A Contextualised Risk-Based Framework for Maven Dependency Updates

David Rodrigues Almeida · Isabel Azevedo
ISEP – Instituto Superior de Engenharia do Porto



2026 International Symposium On Ambient Intelligence and Embedded Systems (AmiEs 2026)



23-26 September 2026



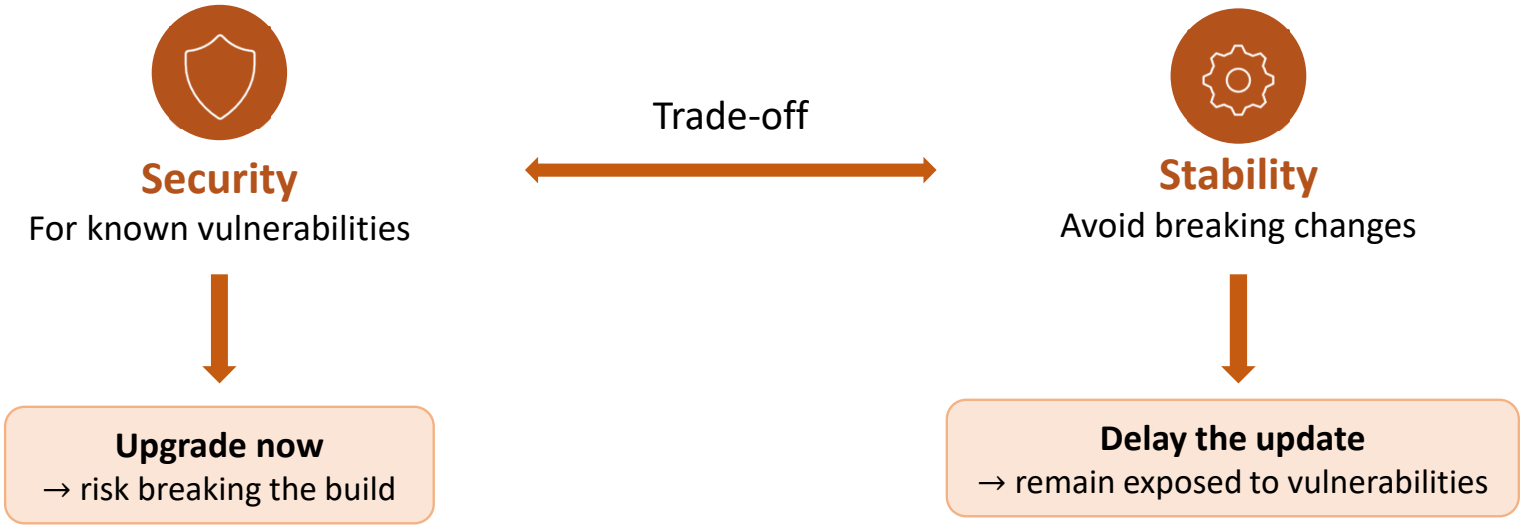
Porto, Portugal

Overview


01
Introduction
02
Design
03
Solution
04
Evaluation
05
Conclusions

Motivation / Problem

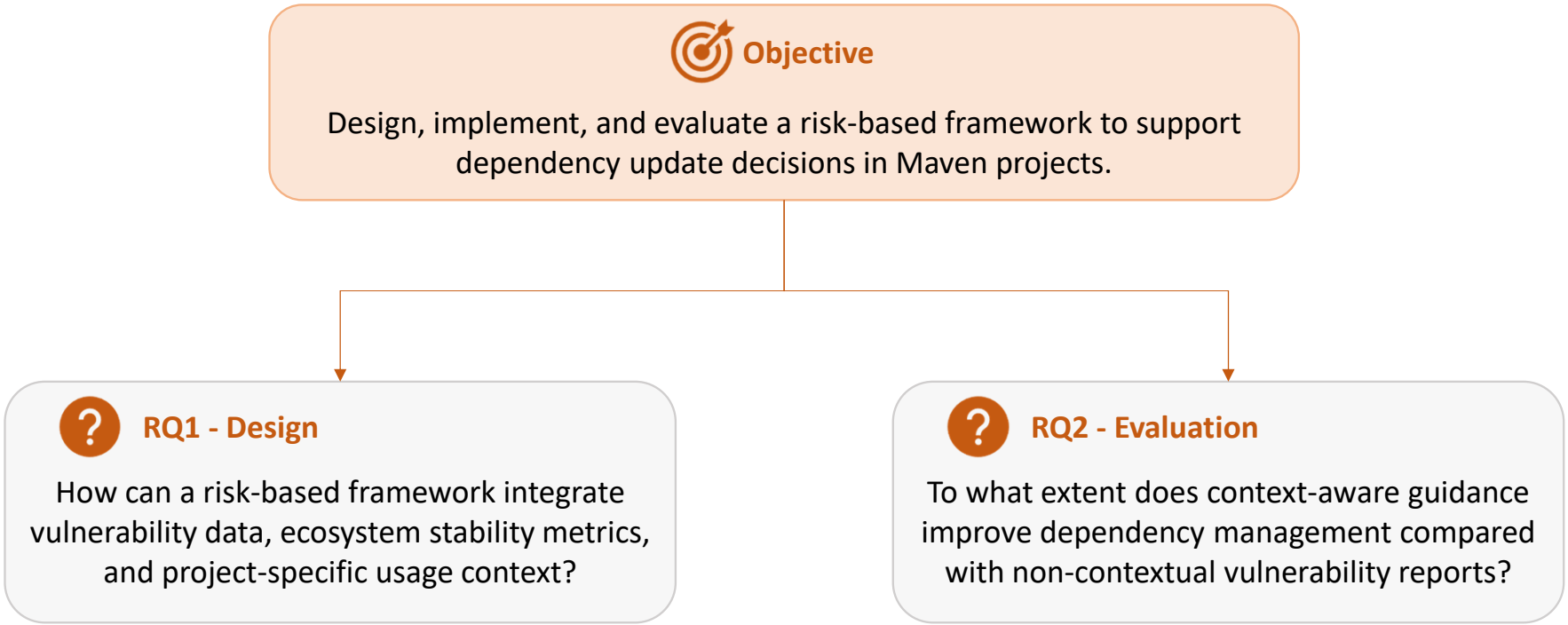
Dependency updates force a security-stability trade-off.



Current vulnerability scanners lack the context needed to support this decision.

- They detect known CVEs but do not determine if the vulnerable code is actually reachable;
- They do not indicate whether the fix version is stable in the wider ecosystem.

Research Questions & Objective



Related Work

Reachability-based tools

- Reduce irrelevant vulnerability alerts
- Check whether vulnerable code is actually used

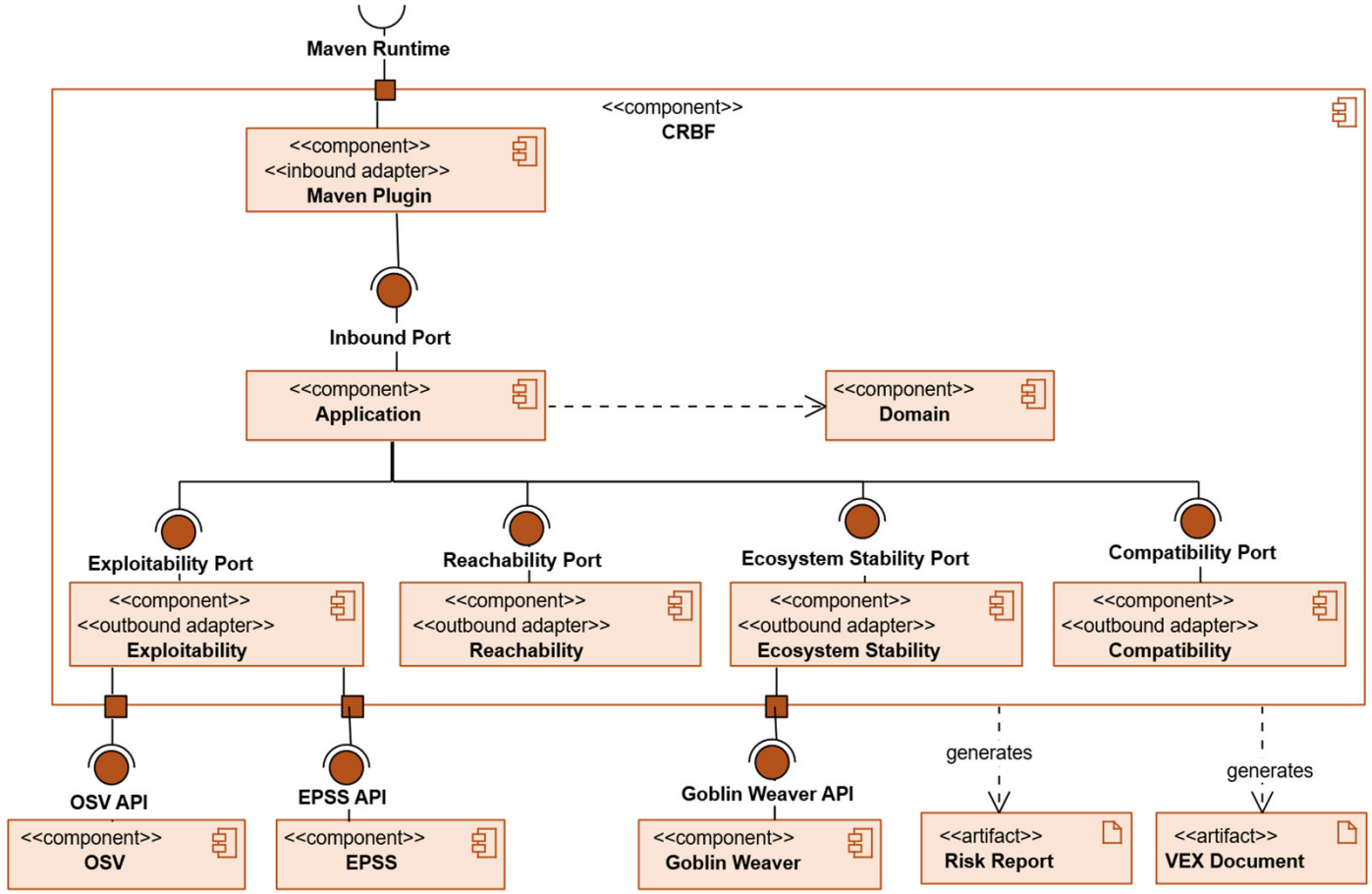
Risk / update decision approaches

- Combine severity, effort, or dependency constraints
- Some consider compatibility or ecosystem adoption

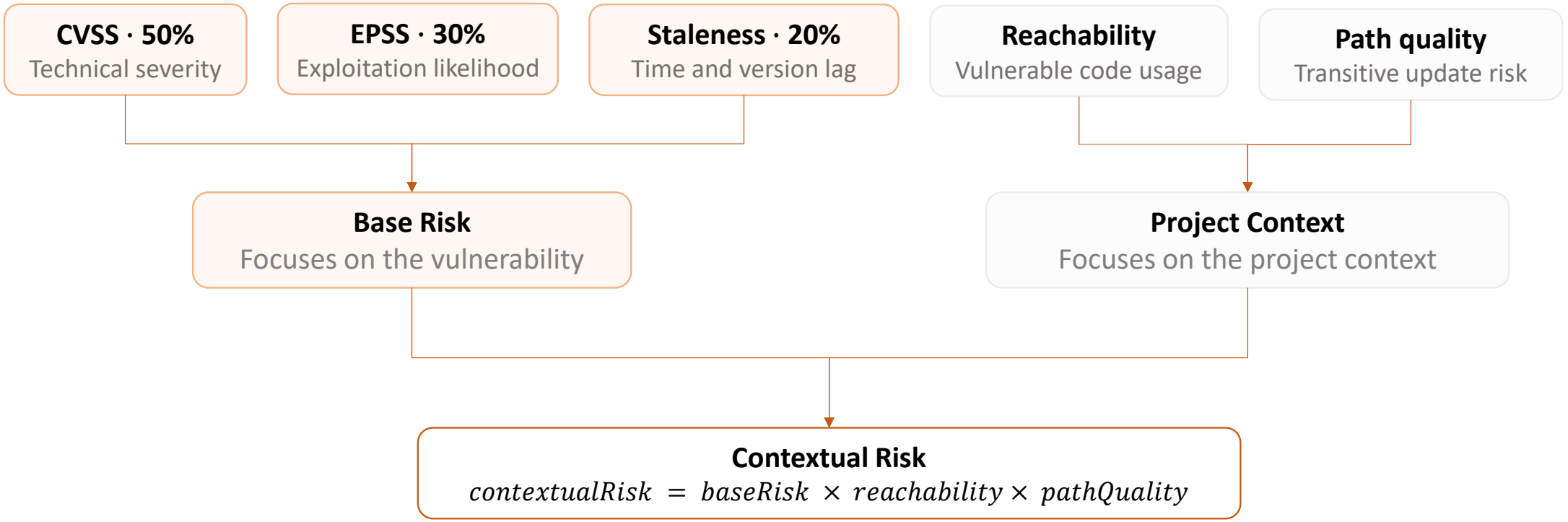
Lack of an integrated approach combining all four signals for dependency update decisions.

Exploitability | Reachability | Ecosystem Stability | Compatibility

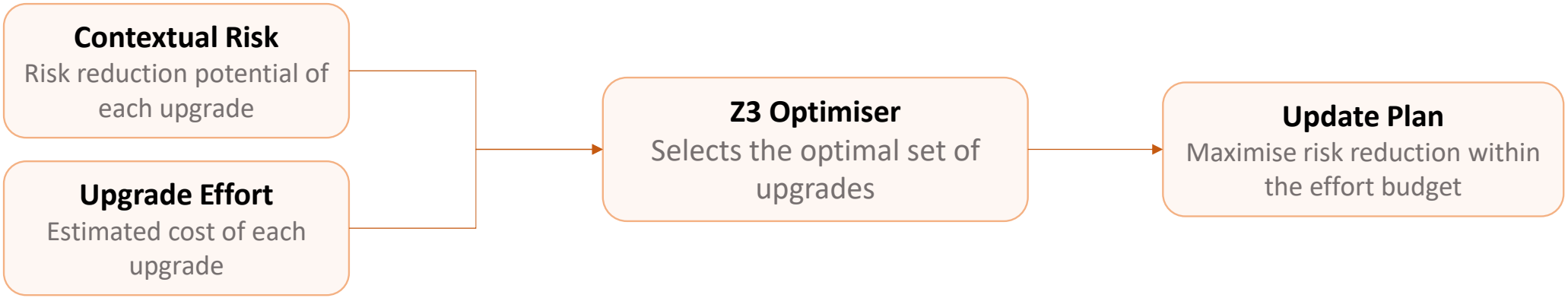
Framework Overview



Risk Scoring Model



Remediation Optimisation



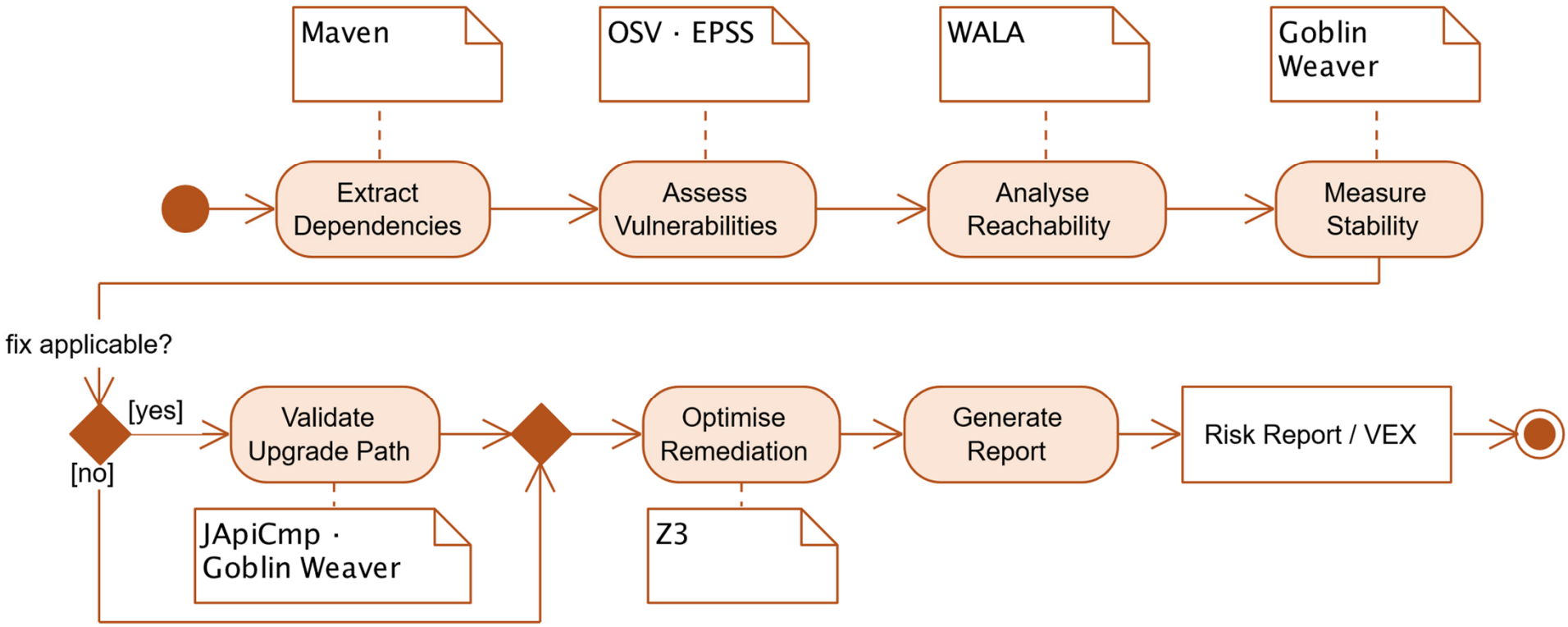
Optimisation Objective

$$Maximise \sum_{i=1}^{n} (upgrade_i \times contextualRisk_i)$$

Constraints

C1 – Stay within the effort budget
C2 – Ensure a fix version is available
C3 – Reachable critical cases are mandatory upgrades

Implementation



Evaluation Design

Compare three configurations on the same set of open-source Maven projects

Configurations

- C1 – OSV-Scanner without call-graph analysis
- C2 – OSV-Scanner with call-graph analysis
- C3 – CRBF with context-aware recommendations

Dataset Selection

- Valid Maven Project
- Executable JAR
- Builds without modification
- Fixed public commit

Status / Preliminary Results

Initial evaluation with three open-source Maven projects across all configurations

Project	**C1** OSV-Scanner	**C2** OSV-Scanner + Call Graph	**C3** CRBF
Spring-boot-security-jwt	Executed	Executed	Executed
my-blog	Executed	Executed	Executed
Websocket-demo	Executed	Executed	Executed

- All three configurations have been executed on the current pilot dataset.
- CRBF additionally provides contextual risk assessment and remediation recommendations;
- The evaluation, including the analysis of the results is currently ongoing.

Conclusions & Future Work

Contribution

CRBF combines:

- Vulnerability severity (CVSS);
- Exploitation likelihood (EPSS);
- Reachability (WALA);
- Project / update context (JApiCmp, Goblin Weaver, Z3)

To support Maven dependency update decisions.

Future Work

- Complete the evaluation on the full dataset;
- Analyse comparative results across C1, C2 and C3;
- Calibrate the scoring model based on evaluation results;

Thank You – questions?

David Rodrigues Almeida
1181210@isep.ipp.pt

Isabel Azevedo
ifp@isep.ipp.pt

Source Code
github.com/CRBF-Project/risk-based-framework