OpenSSF Community Day
NORTH AMERICA 2026

The Architecture of Accountability: Transparency in Software

Hayden Blauzvern
Google Open Source Security Team

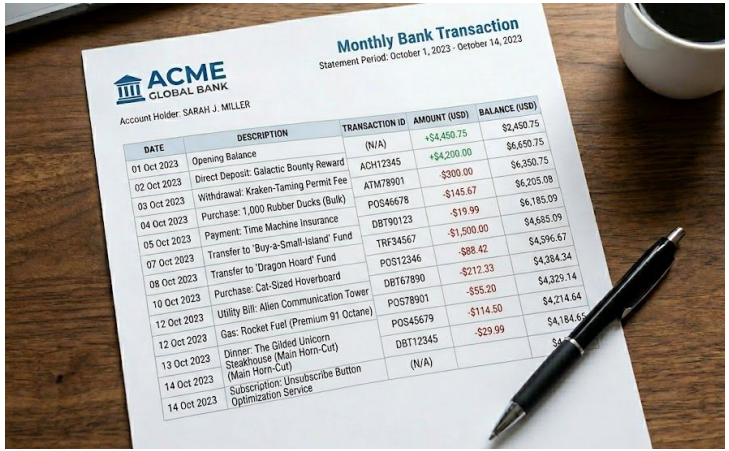
Motivating "Transparency"

Making a "claim"

1. Make a statement
2. Record that statement
3. Be held accountable for it
    a. Can't take it back
    b. Someone will verify it
4. Another party will rely on that claim

Bank Account Transactions

- How do I audit all deposits and withdrawals from my bank account?
- Can someone steal money without me noticing?
- Can my bank take money out without me noticing?



Logging into a Website

- How do I verify no one has my password?
- How about for every website I log into?
- What about if the site is compromised?
- Privacy concerns?

From theverge.com

Vim is Better than Emacs

- An obviously true statement
- Can I be wrong?
- Who is verifying this?

Defining "Transparency"

Discoverability & Auditability

Claimant Model

"There is a **Claimant** that makes a **Claim** that is relied upon by a **Believer** as a precondition to take an action they would not have taken if the claim was false. Claims are represented by signed **Statements**. The veracity of a Claim can be verified by a **Claim Verifier**, who will notify a **Claim Arbiter** of any false Claims." - Claimant Model

Claimant Model - Bank Transactions

Claimant	The bank
Claim	There exists a list of all deposits and withdrawals for a given bank account
Believer	Owner of bank account
Statement	A record of a transaction on a bank account
Verifier	Owner of bank account
Arbiter	Bank owner, the government

Claimant Model - Logging into websites

Claimant	The website?
Claim	All successful login attempts are recorded
Believer	The website
Statement	A record of the website, timestamp & username
Verifier	Human behind the username
Arbiter	Online forums, etc

Claimant Model - Logging into websites

Claimant	The user
Claim	All login attempts are recorded
Believer	The website
Statement	A record of the website, timestamp & username
Verifier	Human behind the username
Arbiter	Online forums, etc

Claimant Model - Vim > Emacs

Claimant	Me
Claim	Vim is better than Emacs <-- This is not falsifiable
Believer	Text editor enthusiasts
Statement	The sentence
Verifier	???
Arbiter	???

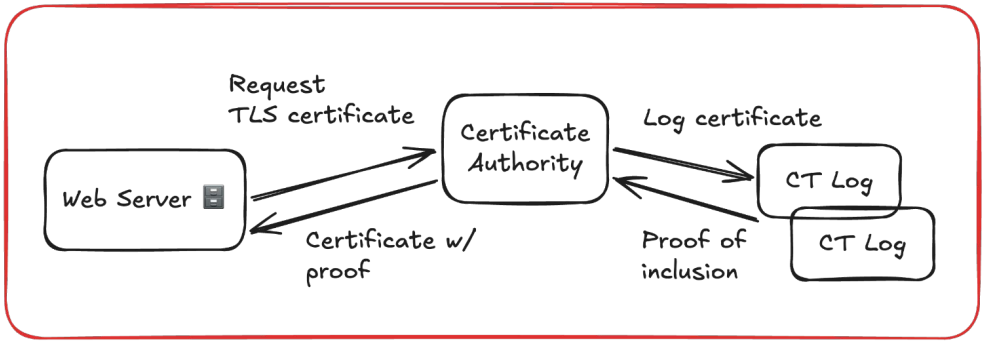
Building Blocks for a Transparent System

- Append-only list of statements, i.e. "log"
- Prove inclusion in the list to a Believer
  - Inclusion means that a Verifier can actually verify the claim
  - No inclusion means the statement shouldn't be relied upon
- Auditability for the Verifier
  - Efficiently verify that all relevant claims are correct

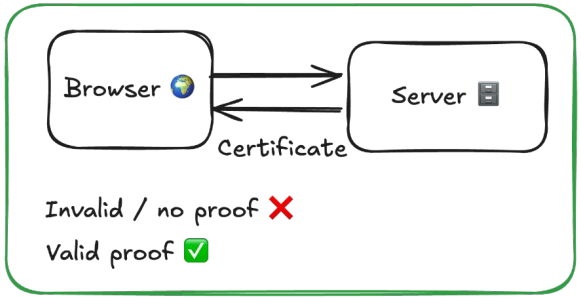
Real-World Examples

Certificate Transparency

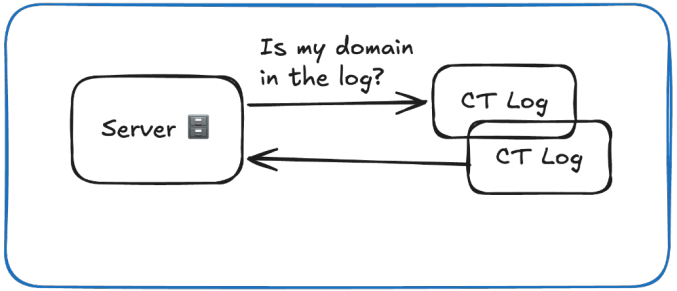
Claimant



Believer

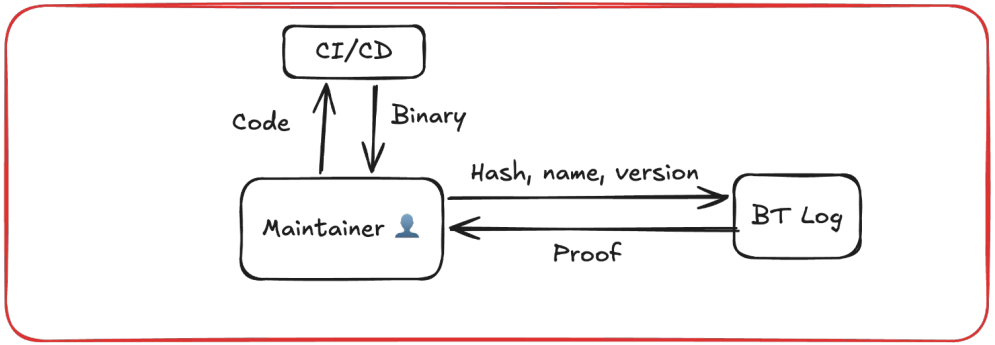


Verifier

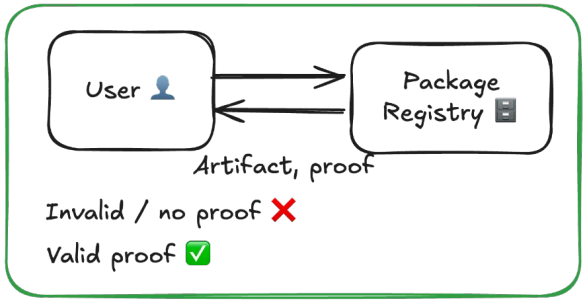


Binary Transparency

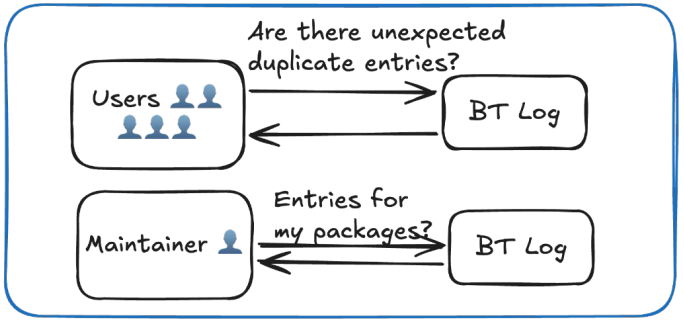
Claimant



Believer



Verifier

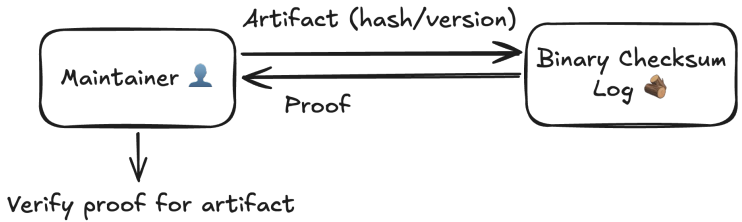


Binary Transparency

- Sigstore for key/identity usage transparency
- Go checksum database
- Android Pixel firmware transparency
- F-Secure USB Armory Drive

Binary Transparency

- Future work
  - More package registries: Registry & developer can guarantee that the same package that was uploaded is also distributed globally to all users

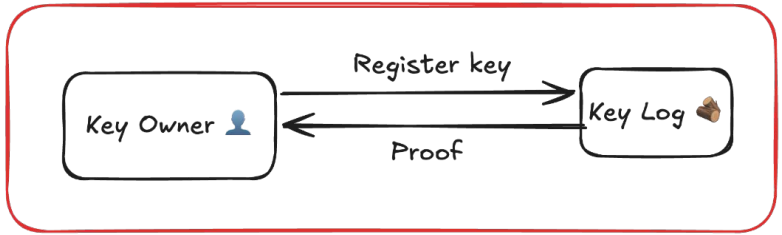


Binary Transparency

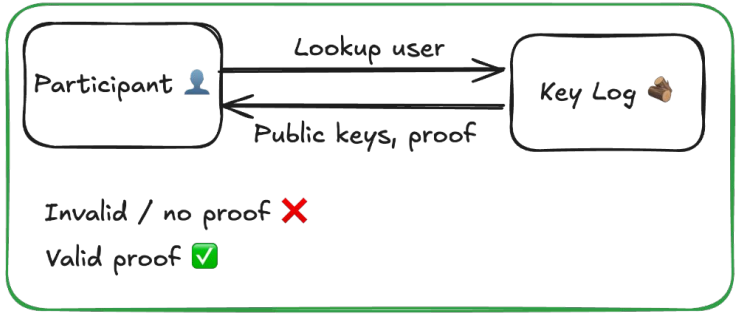
- Future work
  - More package registries: Registry & developer can guarantee that the same package that was uploaded is also distributed globally to all users
  - Web resource transparency
    - Web-based Code Assurance and Transparency (Freedom of the Press Foundation)
    - Web Application Integrity, Consistency, and Transparency (Cloudflare/Mozilla)

Key Transparency

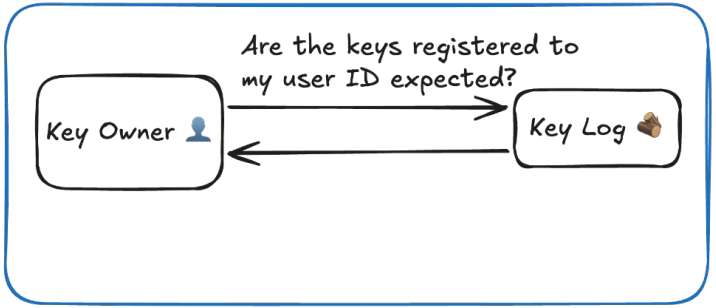
Claimant



Believer



Verifier

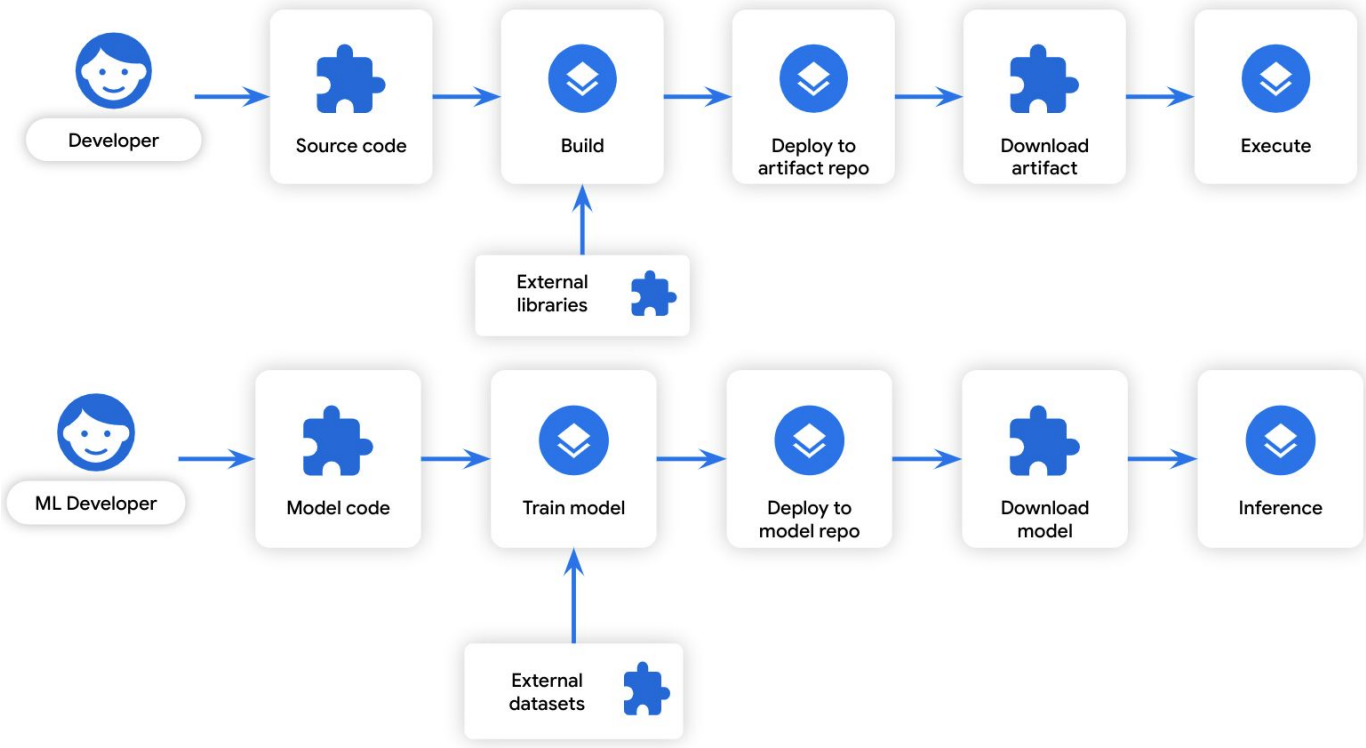


Key Transparency

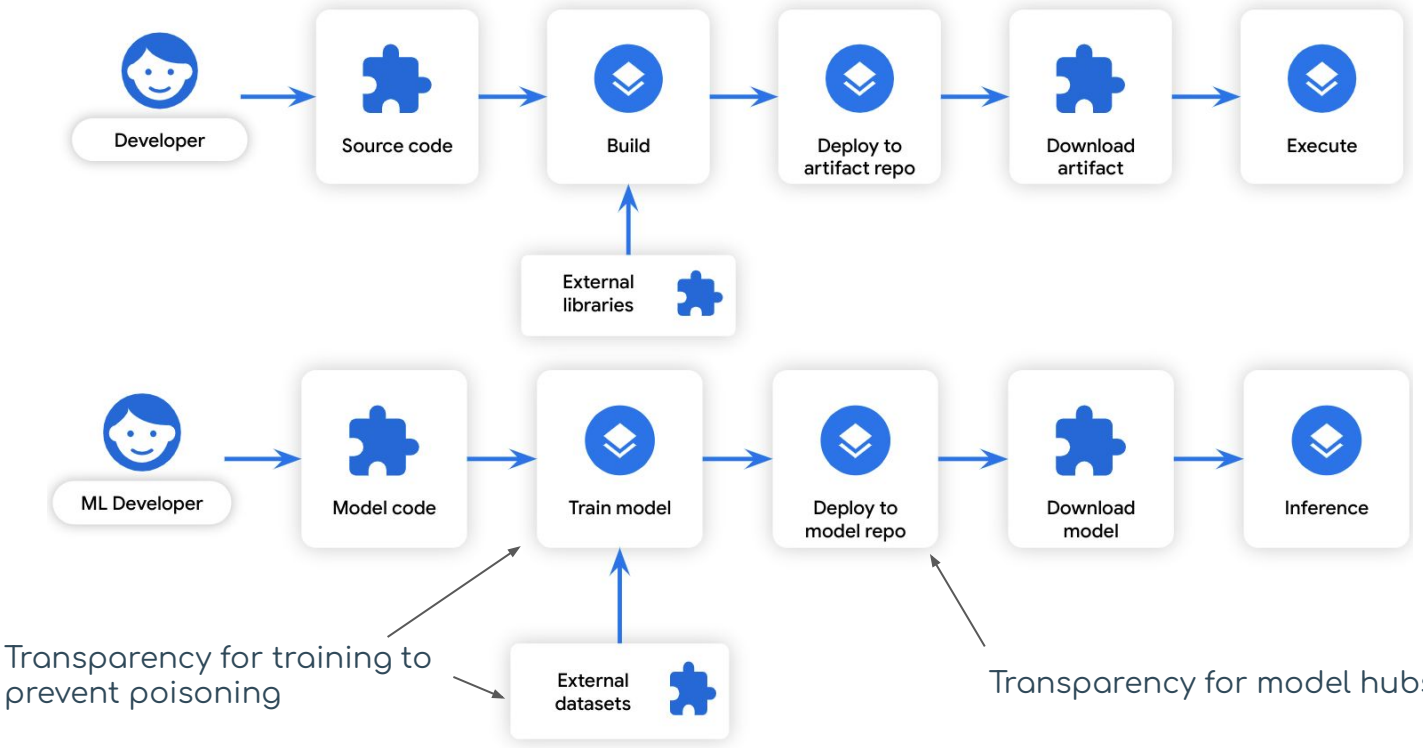
- Messaging applications for end-to-end encryption
  - Apple Messages for iMessage Contact Key Verification
  - WhatsApp
  - Signal
  - Proton Mail
- age public key server
  - https://keyserver.geomys.org/

Future Domains

AI Model Transparency



https://security.googleblog.com/2023/10/increasing-transparency-in-ai-security.html



https://security.googleblog.com/2023/10/increasing-transparency-in-ai-security.html

News Transparency

News Transparency

- Commit to article revisions
  - Research: https://dl.acm.org/doi/abs/10.1145/3579534
- Content authenticity
  - C2PA

Getting Involved

Getting Involved

- Learn more: https://transparency.dev/
- Slack: https://transparency.dev/slack/
- Standards: https://github.com/C2SP/C2SP
- Implementation: https://github.com/transparency-dev/tessera

Thank you!

OpenSSF Community Day
NORTH AMERICA 2026