DrupalCon
BARCELONA2024
24-27 SEPTEMBER

Supply Chain Security

in Drupal and Composer

DrupalCon
BARCELONA2024
24-27 SEPTEMBER

Nils Adermann
Composer &
Private Packagist

Christopher Gervais
Founding Partner
Consensus Enterprises

Neil Drumm
Senior Technologist
Drupal Association

What is a software supply chain?


18
MSC HOME TERMINAL
17
MSC HOME TERMINAL
16
15
14
13
MSC TRIESTE


67
65
63
61
SDI
SDI

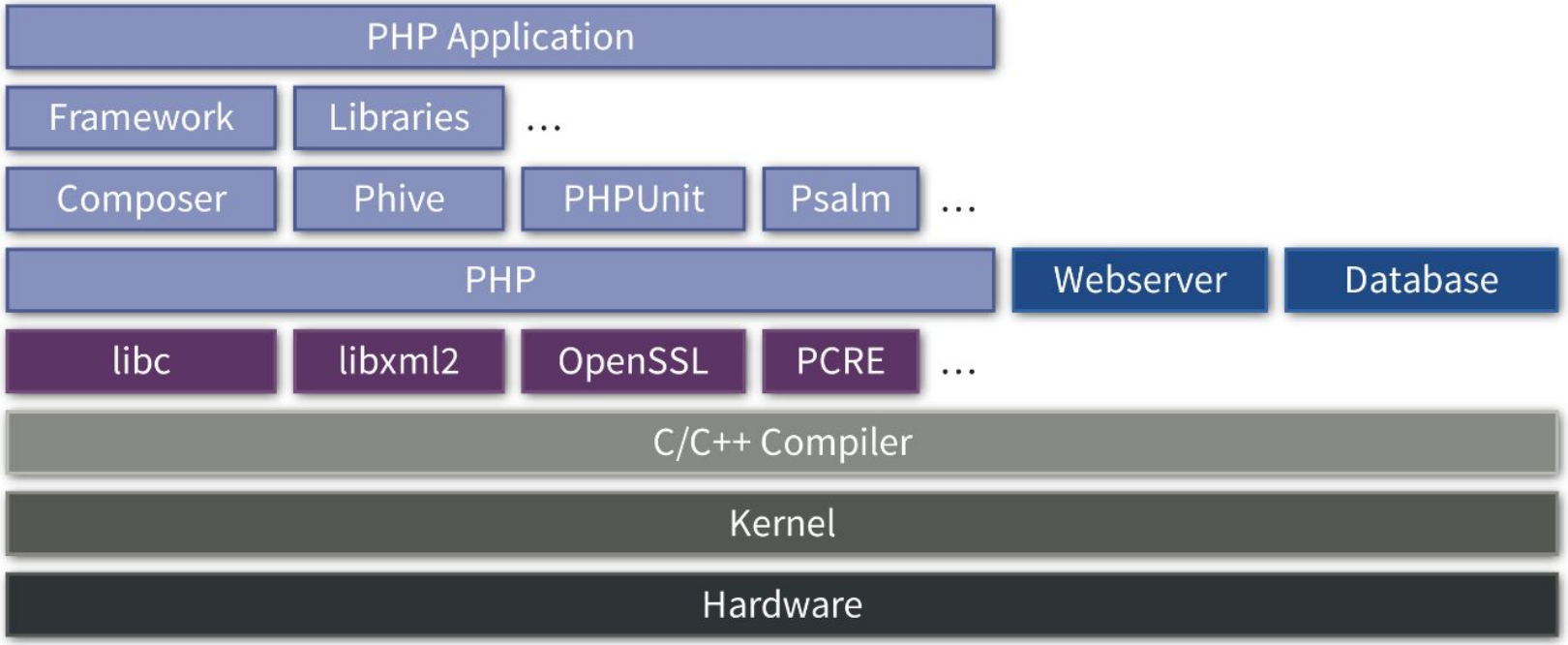
https://commons.wikimedia.org/wiki/File:Geely_assembly_line_in_Beilun,_Ningbo.JPG

A software supply chain is composed of the components, libraries, tools, and processes used to develop, build, and publish a software artifact.

https://en.wikipedia.org/wiki/Software_supply_chain

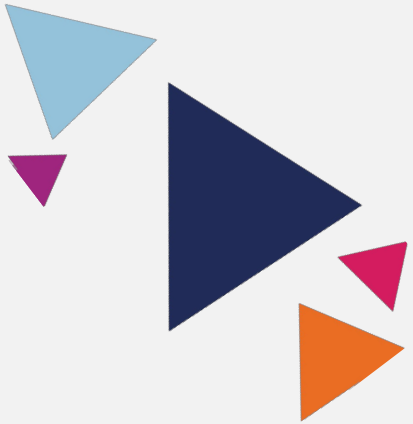
In other words:

The “full-stack” and all processes & tools involved in making and assembling it


PHP Application
Framework
Libraries
...
Composer
Phive
PHPUnit
Psalm
...
PHP
Webserver
Database
libc
libxml2
OpenSSL
PCRE
...
C/C++ Compiler
Kernel
Hardware

Hardware
Webbrowser
Framework
Libraries
...
PHP Application
Framework
Libraries
...
Composer
Phive
PHPUnit
Psalm
...
PHP
Webserver
Database
libc
libxml2
OpenSSL
PCRE
...
C/C++ Compiler
Kernel
Hardware

Supply Chain Attacks



- Heartbleed - https://heartbleed.com/ - 2014
  - OpenSSL: System memory accessible externally
- SolarWinds Orion / 2020 United States federal government data breach
  - attackers gained entry to a build system, likely through a compromised Office 365 account
  - modified software updates to include remote access on any machine installing Orion
  - discovered in December ‘20 after breach Sep ‘19
- Log4Shell
  - https://en.wikipedia.org/wiki/Log4Shell
  - Log4j vulnerability, standard Java logging library
  - existed 2013 - November 24, 2021
  - Arbitrary code execution, extremely widely used, CVSS Score 10/10
- XZ Utils / liblzma
  - https://en.wikipedia.org/wiki/XZ_Utils_backdoor
  - Introduced by covert malicious maintainer
  - Backdoor in compression library running in OpenSSH process granting remote access
  - Fortunately detected very early in distribution on March 29th

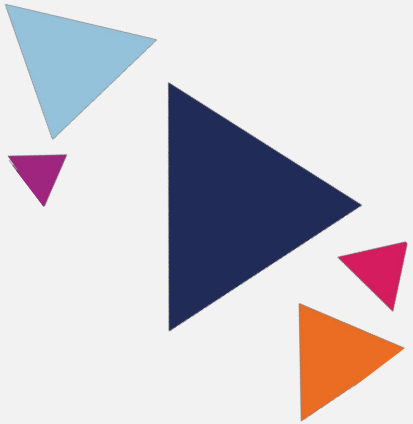
DrupalCon
BARCELONA2024
24-27 SEPTEMBER

Composer & packagist.org

Composer Supply Chain Vulns

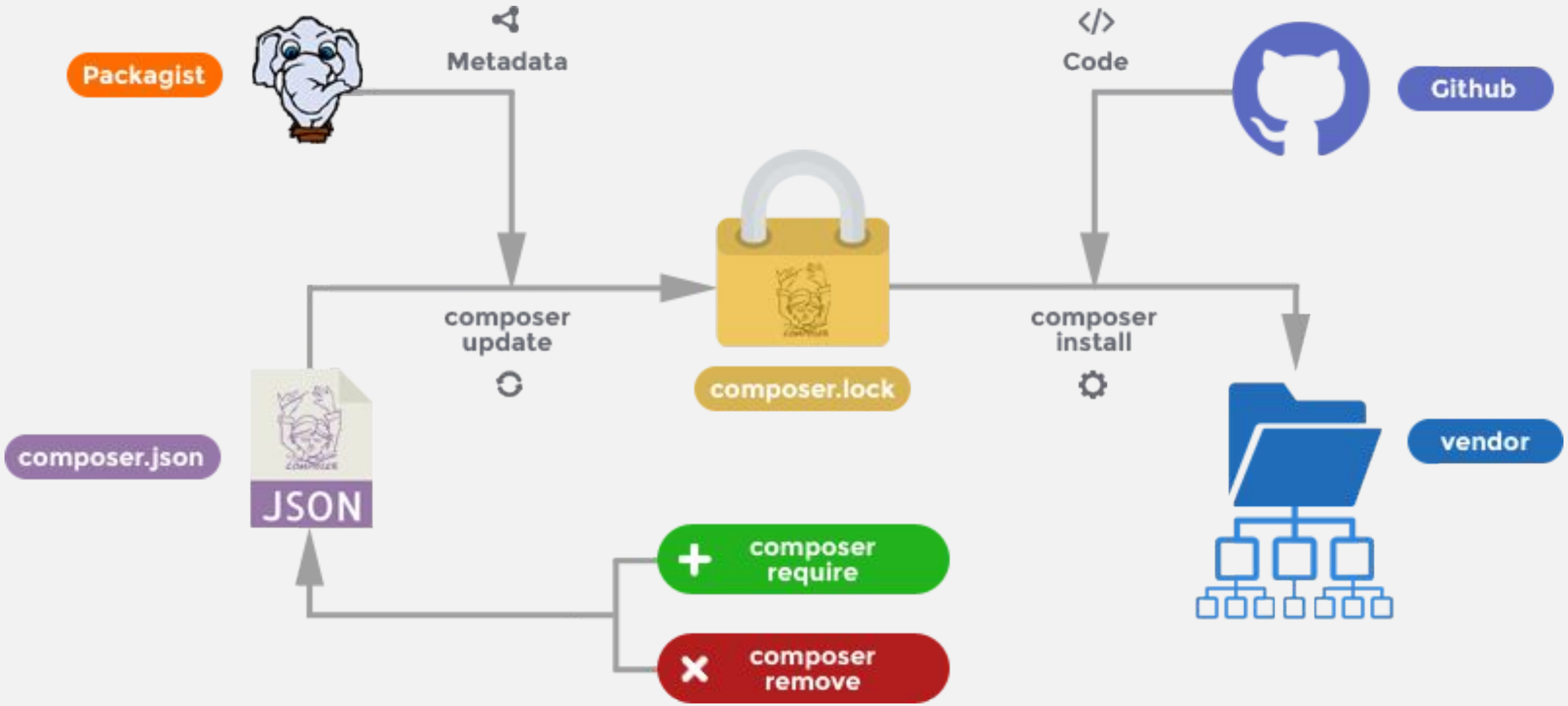
- Mar 11, 2021: Git Clone Security Vulnerability
  - https://blog.packagist.com/git-clone-security-vulnerability/
  - Git vulnerability on case insensitive filesystems can be exploited through Composer if you clone dependencies
- Apr 27, 2021: Composer Command Injection Vulnerability
  - https://blog.packagist.com/composer-command-injection-vulnerability/
  - Code execution through Mercurial repository URL injection
- Apr 13, 2022: Composer Command Injection Vulnerability
  - https://blog.packagist.com/cve-2022-24828-composer-command-injection-vulnerability/
  - Code execution through Git or Mercurial branch names

Composer Supply Chain Attacks

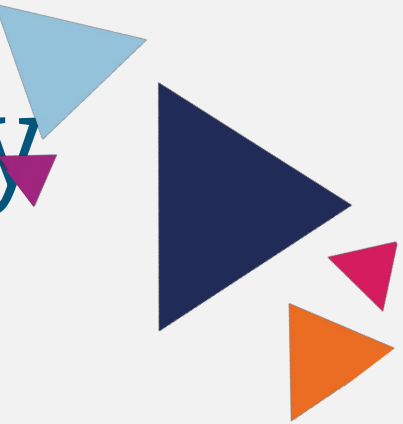


- May 19, 2022: GitHub Repo Jacking
  - Attacker registered GitHub username of former maintainer
  - Republished package with malicious code to steal AWS credentials
  - https://thehackernews.com/2022/05/pypi-package-ctx-and-php-library-phpass.html
  - https://github.blog/2024-02-21-how-to-stay-safe-from-repo-jacking/
    - Problematic with VCS repo URL references in composer.json too
  - Packagist.org uses GitHub repo ids: https://github.com/composer/packagist/pull/1411

- May 1, 2023: Packagist.org maintainer account takeover
  - https://blog.packagist.com/packagist-org-maintainer-account-takeover/
  - Editing of source URLs no longer allowed beyond 50k installs


Packagist
Metadata
Code
Github
composer update
composer.lock
composer install
composer.json
JSON
vendor
composer require
composer remove

Composer Supply Chain Security



- packagist.org metadata provider only
  - code comes from maintainer supplied URL on the internet
  - No checksums for code from GitHub (> 99% of packages)
  - No signatures from maintainers
  - **But:** No way to upload artifacts

- positive:
  - Everything over TLS
  - Installation from GitHub source archive URLs improves trust in artifacts
  - Smaller attack surface on packagist.org

Composer 2.4: composer audit

- **`composer audit`** command
  - Lists vulnerable versions in composer.lock
  - Uses packagist.org vulnerability db API
    - GitHub advisory database
    - FriendsOfPhp/security-advisories
  - Uses packages.drupal.org vulnerability info

- `composer update` implies `audit --format=summary`
- `composer require --dev roave/security-advisories:dev-latest`

Why is vendoring the wrong answer?

- Doesn't work because
  - Still need to update deps
    - still use the package manager to update vendor'd deps
    - or download everything manually
      - Lots of error prone work
      - Hard to spot issues like repo jacking
  - easy to miss removing files that was removed by vendors
  - managing conflicts harder than conflicts in lock file
  - bad actor, e.g. disgruntled employee
    - unmanaged directory hiding attack code in vendor/ tree
    - attack code in small modifications hidden in big update to vendor/ tree
- Instead: Run your own Composer repository

Private Packagist Artifactory Nexus Repository others **Drupal**

Drupal's Automatic Updates Initiative

Automatic Updates for Drupal

- Automate updates using Composer
- We want to be sure updates install what is intended
- The Update Framework (TUF) specification for update systems

packages.drupal.org	Rugged TUF server	PHP-TUF integration plugin	Package manager	Automatic updates	Project browser
Drupal.org server		Composer	Drupal modules		

Packaging Drupal.org projects

- Create module/theme release → queues packaging
- Package zip & tar.gz files
- Update packages.drupal.org metadata for Composer
- Send zip & metadata to Rugged
- Rugged updates TUF metadata

Packaging Drupal core

- Create release → queues packaging
- Subtree splitting to components & templates on GitHub
- Packagist.org handles metadata like any other GitHub project
- packagist-signed.drupalcode.org is a Satis mirror
- Send zip & metadata to Rugged
- Rugged updates TUF metadata

Packaging general projects

- General projects with `composer.json` & a release on Drupal.org
- Git push → notify Packagist.org to update metadata
- packagist-signed.drupalcode.org is a Satis mirror
- Send zip & metadata to Rugged
- Rugged updates TUF metadata

DrupalCon
BARCELONA2024
24-27 SEPTEMBER

Package Verification

Public-key Cryptography, Digital Signatures & Hash Functions

(just the basics)

Asymmetry (real-world example)

To *send* a letter, *you* need:

- my **address** (PUBLIC)

To *read* the letter, *I* need:

- my **mailbox key** (PRIVATE)

*N.B. This does **NOT** work in reverse*

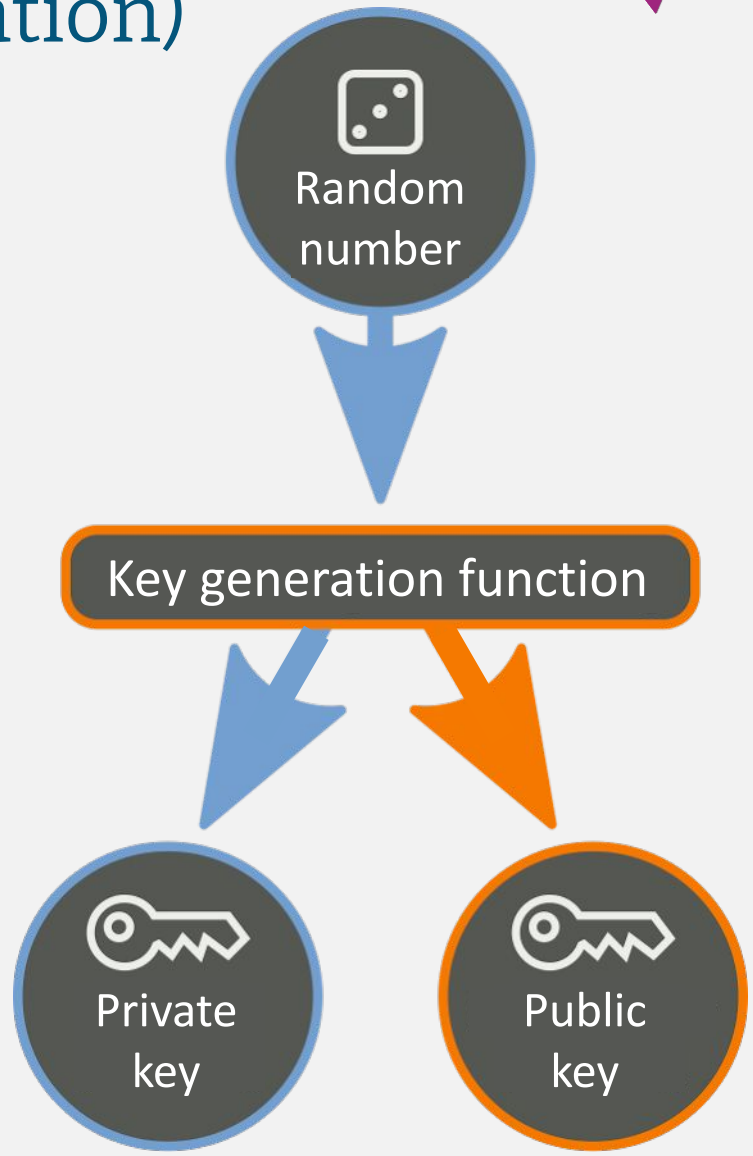
Key pairs (key generation)

Using very complex math, a large random number is used to generate a key pair.

A key pair consists of two files each containing a long string of characters.

Regardless of which one we use to encrypt a message, *only the other one* can be used to decrypt it.

N.B. Either key, used to encrypt a message, ***CANNOT*** *decrypt that message.*



Asymmetry (encrypt/decrypt)

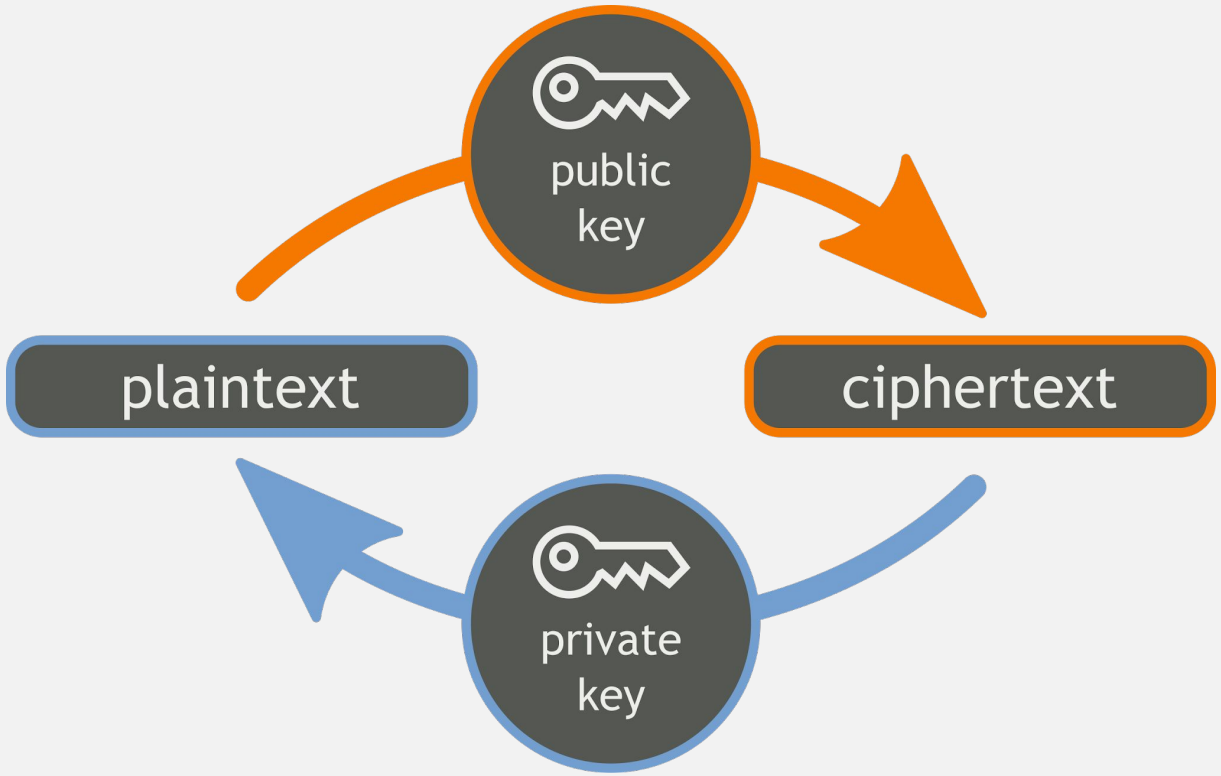
To *encrypt* a message, *you* need:

- my **public key** (PUBLIC)

To *decrypt* the message, *I* need:

- my **private key** (PRIVATE)

*N.B. The message is **secret***



Asymmetry (sign/verify)

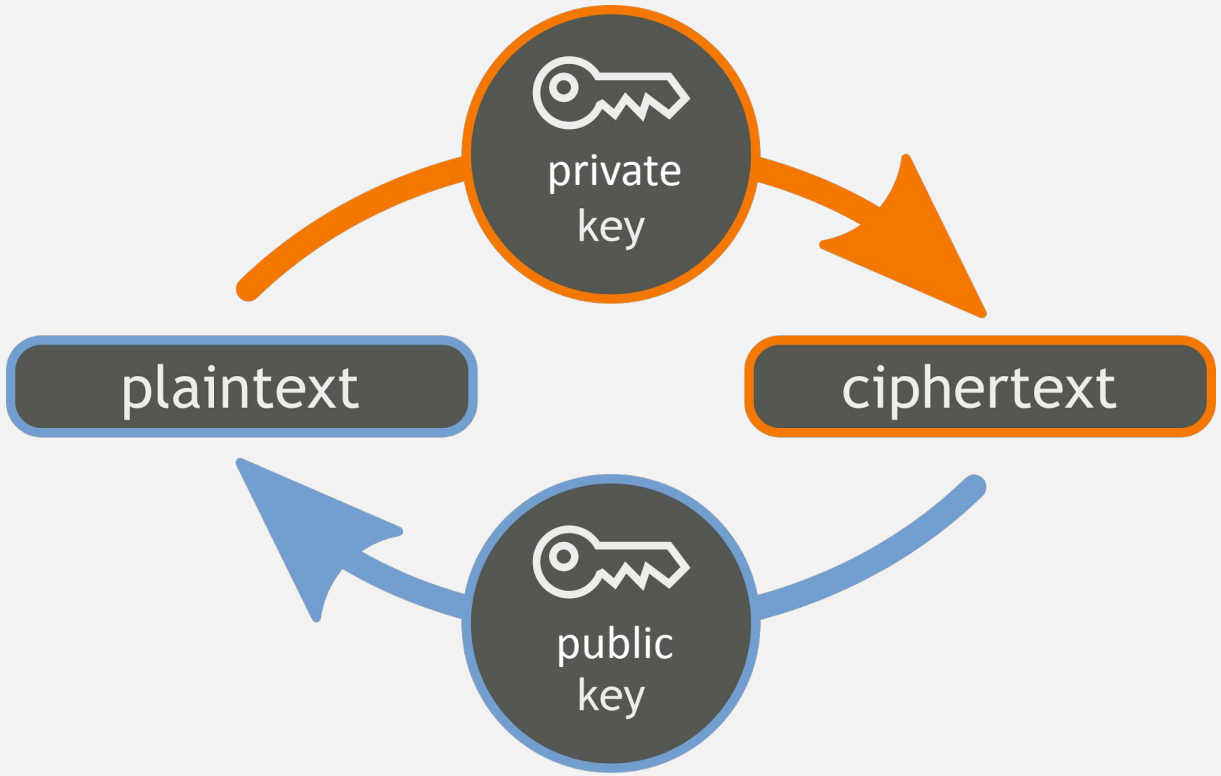
To *sign* a message, *I* need:

- my **private key** (PRIVATE)

To *verify* the signature, *you* need:

- my **public key** (PUBLIC)

*N.B. The message is **not secret***



Hash Functions

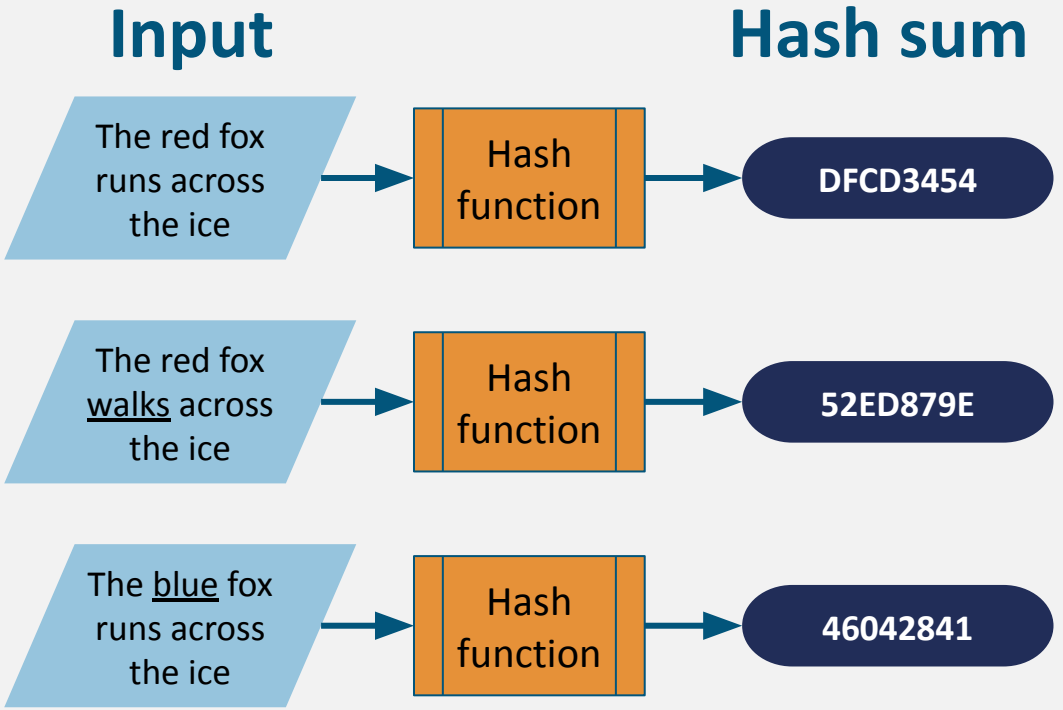
One-way program that scrambles text.

The hash sum **cannot be unscrambled**.

The **same input** always results in the **same hash sum**.

Different input always* results in a **different hash sum**.

*N.B. This can prove that the input has **not been altered***



** effectively always*

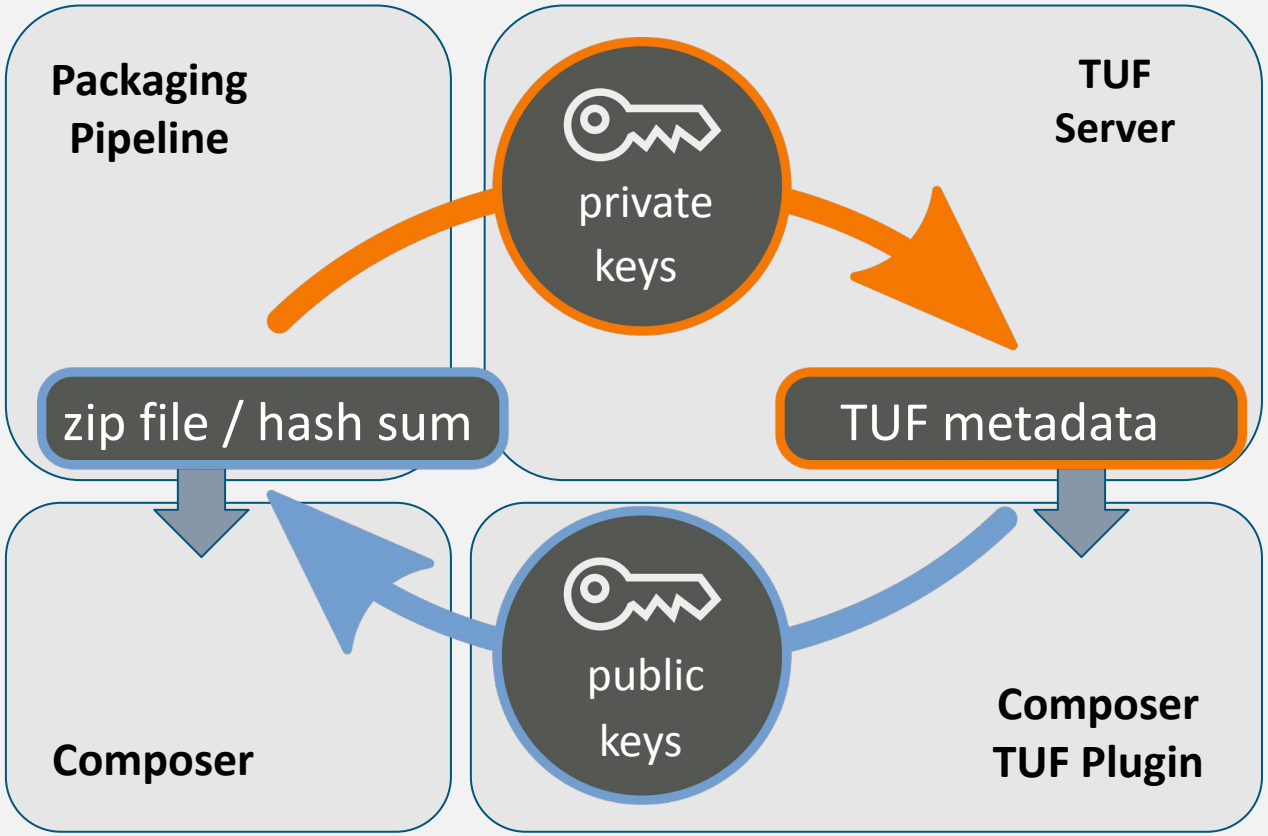
Package Verification

Packaging pipeline generates a zip file of an updated module

TUF server generates a hash of the zip file and signs metadata

Composer downloads zip file

Composer TUF plugin verifies zip file against TUF metadata



The Update Framework (TUF)

Design Principles

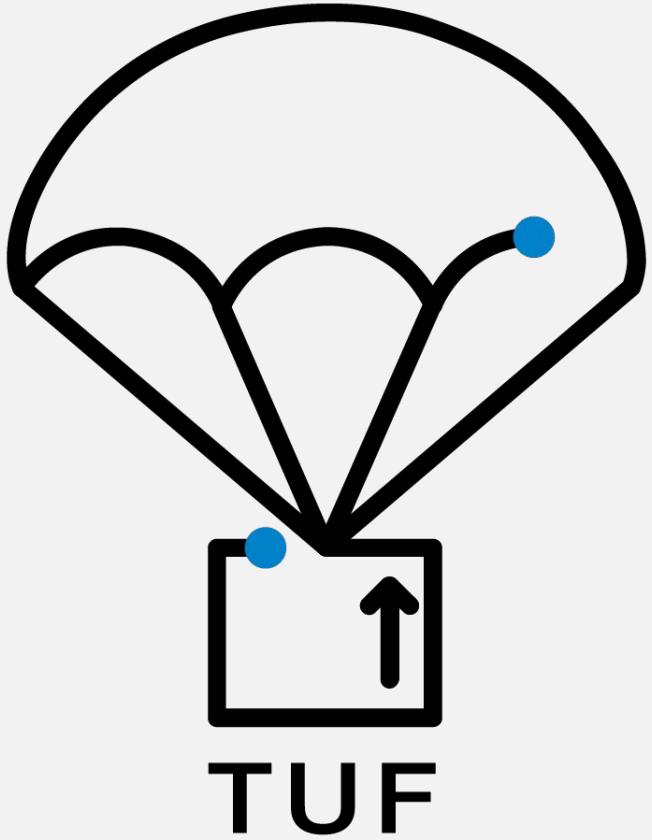
Trust
Compartmentalize signing authority that expires if not renewed.

Compromise Resilience
Use multiple keys. Minimize trust placed in online keys. Easy recovery/remediation.

Integrity
Verify downloaded files are intact, and that the repository overall is correct.

Freshness
Verify that the latest versions of files are available and recognize when a problem occurs.

Implementation Safety
The design of TUF itself must not introduce new attack vectors.



TUF Metadata (principles)

Root Metadata (*n*.`root.json`):
Specifies which keys are *trusted for signing* each of the other metadata; chain of trust.

→ *Trust & compromise resilience*

Timestamp Metadata (`timestamp.json`):
Ensures the *freshness* of the TUF metadata.
Minimizes unnecessary downloads of metadata.

→ *Freshness & repository integrity*

Snapshot Metadata (`snapshot.json`):
Ensures the *integrity* of the TUF Targets metadata.

→ *Repository integrity & implementation safety*

Targets Metadata (`targets.json`):
Ensures the *integrity* of the software packages.
Supports hashed bins and other delegations.

→ *Download integrity & implementation safety*

TUF Metadata (implementation)

n.root.json:
Specifies trusted keys
for the other top-level roles.

timestamp.json:
Lists hash, size, and version
number of the snapshot file.

snapshot.json:
Lists hash, size and version
numbers of all target metadata files

targets.json:
Lists hashes and sizes of
target files.


```
{
 "signatures": [
  {"keyid": "44c6…", "sig": "5783…"}
 ],
 "signed": {
  "_type": "targets",
  "expires": "2024-09-23T20:17:06Z",
  "spec_version": "1.0.31",
  "targets": {
   "test1.txt": {
    "hashes": { "sha256": "634b…" },
    "length": 6
   }
  },
  "version": 2
 }
}
```

Rugged TUF Server

Rugged TUF Server

Rugged is a server-side implementation of The Update Framework (TUF)

Rugged aims to make generating TUF metadata **simple**, and **robust**

Development sponsored by the **Drupal Association**

OSTIF security audit, in January 2024, found **no vulnerabilities**

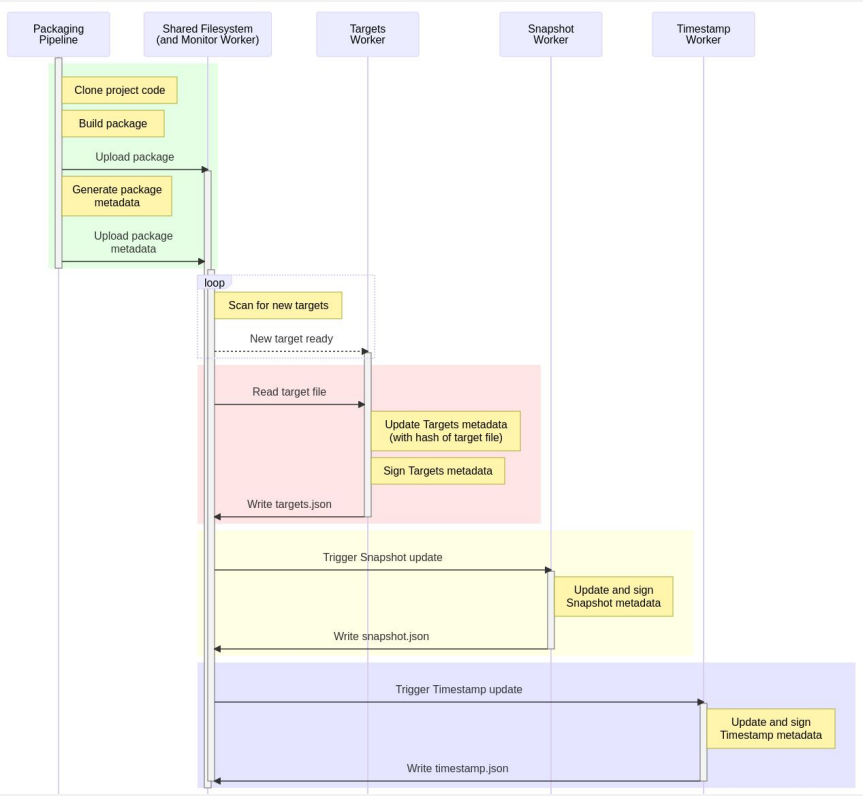
Rugged Components

Command-line (CLI) tool `(rugged)`:

- Initialize TUF repository
- Key-management tasks (`n.root.json`)
- Status reporting and logs
- Other manual maintenance operations

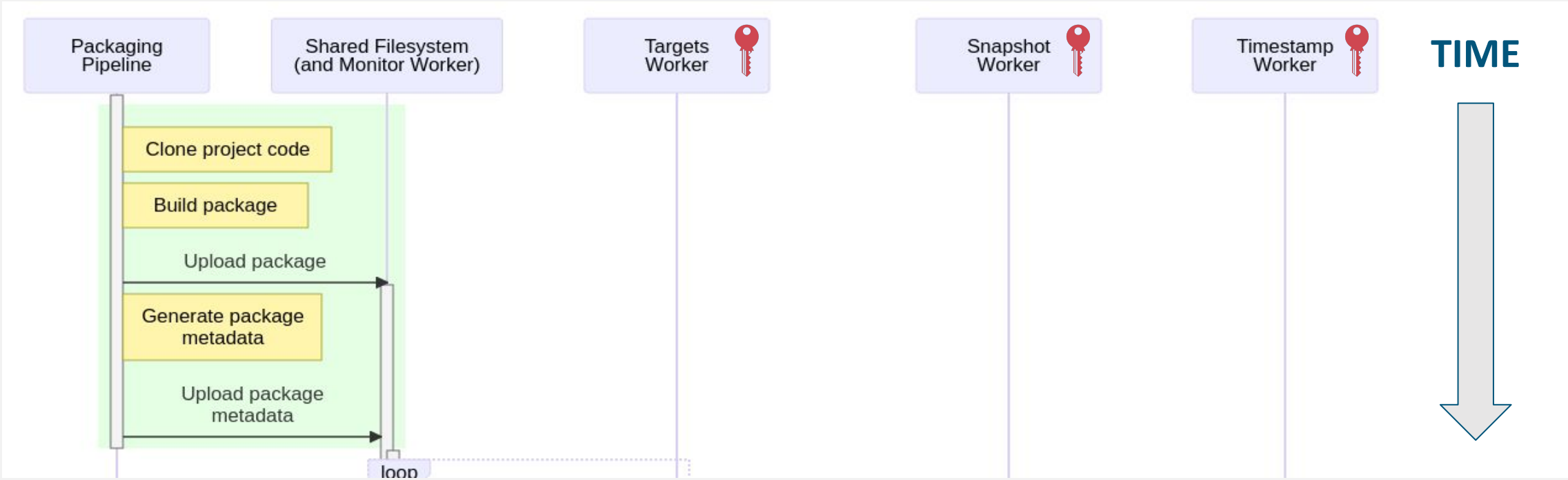
Worker daemons:

- `targets-worker` signs `targets.json`
- `snapshot-worker` signs `snapshot.json`
- `timestamp-worker` signs `timestamp.json`
- `monitor-worker` scans for new targets, periodically refreshes metadata expiry
- `root-worker` initializes TUF repository, generates online keypairs



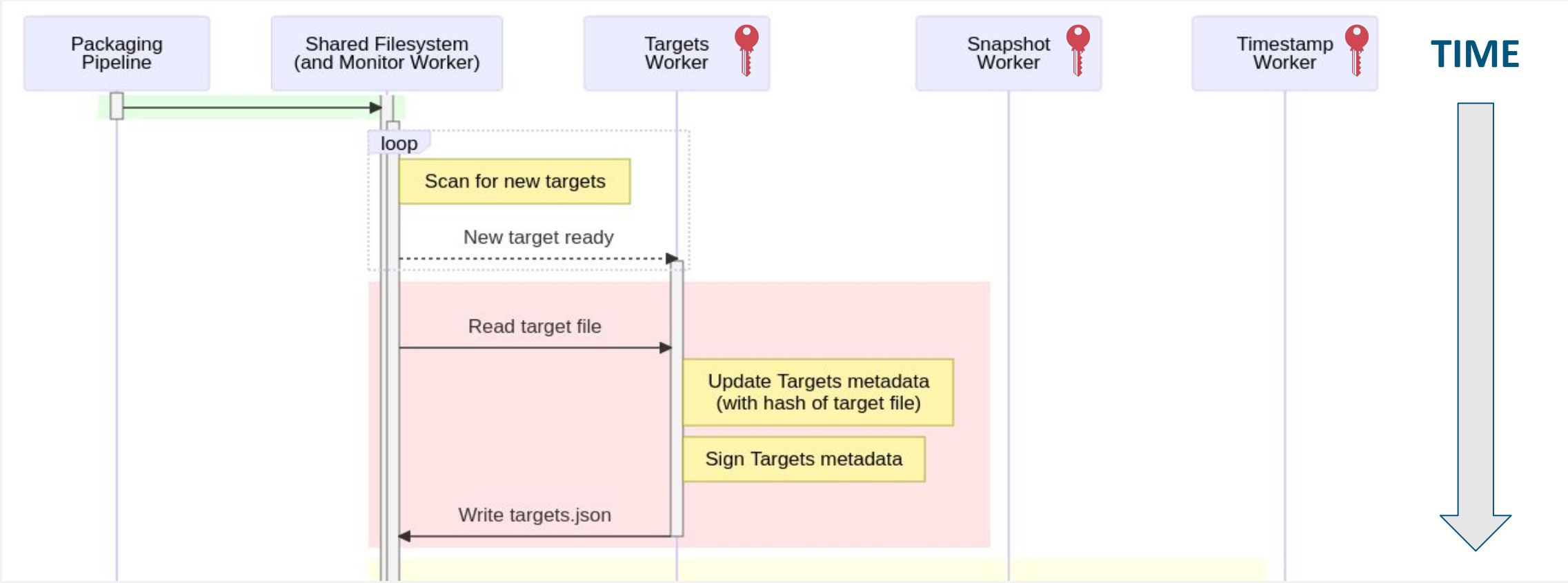
Rugged TUF Server

Packaging pipeline



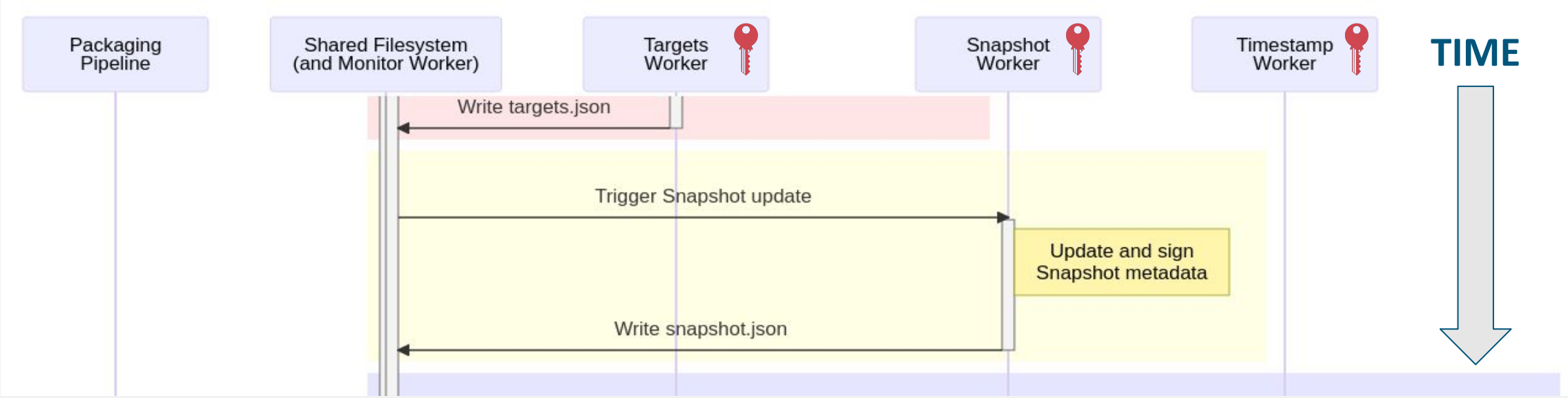
Rugged TUF Server

Targets worker



Rugged TUF Server

Snapshot worker



Rugged TUF Server

Timestamp worker

Client-side TUF Verification

PHP-TUF & Composer Integration Plugin

PHP-TUF Library

PHP-TUF is a PHP implementation of The Update Framework (TUF).

Primarily focused on supporting secure automated updates for PHP CMSes.

Development sponsored by **Acquia,** with support from **Drupal Association**, **TYPO3** & **Joomla.**

OSTIF security audit, in January 2024, found **no significant vulnerabilities**

Composer Plugin

PHP-TUF Composer Integration Plugin adds TUF security to Composer's package discovery process, and packages selected for download.

Expect a slowdown when TUF is enabled.

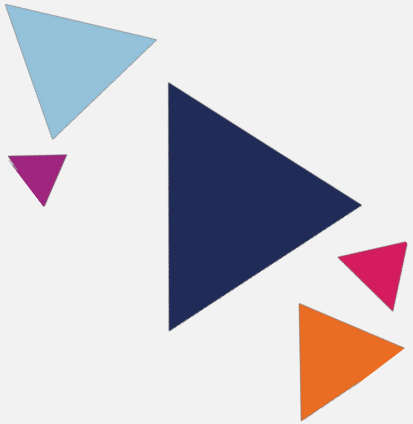
Development sponsored by **Acquia,** with support from **Drupal Association**, **TYPO3** & **Joomla.**

OSTIF security audit, in January 2024, found **no significant vulnerabilities**

TUF

Current Status

Drupal Automatic Updates status



- Server-side components are in production & need testing
- Rugged and PHP-TUF have been formally security reviewed
- **Ready for testing drupal.org/project/automatic_updates**
- **Package manager module → Drupal core**

  drupal.org/i/3319030
- Slack #autoupdates

DrupalCon
BARCELONA2024
24-27 SEPTEMBER

Join us for contribution opportunities!

Mentored Contribution

27 September:
09:00 – 18:00
Room 111

First Time Contributor Workshop

24 September: 16:30 - 17:15
Room BoF 4 (121)
25 September: 11:30 - 12:15
Room BoF 4 (121)
27 September: 09:00 - 12:30
Room 111

General Contribution

24-26 September: 9:00 - 18:00
Area 1
27 September: 09 - 18:00
Room 112

#DrupalContributions

What did you think?

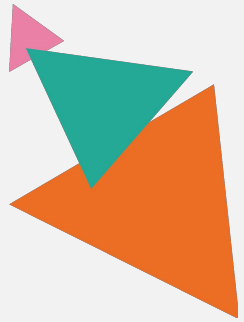
DrupalCon BARCELONA2024 24-27 SEPTEMBER

Please fill out the **Individual session** survey

(in the Mobile App using QR code)

Thank you!

Title

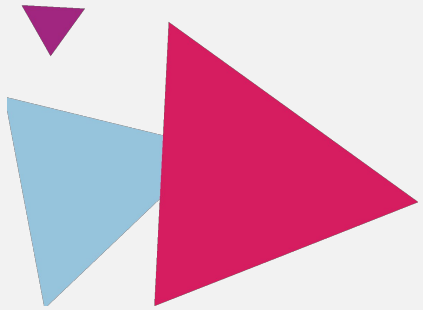


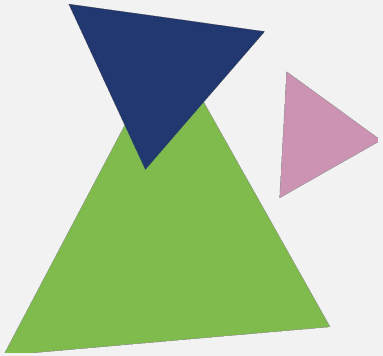
Title

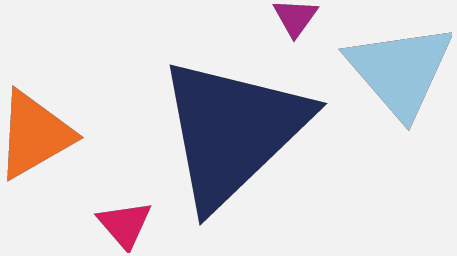
Title

Title


DrupalCon
BARCELONA2024
24-27 SEPTEMBER








DrupalCon
BARCELONA2024
24-27 SEPTEMBER

Title