

Code Inventory

Accurately identify unused and dead code for removal to save time and money

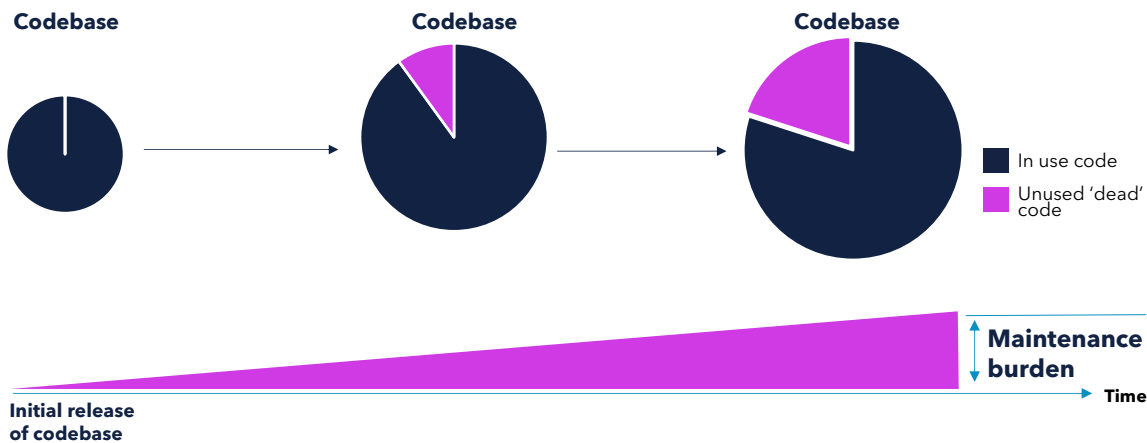


Unremoved Unused and Dead Code Increases Maintenance Burden

Unused and dead code resides in an application’s codebase but is not used by the application. Removing unused and dead code significantly reduces the burden of maintaining the codebase and makes it easier to understand. Developers want to remove unused and dead code but are terrified of doing so because the process of identifying and removing it is manually intensive and can break the application. Traditional tools like IDEs do not provide a complete view of code used in production over time due to their static nature. Profilers are so overhead intensive they cannot practically be used in production settings.

Why It Matters

Unused and dead code increases the cognitive burden for developers to understand the code, makes onboarding new developers difficult, and makes it difficult to update libraries and upgrade from one Java version to the next. Unused and dead code impacts overall developer productivity and slows down velocity of delivering new features. If unaddressed, unused and dead code becomes harder to deprecate and can lead to zombie code that reawakens with unintended consequences.



The Importance of Deleting Unused and Dead Code

A [recent 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
- Reduced codebase size and greater confidence in their testing resulted in time savings and afforded opportunity for other investments

Proof

“Without a timely, complete and trustworthy code use and impact report, there’s a danger that the step of deprecating code can cause bigger problems,” said David Norfolk, Practice Leader, Bloor Research.

“Reliably identifying dead code is the hard part and using automation to find what code is and is not running, or can’t ever be reached, gives developers a huge advantage.”

Code Inventory

Accurately identify unused and dead code for removal to save time and money



Azul Intelligence Cloud

Product Features

A feature of Azul Intelligence Cloud, Code Inventory is the only solution that precisely catalogs the source code actually used in production by Java applications, making it easy to accurately identify unused and dead code for removal. It slashes the burden of maintaining and testing code that is not being used, significantly improving developer productivity and ultimately saving time and money.

- Aggregates what code runs over time
- Strongest, most accurate signal
- Collects information inside the JVM
- Runs in Production, no performance penalty
- Provides a comprehensive view across an enterprise’s Java workloads
- Identifies code down to the class/package and method level

Benefits

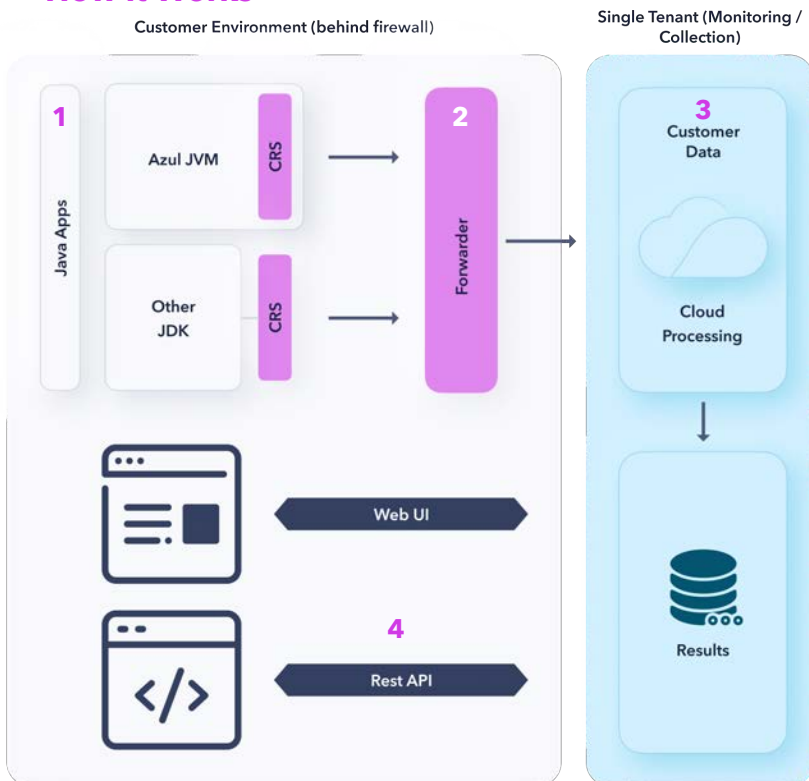
Significant Reduction in Time Spent Automatically finds what code runs, dramatically slashing the time to identify dead code for removal by eliminating the time-intensive manual process of locating and validating dead code.

Remove Unused and Dead Code with Confidence Provides a comprehensive view across an enterprise’s Java workloads of what code runs over time down to the class/package and method level, providing a more accurate signal to confidently remove unused and dead code and reduce the risk of breaking the application.

No Performance Penalty Collects and aggregates detailed code information from inside the JVM without the performance penalty of traditional profilers, and provides production context IDEs lack.

Summarizes and Retains History Collects data across all of an enterprise’s Java workloads and consolidates it to provide a history of when code was first and last run, down to the method level.

How it Works



- 1 The application runs on top of Azul and other JVMs, 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 and last-seen date and time of each method to provide visibility across runs.
- 4 Application owners can call Code Inventory for the results via the REST API and compare against their code to plan removal

Learn more at

www.azul.com/components/code-inventory

Contact Azul

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

sales@azul.com

+1.650.230.6515

www.azul.com