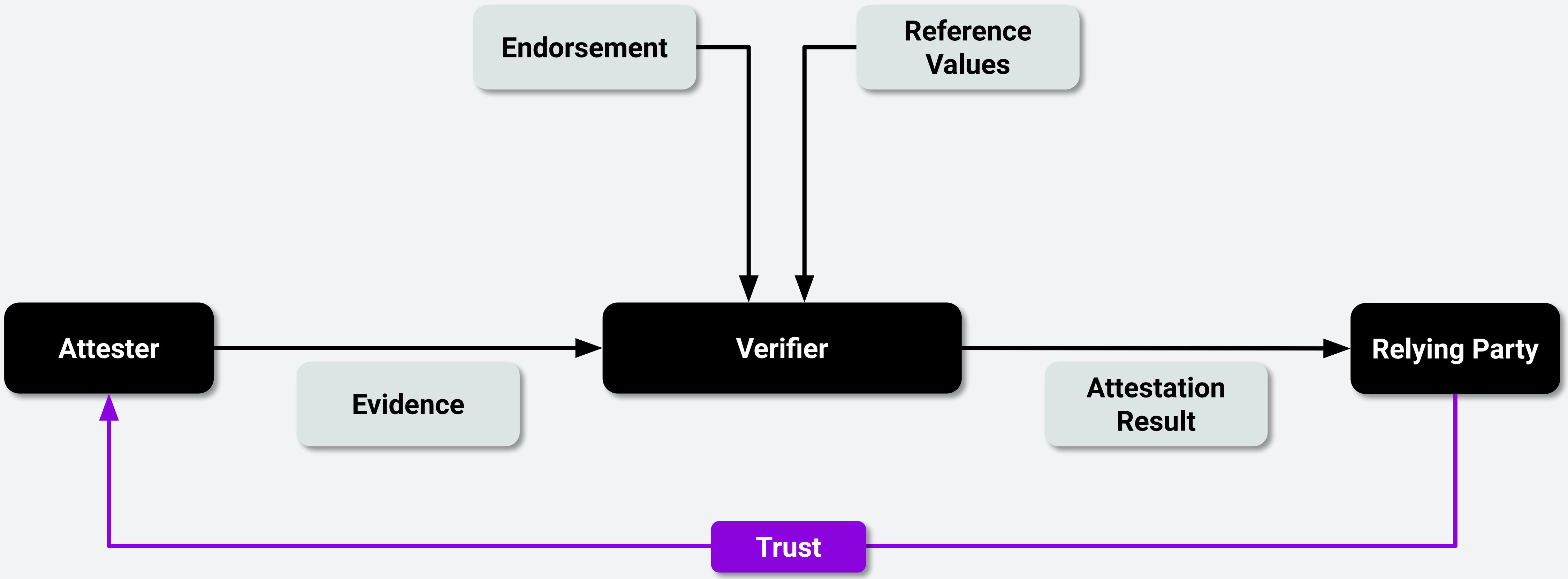
EDGELESS SYSTEMS

Reproducible builds for confidential computing: Why remote attestation is worthless without it

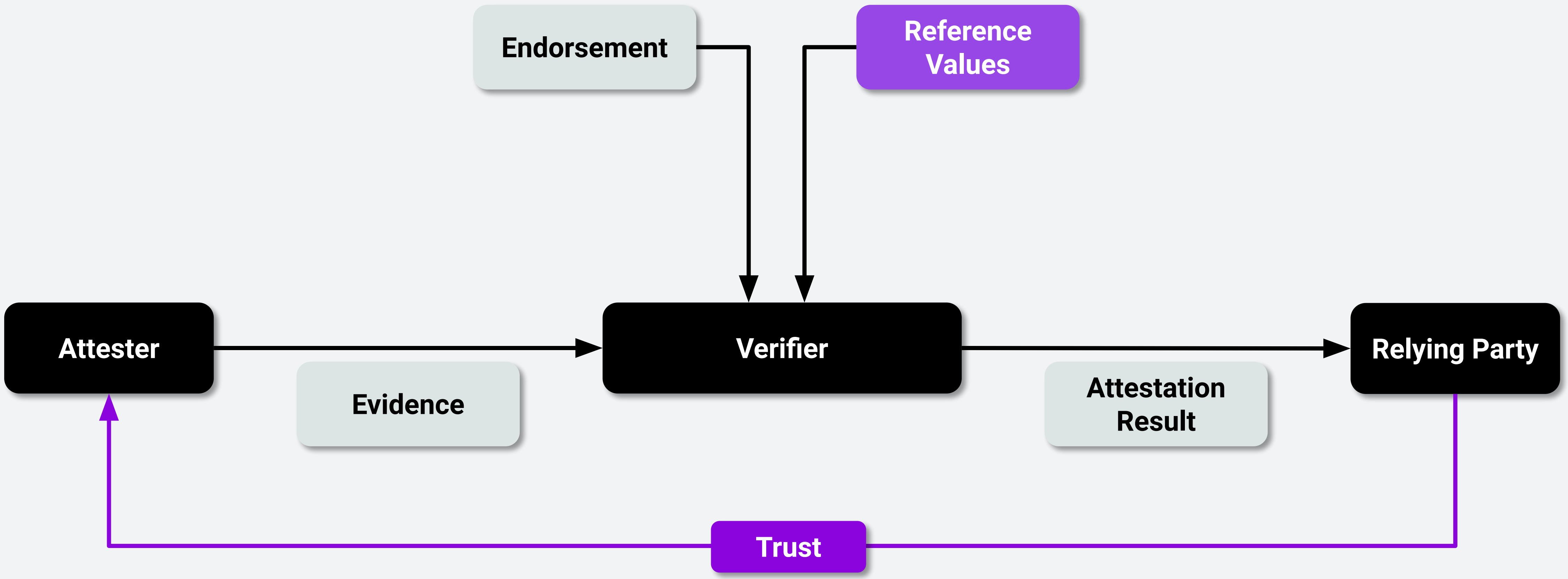
Confidential Computing

- We trust no one!
- Well, beside hardware manufacturer
- Need to establishing trust in the rest
- Remote attestation

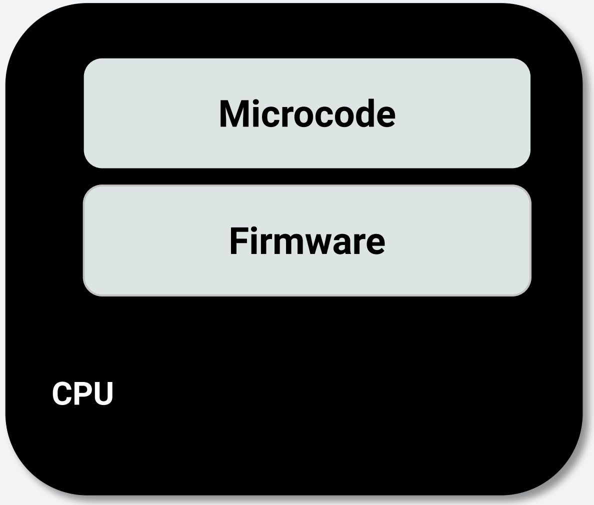
RFC 9334: Remote ATtestation procedureS (RATS)

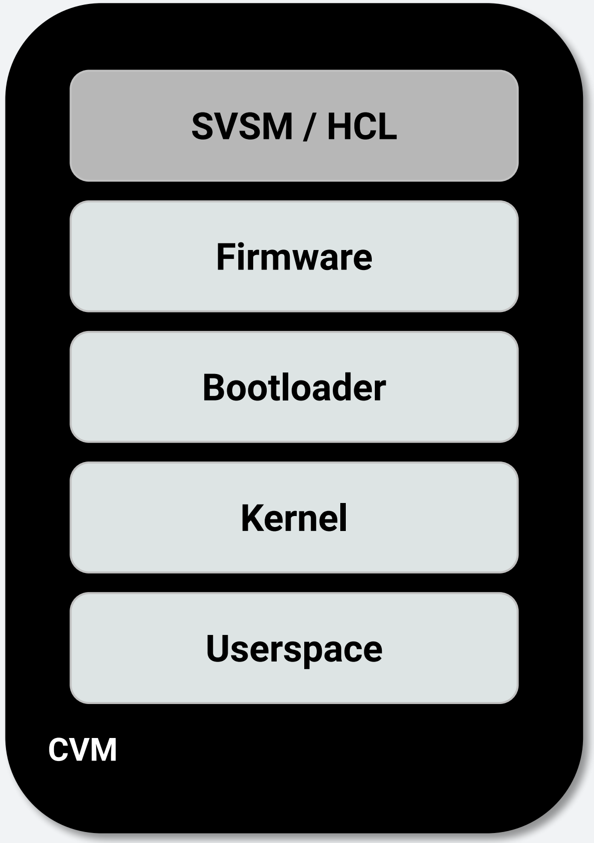


Reference values in Confidential Computing

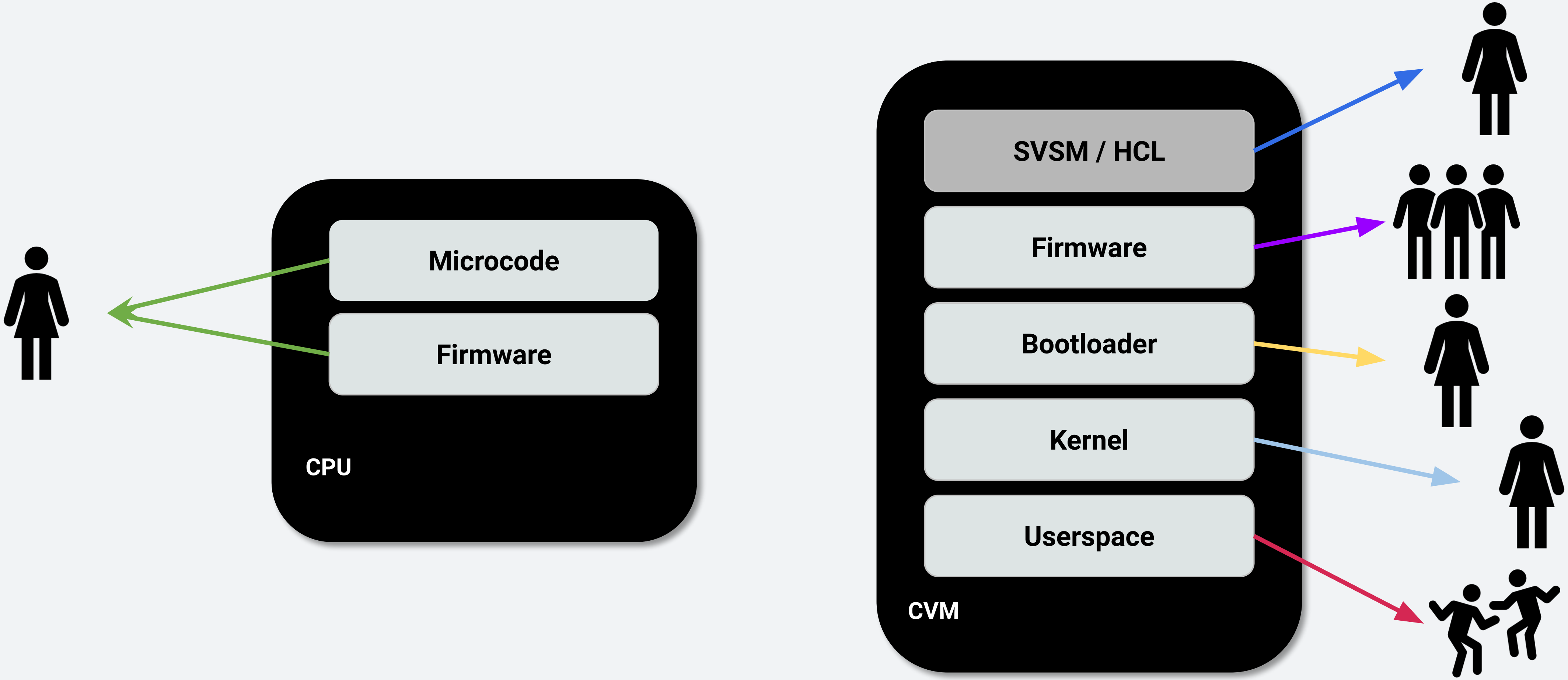


What’s part of my Trusted Computing Base?





Who's part of my Trusted Computing Base?



Remote attestation without reproducible builds

- Every trusted software vendor can run an attack
  - Delivers reference value of malicious binary
  - We can only check the authenticity
  - No insight what code is running
- What if we build everything from source?
  - We are the remaining software vendor that needs to be trusted
- Actual goal: attestation through the *end-user*

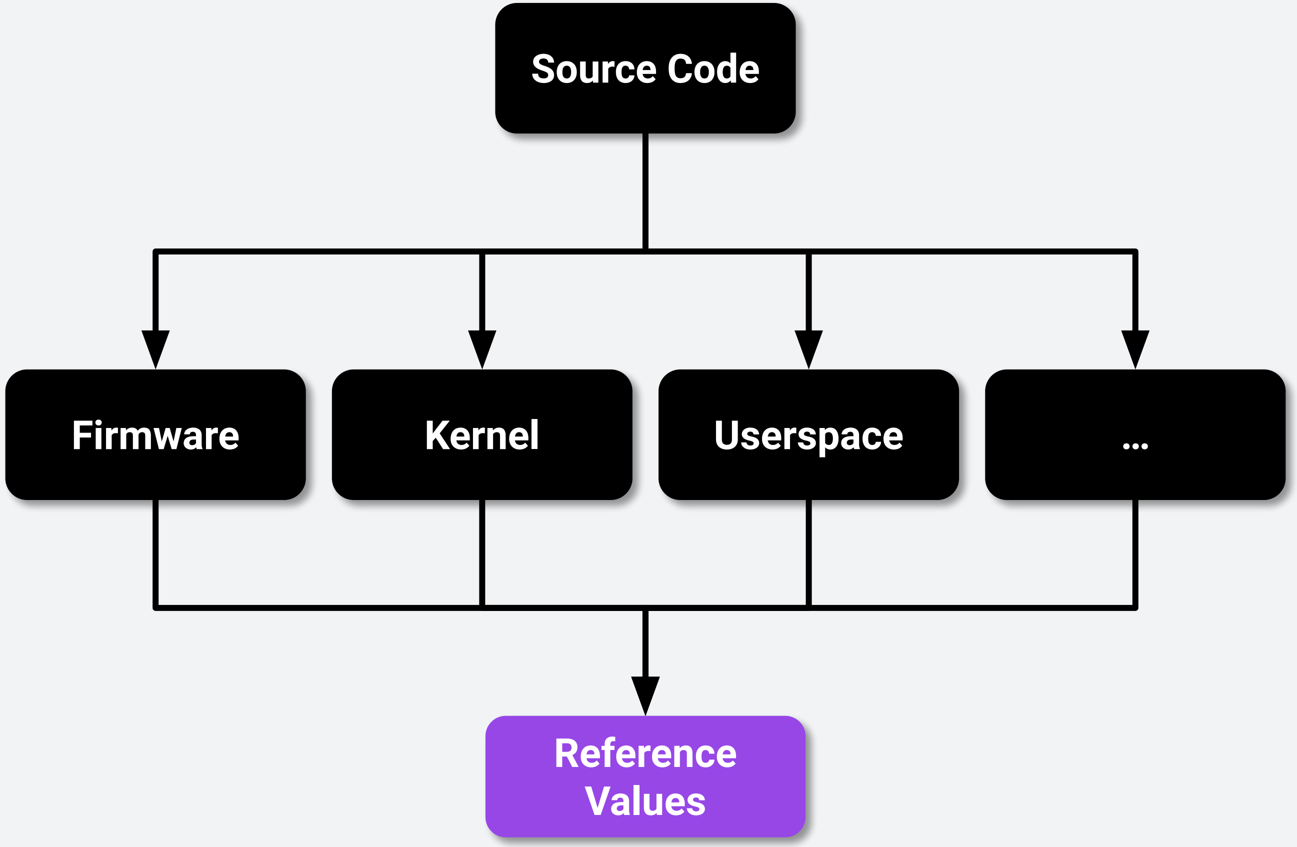
Reproducible Builds

“Reproducible builds are a set of software development practices that create an independently-verifiable path from source to binary code.”

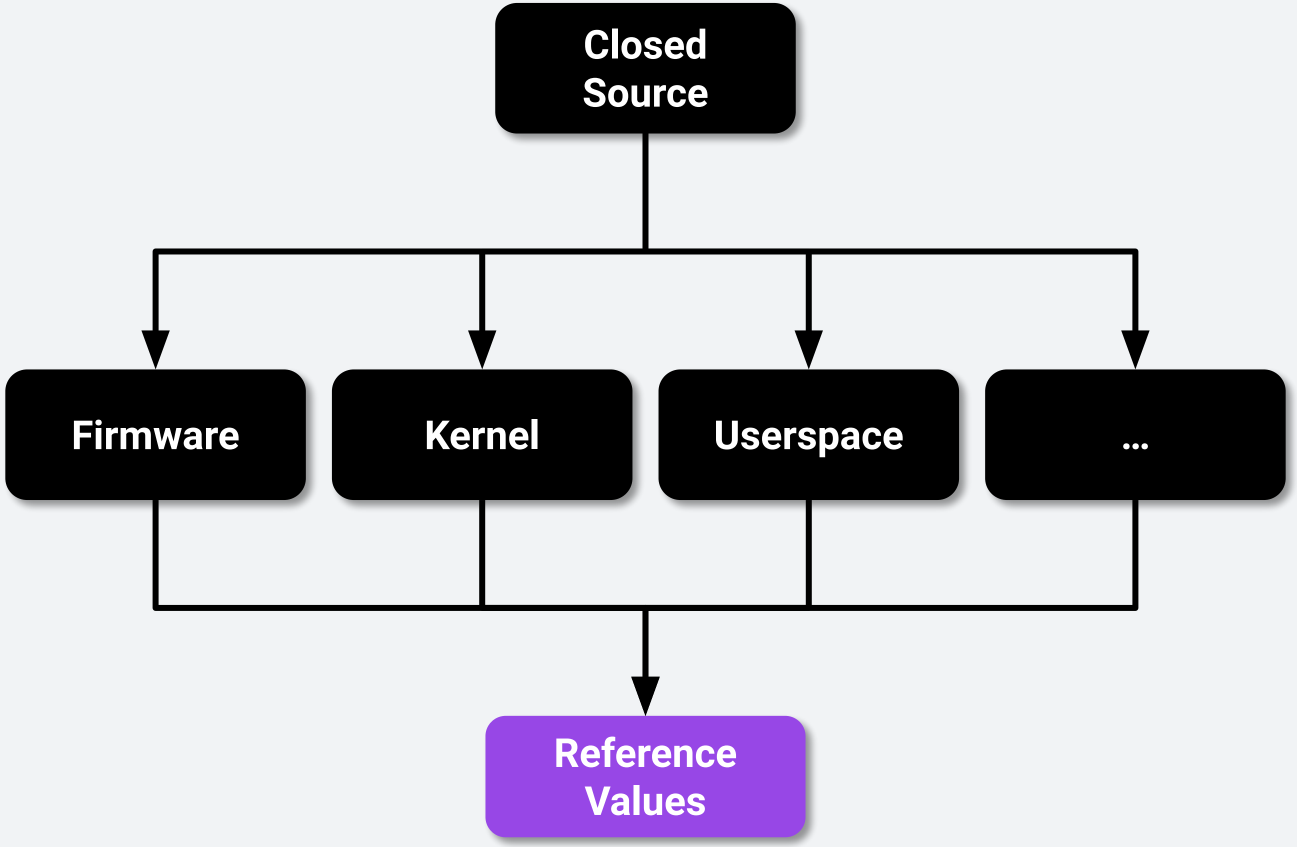
Reference values in Trusted Computing - Expectation

```
referenceValues = f(sourceCode)
```

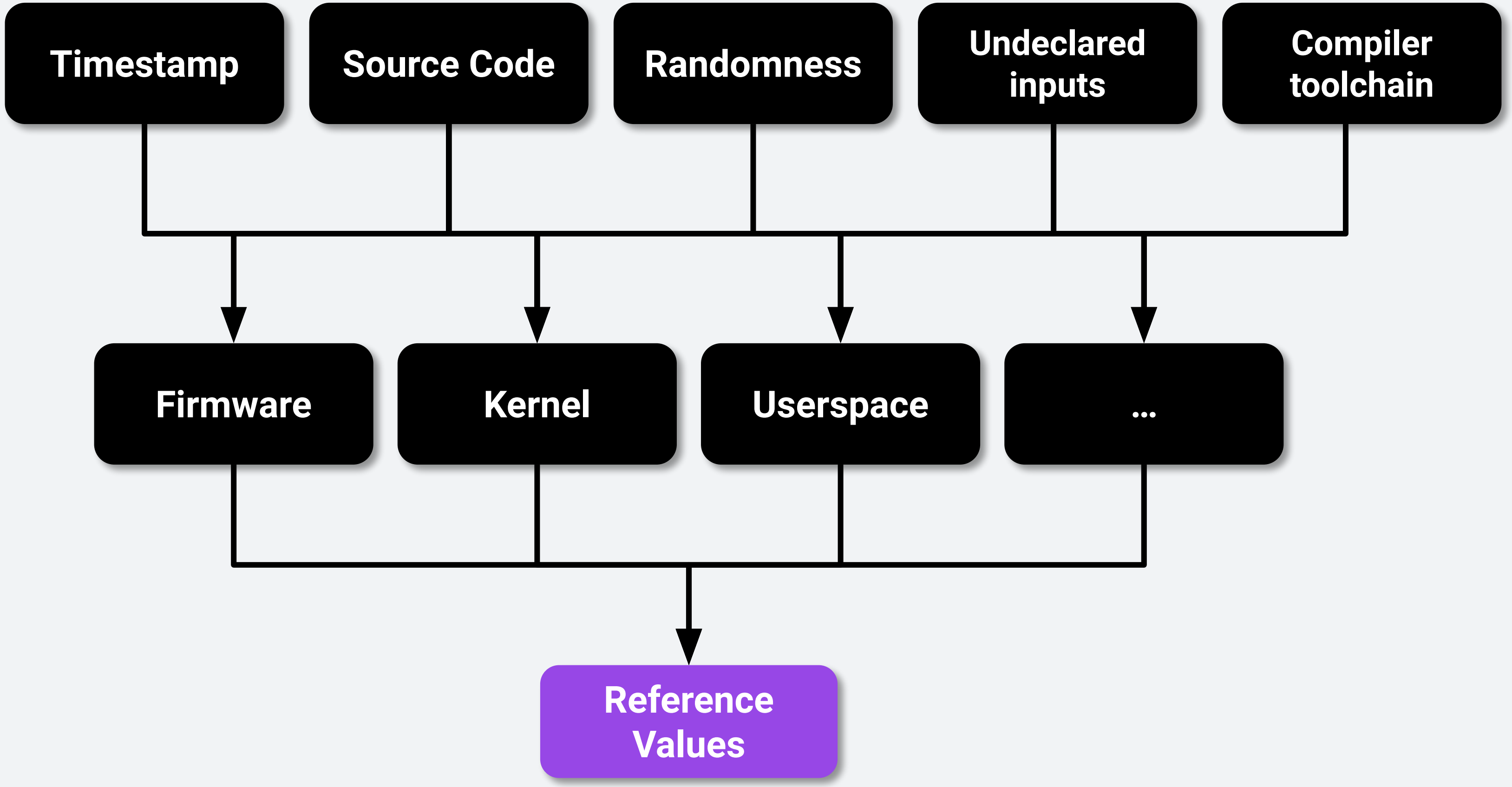
Reference values in Trusted Computing - Expectation



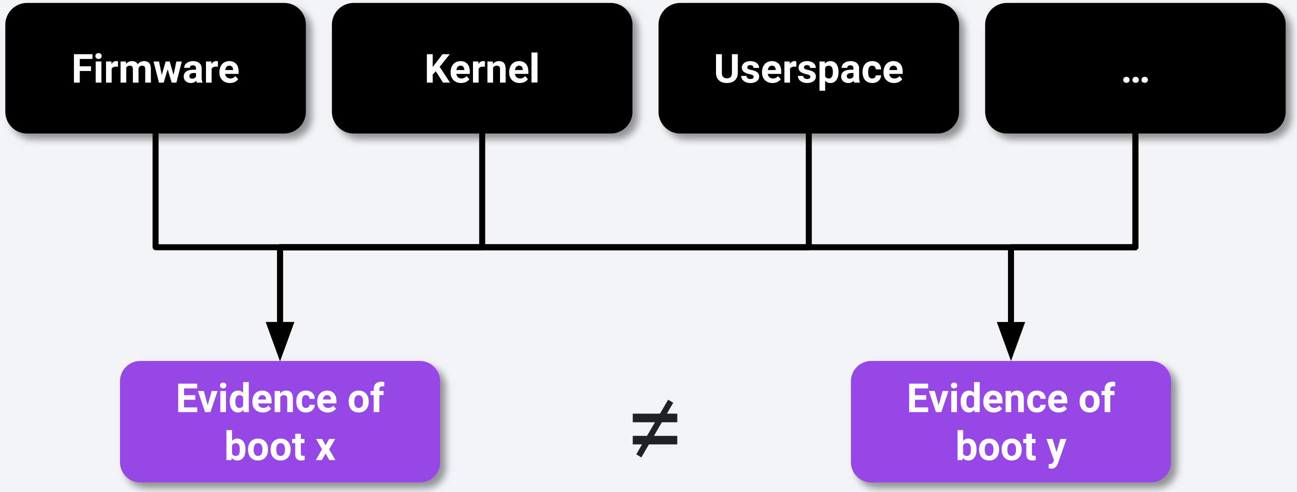
Reference values in Trusted Computing - Reality



Reference values in Trusted Computing - Reality

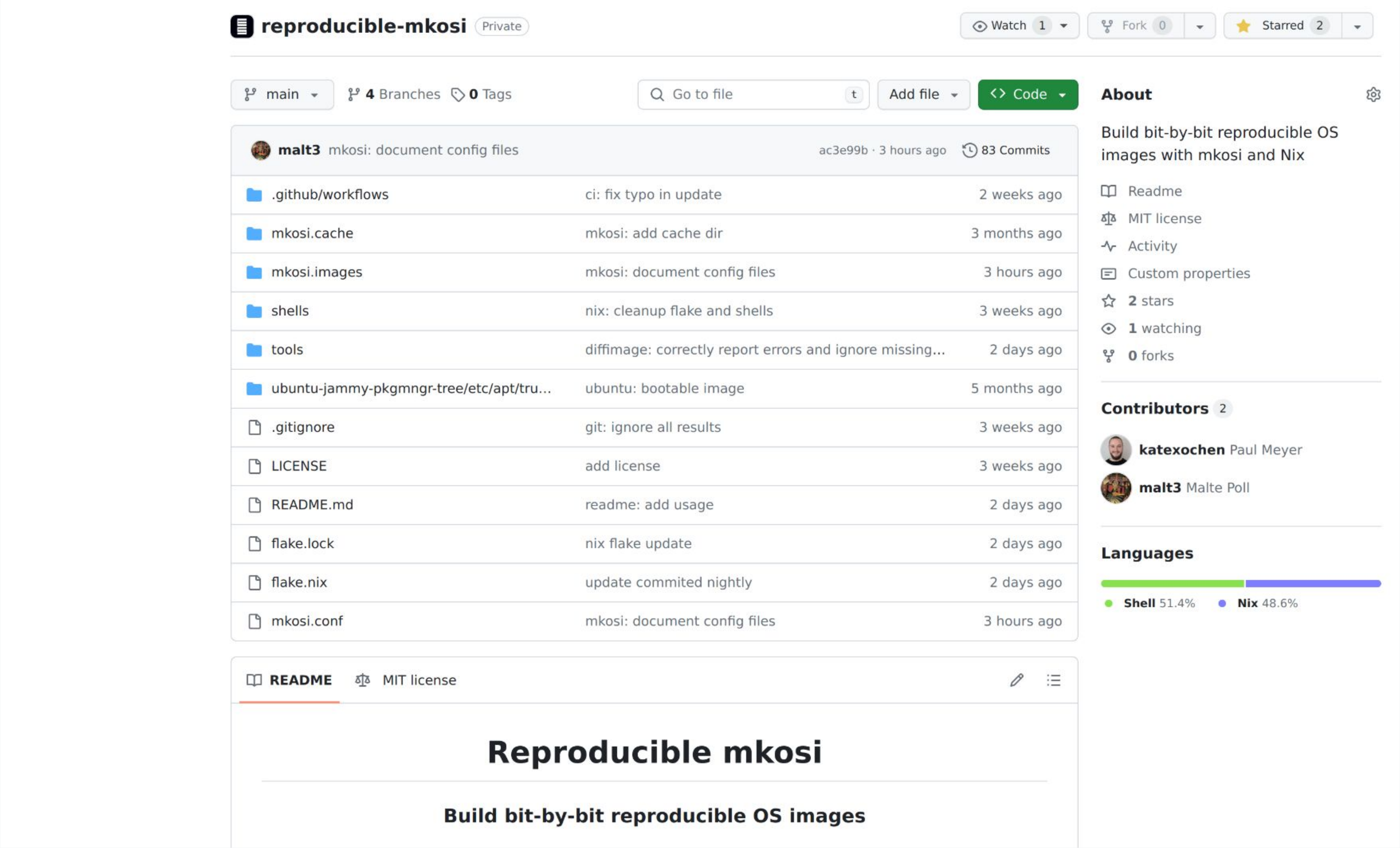


Reference values in Trusted Computing - Reality



Positive examples

- github.com/aws/uefi
- github.com/edgelesssys/constellation
- github.com/confidential-containers/cloud-api-adaptor (podvm-mkosi)
- github.com/edgelesssys/reproducible-mkosi



reproducible-mkosi Private

malt3 mkosi: document config files — ac3e99b · 3 hours ago — 83 Commits

.github/workflows	ci: fix typo in update	2 weeks ago
mkosi.cache	mkosi: add cache dir	3 months ago
mkosi.images	mkosi: document config files	3 hours ago
shells	nix: cleanup flake and shells	3 weeks ago
tools	diffimage: correctly report errors and ignore missing…	2 days ago
ubuntu-jammy-pkgmngr-tree/etc/apt/tru…	ubuntu: bootable image	5 months ago
.gitignore	git: ignore all results	3 weeks ago
LICENSE	add license	3 weeks ago
README.md	readme: add usage	2 days ago
flake.lock	nix flake update	2 days ago
flake.nix	update commited nightly	2 days ago
mkosi.conf	mkosi: document config files	3 hours ago

About

Build bit-by-bit reproducible OS images with mkosi and Nix

- Readme
- MIT license
- Activity
- Custom properties
- **2** stars
- **1** watching
- **0** forks

Contributors 2

- **katexochen** Paul Meyer
- **malt3** Malte Poll

Languages

- **Shell** 51.4%
- **Nix** 48.6%

README | MIT license

Reproducible mkosi

Build bit-by-bit reproducible OS images

How to build reproducible OS images

- Nix provides hermetic build tools
- Pin distro packages with lockfile
- Build in a Sandbox (mkosi, Nix(OS), Bazel)
- Restrict build actions (do not use Hashicorp Packer or Dockerfile)

Thanks!

- Learn about reproducible builds: reproducible-builds.org
- Provide an open software stack for CC
- Enable the community to reproduce reference values

Malte Poll

@malte@chaos.social

@malt3

github.com/malt3

Paul Meyer

@katexochen@infosec.exchange

@katexochen

github.com/katexochen