IETF 102 参加報告
IoT/デバイス認証関連

2018-08-31

株式会社東芝 研究開発センター
安次富 大介 （AJITOMI, Daisuke）

自己紹介

- 株式会社東芝 研究開発センター ネットワークシステムラボラトリー所属
- 2012年頃～ 主にIoT向けのクラウドサービス技術の開発に従事
  - デバイスをGW経由or直接クラウドに繋げ, アプリ・ユーザ側にはWeb I/Fを提供するような基盤
  - 適用先: デジタルプロダクツ, HEMSシステム, 社会インフラシステム等

- これまで自製してきたIETF関連技術
  - SIP, HTTP/1系, WebSocket, OAuth(&OIDC) IdP, etc.
- 標準化活動
  - IETFは99から参加。現状, 標準化提案を行っているわけではない。調査が主。WebとIoT。
  - W3Cでは, 有志で"HTTPS in Local Network" というコミュニティグループを立ち上げ活動中
    - https://www.w3.org/community/httpslocal/

トピック: IoT/デバイス認証関連

- 「デバイス認証」自体は, 基本的に証明書, RPK, PSKベース
  - 通信のセキュリティは, TLSかDTLSが主流
  - あるいは、通信を抽象化し, End-to-Endセキュリティメカニズムを使う方向性
    - JOSE/JWT/JWK, COSE/CWT/CWK など
- 「デバイスには証明書やPSKが事前に埋め込まれています」という前提をとる話が多い
  - どう埋め込むか？埋め込まれた証明書のトラストアンカーは？等のディテールは、Out-of-Scope
- この「Out-of-Scope」な話題に向き合ってそうな取り組みをメインに紹介
  - 認証の前段として、システムの中でIoTデバイスの正当性をどう担保するか？
  - デバイスのトラストアンカー、Attestation
- トピック
  - Security Dispatch (RATS, EAT)
  - SUIT, TEEP
  - ANIMA(IoTではないが), ACE(主に認可だが)

Attestation とは？

- 本発表での定義
  - あるシステム・エンティティ(IoTデバイス等)が, 自身の正当性(システム・エンティティとして認定されたものであること)を証明するための行為

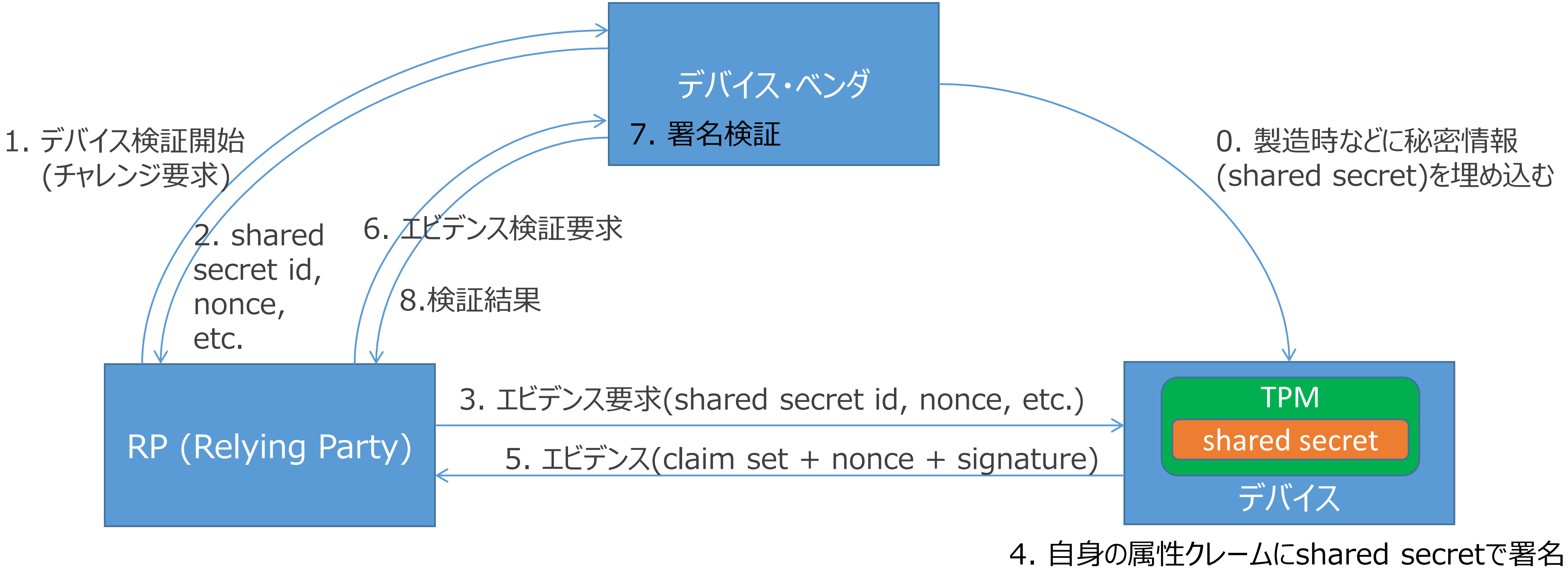
- 例えば
  - デバイスが, あるベンダーに製造されたものであることを, 同じベンダー管轄のFW配布サーバに対して証明する

  - FIDOの認証器(ハードウェアキー)が, FIDOサーバに認定されたものであることをRP (Relying Party)（=FIDOサーバを信頼する第三者）に対して証明する

- システム外のエンティティに向けて証明すること ⇒ **Remote Attestation**

Remote Attestation

- ベンダを信頼する第三者(RP)が, あるデバイスが, そのベンダによって製造されたものであることを検証できる(以下, 実現例)



RATS / EAT

- SECDISPATCH WGにて, Attestationに関する2つの提案あった
  - いずれもWGではないが, 今春にnon-WG mailing list, Githubプロジェクトが立ち上がっている
- RATS (Remote Attestation Procedures)
  - Remote Attestationの用語定義とリファレンスインタラクションモデル定義
  - Fraunhofer, GE, ARM, Huawei, etc.
- EAT (Entity Attestation Tokens)
  - 前頁の「エビデンス」に相当するEATのフォーマット定義
  - QUALCOMM, Cisco, ARM, etc.

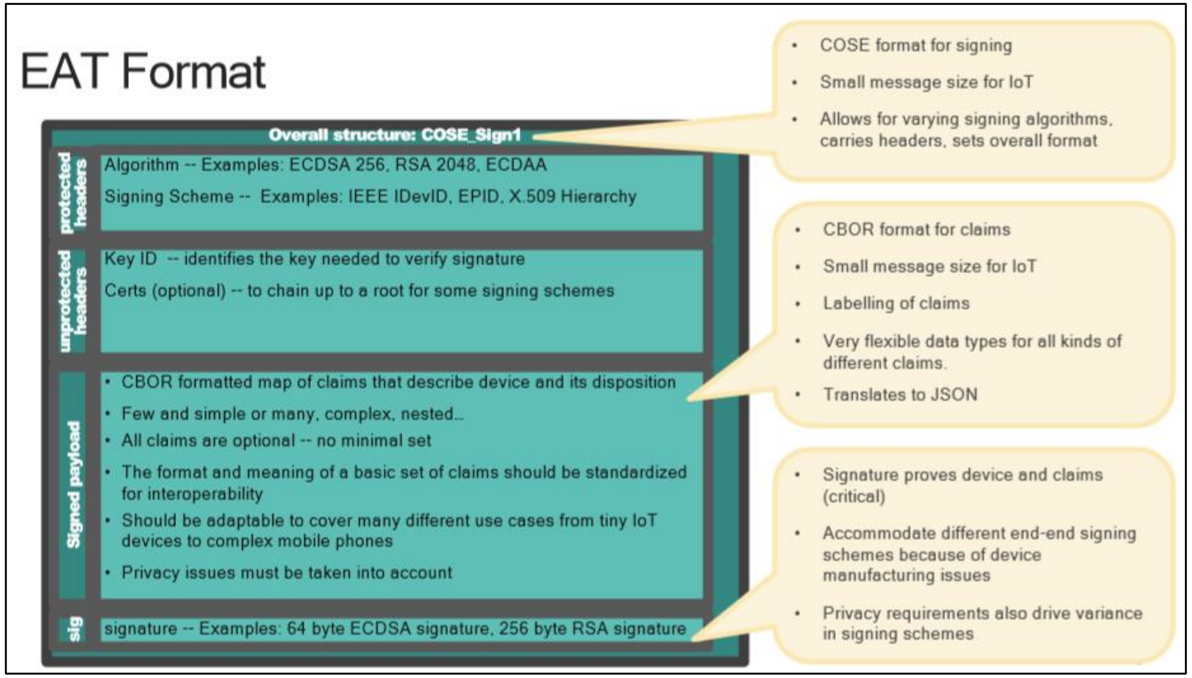
RATS (Remote Attestation Procedures)

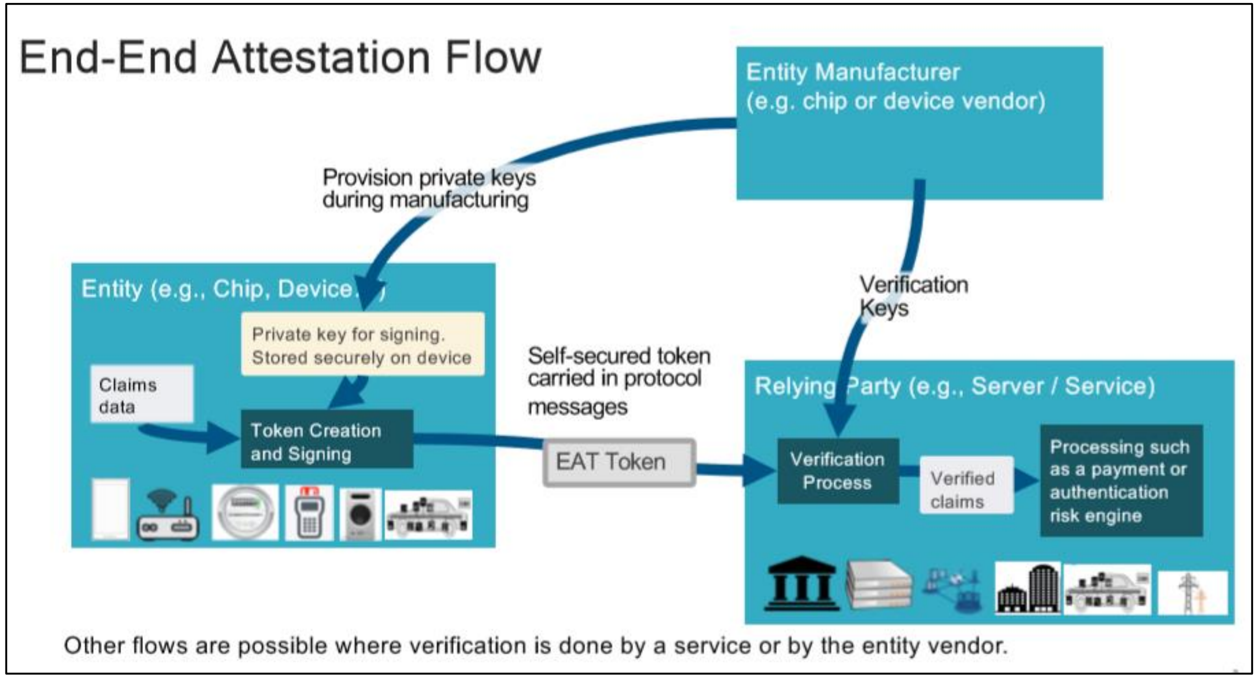
- 用語定義
  - https://datatracker.ietf.org/doc/draft-birkholz-attestation-terminology/
  - Attester(デバイス側)とVerifier(検証側)
  - Remote Attestation: 以下の３つのActivityで構成される手続き。
    - Attestation: Attesterのプロパティに関するクレーム(エビデンス)の生成としての実現されるオブジェクトの完全性認証
    - Conveyance: AttesterからVerifierへのエビデンスの伝送
    - Verification: エビデンスの評価
- リファレンスインタラクションモデル定義
  - https://tools.ietf.org/html/draft-birkholz-reference-ra-interaction-model-00
  - チャレンジレスポンス型のインタラクションモデル

```
[Attestor]                                              [Verifier]
     |                                                       |
     |  <------------ Nonce, Shared Secret ID, [Additonal Info]
     |                                                       |
 Collect Integrity Claims                                    |
     |                                                       |
 Sign Claim Set (Evidence)                                   |
     |                                                       |
     |  Evidence, Identity, Nonce, Signature ----------------->
     |                                                       |
     |                                          Appraise Evidence
     |                                                       |
```

EAT (Entity Attestation Tokens)

- 概要
  - https://github.com/eat-ietf-wg/eat-slides
  - ConstrainedなIoTデバイスを考慮して, CBOR/COSEベースのフォーマットを提案
  - RATSとは別にAttestation Modelを定義





引用元: https://github.com/eat-ietf-wg/eat-slides/blob/master/EAT%20Standardization%20v5%20long.pdf

RATS / EAT

- RATS
  - プロトコルを定義する（チャレンジレスポンス型、Freshnessも検証）
    - 従来のRemote Attestationでは、attestationからverificationまでに時間の縛りがなかった
- EAT
  - プロトコルは決めない
    - FIDOのExtensionや、HTTPヘッダとして、tokenを入れる
- 議論
  - 両者の差は？⇒ 基本的なアーキテクチャは同じ、差は上記
  - 両取り組みを統合できないか、意思疎通は図れているのか? ⇒ 話はしている
  - FIDOのようにグループを対象にしたattestationも対象にするのか? ⇒ 両者Yes
  - attestationは、Replay攻撃対策が重要
  - 特定のWGでの議論を促すコメントは無し

SUIT (Software Updates for IoT)

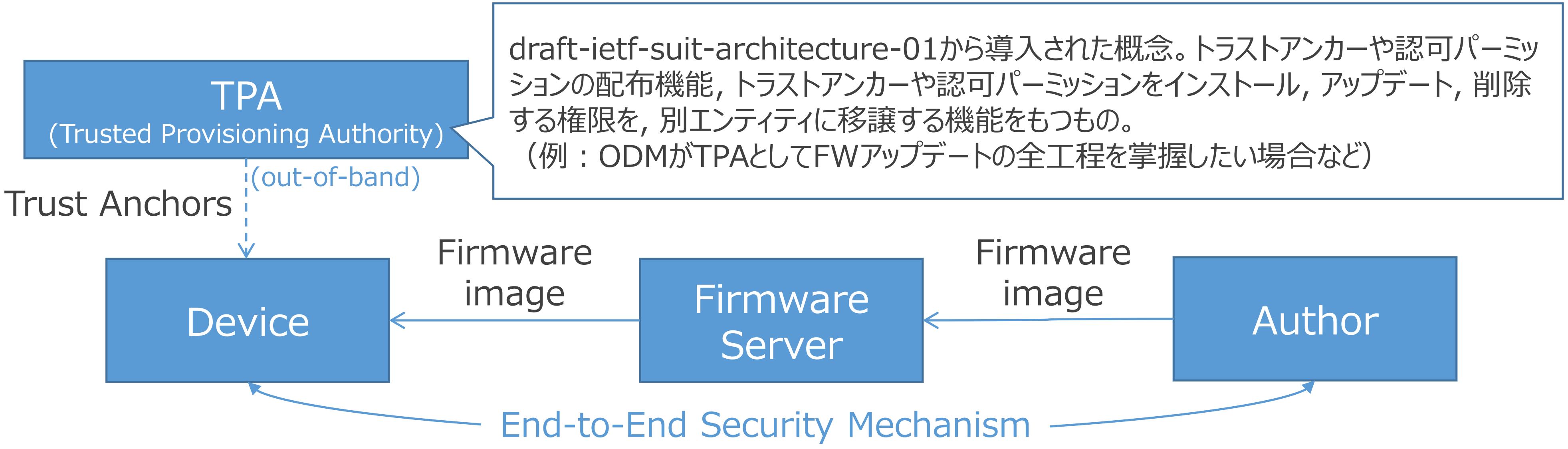
- 概要
  - IoTにおいて堅牢かつセキュアなFWアップデートメカニズムは，重要かつ解決が難しい課題。
  - SUITは，Class1(RFC7228: ~10KiB RAM/~100KiB flash)デバイスに対しても，こうしたFWアップデートが適用できるように要件やアーキテクチャの検討を進めている。
  - IETF100で開催されたBoFにて，上記クラス等のスコープやトランスポート層を定義しない等の基本方針が明確化され，WG化

SUIT (Software Updates for IoT)

- IETF 102 動向
  - 前回に引き続きハッカソン開催
    - 前回, デバイス上でのCOSE署名検証には至らなかったが, 今回, 以下を用いてワンパス通った模様
      - github.com/intel/tinycbor.git, github.com/bergzand/libcose.git
  - 主にアーキテクチャとマニフェストフォーマット(COSE/CBORベース)を議論
    - A Firmware Update Architecture for Internet of Things Devices
      - https://tools.ietf.org/html/draft-ietf-suit-architecture-01
      - 要件とトランスポートを抽象化したアーキテクチャを定義
      - 新しく導入された概念(TPA, etc.)やユースケース(rollback, etc.)に関する議論
    - Firmware Updates for Internet of Things Devices - An Information Model for Manifests
      - https://tools.ietf.org/html/draft-ietf-suit-information-model-01
      - 脅威分析やセキュリティ要件が追加されるなど, 初稿から大幅に拡充された
      - CBORデータ定義の詳細議論(primary objectをmapにするかarrayにするか等々)

SUIT (Software Updates for IoT)

- FWアップデートの通信アーキテクチャとセキュリティ
  - (A Firmware Update Architecture for Internet of Things Devices より)
  - 現時点では, 非対称暗号・PKIの活用を想定
    - リソース制約の非常に厳しいデバイス向けに, 将来的には対象鍵ベースのアプローチにも言及するかも
  - FWイメージを守るため, End-to-Endセキュリティメカニズムを用いる
    - デバイスは, TPAから事前にトラストアンカーリストを取得しておき, これに基づきFWやマニフェストの署名を検証
    - Author側も, FWイメージの機密保護のために, デバイスの証明書/公開鍵, あるいは PSK を保持してデバイス認証できる必要がある

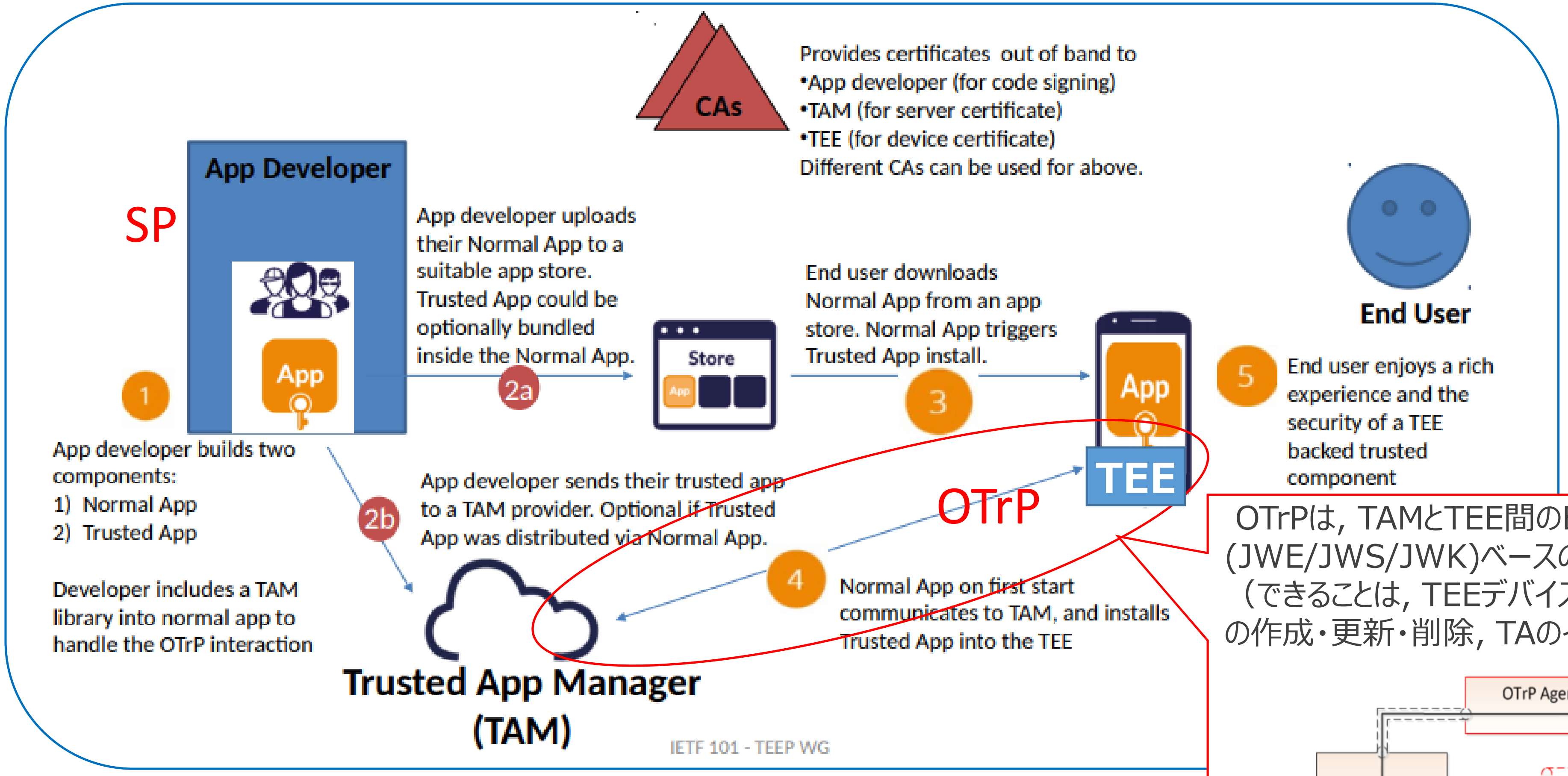


TEEP (Trusted Environment Provisioning)

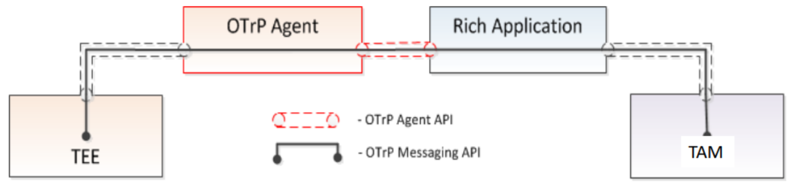
- 概要
  - TEE(Trusted Execution Environment)上で動くトラステッドなアプリケーション(TA: Trusted App)のライフサイクル管理(インストール,アップデートなど)のプロトコル標準化を進めるＷＧ。
    - TAのユースケース: Payment, Authentication, IoT, Confidential cloud computing
  - 主にアーキテクチャとOpen Trust Protocol (OTrP)の2つが, 前回IETFを経てI-Dとして採択
    - https://tools.ietf.org/html/draft-ietf-teep-architecture-00
      - Symantec, ARM, Intercede, Alibaba Group
    - https://tools.ietf.org/html/draft-ietf-teep-opentrustprotocol-01
      - Symantec, Intercede, ARM, Solacia
- IETF102動向
  - 上記2つのドラフトの議論
  - IntelのTEE実装(Intel SGX)の紹介と, SGXを含む形でのTEEPの仕様拡張提案。ARMのTrust Zoneをベースに作られたと思しきTEEPのアーキテクチャと, SGXのアーキテクチャに大きな差異があり, その差異を埋めたい

TEEP (Trusted Environment Provisioning)

- アーキテクチャ と OTrP (Open Trust Protocol)



OTrPは，TAMとTEE間のEnd-to-Endセキュリティメカニズム(JWE/JWS/JWK)ベースのメッセージングプロトコル
（できることは，TEEデバイスの状態取得，SD(Security Domain)の作成・更新・削除，TAのインストール・アップデート・削除）



引用元: https://datatracker.ietf.org/meeting/102/materials/slides-102-teep-open-trust-protocol-otrp-02

TEEP (Trusted Environment Provisioning)

- TEEPにおけるコンポーネント間の認証
  - 各システムコンポーネント(TAM, TEE, SP(Developer))は, それぞれCAに署名された証明書を持つ(PKIベース)

  - TEEは, 予め安全に埋め込まれたデバイスTEE証明書と対応する秘密鍵をもつ これが, TAMによるRemote Attestation に用いられる
    - TAMはトラストアンカーとしてデバイスTEE証明書のCA証明書のリストをもつ

  - TAMも, CAに署名されたTAM証明書をもつ。これが, TAMからのSD(Secure Domain)の作成や, TA(Trusted App)のインストール要求時の認証に用いられる
    - TEEはトラストアンカーとしてTAM証明書のCA証明書リストをもつ

TEEP (Trusted Environment Provisioning)

- TEEPにおけるコンポーネント間の認証
  - キーと証明書のタイプ

Key Entity Name	Location	Issuer	Checked Against	Cardinality
1. TFW key pair and certificate	Device secure storage	FW CA	A white list of FW root CA trusted by TAMs	1 per device
2. TEE key pair and certificate	Device TEE	TEE CA under a root CA	A white list of TEE root CA trusted by TAMs	1 per device
3. TAM key pair and certificate	TAM provider	TAM CA under a root CA	A white list of TAM root CA embedded in TEE	1 or multiple can be used by a TAM
4. SP key pair and certificate	SP	SP signer CA	A SP uses a TAM. TA is signed by a SP signer. TEE delegates trust of TA to TAM. SP signer is associated with a SD as the owner.	1 or multiple can be used by a TAM

TEEP (Trusted Environment Provisioning)

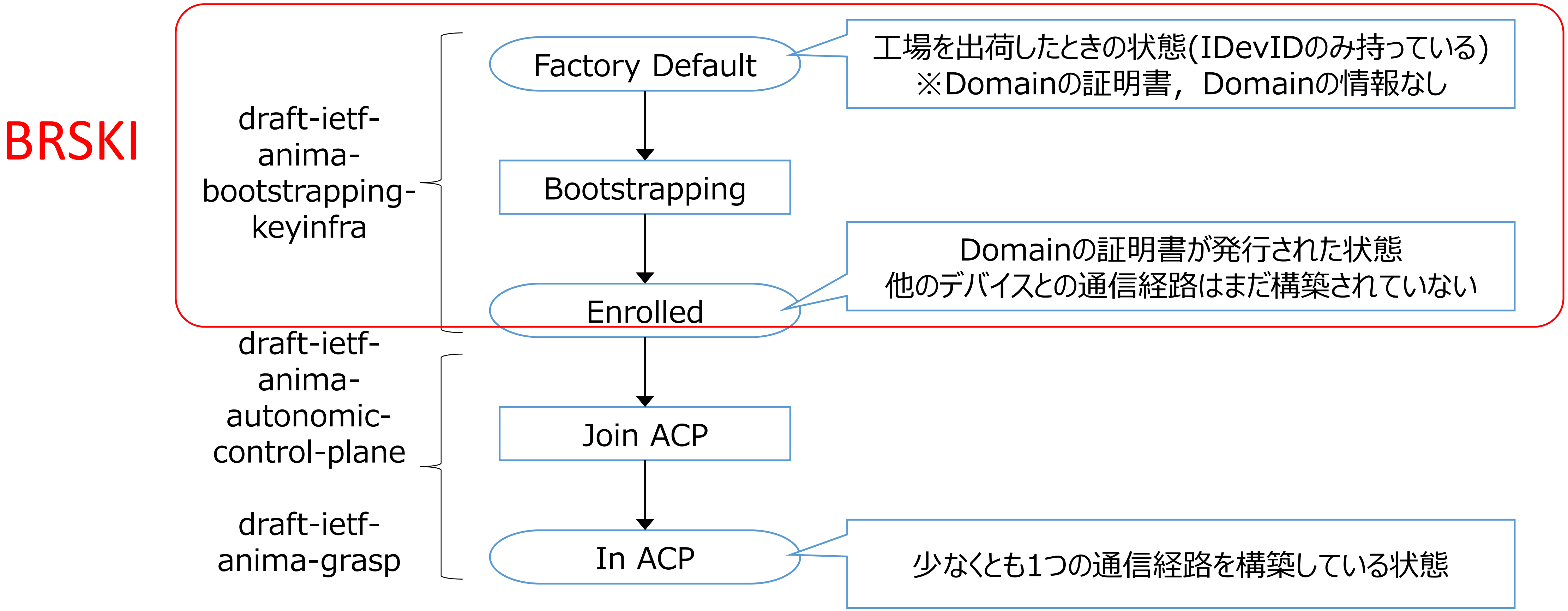
- デバイスセットアップ手順の一例
  1. Prepare Images for devices
     - [TEEベンダ]TEEイメージをdevice OEMに提供
     - [CA] ルートCAホワイトリスト提供
     - [SoC] TFWイメージ提供
  2. Inject key pairs and images to devices
     - [OEM] secure boot key pair 作成
     - [OEM] secure boot keyで署名したTFWイメージとTEEイメージをFlash
  3. Setup attestation key pairs in devices
     - [OEM] secure boot public key とデバイス単位のeFuse keyをFlash
     - [TFW/TEE] ユニークなattestation key ペアを作成し, デバイスの証明書を取得
  4. Setup trust anchors in devices
     - [TFW/TEE] eFuseキーで暗号化した証明書とkeyをストア
     - [TEE vendor or OEM] trustedなCA証明書リストをデバイスにストア

ANIMA (Autonomic Networking Integrated Model and Approach)

- 概要
  - “Autonomic Networking - Definitions and Design Goals“(RFC 7575)を実現するプロトコル等を検討
    - Self-Management
      - Self-healing, Self-optimizing, Self-protection
        ＝環境の変化に対応，通信経路上の問題解決，通信経路の最適化，
        攻撃に備える 等の処理を自律的に行う (no configuration)
      - 例えば，Well-defined Software-Defined Networking
    - Secure by Default
      - 同一ネットワーク内のデバイスが，自身の資格(ネットワークに加入していること)を示すために，identity(domain identity) を使う
      - identityとしては，例えば，ネットワーク内のCAが発行した証明書等
      - identityは, 近隣ノードの把握・ネットワークの境界の判断・ネットワーク内のセキュアな通信を実現に仕様
- IETF102 動向
  - GRASP APIの提案
  - ACP, BRSKI等主要ドラフトのアップデート議論

ANIMA: 各主要ドラフトの位置づけ

- デバイスがAutonomic Networkを作成するまでの状態の変化と，対応するANIMAのドラフト

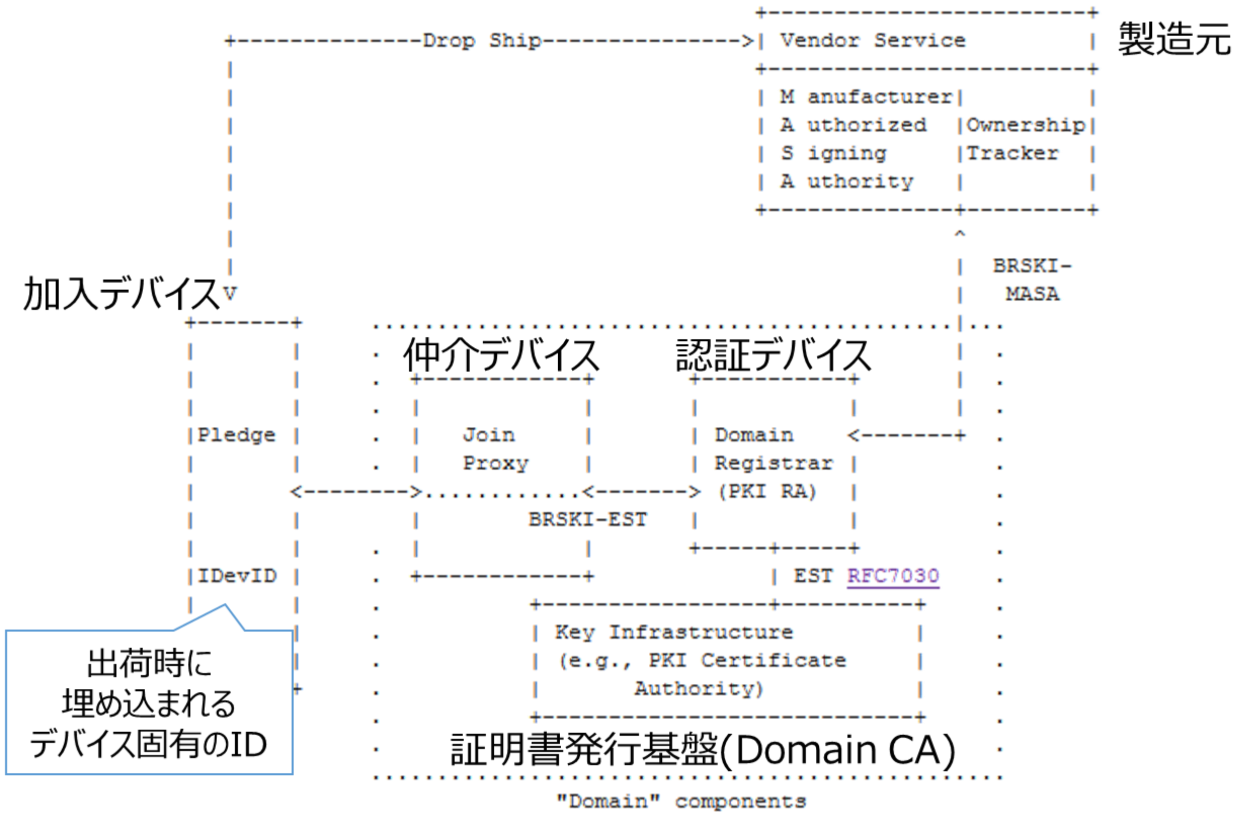


BRSKI（Bootstrapping Remote Secure Key Infrastructures）

- デバイス(Pledge)をDomainに入れるためのプロトコル
  - 工場出荷状態のデバイスのAttestationを行い、Domain内で通信するための証明書を取得するまでをカバー
  - デバイスがDomain内のCAを信頼していない状態でスタートするため、複雑なステップを踏む

1. [製造時] 製造元がデバイスにIDevID(IEEE802.1AR)を書込み

2. デバイスが、認証デバイス(Registrar)と仮TLS接続
3. 仮TLS上で、デバイスが、製造元の署名付きのDomain CA証明書を要求
   - この要求には、IDevIDのプライベートキーでの署名付き（=attestation）
4. 製造元がattestationを検証、Domain CA証明書に署名して返す
5. デバイスは、Domain CA証明書をトラステッドなものとして受け入れ、TLSをはりなおす
6. 正規TLS上で、ESTしてDomain証明書をゲット



ACE (Authn and Authz for Constrained Environment)

- 概要
  - 端的に言えば, OAuthのIoT版。RS(Resource Server)(=デバイス)が Constrainedでプライベートネットワークにつながっているようなケースを想定
  - Constrainedであることに加え, 基本的にPKIが使えない前提
  - AS-RS間, AS-Client間に信頼関係があり, ASを介してRS-Client間の信頼が達成される
  - ClientがアクセストークンとCP一緒に取得するAccess Informationに, RSのRPKやPSKが含まれ, これを使って, RSの認証が行われる

```
+--------+                               +---------------+
|        |---(A)-- Token Request ------->|               |
|        |                               | Authorization |
|        |<--(B)-- Access Token ---------|    Server     |
|        |    + Access Information       |               |
|        |                               +---------------+
|        |                                     ^  |
|        |        Introspection Request  (D)|  |
| Client |             (optional)             |  |
|        |                  Response          |  |(E)
|        |                 (optional)         |  v
|        |                               +--------------+
|        |---(C)-- Token + Request ----->|              |
|        |                               |   Resource   |
|        |<--(F)-- Protected Resource ---|    Server    |
|        |                               |              |
+--------+                               +--------------+
```

Figure 1: Basic Protocol Flow.

ACE (Authn and Authz for Constrained Environment)

- IETF 102 動向
    - EST over Secure CoAP
        - EST (RFC7030)は, HTTPSベースで証明書を発行する規格
            - CSRを投げて、CERTを受け取る
        - これをIoTデバイス向けにCoAPでできるようにしようという提案
        - ユースケース: Secure bootstrapping devices, Distribution of identity
        - ESTクライアント(デバイス側)の認証
            - 事前発行されたクライアント証明書 or
            - 製造過程でベンダがインストールした証明書(ESTサーバは, ベンダのルートCA証明書を信頼する前提)
        - 今回の議論
            - 分オーダーで応答が遅れる場合のエラーハンドリング方法
            - サーバ側でキー生成するケース(証明書とprivate keyが返される)のためのmultipartの導入

ありがとうございました