IETF104
TEEPおよびハッカソンの報告

National Institute of Advanced Industrial Science and Technology(AIST)

National Institute of
Advanced Industrial Science
and Technology
AIST

Akira Tsukamoto, Kuniyasu Suzaki

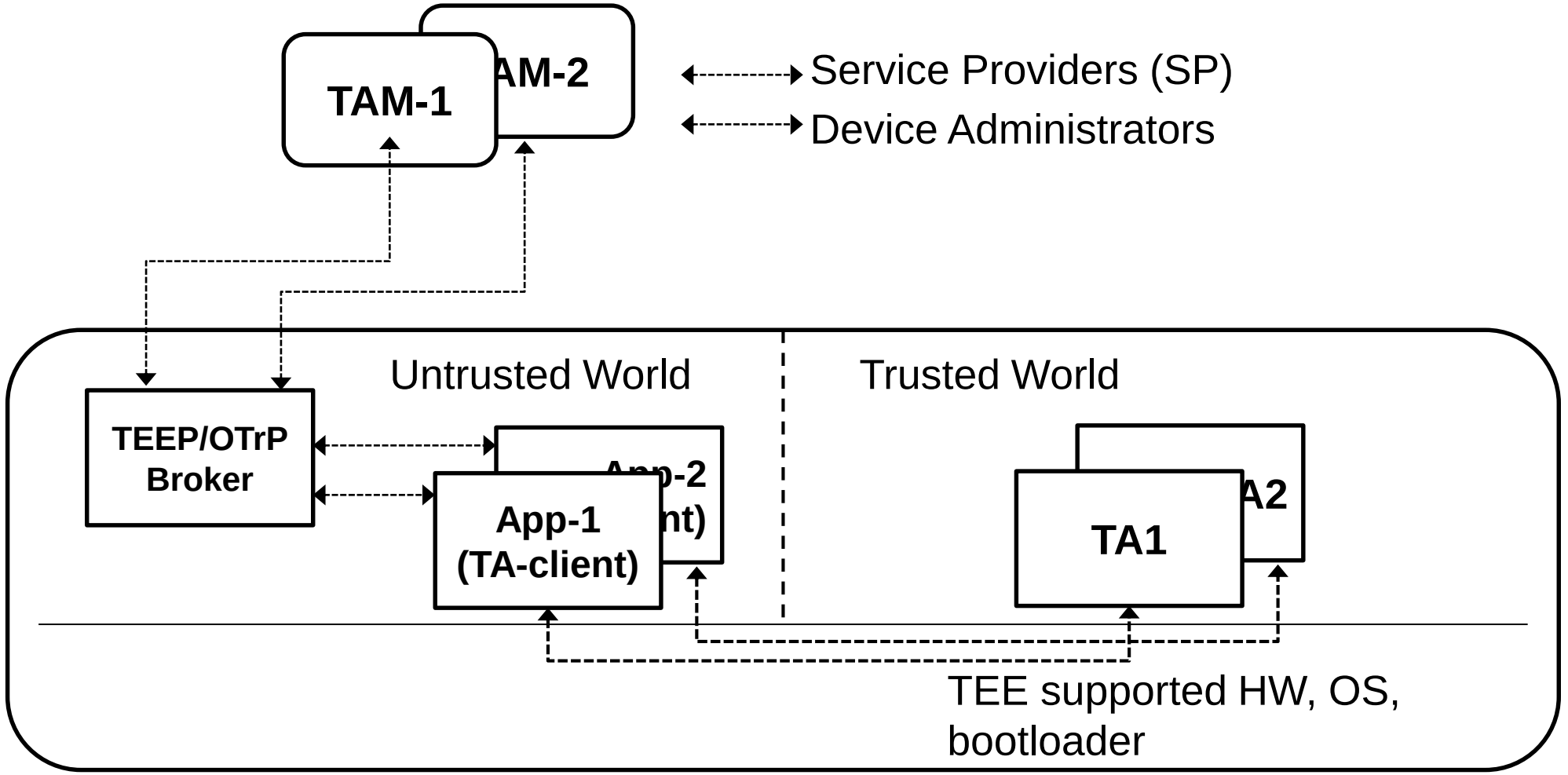
目次

- 総括
- TEEP概要図
- TEEP Hackathon
- TEEP/OTrP セッション
  - TEEPとOTrPの関係の今後の流れ
  - 他の WG の活動との関連性
- その他、SUIT, cbore WG について
- 技術面以外にIETF 104 に参加してみての印象
- まとめ

総括

- TEEP(TEE Provisioning)やSUITなどIETFのいくつかのWorking GroupではTEEをベースとしたセキュリティーのライフサイクル管理に直結する研究開発の議論が行われている。産総研では、NEDOプロジェクトでTEEの研究開発を行っている。初参加であったことから、産総研の意図を紹介し関係者とネットワークを築くことと現在のTEEPやSUITなどのWGの議論の把握に注力した。
- TEEP ではマイクロソフトのDave ThalerのチームはIntel SGX, ARM TrustZoneeの両方で動作するTEE環境であるOpenEnclave を開発していることがわかり、RISC-VのKeystoneとならんで産総研のTEEと直接関係しそうなプロジェクトであることがわかった。また、Hannes(ARM)が我々の研究開発分野に関係するWGに横断的にかかわっていることがわかった。
- 今回の議論にてTEEP WGではIntel SGX, ARM TrustZonee, と並んで RISC-V も含めた を実現する方針となった。

TEEP 概要図


TAM-1
AM-2
Service Providers (SP)
Device Administrators
Untrusted World
Trusted World
TEEP/OTrP
Broker
App-1
(TA-client)
TA1
TEE supported HW, OS,
bootloader

TEEP Hackathon

- 参加者4人、前回の参加者2人から倍に

Team members:
- Dave Thaler
- Akira Tsukamoto
- Kuniyasu Suzaki
- Hannes Tschofenig (co-author)

First timers @ IETF/Hackathon: 2

TEEP Hackathon 成果

- Across 3 types of TEEs (Intel SGX, ARM TrustZone, RISC-V Keystone)
- Participants used Open Enclave SDK branch that supports both SGX and TrustZone
- SGX+TrustZone implementation of OTrP client & server in progress:
- Ported to run over Open Enclave SDK
- Added more of OTrP implementation (more use of JWS & JWE)
- Updated to match latest HTTP transport spec (changes based on MNot feedback), straightforward
- Implemented Trusted Application request mechanism designed (but not implemented) at hackathon 103 but only doc'ed in a github issue

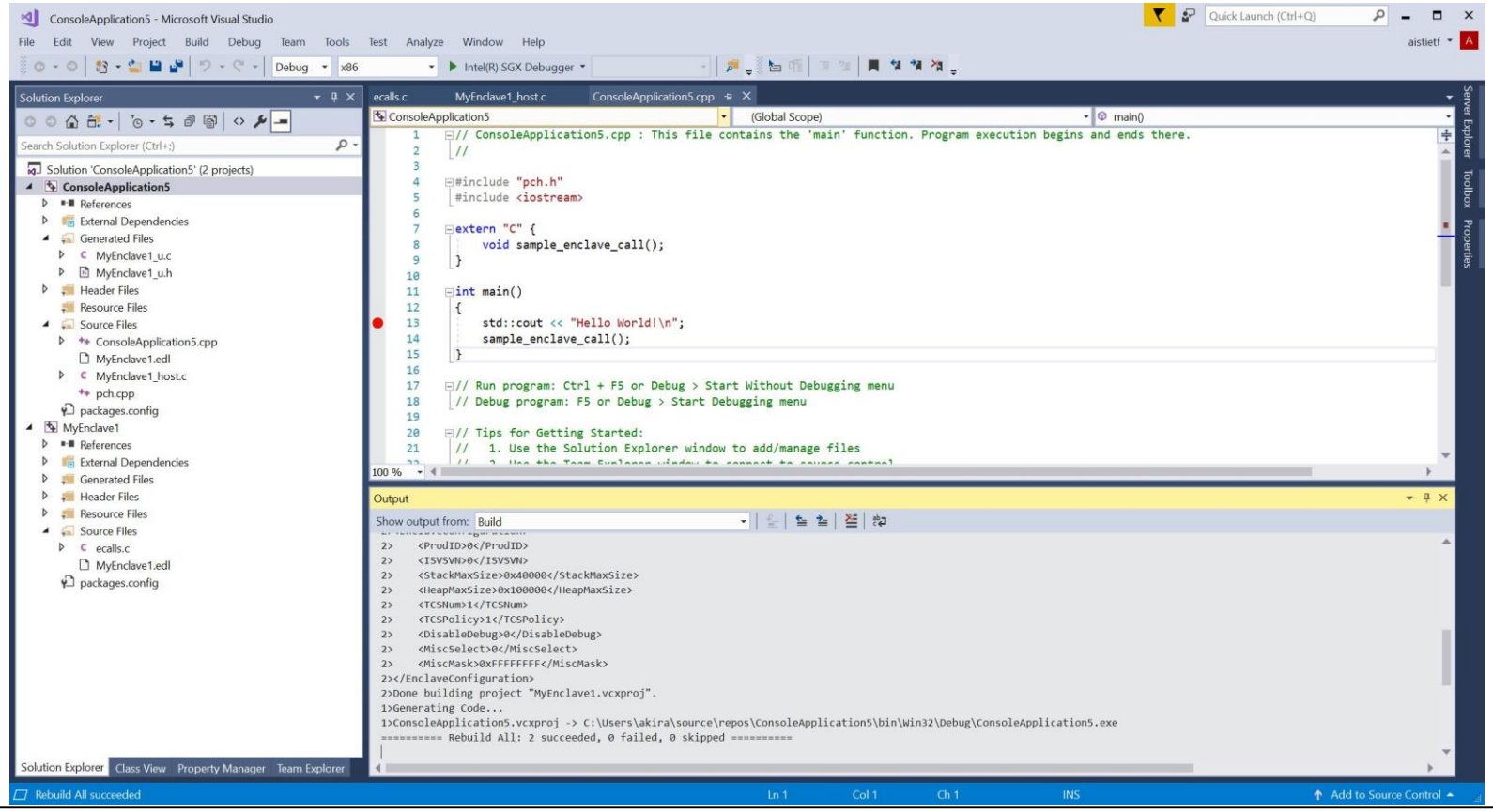
TEEP Hackathon 今後の課題

- Filed Issues: https://github.com/ietf-teep/OTrP
  - –5 new draft issues filed
  - –3 existing issues updated with more info

- Summary of new issues:
  - Relationship between OTrP and attestation (EAT/RATS/etc) needs work (on agenda for this week)
  - Some OTrP fields look redundant with others, opportunity for mismatch
  - OTrP spec uses two slightly different cert chain encoding mechanisms (JWS and custom), complicating code
  - Some OTrP fields (TEE name, TEE version) are underspecified and are interpreted differently by different people

TEEP Hackathon 動作した OpenEnclave

- https://github.com/microsoft/openenclave/tree/feature.new_platforms
- **統合開発環境であるVisual Studio で TEE 開発環境を実現（現在IntelSGXとOP-TEE両対応）**



TEEP Hackathon 動作した OpenEnclave

```
#include "pch.h"
#include <iostream>

extern "C" {
    void sample_enclave_call();
}

int main()
{
    std::cout << "Hello World!\n";
    sample_enclave_call();  ≤1ms elapsed
}

// Run program: Ctrl + F5 or Debug >
// Debug program: F5 or Debug > Star

// Tips for Getting Started:
//   1. Use the Solution Explorer
//   2. Use the Team Explorer wind
//   3. Use the Output window to s
//   4. Use the Error List window
//   5. Go to Project > Add New It
```

```
Hello World!
```

Untrusted APP からの出力

TA からの出力

```
MyEnclave1 | (Global Scope) | ecall_DoWorkInEncla

#include <openenclave/enclave.h>
#include "MyEnclave1_t.h"

void ecall_DoWorkInEnclave(void)
{
    /* Implement your ECALL here. */
    printf("Hello from inside Enclave ");
}

/* Add implementations of any other ECALLs h
```

```
Hello World!
[sgx_create_enclave_exa_ c:¥sgxwindows¥src_rele
ouldn't open file with CreateFile()
[sgx_create_enclave_exa_ c:¥sgxwindows¥src_rele
ouldn't open file with CreateFile()
Hello from inside Enclave
```

TEEP/OTrP セッション での流れ

- TEEP WG では IoT/エッジデバイスで活用可能はセキュリティーアプリのライフサイクル管理をTEEにてセキュリティーを確保する規格の策定を行っている。
- TEEP では TAM, OTrP broker, SP-App, TAなどのコンポーネントがあり、相互を JSONベースのOTrP のフォーマットでやり取りを行う。
- ハッカソンの時にOTrPにおける各フォーマットの各パラメーターの記述に対して、読む人により解釈が違うことが判明。
  - ユースケースの記述があれば人による解釈が違ってしまうことを防止できる
  - 各フォーマットのユースケースについては本会議では明確化できず

TEEP/OTrP セッション での議論と課題

- TAM<->OTrP brokerや、SP<->TA/Enclaveなどは、相互に信頼を担保するために Attestation を行っているが、Attestation の方式は RATS WGの議論を活用してはどうか？
- OTrPにて TA のライフサイクル管理を行う。Install, Update, Deleteに TA のバイナリーのバージョン情報などが必要で、SUIT WG の manifest が使えるのでは？

- TEEP WG では、SUIT WG と RATS WG と連携することが提案される
- OTrP のドラフトに関しては今後詳細を明確化していくことに

その他、SUIT, cbor WG について

- SUIT WG
  - Firmware の情報などを記述する manifest のフォーマットを策定しており、TEEP WG であつかうTAバイナリーの情報記述フォーマットに活用できると個人的にも感じられる
  - OTrP のフォーマットと比較すると SUIT のフォーマットは詳細定義の策定が進んでおりほぼ完成形に近い印象。

- cbor WG
  - SUIT もOTrPも JSONをベースにしており、SUITは JSON記述を cbor方式にてバイナリーとテキスト情報を相互変換している。
  - cbor WG 自体は cbor のパーサーを実装する時にあいまいさが残っている部分をつぶしていく活動を行っていた。議論にされていた内容はニッチな条件の時にどうするかという点が多く、SUIT で使われている定義では問題にならないとおもわれ、現状の cbor で SUIT や OTrPにおける活用では現状のcborで十分とおもわれる。

IETF 104 に参加してみての印象

- 技術面以外で感じたこと
  - イベントの規模と運営
  - 参加者
  - プラハ
  - 2001年2002年ごろの IETF RFC との比較
  - TEEP/OTrPやSUITを日本でうまく普及させるための課題

まとめ

- TEEP/OTrP関係者と相談でき、TEEP WGでは Intel SGX, ARM TrustZone, RISC-Vすべてを対応することに。TEEPが特定のCPUに依存しない形になることは、産総研のTEEの研究開発の成功に必須事項である。

- TEEP WGの方向性が理解できたことで、産総研の研究開発においてTEEP WGと協業するため課題が明確化した。

- IETF に参加して直接キーマンと顔を合わせた議論による進捗は大きなメリット。

- この成果は、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務「高効率・高速処理を可能とするAIチップ・次世代コンピューティングの技術開発/革新的AIエッジコンピューティング技術の開発/セキュアオープンアーキテクチャ基盤技術とそのAIエッジ応用研究開発」の結果得られたものです。