KubeCon | CloudNativeCon

Europe 2022

WELCOME TO VALENCIA

KubeCon CloudNativeCon Europe 2022

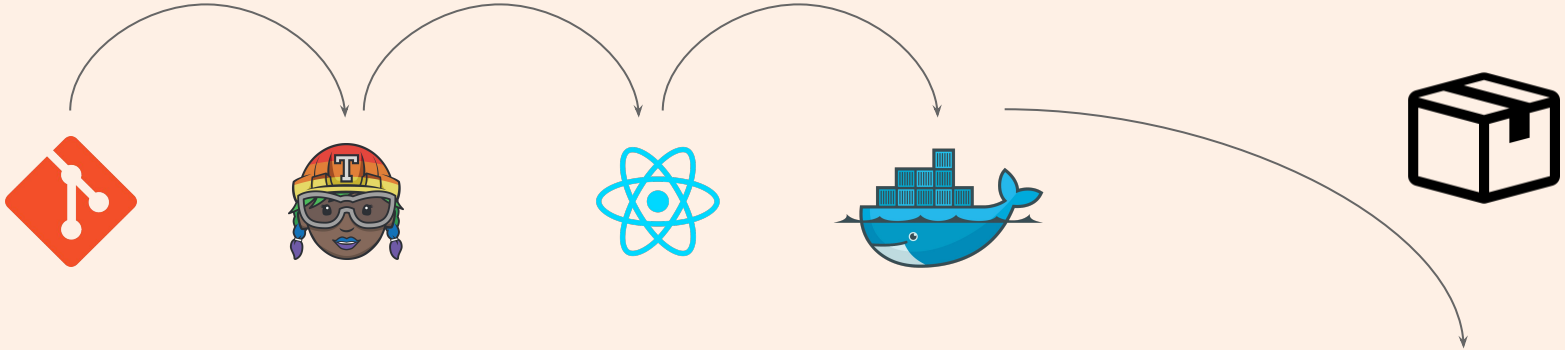
Updates from The Update Framework

Lukas Pühringer, NYU
Jussi Kukkonen, VMware

Agenda

- Content Distribution
- TUF 101
- The TUF Project
- TUF in Practice
- Plans for the Future

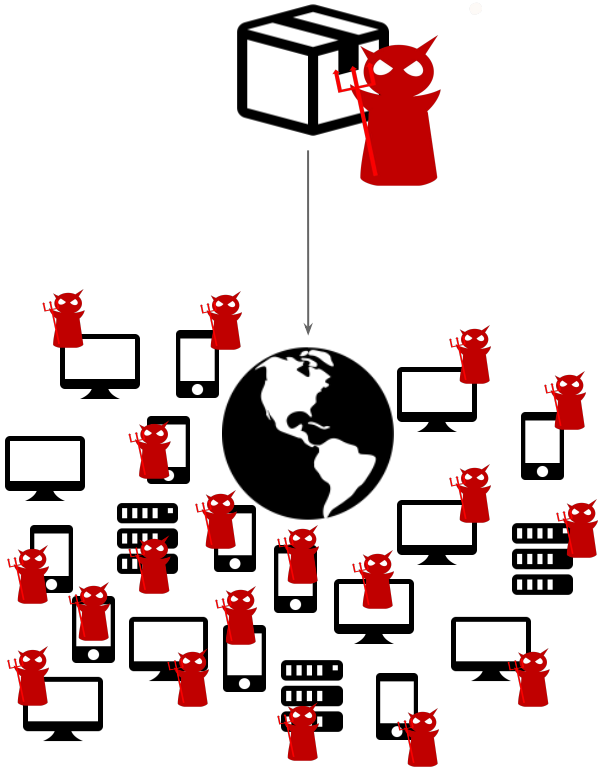
Content Distribution



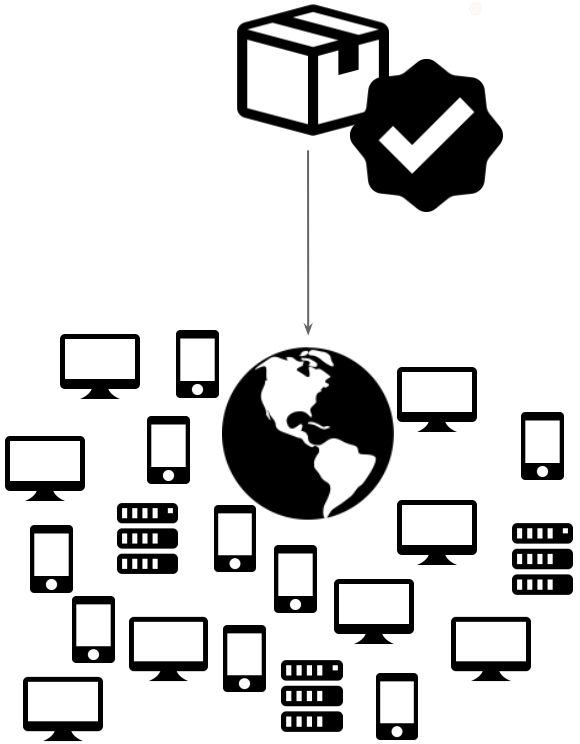
a crucial part of the
software supply chain ...

Compromise Content Distribution

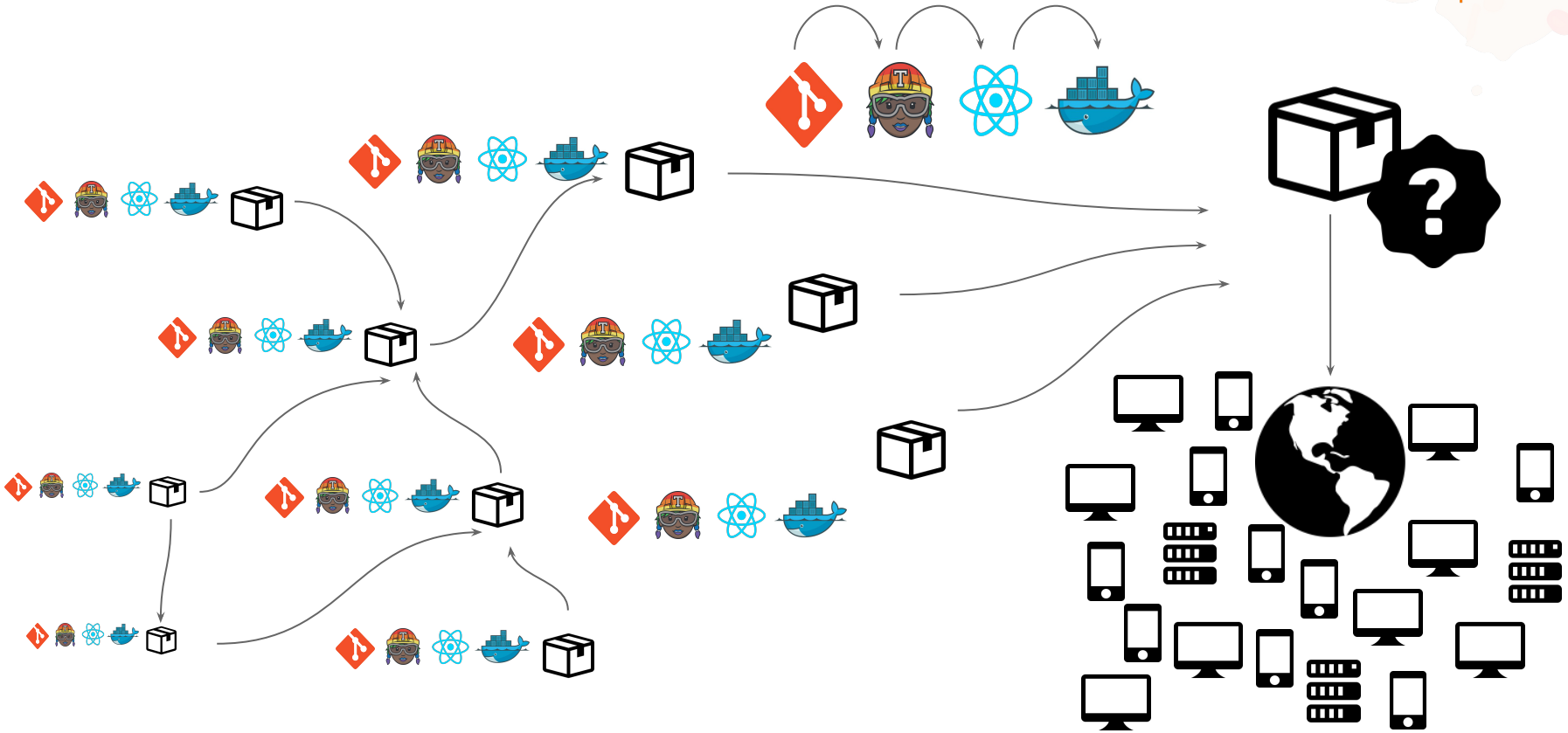
... and an attractive target for attackers



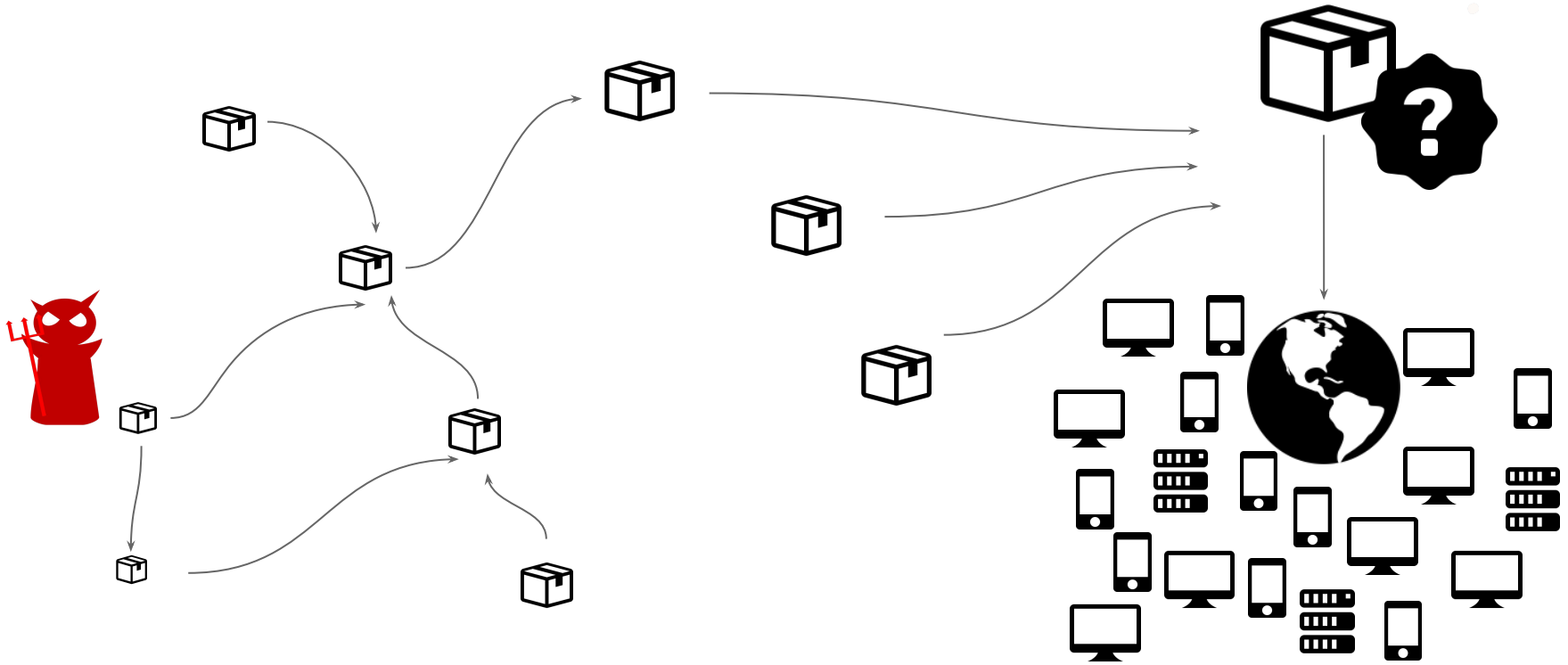
Protect Content …



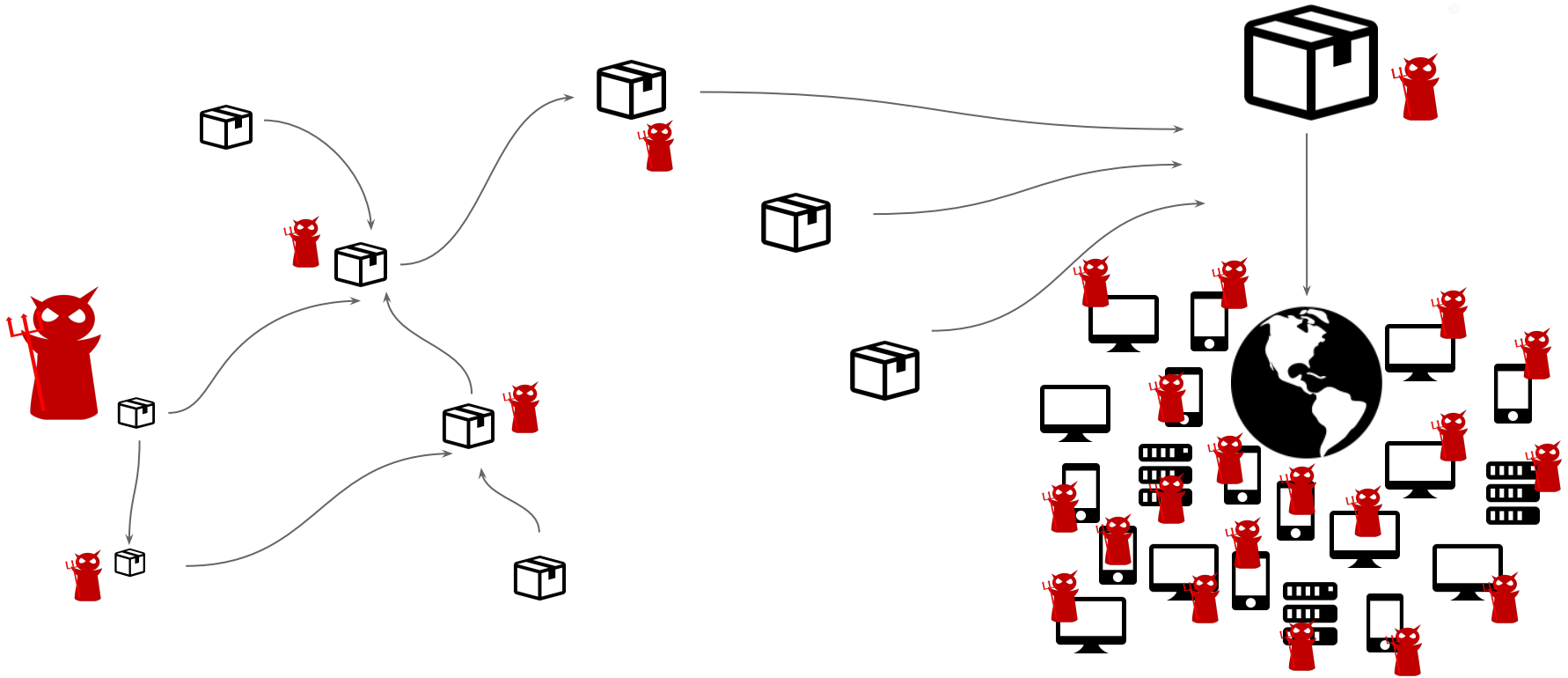
Protect Content at Scale



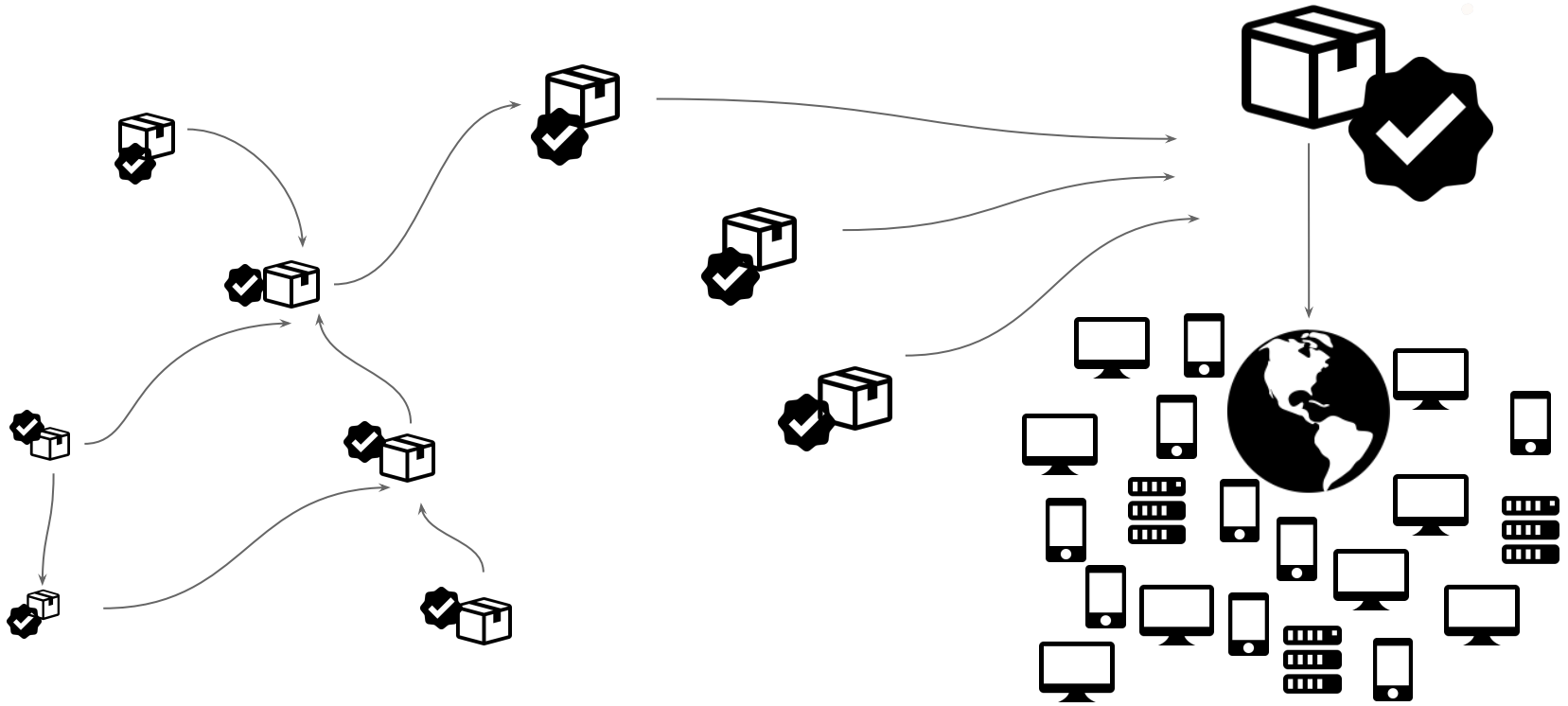
Protect Content at Scale!



Protect Content at Scale!!!



Solution: Sign all the Things!



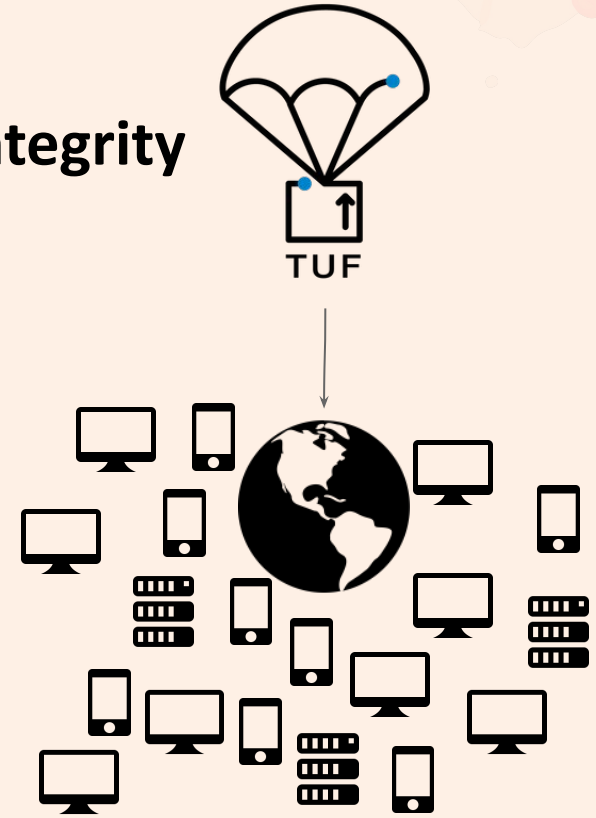
Signing alone is not enough

- Trust decisions at scale
- Compromise is fatal
- Freshness, consistency

source: poorlylockedbikes.com

The Update Framework 101

- Protect **freshness**, **consistency** and **integrity**
- Delegate **trust** at scale
- Reduce **impact** of compromise
- Allow in-band **recovery**



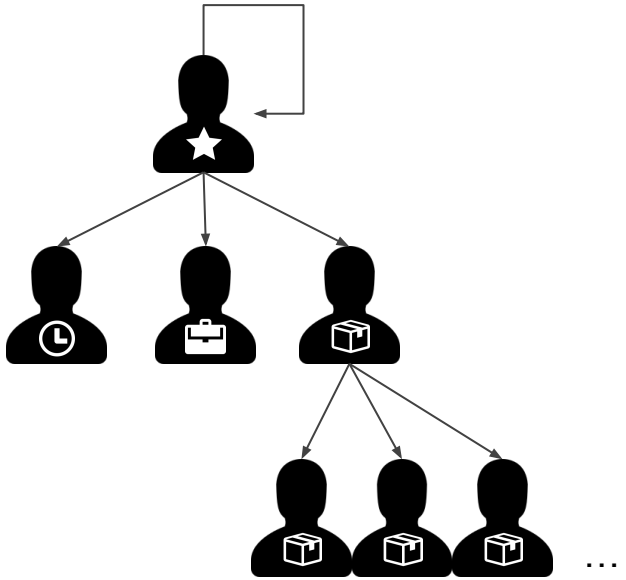
Protect Content

- Cryptographic signatures
  - content → **integrity**
  - repository → **consistency**

- Rapid signature expiration → **freshness**

Delegate Trust at Scale

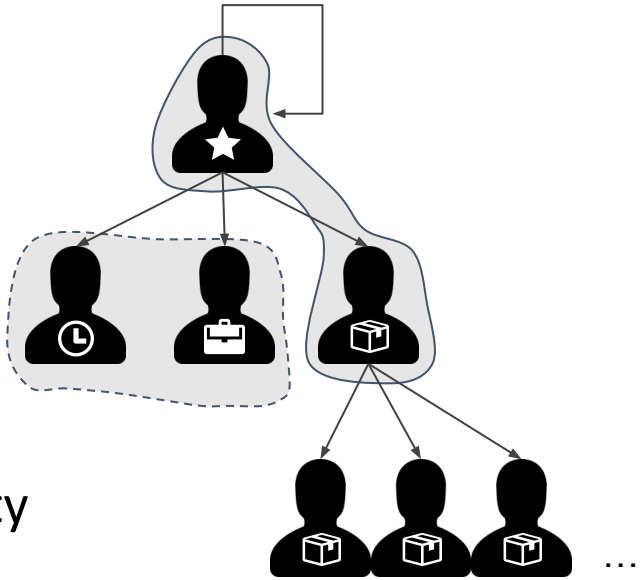
Separate responsibilities by **role**



Separate responsibilities by **content**

Share responsibility

Reduce Impact of Compromise

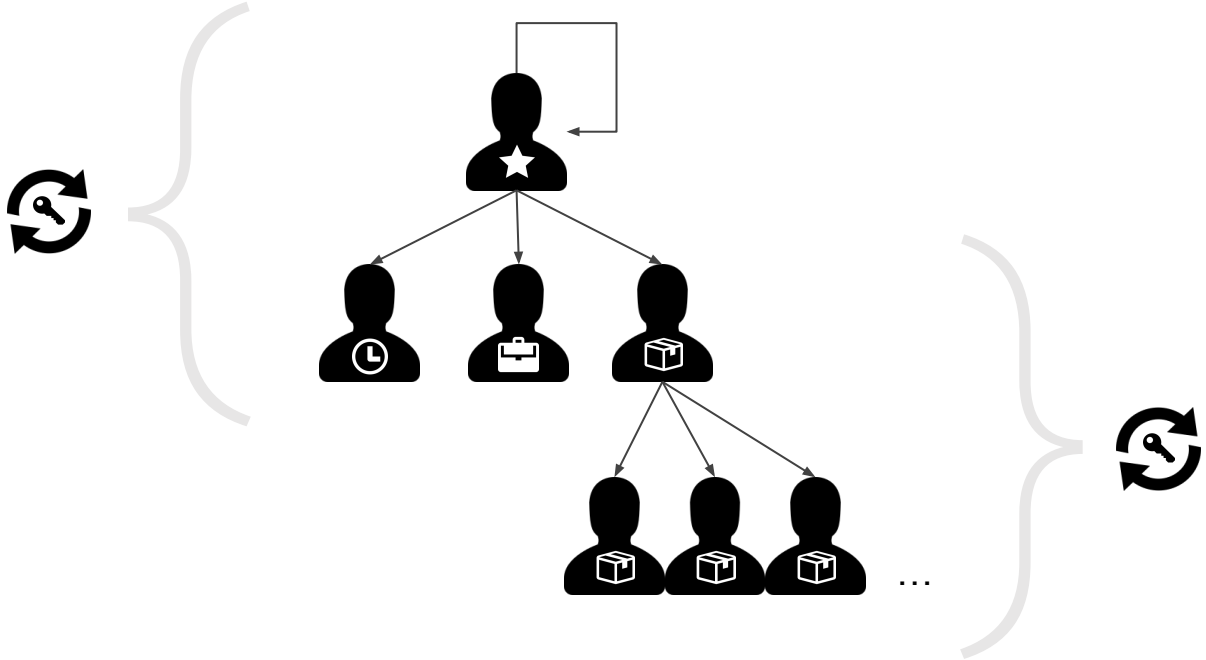


…

High availability
⟶ minimize responsibility

High responsibility
⟶ minimize risk

Allow in-band Recovery



Delegators can revoke and rotate keys for delegatees

The TUF Project

- Core TUF
  - Specification
  - TAP
  - Reference implementation
- More TUF

Specification

Defines signed role metadata **formats**, how to **maintain** them in a content repository and how to **consume** them on the client.

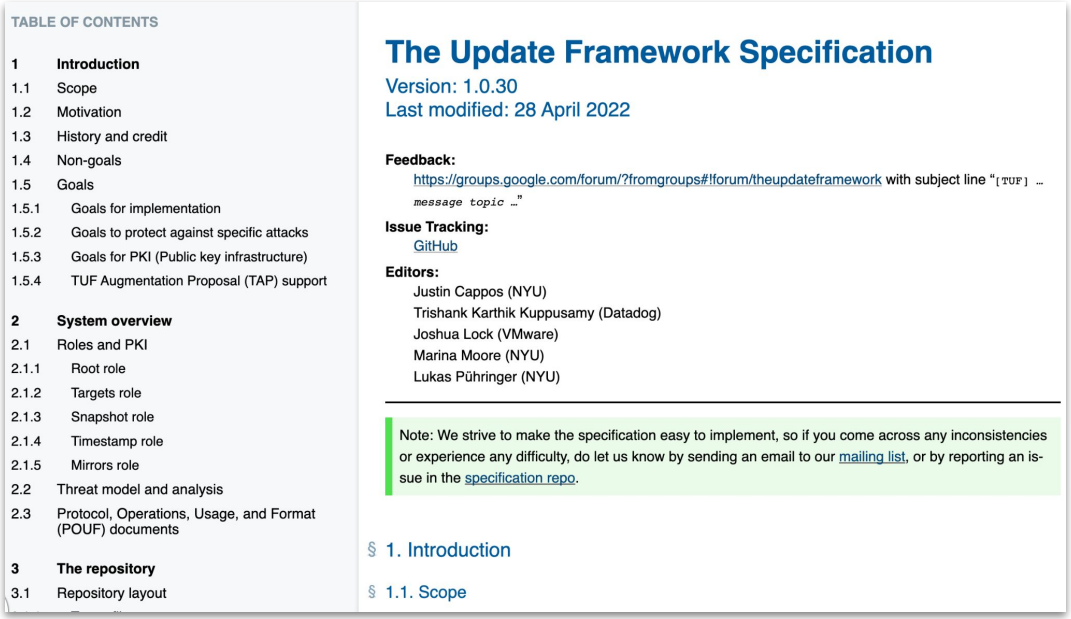


TABLE OF CONTENTS

The Update Framework Specification

Version: 1.0.30
Last modified: 28 April 2022

Feedback:
https://groups.google.com/forum/?fromgroups#!forum/theupdateframework with subject line "`[TUF] … message topic …`"

Issue Tracking:
GitHub

Editors:
Justin Cappos (NYU)
Trishank Karthik Kuppusamy (Datadog)
Joshua Lock (VMware)
Marina Moore (NYU)
Lukas Pühringer (NYU)

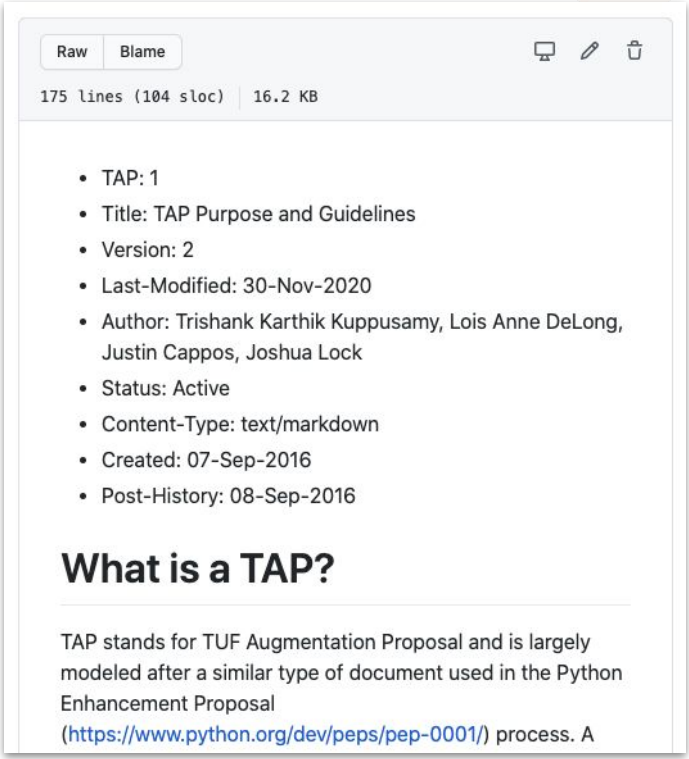
Note: We strive to make the specification easy to implement, so if you come across any inconsistencies or experience any difficulty, do let us know by sending an email to our mailing list, or by reporting an issue in the specification repo.

§ 1. Introduction

§ 1.1. Scope

TUF Augmentation Proposals (TAPs)

The **specification** is a dependable but living document and **evolves** responsibly through the TAP process.



Raw Blame

175 lines (104 sloc) 16.2 KB

- TAP: 1
- Title: TAP Purpose and Guidelines
- Version: 2
- Last-Modified: 30-Nov-2020
- Author: Trishank Karthik Kuppusamy, Lois Anne DeLong, Justin Cappos, Joshua Lock
- Status: Active
- Content-Type: text/markdown
- Created: 07-Sep-2016
- Post-History: 08-Sep-2016

What is a TAP?

TAP stands for TUF Augmentation Proposal and is largely modeled after a similar type of document used in the Python Enhancement Proposal (https://www.python.org/dev/peps/pep-0001/) process. A

Reference Implementation

- **Client** and **Metadata** API
- Readability and recognizability
- Production-quality code
- python-tuf v1.0

More TUF

Libraries




Adopted







Work In Progress




Derived

Adapted



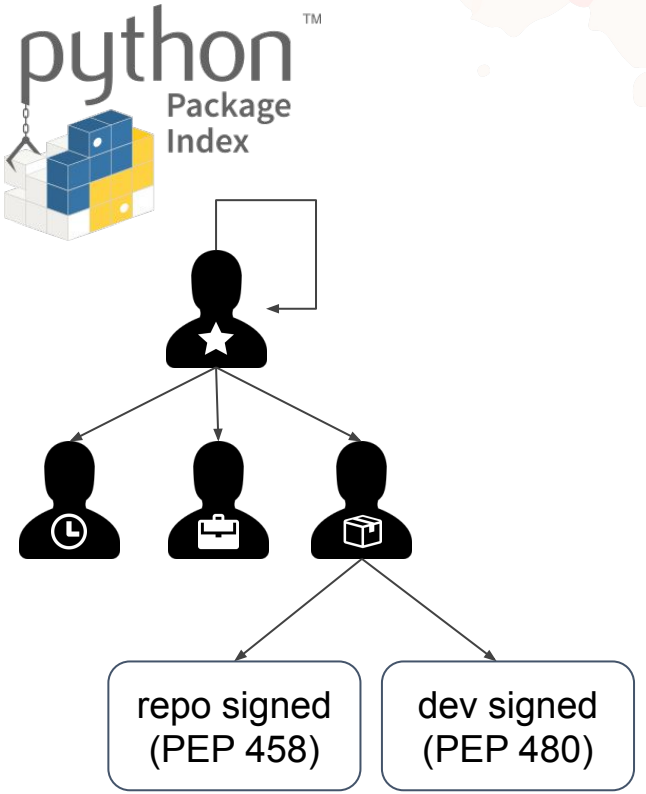


TUF in Practice

- Client
  - Well-defined workflows
  - Off-the-shelf TUF libraries work
- Repository
  - Large variance in implementations
  - No off-the-shelf libraries

Case: PyPI – *community repo challenges*

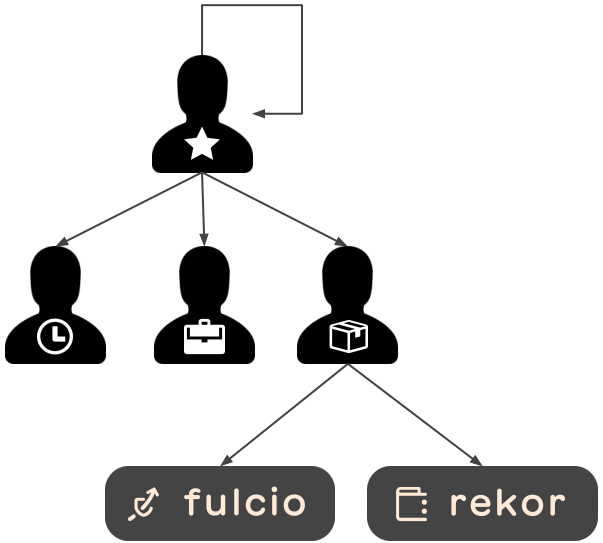
Serves content **signed in repo** (PEP 458), or, preferably, by **developers** in controlled **namespaces** (PEP 480).



Case: Sigstore – *keys can be content too*

Uses TUF to establish a trust root for its certificate authority and identity log (**fulcio**) and artifact log, (**rekor**) which in turn are used by its content signing and verifying tool (**cosign**).

sigstore



Plans for the Future

- Repository Playground
- Sigstore: ephemeral keys + OIDC
- …

- theupdateframework.io
- #tuf on cloud-native.slack.com
- theupdateframework@googlegroups.com

Thank you!

(references on next pages)

References (1/2)

TUF core

- Project Website
  https://theupdateframework.io/
- Specification
  https://github.com/theupdateframework/specification
- TUF Augmentation Proposals (TAP)
  https://github.com/theupdateframework/taps
- Reference Implementation
  https://github.com/theupdateframework/python-tuf/

TUF Libraries / Frameworks

- Go
  https://github.com/theupdateframework/go-tuf
- PHP
  https://github.com/php-tuf/php-tuf
- Rust
  https://github.com/theupdateframework/rust-tuf
  https://github.com/awslabs/tough

References (2/2)

Ongoing and Upcoming

- Python TUF Reaches Version 1.0.0
  https://ssl.engineering.nyu.edu/blog/2022-02-21-tuf-1_0_0
- PEP 458 – Secure PyPI downloads with signed repository metadata
  https://www.python.org/dev/peps/pep-0458/
- The Sigstore Trust Model
  https://dlorenc.medium.com/the-sigstore-trust-model-4b146b2ecf2c
- Sigstore: Bring-your-own sTUF with TUF
  https://blog.sigstore.dev/sigstore-bring-your-own-stuf-with-tuf-40febfd2badd
- Anaconda Content Trust: Conda Signature Verification
  https://www.anaconda.com/blog/conda-signature-verification
- TAP 15: Succinct hashed bin delegations
  https://github.com/theupdateframework/taps/blob/master/tap15.md
- TAP 17: Remove Signature Wrapper from the TUF Specification
  https://github.com/theupdateframework/taps/blob/master/tap17.md
- Repository Playground
  https://github.com/jku/repository-playground/
- Python-tuf repository API proposal
  https://github.com/theupdateframework/python-tuf/blob/develop/docs/repository-library-design.md