TTC デジュール及びフォーラム標準に関する国際標準化活動動向調査

IETFにおける IoT機器管理の標準化動向調査

セコム株式会社　IS研究所
髙山献

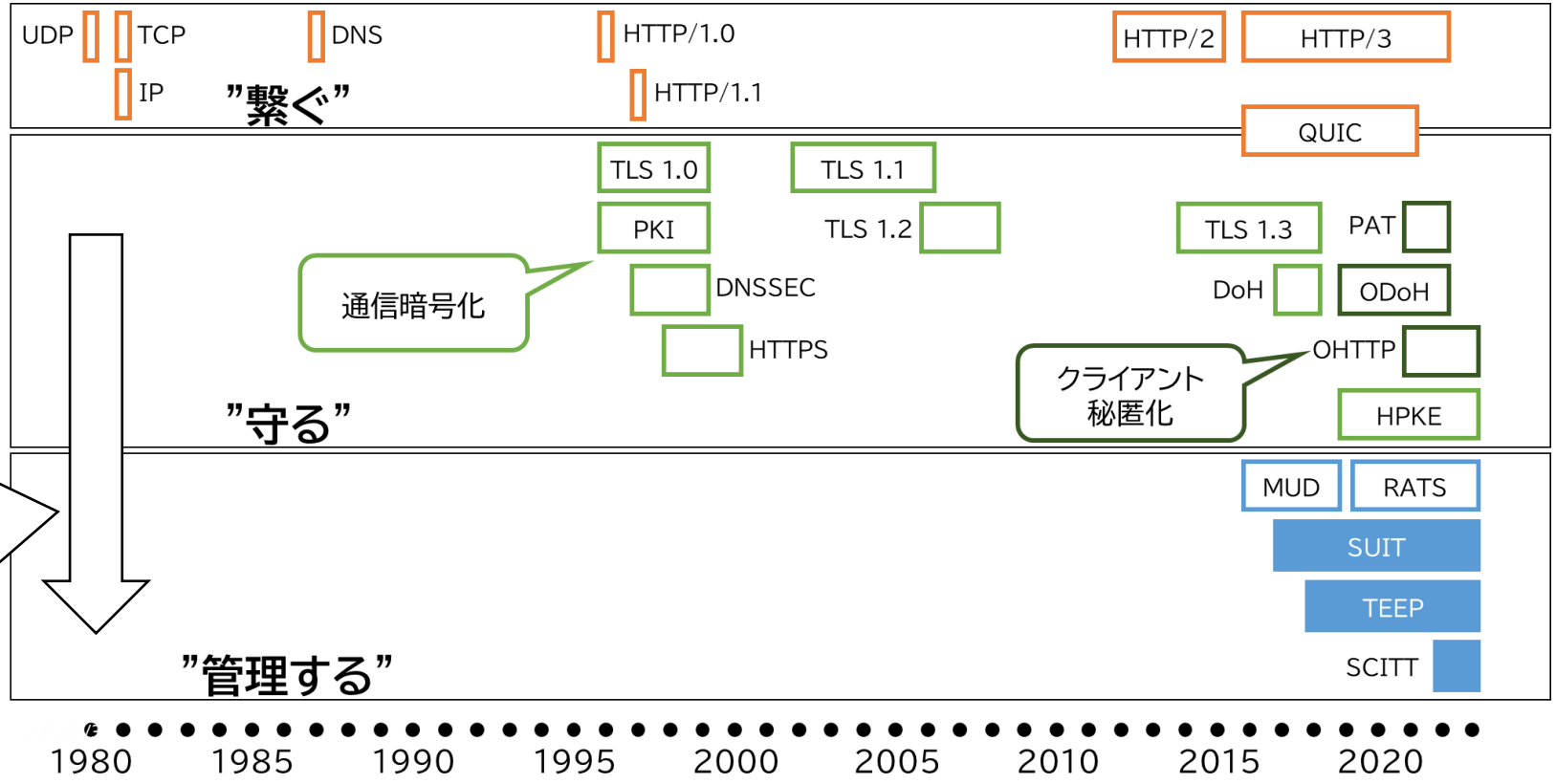
本日お話しすること

- IETFとは
- SCITT Working Groupの標準化動向
- TEEP WG、SUIT WGの標準化動向
- まとめ

IETFとは

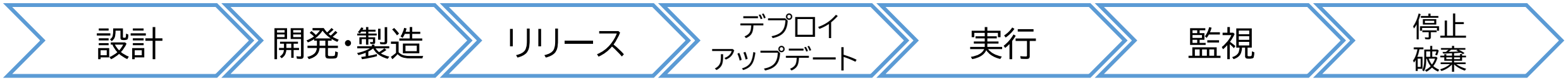
- Internet Engineering Task Force
  - インターネットに関連する技術標準をまとめる団体の一つ
  - 有名な標準(RFC)はTCP、IP、HTTP、HTTPS、QUICなど

近年は通信プロトコル
だけではなく、
暗号技術やその利用方法、
インターネットに接続する
機器の管理や連携
なども標準化対象に



本日ご紹介するIETF WGの取り組み概要

- SCITT（Supply Chain Integrity, Transparency and Trust）
  - 目的：ソフトウェアサプライチェーンの完全性・透明性・信頼性を担保する
  - 方法：既存技術を利用しつつ、検証フローとアーキテクチャを標準化する
- TEEP（Trusted Execution Environment Provisioning）
  - 目的：セキュアな隔離実行環境の初期化・パーソナライズ・アップデート処理を行う
  - 方法：隔離実行環境の検証、各デバイスに合わせたプログラムのデプロイ、プログラム更新の実行結果を伝送するプロトコルを定義する
- SUIT（Software Updates for Internet of Things）
  - 目的：改ざんされたファームウェアがインストール・実行されることを防ぐ
  - 方法：開発者の署名を付加したインストール・実行手順のフォーマットを定義する



IoT機器のライフサイクルと各標準の対応

本日ご紹介するIETF WGの取り組み概要

- SCITT（Supply Chain Integrity, Transparency and Trust）
  - 目的:ソフトウェアサプライチェーンの完全性・透明性・信頼性を担保する
  - 方法:既存技術を利用しつつ、検証フローとアーキテクチャを標準化する
- TEEP（Trusted Execution Environment Provisioning）
  - 目的:セキュアな隔離実行環境の初期化・パーソナライズ・アップデート処理を行う
  - 方法:隔離実行環境の検証、各デバイスに合わせたプログラムのデプロイ、プログラム更新の実行結果を伝送するプロトコルを定義する
- SUIT（Software Updates for Internet of Things）
  - 目的:改ざんされたファームウェアがインストール・実行されることを防ぐ
  - 方法:開発者の署名を付加したインストール・実行手順のフォーマットを定義する

IoT機器のライフサイクルと各標準の対応

IETF SCITT Working Group

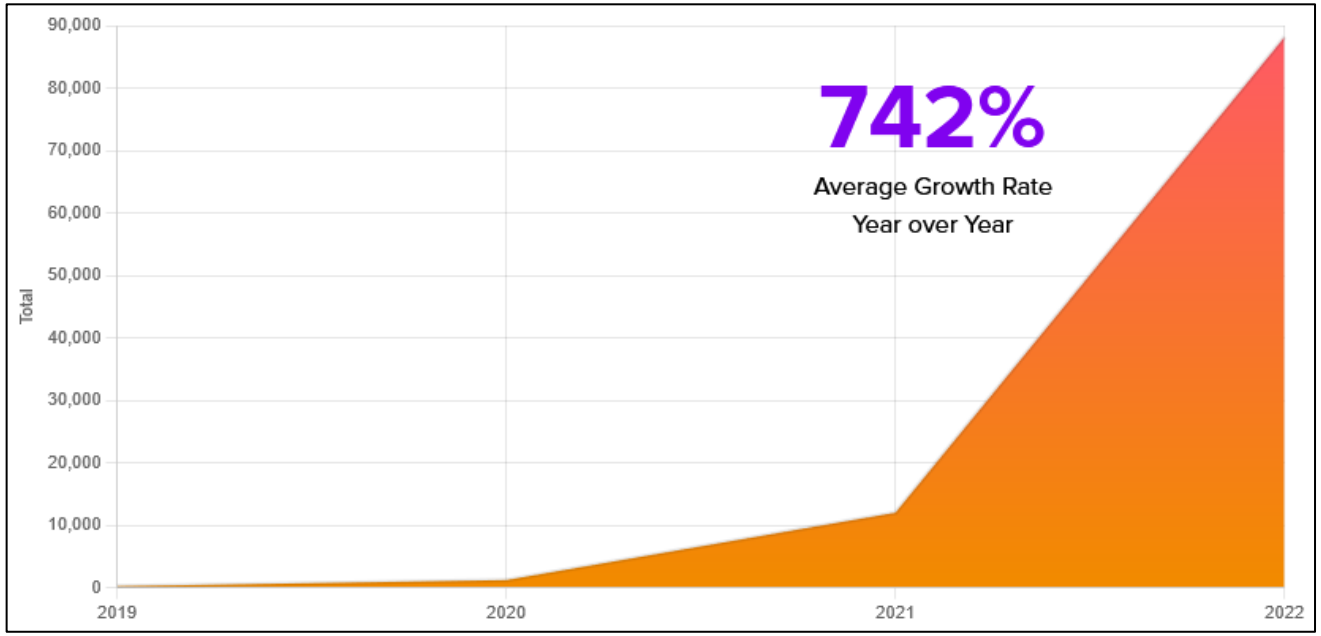
- Supply Chain Integrity, Transparency and Trust
  - 構成要素の出自を明確にする(Transparency＝透明性)
  - それを維持する(Integrity＝完全性)
  - それを検証できるようにする(Trust＝信頼性)
- まずはソフトウェアサプライチェーンを対象に取り組む
  - ユースケースの検討が進む https://datatracker.ietf.org/doc/html/draft-birkholz-scitt-software-use-cases-00

時期	出来事
2022年3月	IETF113 SecDispatchミーティングにてSCITT構想発表、グループ化
2022年6月	SCITT Interimミーティングにて BoF開催、問題設定と方向性の審議開始
2022年7月	IETF114にてSCITT BoF開催
2022年10月	SCITT WG Charterが承認され正式なWGへ
2022年11月	IETF115にてSCITTミーティング開催
2023年1月23日	SCITT 週例Interimミーティングにて Sigstoreを紹介

背景:サプライチェーンに対する攻撃

- 商品やサービスを提供するまでの取引先組織への攻撃
  - 子会社や業務委託先などがランサムウェアの攻撃を受けて機密を暴露される
  - ソフトウェア開発で利用しているOSS(オープンソースソフトウェア)などを介して攻撃されることもある

- ソフトウェアを介した攻撃は毎年数倍の勢いで増加中
  - IoT機器を含めた様々なデバイス内のソフトウェアが攻撃の対象になりうる



Sonatype社が観測したソフトウェアサプライチェーンへの攻撃件数
https://blog.sonatype.com/8th-annual-state-of-the-software-supply-chain-report

背景:アメリカ政府が対策を指示

- アメリカ政府や重要インフラに大規模なサイバー攻撃を受け政権が対応
  - 米国内で一気にサプライチェーン攻撃に対する警戒が強まった

時期	出来事
2020年3-12月	米国SolarWinds社が狙われネットワーク監視製品にバックドアを挿入された アメリカ政府機関を含む100弱の組織内の情報が収集された
2021年4-5月	米国Colonial Pipeline社が狙われランサムウェアによるサイバー攻撃を受けた アメリカ東海岸の燃料供給パイプラインが1週間停止した
2021年5月	米国バイデン大統領がサイバーセキュリティ大統領令 防衛・エネルギーなどで使用する重要製品の対策強化を命令
2021年7月	米国NTIAがソフトウェアに関してSBOMの必要要件を発表
2022年1月	OSS Security Summit（同年5月にも第2回が開催）
2022年2月	米国商務省と国土安全保障省がソフトウェア開発でのOSS利用や外注に言及
2022年2月	米国防省がOSSを優先採用することを定め、従業員がOSSに貢献することも認める

アメリカでは行政が積極的に動いて対策が続けられている

背景:日本でも進むOSS利用と対策

- 行政・民間企業のOSS依存が進む中、利活用と対策を模索している

時期	出来事
2020年3月	東京都が新型コロナウイルス感染症対策サイトのソースコードを公開
2020年9月	厚生労働省が接触確認アプリCOCOAのソースコードを公開
2021年4月	経済産業省がOSSの利活用及びそのセキュリティ確保に向けた管理手法に関する事例集を各企業からヒアリングして作成
2022年2月	公正取引委員会が競争政策上の検討事項として、オープンな仕様の設計や情報システムのオープンソース化を挙げる
2022年5月	経済安全保障推進法成立、サプライチェーン強靭化のための支援も盛り込まれる
2022年6月	デジタル庁情報システム調達改革検討会にて、ベンダーロックイン予防のためにオープンソースソフトウェアの活用やオープンソース化・官公庁内での共有を議論
2022年8月	OSS Security Summit Japan開催（主催：Linux Foundation、OpenSSF）
2023年1月	IPA情報処理推進機構が情報セキュリティ脅威の組織部門2位に「サプライチェーンの弱点を悪用した攻撃」に挙げる

日本でも各省庁がOSSの利用やOSS化などについて指針を示している

SCITTで目指していること

- ソフトウェアサプライチェーン関連の署名・検証フローの標準化
  - 必要に応じてIETFが標準化してきた技術を再利用する
    - COSEやRATSなど
  - 外部の標準化組織と連携する
    - OpenSSF (Open Software Security Foundation)、W3C、ISO、Trusted Computing Groupなど

➢IoT機器のプログラム管理にも利用可能な仕組み

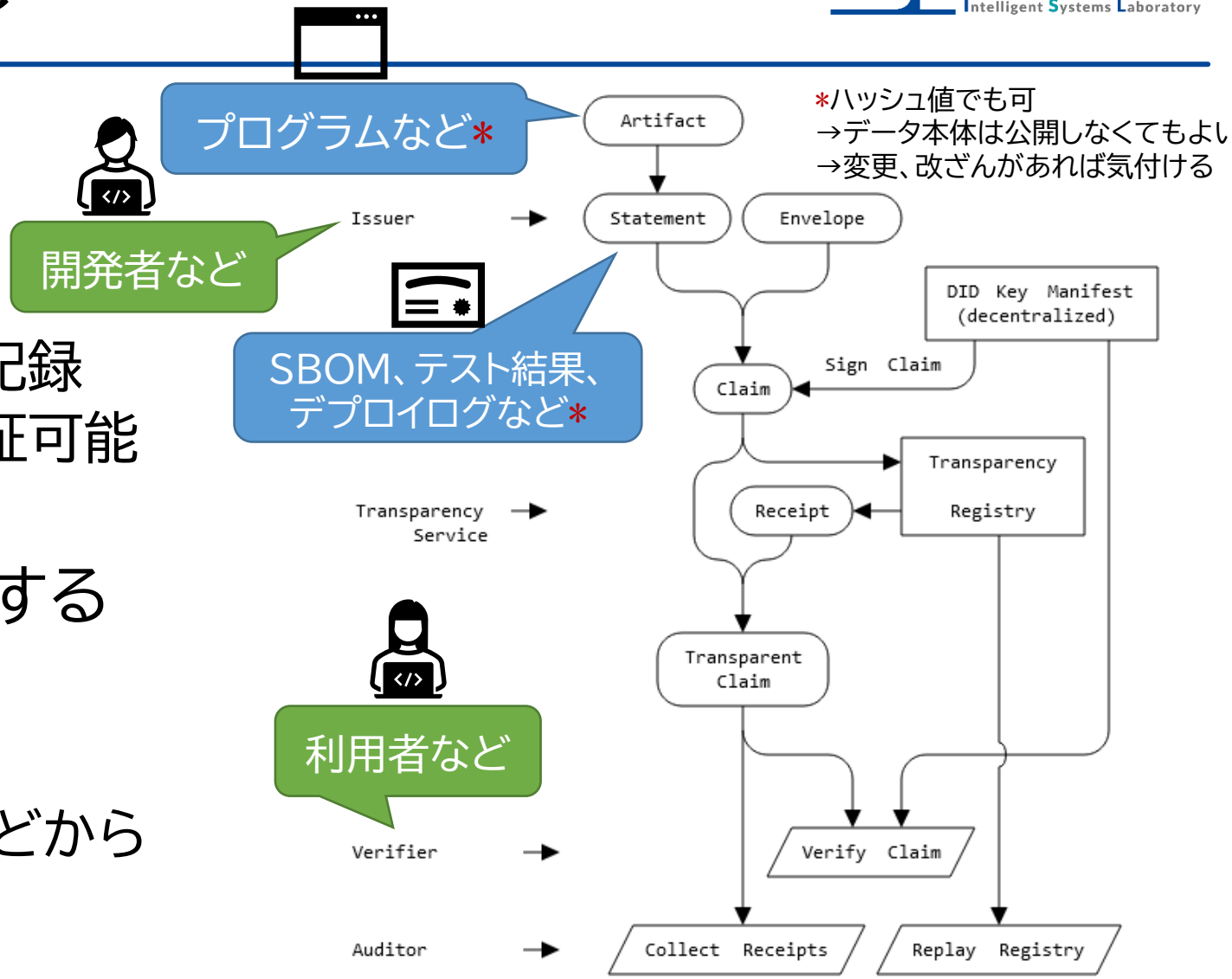
- 利用するプログラムがいつ、誰に作られたかの記録が残る
- 採用前、攻撃や脆弱性が発覚した時に、診断に用いることができる
  - 誰によって開発・頒布された、何のプログラムを利用しているか

SCITT Architecture

- 各者の役割とフローを定義
  - Issuerは作成したプログラムとそのSBOMなどを提出
  - Transparency Serviceが記録
  - Verifierは証拠の正しさを検証可能

➢OSSの出自などを記録し検証を可能とする場として適する

➢ログはオープンであるため用途が限られる場合も
  - 署名者、ハッシュ値、時期・量などから部外者が何らかの情報を得られるかもしれない



https://datatracker.ietf.org/doc/html/draft-ietf-scitt-architecture

既存のサプライチェーン攻撃対策例

- SBOM（Software Bill of Materials）
  - 目的：開発時に、ファームウェア内へ意図しないプログラムが混入することを防ぐ
  - 方法：依存するプログラムの一覧を表示し確認する
- Sigstoreプロジェクト
  - 目的：開発時に、依存するプログラム等に改ざんがないことを検証可能にする
  - 方法：依存するOSSプログラムのハッシュ値と開発者の署名を公開ログに残す

- Secure Boot
  - 目的：改ざんされたファームウェアが実行されることを防ぐ
  - 方法：ブート時に、ファームウェアのバイナリイメージに対する署名を検証する（こちらはハードウェア支援機能やSUIT、TEEPなどが関連します）

SBOM

- 主要なデータフォーマットはSPDX（Software Package Data Exchange）
  - 依存しているプログラム名、バージョンなどの一覧をSPDX Viewerで表示可能
  - SPDX 2.2.1はISO/IEC 5962:2021として標準化された
  - もともとはOSSライセンス遵守を主目的に作られた

Apacheライセンス版Mbed TLSの記述例

https://github.com/OpenChain-Project/OpenChain-JWG/blob/master/subgroups/sbom-sg/outcomes/SPDX-Lite/sample/mbedtls/mbedtls-apache.txt

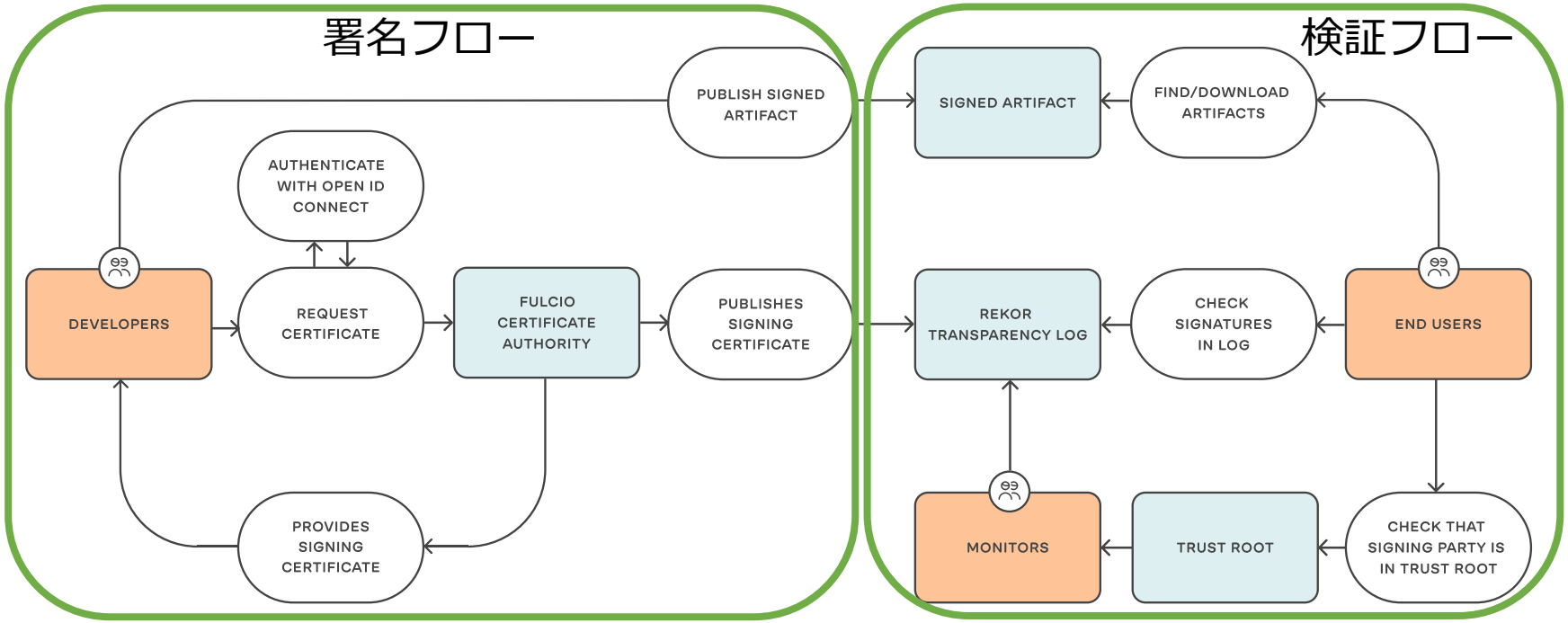
```
SPDXVersion: SPDX-2.1
DataLicense: CC0-1.0
SPDXID: SPDXRef-DOCUMENT
DocumentName: mbedtls-2.16.2
DocumentNamespace: http://spdx.org/spdxdocs/mbedtls-2.16.2-
d92dab8c-6849-42f7-b4ef-843092878b27
Creator: Person: Tomo Dote(fu7mu4@gmail.com)
Creator: Organization: OpenChainProject ()
Created: 2019-01-25T01:01:01Z

PackageName: mbedtls
SPDXID: SPDXRef-1
PackageVersion:   2.16.2
PackageFileName: mbedtls-2.16.2-apache.tgz
PackageDownloadLocation: https://tls.mbed.org/download
FilesAnalyzed: false
PackageHomePage: https://tls.mbed.org/
PackageLicenseConcluded: Apache-2.0
# PackageLicenseInfoFromFiles: Apache-2.0
PackageLicenseDeclared: (Apache-2.0 OR GPL-2.0-only)
```

Sigstore

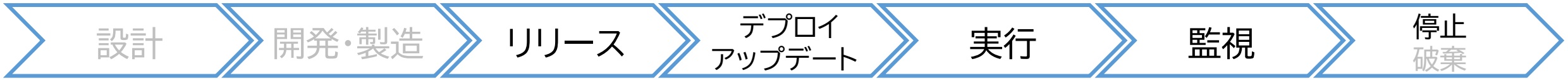
- OSSの頒布イメージに対する署名・検証プロジェクト
  - 誰が、いつ、どこで頒布したものかを誰でも検証できる
  - 自動化されたツールを使って登録できる
  - OSSの利用者は変化があれば検知できる
- SCITTとの関連
  - 電子署名を利用したサプライチェーンのセキュリティ構築
  - SCITTのアーキテクチャと類似
  - Sigstoreは具体的な実装例

https://www.sigstore.dev/how-it-works



本日ご紹介するIETF WGの取り組み概要

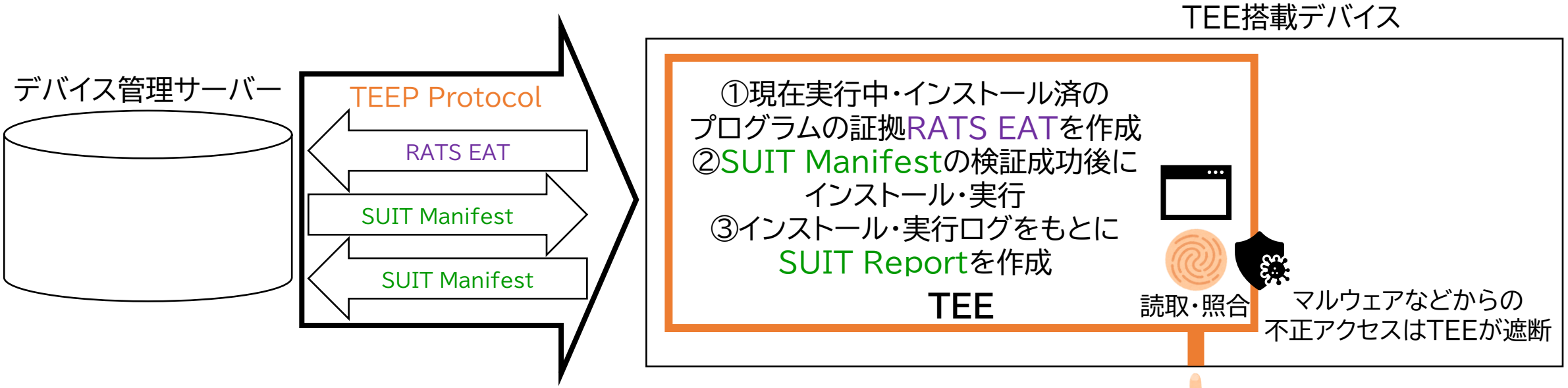
- SCITT（Supply Chain Integrity, Transparency and Trust）
  - 目的：ソフトウェアサプライチェーンの完全性・透明性・信頼性を担保する
  - 方法：既存技術を利用しつつ、検証フローとアーキテクチャを標準化する
- TEEP（Trusted Execution Environment Provisioning）
  - 目的：セキュアな隔離実行環境の初期化・パーソナライズ・アップデート処理を行う
  - 方法：隔離実行環境の検証、各デバイスに合わせたプログラムのデプロイ、プログラム更新の実行結果を伝送するプロトコルを定義する
- SUIT（Software Updates for Internet of Things）
  - 目的：改ざんされたファームウェアがインストール・実行されることを防ぐ
  - 方法：開発者の署名を付加したインストール・実行手順のフォーマットを定義する



IoT機器のライフサイクルと各標準の対応

TEEP進捗

- 実行中のプログラムの証拠を報告する方式の検討が進む
  - 実行するプログラムを差し替える攻撃を検知可能
  - 証拠はRATS WGが策定するデバイスの署名付きEAT（RFC化の準備中）に格納して送信



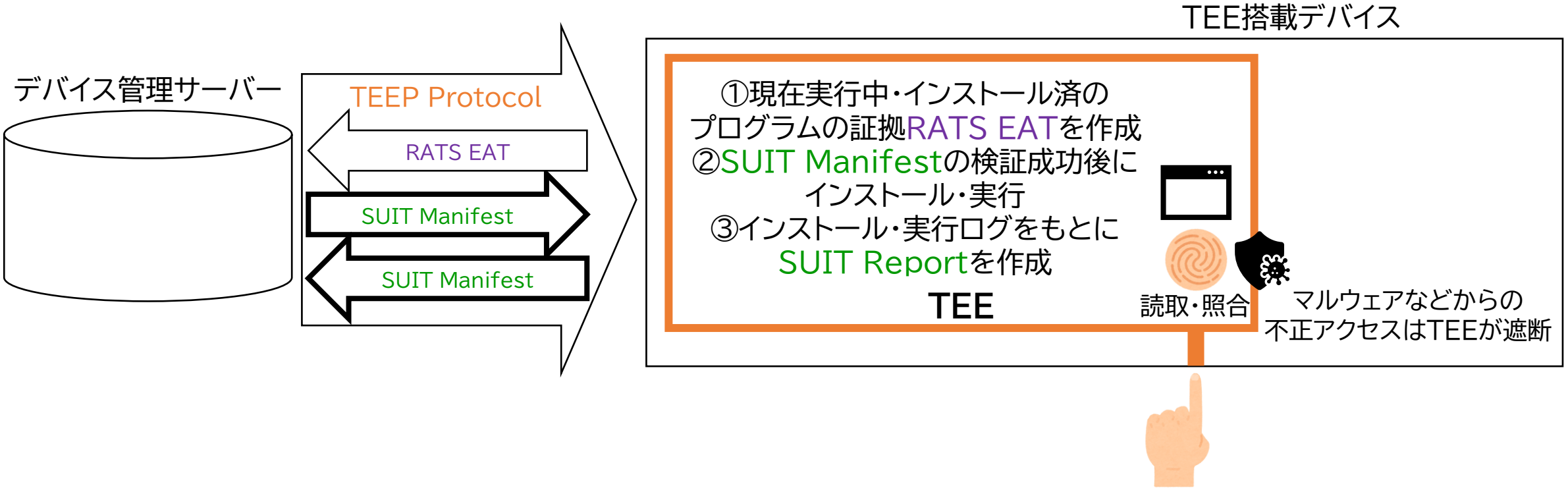
証拠の内容や提出方法の詳細についてGitHub上で活発に議論中

#289, #285, #281, #224, #221, #189, #185, #184, #171, #170

解決済み

SUIT進捗

- SUIT Manifestのコア部分がWG内でLast Call
- プログラムの実行結果を格納するSUIT Reportも進行中
  - TEEP WGからの要望も取り込みつつ、実行結果を格納するフォーマットを検討



まとめ

- IETF SCITT
  - ソフトウェア開発時のソフトウェアサプライチェーン攻撃への対策方法を検討
  - 攻撃や脆弱性などが判明した時に影響を受けるIoT機器等の診断が容易に
  - 誰がどのようなフローで署名・検証を行うべきか、アーキテクチャの本質を標準化
- IETF TEEP
  - 高信頼なTEEでの実行を要求するソフトウェアの安全なデプロイ方法を検討
  - 遠隔のデバイス管理者が状況を把握できる通信プロトコルを標準化
- IETF SUIT
  - ソフトウェア頒布時のソフトウェアサプライチェーン攻撃への対策方法を検討
  - IoT機器でも利用できるよう数百kb程度の軽量フォーマットを標準化