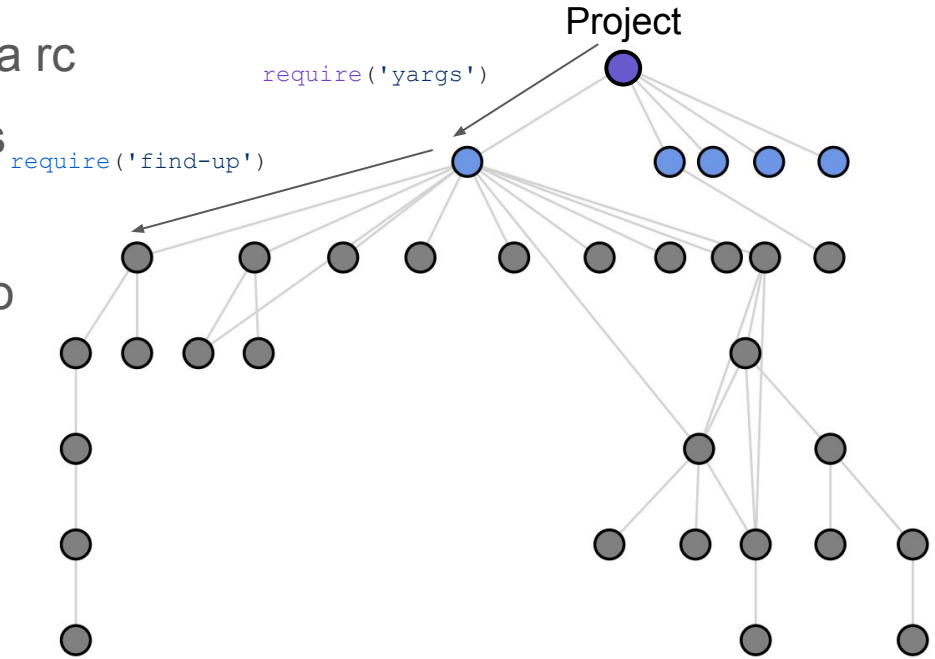
REM: Visualizing the Ripple Effect on Dependencies Using Metrics of Health

Zhe Chen, Daniel German
University of Victoria, Canada
September 2020

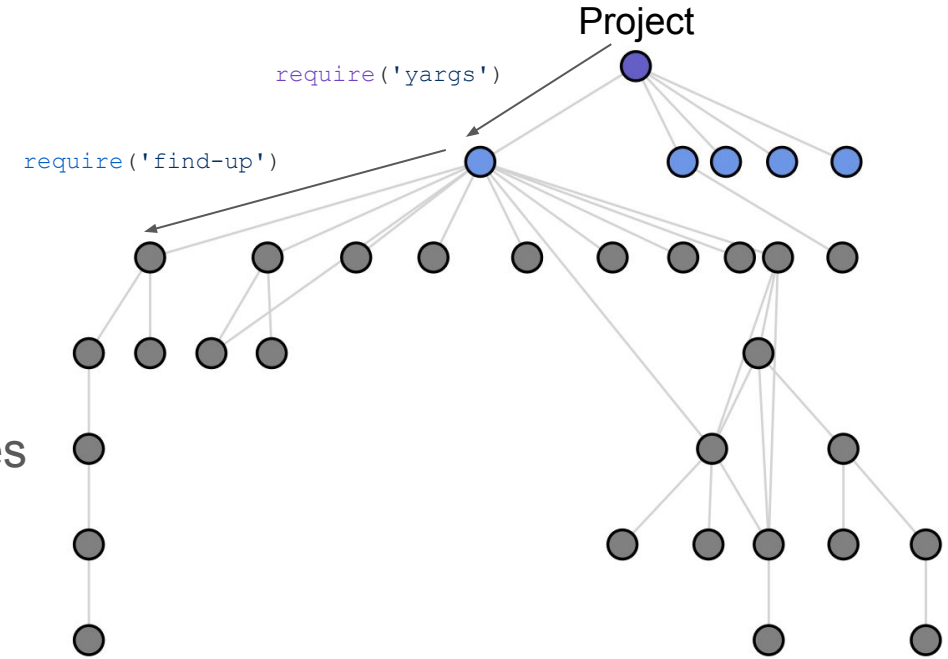
Scenario

- Build a project - configurable via rc
- Using external dependencies is common
- These dependencies could also rely on other dependencies



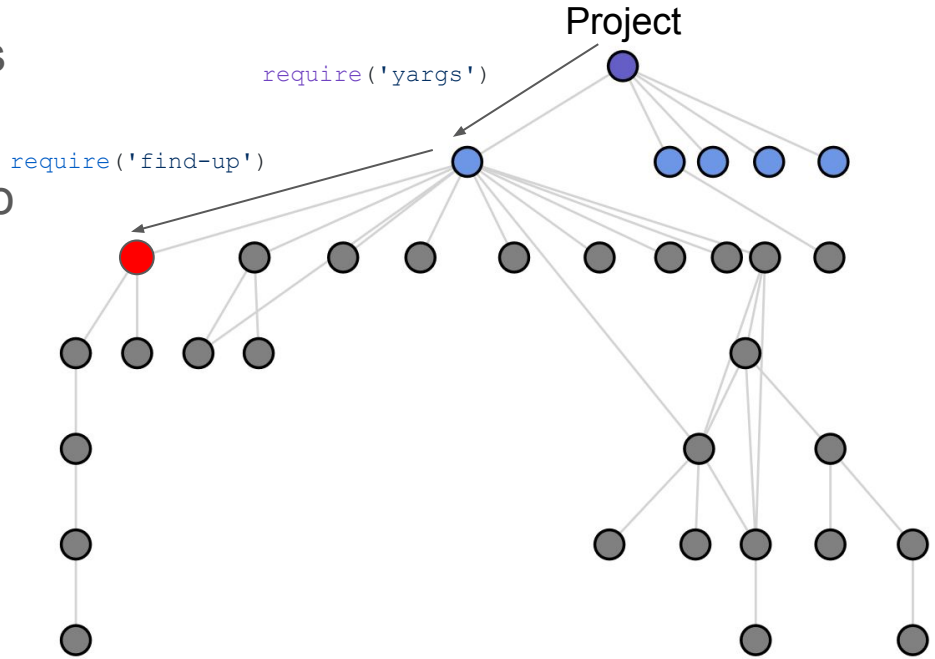
Scenario - Dependency Graph

- To describe software reuse
- Hierarchical layout -
  top requires bottom
- Direct dependency: software
  used by project
- Transitive dependency:
  software used by dependencies
  of project



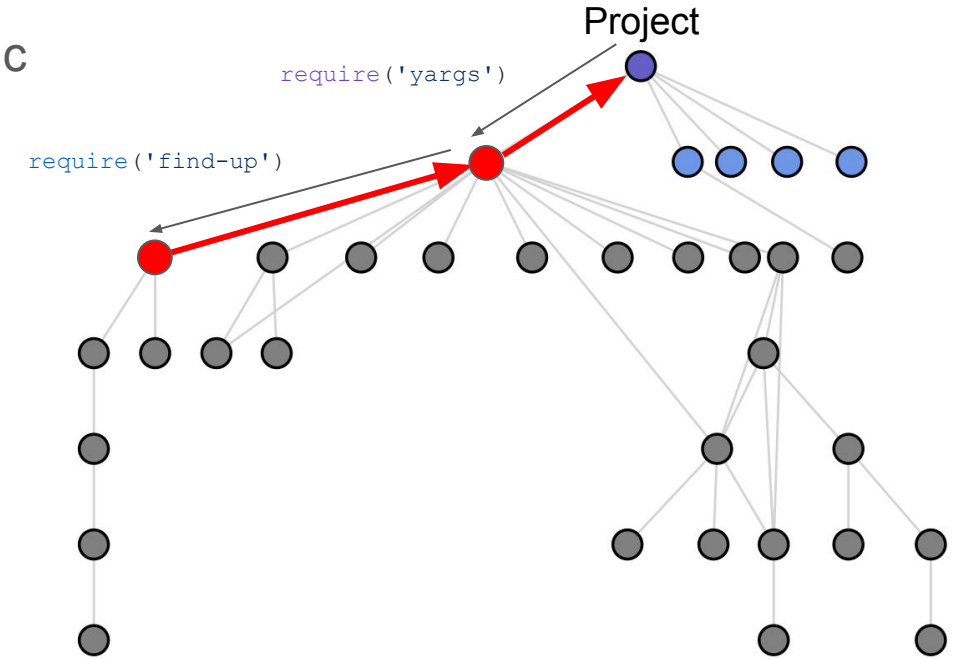
Scenario

- Using external dependencies is common
- These dependencies could also rely on other dependencies
- What if one of them is broken?



Scenario - Ripple Effect

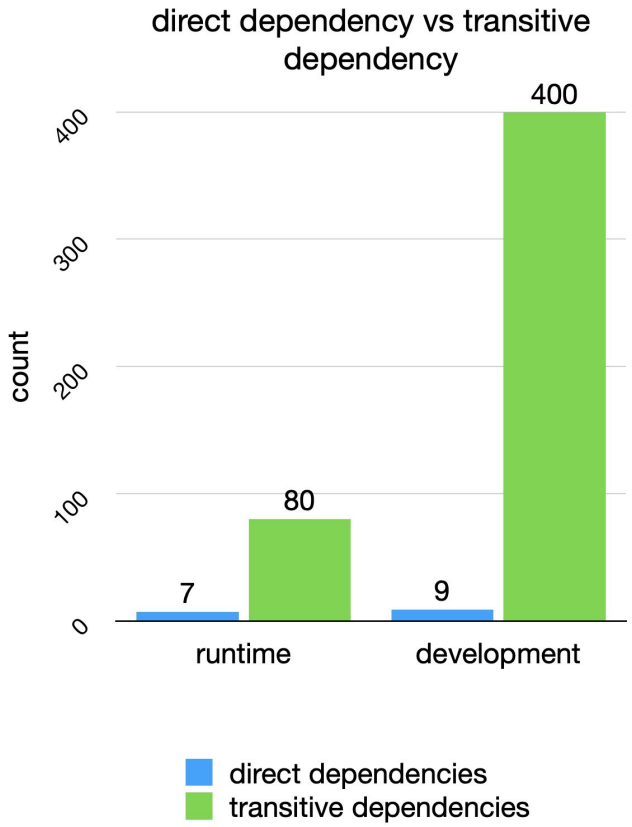
- Propagation from a problematic dependency to the software application
- Any dependency (direct or transitive) can break the software application



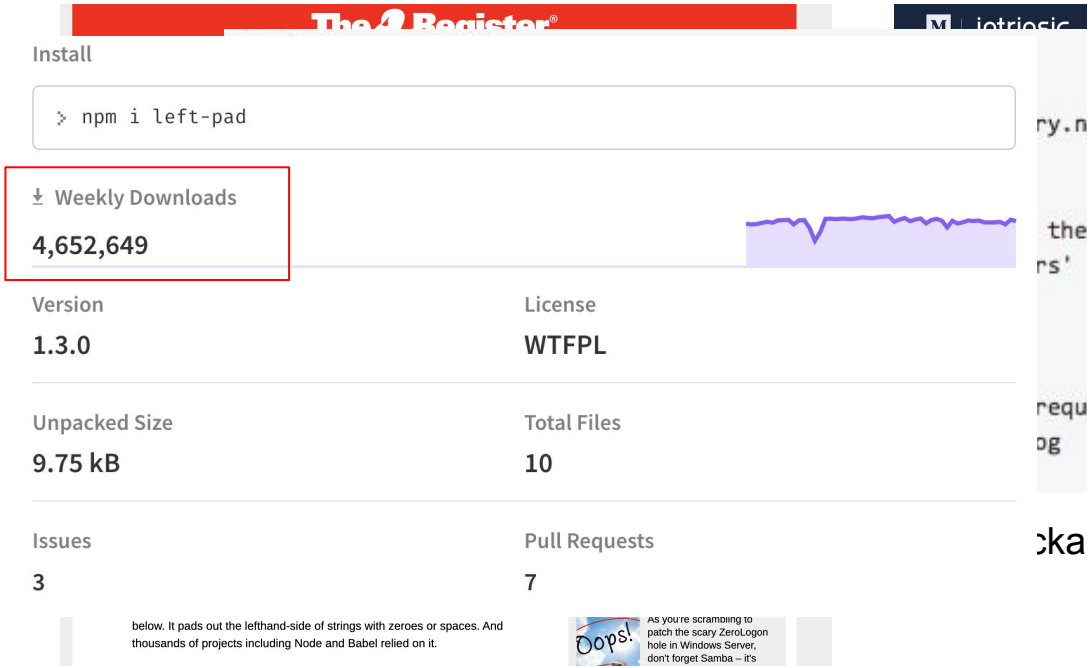
Transitive Dependency

- hidden from application developers
- hard to track

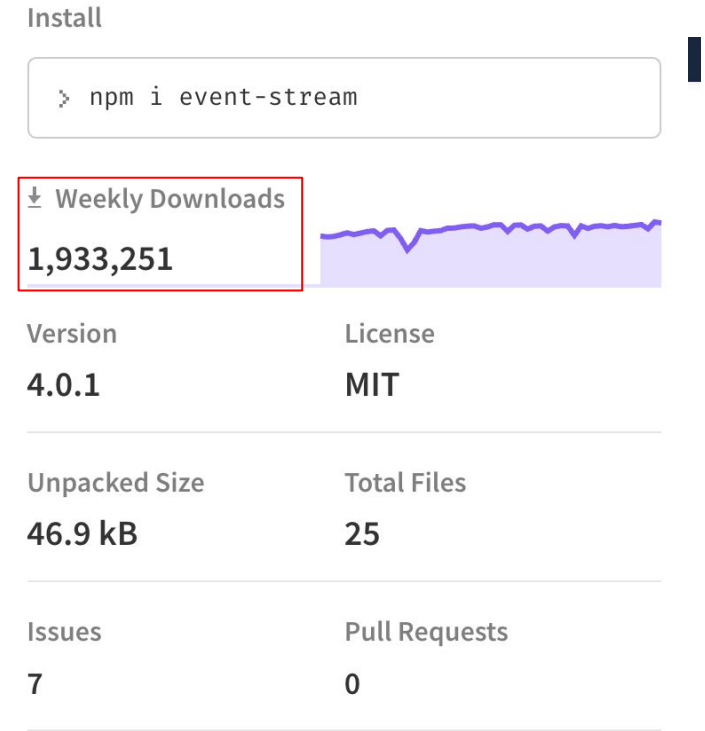
Collected dependency data on 50,000+ GitHub NPM applications: huge number of transitive dependencies could be hiding in applications with small number of direct dependencies



Failures in Transitive Dependencies



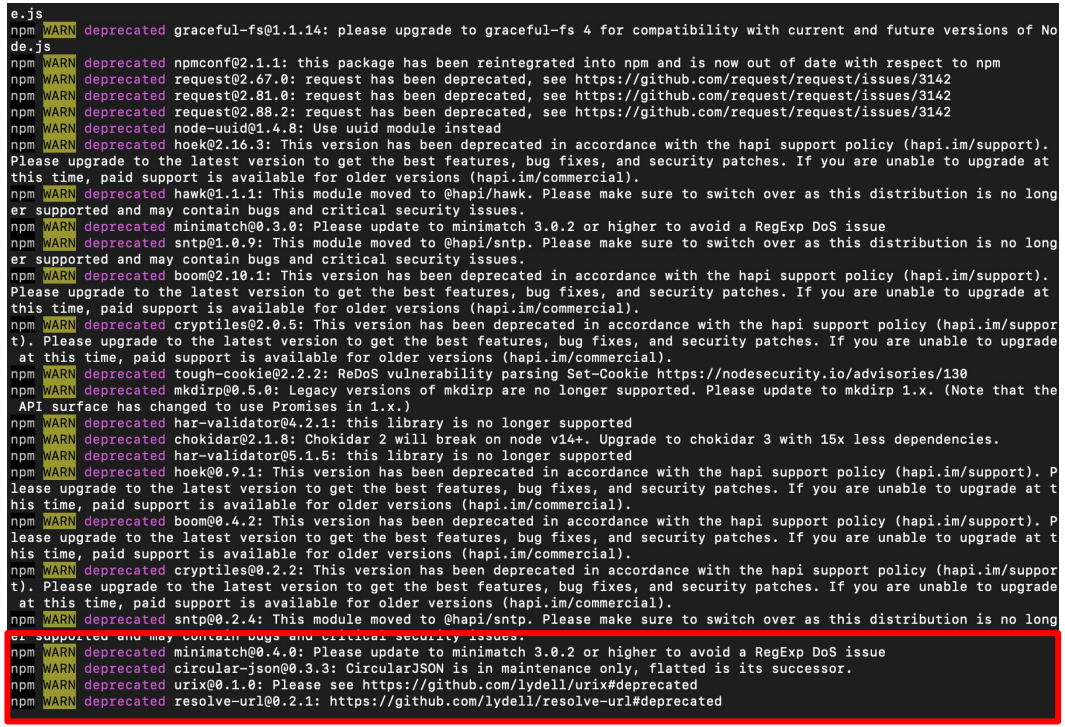
left-pad: removal of a package



event-stream: malicious code injected

How NPM shows Problem

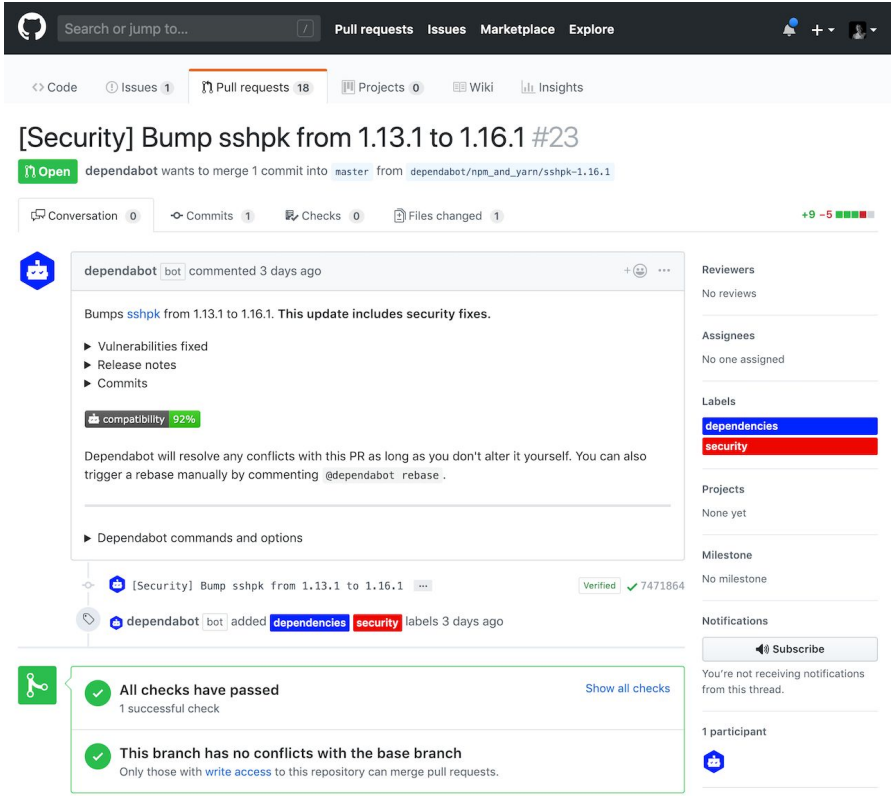
- Node Package Manager (NPM) for JavaScript/Node.js
- Overwhelming NPM build message
- unhelpful warning message



npm WARN deprecated minimatch@0.4.0: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue
npm WARN deprecated circular-json@0.3.3: CircularJSON is in maintenance only, flatted is its successor.
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated

How GitHub handles this Problem

1. Marketplace Plugin - GitHub **dependabot**
2. Via automated Pull Requests
   a. Version bumps
   b. Temporary dependency file to fix dependencies being used transitively
3. No visual aid on how dependencies are structured

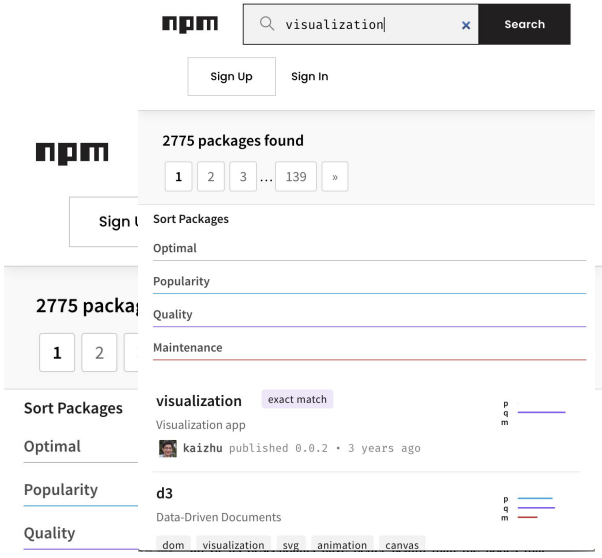


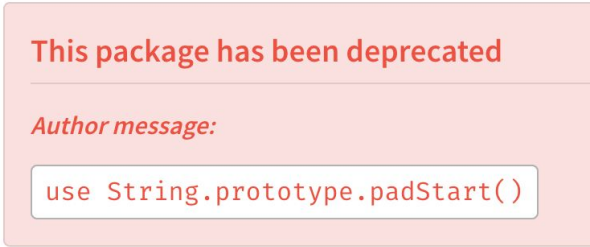
A version bump PR created by dependabot

Metrics of Health

A measurement of aspects of development process

- Scores used in ranking search results maintained by NPM
  - popularity - downloads
  - quality - presence of quality code
  - maintenance - activities
  - final (optimal) - combined
- Package deprecation flag
  - Package is no longer maintained
  - Do not recommend using the package

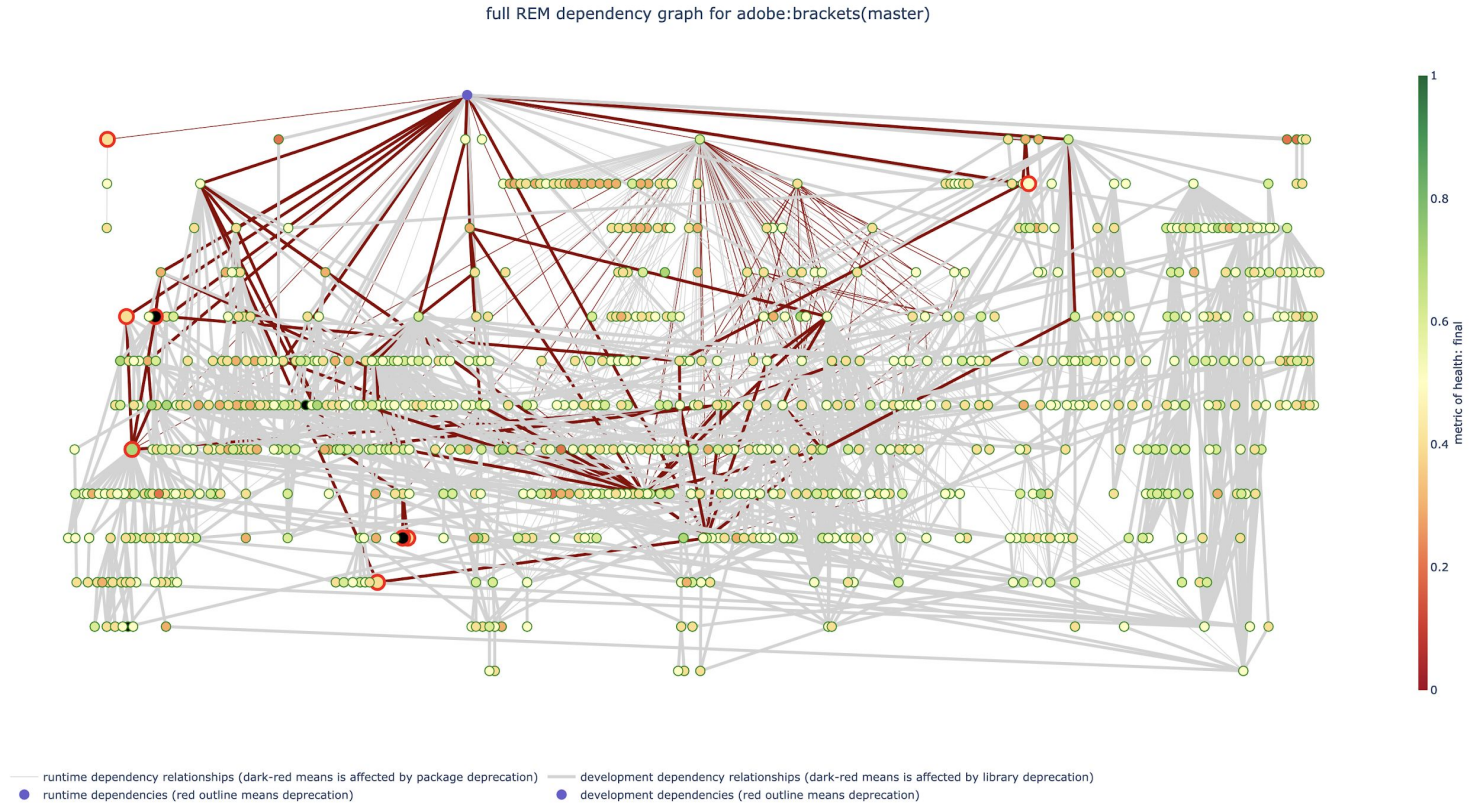




left-pad

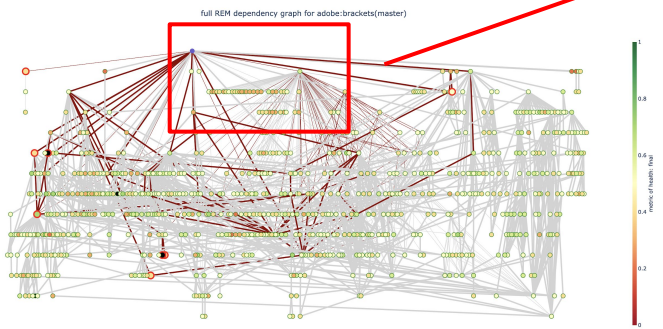
1.3.0 • Public • Published 2 years ago

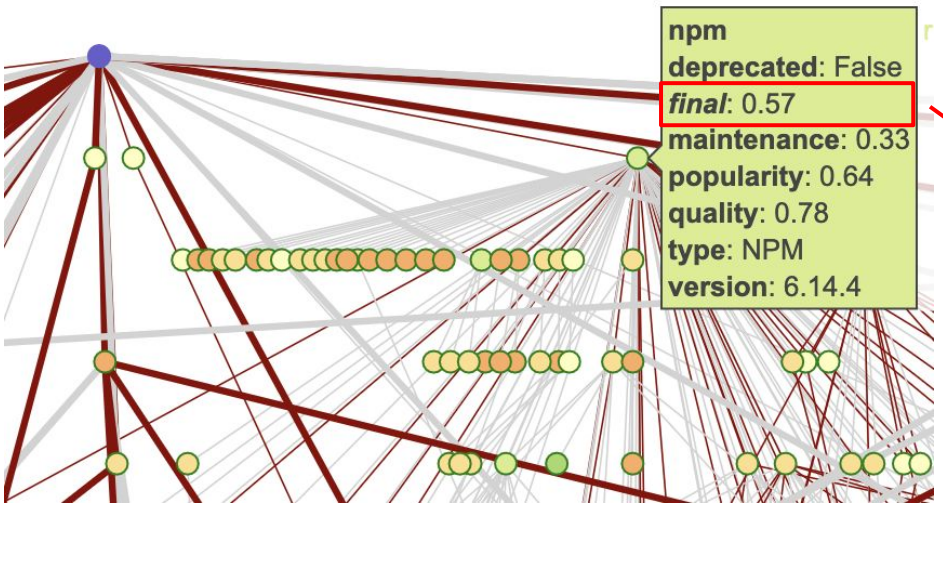
Ripple Effect of Metrics (REM)



REM - Metrics of Health

- Node annotation
- Colour legend

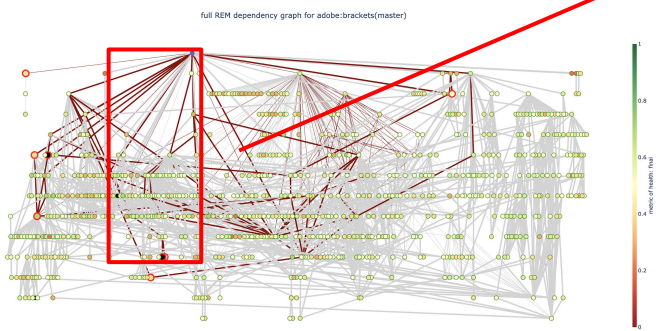


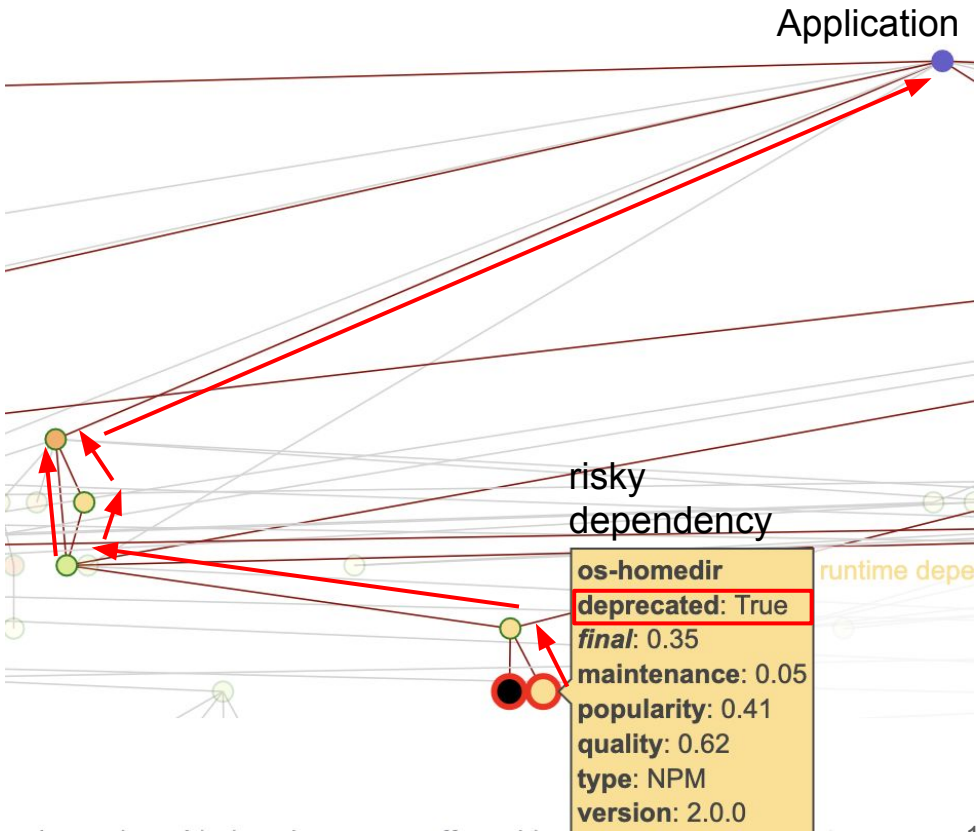


REM - Ripple Effect

Risky dependency to application root

- Ripple-Effect Metric: node outline
- Ripple-Effect Path: affected edge

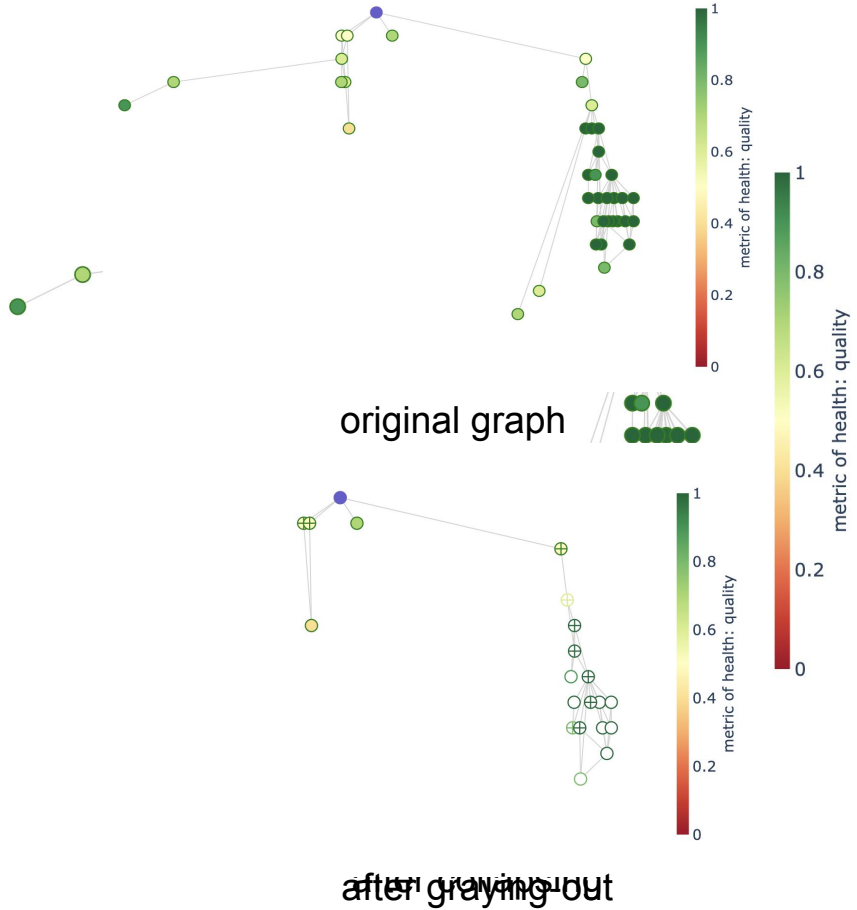




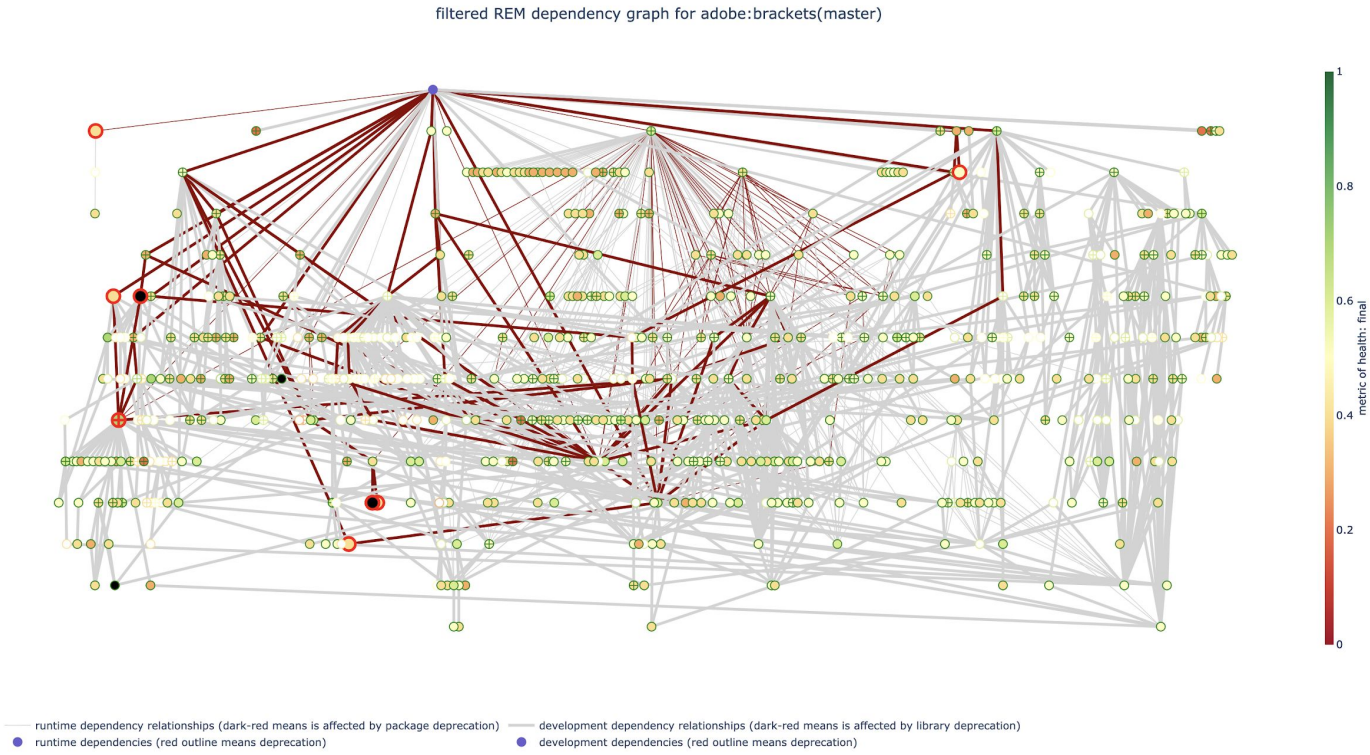
REM - Filtering

Reduce the number of nodes to highlight those considered more risky

1. Collapsing - nodes healthier than parent nodes
2. Graying-out - nodes healthier than ancestors in direct dependency nodes



REM - Filtering



DEMO

turingmachine.org/rem_demo

REM: Visualizing the Ripple Effect on Dependencies Using Metrics of Health

Ripple Effect

Metrics of Health

Filtering