Towards Reliable Dependency Management

Cathrine Paulsen, C.R.Paulsen@tudelft.nl
Supervised by Sebastian Proksch, Arie van Deursen, Fernando Kuijpers

What's the problem with dependency management?

Developers rely on dependency managers to resolve dependency sets but they are not guaranteed to be **up-to-date**, **compatible**, nor **secure**.

- Keeping dependencies **up-to-date** is crucial to receive bug fixes and security patches
- Open version ranges automate updates but risk **breaking changes** [1]
- Low Semantic Versioning compliance encourages update aversion among developers [2-3]
- Pinned versions risk version conflicts, and propagate stale and **vulnerable** dependencies [4]

Dependency managers *cannot* ***reason*** *about all these properties so ensuring them is often a labor-intensive task left to the developer…* ***But what if they could?***

A new approach to dependency management

How can we provide and maintain dependencies that are ***up-to-date***, ***compatible***, *and* ***secure****?*

- A **dependency metadata store** to better inform constraints for the resolution process
  - Enable reasoning over compatibility, vulnerabilities, network resources, performance requirements, …
- A **constraint-based resolver** that can extract useful constraints and optimize the resolution result

Proof-of-concept: Compatible Version Ranges in Maven with MaRCo

Paper under submission: Reliable Dependency Resolution Using Client-Agnostic Compatibility Detection

~73% of projects at risk of future dependency-related issues

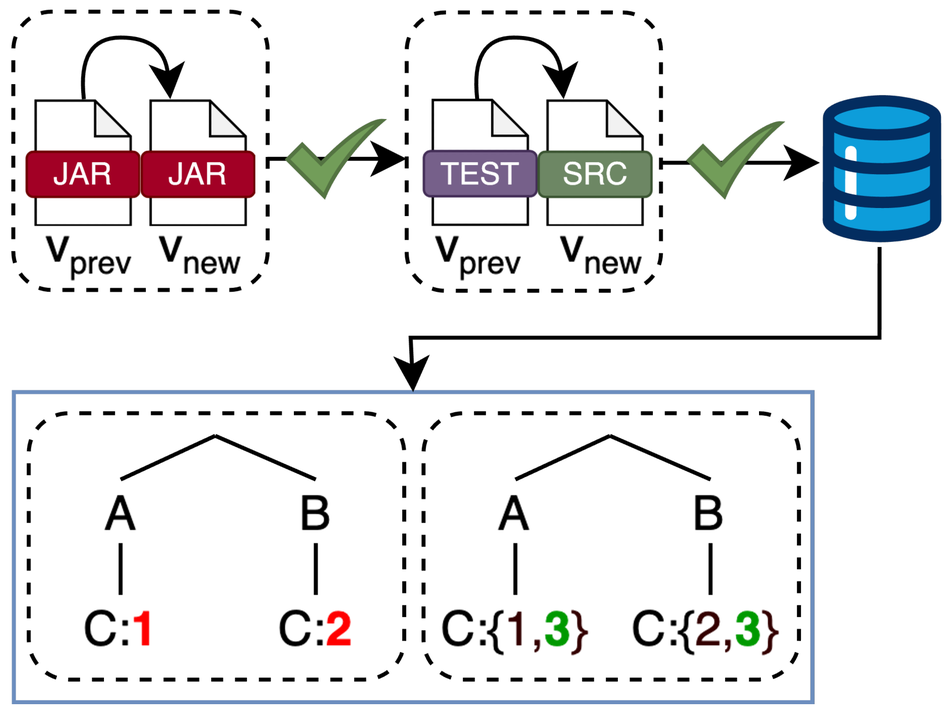
- 73% use transitive dependencies directly
- 66% have conflicting pinned versions
  - 67% resolve conflicts manually

We built MaRCo to mitigate these issues

- Injects missing direct dependencies in 91% of projects
- Replaces pinned versions with compatible ranges in 79% of projects
  - Replaces 13% of dependencies on average

Client-agnostic compatibility allows reuse across clients

- Stable but strict ranges (0.99 breaking vs 0.75 non-breaking recall)
- Limitation: Inability to find runnable test suites for many libraries



MaRCo *serves as a* ***proof-of-concept*** *showcasing how we can* ***extract better constraints*** *from pre-computed data to* ***ensure compatible dependency resolution*** *in a* ***scalable*** *way.*

Future work

- **Fine-grained** compatibility data via **test generation**
- **Filtering** of constraints on **client usage**
- Feasibility of **constraint-based resolver** with **different constraint types**

References

- [1] Dietrich et al. "Dependency versioning in the wild." *MSR, 2019.*
- [2] Raemaekers et al. "Semantic versioning and impact of breaking changes in the Maven repository," Journal of Systems and Software, 2017.
- [3] Pashchenko et al. "A qualitative study of dependency management and its security implications," CCS, 2020.
- [4] Zhang et al. "Mitigating persistence of open-source vulnerabilities in Maven ecosystem," ASE, 2023.