RA-WEBs: Remote Attestation for WEB services

Kosei Akama[*1], **Yoshimichi Nakatsuka**[*2],
Korry Luke[*1], Masaaki Sato[*3], Keisuke Uehara[*1]

[*1] Keio University, [*2] ETH Zurich, [*3] Tokai University

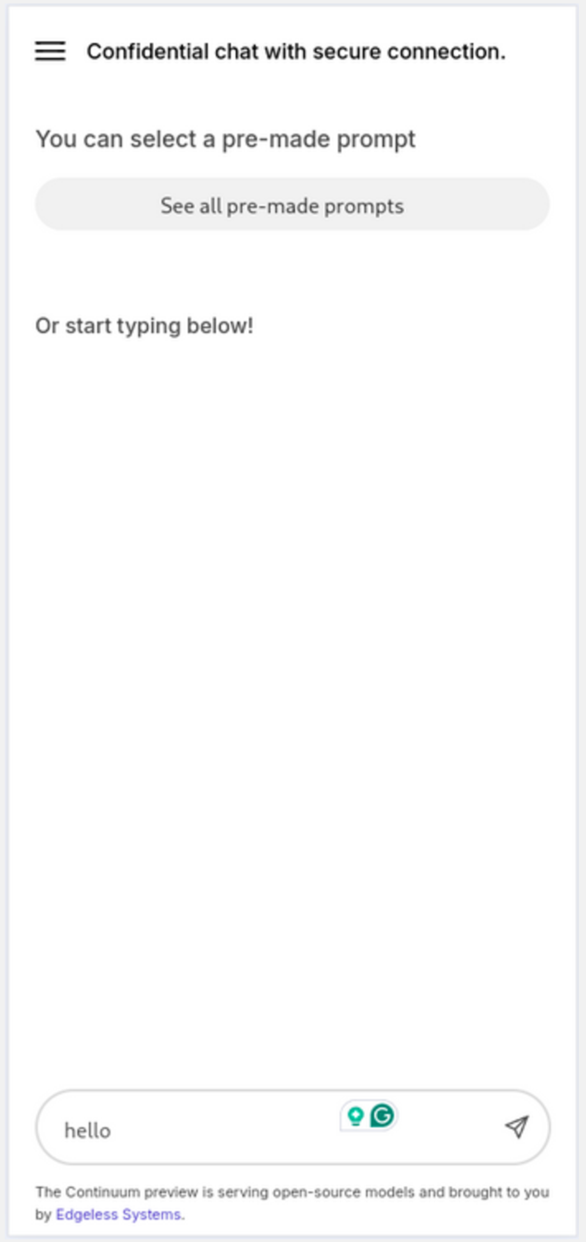
Paper

Live Demo

TEE-enabled Web services

LLM Inference (Continuum)

LLM chat service that protects user prompts



Confidential chat with secure connection.

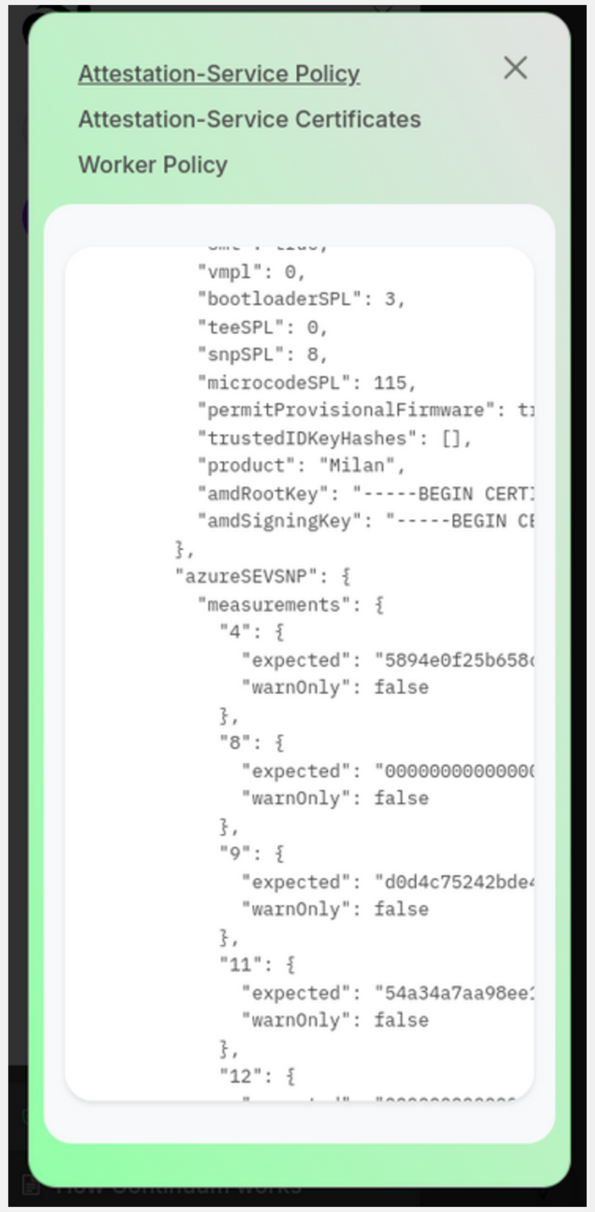
You can select a pre-made prompt

See all pre-made prompts

Or start typing below!

hello

The Continuum preview is serving open-source models and brought to you by Edgeless Systems.

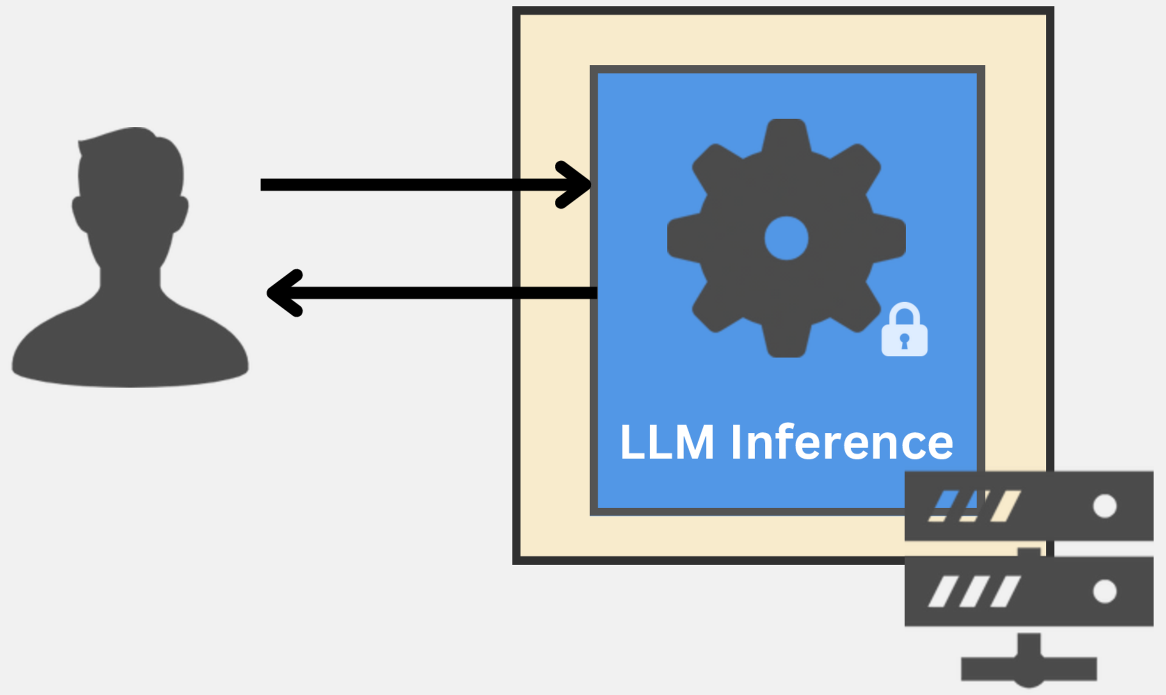


Attestation-Service Policy

Attestation-Service Certificates

Worker Policy

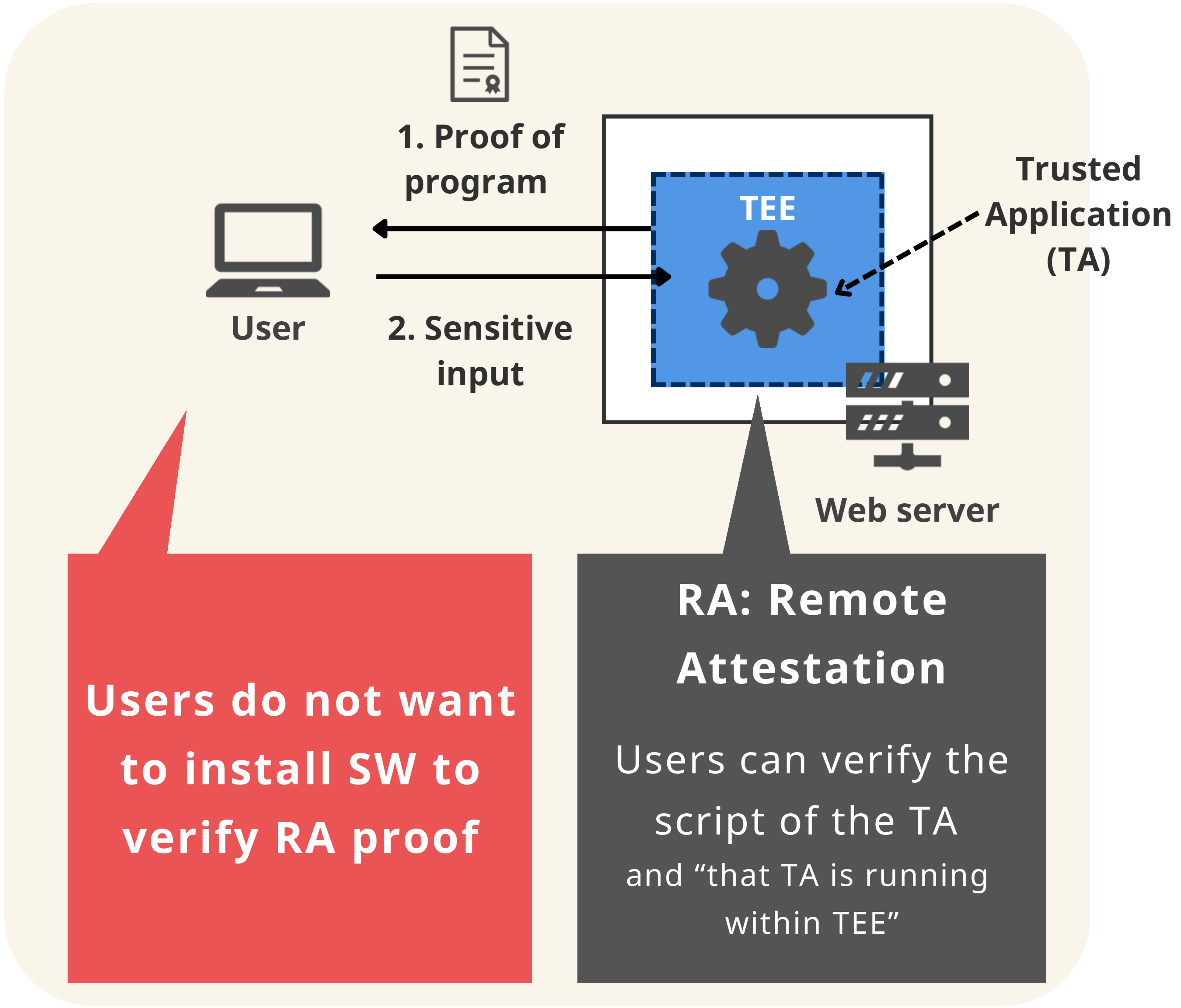
```
        "vmpl": 0,
        "bootloaderSPL": 3,
        "teeSPL": 0,
        "snpSPL": 8,
        "microcodeSPL": 115,
        "permitProvisionalFirmware": t
        "trustedIDKeyHashes": [],
        "product": "Milan",
        "amdRootKey": "-----BEGIN CERT
        "amdSigningKey": "-----BEGIN C
      },
      "azureSEVSNP": {
        "measurements": {
          "4": {
            "expected": "5894e0f25b658
            "warnOnly": false
          },
          "8": {
            "expected": "0000000000000
            "warnOnly": false
          },
          "9": {
            "expected": "d0d4c75242bde
            "warnOnly": false
          },
          "11": {
            "expected": "54a34a7aa98ee
            "warnOnly": false
          },
          "12": {
```



Other use-cases: Non-repudiable Logger, Secure Questionnaire, Privacy-preserving Deep-learning, etc…

Problem with TEE-enabled Web services

- RA Compatibility
- Users need to install software to run RA
  - User friction
  - Information leakage
- Wait for standardization?
  - May not result in standard
  - Cannot force to follow standard



Users do not want to install SW to verify RA proof

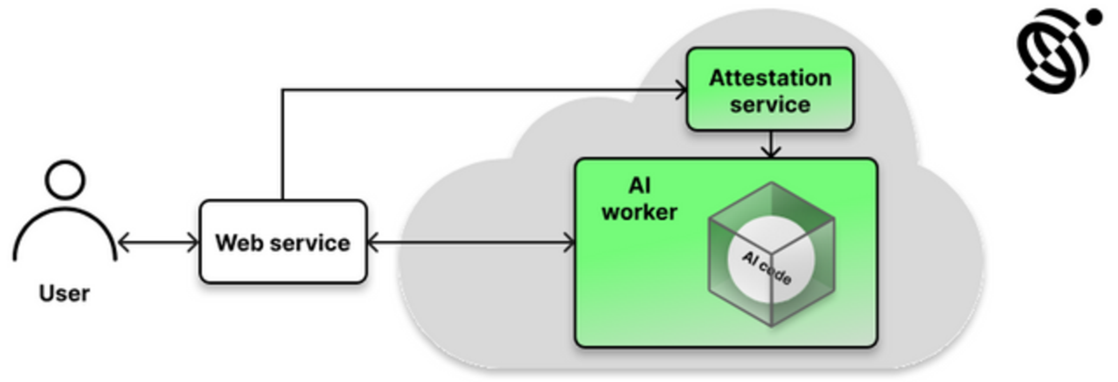
RA: Remote Attestation

Users can verify the script of the TA and "that TA is running within TEE"

Related Work

Continuum Verifier

- **Pro:** Compatible with browsers
- **Con:** Confidentiality issues



Continuum proxy verifier architecture
(Taken from [1])

Remote ATtestation procedureS (RATS)

- **Pro:** Standardized RA model
- **Con:** Issues not clear when adapting RA to Web context

Integrating Attestation into TLS & DTLS

- **Pro:** Fully transparent to user
- **Con:** Standardization & Development process expected to take a long time

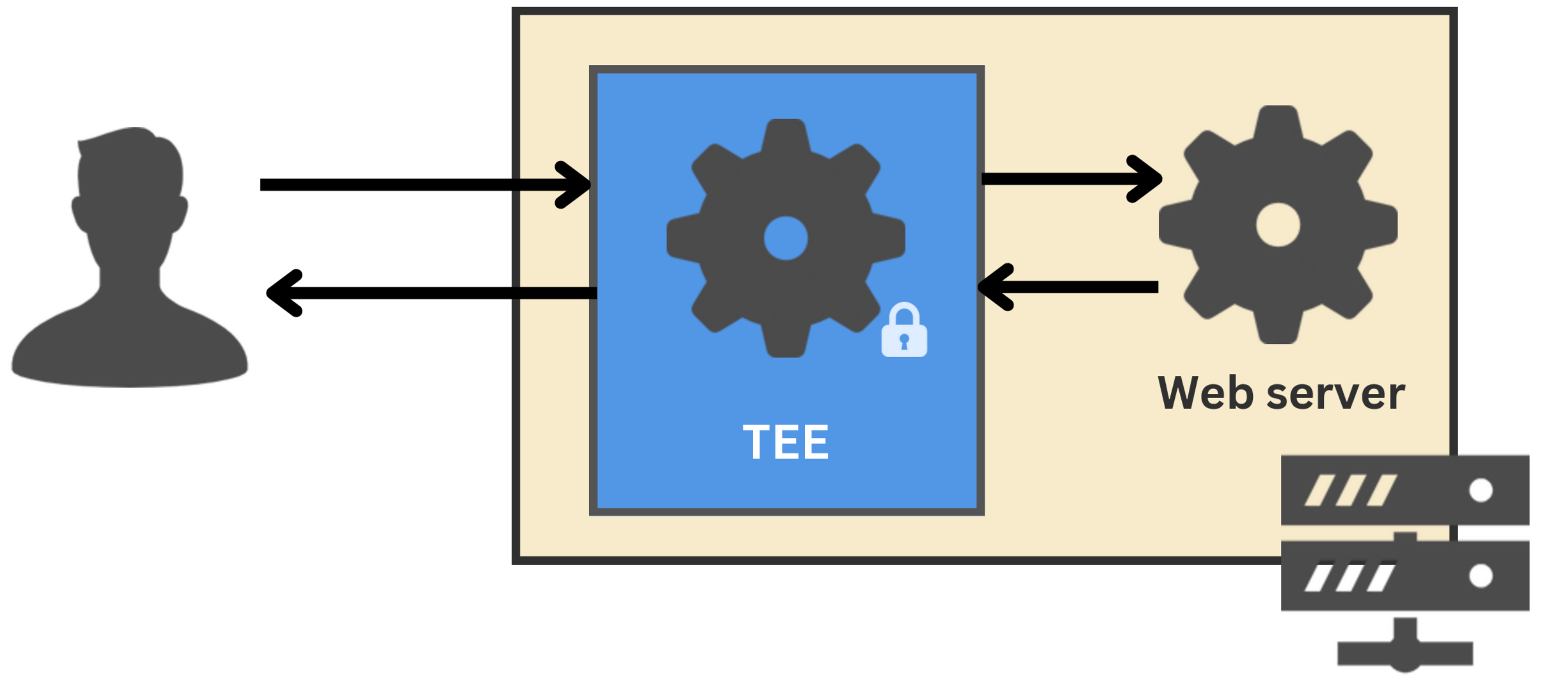
[1] https://docs.edgeless.systems/continuum/under-the-hood/overview

Introducing RA-WEBs (Remote Attestation for WEB services)

RA-WEBs is:

Highly compatibile with the current web ecosystem

- Built using known, well-established web mechanisms
- Users can verify RA proofs using existing browsers without installing any additional software

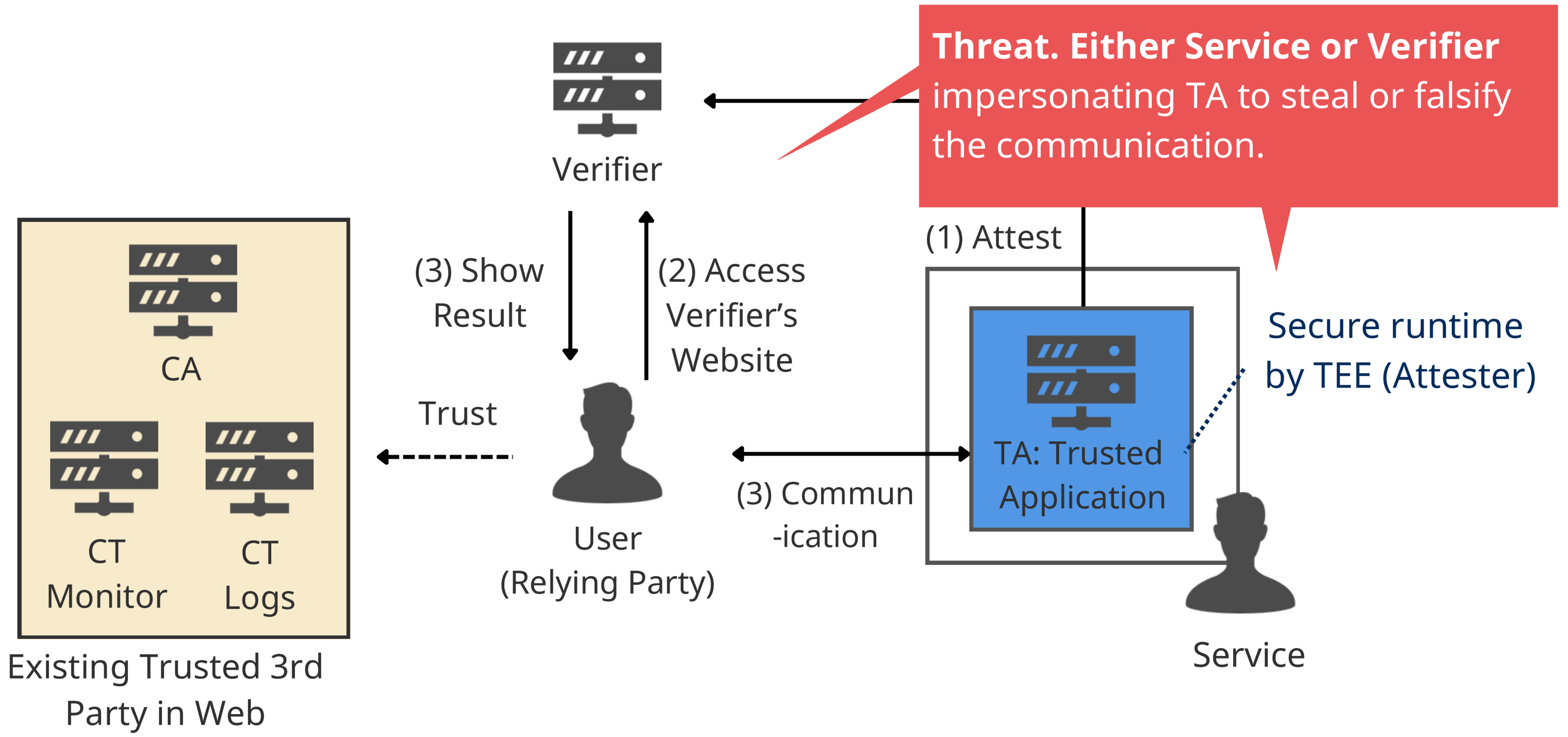


Immediately deployable

Live Demo

System & Threat Model

- Introduce **untrusted** third party: Verifier
  - Verify TA on User's behalf
  - Publishes verification results via website
- Note: Service and Verifier are assumed to not collude



Threat. Either Service or Verifier impersonating TA to steal or falsify the communication.
Verifier
(1) Attest
(3) Show Result
(2) Access Verifier's Website
CA
CT Monitor
CT Logs
Existing Trusted 3rd Party in Web
Trust
User (Relying Party)
(3) Commun -ication
TA: Trusted Application
Secure runtime by TEE (Attester)
Service

Challenges & Solutions 1/2

Background:

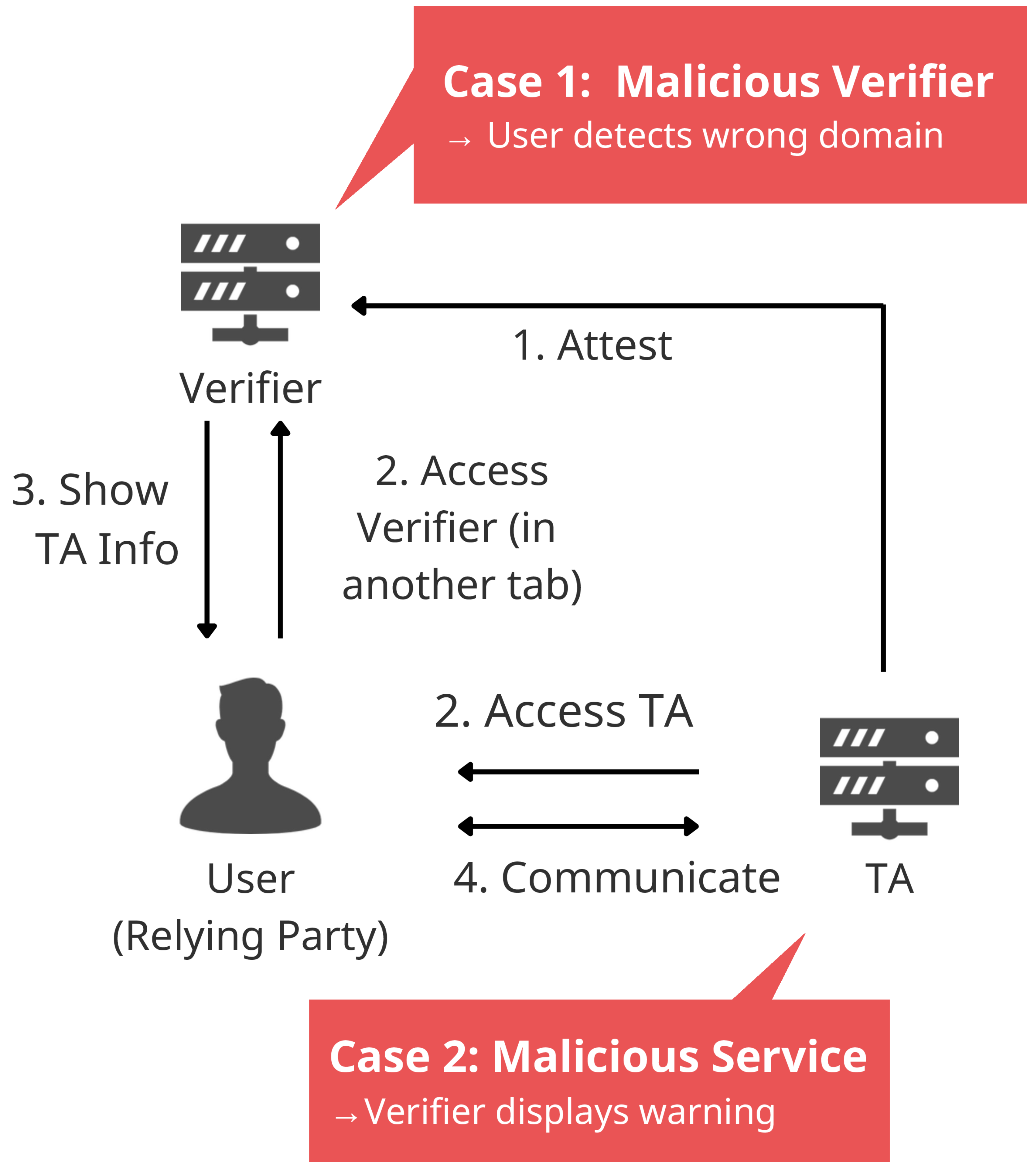
Users must check the TA information

Challenge:

How to obtain & verify TA information?

Solution:

Verifier verifies the proof of TA on User’s behalf and shows TA information to Users


Case 1: Malicious Verifier
→ User detects wrong domain
Verifier
1. Attest
3. Show TA Info
2. Access Verifier (in another tab)
2. Access TA
User (Relying Party)
4. Communicate
TA
Case 2: Malicious Service
→ Verifier displays warning

Challenges & Solutions 2/2

Background:

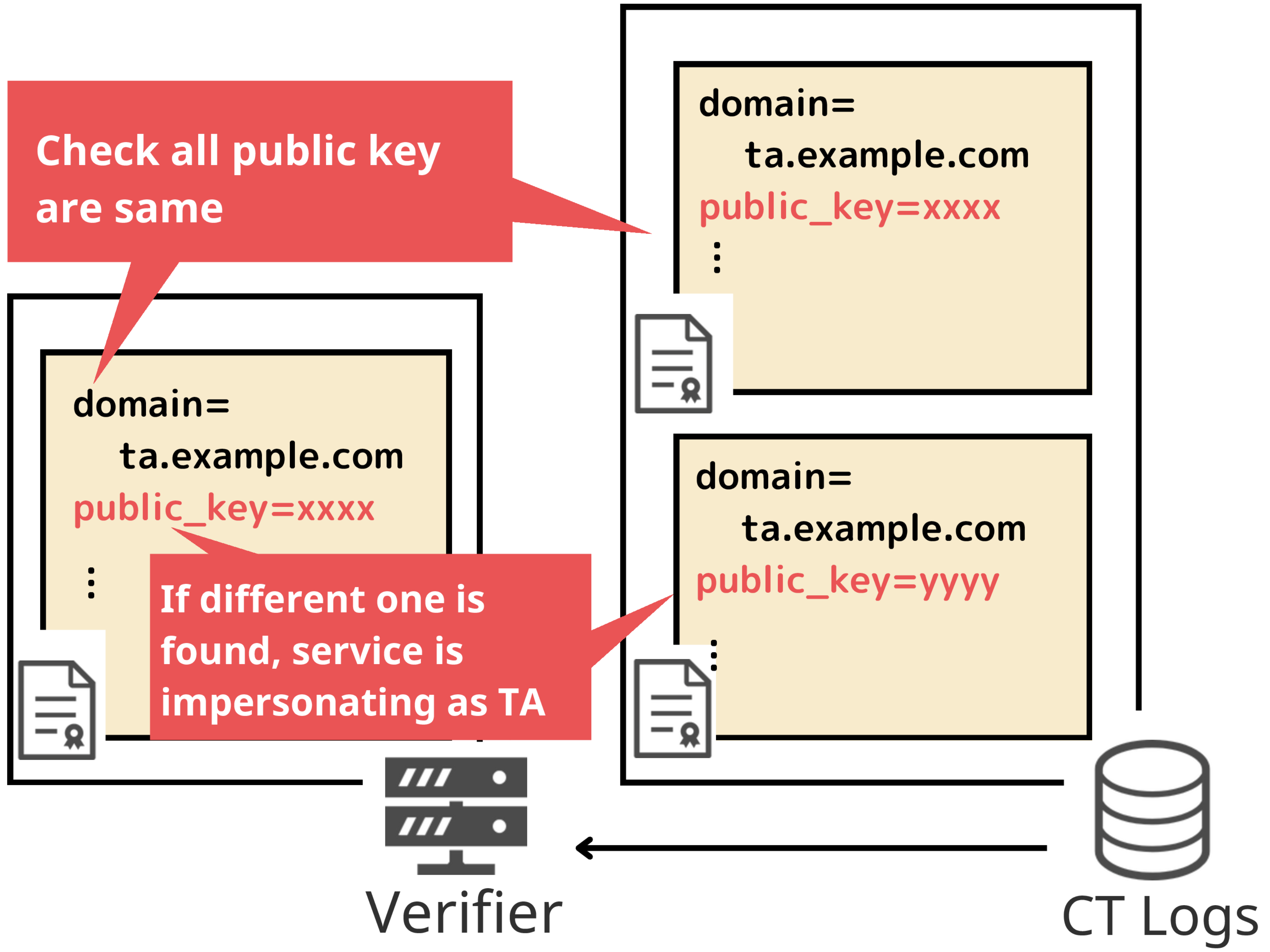
Specific address must be assigned to TA

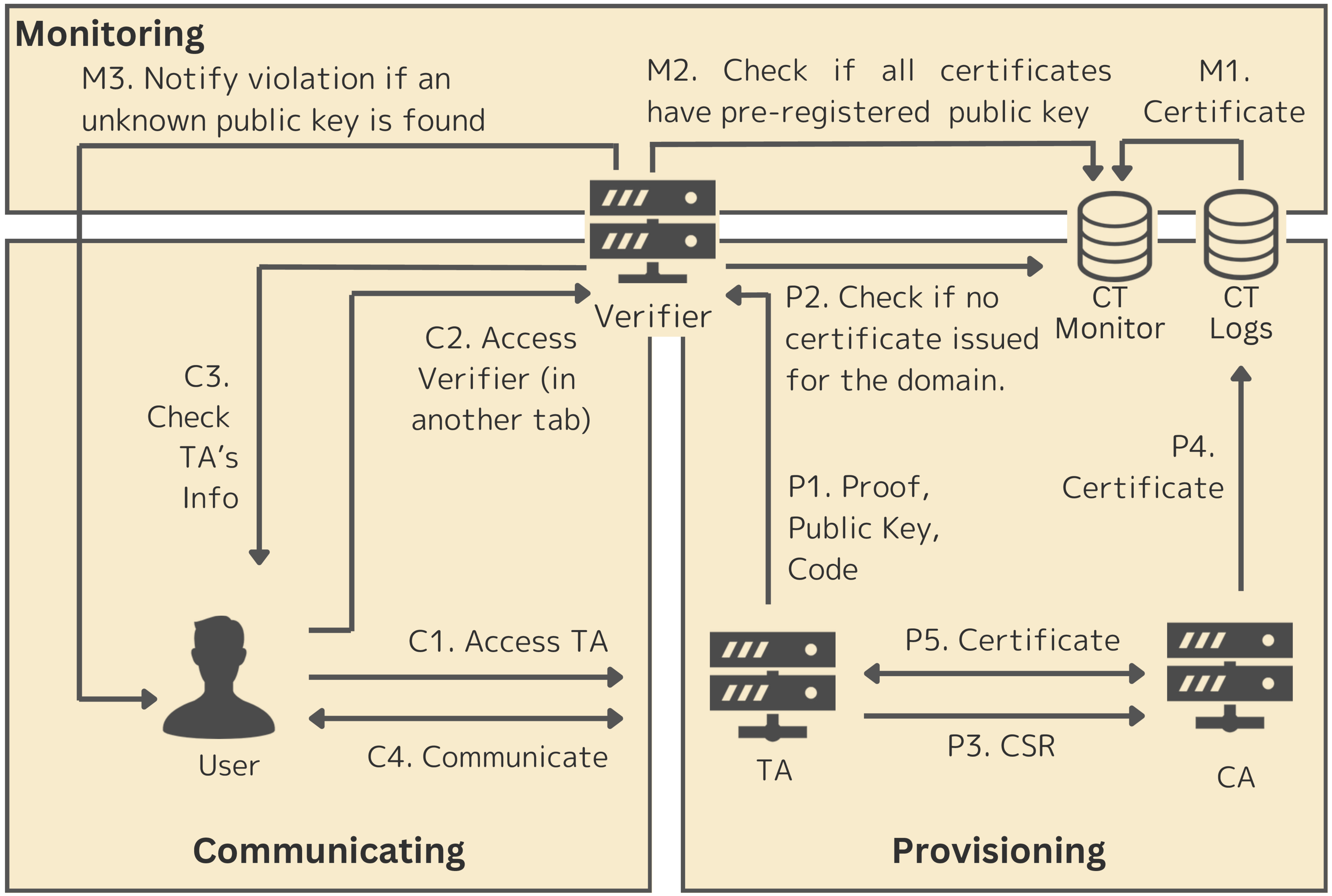
Challenge:

Service reassigns TA address to another machine

Solution:

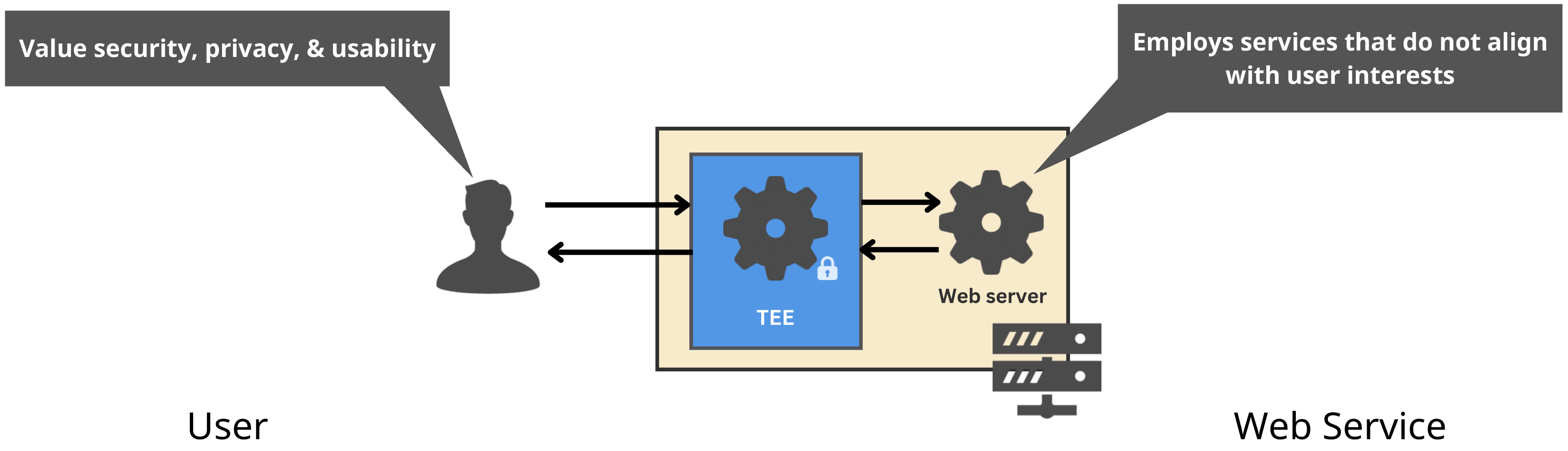
Verifier monitors all public keys assigned to TA address → **Monitor using CT**




Monitoring
M3. Notify violation if an unknown public key is found
M2. Check if all certificates have pre-registered public key
M1. Certificate
Verifier
CT Monitor
CT Logs
C3. Check TA's Info
C2. Access Verifier (in another tab)
P2. Check if no certificate issued for the domain.
P4. Certificate
P1. Proof, Public Key, Code
C1. Access TA
C4. Communicate
User
P5. Certificate
P3. CSR
TA
CA
Communicating
Provisioning

How RA-WEBs Works in Real-World Scenarios

- Security- & Privacy-conscious Users
- TEE-enabled Web services


Value security, privacy, & usability
Employs services that do not align with user interests
TEE
Web server
User
Web Service

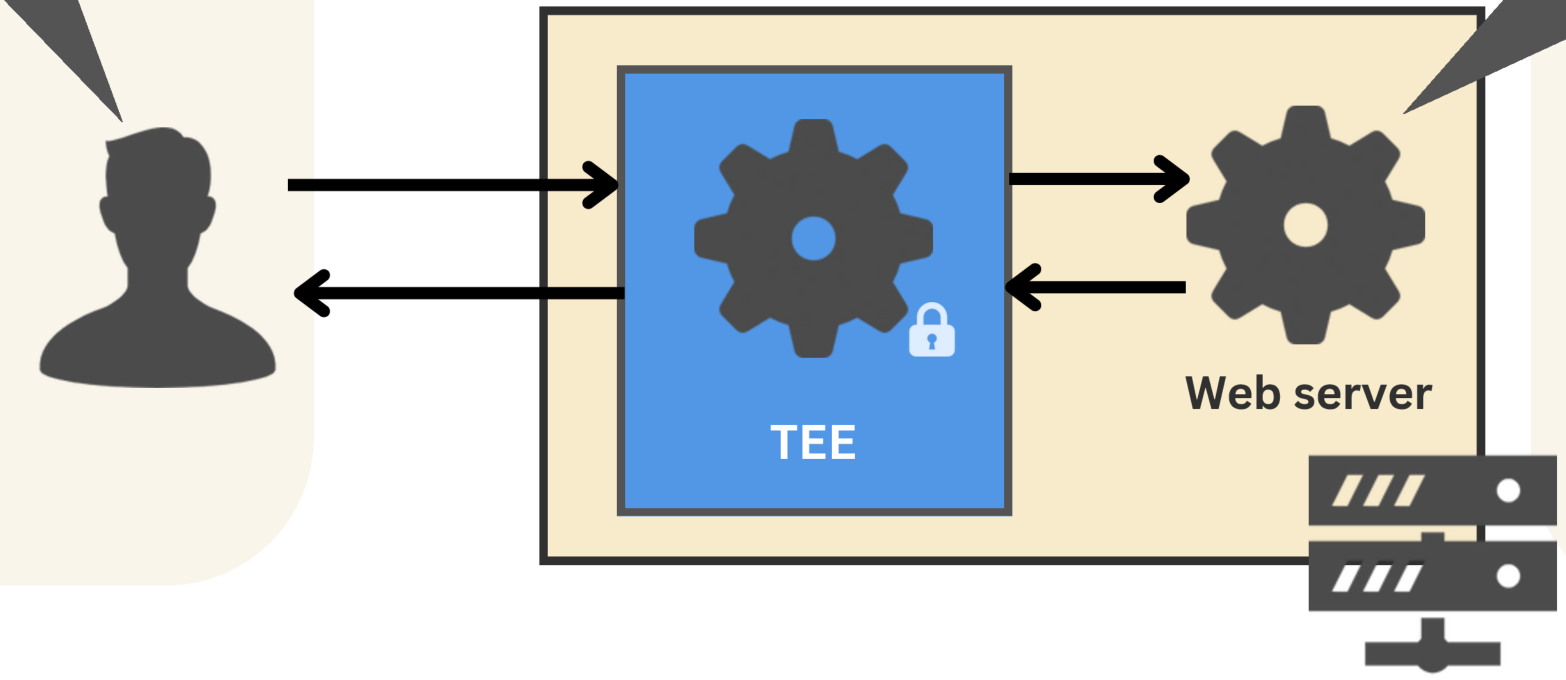
How RA-WEBs Works in Real-World Scenarios

- **Example use-case: Training ML models with Web data**

Value security, privacy, & usability

- Do not want their data to be exposed during training
- Want to benefit from trained ML model

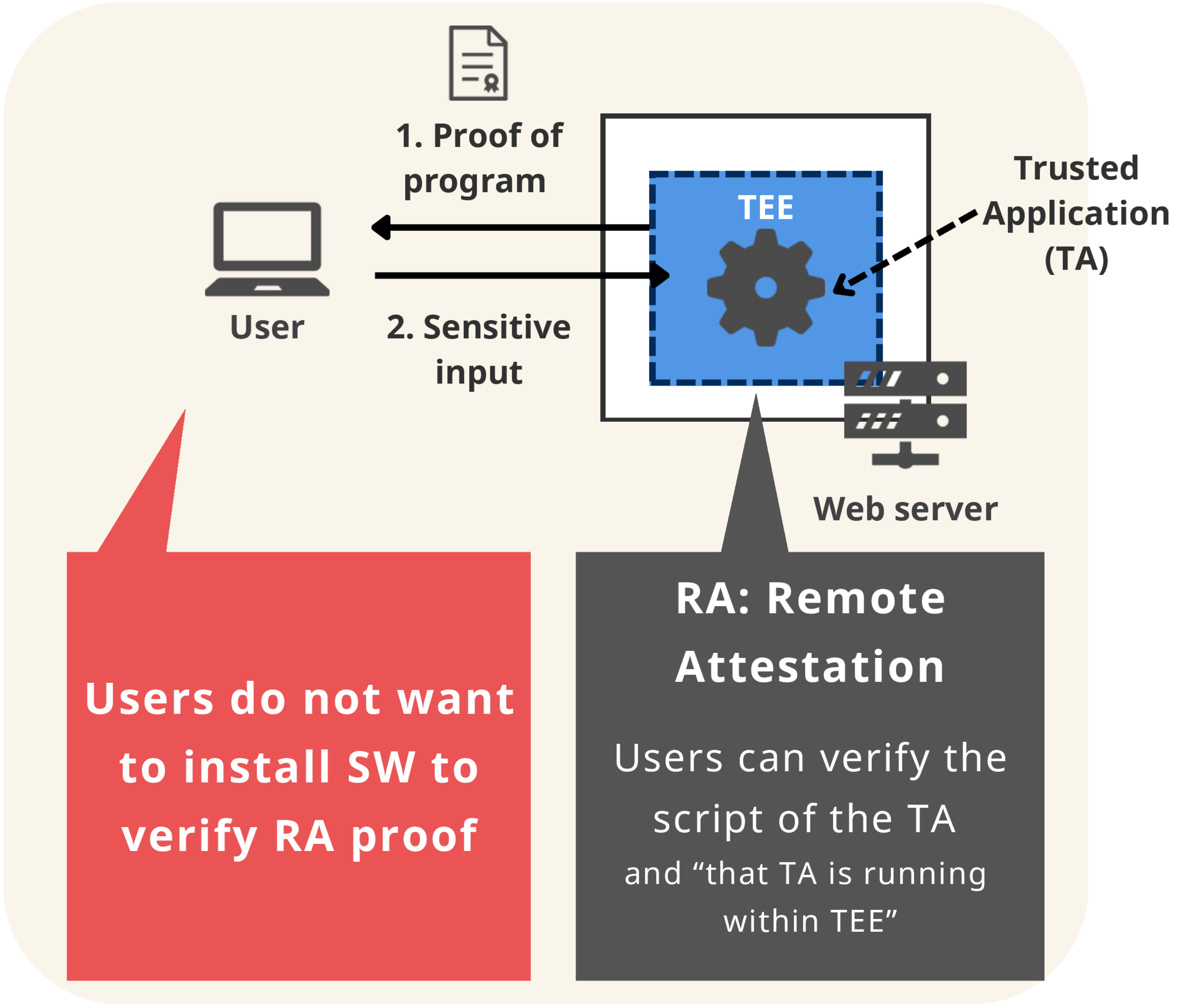
User



Employs services that do not align with user interests

- Want to train ML model using User data
- Want to respect User privacy and comply to data protection laws

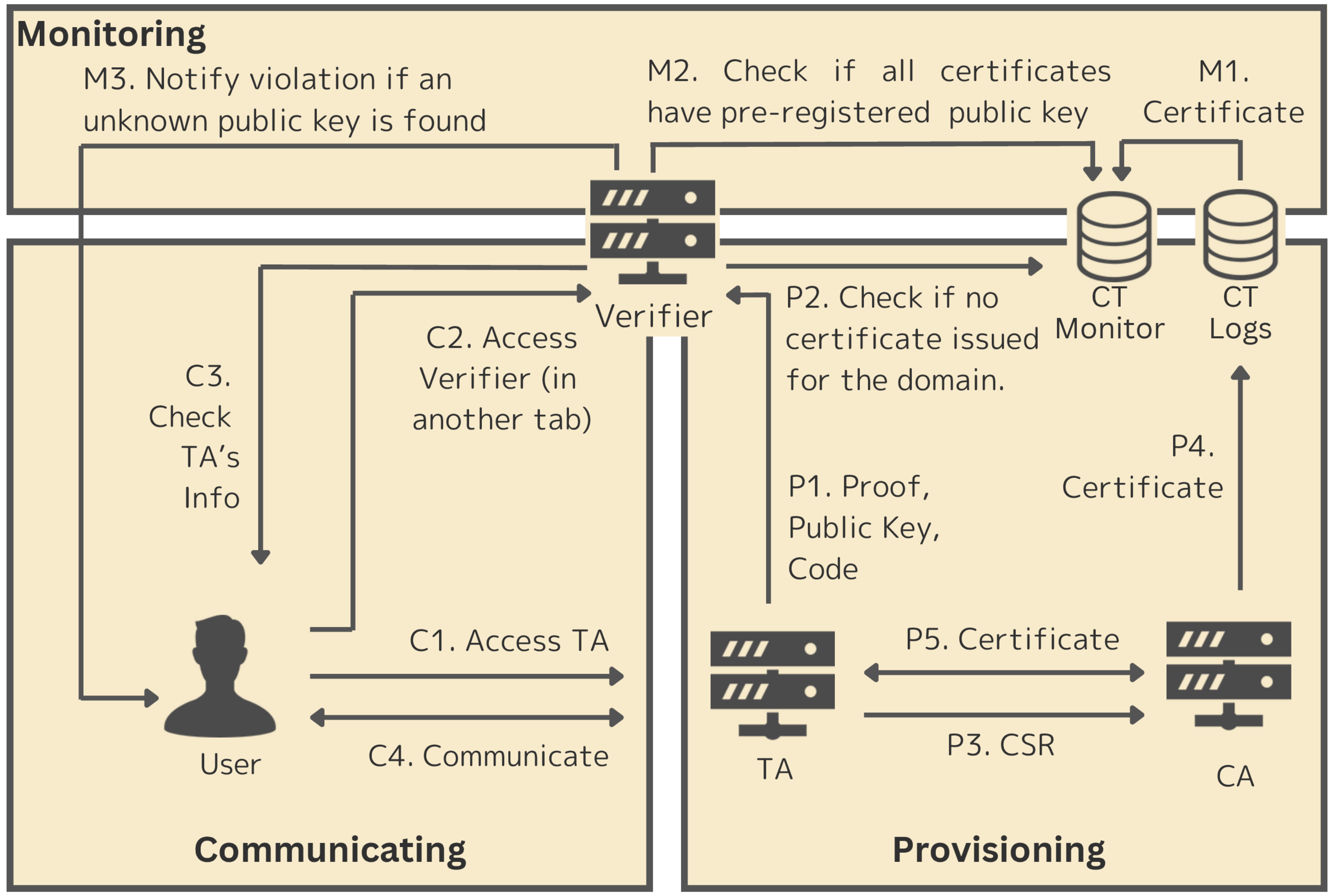
Web Service



Users do not want to install SW to verify RA proof

RA: Remote Attestation

Users can verify the script of the TA and "that TA is running within TEE"


Monitoring
M3. Notify violation if an unknown public key is found
M2. Check if all certificates have pre-registered public key
M1. Certificate
Verifier
CT Monitor
CT Logs
C3. Check TA's Info
C2. Access Verifier (in another tab)
P2. Check if no certificate issued for the domain.
P4. Certificate
P1. Proof, Public Key, Code
C1. Access TA
C4. Communicate
User
P5. Certificate
P3. CSR
TA
CA
Communicating
Provisioning

Paper

Live Demo

Questions?

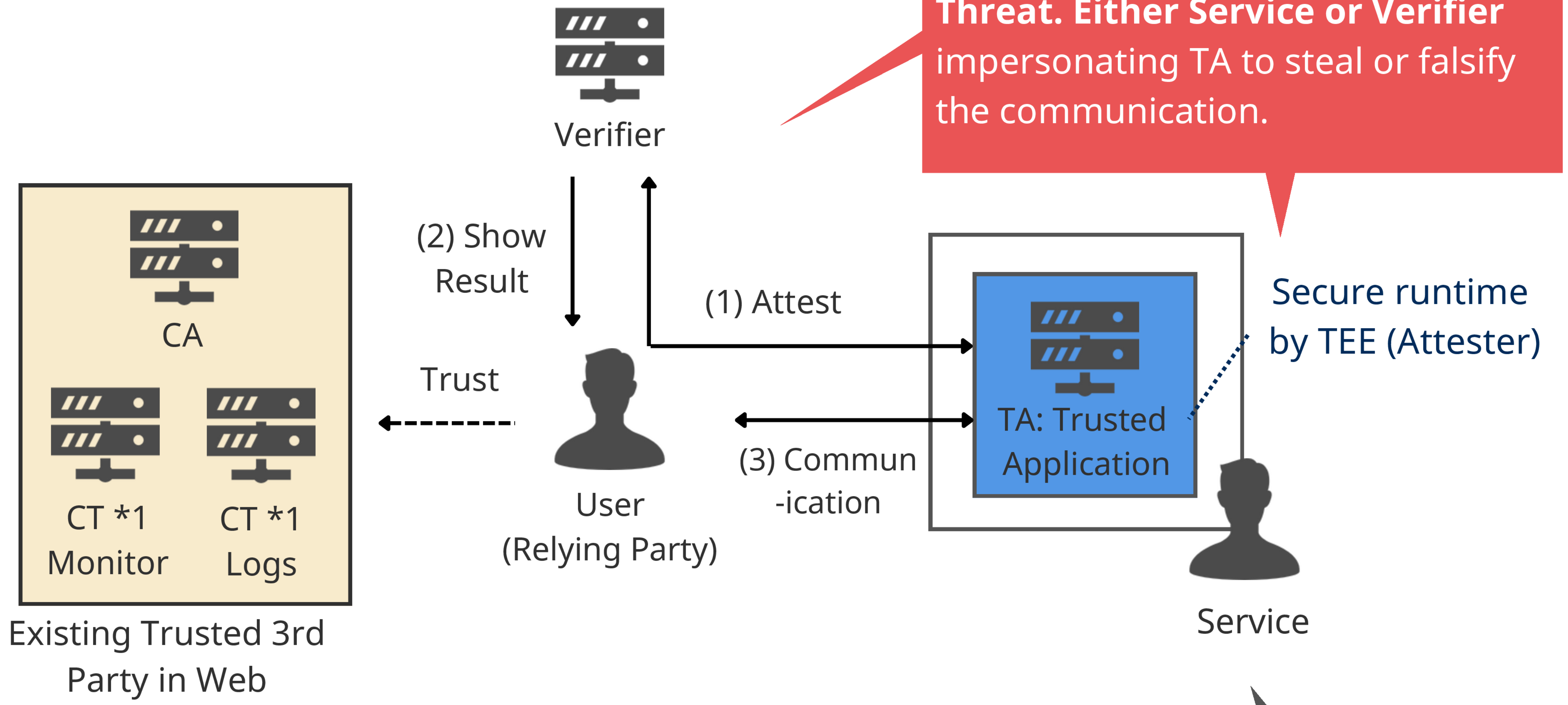
Email: yoshimichi.nakatsuka@inf.ethz.ch
Website: yoshinakatsuka.com

Appendix

System & Threat Model

- Introduce **untrusted** third party: Verifier
- Service and Verifier are assumed to not collude (similar to ODoH, OHTTP).

Threat. Either Service or Verifier impersonating TA to steal or falsify the communication.



Goals

Security

- Confidentiality
- Integrity

Non-Security

- Compatibility

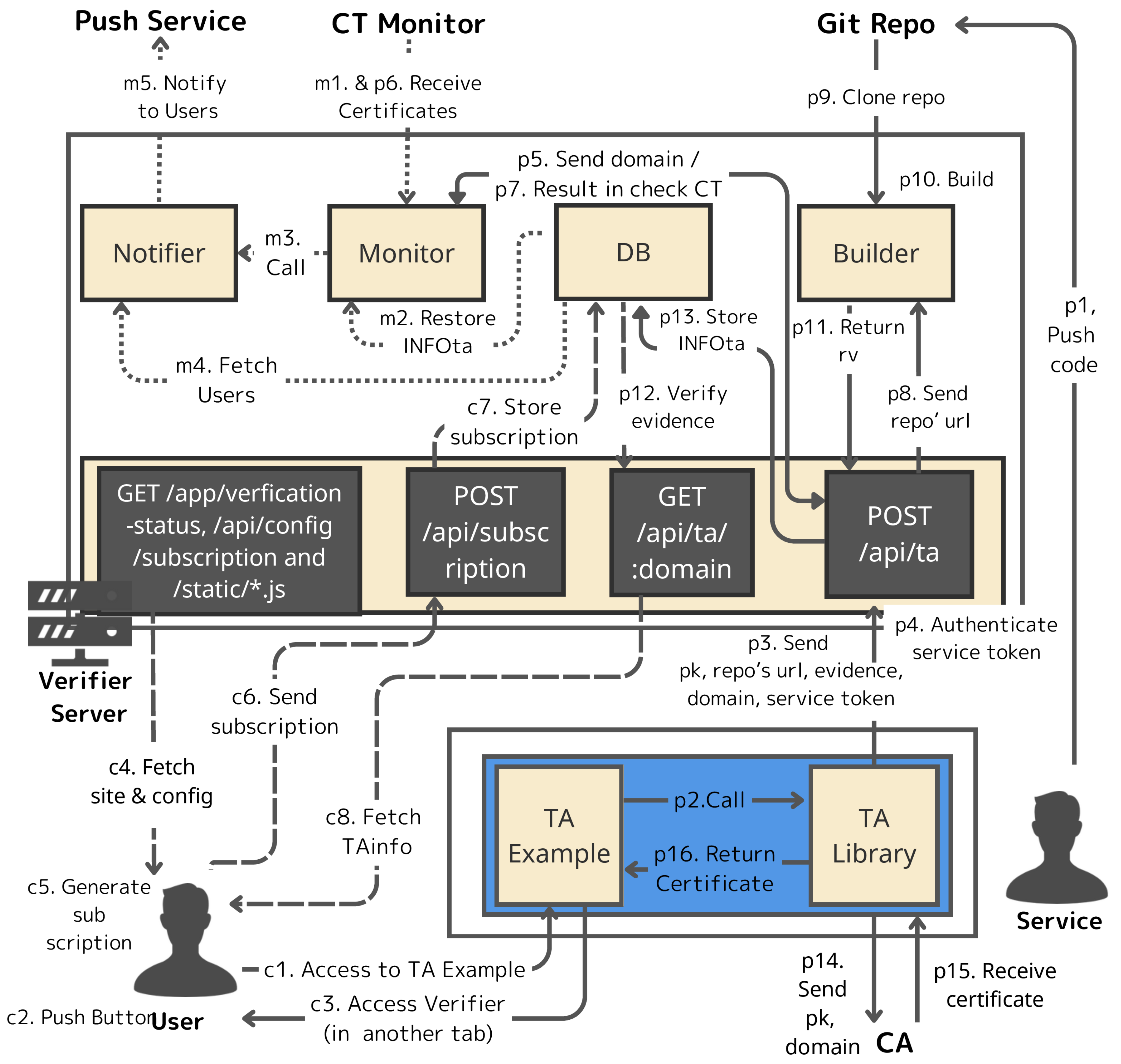
*1 Authorities for auditing CA.
They all have certificates in Web PKI.[5]
CT : Certificate Transparency

Assumption of General Web:
e.g., Users checks the domain, Users trust CA, CT Logs, CT Monitor

Assumption of General RA:
e.g., Users can read TA source code (or depends on OSS ecosystem) and Service publishes TA source code.

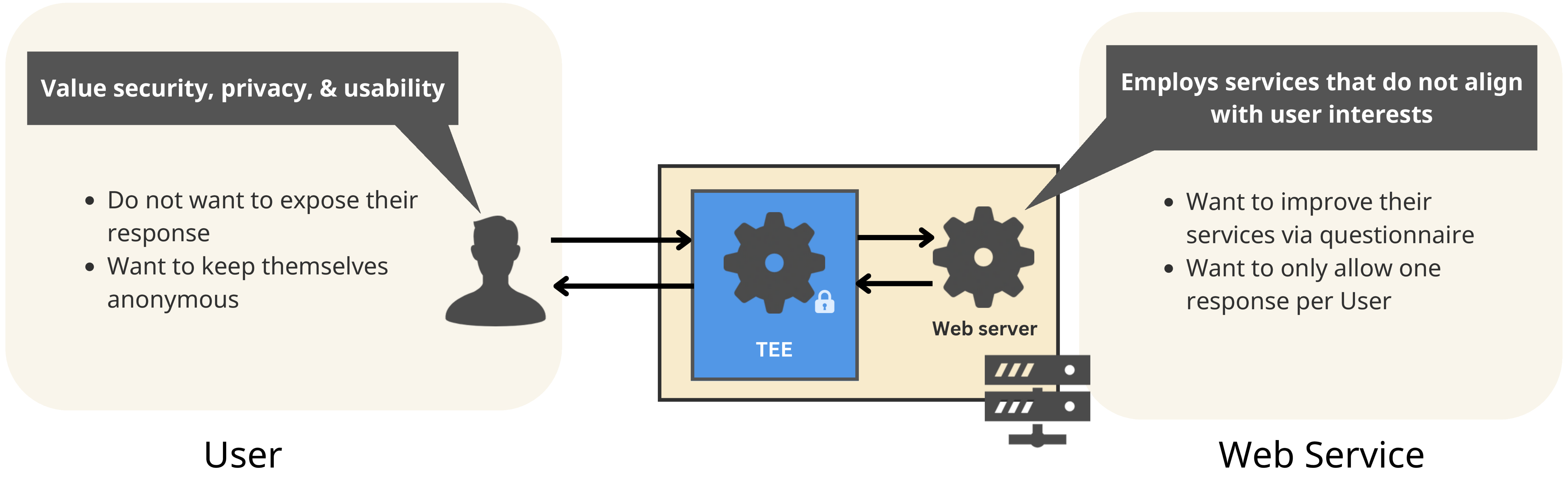
Implementation

Code



How RA-WEBs Works in Real-World Scenarios

- **Example use-case 2: Online questionnaires**


Value security, privacy, & usability
Do not want to expose their response
Want to keep themselves anonymous
User
TEE
Web server
Employs services that do not align with user interests
Want to improve their services via questionnaire
Want to only allow one response per User
Web Service

Security Analysis

By analyzing the security of RA-WEBs, we show that RA-WEBs are secure.

- **Manual Analysis:** Showing threats and how RA-WEBs prevent them.
- **Formal Verification:** Automatic security analysis (Verifpal)
  - We have made two Verifpal models (with either malicious Verifier or Service respectively).
  - While running Verifapl, we found and reported the critical bugs in the Verifpal to a developer and they said “we will fix it soon” but...

Reference

[1] https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/03/CCC_outreach_whitepaper_updated_November_2022.pdf
[2] https://ai.confidential.cloud/#/
[3] Dominik Meißner, Felix Engelmann, Frank Kargl, and Benjamin Erb. 2021. PeQES: A platform for privacy-enhanced quantitative empirical studies. In Proceedings of the 36th Annual ACM Symposium on Applied Computing. 1226–1234.
[4] Pierre-Louis Aublin, Florian Kelbert, Dan O’Keeffe, Divya Muthukumaran, Chris tian Priebe, Joshua Lind, Robert Krahn, Christof Fetzer, David Eyers, and Peter Pietzuch. 2018. LibSEAL: Revealing service integrity violations using trusted execution. In Proceedings of the Thirteenth EuroSys Conference. 1–15.
[5] https://datatracker.ietf.org/doc/html/rfc6962
[6] https://datatracker.ietf.org/doc/html/rfc9334
[7] https://datatracker.ietf.org/doc/rfc9458/
[8] Aozhuo Sun, Jingqiang Lin, Wei Wang, Zeyan Liu, Bingyu Li, Shushang Wen, Qiongxiao Wang, and Fengjun Li. [n. d.]. Certificate Transparency Revisited: The Public Inspections on Third-party Monitors. In Proceedings 2024 Network and Distributed System Security Symposium (San Diego, CA, USA, 2024). Internet Society. https://doi.org/10.14722/ndss.2024.24834
[9] https://gist.github.com/thesamesam/223949d5a074ebc3dce9ee78baad9e27
[10] https://docs.edgeless.systems/continuum/apis/continuum-proxy
[11] https://www.heady.io/blog/market-study-mobile-customer-experience-issues-highlight-use-cases-for-ios-app-clips

Discussion 1/2

Colluding Verifier and Service (Security)

- Verifier and Service may collude to violate user privacy (This is our out-of-scope)
- Countermeasures:
  - Multiple Verifier
  - Verifier running in TEE *

Delay of CT system (Security)

- Service may impersonate TA during a CT delay (3 days ~ 7 days)[8].
- We consider that delays are enough to short
  given that the discovering backdoor takes several days to months [9].
- Also, we can pursue the responsibility of services’ malicious activities.

*but require a distributed Verifier to monitor the Verifier

Discussion 2/2

Pursuing responsibility (Security)

- Service may violate user privacy using malicious TA, or impersonation during CT delay.
- Verifier can pursue the responsibility of the violations to service.

The burden of checking source code (Usability)

- Checking the source code is a tough task for non-developer users.
- We can utilize the Open Source Software(OSS) ecosystem.
  e.g. Many people do not read Linux Kernel but trust it because others check.
- We can also integrate with OSS auditing service.