Rethinking Trust in Forge-Based Git Security

Aditya Sirish A Yelgundhalli, Patrick Zielinski,
Reza Curtmola, Justin Cappos

NJIT New Jersey Institute of Technology

NYU | TANDON SCHOOL OF ENGINEERING

Bio @adityasaky

- Ph.D. Candidate @ New York University
- Maintainer:
    - **in-toto**
    - **gittuf**
    - **SLSA Specification**
- Contributor:
    - TUF
    - CNCF TAG-Security Supply Chain WG
    - OpenSSF SCI WG

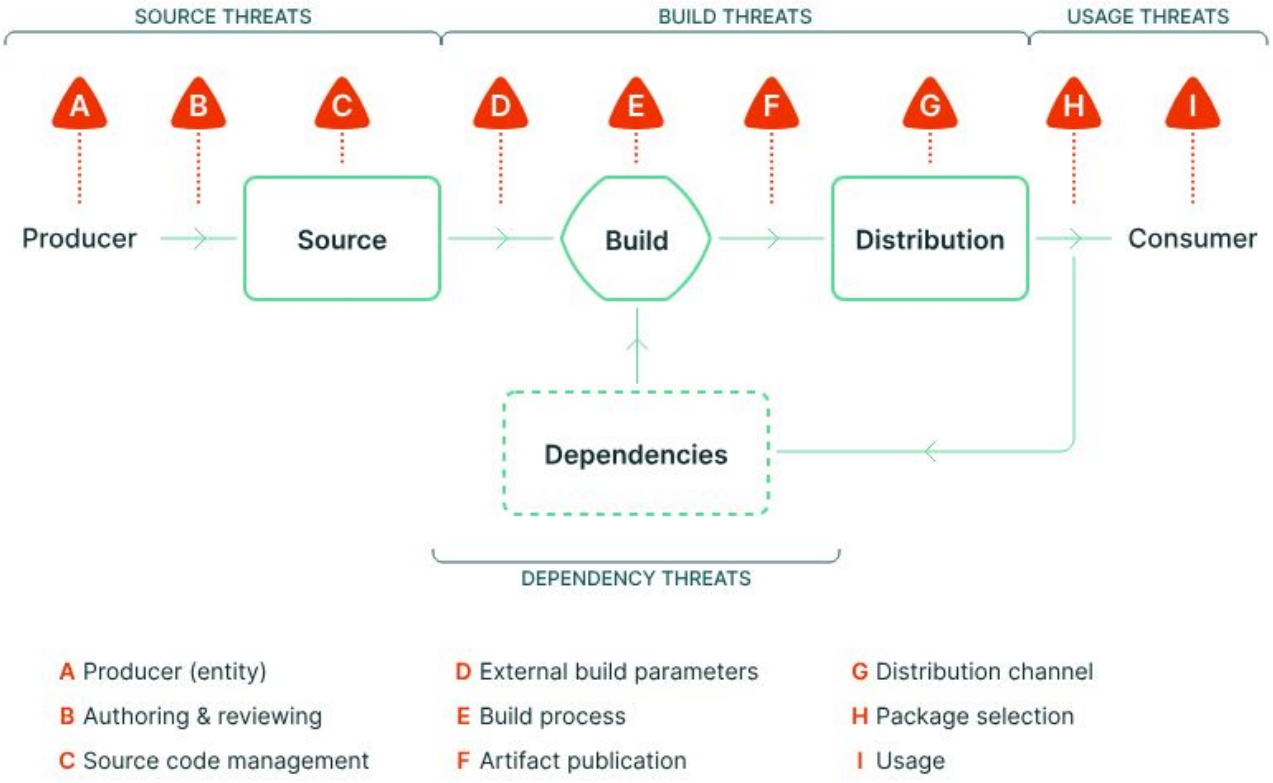
NYU | TANDON SCHOOL OF ENGINEERING

Bio @patzielinski

- Ph.D. Student @ New York University
- Maintainer:
  - **gittuf**
- Contributor:
  - **TAF**
- Moonlights as a pleasant meadow

NYU | TANDON SCHOOL OF ENGINEERING

Software Supply Chain



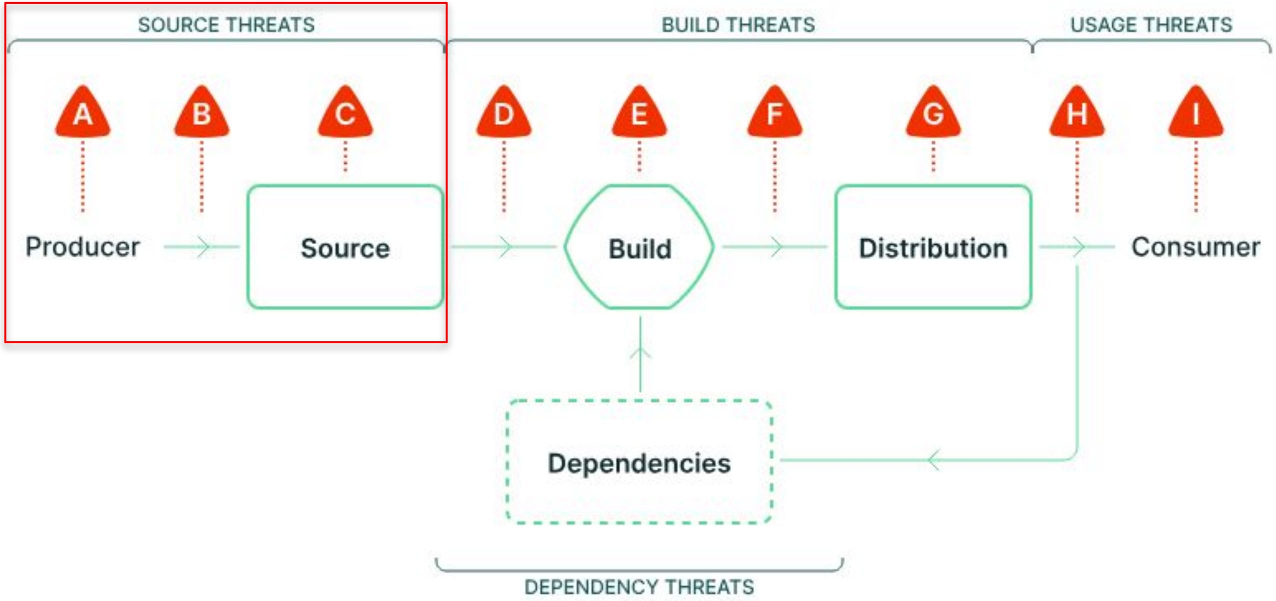
A Producer (entity)
B Authoring & reviewing
C Source code management

D External build parameters
E Build process
F Artifact publication

G Distribution channel
H Package selection
I Usage

From: slsa.dev

Software Supply Chain



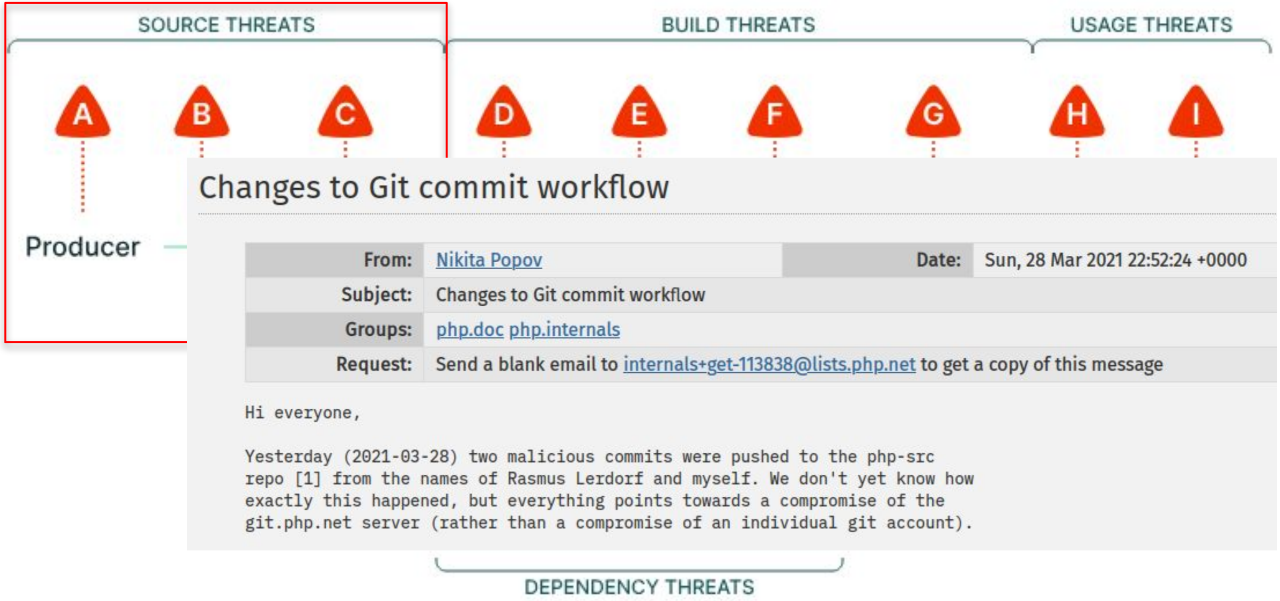
A Producer (entity)
B Authoring & reviewing
C Source code management

D External build parameters
E Build process
F Artifact publication

G Distribution channel
H Package selection
I Usage

From: slsa.dev

Software Supply Chain



Changes to Git commit workflow

From:	Nikita Popov	**Date:**	Sun, 28 Mar 2021 22:52:24 +0000
Subject:	Changes to Git commit workflow		
Groups:	php.doc php.internals		
Request:	Send a blank email to internals+get-113838@lists.php.net to get a copy of this message		

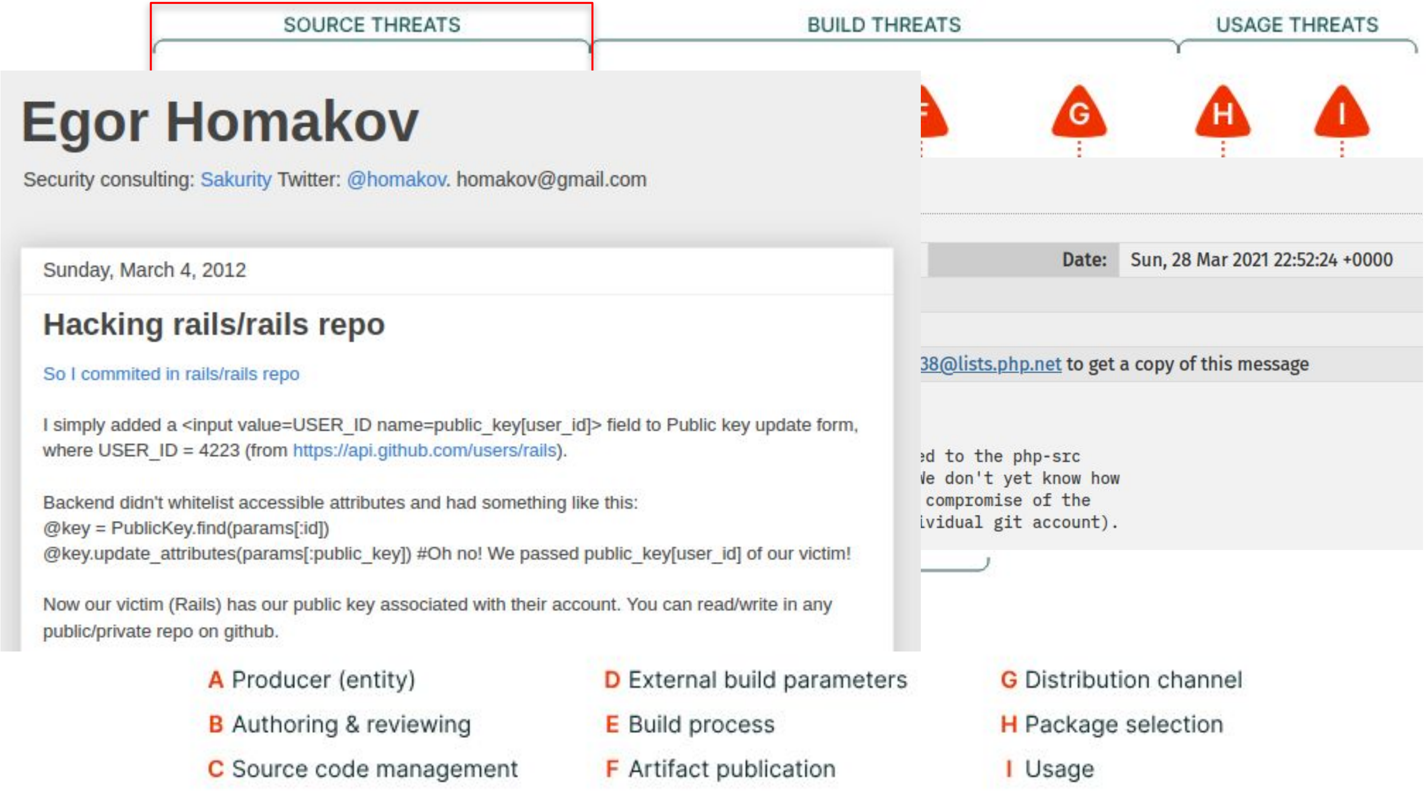
```
Hi everyone,

Yesterday (2021-03-28) two malicious commits were pushed to the php-src
repo [1] from the names of Rasmus Lerdorf and myself. We don't yet know how
exactly this happened, but everything points towards a compromise of the
git.php.net server (rather than a compromise of an individual git account).
```

A	Producer (entity)
B	Authoring & reviewing
C	Source code management
D	External build parameters
E	Build process
F	Artifact publication
G	Distribution channel
H	Package selection
I	Usage

From: slsa.dev

Software Supply Chain



Egor Homakov

Security consulting: Sakurity Twitter: @homakov. homakov@gmail.com

Sunday, March 4, 2012

Hacking rails/rails repo

So I commited in rails/rails repo

I simply added a <input value=USER_ID name=public_key[user_id]> field to Public key update form, where USER_ID = 4223 (from https://api.github.com/users/rails).

Backend didn't whitelist accessible attributes and had something like this:
@key = PublicKey.find(params[:id])
@key.update_attributes(params[:public_key]) #Oh no! We passed public_key[user_id] of our victim!

Now our victim (Rails) has our public key associated with their account. You can read/write in any public/private repo on github.

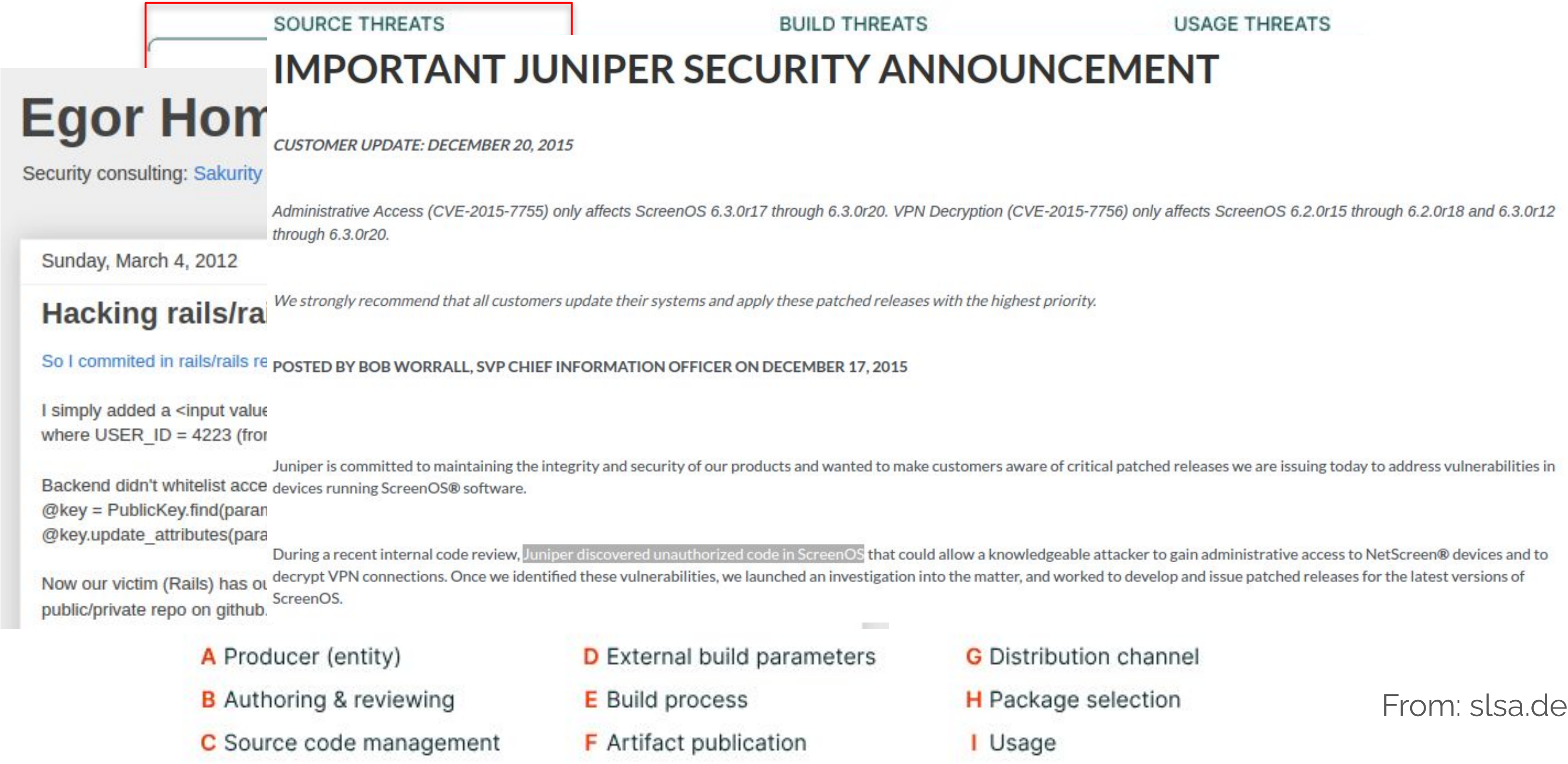
Date: Sun, 28 Mar 2021 22:52:24 +0000

38@lists.php.net to get a copy of this message

```
ed to the php-src
We don't yet know how
 compromise of the
ividual git account).
```

From: slsa.dev

Software Supply Chain



SOURCE THREATS

BUILD THREATS

USAGE THREATS

Egor Hom

Security consulting: Sakurity

Sunday, March 4, 2012

Hacking rails/rai

So I commited in rails/rails re

I simply added a <input value
where USER_ID = 4223 (fror

Backend didn't whitelist acce
@key = PublicKey.find(paran
@key.update_attributes(para

Now our victim (Rails) has ou
public/private repo on github.

IMPORTANT JUNIPER SECURITY ANNOUNCEMENT

CUSTOMER UPDATE: DECEMBER 20, 2015

Administrative Access (CVE-2015-7755) only affects ScreenOS 6.3.0r17 through 6.3.0r20. VPN Decryption (CVE-2015-7756) only affects ScreenOS 6.2.0r15 through 6.2.0r18 and 6.3.0r12 through 6.3.0r20.

We strongly recommend that all customers update their systems and apply these patched releases with the highest priority.

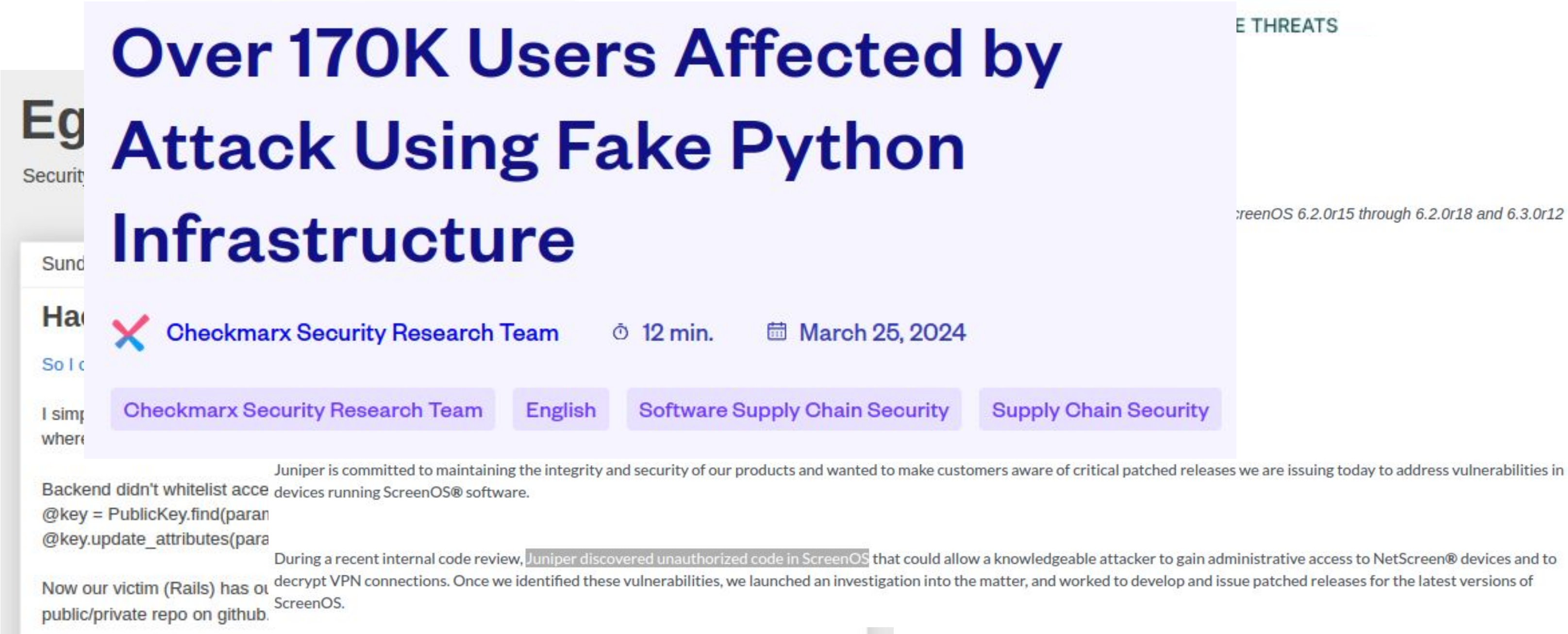
POSTED BY BOB WORRALL, SVP CHIEF INFORMATION OFFICER ON DECEMBER 17, 2015

Juniper is committed to maintaining the integrity and security of our products and wanted to make customers aware of critical patched releases we are issuing today to address vulnerabilities in devices running ScreenOS® software.

During a recent internal code review, Juniper discovered unauthorized code in ScreenOS that could allow a knowledgeable attacker to gain administrative access to NetScreen® devices and to decrypt VPN connections. Once we identified these vulnerabilities, we launched an investigation into the matter, and worked to develop and issue patched releases for the latest versions of ScreenOS.

From: slsa.dev

Software Supply Chain



Over 170K Users Affected by Attack Using Fake Python Infrastructure

Checkmarx Security Research Team · 12 min. · March 25, 2024

Checkmarx Security Research Team | English | Software Supply Chain Security | Supply Chain Security

E THREATS

reenOS 6.2.0r15 through 6.2.0r18 and 6.3.0r12

Eg

Securit

Sund

Ha

So I c

I simp
where

Backend didn't whitelist acce
@key = PublicKey.find(paran
@key.update_attributes(para

Now our victim (Rails) has ou
public/private repo on github.

Juniper is committed to maintaining the integrity and security of our products and wanted to make customers aware of critical patched releases we are issuing today to address vulnerabilities in devices running ScreenOS® software.

During a recent internal code review, Juniper discovered unauthorized code in ScreenOS that could allow a knowledgeable attacker to gain administrative access to NetScreen® devices and to decrypt VPN connections. Once we identified these vulnerabilities, we launched an investigation into the matter, and worked to develop and issue patched releases for the latest versions of ScreenOS.

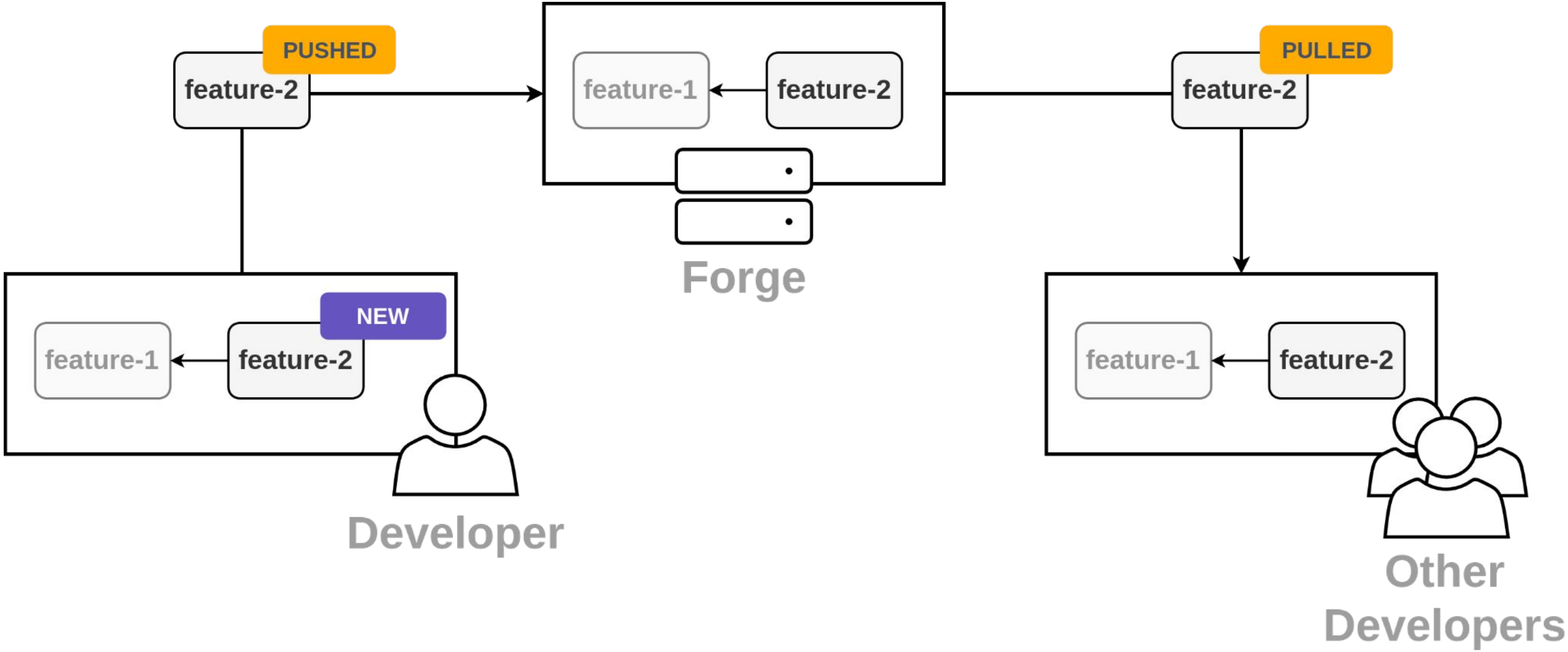
A Producer (entity)
B Authoring & reviewing
C Source code management

D External build parameters
E Build process
F Artifact publication

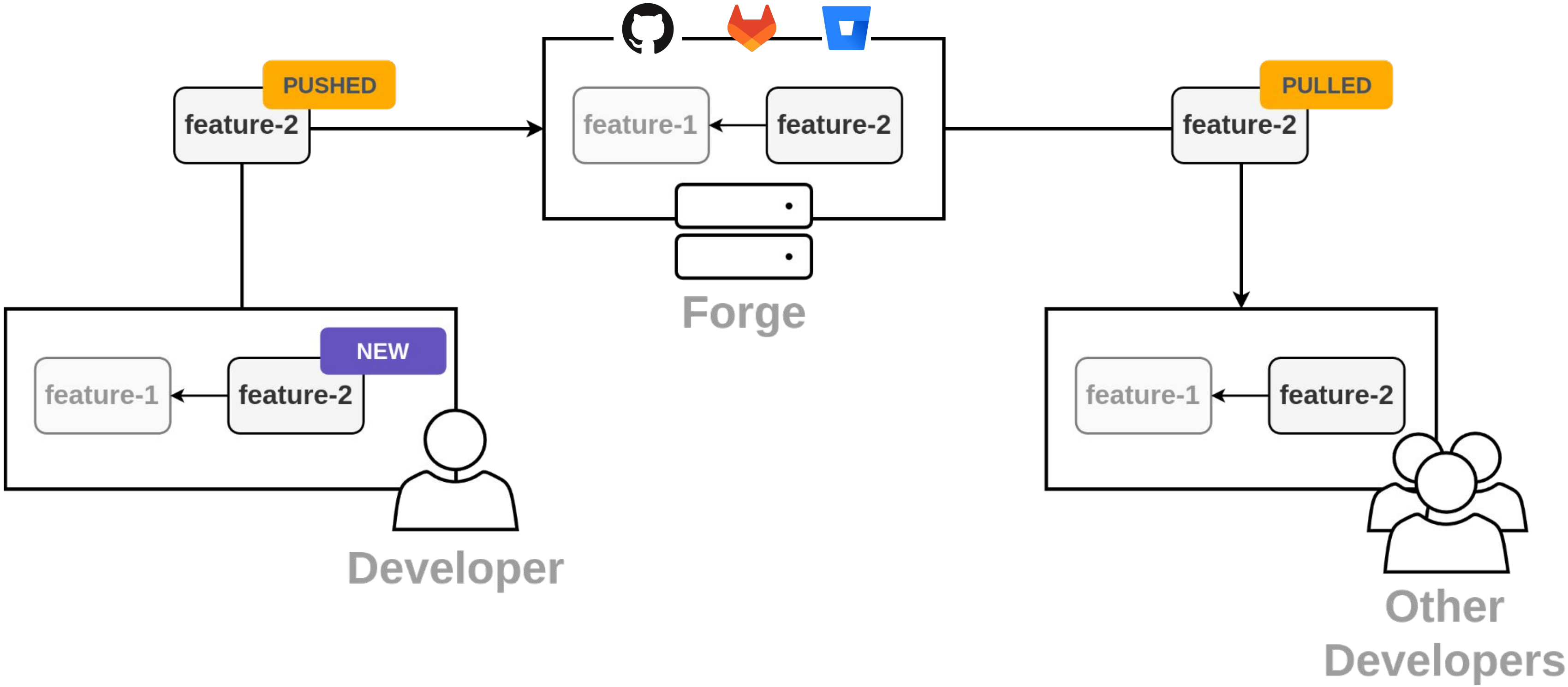
G Distribution channel
H Package selection
I Usage

From: slsa.dev

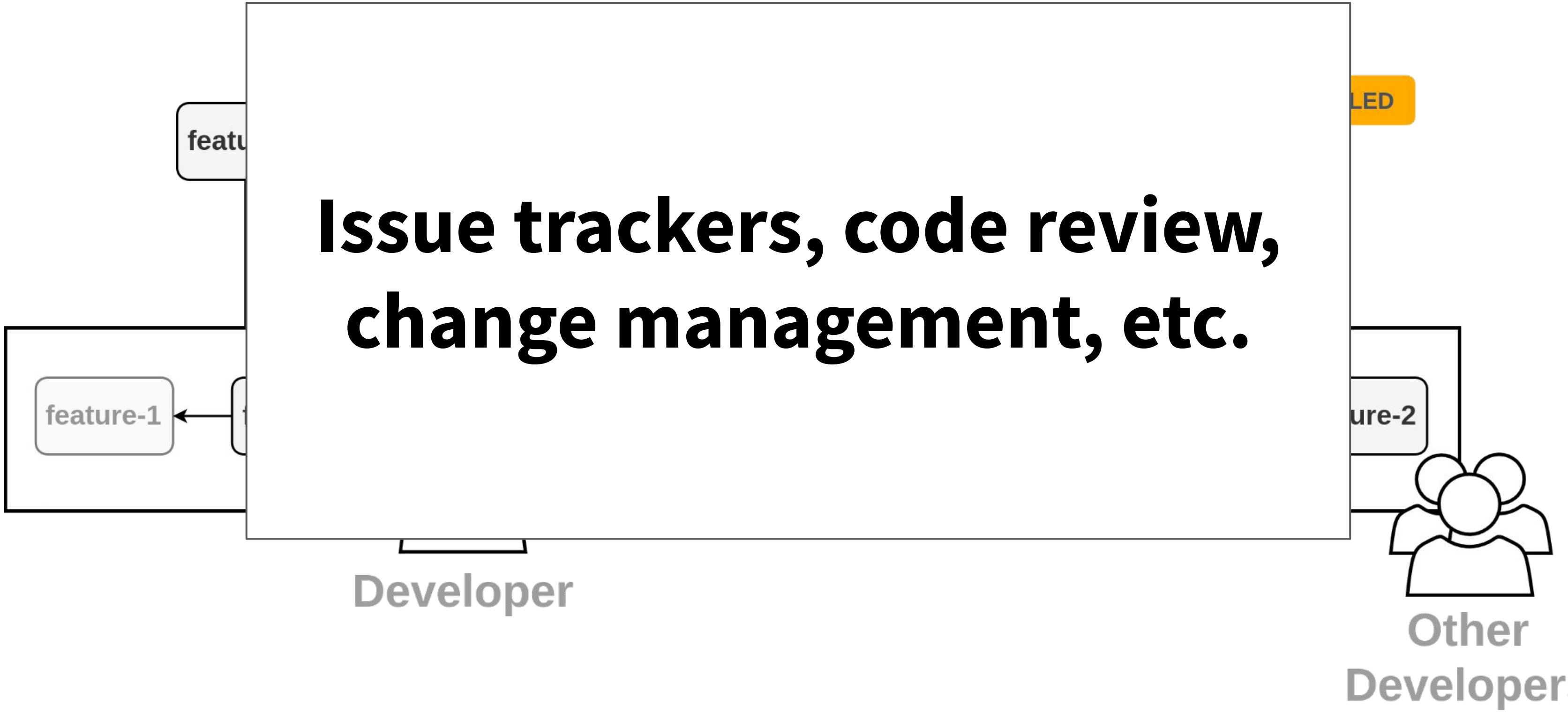
Typical Developer Workflow



Typical Developer Workflow

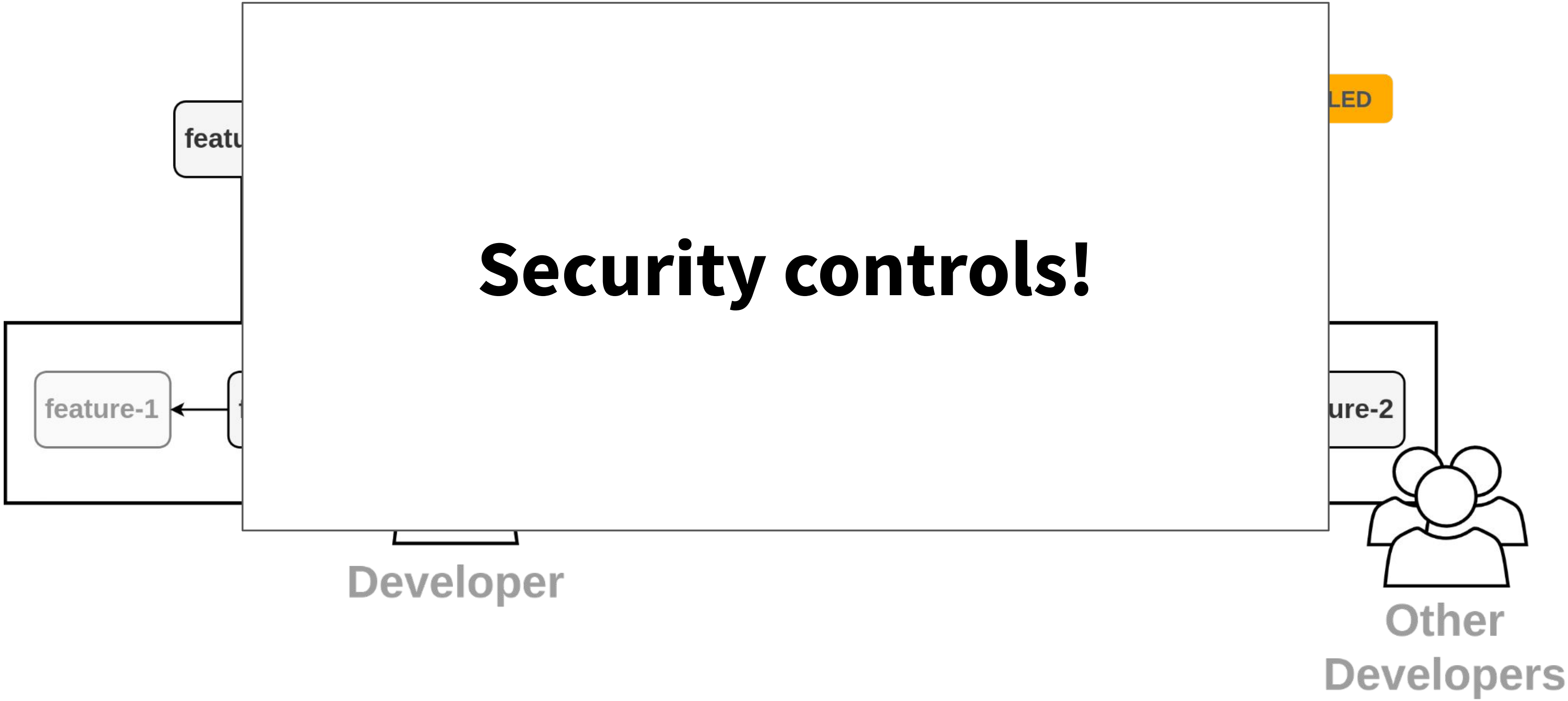


Typical Developer Workflow



Issue trackers, code review, change management, etc.

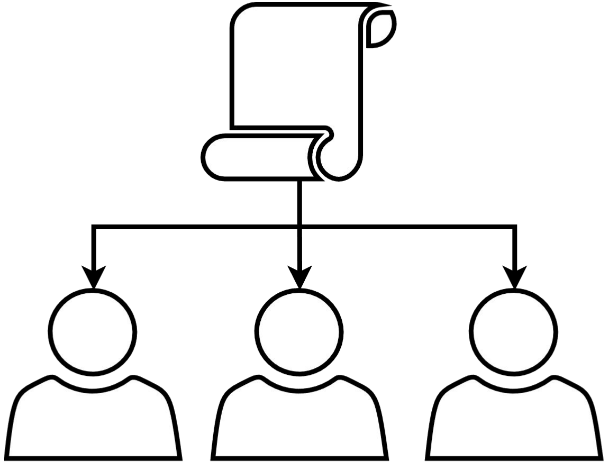
Typical Developer Workflow



Security controls!

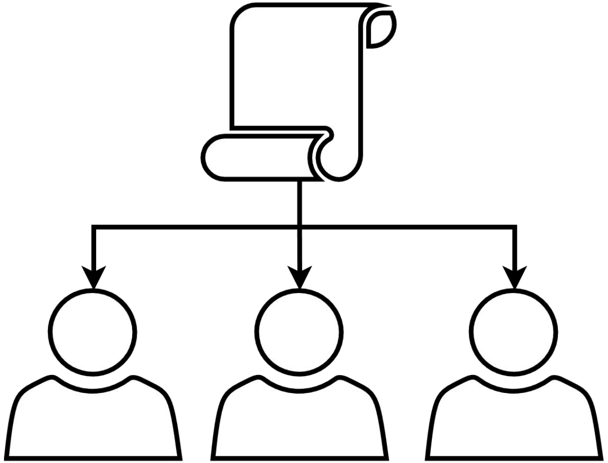
Forges Offer Security Controls

Forges Offer Security Controls

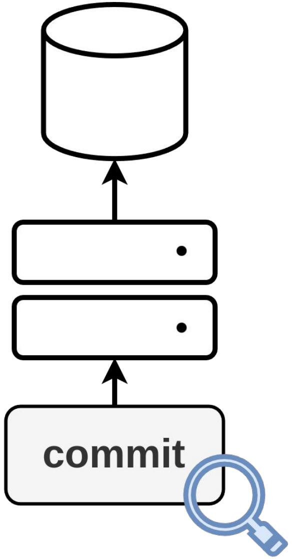


Policy Declaration

Forges Offer Security Controls

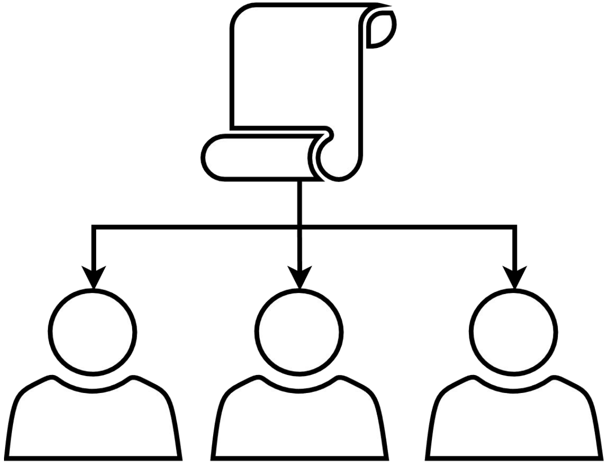


Policy Declaration

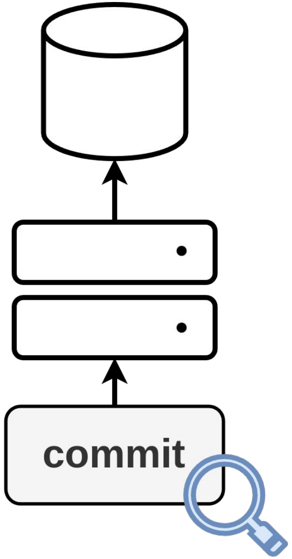


Activity Tracking

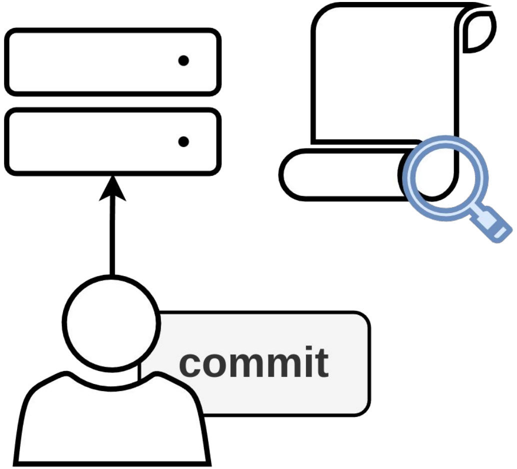
Forges Offer Security Controls



Policy Declaration

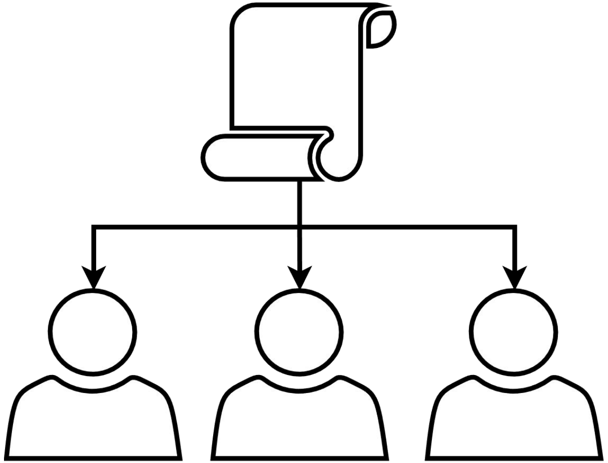


Activity Tracking

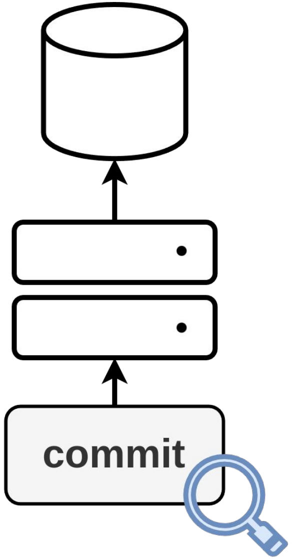


Policy Enforcement

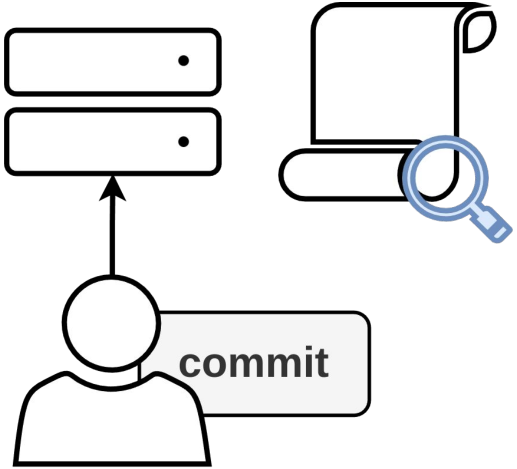
Problems with Forge Security Controls



Policy Declaration



Activity Tracking

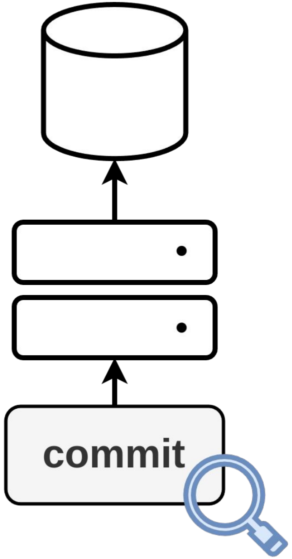


Policy Enforcement

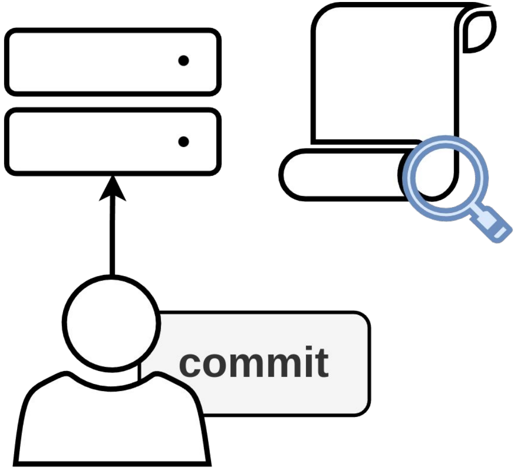
Problems with Forge Security Controls

Who can edit policy?

Policy Declaration



Activity Tracking



Policy Enforcement

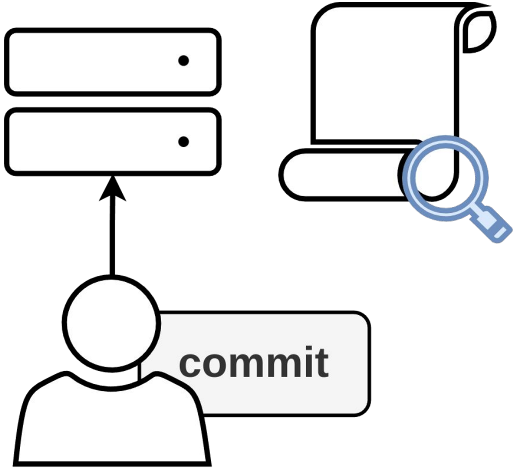
Problems with Forge Security Controls

Who can edit policy?

Policy Declaration

Has the log been tampered with?

Activity Tracking



Policy Enforcement

Problems with Forge Security Controls

Who can edit policy?

Policy Declaration

Has the log been tampered with?

Activity Tracking

Is enforcement working properly?

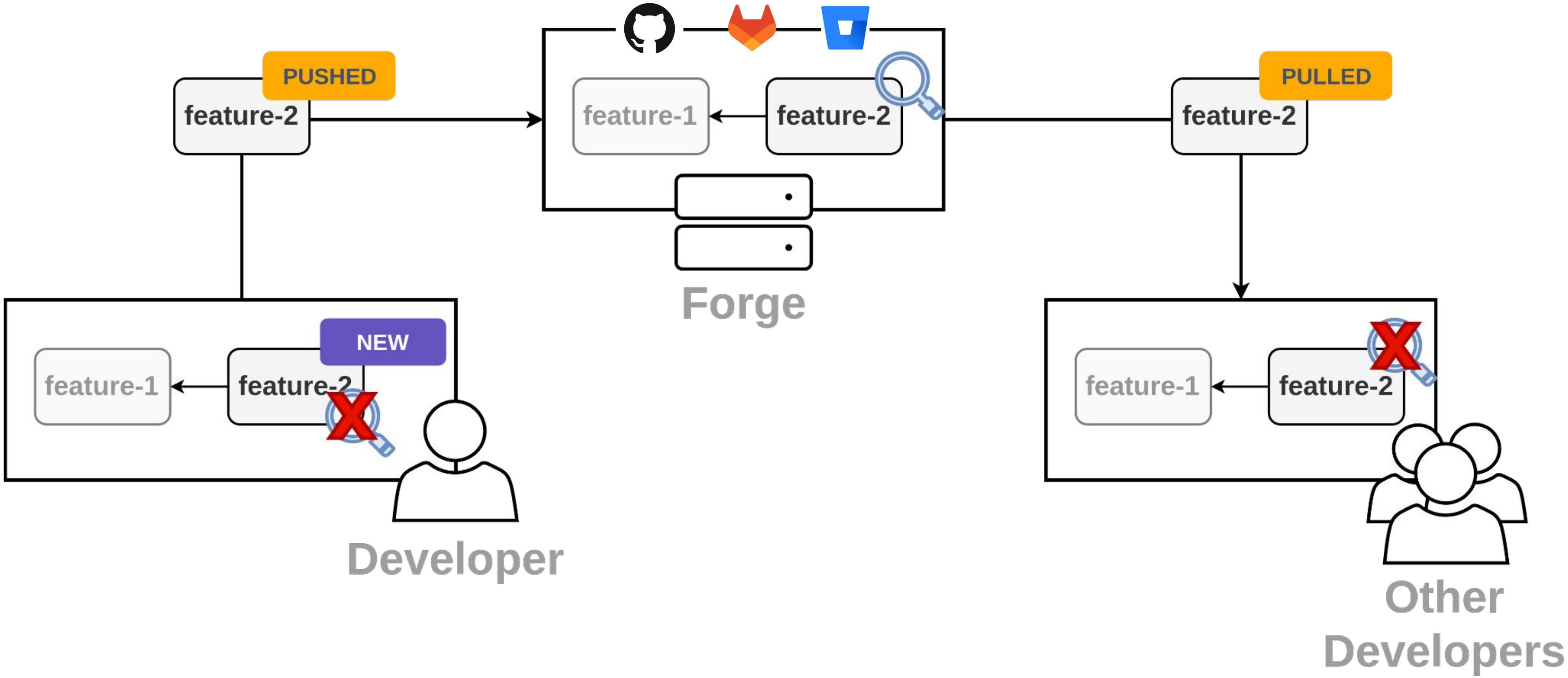
Policy Enforcement

Threat Model

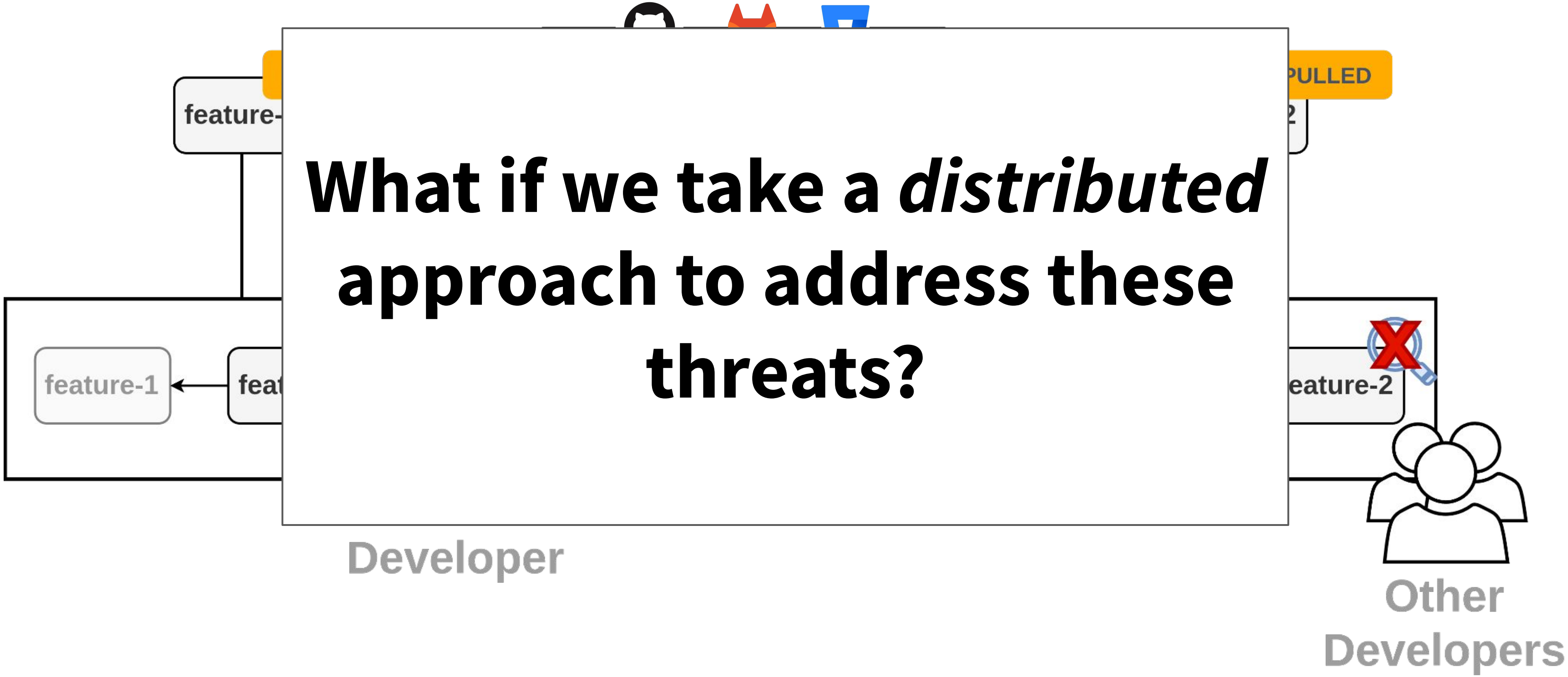
Any trusted party (maintainers, forge, bots) may be compromised and act in an arbitrarily malicious manner, such as:

- T1: **Modifying configured repository security policies**, such as to weaken them
- T2: **Tampering with the contents of the repository's activity log**, such as by reordering, dropping, or otherwise manipulating log entries
- T3: **Subverting the enforcement of security policies**, such as by accepting invalid changes instead of rejecting them

Problems with Forge Security Controls

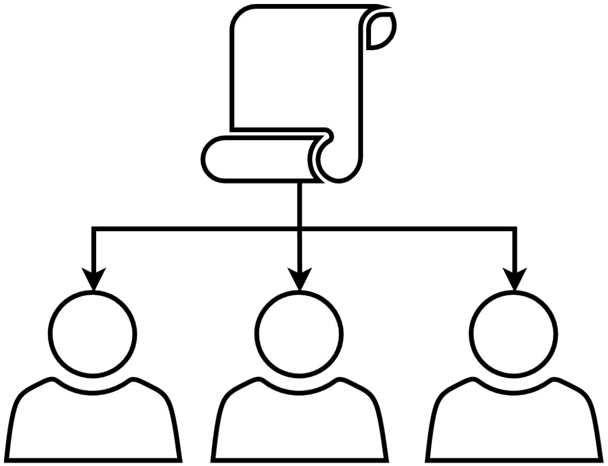


Problems with Forge Security Controls

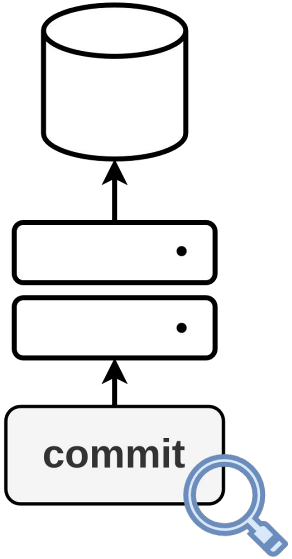


What if we take a *distributed* approach to address these threats?

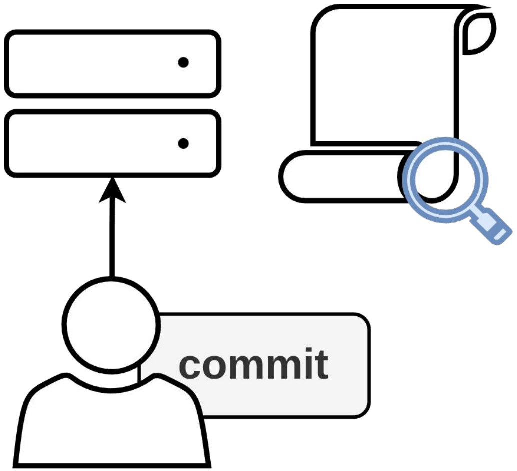
Security Goals



Policy Declaration

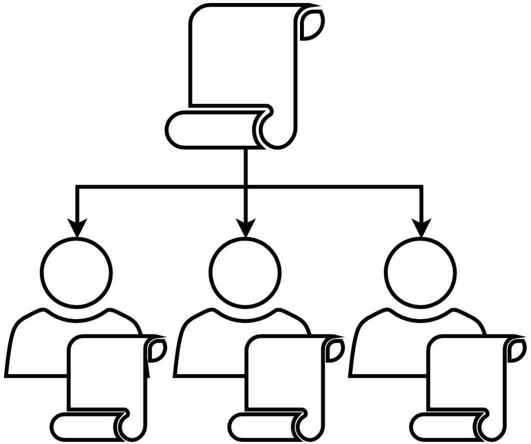


Activity Tracking

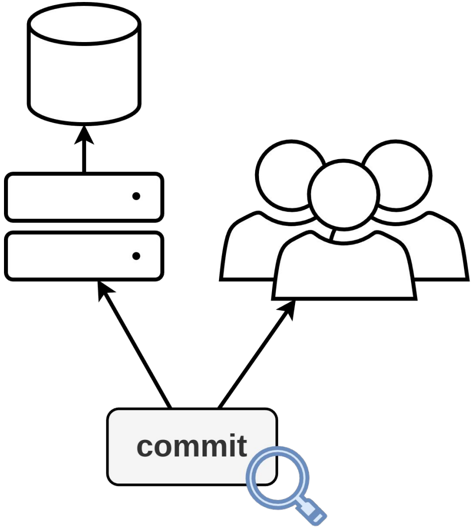


Policy Enforcement

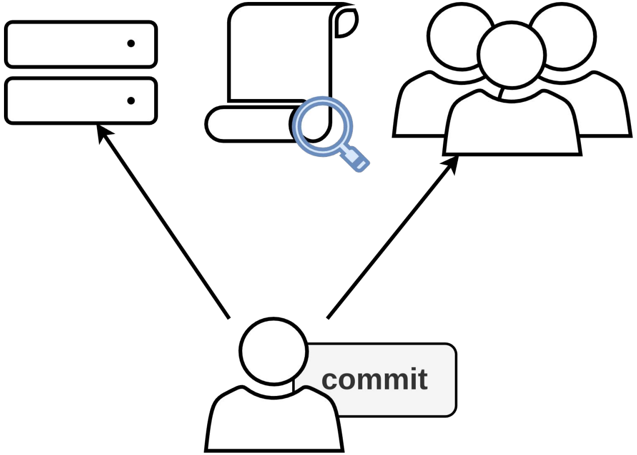
Security Goals



Distribute
Policy
Declaration



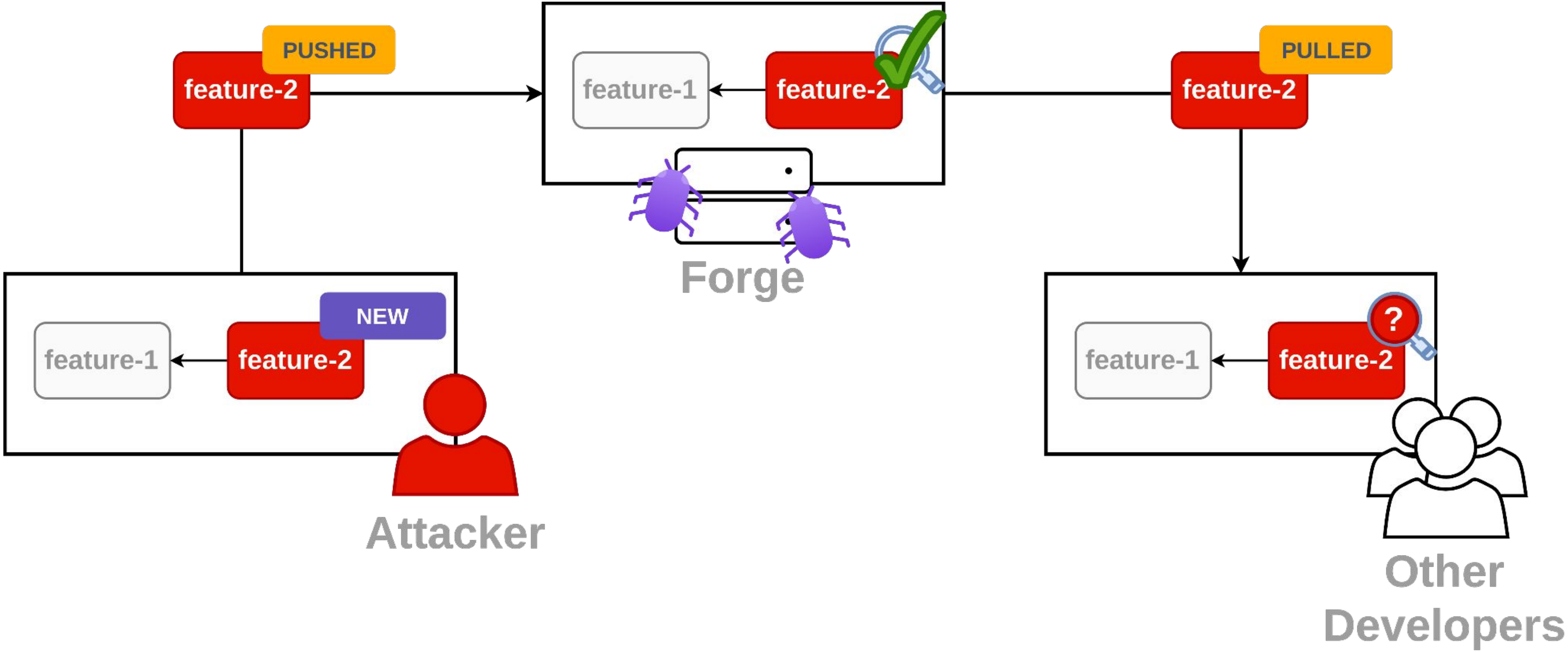
Distribute
Activity
Tracking



Distribute
Policy
Enforcement

gittuf

gittuf: A Scenario

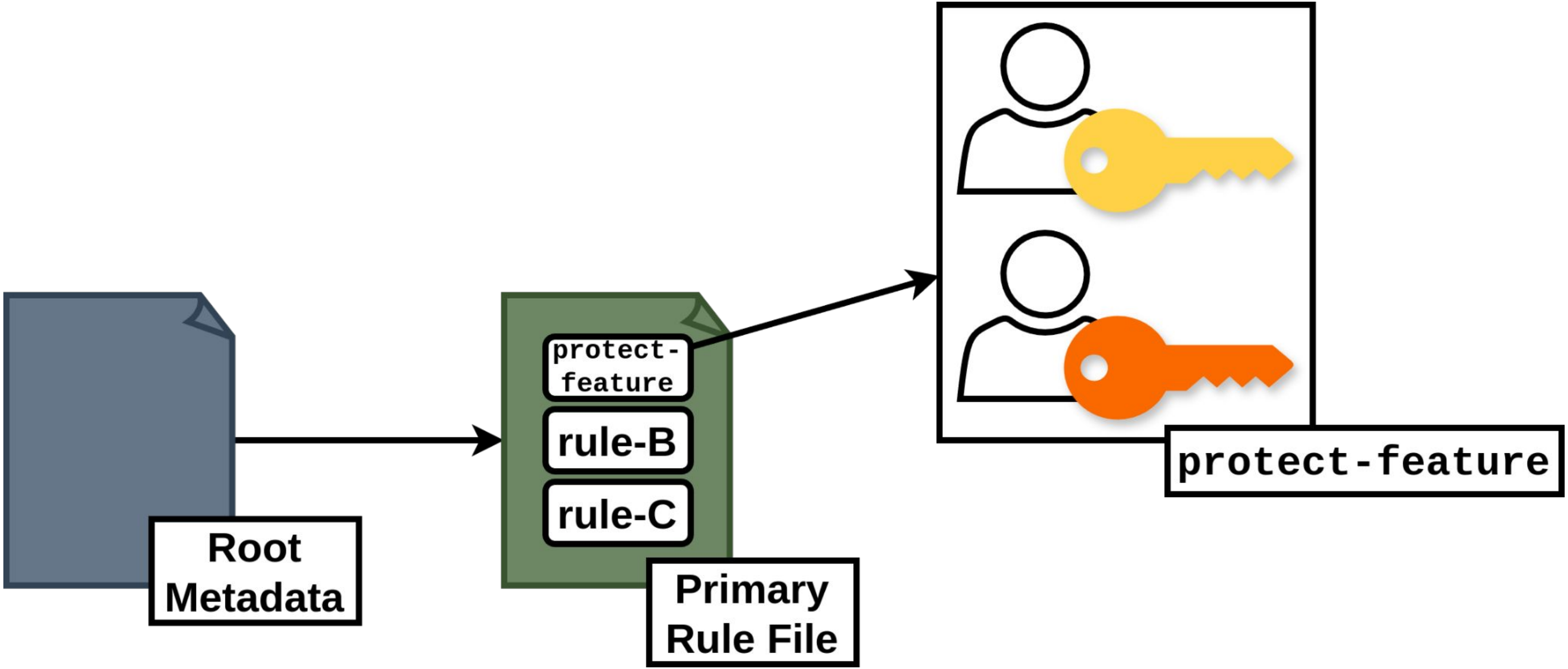


gittuf: A Scenario

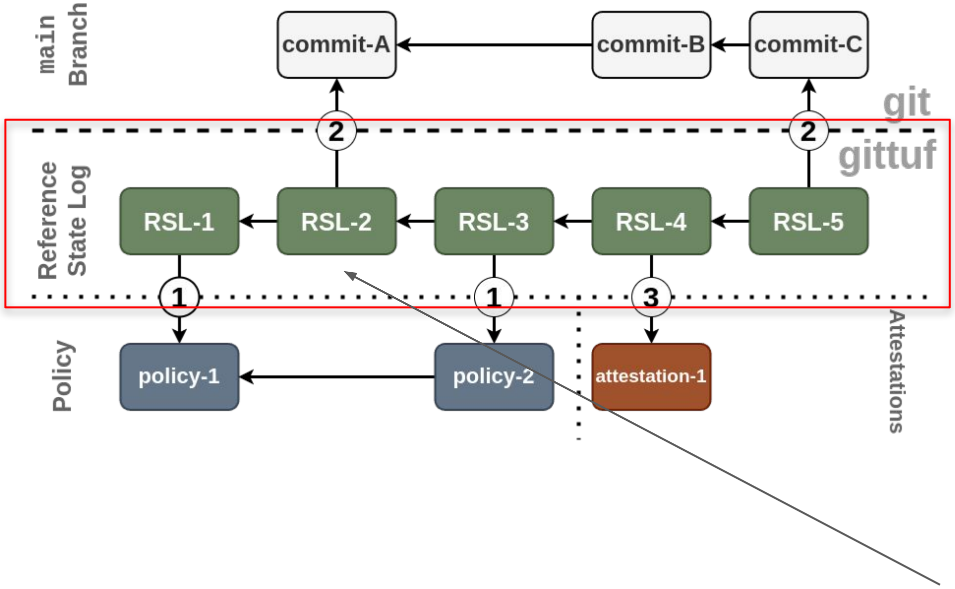
PUS
feature-2
PULLED
re-2
feature-1
featur
feature-2
?
Attacker
Other Developers

What if everyone has the policy and the activity log?

gittuf Internals: Policy



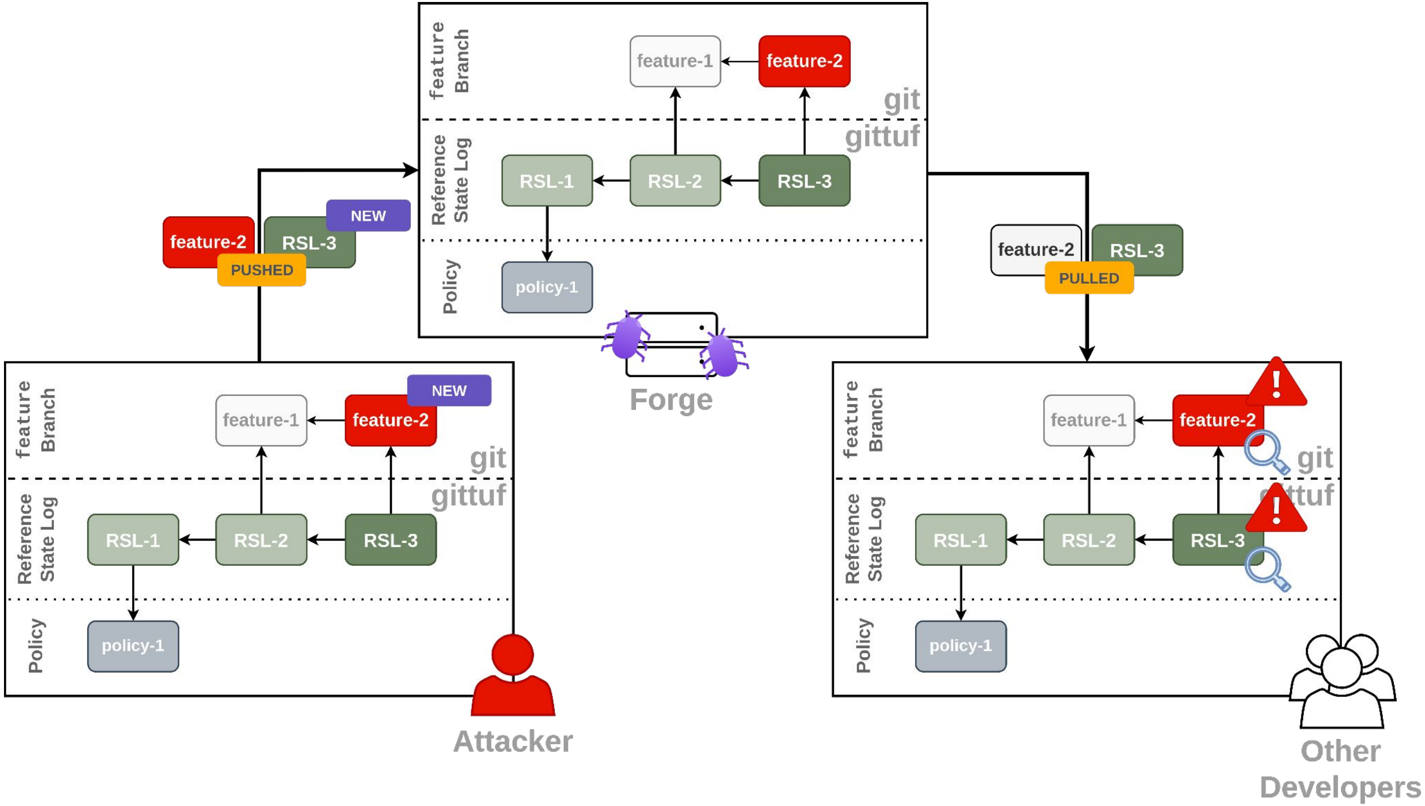
gittuf Internals: Activity Tracking



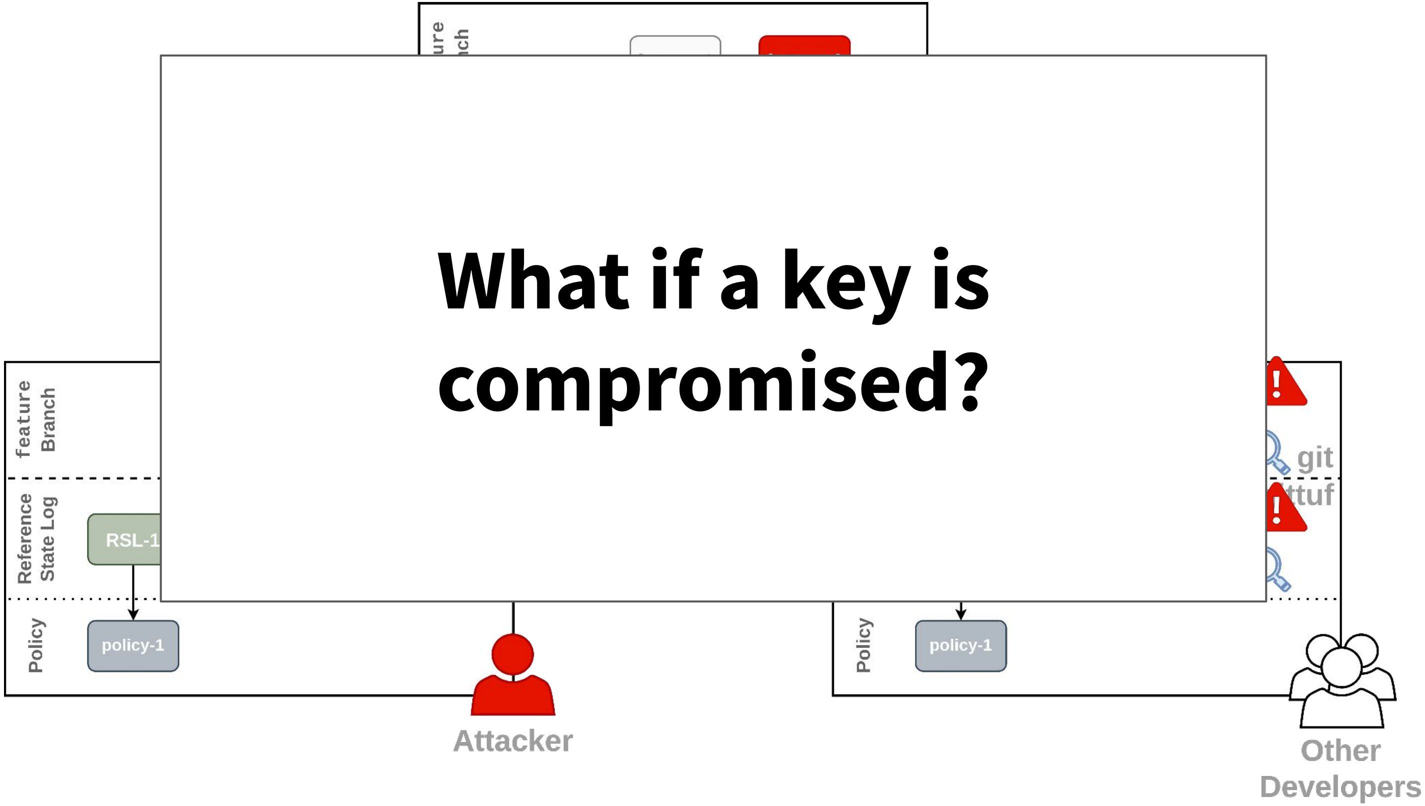
Record of Pushes

E.g., "Alice pushed `main` to `commit-A`"

gittuf: A Scenario

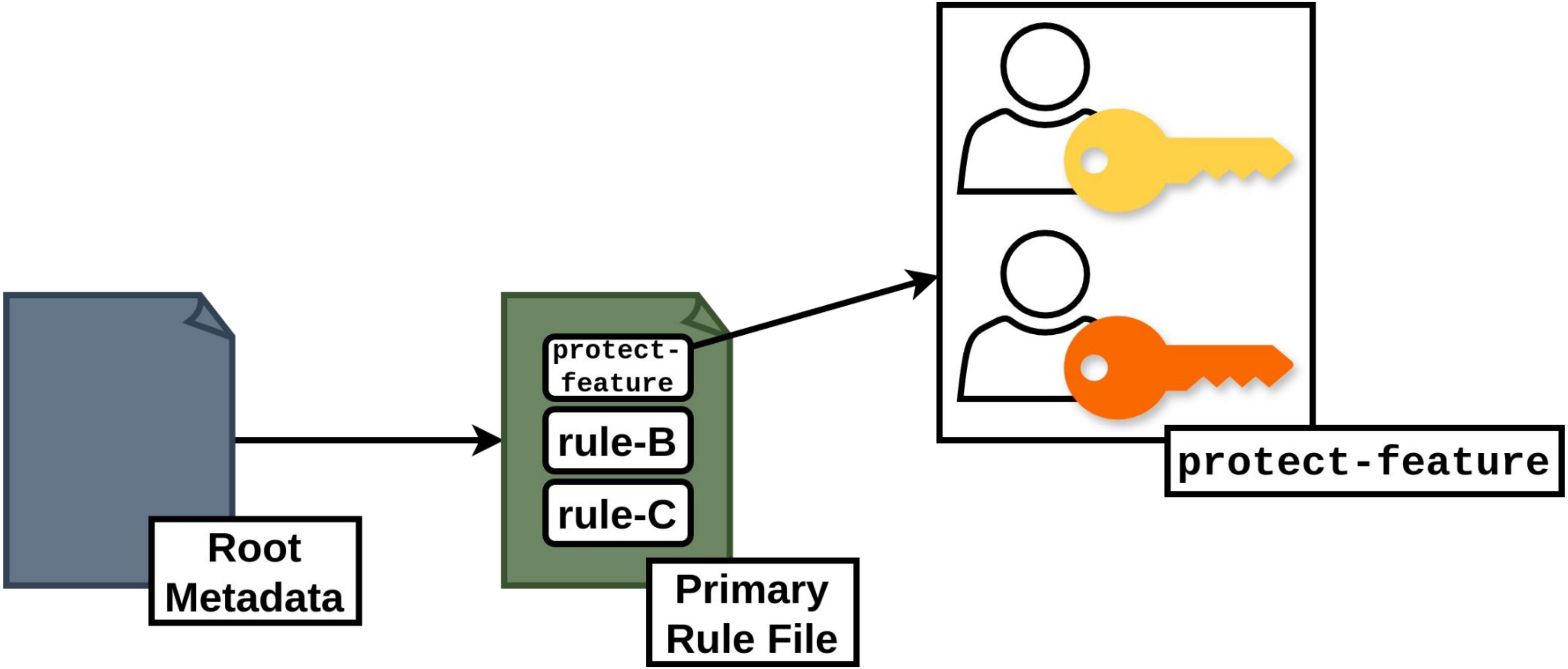


gittuf: A Scenario

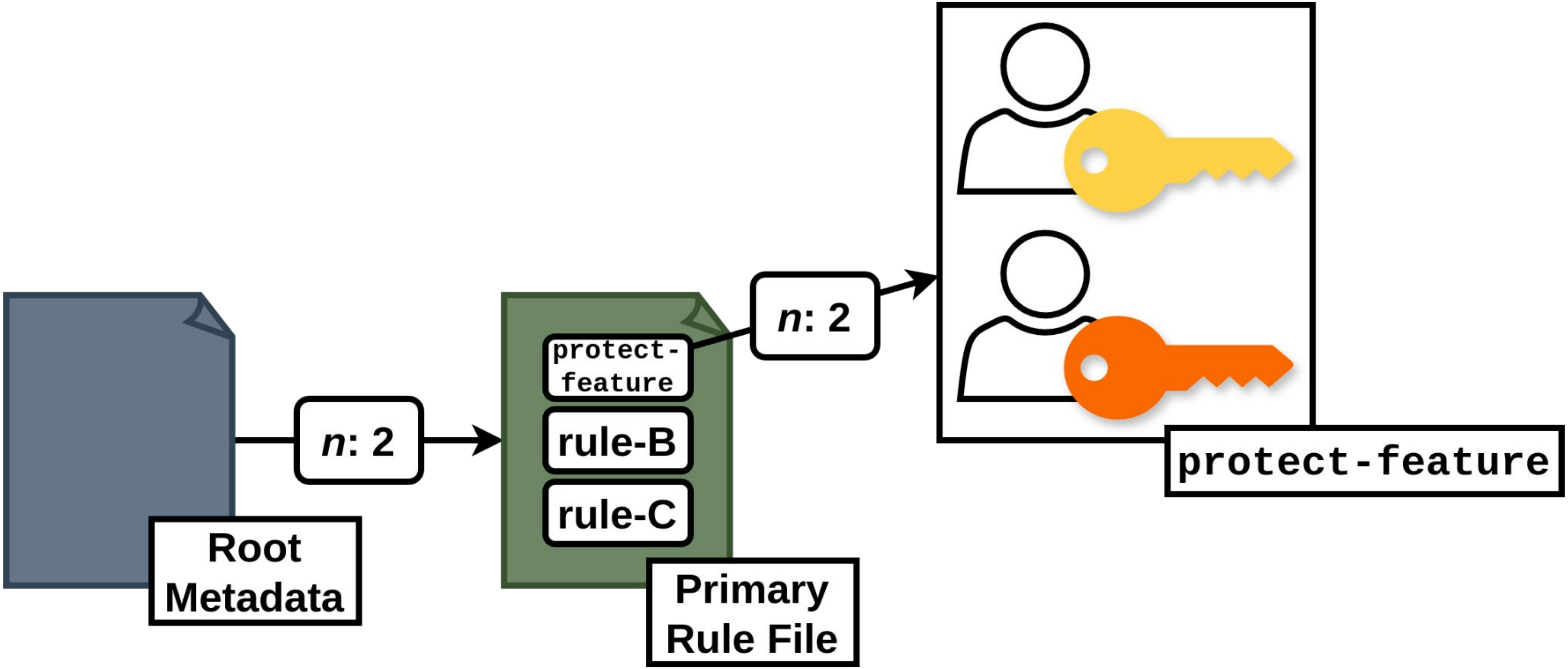


What if a key is compromised?

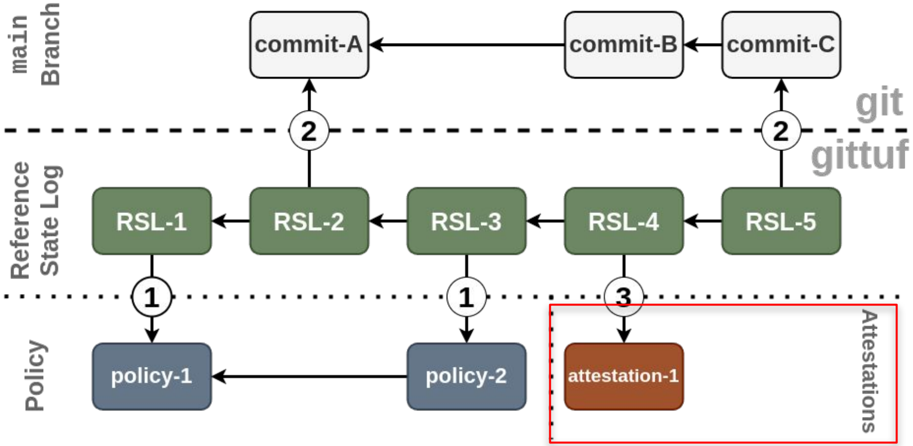
gittuf Internals: Delegations



gittuf Internals: Delegations



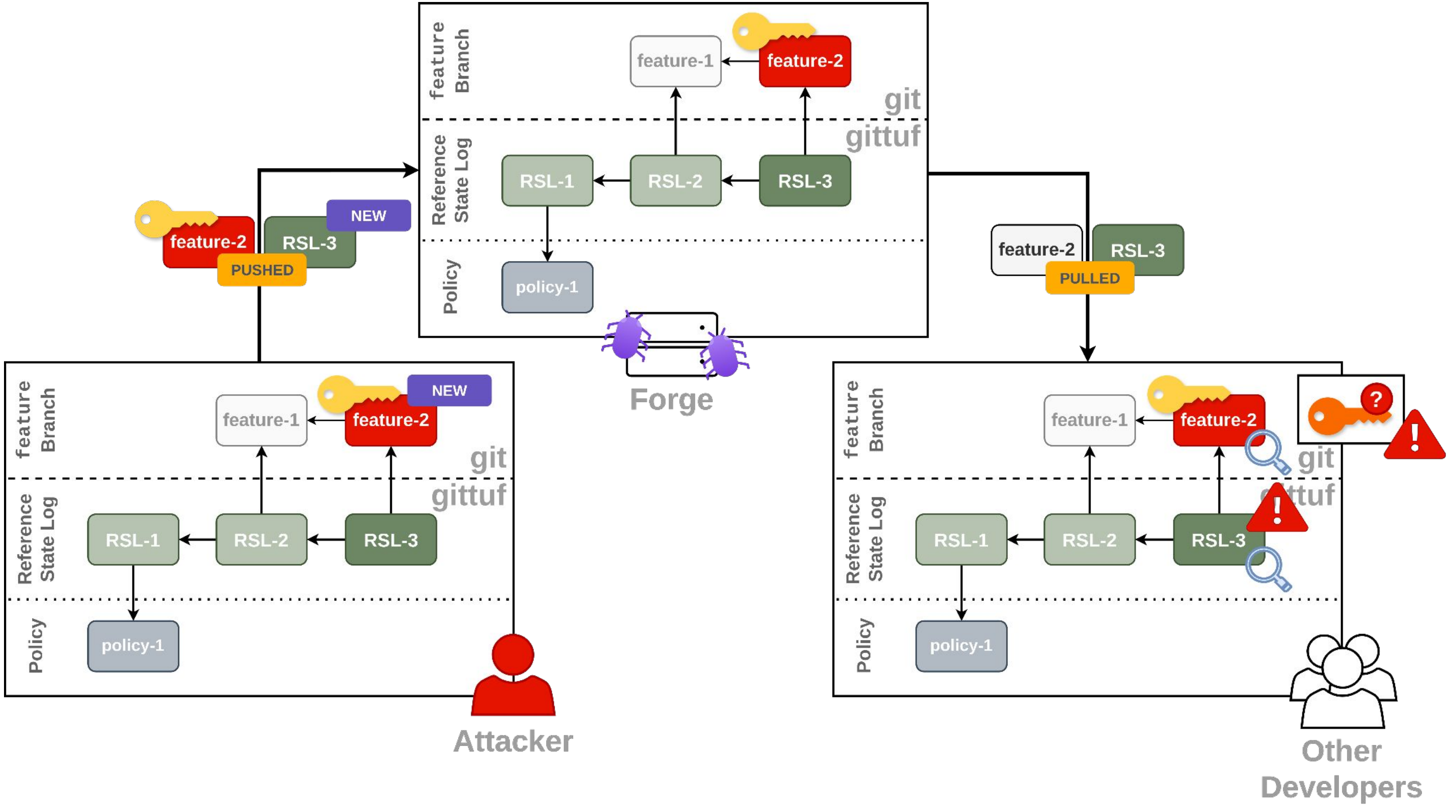
gittuf Internals: Activity Tracking



Other Activity

E.g., “Alice **approved** updating `main` to `commit-C`”

gittuf: A Scenario



Implementation & Deployment

Implementation & Deployment

- **Open source**, part of the **OpenSSF Sandbox** at the **Linux Foundation**

Implementation & Deployment

- **Open source**, part of the **OpenSSF Sandbox** at the **Linux Foundation**
- Developed by **academics, industry,** and **independent developers**

Implementation & Deployment

- **Open source**, part of the **OpenSSF Sandbox** at the **Linux Foundation**
- Developed by **academics, industry,** and **independent developers**
- **Backwards compatible** with the current Git ecosystem

Implementation & Deployment

- **Open source**, part of the **OpenSSF Sandbox** at the **Linux Foundation**
- Developed by **academics, industry,** and **independent developers**
- **Backwards compatible** with the current Git ecosystem
- Being used in a pilot at **Bloomberg**

Implementation & Deployment

Repository

https://github.com/gittuf/gittuf

Simulation / Demo

https://github.com/adityasaky/gittuf-ndss-eval

Implementation & Deployment

Artifact Evaluated
NDSS
SYMPOSIUM
Available
Functional
Reproduced

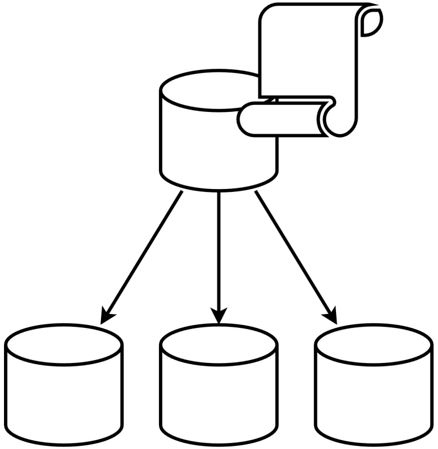
Repository

https://github.com/gittuf/gittuf

Simulation / Demo

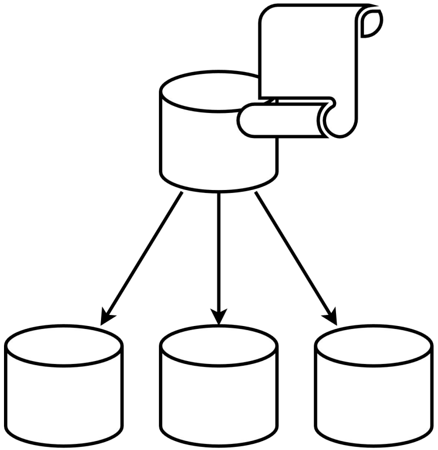
https://github.com/adityasaky/gittuf-ndss-eval

Future Work

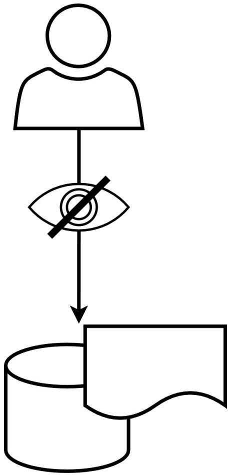


Policies Across Multiple Repositories

Future Work

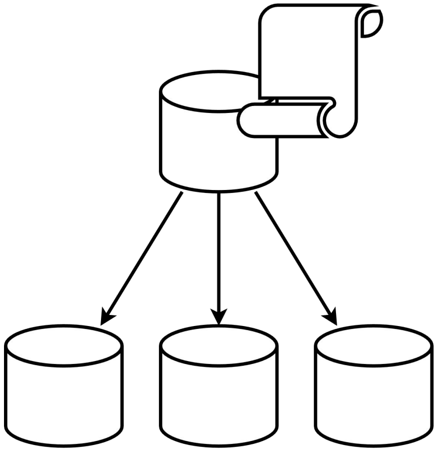


Policies Across Multiple Repositories

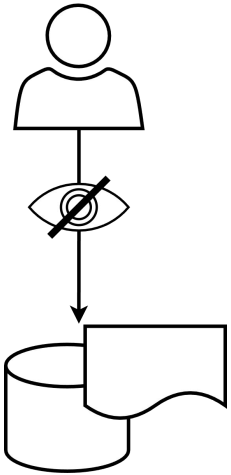


Read Access Control

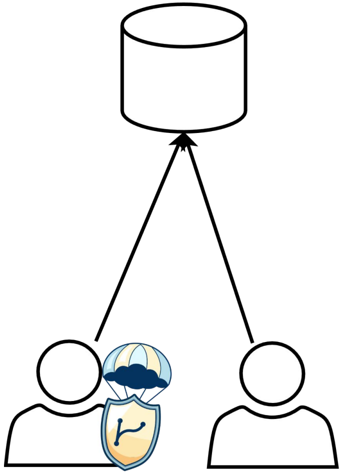
Future Work



Policies Across Multiple Repositories



Read Access Control



Better Support in Mixed Environments

Thank you!
Questions?

aditya.sirish@nyu.edu

patrick.z@nyu.edu

Extra Slides

gittuf Internals: Policy Declaration

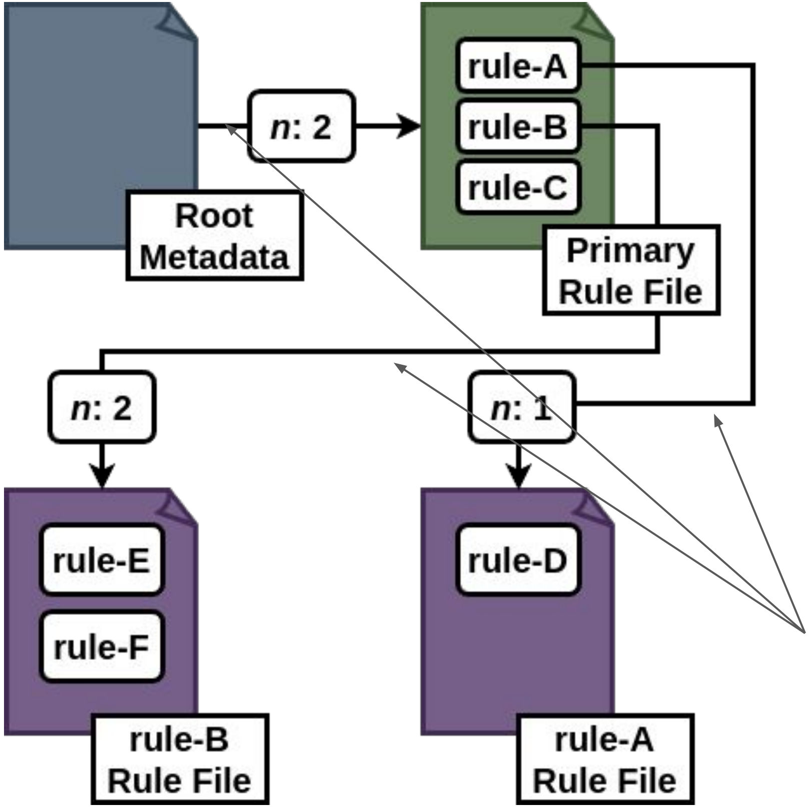
```
rootOfTrust:
keys: {R1, R2, R3, P1, P2, P3}
signers:
      rootOfTrust: (2, {R1, R2, R3})
      primary: (2, {P1, P2, P3})

ruleFile: primary
keys: {Alice, Bob, Carol, Helen, Ilda}
rules:
      protect-main-prod: {git:refs/heads/main,
                          git:refs/heads/prod}
            -> (2, {Alice, Bob, Carol})
      protect-ios-app: {file:ios/*}
            -> (1, {Alice})
      protect-android-app: {file:android/*}
            -> (1, {Bob})
      protect-core-libraries: {file:src/*}
            -> (2, {Carol, Helen, Ilda})

ruleFile: protect-ios-app
keys: {Dana, George}
rules:
      authorize-ios-team: {file:ios/*}
            -> (1, {Dana, George})

ruleFile: protect-android-app
keys: {Eric, Frank}
rules:
      authorize-android-team: {file:android/*}
      -> (1, {Eric, Frank})
```

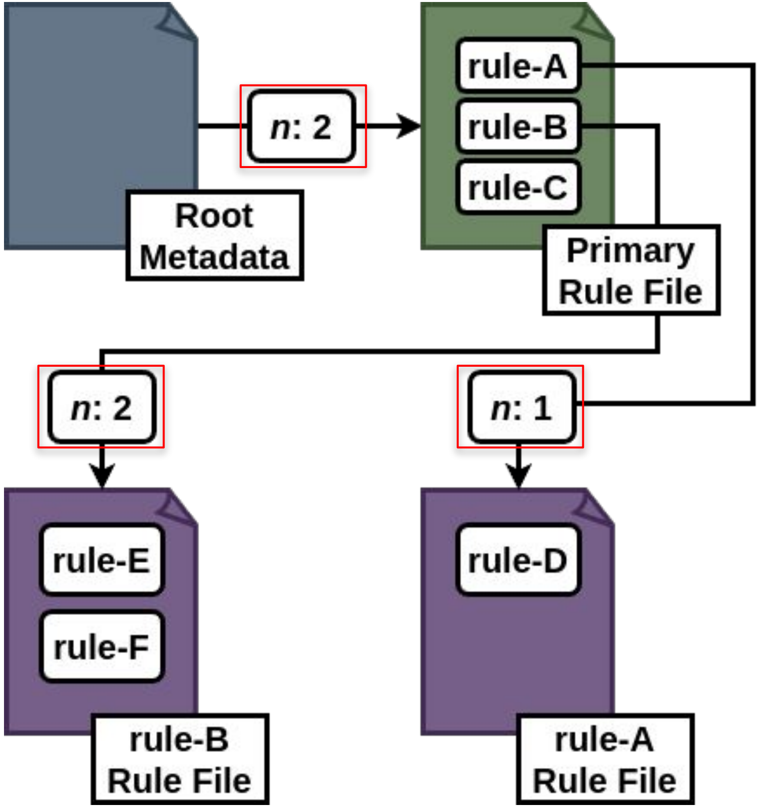
gittuf Internals: Policy



Delegations

E.g., “2 of Alice, Bob, Carol must approve changes to `main`”

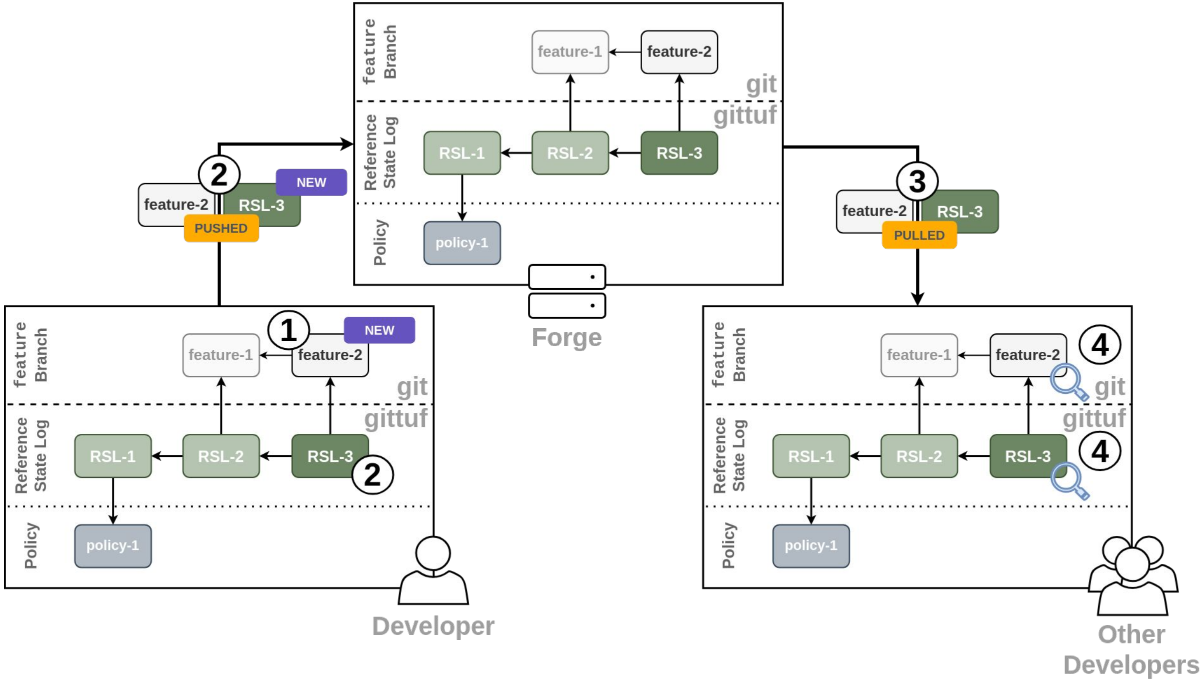
gittuf Internals: Policy Declaration



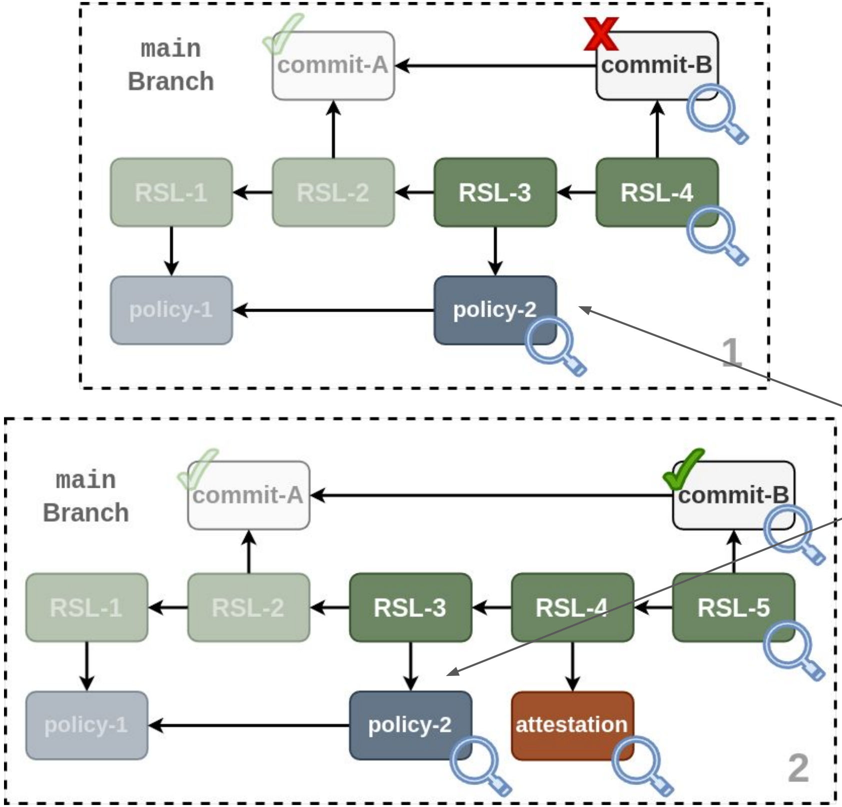
Delegations

E.g., “**2** of Alice, Bob, Carol must approve changes to `main`”

gittuf Internals: Policy Enforcement

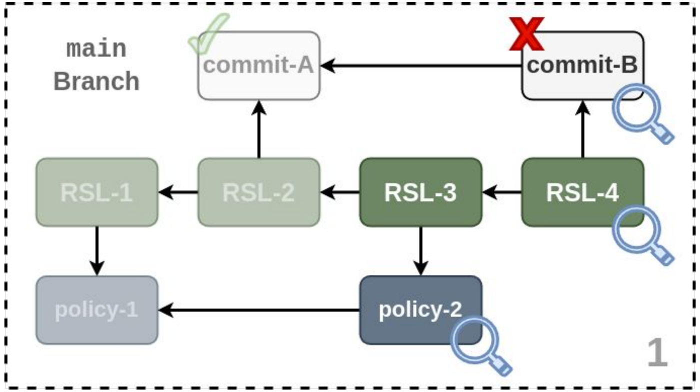


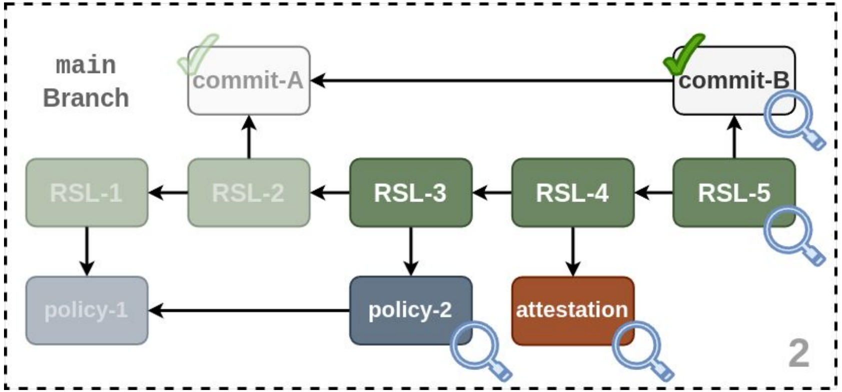
gittuf Internals: Policy Enforcement

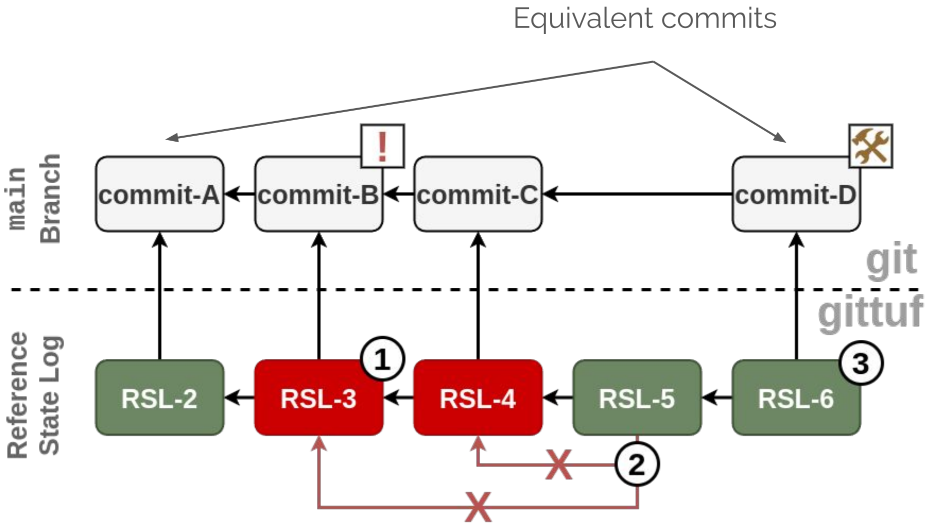


Additional approval required for `main`

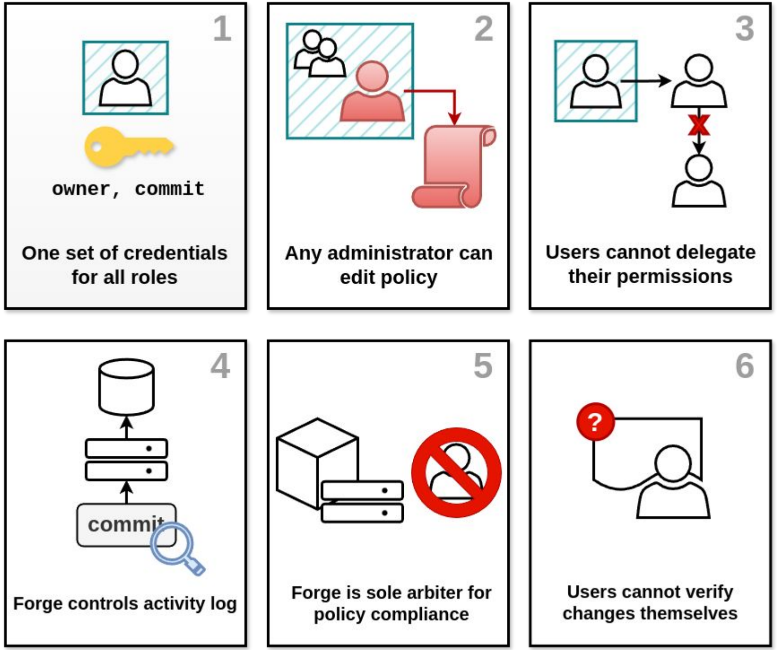
gittuf Internals: Policy Enforcement





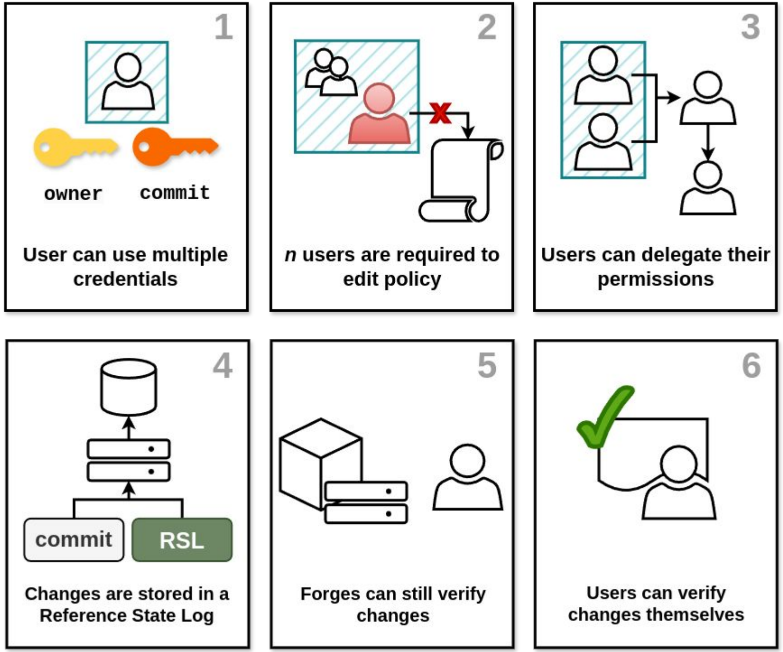


Forges <-> gittuf



1. One set of credentials for all roles
2. Any administrator can edit policy
3. Users cannot delegate their permissions
4. Forge controls activity log
5. Forge is sole arbiter for policy compliance
6. Users cannot verify changes themselves

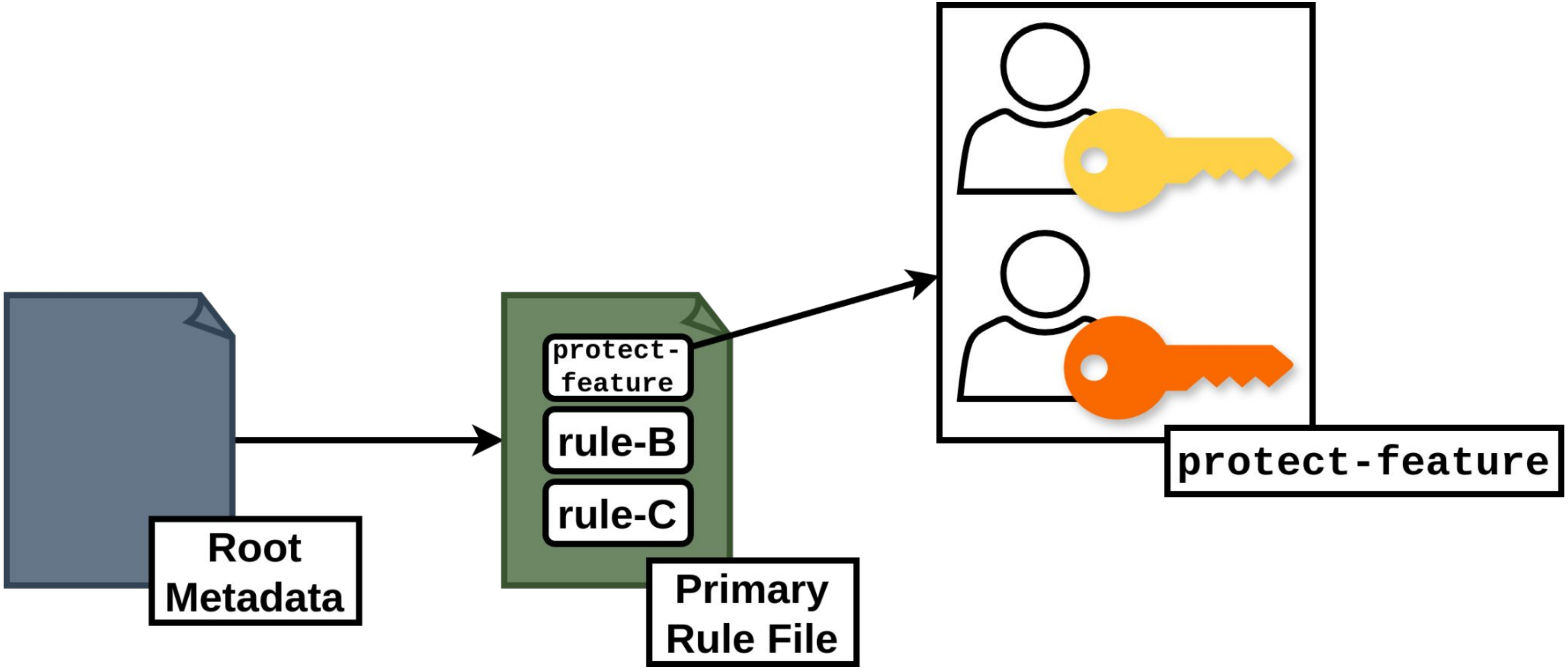
Forges



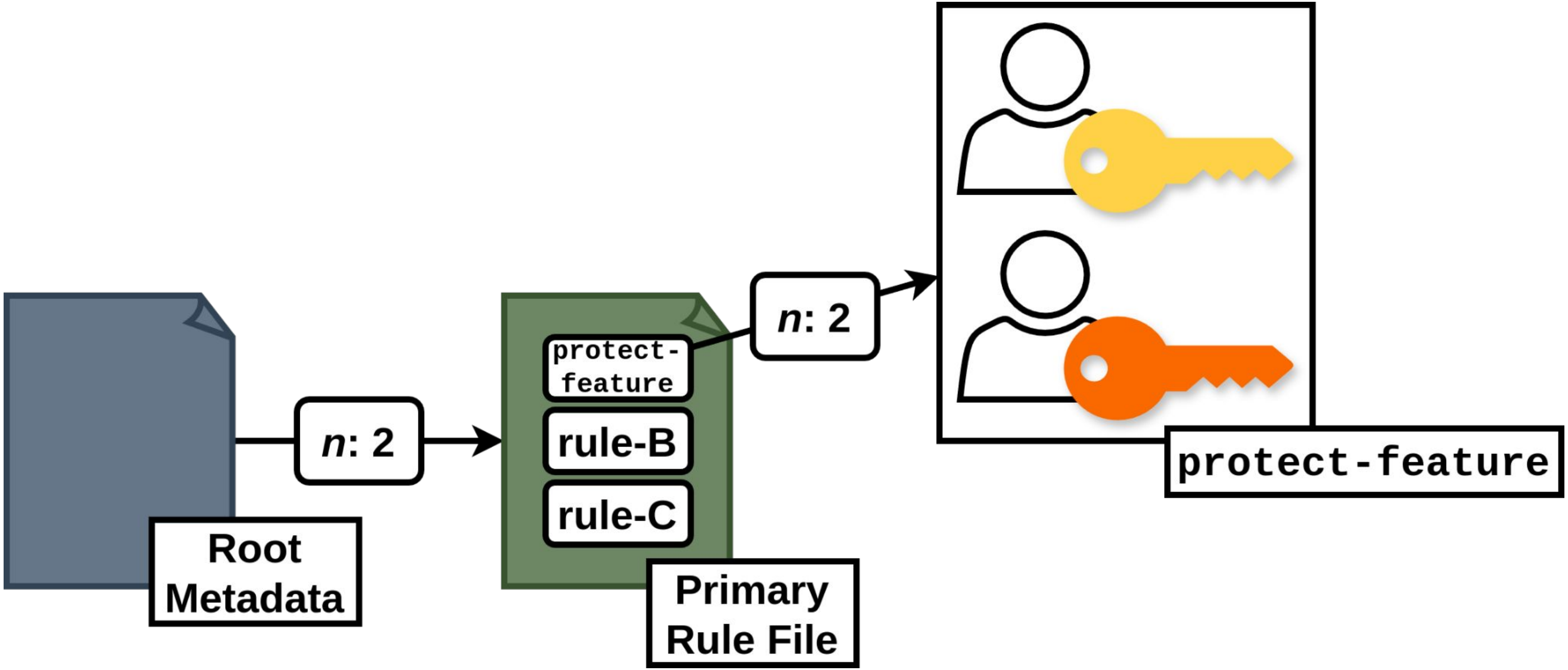
1. User can use multiple credentials
2. *n* users are required to edit policy
3. Users can delegate their permissions
4. Changes are stored in a Reference State Log
5. Forges can still verify changes
6. Users can verify changes themselves

gittuf

gittuf Internals: Delegations



gittuf Internals: Delegations


Root Metadata
n: 2
protect-feature
rule-B
rule-C
Primary Rule File
n: 2
protect-feature