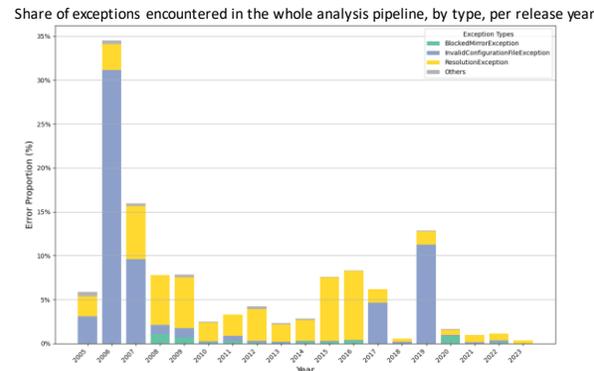
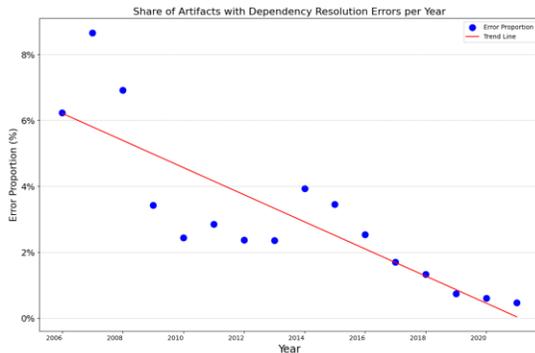
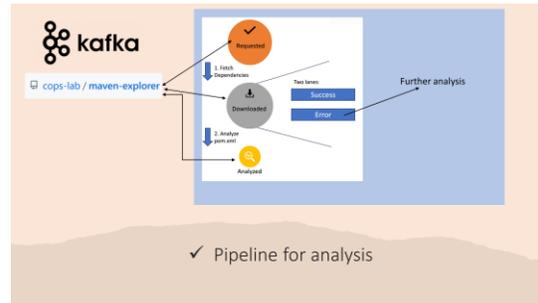
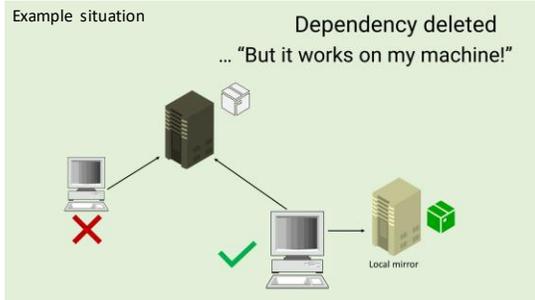
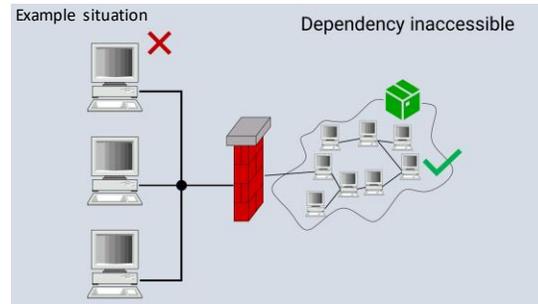


Research Questions

1. Age vs Number of Errors
2. Most Common Errors
3. Strategies for Circumventing Errors



Introduction

The study explores the impact of aging on projects in Maven Central on dependency resolution. Maven Central preserves all versions of artifacts, but dependencies often exist in alternative repositories that may not enforce this policy, or as individual files which may be unavailable in certain environments or disappear. Repositories are sometimes taken offline as well, such as in the case of JCenter [1]. We estimate the proportion of artifacts that are no longer resolvable, and whether this is more likely to happen with older artifacts.

Methods

The study employs a quantitative approach, using the Maven Explorer [2] indexer to assess the number of artifacts in Maven Central with missing dependencies. The methodology involves indexing artifacts, checking for insecure HTTP-based repositories, and gathering information about the original dependencies causing resolution failures. We built an extension to Maven Explorer that structures data collected from Maven Central using Maven Explorer into a relational database, recording artifact creation dates, error messages, and other metadata, and then analyzing this data using statistical methods. We encourage the program to be reused in future research, publishing it under an MIT license: [3].

Results

The research found that older dependencies are more likely to be unresolvable: notice a decline from around 6% of artifacts published in 2006 to less than 1% of artifacts of 2021. We estimate that around 0.3% of artifacts might fail to download due to reliance on non-HTTPS repositories, however, recommending to re-enable only those repositories that are strictly necessary for each individual project. The study examines common exceptions in the Maven Explorer pipeline and identified popular causes encountered per artifact release year. A detailed analysis of these dependencies showed specific missing artifacts that are useful to be researched.

Conclusion

The study concludes that older packages in Maven Central are more prone to dependency resolution failures, often due to the absence of Transport Layer Security in repositories. The research recommends that developers and maintainers of popular open-source projects upload new versions of their libraries to multiple repositories to mitigate these issues. The findings highlight the importance of better dependency management practices and active maintenance for enhancing the longevity of software projects in the Maven ecosystem. To an extent, this study's insights extend to other package management systems, such as Debian's APT (requires GPG signatures, as opposed to Maven), NuGet (here repositories without TLS cause a warning).

References:

- [1] S. Greene, "JCenter Shutdown Impact on Gradle Builds," Gradle, Feb. 2021. [Online]. Available: <https://blog.gradle.org/jcenter-shutdown>.
- [2] cops lab, "Maven Explorer," GitHub, Jan 2024 [Online, accessed 28 Jan 2024]. Available: <https://github.com/cops-lab/maven-explorer>.
- [3] G. Kulikovskis, "Maven Error Stats", Jan 2024 [Online, accessed 28 Jan 2024]. Available: <https://gitlab.com/gintsk/maven-error-stats>.