PROTECT YOUR RUST LIBRARY RELEASES FROM BREAKING API CHANGES

THE IDEA

You have most likely seen the "1.0.2" format of numbers that indicates the versions of code packages. It is called semver (semantic versioning) – a set of rules dictating how version numbers for libraries are assigned and incremented, reflecting the changes in their public APIs.

CONSEQUENCES

More than 1 in 30 releases of the largest Rust libraries have violated semver, causing the users' code to suddenly stop compiling. In such cases, the packages' maintainers have to delete the faulty releases or create patches to fix them.

DIFFICULTIES

The surprising frequency of these violations is caused by the complexity of many changes that qualify as invalid. Unfortunately there are not many tools available for keeping semver under control. It is because the semantics of popular languages make complete and automatic verification hard to achieve.

SOLUTION

Instead of looking at the source code, the tool compares the automatically generated documentation of two versions of the library's public API. To parse this data, it uses a special query engine that can read from any organized database. Thanks to this, the checks run by the tool are in the form of short queries.

CARGO-
semver
-CHECKS

OUR TEAM

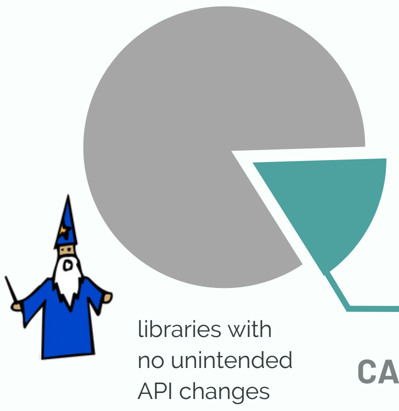
TOMASZ NOWAK

MICHAŁ STANIEWSKI

MIESZKO GRODZICKI

BARTOSZ SMOLARCZYK

PROJECT OWNER	SUPERVISOR	PATRONAGE
Predrag Gruevski	Michał Możdżonek	JetBrains



more than **1 in 6** of the top 1000 **Rust libraries broke their users'** projects with new releases because of:

- missing functions
- removing enums
- changing struct fields
- modifying traits
- . . . and many more

CARGO-SEMVER-CHECKS PREVENTS THESE PROBLEMS

The tool is to become part of the official package manager and build system for the Rust language.

IT IS ALREADY USED BY

global companies

Amazon

Adobe

Microsoft

Mozilla

Rust libraries with 150 mln downloads

libp2p

pest

data-encoding

tokio