

Handling multiple verifiers in RATS architecture

[draft-zhang-rats-multiverifiers-00](#)

[Jun Zhang](#), [Houda Labiod](#) , [Tieyan Li](#) ,
[Thanassis Giannetsos](#) , [Henk Birkholz](#)

RATS Interim meeting
27th September 2024

Motivation

- RFC9334 specifies the information flow between 1 Attester, 1 Verifier and 1 Relying Party for RATS.
- Single verifier may face single-point of failure, “achilles’ heel” of the attestation verification system (AVS).
- Augment the RATS architecture to explicitly multiple Verifiers scenarios?
 - Scalability: How many evidences to generate?/How many verifiers to contact?
 - Robustness: How to handle heterogeneous verifiers (delayed update, compromised, different RIM policy, ...)

Goal

Acknowledge and address potential inconsistent behaviors of Verifier by:

- Aggregating Attestation Results collected from multiple Verifiers at the Relying Party
- Simplifying the policy and operation

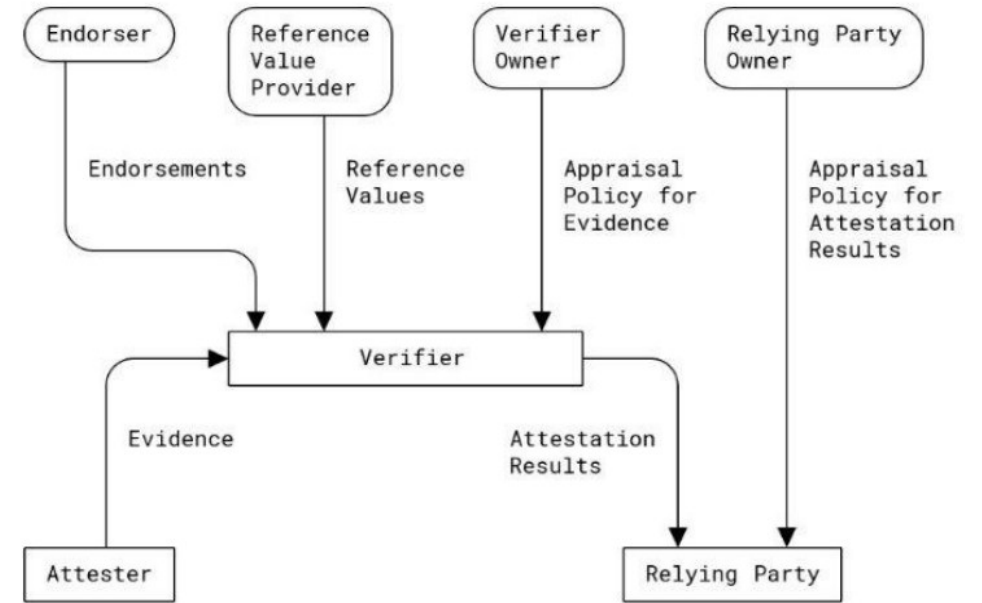


Fig. 1: RATS information flow in RFC9334

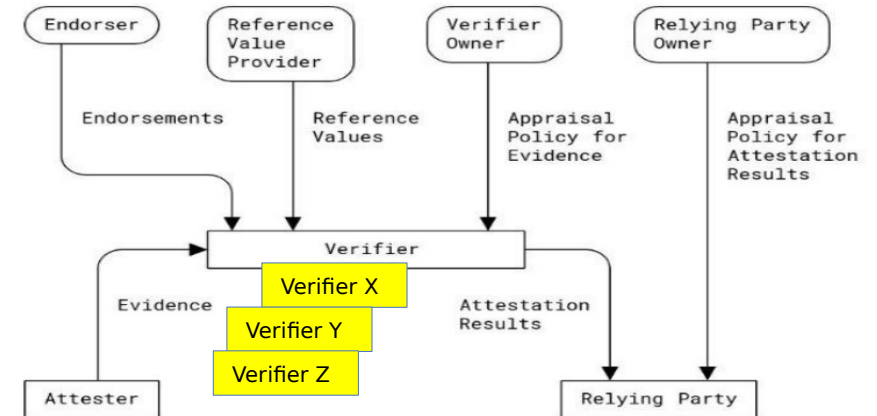
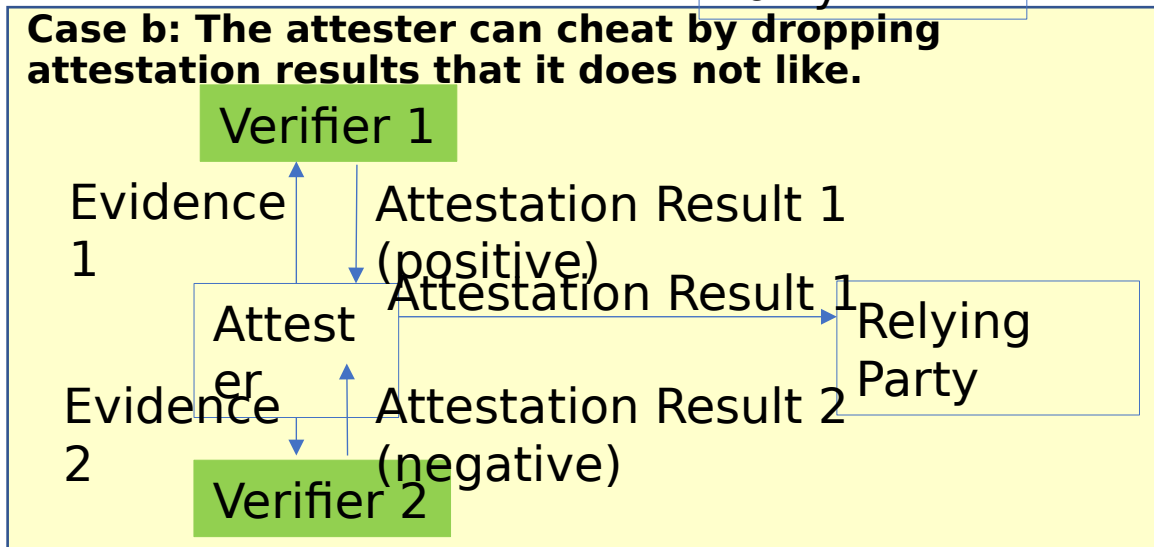
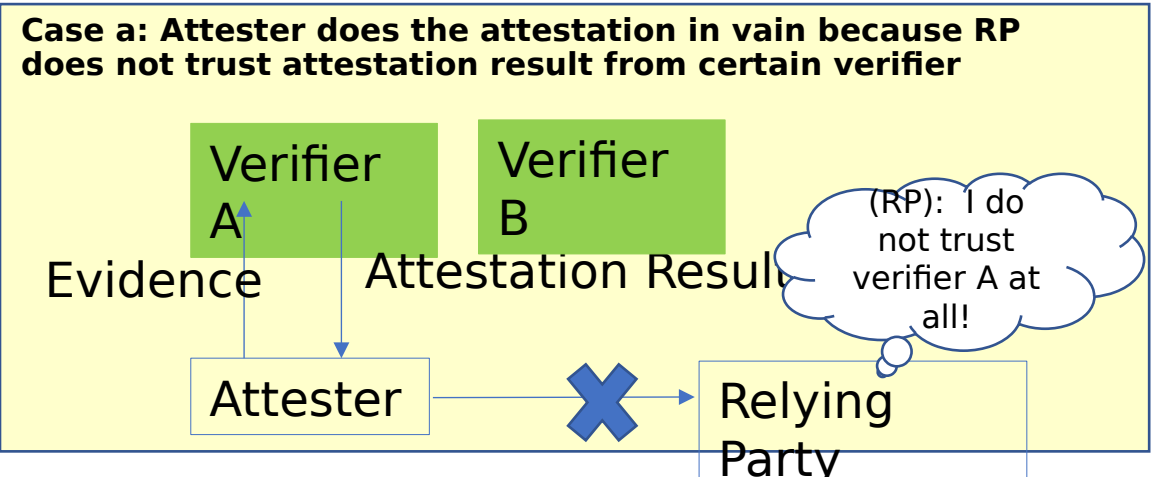


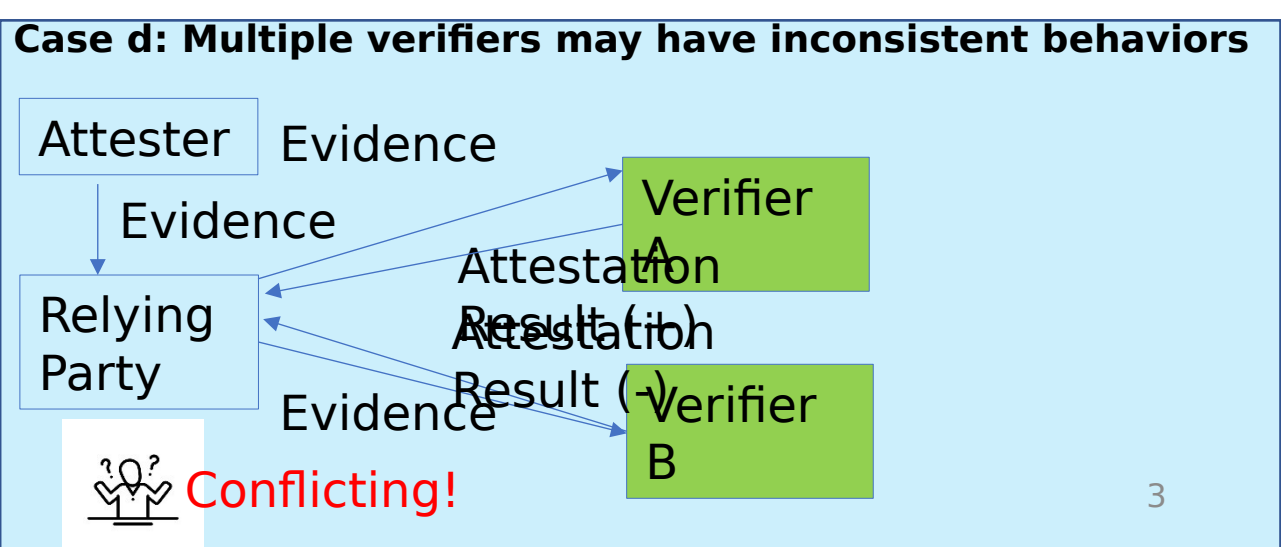
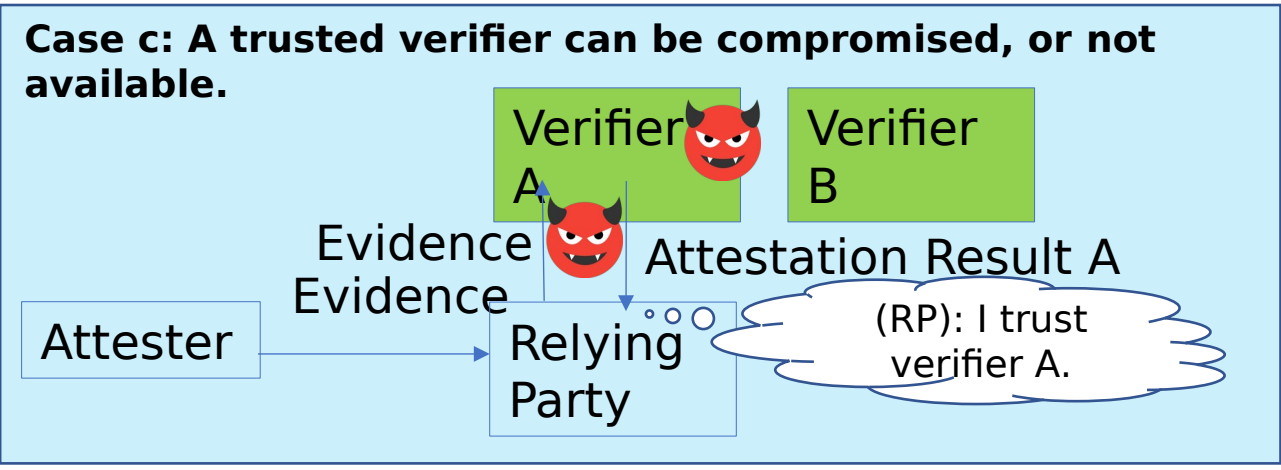
Fig. 2: RATS architecture with multiple Verifiers

Possible issues to handle multiple verifiers in current RATS architecture

Passport model



Background-check model



Handling multiple verifiers

Extension of background-check model

Objective: introduce multiple Verifiers to ensure the resilience of the attestation service

Extension of the background-check model

- Mostly on relying party side
 - RP initiates the attestation flow and generates the nonce.
 - RP forwards evidence to all its recommended Verifiers.
 - RP aggregates the attestation results from these Verifiers.
- Benefit
 - Ensure attestation is not done in vain.
 - Ensure availability & security of remote attestation by “**Wisdom of Crowds**”
 - Avoid redundant generation of evidence
 - Shortlisted Verifiers (detailed in **Verifier Manager** part)

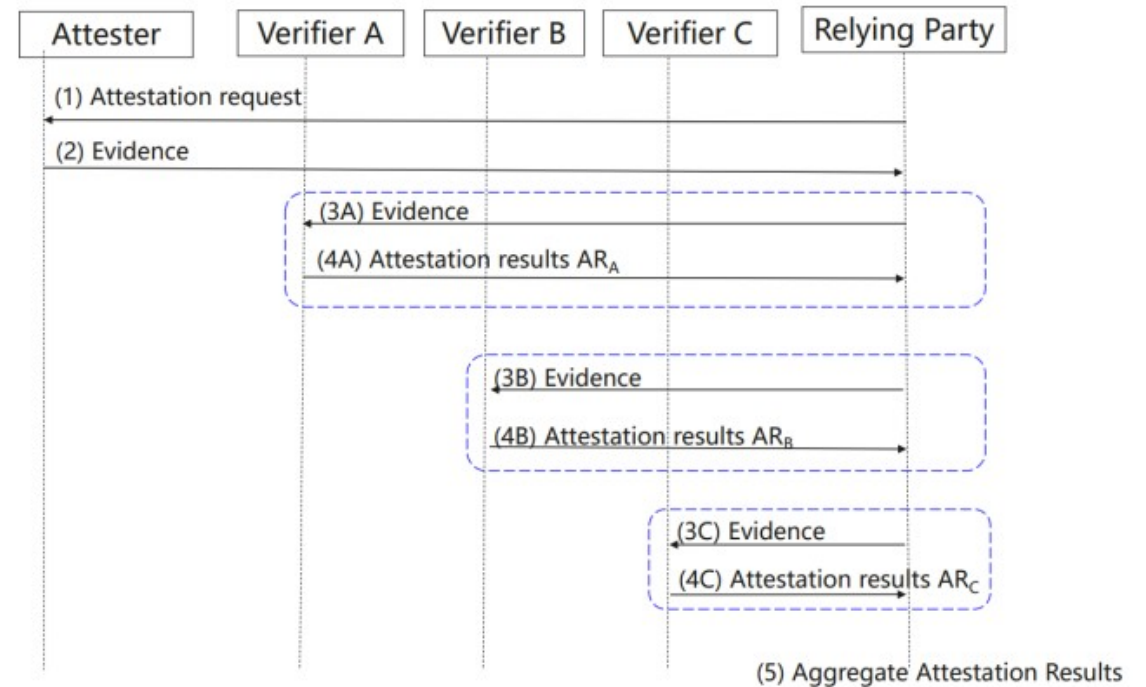
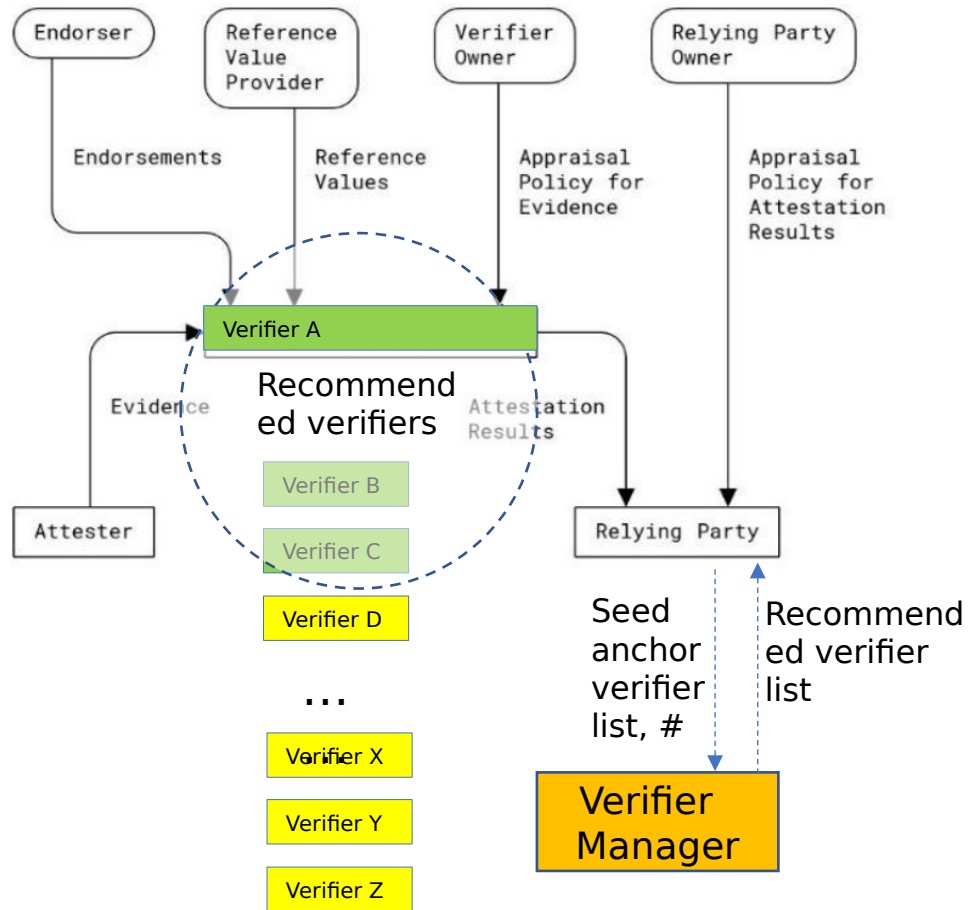


Fig. 3 Augmented Background-Check Model to support the aggregation of Attestation Results from multiple Verifiers

Verifier manager

Goal: support for flexible verifiers configuration at relying party



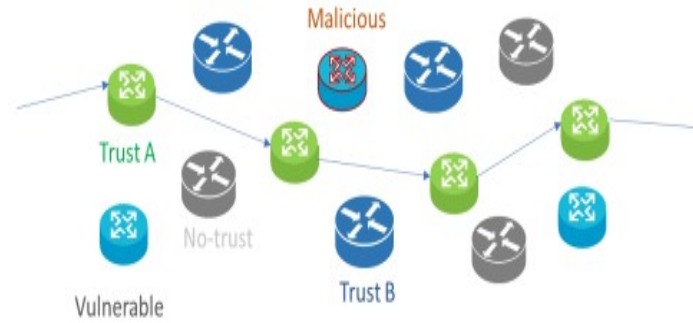
Introduce verifier manager in the architecture to:

- Provide a recommended Verifier list to RP based on input of a seed anchor verifier list and required # of Verifiers
- Enable RP contacts small set of Verifiers that follow same Reference Values
- Enable RP flexible configuration of policy

Fig. 4 Interaction between a verifier manager and a

Use case 1: Node Attestation for Trusted Routing

Trusted Routing requires traffic to go through trusted nodes while they can be appraised by the Verifier (Fig. 5).



Challenge:

Single Verifier may not be available or may be compromised

Fig. 5 Trusted routing Use Case from [NASR use cases](#)

Solution:

Provide multiple Verifiers (primary and secondaries) to ensure the availability of the attestation verification service (AVS) for nodes in the network (Fig. 6)

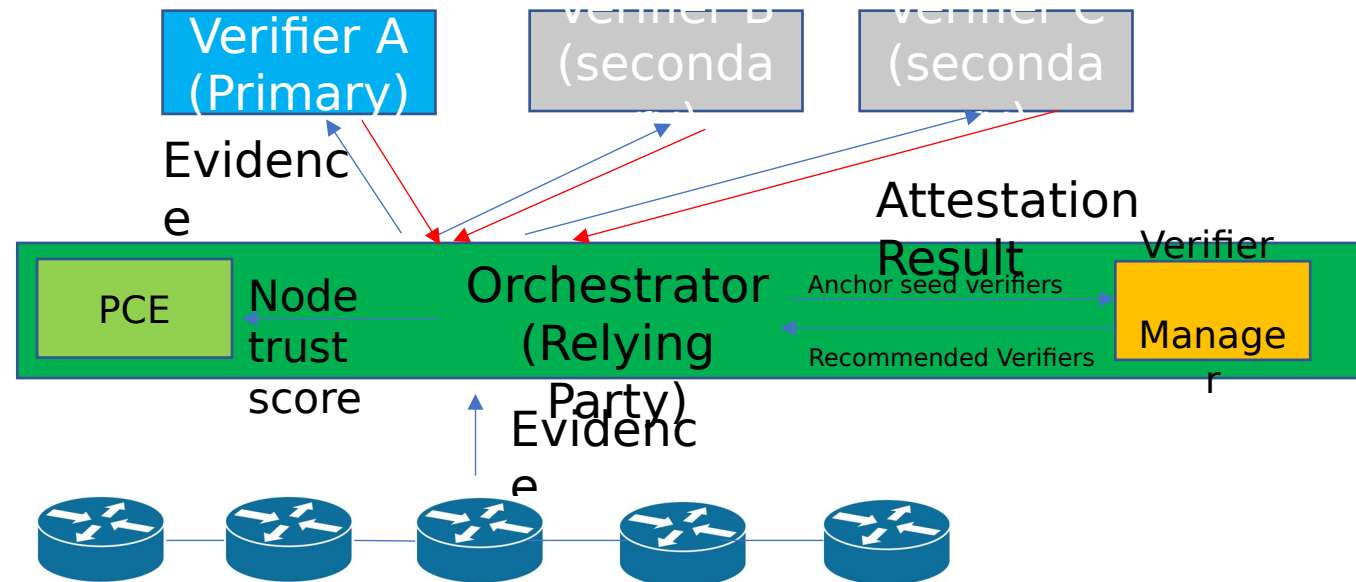


Fig. 6 Node attestation for trusted routing with multiple Verifiers

Use case 2: Attestation in Data Centers

In Data Center (Fig. 7), units (CPU, DPU, GPU) appraise each other to guarantee the trustworthiness of the workload inside the units. Each unit (as attester) can be appraised by many Verifiers (Fig. 8). This results in large amount of request of evidence from the Attester.

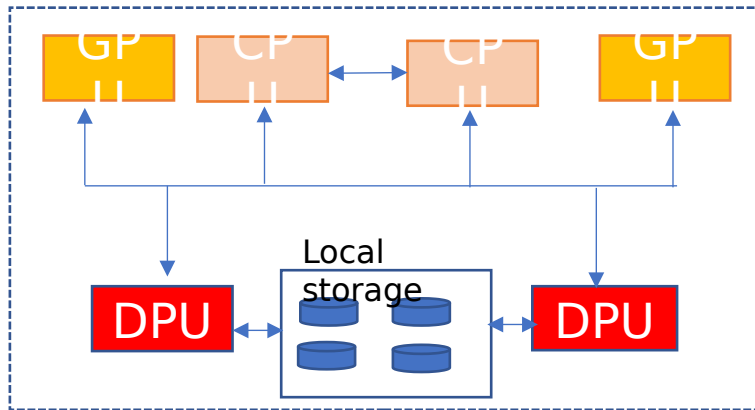


Fig. 7 Interaction between units in Data Center

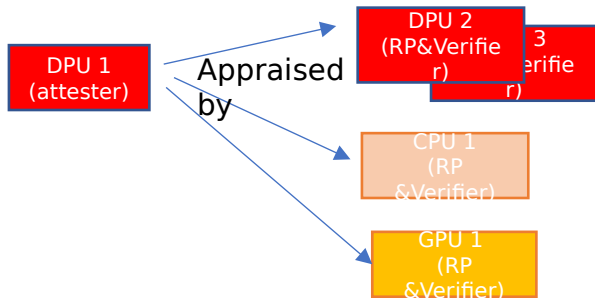


Fig. 8 Attestation between units in Data Center

Two challenges to address

- **Handle the case when some Verifiers do not work** (not available, compromised)
- **Reduce the number of evidence to generate**

Following our proposal to handle multiple verifiers (Fig. 9),,

- Resilience to the dysfunction of certain verifiers
- One evidence is sufficient for verification requirement from n units

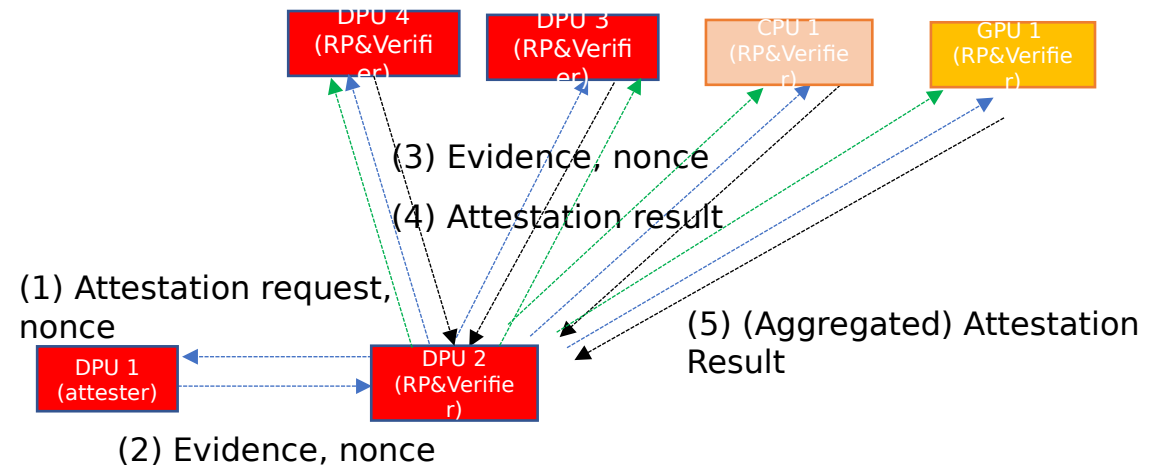


Fig. 9 Handling multiple Verifiers for the attestation in Data Center

Thank you