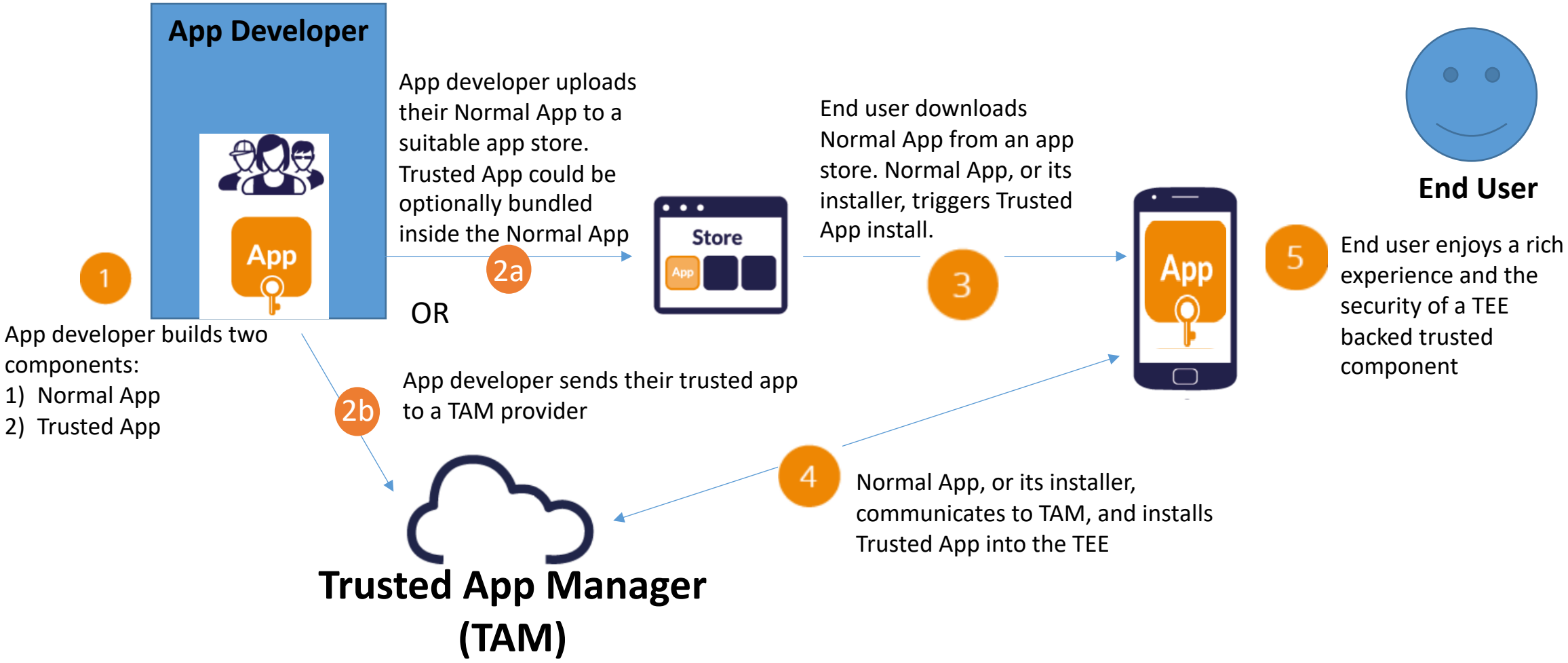
TEEP + RATS Alignment

Dave Thaler <dthaler@microsoft.com>

Topics

1. **TEEP Background for RATS folks**
2. TEEP's Use of RATS

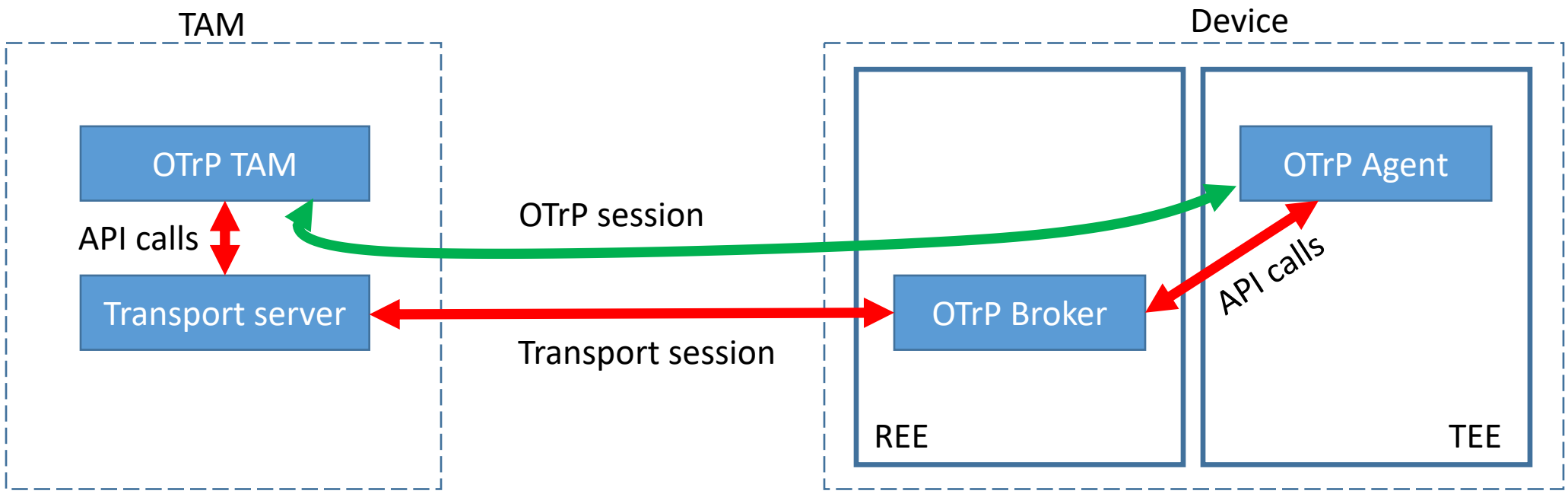
Entity Roles and Example Experience



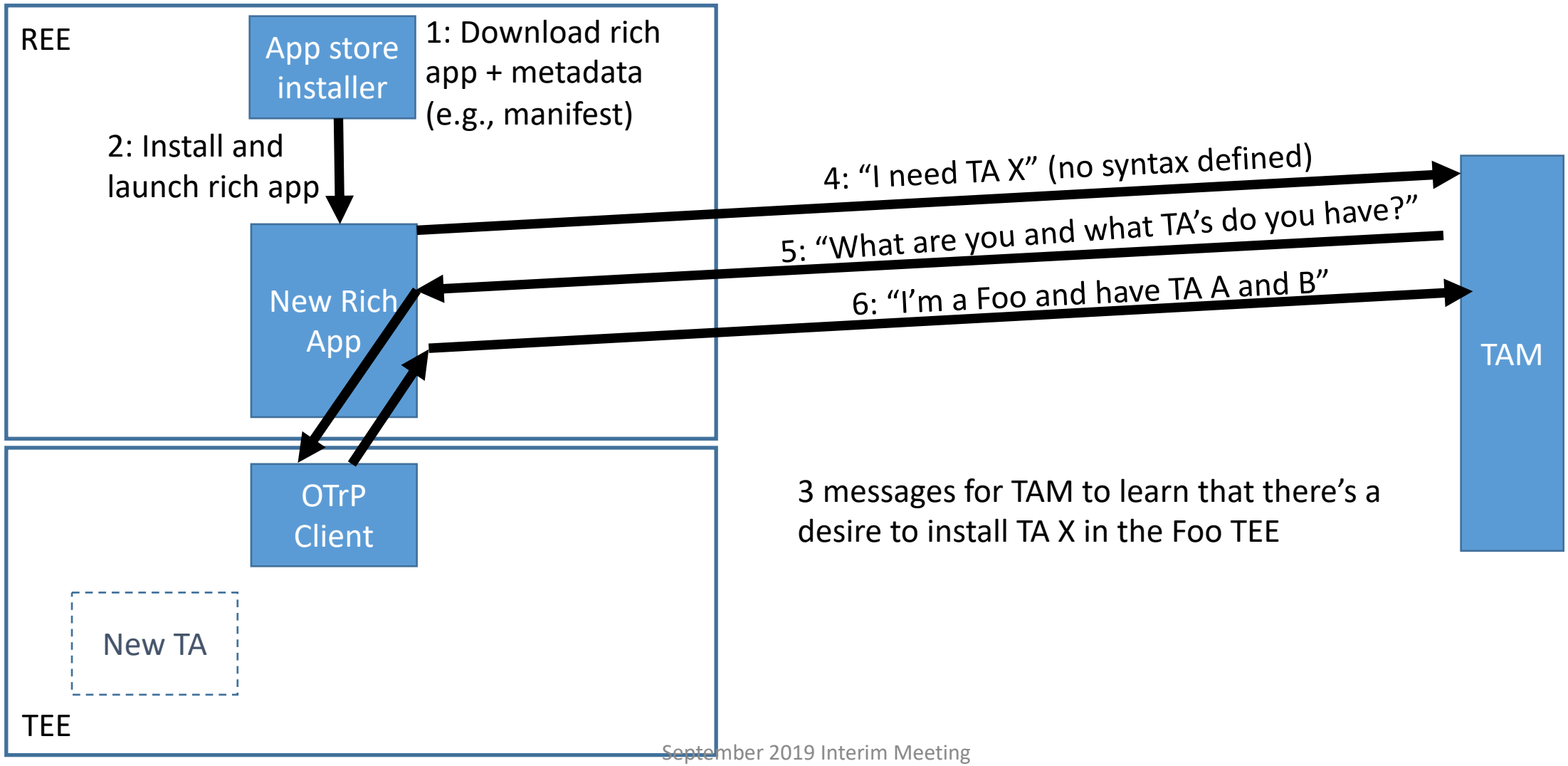
Entities with potential requirements

- **Device/TEE admin** wants to manage what TA's are allowed in its TEE (e.g., because of limited secure storage capacity)
- **Device/TEE admin** wants to keep a given TA and/or its config encrypted (independent of anything the author does) so needs to be in the loop when the TA is installed
- **Trusted Application author** wants to keep the TA code and/or its config encrypted (independent of anything the device/TEE admin does) and only let it be decryptable within a kind of TEE that it trusts to keep the info private, so needs to somehow be in the loop when the TA is installed
- **TEE chip vendor** wants to only allow authorized TA's to run in its chip, e.g., first vet the code as being safe under the assumptions that TEE chip makes
- **Device manufacturer** wants to only allow authorized TA's to run in the TEE on its devices, e.g., first vet the code as being safe under the assumptions that TEE chip makes

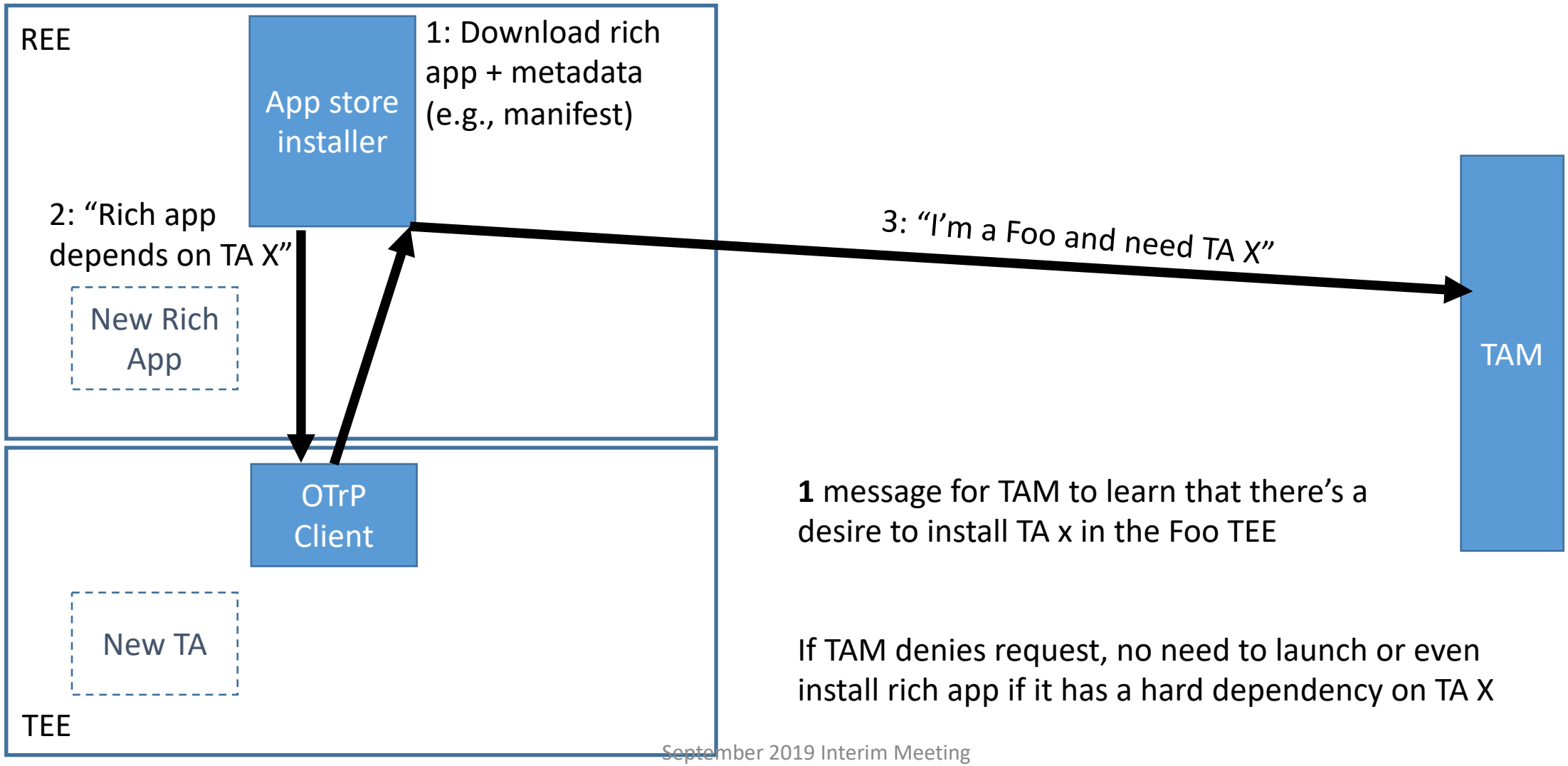
Protocol Roles



Connection model #1: Broker in app



Connection model #2: Broker in installer



1 message for TAM to learn that there's a desire to install TA x in the Foo TEE

If TAM denies request, no need to launch or even install rich app if it has a hard dependency on TA X

Topics

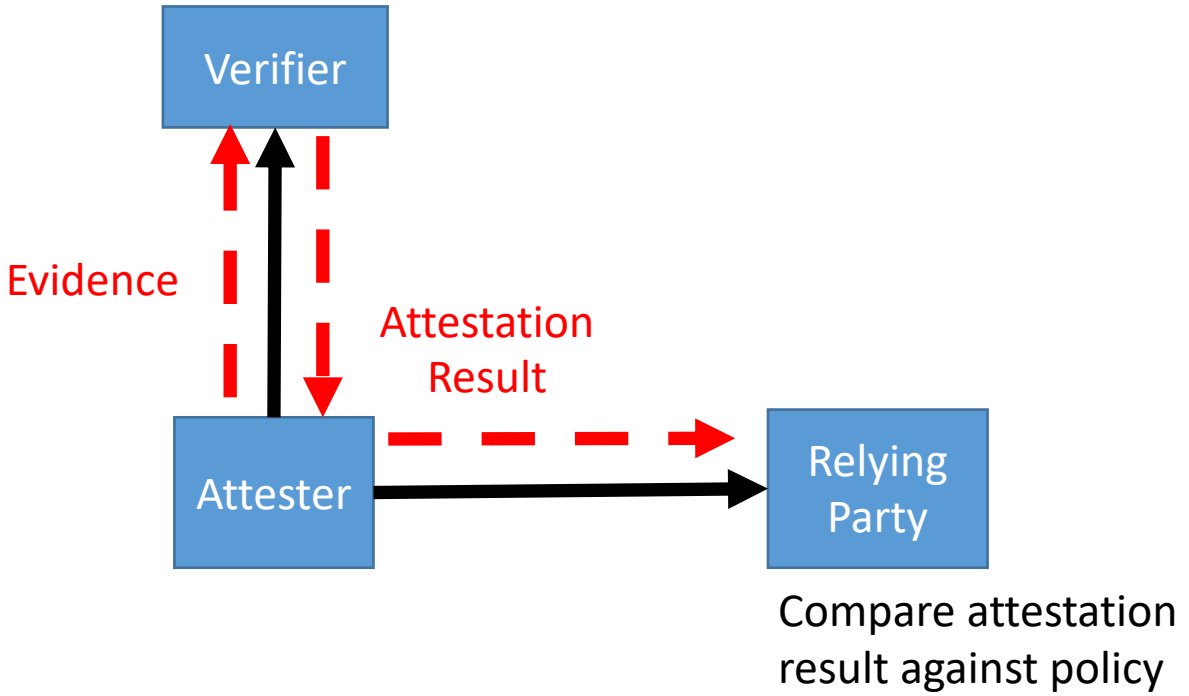
1. TEEP Background for RATS folks
2. **TEEP's Use of RATS**

Past discussion in TEEP

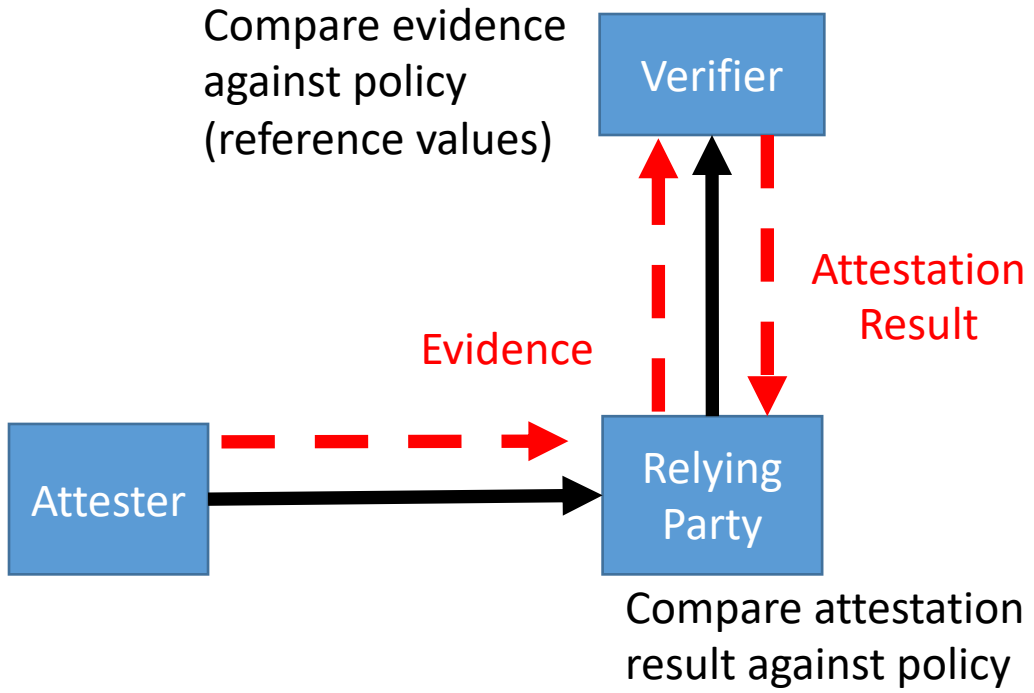
- Much of OTrPv1 “GetDeviceStateResponse” overlaps with what Remote Attestation (RATS) WG is chartered to do
  - Aligning with RATS would prevent duplication/conflict, and provide better modularity
  - Aligning with RATS would supposedly break compat with GlobalPlatform
  - RATS is much less far along than TEEP

RATS models

“Passport” model:



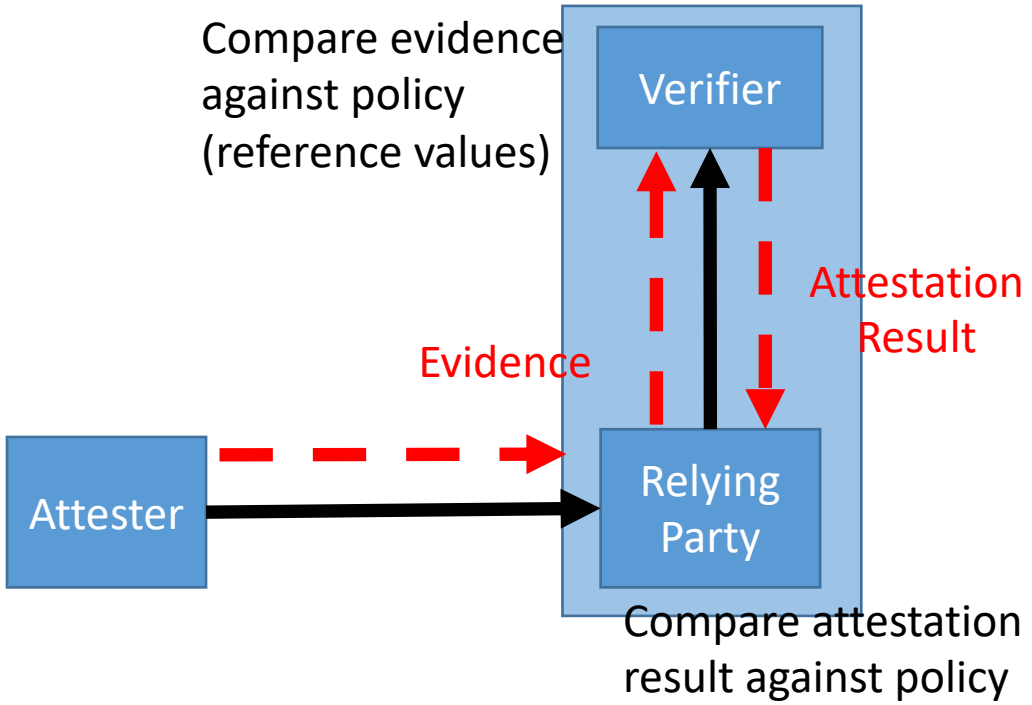
“Background check” model:



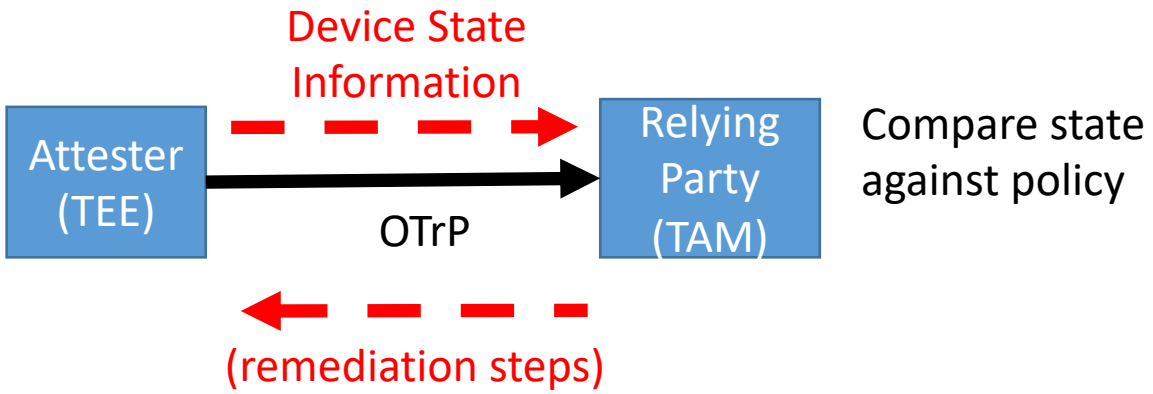
RATS models

Verifier could also be combined into same device Relying Party

"Verifying RP" model:



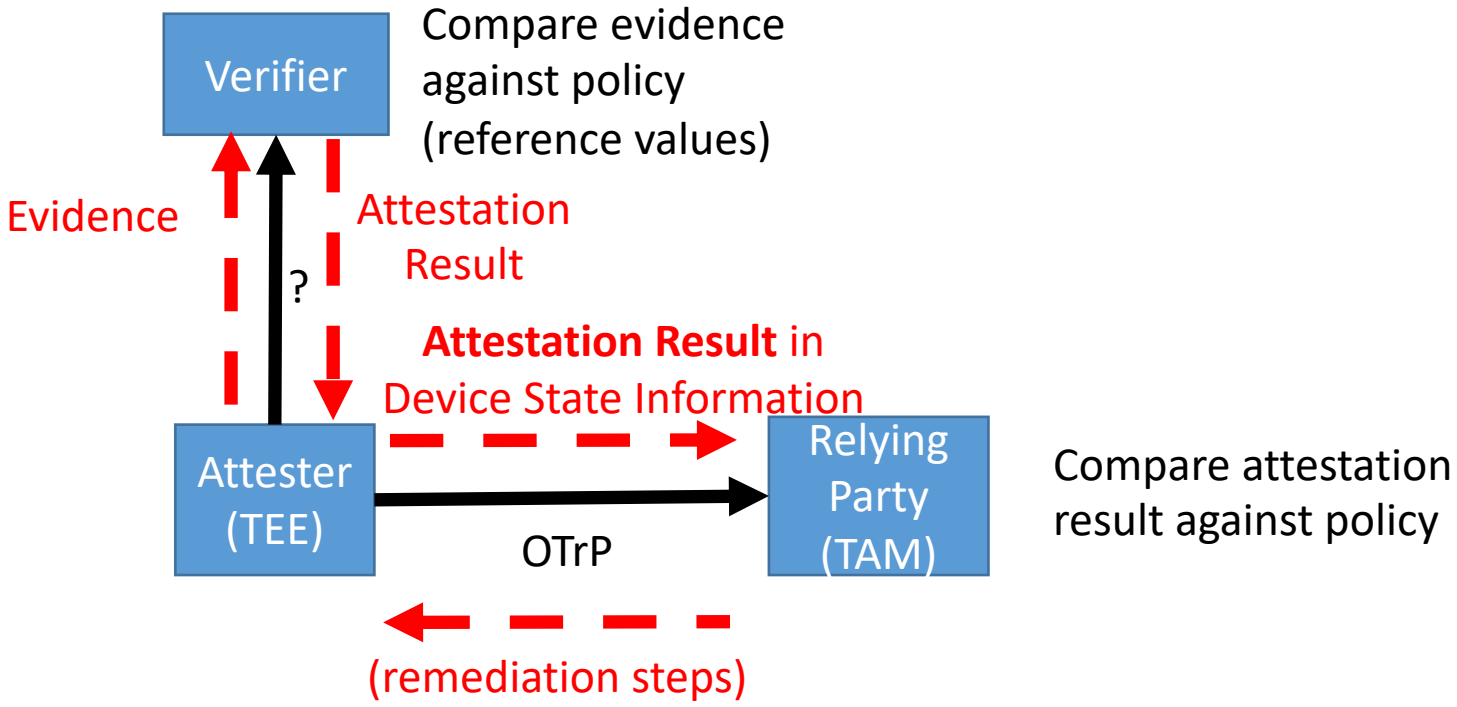
OTrP model for device state



There are at least 3 ways this *could* be combined with RATS models

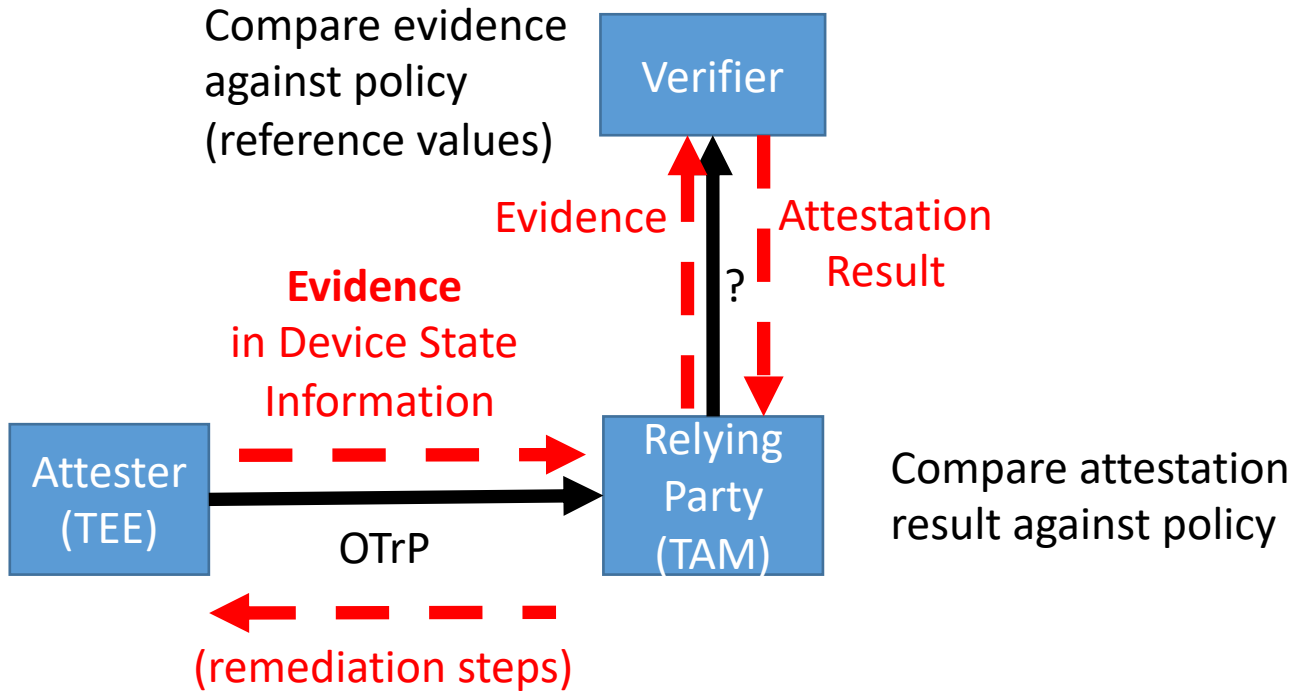
Option 1: Verifier and TAM used separately

Based on “Passport” model:



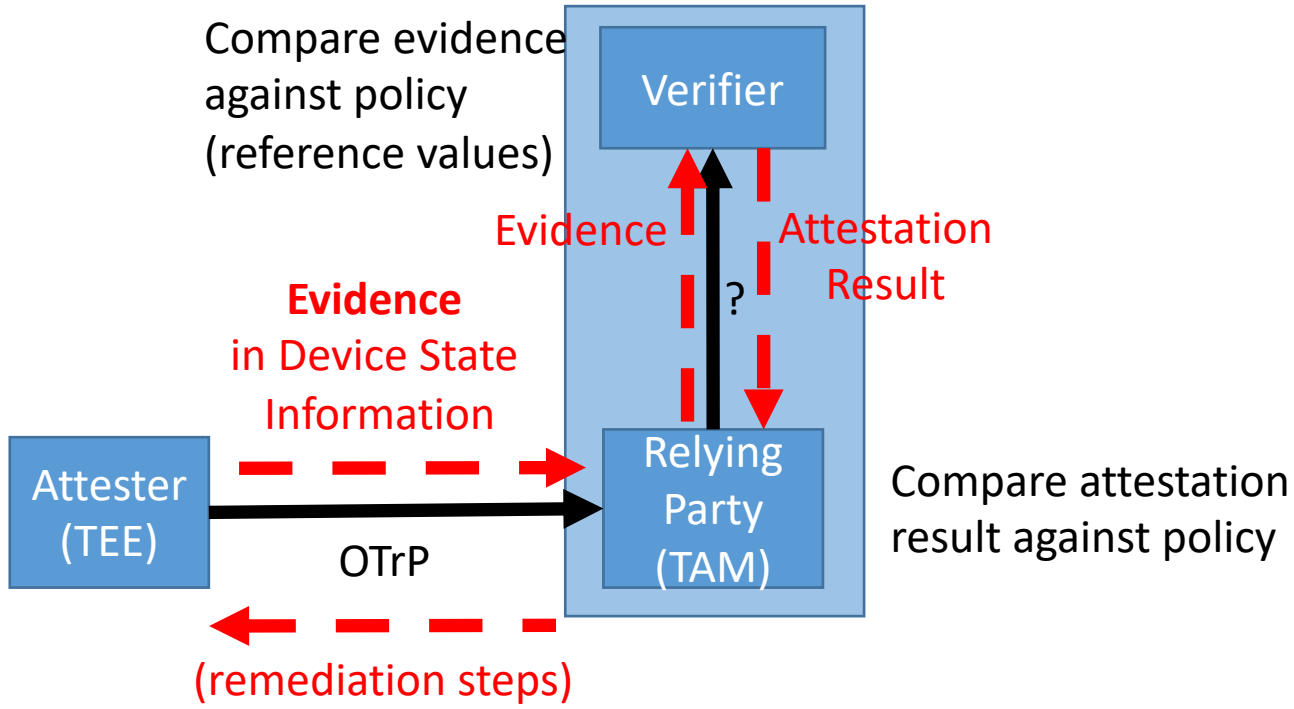
Option 2: Chained roles

Based on “Background check” model:

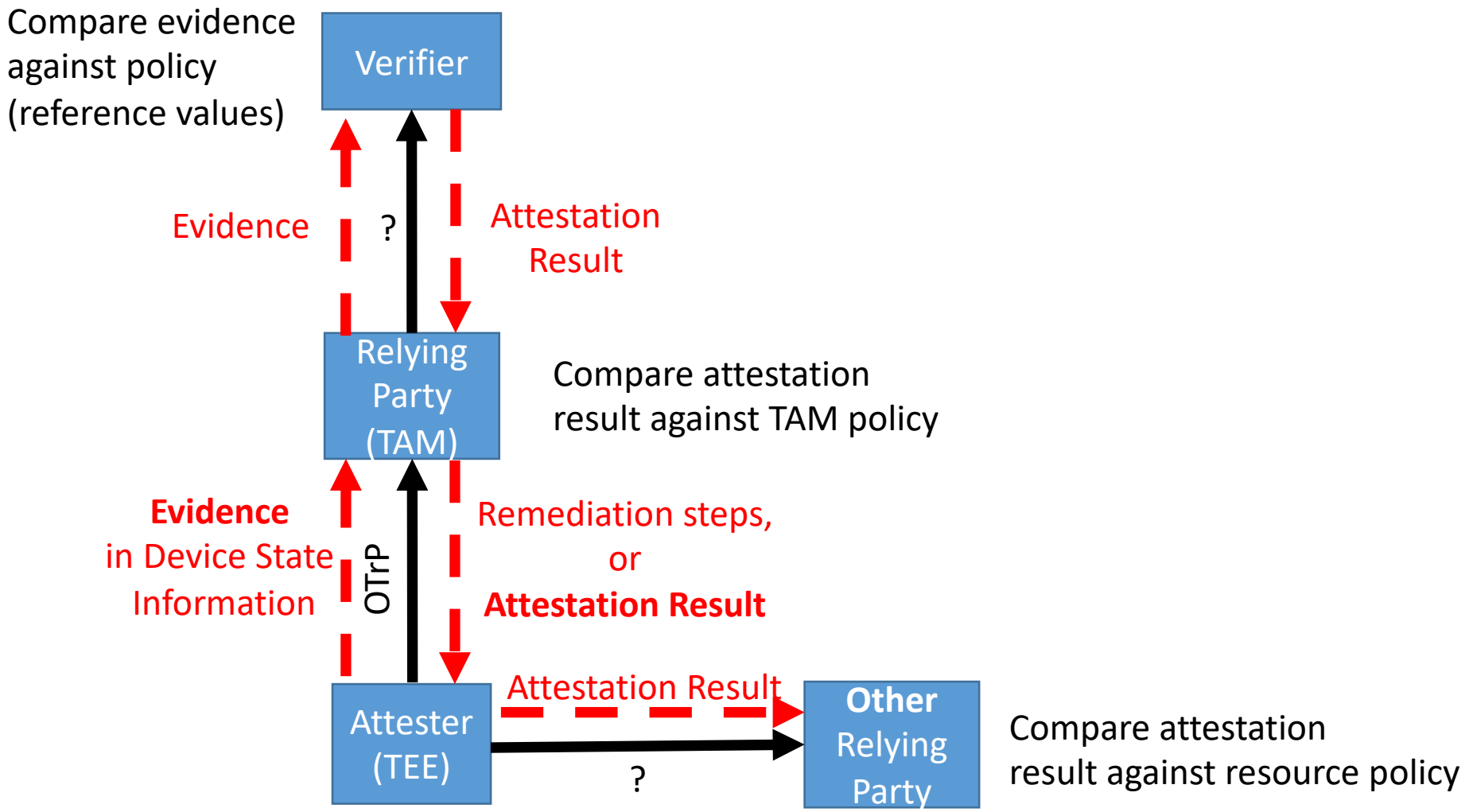


Option 3: Combined TAM/Verifier

Based on “Verifying RP” model:



Advanced use of OTrP in “Passport model”



Freshness

- RATS wants a nonce in a challenge ensure freshness of info
  - OTrPv1 has RID in GetDeviceStateRequest,
    and in signed GetDeviceState response,
    but not inside the encrypted DSI part of the response
  - OTrPv2 proposal has NONCE in QueryRequest,
    and inside EAT in QueryResponse
- Nonce alone does not ensure result is still valid at time of receipt
  - Policy might have changed since sending the attestation result
    - Covered in OTrP by accepting a time window for periodic policy change checks
  - Device might have rebooted since sending the evidence
    - Covered in OTrP by restarting TEEP Agent (Attester)<->TAM (RP) exchange

Claim sets for TEEP use

- draft-ietf-teep-architecture-03, section 7.3:
  - “it is expected that extensions to the attestation claims will be required as new TEEs and devices are created, the set of attestation claims required by TEEP SHALL be defined in an IANA registry. That registry SHALL be defined in the OTrP protocol with sufficient elements to address basic TEEP claims, expected new standard claims (for example from https://www.ietf.org/id/draft-mandyam-eat-01.txt), and proprietary claim sets.”

Questions/Discussion