May 29, 2018

Tracking Package Repository Release Using Blockchain

Alfred Pang

TRACKING PACKAGE REPOSITORY RELEASE USING BLOCKCHAIN

ABSTRACT

A system and a method are disclosed for tracking package repositories using blockchain. The method includes publishing a transaction on a blockchain when a package maintainer wants to put out a new release of software packages or libraries. The blockchain provides a mechanism for agreement between package maintainers to provide a snapshot of all the versions of the software packages in time. The disclosed system is configured for tracking package repositories thereby giving the users build reproducibility and agreement on canonical versions of libraries. Further, a token mechanism may be provided to reward the package maintainers for using the blockchain. The package maintainers may use these tokens to pay for transactions or make announcement related to version release on the blockchain.

KEYWORDS: Software packaging, versioning, blockchain, version compatibility

BACKGROUND

Package dependency is unreliable in many programming languages. For example, in Golang, “go get” command may be entered by a user to pull package dependencies. The “go get” command downloads a snapshot, which is anything at the head of each GitHub repository for each package dependency. The snapshot representing what packages were downloaded at a specific point in time is not necessarily reproducible by another user running "go get". Version control of packages is a mix and match of whatever happened to be downloaded and sometimes selectively updated by the user. In short, there is no global agreement at all on what set of package versions is canonical. In some instances, this may lead to difficulty in figuring out which versions safely work with each other as package maintainers themselves have difficulty in

providing any guarantees about what dependencies are tested to work well together. Existing solutions address the unreliability at the level of individual reproducibility by taking a snapshot of the specific downloaded package version(s). However, such individual-level solutions leave the user at the mercy of that snapshot taken by another user who may not necessarily have a well-tested set of packages.

DESCRIPTION

A system and a method for tracking package repositories using blockchain are provided. The method, as illustrated in FIG. 1, includes putting out a new release of software packages or libraries by a package maintainer at 102. The package maintainer publishes a transaction on a blockchain when at 104. For instance, the package maintainer may publish a known good git commit hash to a blockchain to put out a new release of library.

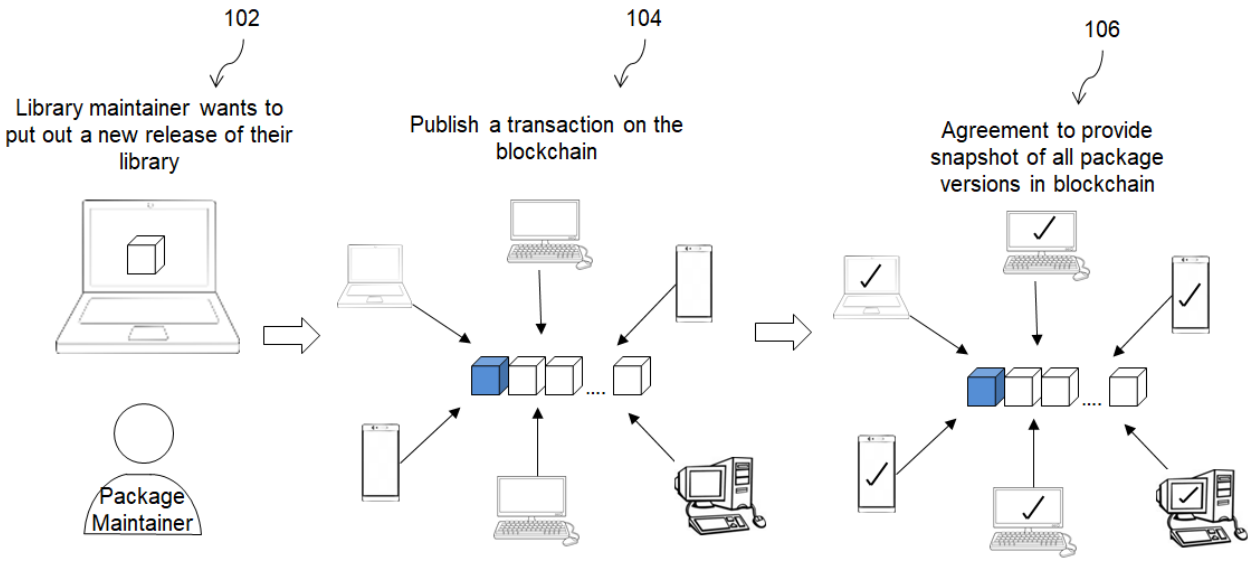

FIG. 1: Method for tracking package repositories using blockchain

Further, the blockchain provides a mechanism for agreement between package maintainers such that the blockchain provides a view or snapshot of the versions of all the software packages in time, at block 106. The library maintainers may make better claims and

guarantees about what versions of libraries should work together at a specific blockchain epoch. This provides reproducibility and agreement on canonical versions of libraries.

Further, a typical token mechanism may be used in order to reward package maintainers for using the blockchain. The package maintainers may use these tokens to pay for transactions or version release announcements on the blockchain. This would be useful for this particular blockchain implementation to allow easier bifurcation, if necessary. For instance, if there are particular harmful or malicious packages that should not be made available, a small set of trusted blockchain administrators should be able to make the necessary overrides and stop the transaction.

The use of blockchain for solving the package dependency issue is an advantageous novel solution with new capabilities. The method enables tracking which version of packages are compatible and safe to work with. Since the blockchain is ordered, the same snapshot of the versions of the software packages that are currently used may be provided to many users. The blockchain implementation also totally removes the need to figure out compatibility of the software package versions while providing a coordination mechanism between package maintainers.