Azul Intelligence Cloud

Code Inventory

Scope AI and modernization efforts to code that actually runs

AI Coding Tools Run Blind Against Unused and Dead Code

AI mandates are intensifying, but AI codegen and modernization tools run blind against legacy Java estates, generating new code on top of — and refactoring — code that never runs. An estimated 20–50% of enterprise Java code is unused or dead*, overloading AI coding tool context windows and shrinking reasoning headroom. The result is an increase in hallucinations, wasted token spend, and more code rework while the DevOps teams asked to lead AI initiatives remain buried under the burden of maintaining, testing, and patching code no one uses.

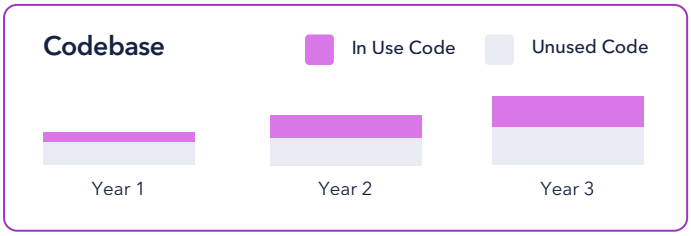
Why Unused & Dead Code Matters

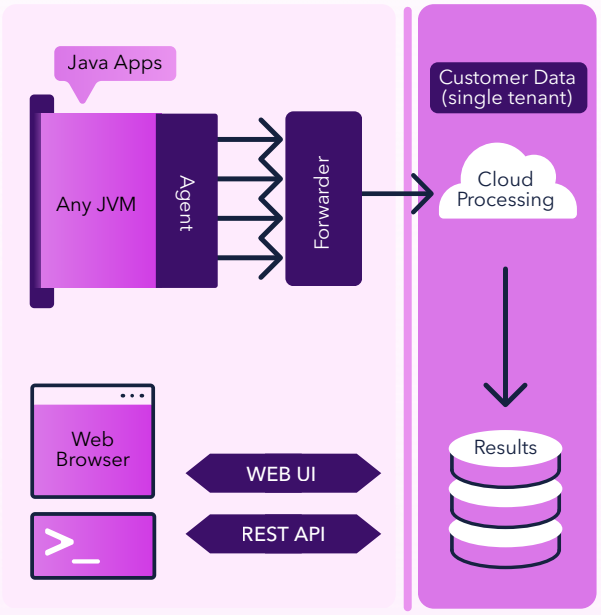
Unused and dead code bloats an application's codebase but is never executed. Removing it shrinks the target for every AI codegen, modernization, and vulnerability remediation effort while reducing the burden of maintaining the codebase. Static analysis tools and IDEs can clean up only ~20% of unused and dead code. Code Inventory finds the other 80% by automatically identifying and flagging code that never executes in production so it can be safely removed.

The Proven Impact

A study from Goldman Sachs' DevOps organization underscores the importance of deleting unused and dead code by revealing that they:

- Reduced the size of a codebase by 67% for a recent project.
- Improved product release cadence to more than 250 releases per year.
- Realized time savings that afforded new opportunities as a result of their reduced codebase size and greater testing confidence.





1. Applications run on top of any JVM enabled with Code Inventory.
2. Data goes through the Forwarder to a point of collection.
3. Each customer's tenant collects and aggregates the data, marking the first-seen time of each method to provide visibility across runs.
4. Code Inventory's GUI or REST API returns results that can be used to optimize AI codegen and modernization efforts.

Code Inventory Boosts DevOps Capacity for AI

A feature of Azul Intelligence Cloud, Code Inventory is the only solution that uses live production runtime data to precisely catalog the code actually used by Java applications, giving AI codegen, modernization, and vulnerability remediation tools a leaner, more accurate target to work against instead of the entire codebase. By slashing the burden of maintaining and testing unused code, enterprises can reclaim developer capacity for AI initiatives.

Benefits of Code Inventory

Accelerate AI Modernization Projects	Identifies dead and unused code down to the class and method level – an accurate signal that can be used to confidently leave behind technical debt and shrink refactoring efforts
Reclaim DevOps Capacity for AI	Eliminates the need to maintain, upgrade, test, or investigate vulnerabilities in code that never executes, freeing up resources to focus on AI initiatives
Cut AI Token Spend and Sharpen AI Output	Feed AI tools only the code that actually runs to decrease token spend and maximize context window reasoning headroom resulting in fewer hallucinations and less code rework
Scope Security Remediation to Real Exposure	Addresses AI-era threat velocity by providing instant information about whether a vulnerable class or method runs in production, eliminating the need for investigation
Safely Remove Dead & Unused Code Automatically	Leverage AI tools like Moderne or OpenRewrite recipes to flag, deprecate and then safely execute large-scale code changes to shrink large codebases
Manage Risk More Effectively	An English language AI assistant provides natural language querying of application code, enabling DevOps to forego reporting in favor of asking the assistant directly which code is dead and unused, whether an edge case actually occurred, whether vulnerable code ran, etc
No Performance Penalty; Full Production History	Collects data from inside the JVM without the performance penalty of traditional profilers and retains a history of when code was first and last run, down to the method level
AI Assistant	Rather than generating, updating and investigating reports, you can ask natural-language queries about live runtime data to pinpoint where dead and unused code lies, such as whether a class or method has been deprecated, whether a feature is still in use, what a 3rd-party library is used for, and more

Learn more about Azul Code Inventory

Copyright© 2026 Azul Systems, Inc. All rights reserved. Azul Systems, the Azul Systems logo, Zulu and Zing are registered trademarks of Azul Systems Inc. Java and OpenJDK are trademarks of Oracle Corporation and/or its affiliated companies in the United States and other countries. Other marks are the property of their respective owners and are used here only for identification purposes.

azul | www.azul.com | +1.650.230.6500

385 Moffett Park Drive, Ste 115
Sunnyvale, CA 94089 USA