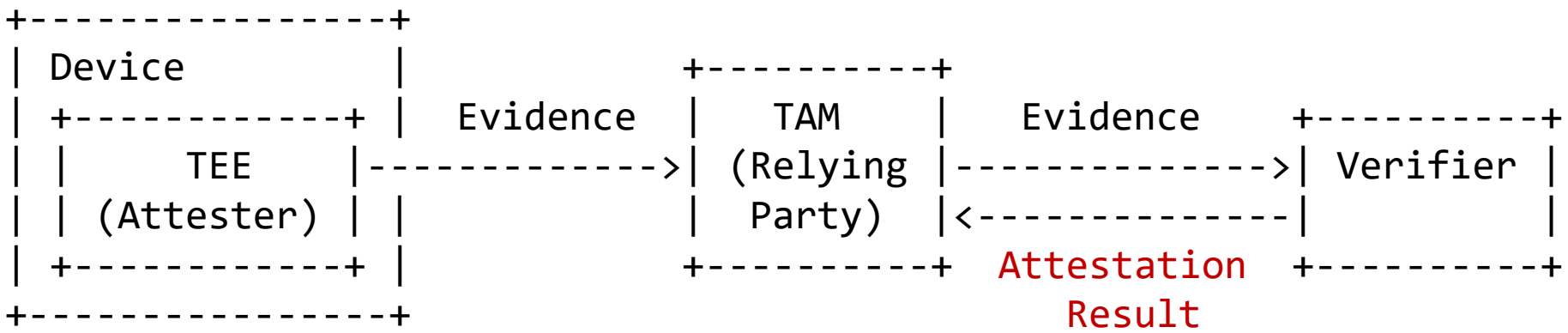
This work was supported by JST K Program Grant Number JPMJKP24U4, Japan.

Hackathon Report: Secure Software Provisioning with VERAISON

Ken Takayama, Shinichi Miyazawa, and Yuma Nishihira

Hackathon Objectives

- To evaluate whether TEEP (TEE Provisioning) Protocol incorporating Remote Attestation is practically viable by using VERAISON as a Verifier

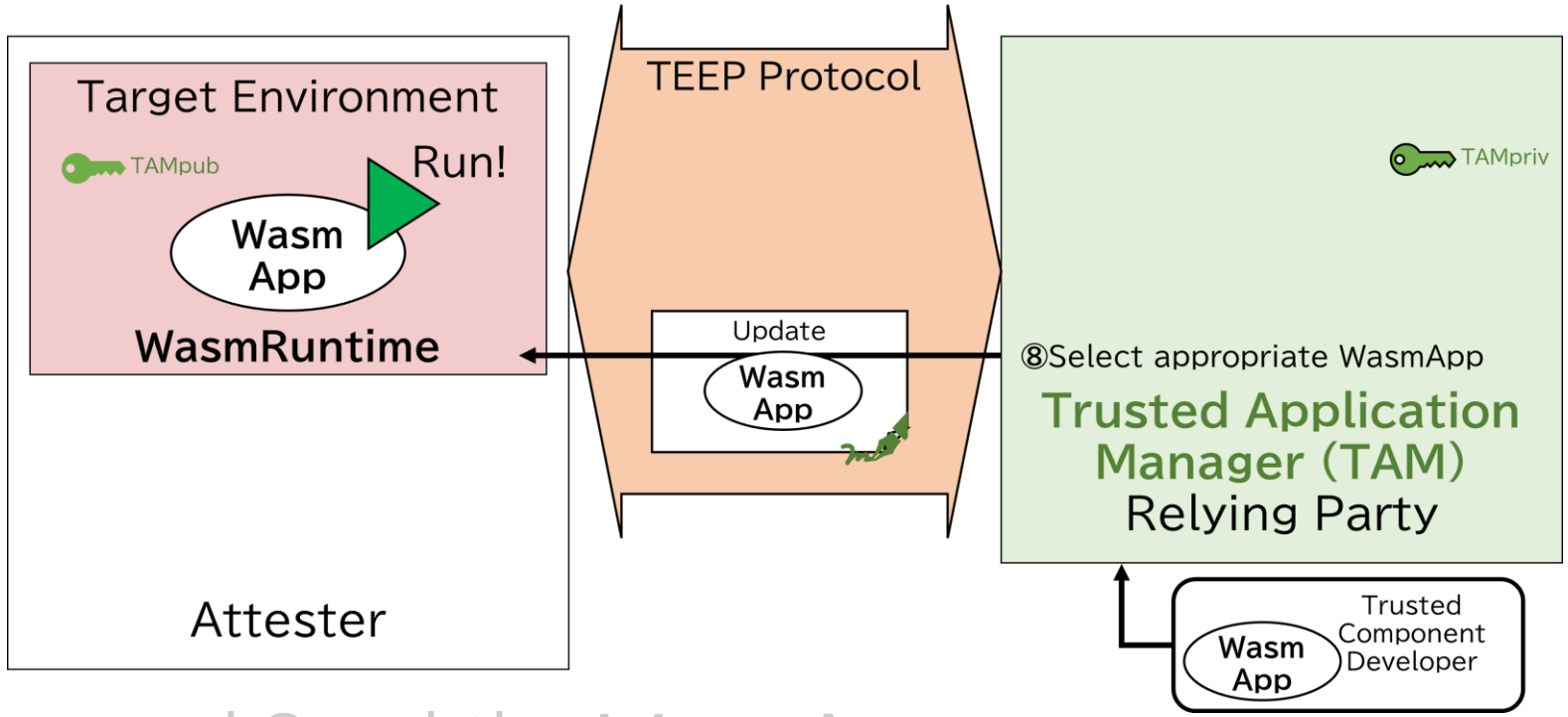


TEEP Attestation Roles (from RFC 9397: TEEP Architecture)

- To share feedback to the RATS documents and VERAISON from the perspective of TEEP utilizing RATS

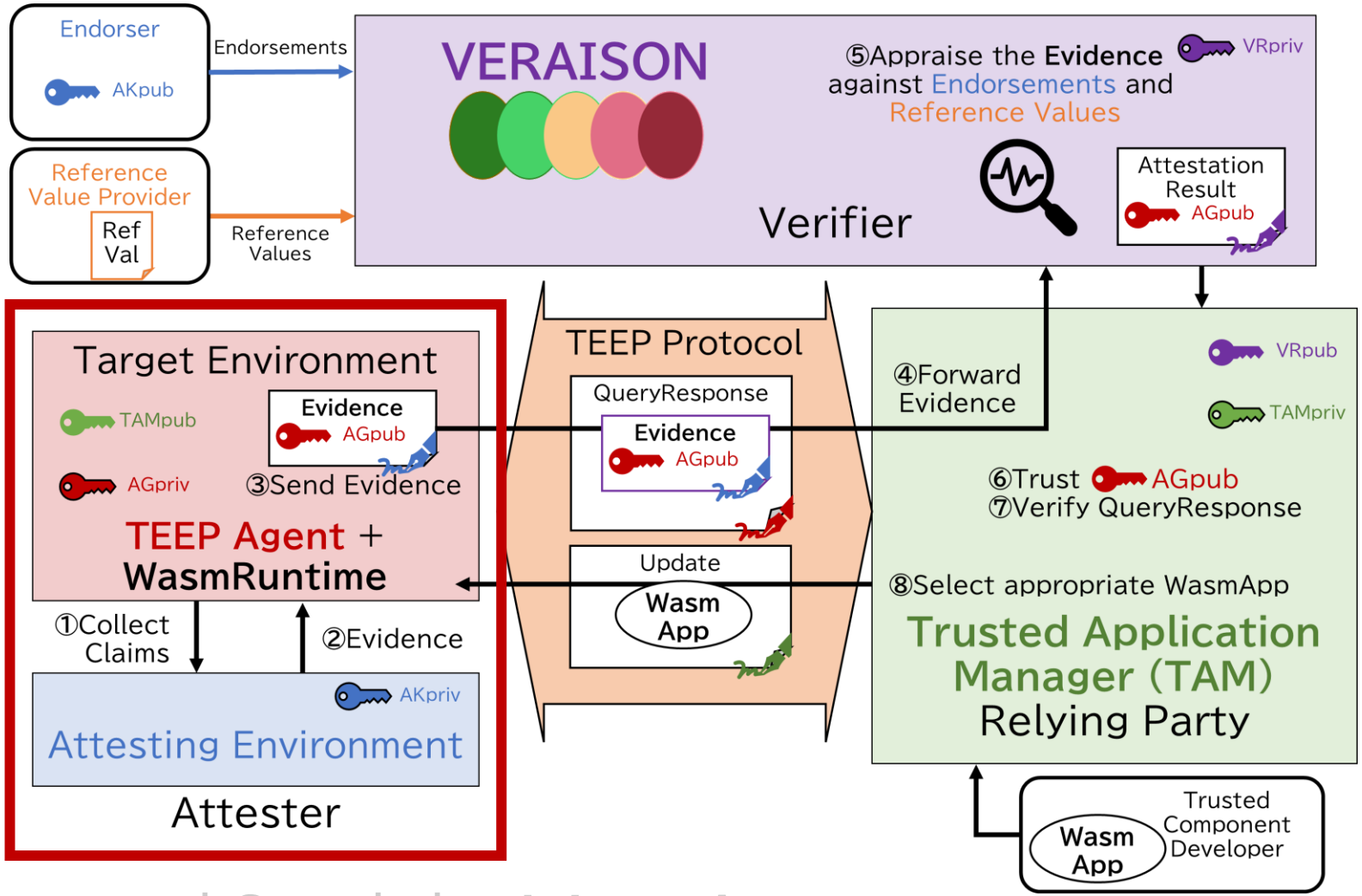
Hackathon Design

- Goal
  - Run a **WasmApp** only on attested **WasmRuntime**
- Steps
  - The Attester creates Evidence of the TEEP Agent and the **WasmRuntime**
  - VERAISON verifies it
  - The TAM trusts TEEP Agent's public key within **cnf** claim in AR
  - Verify the TEEP Message and Send the **WasmApp**



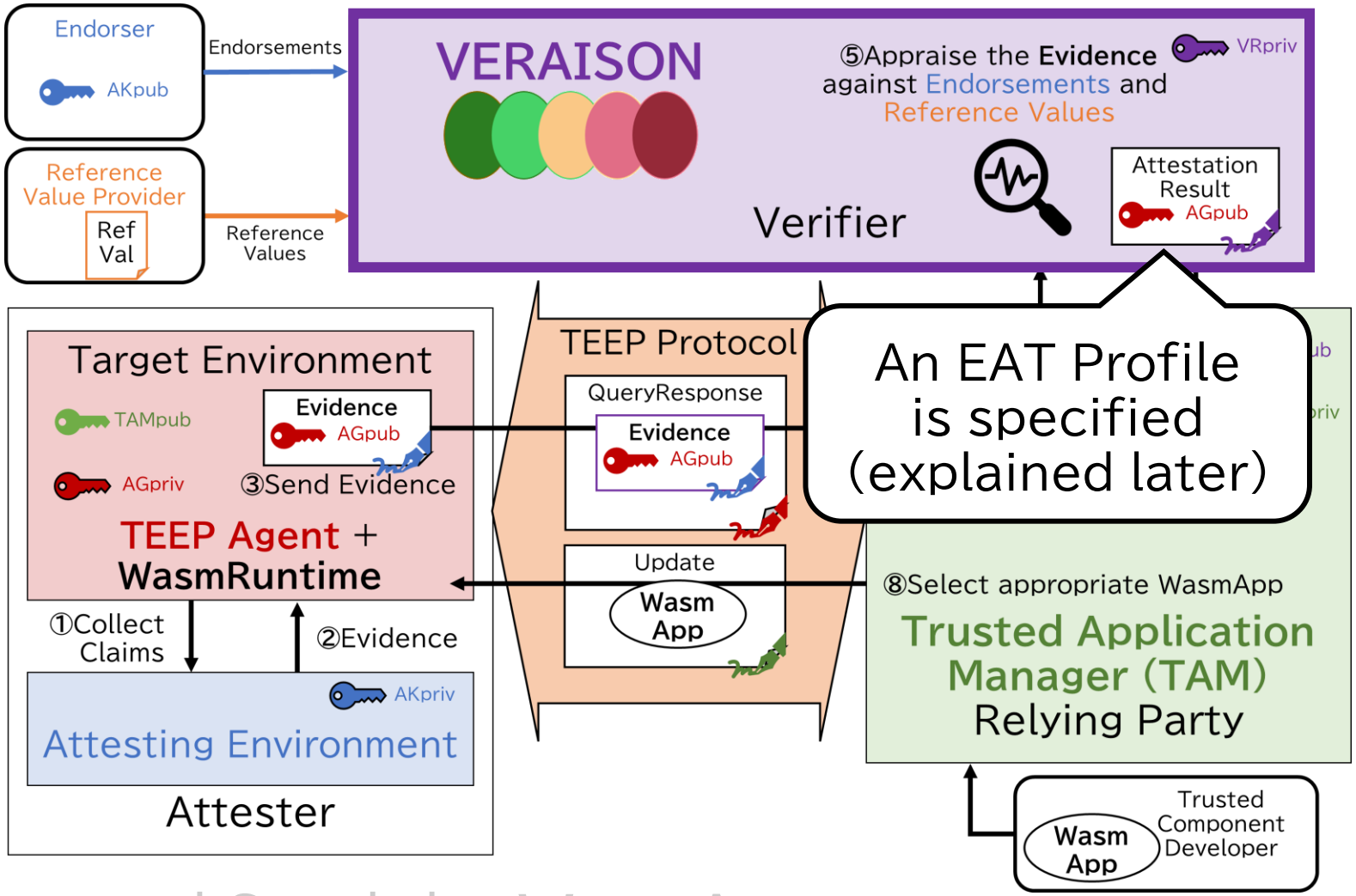
Hackathon Design

- Goal
  - Run a **WasmApp** only on attested **WasmRuntime**
- Steps
  - The Attester creates Evidence of the **TEEP Agent** and the **WasmRuntime**
  - VERAISON verifies it
  - The TAM trusts TEEP Agent's public key within cnf claim in AR
  - Verify the TEEP Message and Send the **WasmApp**



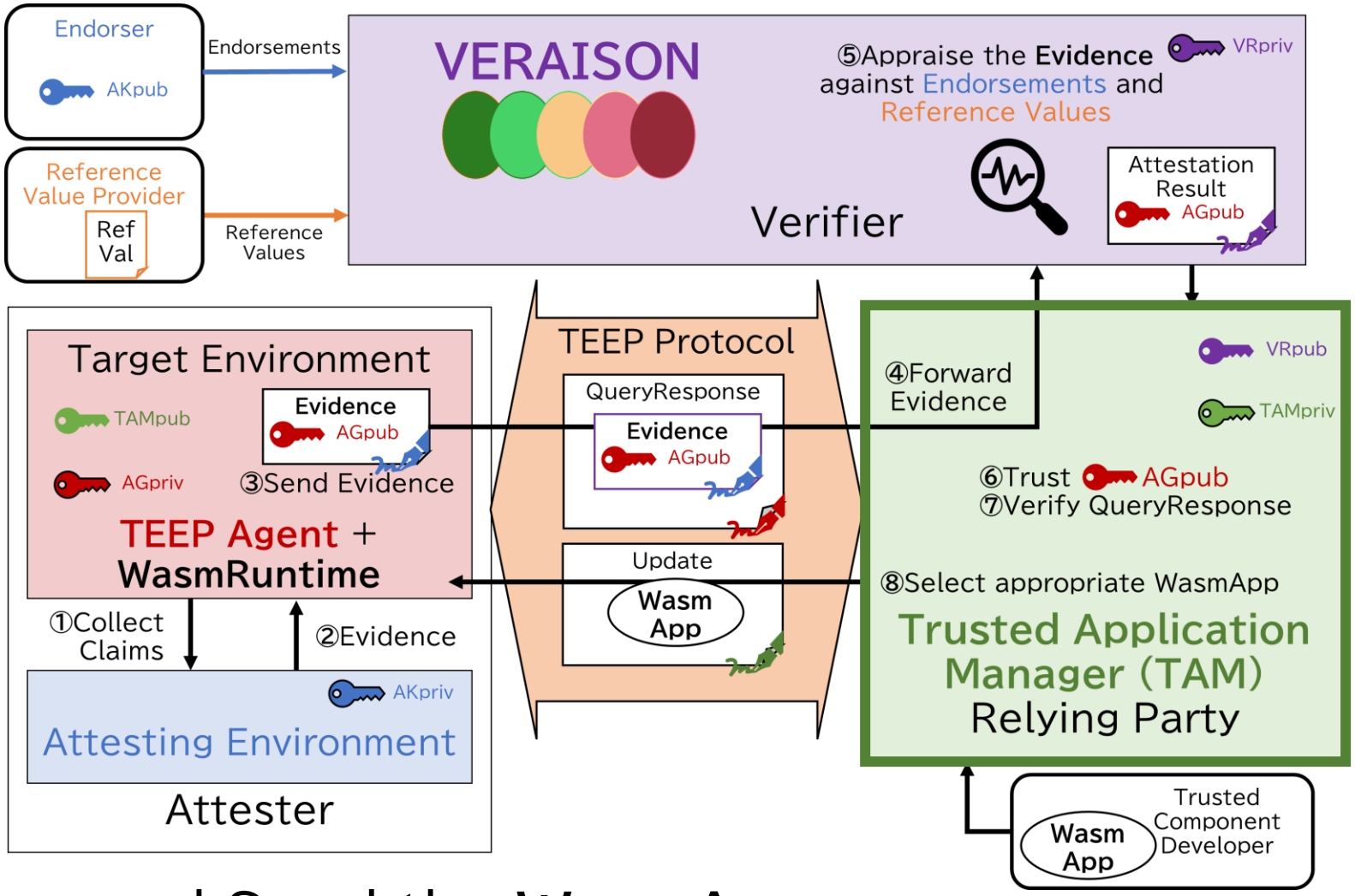
Hackathon Design

- Goal
  - Run a **WasmApp** only on attested **WasmRuntime**
- Steps
  - The Attester creates Evidence of the TEEP Agent and the WasmRuntime
  - VERAISON verifies it
  - The TAM trusts TEEP Agent's public key within cnf claim in AR
  - Verify the TEEP Message and Send the WasmApp



Hackathon Design

- Goal
  - Run a **WasmApp** only on attested **WasmRuntime**
- Steps
  - The Attester creates Evidence of the TEEP Agent and the WasmRuntime
  - VERAISON verifies it
  - The TAM trusts TEEP Agent's public key within cnf claim in AR
  - Verify the TEEP Message and Send the **WasmApp**



EAT Profile for TEEP Protocol

- The Verifier should generate an AR with this profile

Options	Choice
Use of JSON, CBOR, or both	CBOR only
CBOR Preferred Serialization	YES
Required claims	ueid, oemid, hwmodel, hwversion, manifests, and cnf
…	…

What We Faced

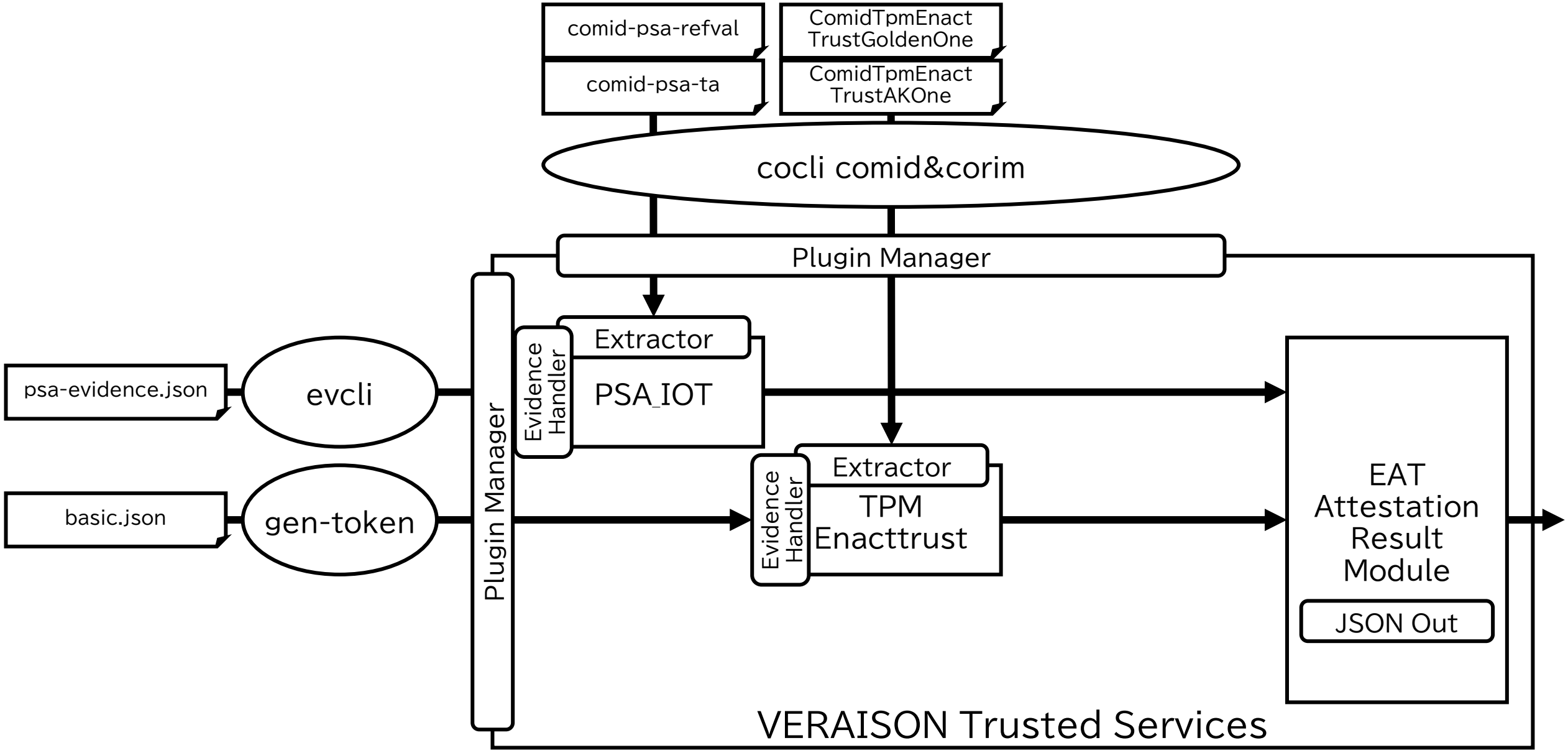
- No existing VERAISON plugin supports all of claims required in the EAT Profile of TEEP Protocol
  - e.g. RFC 9783: PSA Attestation Token are designed not to carry oemid, hwmodel, ..., **cnf** claims (while this is mainly for Evidence)

- Current EAT Attestation Results module doesn't support CBOR output

What We Implemented

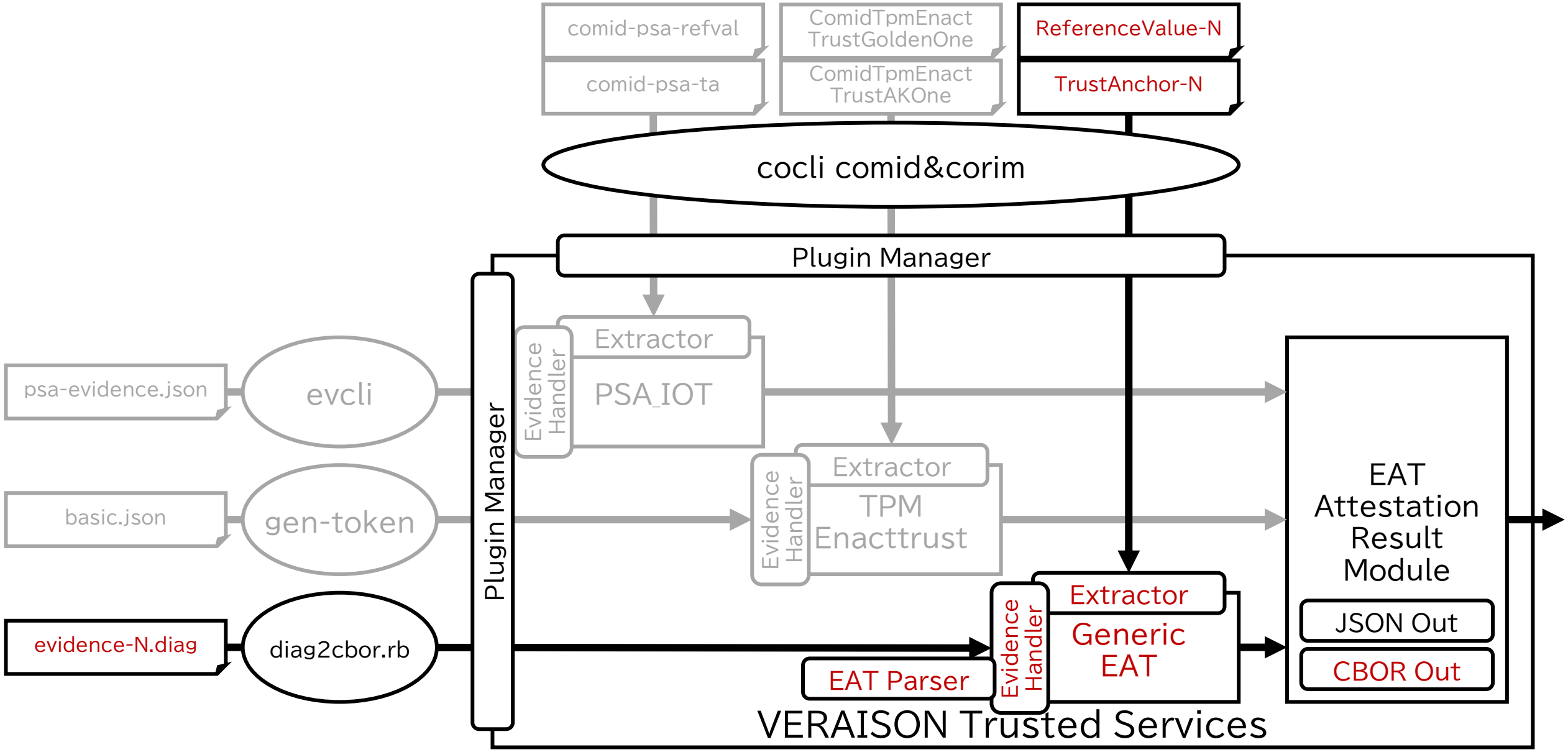
- No existing VERAISON plugin supports all of claims required in the EAT Profile of TEEP Protocol
  - e.g. RFC 9783: PSA Attestation Token are designed not to carry oemid, hwmodel, ..., **cnf** claims (while this is mainly for Evidence)
  - ➢ We've updated the EAT parser reflecting published RFC 9711 EAT
    - Pull Request #36 to github.com/veraison/eat (✅Merged)
  - ➢ We've added *Generic EAT* scheme plugin designed to support any EAT claims
    - on my forked branch github.com/kentakayama/services
- Current EAT Attestation Results module doesn't support **CBOR** output
  - ➢ We've updated the EAR module
    - Pull Request #46 to github.com/veraison/ear (🚧WIP)

To VERAISON Modules



To VERAISON Modules

We've updated/ implemented



Shared Feedback (discussing in ML)

- To RATS EAR document authors
  - TEEP defines an EAT Profile with some required claims, but optional claims are also allowed

```
*  Claims Requirements:

   -  The following claims are required: ueid, oemid, hwmodel,
      hwversion, manifests, and cnf. See Section 4.3.1 for
      discussion. Other claims are optional.
```

EAT Profile for TEEP Protocol

```
ear-teep-claims = non-empty<{
  ? eat.nonce-label => eat.nonce-type
  ? eat.ueid-label => eat.ueid-type
  ? eat.oemid-label => eat.oemid-pen / eat.oemid-ieee / eat.oemid-random
  ? eat.hardware-model-label => eat.hardware-model-type
  ? eat.hardware-version-label => eat.hardware-version-type
  ? eat.manifests-label => eat.manifests-type
}>
```

CDDL Definition of TEEP extension in EAR

  - The extension could be general purpose

- To TEEP Protocol document authors
  - a lot (reflecting documents updates)

Wrap Up

- VERAISON proved to be a dependable backend Verifier for TEEP TAMs
  - with minimum modifications
- *Generic EAT* scheme is applicable not only TEEP Protocol but also other (including future) EAT-based protocols
  - for reference implementations, not intended for production use

- You can try it here!
  - github.com/s-miyazawa/teep-wasm-demo
- Nest steps
  - Use real TEEs, Update an old WasmApp (currently install-only)
  - Propose an interface for customizable EAR output