Handling multiple verifiers in RATS architecture

draft-zhang-rats-multiverifiers-01

Jun Zhang, Houda Labiod , Tieyan Li , Thanassis Giannetsos , Henk Birkholz

IETF 121 RATS

5th November 2024

Progress from 00 to 01

- Moved to kramdown, github@ ietf-rats/draft-zhang-rats-multiverifiers
- In section 3
  - add examples that RP needs multiple Verifiers: 1) multiple (homogeneous) Verifiers for resilience, 2) multiple (heterogeneous) Verifiers for the aggregation of Attestation Results.
  - add reasoning that multiple Verifiers cannot be replaced by the current definition of a single conceptual Verifier (e.g., resilience)
  - Add more detail of the function of the Verifier Manager and how to handle misbehaved Verifiers in Section 3.2
- In section 4, add more detail to the two use cases.

RATS Virtual Interim Meeting – Friday Sep 27th Q&A (1/2)

- **Q1**: Are Verifiers same? Are the same Evidences sent to all selected Verifiers? Is there an interoperability issue? How do we reason about the results? (Hannes Tschofenig, Ghada Afraoui)
- **A:**
  - Verifiers are not necessary same. They can be 1) Homogenous Verifiers, 2) Heterogeneous Verifiers that Attestation Results from them are aggregable based on RP's aggregation policy.
  - The Evidences that are sent to all selected Verifiers via RP are the same as the one from the Attester. The RP dose not change the Evidence's format and content.
  - How to handle Attestation Results depends on RP's local policy. RP can make decision according to received Attestation Results, or generated an aggregated Attestation Result. Consensus decisions are possible ways, but we leave the decision policy open in the draft.

- **Q2**: Is a single "logical" Verifier enough to cover use case with multiple Verifiers? Is the concept of multiple Verifier more implementation specific rather being architectural (Hannes Tschofenig, Nancy)
- **A:**
  - In terms of resilience, conceptual multiple Verifiers cannot be replaced by a conceptual single Verifier as this single Verifier may still has the availability issue.
  - In terms of interoperability, current RATS architecture does not describe the information flows to guide "connecting RPs to appropriate groups of Verifiers", while the introduction of Verifier Manager, and its interaction with the RP can fulfill this duty.
  - In terms of time of freshness, multiple Verifier architecture provides a way to align the nonce/Epoch Maker for Evidence sent to different Verifiers

RATS Virtual Interim Meeting – Friday Sep 27th Q&A (2/2)

- **Q3**: How a misbehaved Verifier is adjudicate? (Carl Wallace)
- **A:** When a Relying Party receives certain outlining Attestation Results from certain Verifiers, RP can inform the Verifier Manager about this incidence, and the Verifier Manager will deal with this incidence, and reduce the probability to recommend these Verifiers to Relying Parties Later. So in the long run, the outlining Verifiers will be managed.

- **Q4:** "Simplifying policy"? Wouldn't each verifier still need a separate appraisal policy? One VM will be able to disperse Appraisal policies to the verifiers pool. This does not simplify appraisal policy. (Giridhar Mandyam)
- **A:** RP configures its desired policy for appraisal for Attestation Results by configuring seed anchor verifier list, so as to obtain desired recommended Verifiers from the interaction with the Verifier Manager.

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

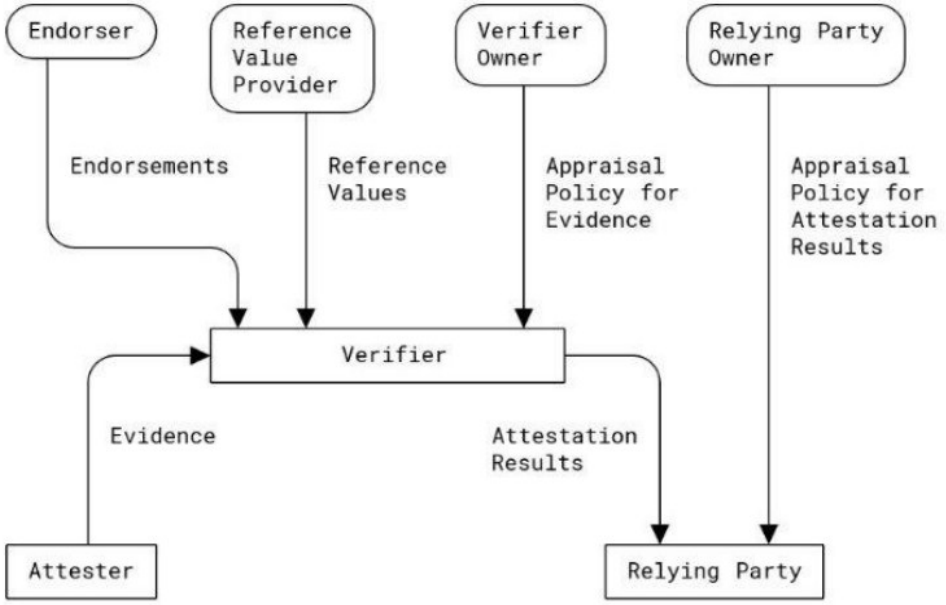


Fig. 1: RATS information flow in RFC9334

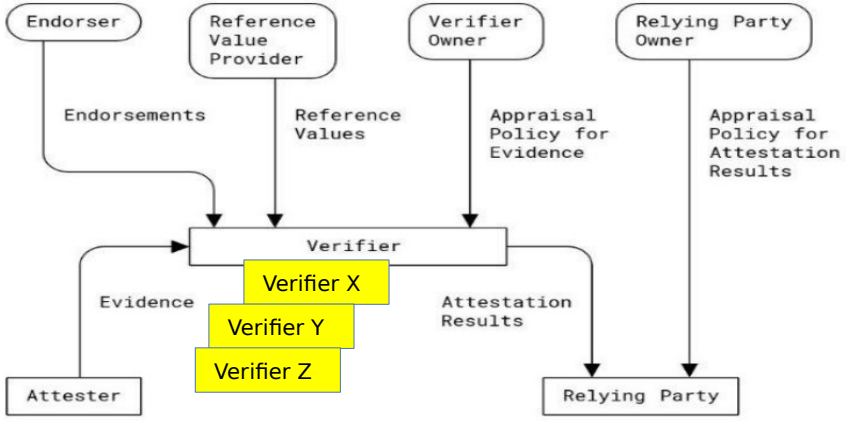
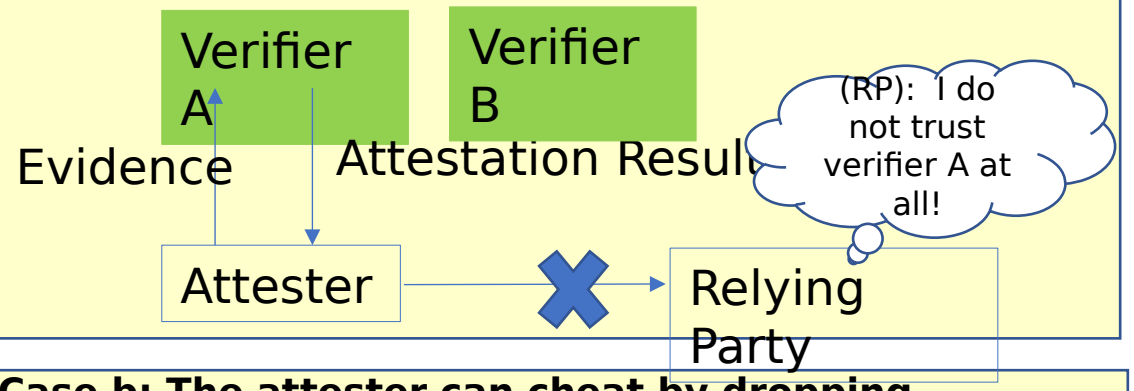


Fig. 2: RATS architecture with multiple Verifiers

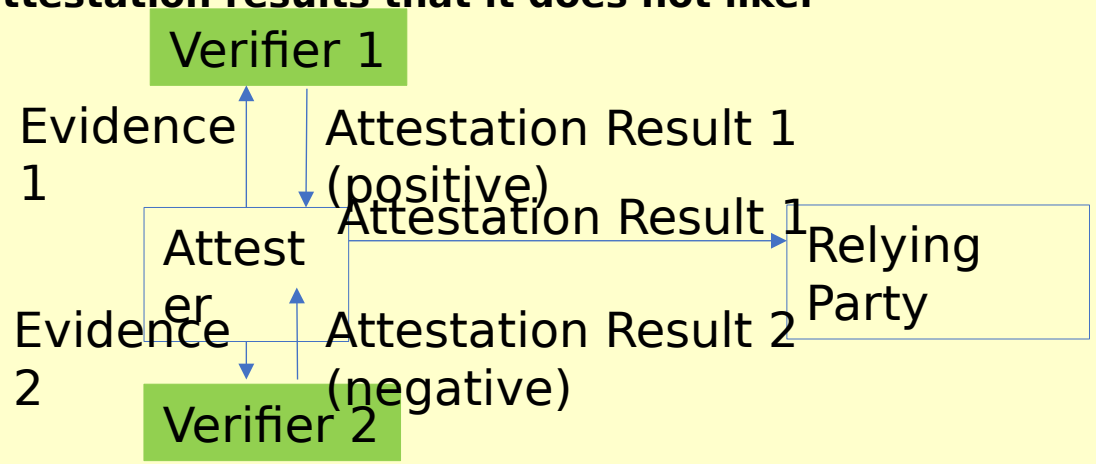
Possible issues to handle multiple verifiers in current RATS architecture

Passport model

Case a: Attester does the attestation in vain because RP does not trust attestation result from certain verifier

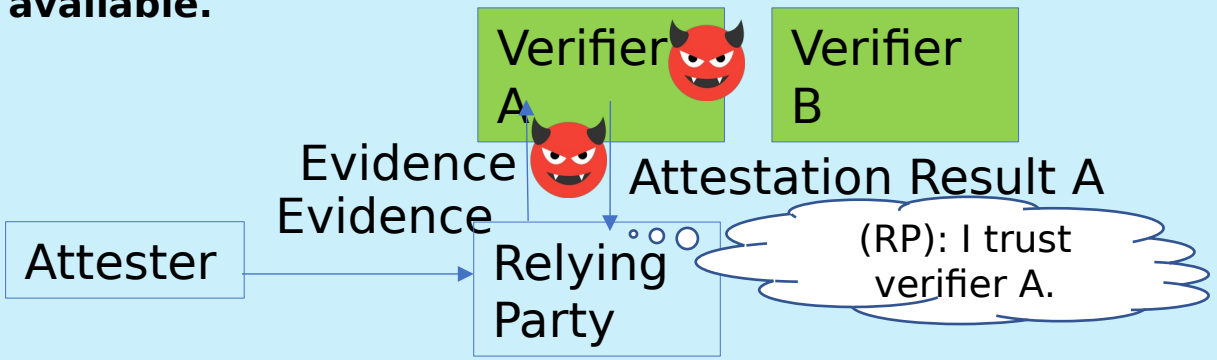


Case b: The attester can cheat by dropping attestation results that it does not like.



Background-check model

Case c: A trusted verifier can be compromised, or not available.



Case d: Multiple verifiers may have inconsistent behaviors

Handling multiple verifiers

Extension of background-check model

Objective: introduce multiple Verifiers to ensure the resilience of the attestation service, and the support for Attestation Result aggregation based on the same Evidence.

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”** & support aggregation of Attestation Results from heterogeneous Verifiers
  - Avoid redundant generation of evidence
  - Shortlisted Verifiers (detailed in **Verifier Manager** part)

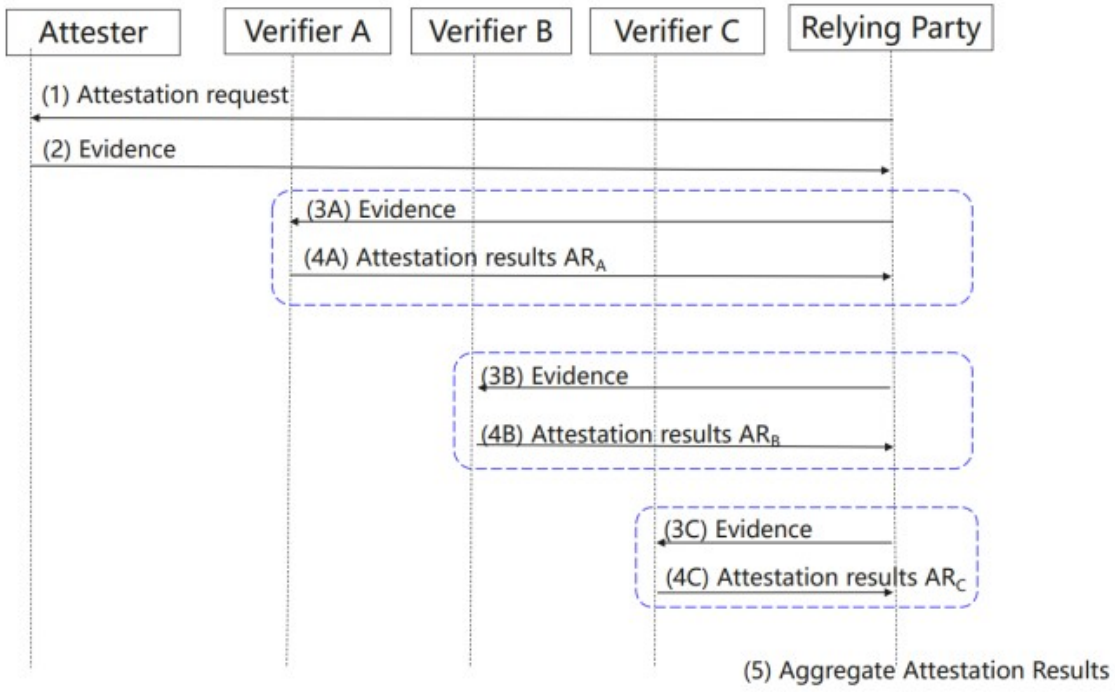


Fig. 3 Augmented Background-Check Model to support the aggregation of Attestation Results from multiple Verifiers

Verifier manager

e: support for flexible verifiers configuration at relying party

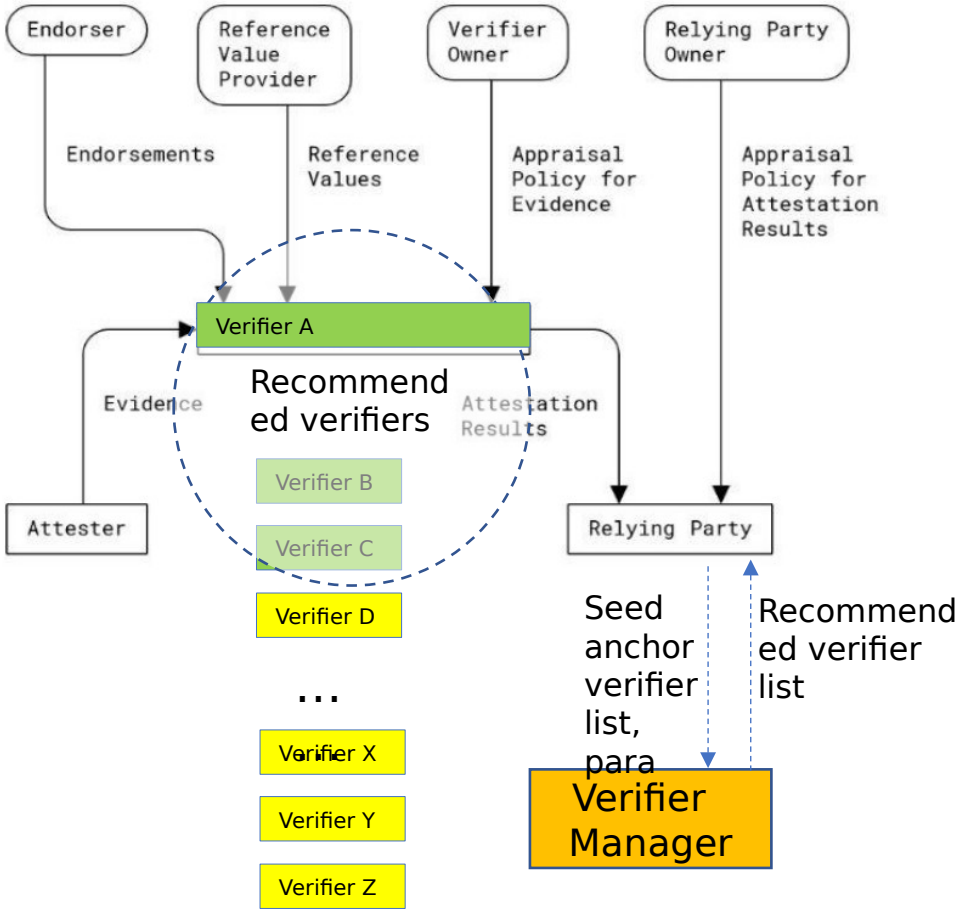


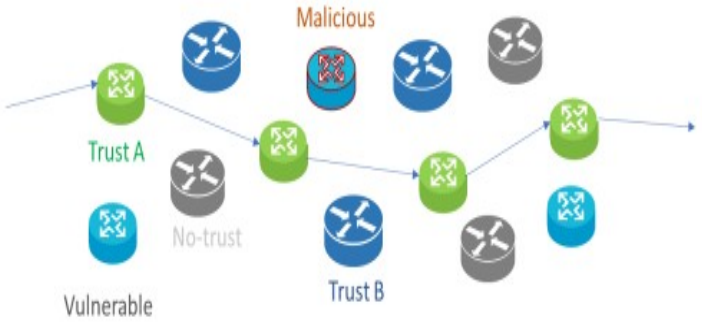
Fig. 4 Interaction between a verifier manager and a

Introduce verifier manager in the architecture to:

- Provide a recommended Verifier list to RP based on input of a seed anchor verifier list and parameters
- Enable RP to contact a list of selected Verifiers
- Enable RP to configure its desired verification policy by configuring seed anchor verifier list

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)

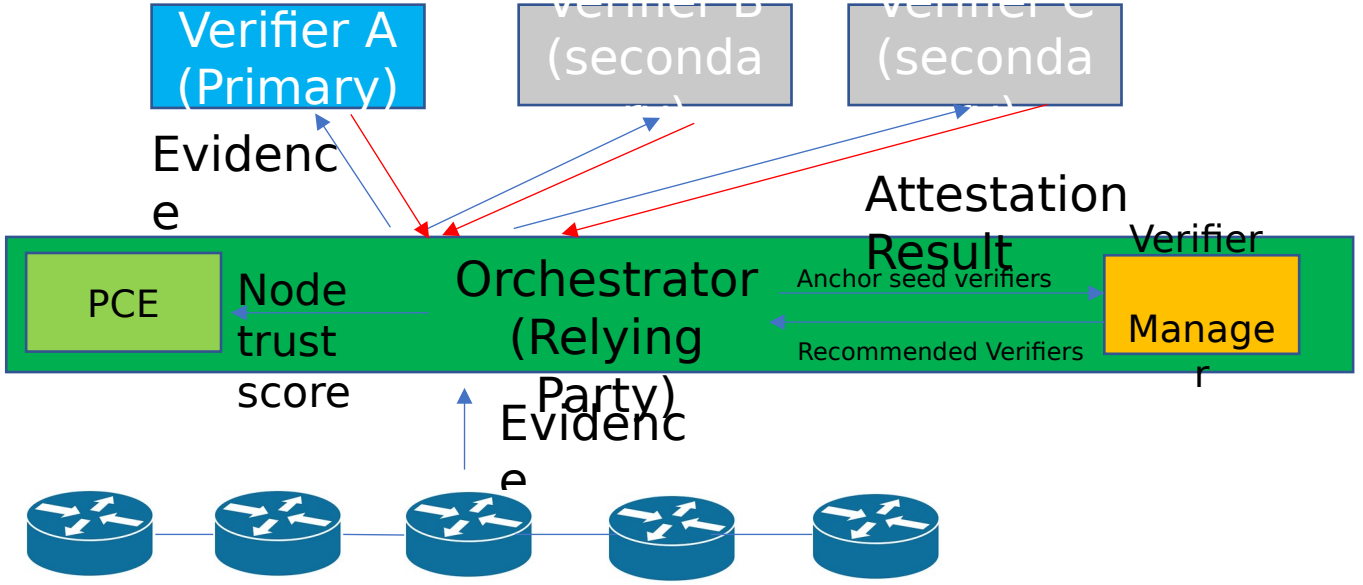


Fig. 6 Node attestation for trusted routing with

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.

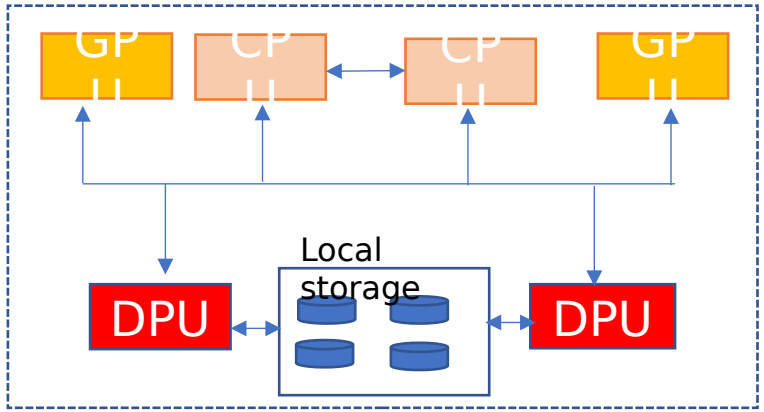


Fig. 7 Interaction between units in Dater Center

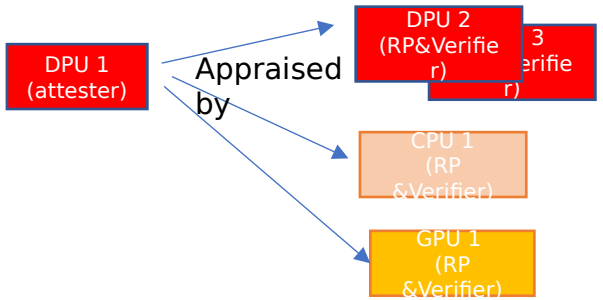


Fig. 8 Attestation between units in Dater Center

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

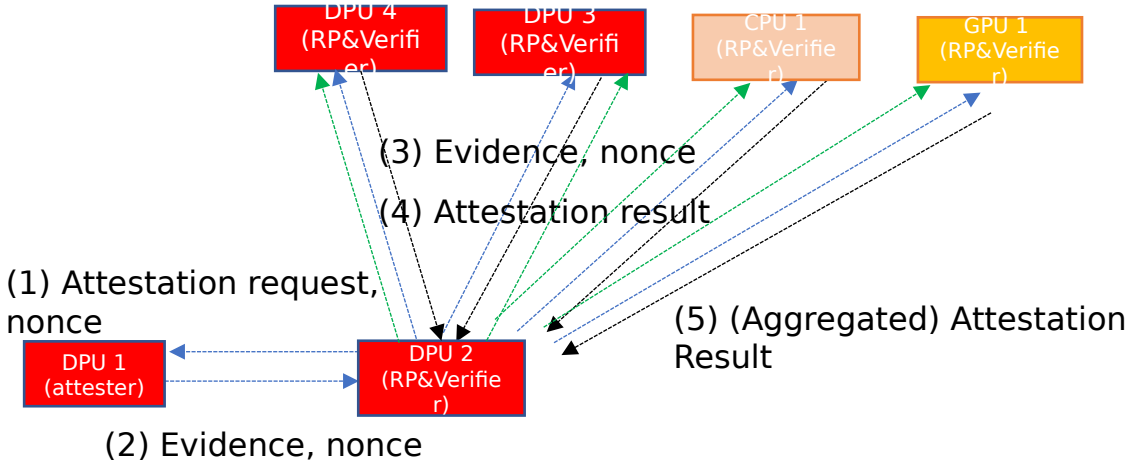


Fig. 9 Handling multiple Verifiers for the attestation in Data Center

Thank you