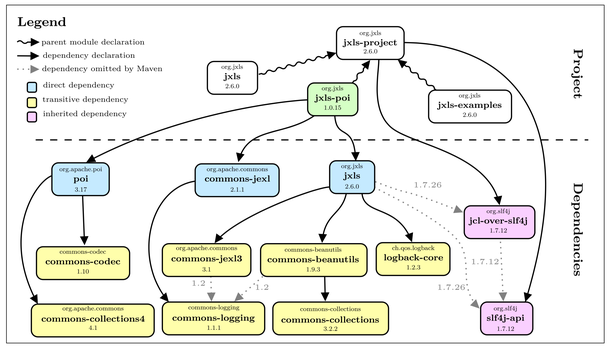
Automatic Software Debloating

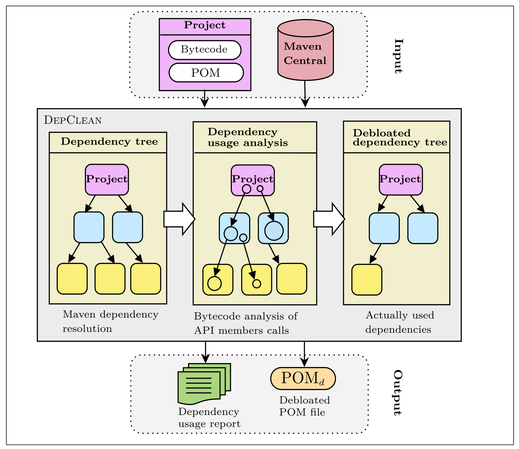
César Soto-Valero [cesarsv@kth.se]
Proudly supervised by Thomas Durieux, Martin Monperrus, and Benoit Baudry
Thankfully funded by WASP

Debloating Java Dependencies (Statically)

"Developers declare software dependencies that they actually do not use in their projects."

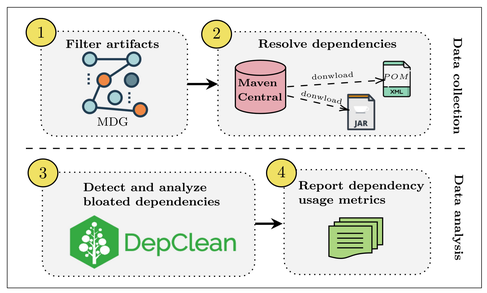


Methodology & Tool

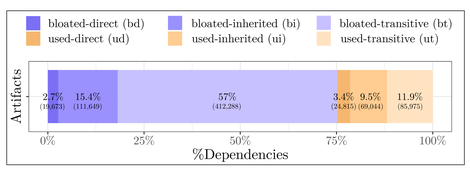


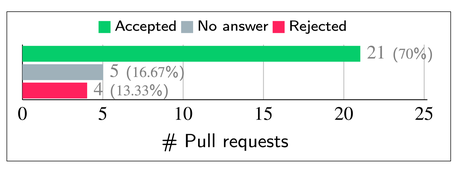
https://github.com/castor-software/depclean

DepClean automatically detects and removes bloated dependencies in Java projects through static analysis.



Results



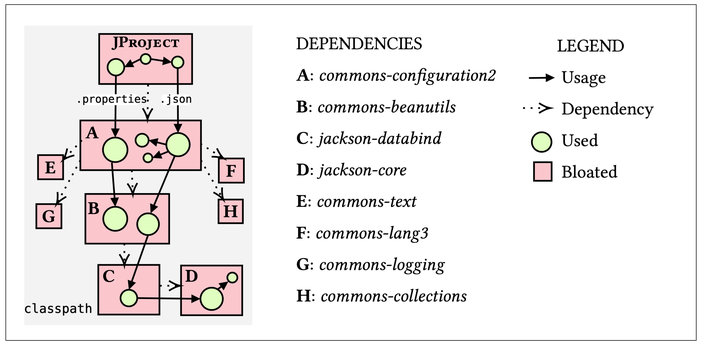


- 75% of the dependencies in the Maven Central repository are bloated dependencies.
- 21/26 of answered pull requests have been merged, removing 140 bloated dependencies in total.

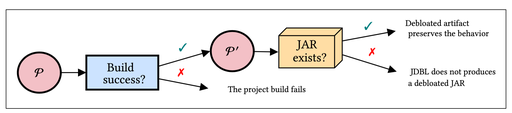
A Comprehensive Study of Bloated Dependencies in the Maven Ecosystem. Springer Empirical Software Engineering.

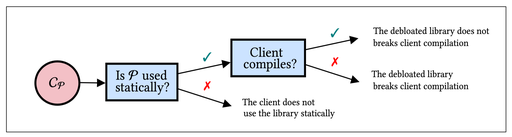
Debloating Java Bytecode (Dynamically)

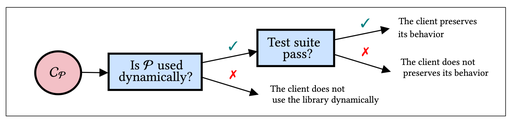
"The existence of disjoint execution paths makes Java projects susceptible to include unnecessary functionalities."



Methodology & Tool

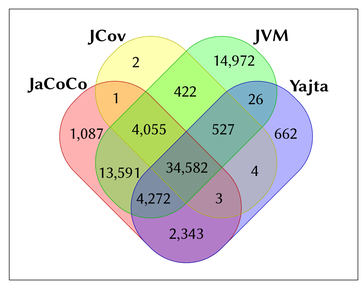




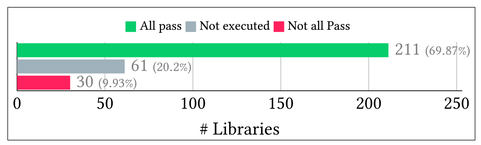


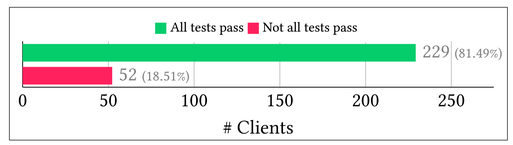
https://github.com/castor-software/jdbl

JDBL automatically removes unnecessary bytecode from Java projects using coverage tools and dynamic analysis.



Results





- 68.3% of the libraries' bytecode and 20.3 % of their total dependencies can be automatically debloated.
- 81.5% of the clients preserve their behavior when the original library is replaced by its debloated version.

Coverage-Based Debloating for Java Bytecode. ACM Transactions on Software Engineering and Methodology.