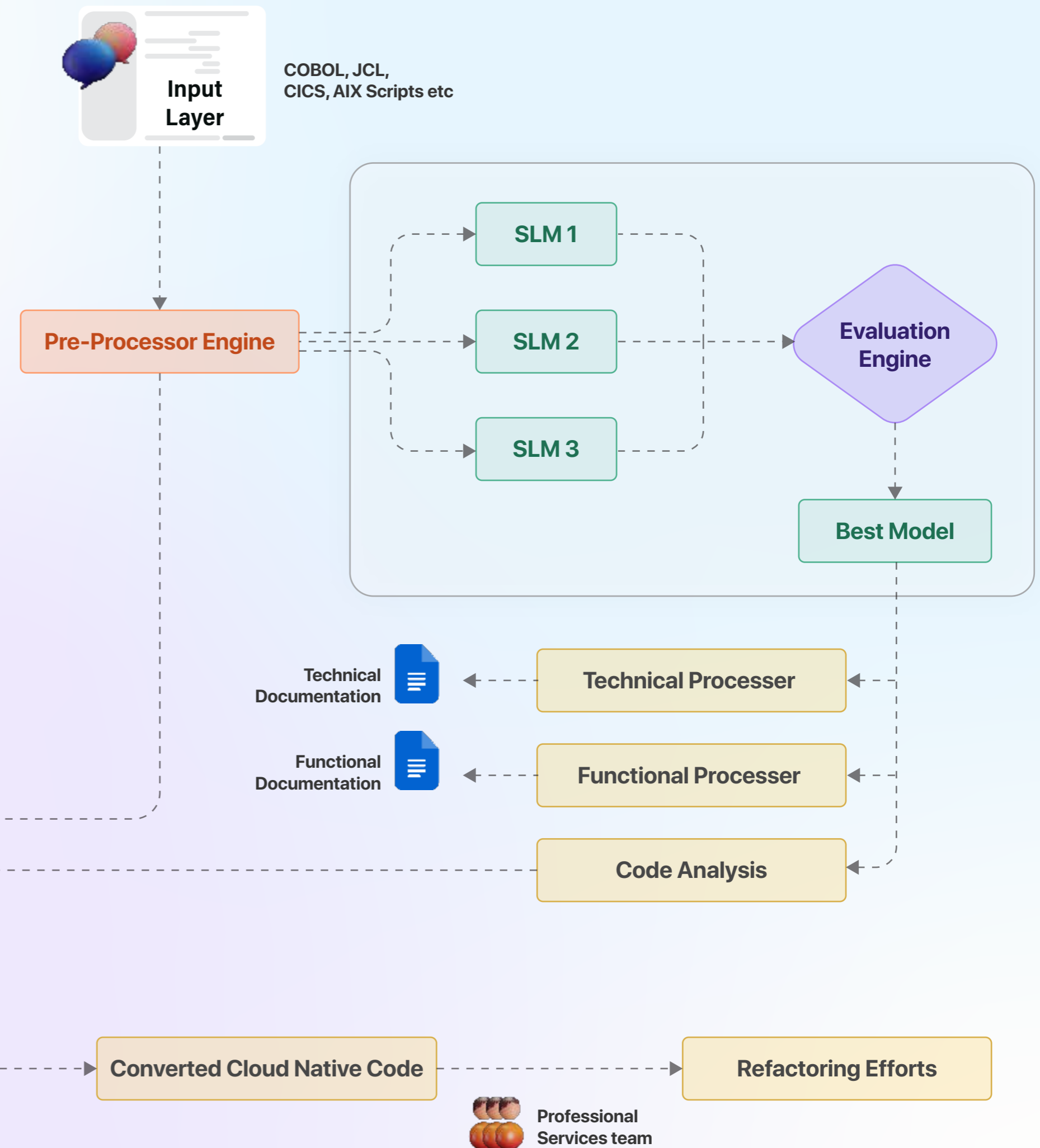
Utilizes secure, containerized Gen AI for data protection

- **Intuitive User Interface**

Simplifies the code upload and transformation process

- **Seamless Integration**

Integrates easily with existing systems and applications



# Benefits

50%

## AI-Powered Documentation

lower pricing compared to traditional market tools, reducing overall modernization expenses

23%

## Faster go-to-market time

by automating code modernization processes.

61%

## Improve documentation quality and speed

ensuring comprehensive and accurate project records

Feature	Our IP	Traditional Market Tools
<b>Documentation Generation</b>	Generates comprehensive documentation automatically, ensuring accuracy and reducing manual effort.	Documentation is often outdated or incomplete, requiring significant manual input from developers
<b>Code Optimization</b>	Automatically refactors and optimizes legacy code using generative AI, enhancing performance and maintainability	Manual refactoring is often required, leading to higher chances of errors and inefficiencies
<b>Adaptability</b>	Adapts to diverse datasets and learns patterns for flexibility in code generation and modernization	Rigid and rule-based, requiring frequent updates to accommodate new requirements or technologies
<b>Testing Automation</b>	Automates test case generation and maintenance, ensuring reliability with less manual intervention	Testing processes are typically manual and time-consuming, increasing the risk of undetected issues
<b>Integration Capabilities</b>	Seamlessly integrates with existing systems, facilitating smoother transitions to modern architectures	Integration can be cumbersome and often requires extensive manual coding efforts
<b>Security Enhancements</b>	Utilizes AI to identify vulnerabilities and suggest secure coding practices proactively	Security measures are often reactive, addressing issues only after they arise due to outdated practices
<b>Cost Efficiency</b>	Reduces overall modernization costs by automating processes and minimizing the need for extensive developer resources	Higher costs associated with manual processes and prolonged project timelines due to inefficiencies