Sui Provenance Suite

Deploy what you trust. Verify what you see.

zktx.io

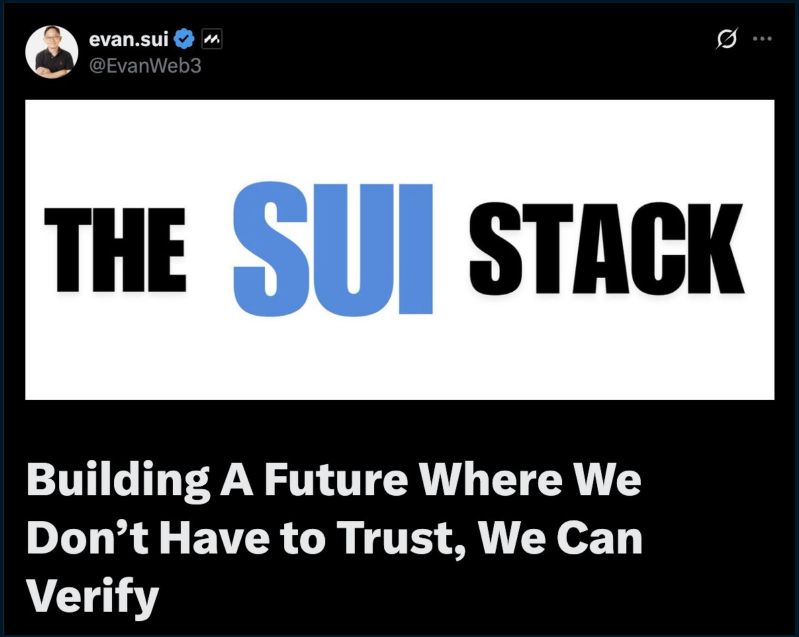
Can Others Verify What You Deploy?

“Trust” is still the default in Web3

- Can you prove your frontend hasn’t been tampered with?
- Can others verify your Move package came from your repo?
- Hashes alone don’t tell the full story.

➡ There’s a missing link between GitHub and on-chain code.

Don’t Trust. Verify.



evan.sui

@EvanWeb3

Building A Future Where We Don’t Have to Trust, We Can Verify

https://x.com/EvanWeb3/status/1905728305569128649

- Trust is not a UX problem.

  It’s **an infrastructure design problem**.

- **Provenance**, not as a feature — but as a **new standard for blockchains**.

- **First** to make end-to-end **provenance native — on the Sui Stack**.

The Toolkit

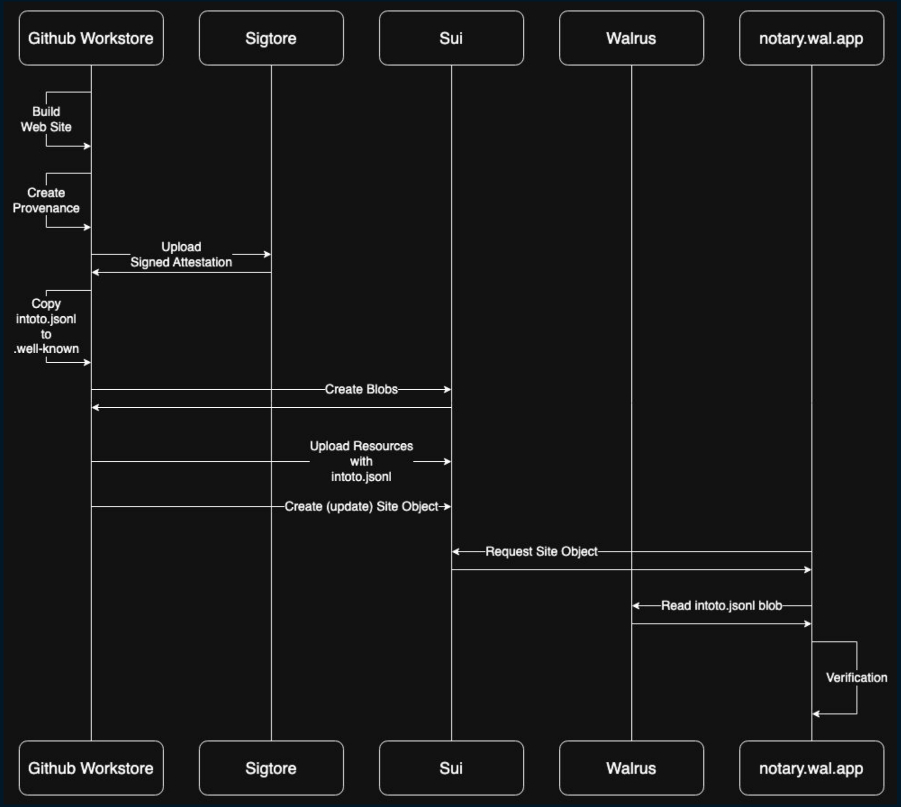
Sui Provenance Suite

- **Walrus Sites Provenance** – verifiable frontend deployment.
- **Sui MVR Provenance** – verifiable Move package registry.
- **Notary** – browser-based verification UI.
- **GitSigner** – secure PIN-based external signing.

➡ One suite, Full-stack provenance, and live.

Walrus Sites Provenance

1. Website (GitHub)
2. GitHub Actions (CI)
    a. npm run build
    b. Sigstore signs → generate *site.intoto.jsonl*
    c. Walrus sites deploy (*resources + site.intoto.jsonl*)
3. Verified on ***notary.wal.app***

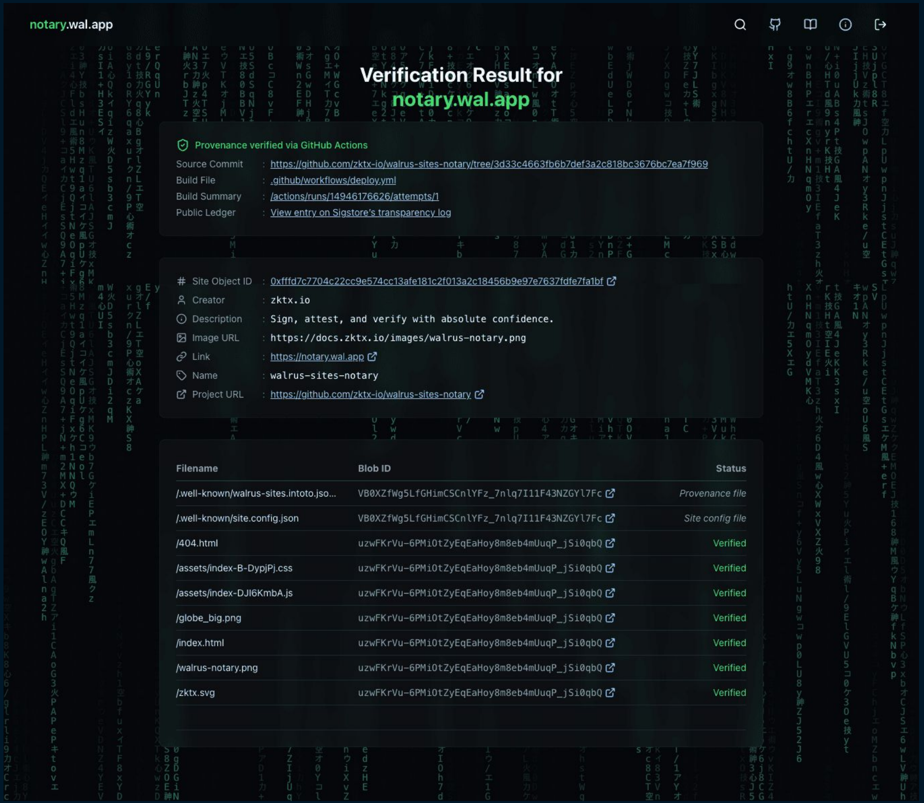


Self-Verifying Example

Case Study: notary.wal.app

- Notary doesn’t just verify others — it proves its own deployment.
- It is built, signed, and deployed via GitHub Actions using the same Walrus + Sigstore pipeline.
- Its .intoto.jsonl provenance is public and verifiable on itself.

➡ A trust tool that proves it can be trusted.



notary.wal.app

Verification Result for
notary.wal.app

Provenance verified via GitHub Actions

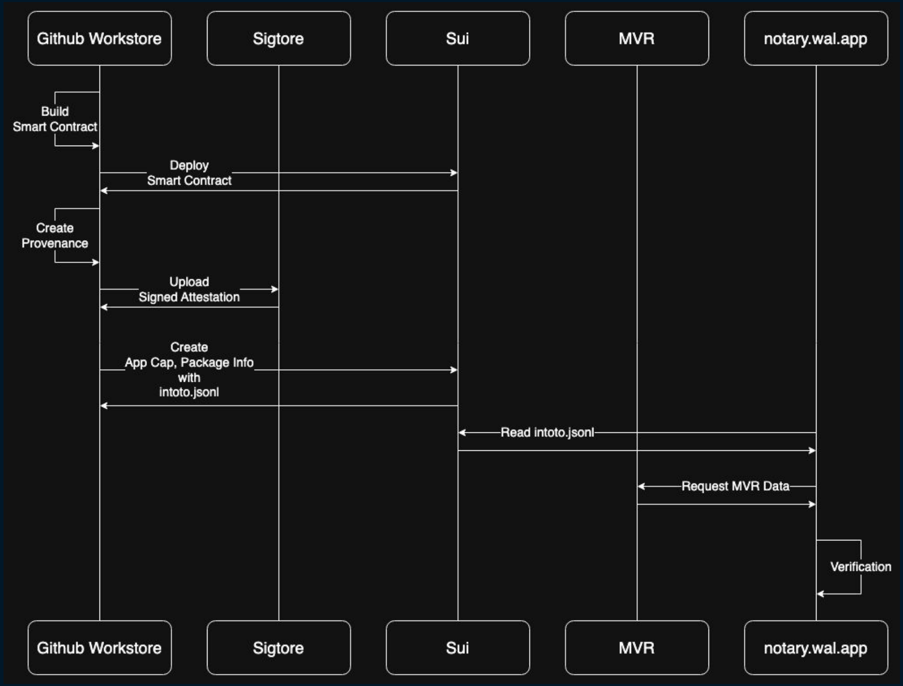
Source Commit	: https://github.com/zktx-io/walrus-sites-notary/tree/3d33c4663fb6b7def3a2c818bc3676bc7ea7f969
Build File	: .github/workflows/deploy.yml
Build Summary	: /actions/runs/14946176626/attempts/1
Public Ledger	: View entry on Sigstore’s transparency log

# Site Object ID	: 0xfffd7c7704c22cc9e574cc13afe181c2f013a2c18456b9e97e7637fdfe7fa1bf
Creator	: zktx.io
Description	: Sign, attest, and verify with absolute confidence.
Image URL	: https://docs.zktx.io/images/walrus-notary.png
Link	: https://notary.wal.app
Name	: walrus-sites-notary
Project URL	: https://github.com/zktx-io/walrus-sites-notary

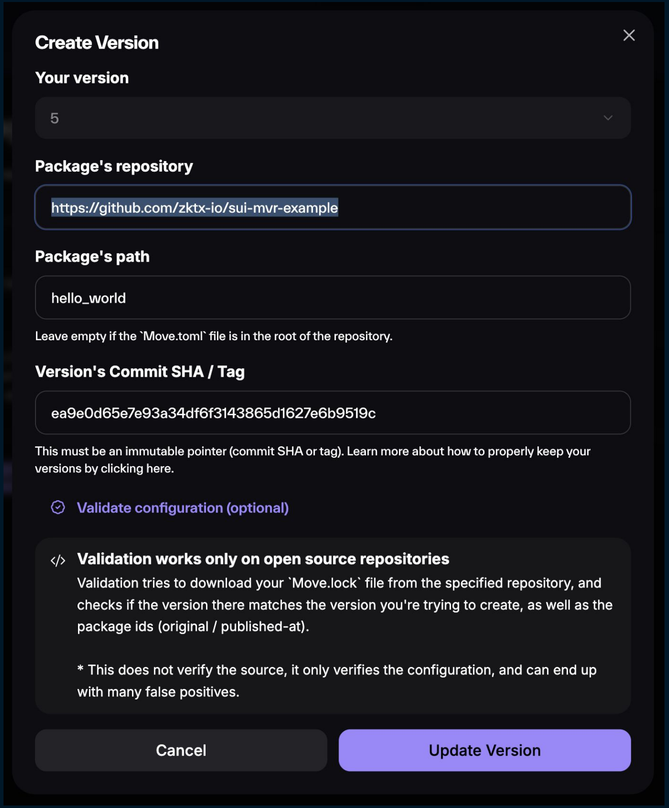
Filename	Blob ID	Status
/.well-known/walrus-sites.intoto.jso...	VB0XZfWg5LfGHimCSCnlYFz_7nlq7I11F43NZGYl7Fc	*Provenance file*
/.well-known/site.config.json	VB0XZfWg5LfGHimCSCnlYFz_7nlq7I11F43NZGYl7Fc	*Site config file*
/404.html	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified
/assets/index-B-DypjPj.css	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified
/assets/index-DJI6KmbA.js	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified
/globe_big.png	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified
/index.html	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified
/walrus-notary.png	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified
/zktx.svg	uzwFKrVu-6PMiOtZyEqEaHoy8m8eb4mUuqP_jSi0qbQ	Verified

Sui MVR Provenance

1. Move Package (GitHub)
2. GitHub Actions (CI)
   a. Build with sui move build → generate bytecode.dump.json.
   b. Sign and deploy move package.
   c. Sigstore signs → generate *mvr.intoto.jsonl*.
   d. MVR Register Move Package (*tx digest + mvr.intoto.jsonl*)
3. Verified on ***notary.wal.app***



From Form to Cryptographic Provenance



Create Version

Your version

5

Package's repository

https://github.com/zktx-io/sui-mvr-example

Package's path

hello_world

Leave empty if the `Move.toml` file is in the root of the repository.

Version's Commit SHA / Tag

ea9e0d65e7e93a34df6f3143865d1627e6b9519c

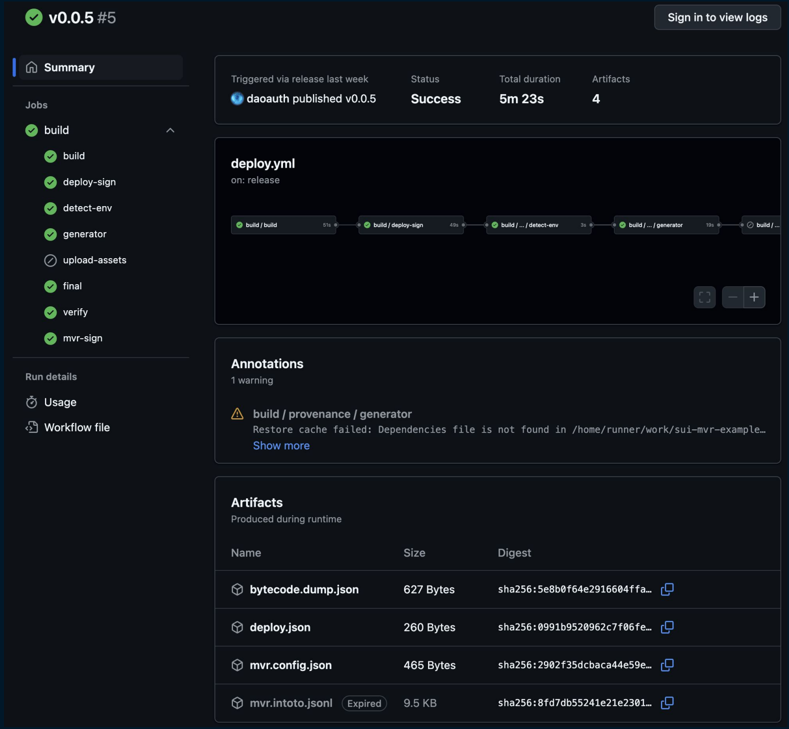
This must be an immutable pointer (commit SHA or tag). Learn more about how to properly keep your versions by clicking here.

Validate configuration (optional)

Validation works only on open source repositories

Validation tries to download your `Move.lock` file from the specified repository, and checks if the version there matches the version you're trying to create, as well as the package ids (original / published-at).

* This does not verify the source, it only verifies the configuration, and can end up with many false positives.



✓ v0.0.5 #5

Jobs

- ✓ build
  - ✓ build
  - ✓ deploy-sign
  - ✓ detect-env
  - ✓ generator
  - ⊘ upload-assets
  - ✓ final
  - ✓ verify
  - ✓ mvr-sign

Run details

Triggered via release last week	Status	Total duration	Artifacts
daoauth published v0.0.5	Success	5m 23s	4

deploy.yml

on: release

Annotations

1 warning

build / provenance / generator

```
Restore cache failed: Dependencies file is not found in /home/runner/work/sui-mvr-example…
```

Show more

Artifacts

Produced during runtime

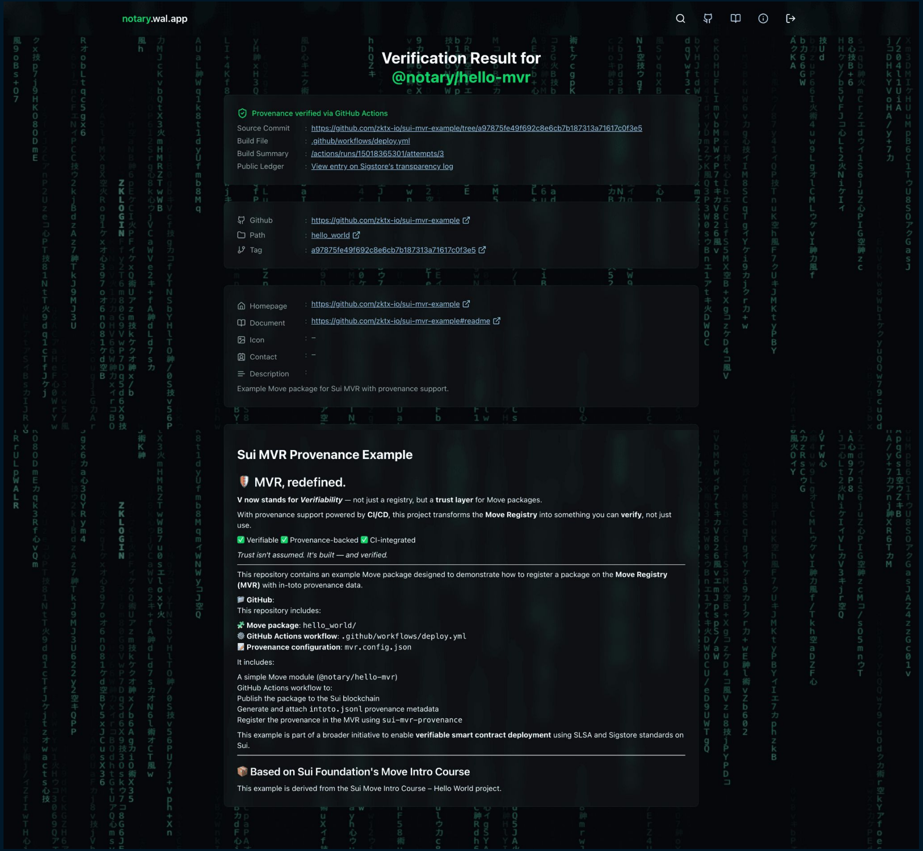
Name	Size	Digest
bytecode.dump.json	627 Bytes	sha256:5e8b0f64e2916604ffa…
deploy.json	260 Bytes	sha256:0991b9520962c7f06fe…
mvr.config.json	465 Bytes	sha256:2902f35dcbaca44e59e…
mvr.intoto.jsonl Expired	9.5 KB	sha256:8fd7db55241e21e2301…

Beyond Registration

MVR = Move ***Verifiable*** Registry

- “V” is for Verifiable.
- Every package is signed, linked to commits, and reproducible.
- Metadata is traceable across GitHub and chain.

➡ This isn’t just a registry — it’s a trust layer.



notary.wal.app

Verification Result for @notary/hello-mvr

Provenance verified via GitHub Actions

Source Commit : https://github.com/zktx-io/sui-mvr-example/tree/a97875fe49f692c8e6cb7b187313a71617c0f3e5
Build File : .github/workflows/deploy.yml
Build Summary : /actions/runs/15018365301/attempts/3
Public Ledger : View entry on Sigstore’s transparency log

Github : https://github.com/zktx-io/sui-mvr-example
Path : hello_world
Tag : a97875fe49f692c8e6cb7b187313a71617c0f3e5

Homepage : https://github.com/zktx-io/sui-mvr-example
Document : https://github.com/zktx-io/sui-mvr-example#readme
Icon : –
Contact : –
Description :
Example Move package for Sui MVR with provenance support.

Sui MVR Provenance Example

🛡️ MVR, redefined.

V now stands for *Verifiability* — not just a registry, but a **trust layer** for Move packages.

With provenance support powered by **CI/CD**, this project transforms the **Move Registry** into something you can **verify**, not just use.

✅ Verifiable ✅ Provenance-backed ✅ CI-integrated

Trust isn't assumed. It's built — and verified.

This repository contains an example Move package designed to demonstrate how to register a package on the **Move Registry (MVR)** with in-toto provenance data.

📁 **GitHub**:
This repository includes:

🧩 **Move package**: `hello_world/`
⚙️ **GitHub Actions workflow**: `.github/workflows/deploy.yml`
📝 **Provenance configuration**: `mvr.config.json`

It includes:

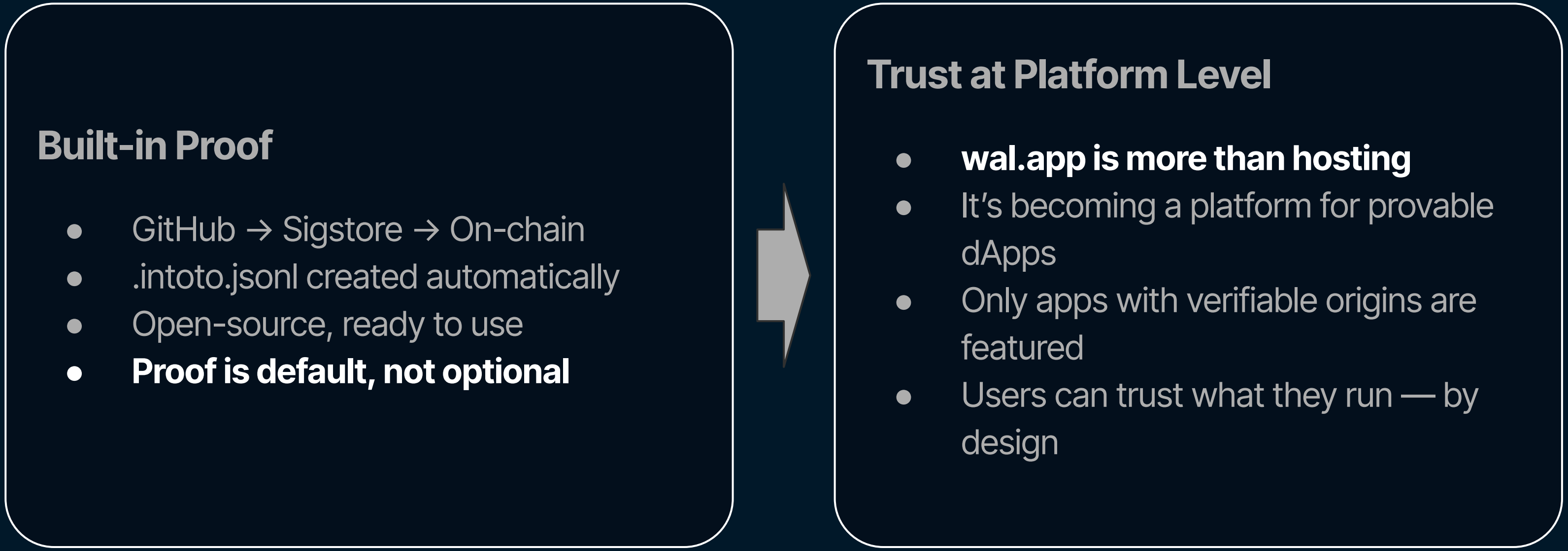
A simple Move module (`@notary/hello-mvr`)
GitHub Actions workflow to:
Publish the package to the Sui blockchain
Generate and attach `intoto.jsonl` provenance metadata
Register the provenance in the MVR using `sui-mvr-provenance`

This example is part of a broader initiative to enable **verifiable smart contract deployment** using SLSA and Sigstore standards on Sui.

📦 Based on Sui Foundation's Move Intro Course

This example is derived from the Sui Move Intro Course – Hello World project.

Verifiability is Infrastructure



Built-in Proof

- GitHub → Sigstore → On-chain
- .intoto.jsonl created automatically
- Open-source, ready to use
- **Proof is default, not optional**

Trust at Platform Level

- **wal.app is more than hosting**
- It’s becoming a platform for provable dApps
- Only apps with verifiable origins are featured
- Users can trust what they run — by design

wal.app & MVR is where trust begins — with **provenance**, by default.

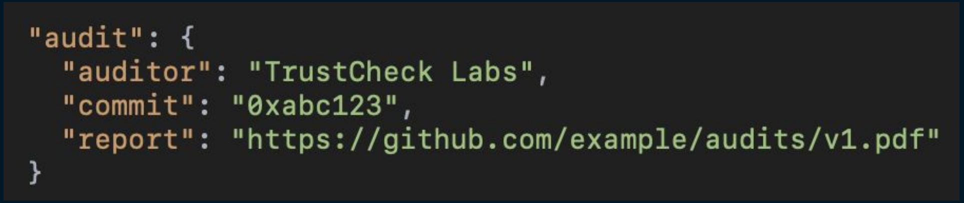
Proof Doesn’t End at Build

We can go further:

- **Audits** can be registered as **metadata** alongside the .intoto.jsonl file in MVR.

This metadata includes:

- Who audited it
- Which commit was reviewed
- Link to the public audit report



```
"audit": {
  "auditor": "TrustCheck Labs",
  "commit": "0xabc123",
  "report": "https://github.com/example/audits/v1.pdf"
}
```

➡ This way, expert reviews become part of the on-chain trust layer — verifiable, inspectable, and tamper-proof.

Trust That Grows, Not Freezes

Technical proof is only the beginning. Trust needs to grow — not stay frozen in time.

We build a living layer of trust:

- Bounty programs for continuous review
- Developer–user challenges to test and improve code

➡ This turns MVR into a dynamic trust ecosystem — where trust evolves, not just gets archived.

Links

- **End-to-end provenance tooling** for MVR.
  - https://github.com/zktx-io/sui-mvr-provenance
- A test repo that **passes validation without any real source code**.
  - https://github.com/zktx-io/sui-mvr-pass-but-fake
- **Frontend provenance pipeline** based on Walrus and Sigstore.
  - https://github.com/zktx-io/walrus-sites-provenance
- **Frontend UI for verifying provenance** files and site objects
  - https://github.com/zktx-io/walrus-sites-notary
- **Live explorer** to verify frontend provenance interactively
  - https://notary.wal.app

Sui Provenance Suite

Deploy what you trust. Verify what you see.

zktx.io