TEEP Architecture
draft-ietf-teep-architecture-19

Ming (presenting)

Ming Pei, Dave Thaler, David Wheeler, Hannes Tschofenig

Timeline

- JAN 2020: WGLC completed
- JUL 2021: Draft-15 submitted to IESG
- JAN 2022: AD (Ben) feedback
  - https://mailarchive.ietf.org/arch/msg/teep/QJCjyUP-0vErQODB5-_PkvrKMXc/
- FEB 2022: Updated draft-16 to address AD feedback
  - https://mailarchive.ietf.org/arch/msg/teep/eW5OokuMPYGIIdRGKn1vbF14uQs/
- JUN 2022: Updated draft-17 / draft-18 to address AD feedback
  - Russ Housley (review of -16), Brendan Moran, Carl Wallace, Ben Kaduk
- NOV 2022: Updated draft-19 to address AD feedback / reviews
  - Secdir Benjamin M Schwarts (review of -16)
  - Genart Paul Kyzivat (review of -16)
  - Intdir Telechat Rob Halley (review of -18)
  - Iotdir Telechat Ines Robles (review of -18)
  - Compromised TAM description Ben Schwarts / Brendan Moran (review of –18)
  - Comments from Roman Danyliw
  - Comments from Robert Wilton

Section 1: Terminology Clarifications

Section1 days:

- "An application component ... is referred to as a Trusted Application”.

Ben S’s comments:

1. This is confusing. A component, explicitly **not an entire "application"**, is referred to as an "application". **"Trusted Component" would be more consistent**.
2. Regarding “Trusted Component”, **"trusted" seems to be the wrong adjective here**, as it is the environment, and not the software, that carries an elevated level of trust. "Isolated" might be a better descriptor.

Clarifications:

3. Clarifications and changes made in draft-19
    1. The protocol spec uses the term "Trusted Component”
    2. TA is already defined in the doc to mean both an app and a component. "**Trusted Application (TA): An application (or, in some implementations, an application component)** that runs in a TEE”.
    3. TA is also a common industry term. Added references about this.
4. We believe the word “trusted” is the right adjective. By trusting the TAM, you also trust (to not be malicious) the components the TAM installs.

Fixed in draft-19 with the following:

5. “An application component running inside a TEE is referred to **(e.g., in {{GPTEE}}, {{OP-TEE}},** etc.) as a Trusted Application (TA),”

Section 1: Wording Clarifications and Nits

Ben S's comments:

1. I would appreciate **some discussion of whether the Device Owner needs to trust the Trusted Application**, i.e. interaction between enclaves and sandboxes.
2. "verify the … rights of TA developers": "**rights**" is a loaded term. Rather than get into constitutional law, consider "**permissions**".
3. "so that the Untrusted Application can complete" -> "so that **installation of the** Untrusted Application can complete"
4. "is considered confidential" -- **By whom? From whom?** Consider "A developer who wants to provide a TA without revealing its code to the device owner…"

Roman Danyliw commented:

5. Is it the "danger of attacks" or the "consequences of attacks"?

Clarifications:

- We considered this is covered in the description of Device Administrator. A Device Owner may not always have control over TAs that go to its device regardless of the owner's trust. A Device Administrator may often decide when the Administrator is different from the owner. Secondly, the trust to a TA is delegated to the TAMs that may manage the TAs for installation to devices.
- (2) – (5) Changed as suggested

Fixed in draft-19 with the following:

2. Changed as suggested: "verify the … **rights** of TA developers" → "verify the … **permissions** of TA developers"
3. Changed as suggested: "so that **the installation of the Untrusted Application** can complete".
4. Changed as suggested: "if the code is considered confidential, **for example, when a developer who wants to provide a TA without revealing its code to others."**
5. Changed to "risk of attacks"

Section 3: Use Case Examples - Payment

Section 3.1. Payment says

- “**A trusted user interface (UI)** may be used in a mobile device to prevent malicious software from stealing sensitive user input data. "

Ben commented:

- Can you cite an example of a mobile device with a trusted peripheral that **is not accessible to the REE OS**? This seems theoretical.

Fixed in draft-19 with the following resolutions:

- “A trusted user interface (UI) may be used in a mobile device **or point-of-sale device**”

Section 3: Use Case Examples - IoT

Ben commented:

- Similarly, are there any examples of IoT devices that prevent the REE OS from operating certain actuators?

Ines Robles asked:

- Having an IoT scenario, in your opinion which type of Classes of Constrained Devices (Class 0, Class 1, etc. [RFC7228]) can participate in the TEE as a "Device" in Figure 1.

Clarifications:

- Multiple examples from David Thaler and Brendan's email comments, and Ben suggested to add a reference. Added a reference about Global platform TEE and text.
- On Ines's comment, there is no clear spec from RFC 7228 to say which classes of IoT devices may fit. We will not specify it and leave such recommendation to the adopters. And the TEEP allows any code as long as the capacity fits.

Fixed in draft-19 with the following:

```
For example, [GPTEE] uses the term "trusted peripheral" to refer to such things being accessible only
from the TEE, and this concept is used, and this concept is used in some GlobalPlatform-compliant devices
today.
```

Section 4.1 / 9.5: Threat model of a Hostile / Compromised TAM

Section 4.1 says

- “For a TAM to be successful, it must have its public key or certificate installed in a device's Trust Anchor Store.”

Ben S’s comment:

- This needs discussion of threat model. What damage can a hostile TAM do? What does the device administrator need to know for adding a trust anchor to be safe?

Brendan M’s comment:

- Insufficient description about Compromised TAM.

Fixed in draft-19 with the following:

- Expanded the threat in Section 9.5 about “Compromised TAM” as follows, thanks Brendan for initial verbiage that we revised over.

```
Device TEEs are responsible for validating the supplied TAM certificates. A compromised TAM may bring multiple threats and damage to
user devices that it can manage and thus to the Device Owners. Information on devices that the TAM manages may be leaked to a bad
actor. A compromised TAM can also install many TAs to launch a DoS attack on devices, for example, by filling up a device's TEE
resources reserved for TAs such that other TAs may not get resources to be installed or properly function. It may also install
malicious TAs to potentially many devices under the condition that it also has a Trusted Component signer key that is trusted by the
TEEs. This makes TAMs high value targets. A TAM could be compromised without impacting its certificate or raising concern from the
TAM's operator.

To mitigate this threat, TEEP Agents and Device Owners have several options, including but potentially not limited to those listed
below, for detecting and mitigating a compromised TAM:

1. Apply an ACL to the TAM key, limiting which Trusted Components the TAM is permitted to install or update.
2. Use a transparency log to expose a TAM compromise: TAMs publish an out-of-band record of Trusted Component releases, allowing a
   TEE to cross-check the Trusted Components delivered against the Trusted Component installs in order to detect a TAM compromise.
3. Use remote attestation of the TAM to prove trustworthiness.
```

- A “hostile TAM” is considered the same as a ”Compromised TAM” where the TAM administration or team is compromised.

Section 4.1: TAM trust by public key with constraints

Section 4.1 says

- “The TAM is trusted by a device if the TAM's public key is, or chains up to, an authorized Trust Anchor in the device.”

Bob Halley commented:

- If you have read carefully and remember the definition of Trust Anchor, you realize this means ***the TAM is trusted subject to the constraints*** on its authority, but it might be good to reiterate this point here, as it reads like "is unconditionally trusted" if you don't remember the definition. Also, it was not clear if the chaining process could have further restricted the scope of the TAM, e.g. due to additional restrictions on certificates beneath the trust anchor.

Fixed in draft-19 with the following:

- Added a sentence about conforming to “constraints” as follows.

“The TAM is trusted by a device if the TAM's public key is, or chains up to, an authorized Trust Anchor in the device, **and conforms with all constraints defined in the Trust Anchor**.”

Section 4.1: TEEP Broker contacts TAM

Section 4.1 says

> "As shown in Figure 1, the TAM cannot directly contact a TEEP Agent, but must wait for the TEEP Broker to contact the TAM requesting a particular service. This architecture is intentional in order to accommodate network and application firewalls that normally protect user and enterprise devices from arbitrary connections from external network entities."

Bob Halley commented:

- This text says that the broker must contact the TAM "requesting a particular service". This is unclear; does it mean "the TEEP Agent must ask the broker to initiate a TAM connection", or "the broker can initiate the connection, but only when it has some specific service it wants", or something else?
- While I understand the rationale to permit broker initiation of connections to avoid firewall issues, it seems overly restrictive to require it.

Fixed in draft-19 with the following (David T PR #247):

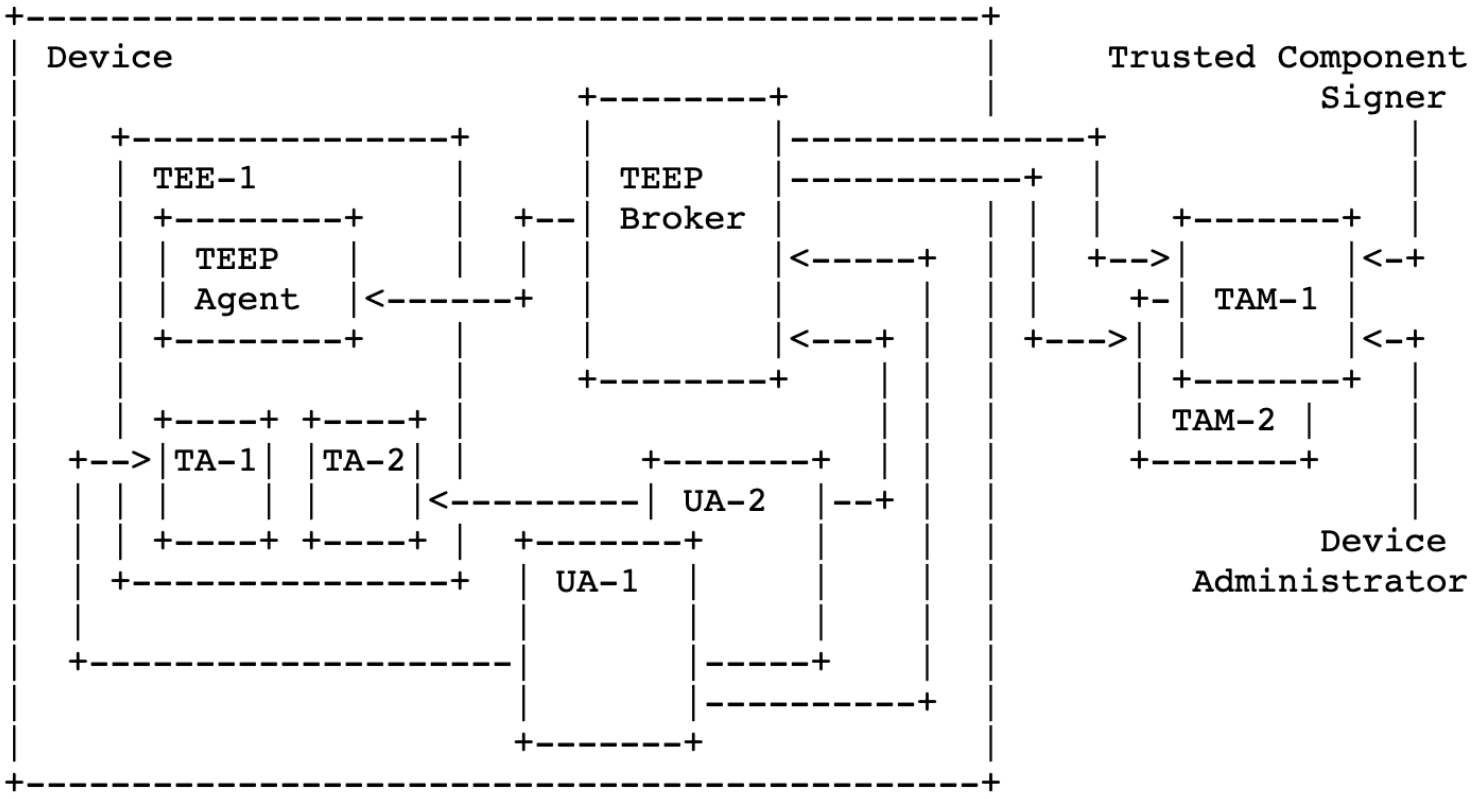
> "When the TEEP Agent runs in a user or enterprise device, network and application firewalls normally protect user and enterprise devices from arbitrary connections from external network entities. In such a deployment, a TAM outside that network might not be able to directly contact a TEEP Agent, but needs to wait for the TEEP Broker to contact it. The architecture in Figure 1 accommodates this case as well as other less restrictive cases **by leaving such details to an appropriate TEEP transport protocol** (e.g., [I-D.ietf-teep-otrp-over-http], though other transport protocols can be defined under the TEEP protocol for other cases)."

Section 4.1: Figure 1 Inconsistent Notations

Bob Halley commented:

- In Figure 1 and 2, "App-1" and "App-2" should probably emphasize that they are Untrusted Applications, so **perhaps "UA-1" and "UA-2" would be better**. Also in these figures, the Trusted Applications are labeled "TA1" and "TA2" which is a little strange as all of the other labels have "-", so **"TA-1" and "TA-2" would be better**.

Fixed in draft-19 with the following: PR #243



Section 4.4: Lower case use of “must”

Section 4.4 says

1. “**Implementations must support** encryption of such Personalization Data to preserve ….”
2. “**must support** integrity protection of the Personalization Data.”

Ben’s comment:

3. Implementation of what?
4. Lower-case "**must**" without explanation. Why, and is this a normative requirement?
5. “For example, if all instances of a TA share a secret key, used for decrypting the Personalization Data

Fixed in draft-19 with the following resolutions:

- “**`Implementations of TEEP protocol`** `must support encryption`…”
- “Personalization Data is encrypted with **a key unique to that specific TEE**, as discussed in {{trustanchors}}.”
- David: Some parts of the IETF have a negative reaction to using normative MUST type language in an informational architecture document, so this uses normal English rather than 2119 language per such feedback.

Section 4.4: Clarify “support encryption” purpose

Section 4.4 says

- “**Implementations must support** encryption of such Personalization Data to preserve ….”

Roman Danyliw commented:

- Please clarify what it means to “support encryption.” Is this saying that all personalization data at rest must be encrypted? No personalization data can be sent in the clear?

Fixed in draft-19 with the following:

- “The encryption is used to ensure that no personalization data is sent in the clear."

Section 4.5: Figure 3 sequence update

Paul Kyzivat commented:

- I find this figure confusing. It starts out looking like a sequence diagram, where time flows from top to bottom. But then overlayed on it is a nested text outline that seems to interact with the sequence diagram. Based on the outline numbering I expect the time sequence to be 2a,2b,3,4. but based on positioning within the sequence diagram it seems that the order should be 2a,3,2b,4. I don't understand how this is intended to be read.

Robert Wilton similar comments:

- I found this quite unclear from the diagram, it looked like the messaging is initiated from the TAM to the Device with the TEE. I also found the time flow in this diagram to be somewhat unclear, since normally time flows downwards with sequence diagrams, but I wasn't convinced that was the case here. E.g. TA -- 2b is lower than 3. Install.

OLD Draft-18

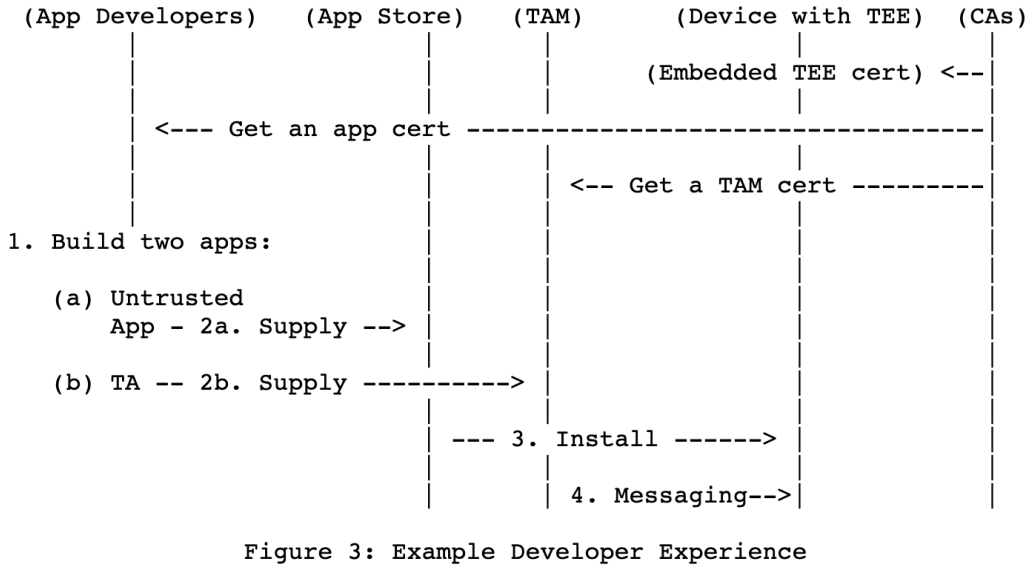


Figure 3: Example Developer Experience

NEW Draft-19

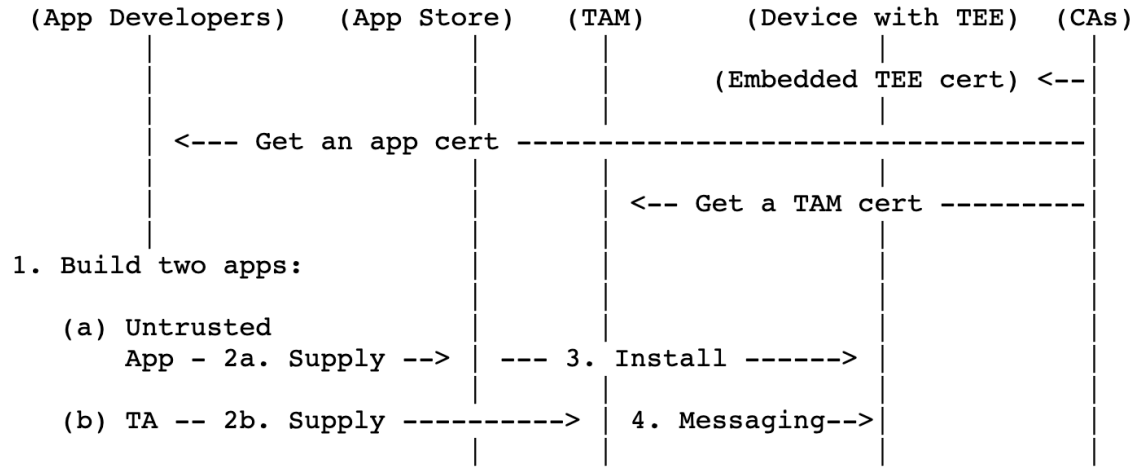


Figure 3: Example Developer Experience

Section 6.1: TEEP Broker “abstracts” update

Section 6.1 Role of the TEEP Broker says

- “A TEEP Broker abstracts the message exchanges with a TEE in a device.”

Ben / Roman commented:

- What does it mean for the broker to “abstract” the message exchange?

Fixed in draft-19: Rewritten the entire paragraph

```
A TEEP Broker interacts with a TEEP Agent inside a TEE,
relaying messages between the TEEP Agent and the TAM, and may also interact with
one or more Untrusted Applications (see {{apis}}).
The Broker cannot parse encrypted TEEP messages between a TAM and a TEEP agent
but merely relays them.


When a device has more than one TEE, one TEEP Broker per TEE could
be present in the REE or a common TEEP Broker could be used by multiple TEEs
where the transport protocol (e.g., {{I-D.ietf-teep-otrp-over-http}}) allows
the TEEP Broker to distinguish which TEE is relevant for each message from a TAM.
```

Section 6.2: TEEP Broker API spec

Section 6.2.1 TEEP Broker APIs says

- “The following **conceptual APIs** exist from a TEEP Broker to a TEEP Agent:”

Robert Wilton commented:

- I'm slightly surprised that the conceptual TEEP Broker APIs are contained in this document when the other equivalent TEEP APIs are not.

Clarification:

- The protocol APIs are very specific via a separate RFC. These Broker APIs are illustrative or "conceptual" only, which we cannot specify exactly what forms it take, which is up to the provider, usually a TEE provider or device provider to bundle some TEEP Broker that can work with the TEE they embed.

Section 5.4: Rules about supported x.509 extensions?

Section 5.4 says

- “When a PKI is used, many intermediate CA certificates can chain to a root certificate, each of which can issue many certificates”

Ben’s comment:

- Intermediate CAs have a troubled history (e.g. [1]), and techniques that make them safer (e.g. x.509 name constraints) can't be deployed as a retrofit. Does TEEP need some rules about supported x.509 extensions?

Replies:

- We leave this to the device provider for the constraints on X509 extensions it supports and uses in Trust Anchor validation. They may select to trust only a selected intermediate CA instead of the root as the Trust Anchor.

Section 6.2: Mitigate leaking a TA upon removal

Ben S's comment:

- If an Untrusted Application is summarily deleted, how do you avoid leaking the TA?

Clarifications:

- Similar to the removal of a buggy or malicious TA, this is up to a device to have some scheme to contact TAM or be contacted to initiative a removal of TAs that are not needed anymore.

Section 7: Clarify “class of device” in Attestation

Section 5.4 says

- “In some use cases it may be sufficient to identify only the class of the device.”

Ines Robles commented:

- What do you mean with class of device? Perhaps would be nice to add between brackets some examples

Fixed in draft-19: PR #260

```
In some use cases it may be sufficient to identify only the model or class of the device,
for example, a DAA Issuer's group public key ID when the attestation uses DAA, see [
I-D.ietf-rats-daa]. Another example of models is the hwmodel (Hardware Model) as defined
in [I-D.ietf-rats-eat].
```

Section 9: Compromised TEE?

Section 9. Security Considerations

Robert Wilton commented:

- I note that there is no security considerations for a compromised TEE. Should this be considered, or is it the case that TEEs cannot be compromised?
- Similarly, is a compromised TA something that should be considered here?

Clarification:

- We considered "Compromised TEE" out of scope for this. The device provider or owner needs to monitor and patch the entire device. There isn't much the TA that the protocol manages here can do.
- Compromised TA has been documented (draft-18). Robert’s review was about draft-16.

General: Inclusive language

Lars Eggert / Roman Danyliw suggested to use “inclusive language”

- Found terminology that should be reviewed for inclusivity; see https://www.rfc-editor.org/part2/#inclusive_language for background and more guidance:
- Term man; alternatives might be individual, people, person
- Terms she and he; alternatives might be they, them, their

Fixed in draft-19: PR #257

- Man-in-the-middle **manipulator**-in-the-middle

Other Minor / Nits Update (Multiple Reviewers)

1. Section 1
    1. Ben Schwartz: "A TEE … **wants** to determine…". Ben commented: "I suggest '**needs'**". Changed as suggested.
    2. Ben Schwartz / Robert Wilton: "so that the Untrusted Application can complete". "so that **installation of the** Untrusted Application can complete"
    3. Paul Kyzivat: s/its/their in "TEEs use hardware enforcement combined with software protection to secure TAs and its data."
2. Section 2: Device User definition has a broken line "Relates to Device Owner and Device Administrator."
    - Bob Halley / Paul Kyzivat. Removed the line.
3. Section 4.4.2 "e.g., OP-TEE"
    - Ben S / Robert Wilton commented: "What is this?". Changed to "**(e.g., OP-TEE {{OP-TEE}})**", where a reference to OPTEE is added.
4. Section 9.3: "We have already seen examples of attacks on the public Internet **with billions of compromised devices** being used to mount DDoS attacks."
    1. Ben commented: "Citation please. Also, are you sure it has reached into billions?"
    2. Changed to "a large number of devices"
5. Lars Eggert findings
    1. unauthorised unauthorized, Untrused Untrusted, faciliates faciliatates, "smart phones" "smartphones, "long lived" long-lived
6. Robert Wilton findings
    1. "TEEs will be allowed to be personalized" "TAs in the TEEs will be allowed…"
    2. "OEM time" "device manufacturing time"
7. Roman Danyliw findings
    1. "A TEE can be the best way" "A TEE can be one of the best ways"
    2. Administators Administrators, "verifier" "Verifier", "a TEEP Broker broker"