Delegating Verification for Remote Attestation using TEE

Takashi Yagawa[1], Tadanori Teruya[2],
Kuniyasu Suzaki[3], Hirotake Abe[1]

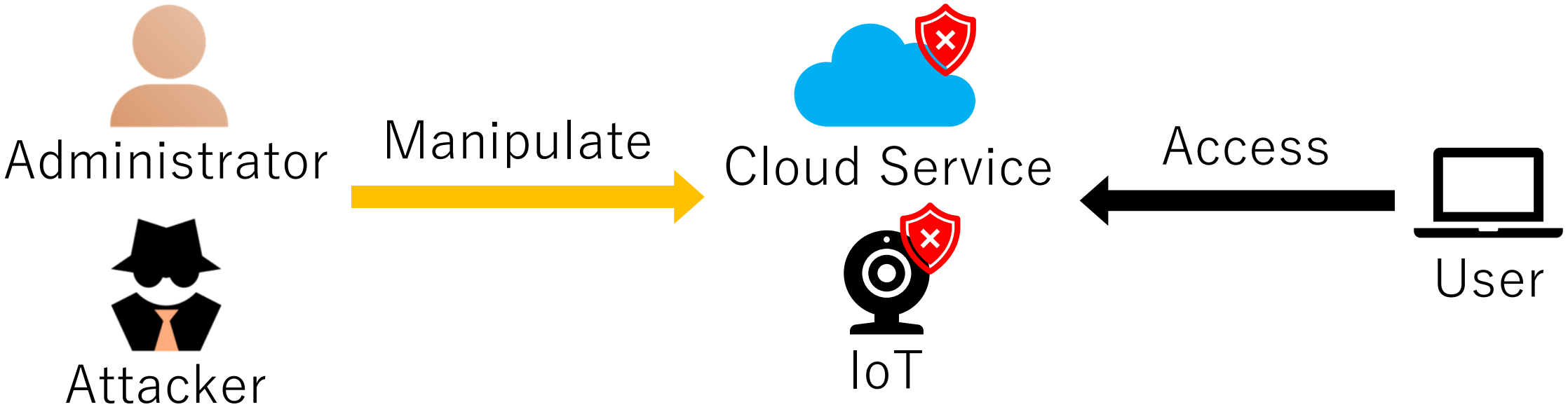
[1]University of Tsukuba, Japan
[2]AIST, Japan
[3]Institute of Information Security, Japan

7th Workshop on System Software for Trusted Execution
July 8th, 2024

Background: Remote Platform is unsafe

Cloud Service and IoT are becoming more widespread.
They are both managed in physically remote location.

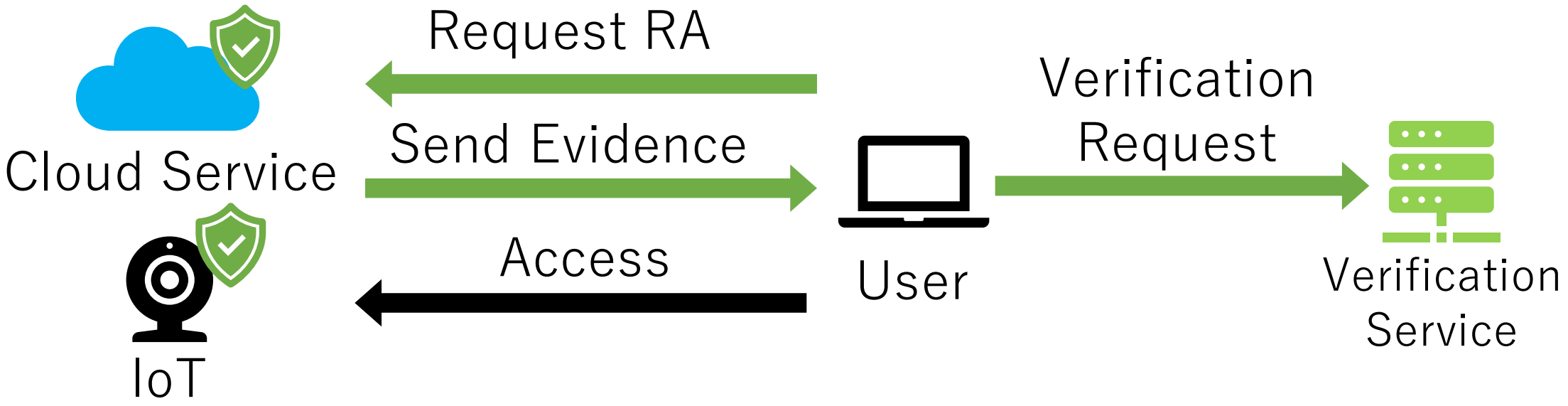
However, that situation allows a malicious administrator or attacker to manipulate the remote platforms.



Remote Attestation required

To prevent such threat, Remote Attestation (RA) is important.

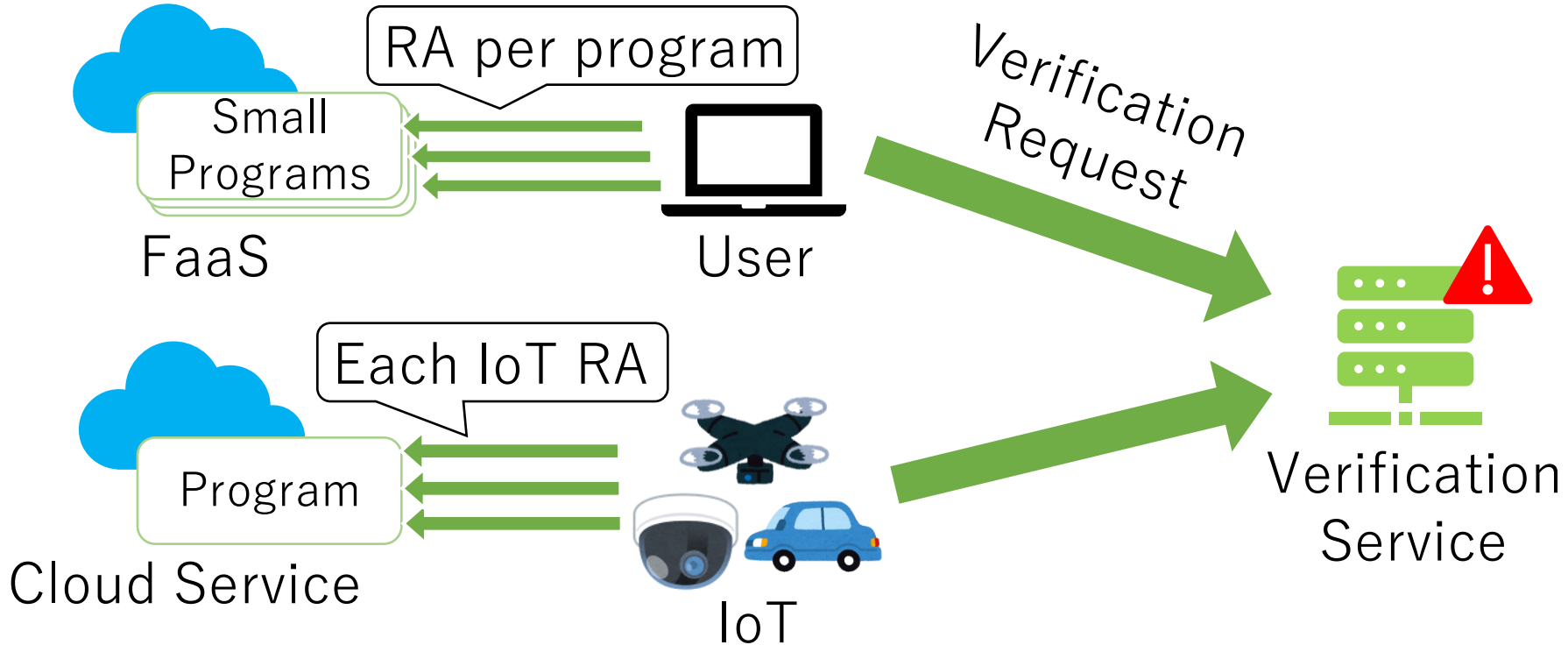
RA allows users to remotely check the status of a device and service.



Challenge: Increase in Remote Attestation

With Function as a Service (FaaS) and Edge Computing becoming more popular, RA is more required.

However, existing verification services do not have the shorter response time nor scalability.

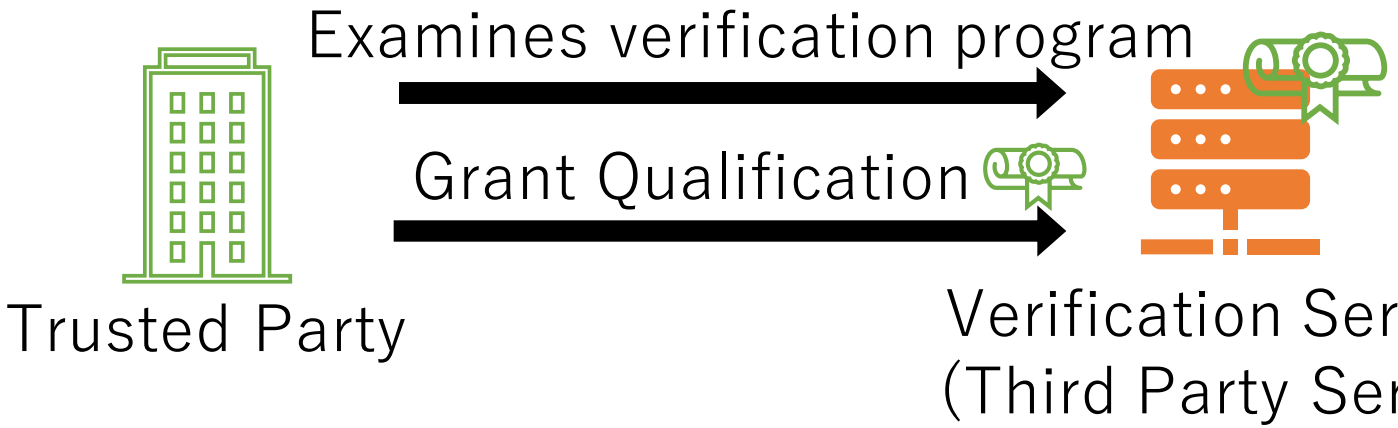


Our Approach：Delegating Verification

There is a limitation that only Trusted Party (e.g. CA) can verify an evidence.

=> We propose **Delegating Verification :**

- Trusted Party delegates a qualification of verification to third parties.
- In delegation process, verification program is run with Trusted Execution Environment (TEE) and Trusted Party examines it.

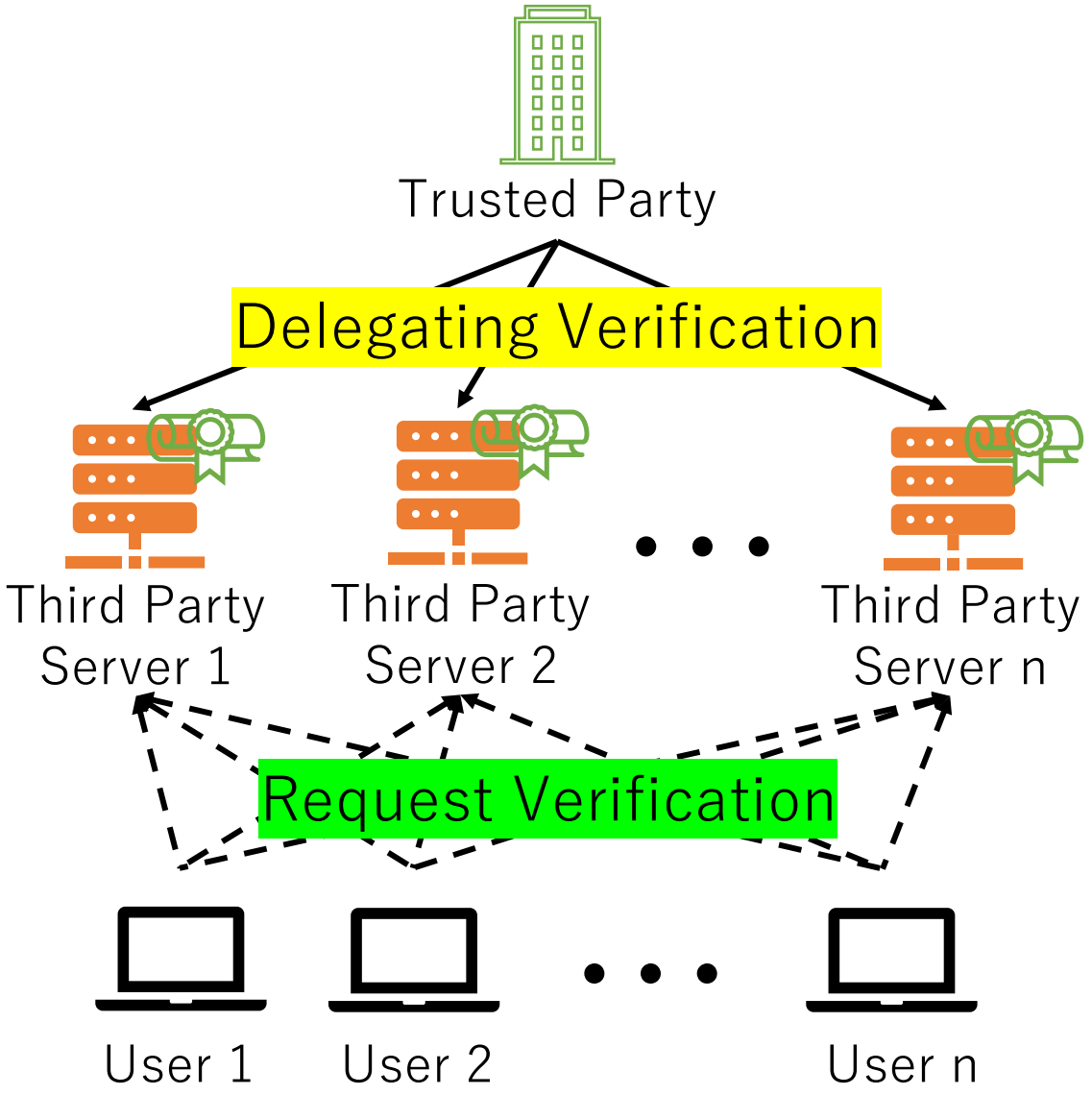


Benefits of Our Approach

- **Quick Response**
  => Third Party Servers can be geographically distributed.

- **Trustworthy**
  => Trust anchor is Trusted Party as now.

- **Scalable**
  => Delegating Verification setup new verifier more quickly than PKI mechanism.



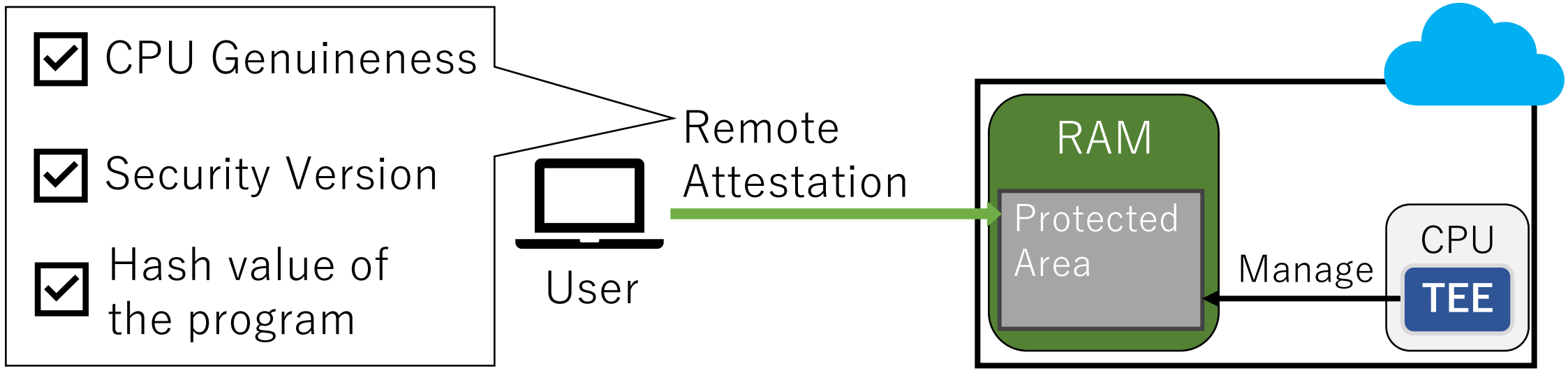
Contents

- Introduction
- Design
- Implementation & Evaluation
- Related Works
- Conclusion & Future Works

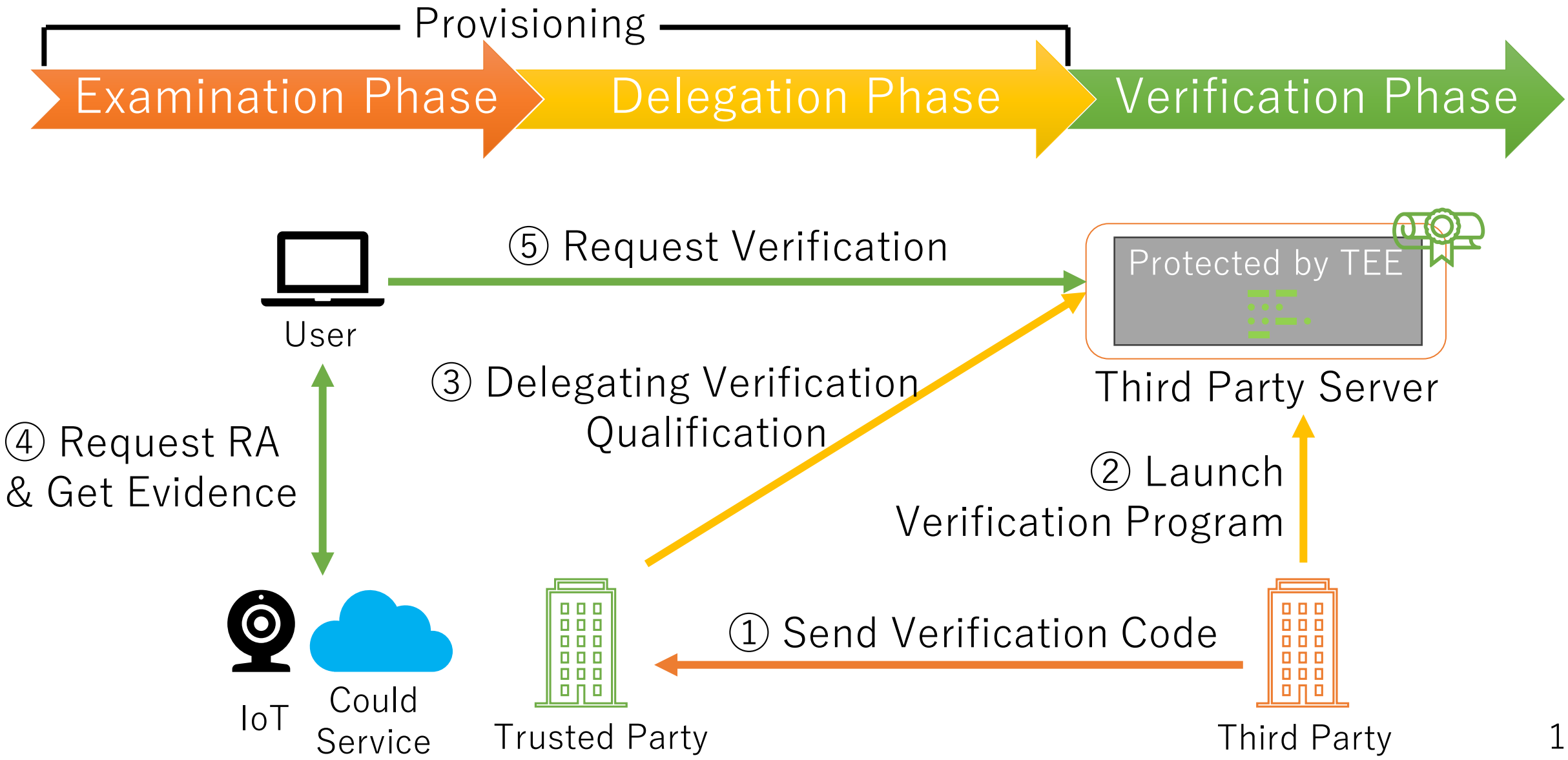
Contents

TEE and its RA

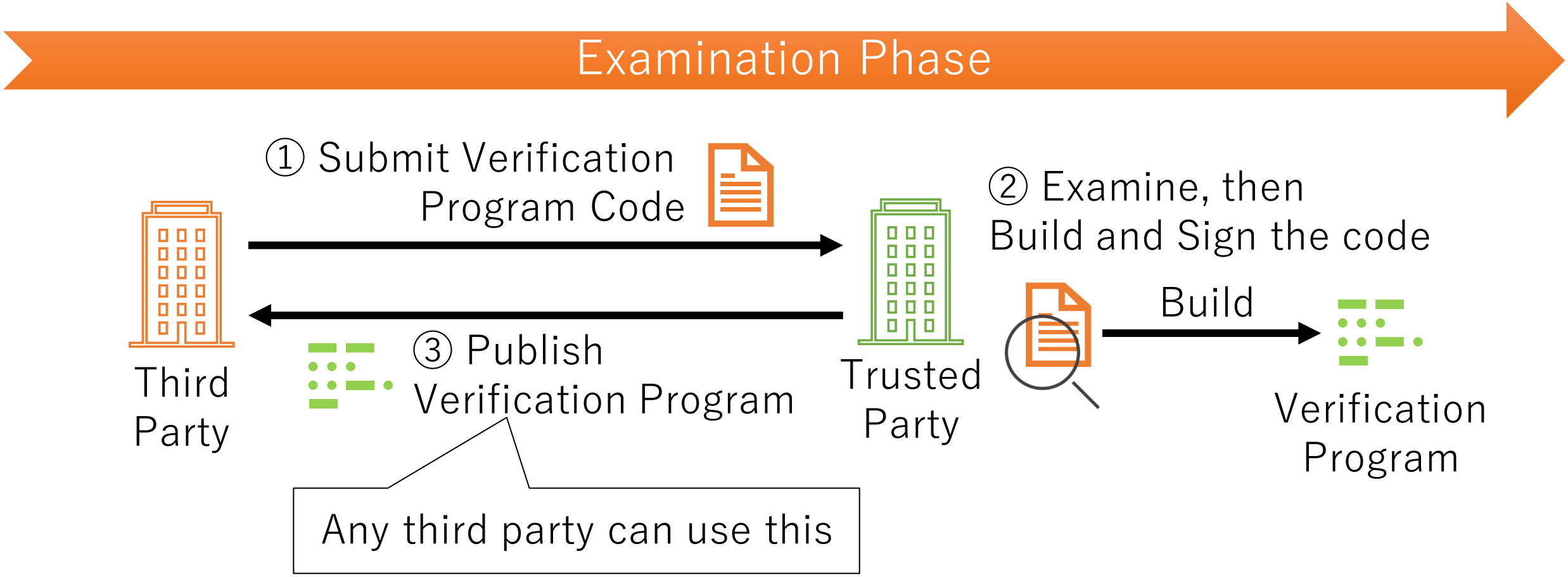
- An extension of CPU for memory protection that uses a key burned into CPU as the Root of Trust.
- TEE keeps confidentiality in Protected Area through memory encryption and privilege management.
- Its RA will confirm that the integrity of TEE and the program in Protected Area.



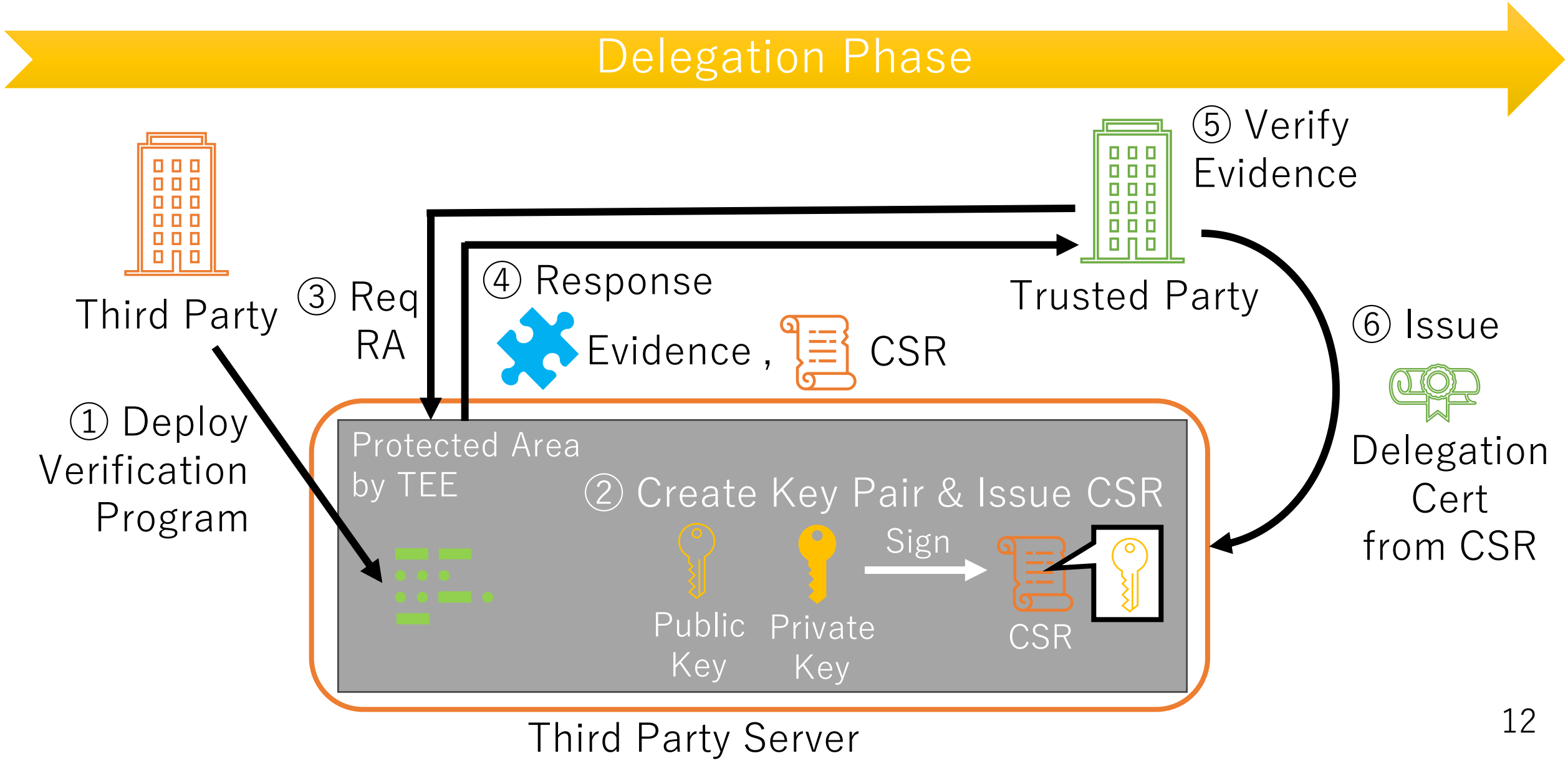
Design Overview



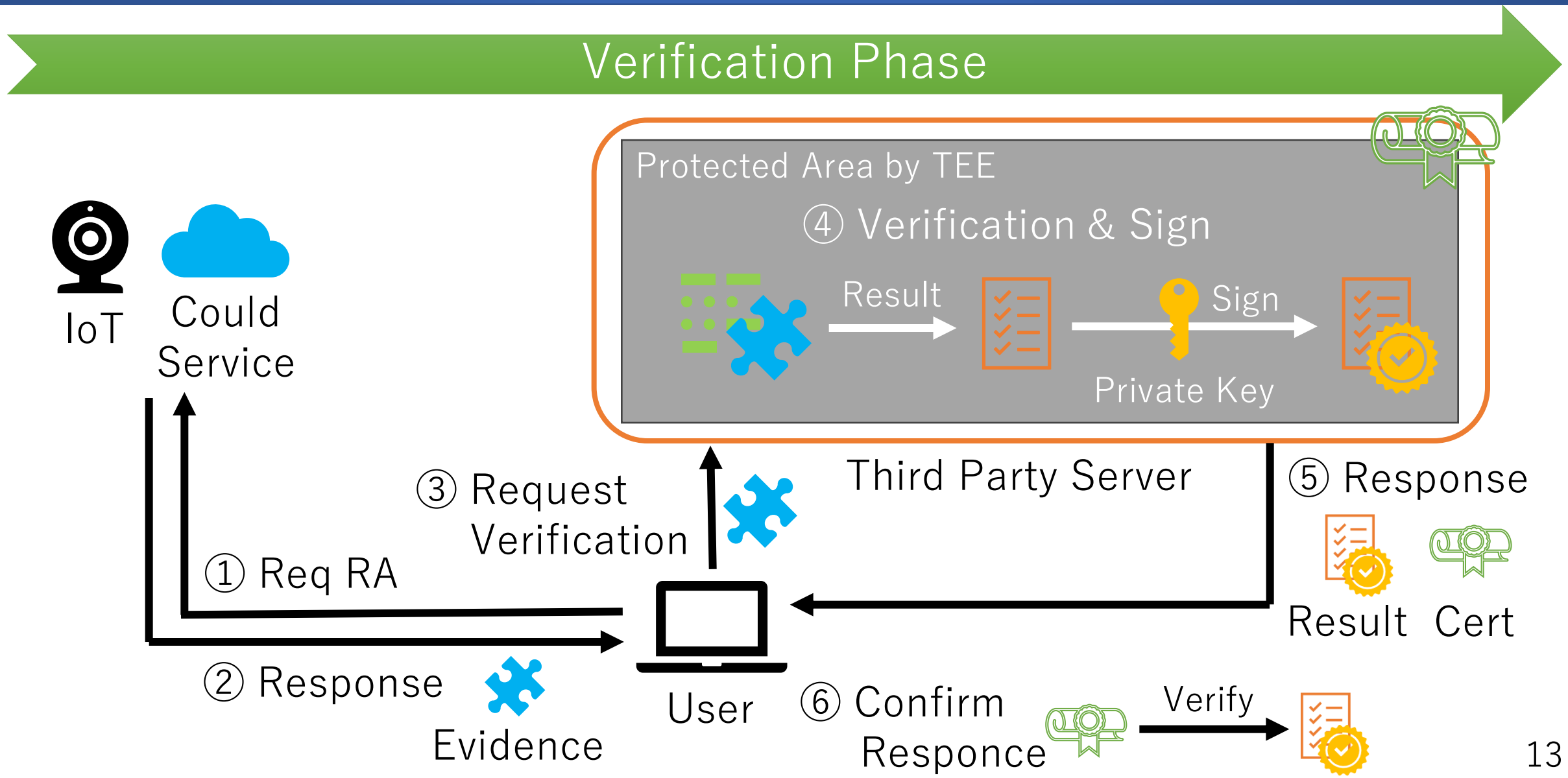
Examination Phase



Delegation Phase



Verification Phase



Contents Table

Implementation

TEE：Intel Software Guard Extensions (SGX)
Verification Program：SGX-TDX-DCAP-QuoteVerificationService*

- Our implementation is a docker container and we used Gramine tool for SGX application.

- We added the process that generating key pair and Certificate Signing Request (CSR) to the Verification Program.

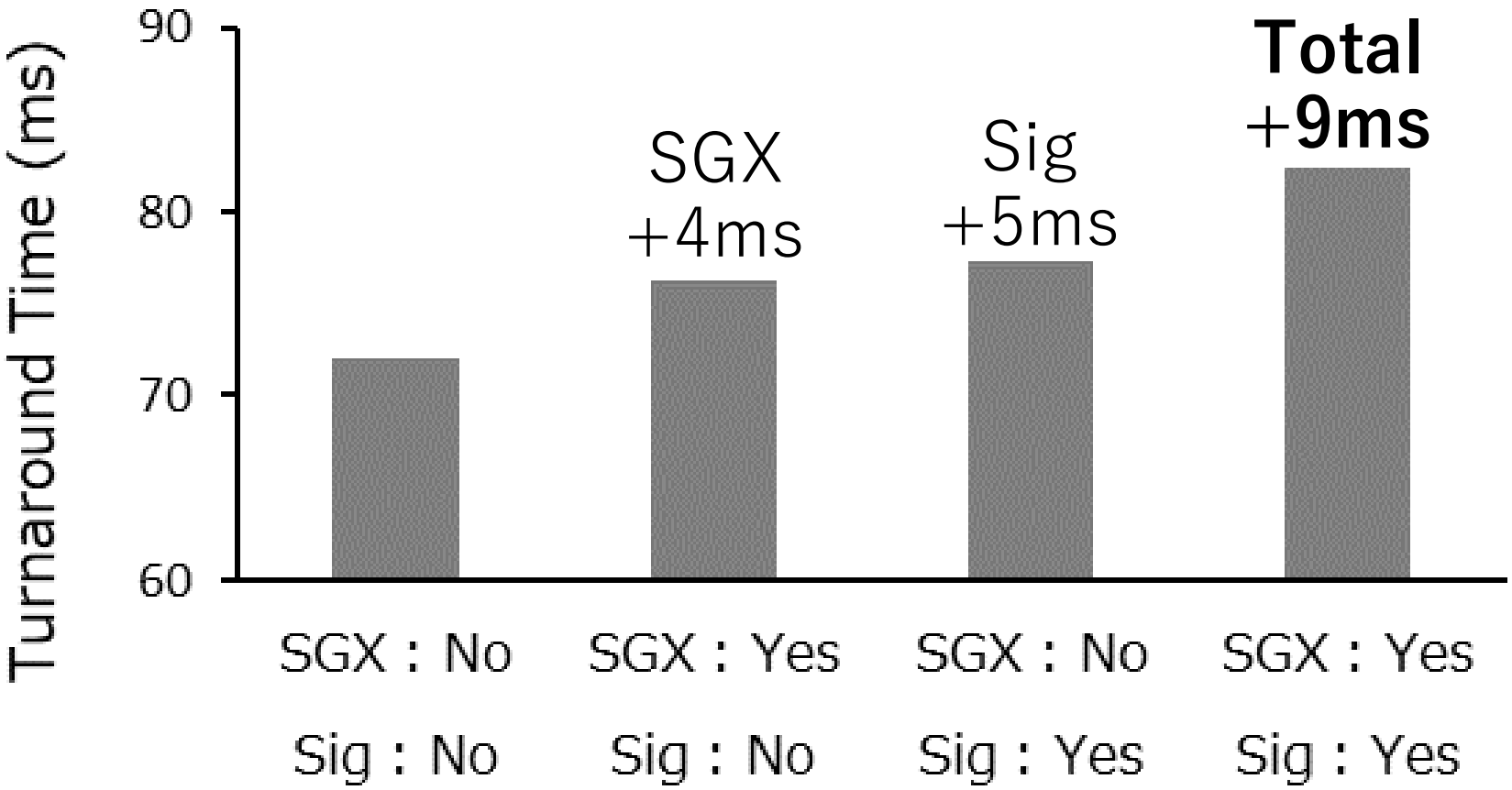
- We used k6 for our evaluation.
  Measurements were taken for 10 seconds and the average value was used.

* Intel, https://github.com/intel/SGX-TDX-DCAP-QuoteVerificationService

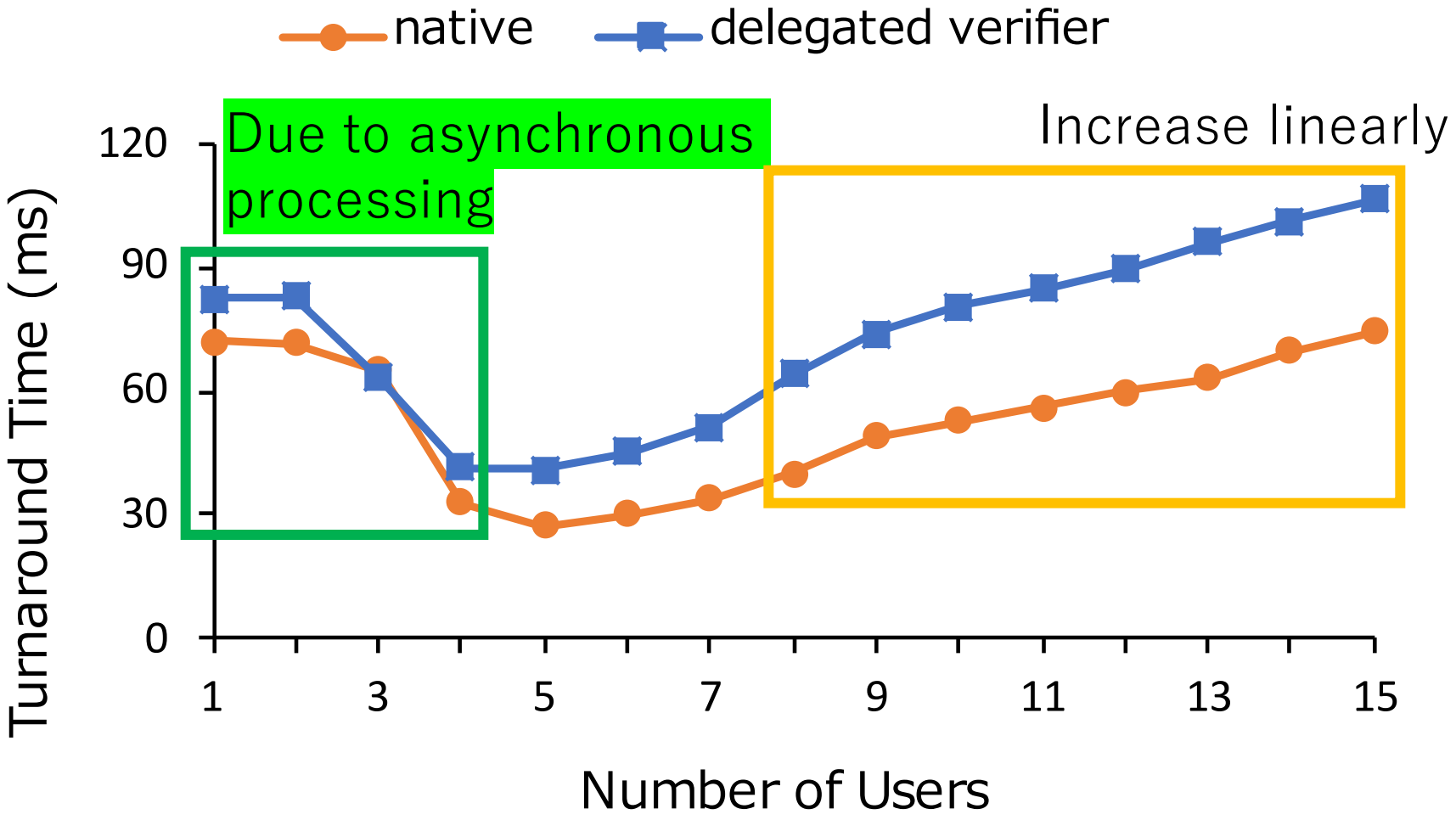
Evaluation Environment

OS	Ubuntu 22.04 LTS
Linux kernel	6.2.0-36-generic
CPU	Intel Xeon Silver 4314
SGX SDK	2.22.100.3
SGX PSW	1.19.100.3-jammy1

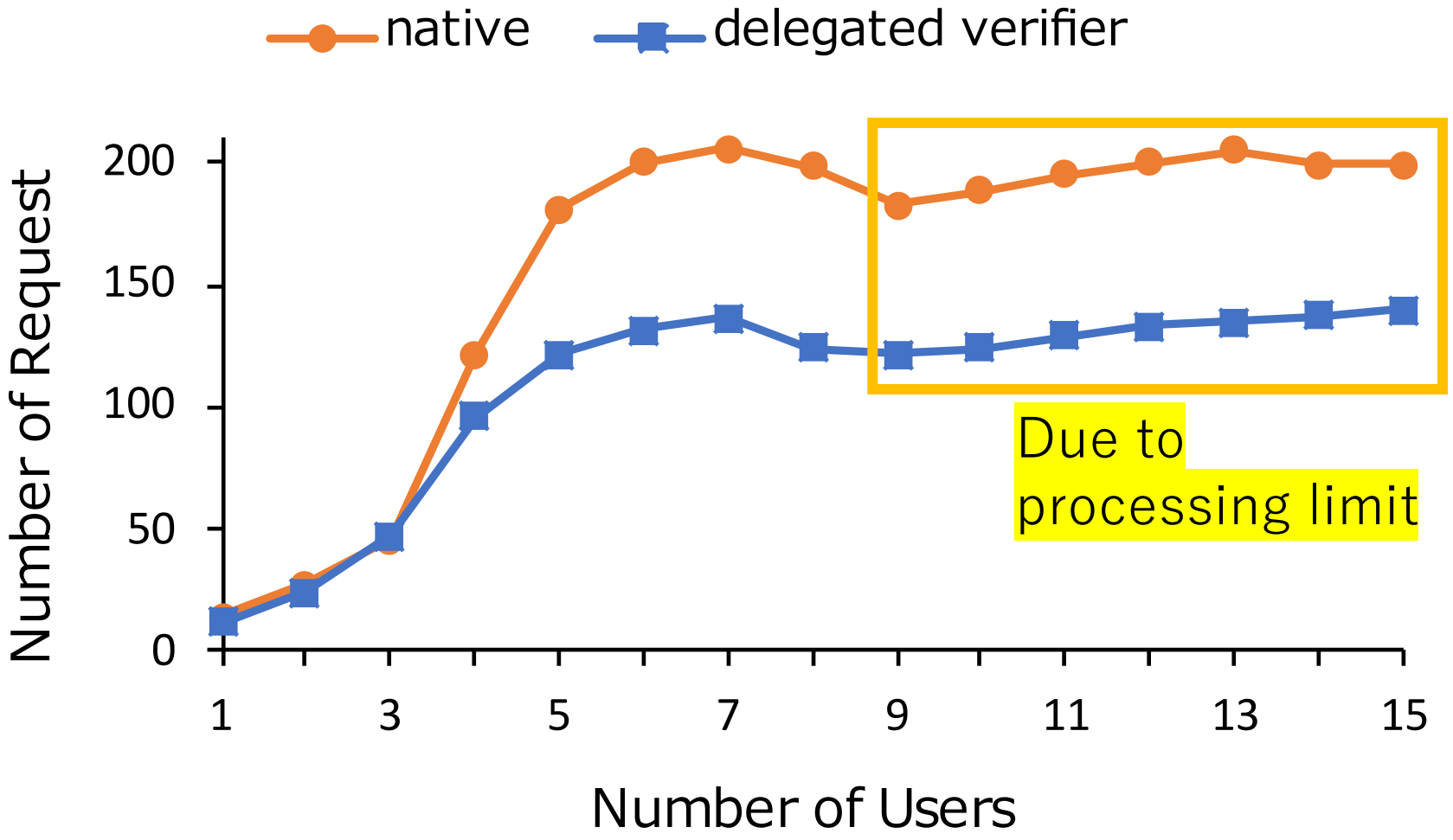
Evaluation：Runtime Overhead



Evaluation：Turnaround Time



Evaluation：Number of Request

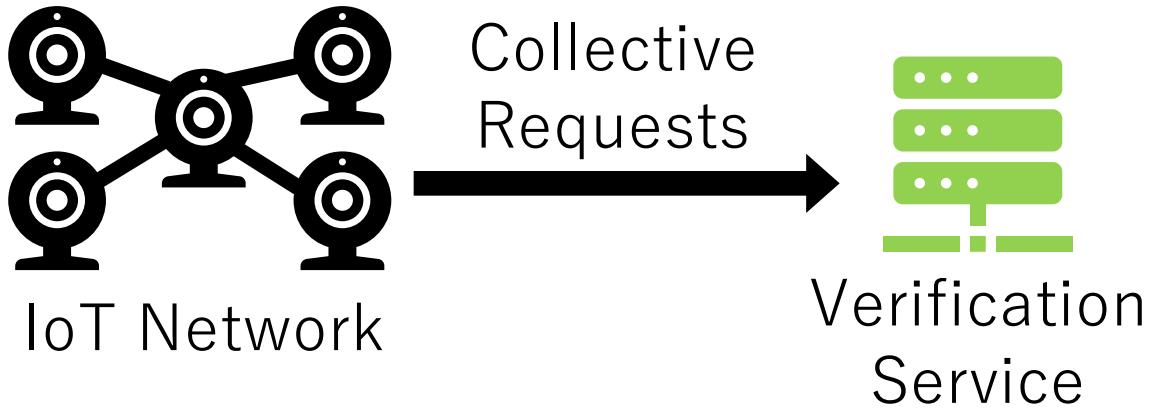


But our proposal can solve the processing limit by scale-out!

Contents Table

Related Works(1/2)

- Proxy Signature (Mambo, et al. 1996) is a cryptographic scheme for erecting a proxy for the signer. The delegate issues a certificate for the signature key of the proxy signer. We apply this scheme on a per-verification-program basis.
- Swarm Attestation (Nadarajah, et al. 2015) improves RA scalability by collectively verifying IoT devices. However, it is not applicable when RA is requested centrally from an unspecified number of devices.



Related Works(2/2)

- Intel Trust Authority* is an online verification service provided by Intel for multiple TEEs. Our proposal will allow verification services to be deployed on third-party servers that are not managed by a trusted authority.
- By using SGX DCAP (Simon, et al. 2018), SGX verification servers can be built by users themselves. However, it is too costly for each SGX user to build a verification environment.

* Intel® Trust Authority, https://www.intel.com/content/www/us/en/security/trust-authority.html

Conclusion

- For increasing in RA, we proposed a delegating verification for secure verification on third-party servers.
- We implemented a proof of concept for our approach with Intel SGX.
- The limit on the number of requests processed suggests that our idea is helpful.

Future Works

- Integration of traditional RA and certificate issuance
  - Modifying Gramine's RA.
- Automation of the delegation phase
  - Accept delegating verification by Trusted Party via API.
- Measuring scalability
  - How quickly can we increase the number of verification servers?
  - What are the challenges in doing so dynamically?

Appendix

Adversarial Model and Assumption

The goal of the adversary:
Users can execute RA on vulnerable or malicious platforms without them being detected.

The adversary can

- manipulate any program with administrator privileges.
- eavesdrop, delete, or tamper with packets in any network.
- erect new unauthorized verification servers.

However, the following attacks are not considered:

- Side-channel attacks
- Attacks that threaten availability (e.g. DoS attack)