トラストを確立する技術の概要

どのような技術がなぜ作られてきたのか

セコム株式会社IS研究所
宮澤慎一

PKI and Trust Days 2021
2021年4月15日

Disclaimer

- 本日の講演は文献やインターネット上の情報をもとに歴史を考察したものです。
- 細心の注意を払って調査しましたが事実と異なる点もあるかもしれませんが、ご了承ください。
- 社名や製品名も出てきますので、もし問題がありましたら後日修正いたしますので、ご指摘いただければと思います。

本講演の目的

- この後に続く講演のための
  - 基礎知識
  - 歴史背景
  - を知っていただきたい
- 温故知新の技術にもっと光を当てたい
  - 聴衆の皆様の中には、今日紹介する技術の中の人（中だった人）がいるかと思います。
  - 間違い、補足、批判、思い出話がある場合は、Twitter等でご指摘頂くと幸いです。
- 今回のPKI & Trust Days2021で紹介される技術がもっとSNSで話題になると嬉しいです。

本日の流れ

- トラストの技術を活用した具体例：DRM
- 歴史
  - ICカード（Smart Card）の歴史
  - モバイルの歴史
  - PC/サーバの歴史
- Intel SGXの特殊性
- まとめ

本講演でのTrust

Trust ＝想定したプログラム（データ）だけが動く

Trustを確立する技術

- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

本日の流れ

- **トラストの技術を活用した具体例：DRM**
- 歴史
  - ICカード（Smart Card）の歴史
  - モバイルの歴史
  - PC/サーバの歴史
- Intel SGXの特殊性
- まとめ

本講演でのTrust

Trust ＝想定したプログラム（データ）だけが動く

Trustを確立する技術

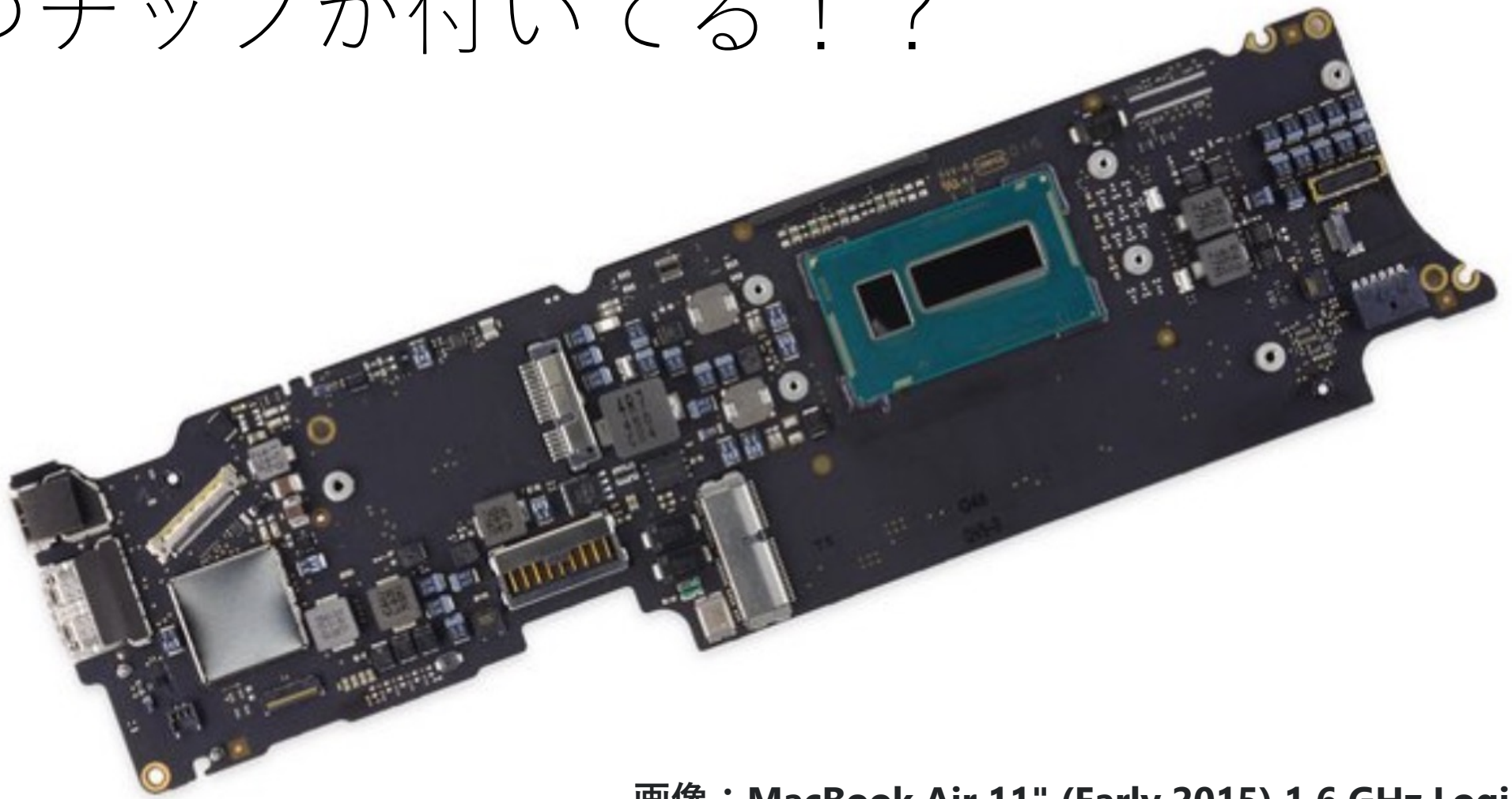
- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

DRM (Digital Rights Management) とは

- 有料の映画、音楽、アプリケーションなどについて、お金を払った人だけが視聴・利用できるようにさせる技術。
- 違法コピー、海賊版を許さない技術
- 有料コンテンツだけでなく、社内文書などの利用制限など適用が期待されている。
- メディアの作成、配信、配布、再生装置などの会社でコンソーシアムを作ることも

- 課題：
  自分の作ったバイナリデータが、他人の機器へ入った後も制御したい

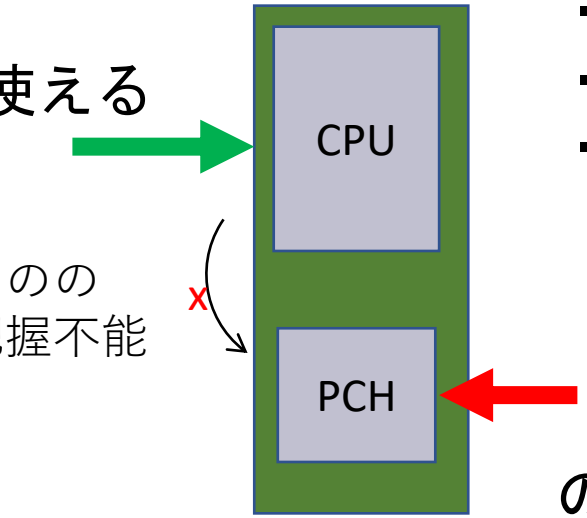
IntelのCPUをよく見ると
２つチップが付いてる！？



画像：MacBook Air 11" (Early 2015) 1.6 GHz Logic Board
https://jp.ifixit.com/Store/Mac/MacBook-Air-11-Inch-Early-2015-1-6-GHz-Logic-Board/IF108-058?o=1

ユーザーが制御できないチップ（PCH）
ユーザーが制御できるチップ（CPU）

PC購入者が自由に使える
（Win10/Linux/.....）



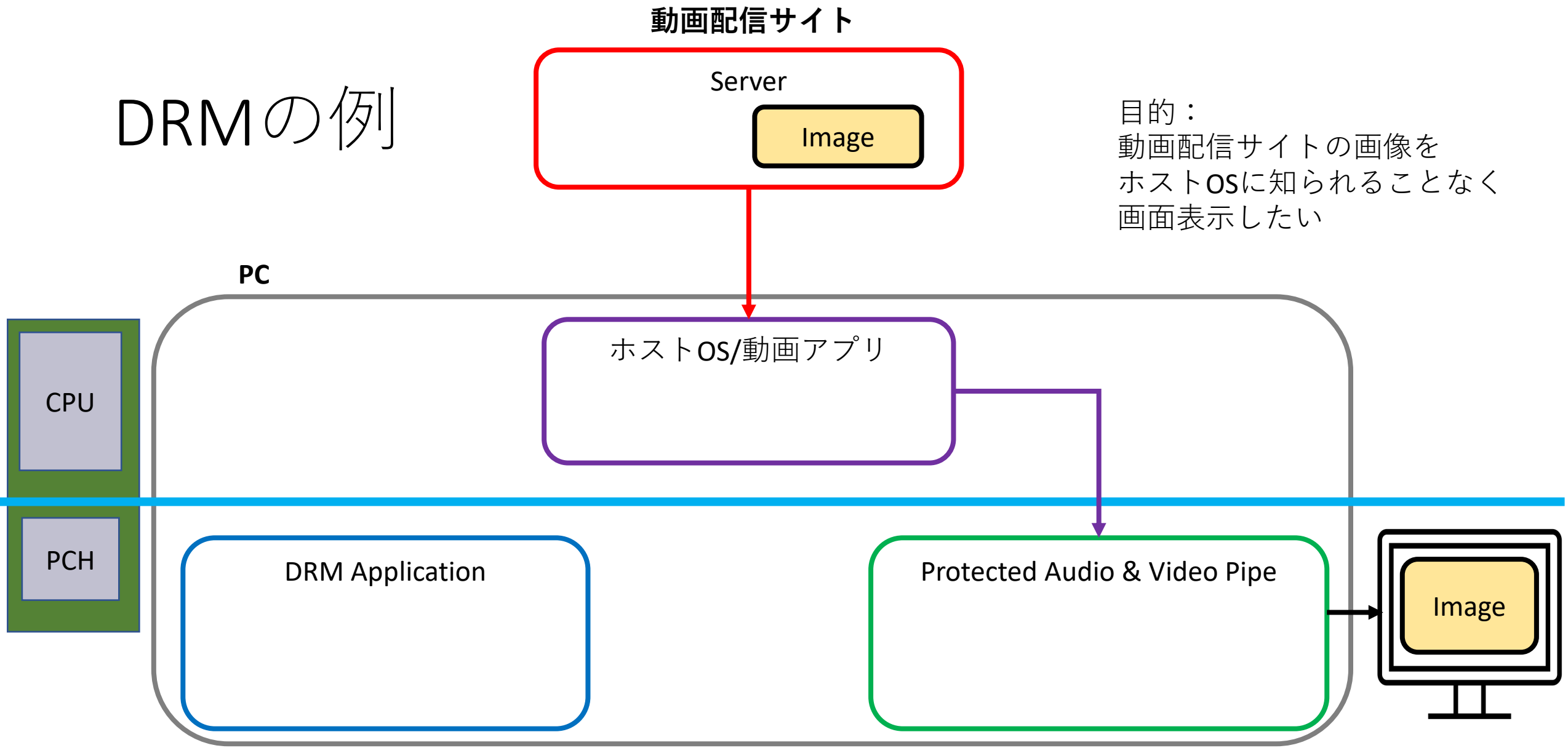
通信できるものの
処理内容の把握不能

・マザーボードの管理やセキュリに関する処理を担当
・Intel
・Intel OEM
  ・マザーボード会社
  ・PC/サーバ販売会社
  ・OS/DRMシステム会社
  ・等々

のみのアプリ実行や設定が有効。
PC/サーバ購入者は利用不可

専用I/Oとの接続
例：暗号化HDMI (HDCP) との直結

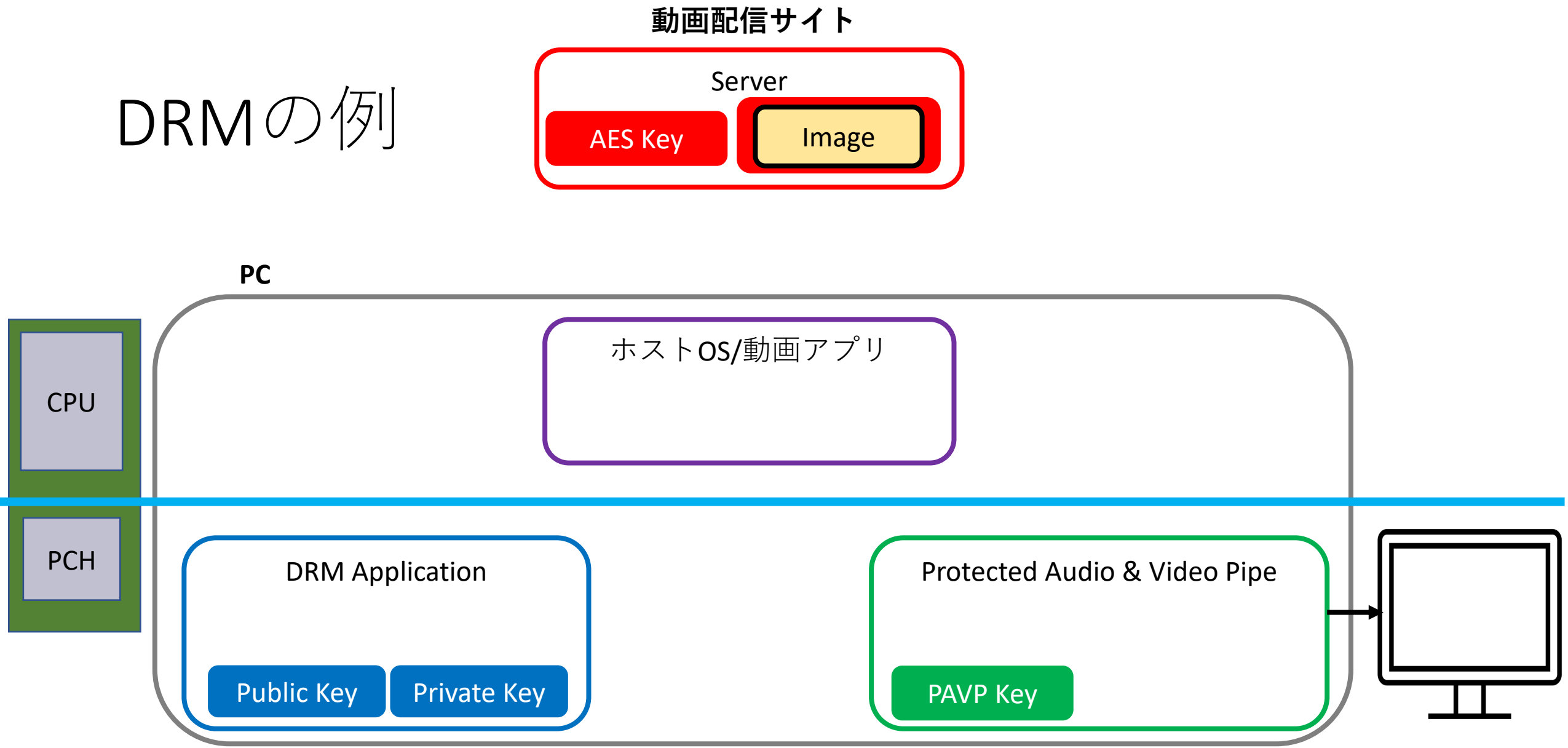
DRMの例



目的：
動画配信サイトの画像を
ホストOSに知られることなく
画面表示したい

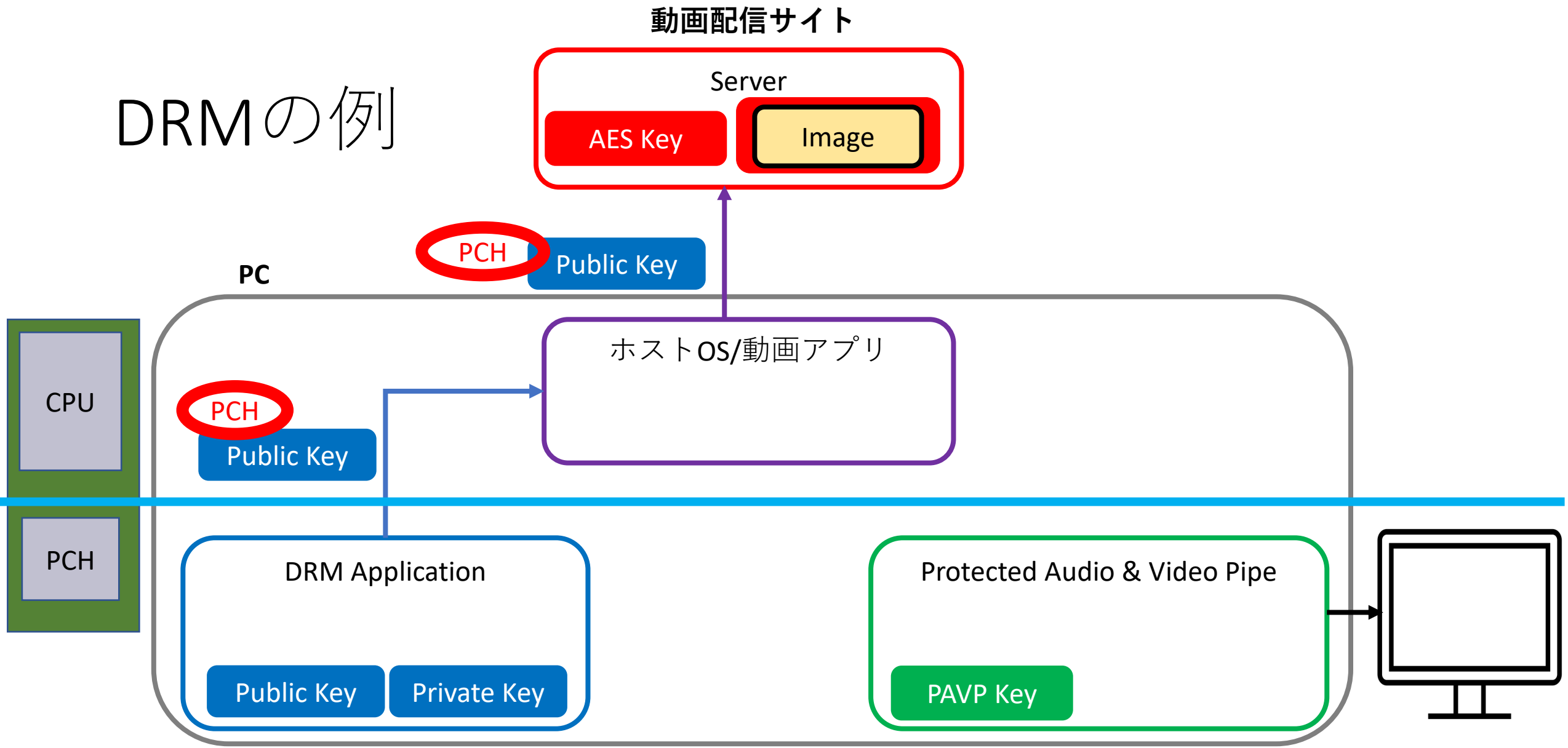
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



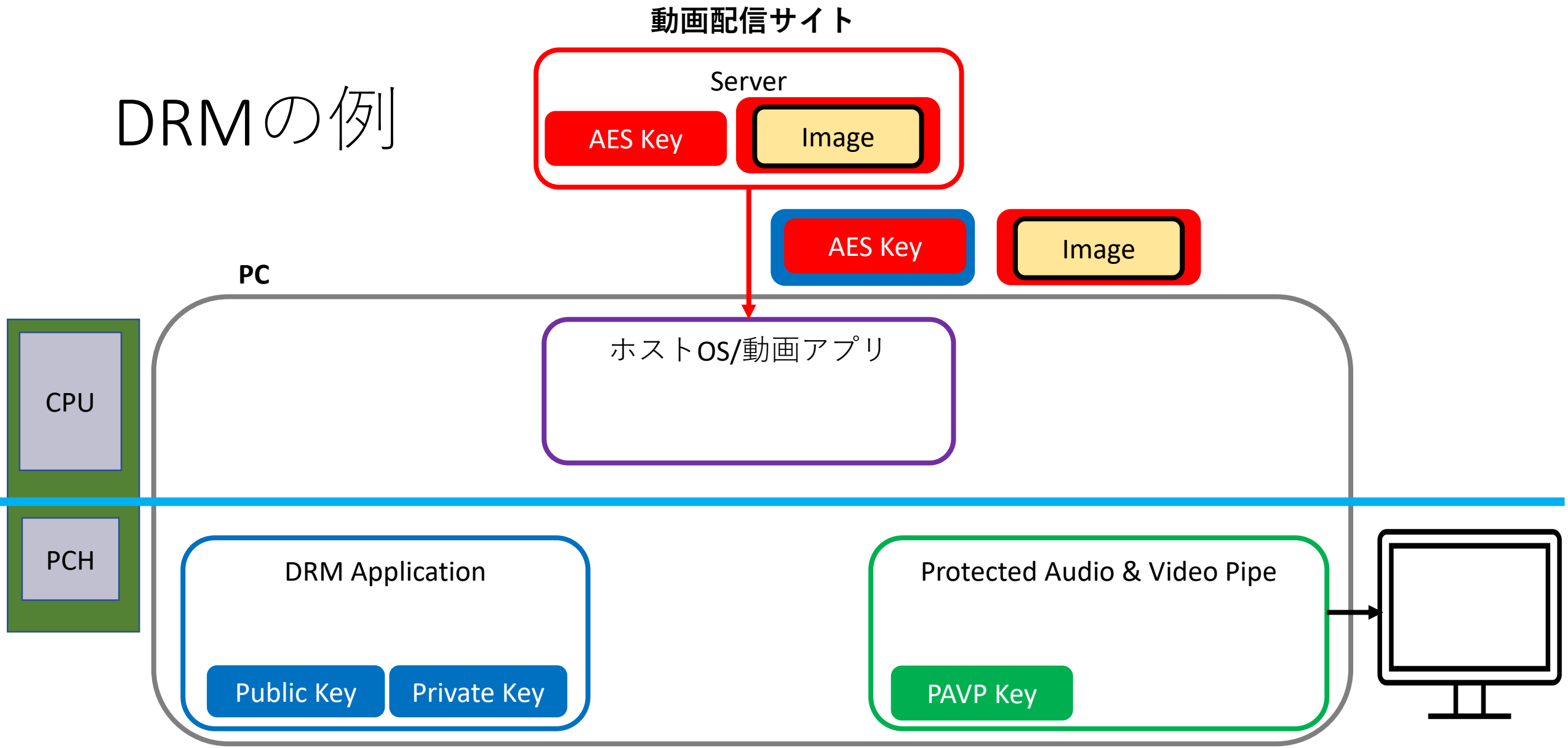
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



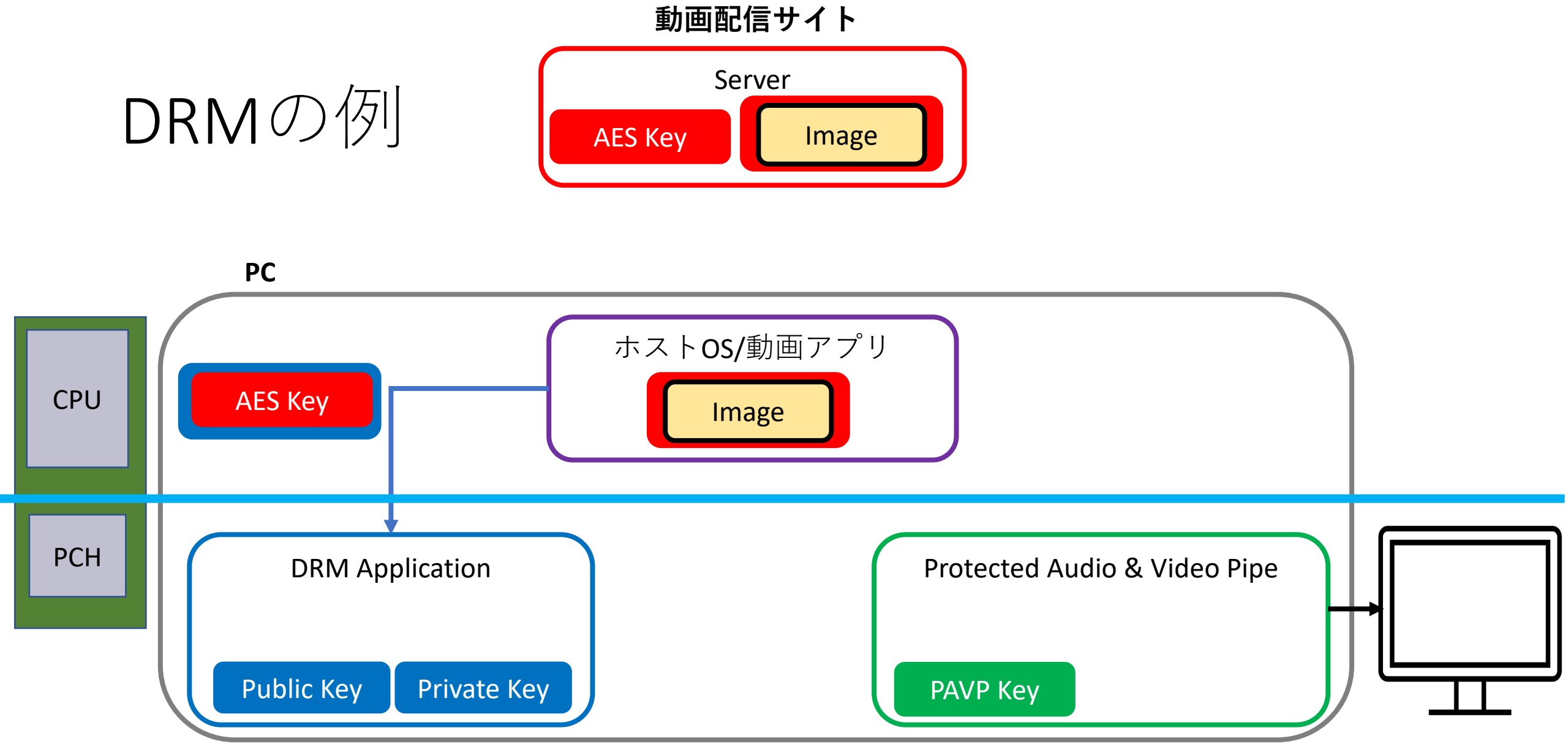
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



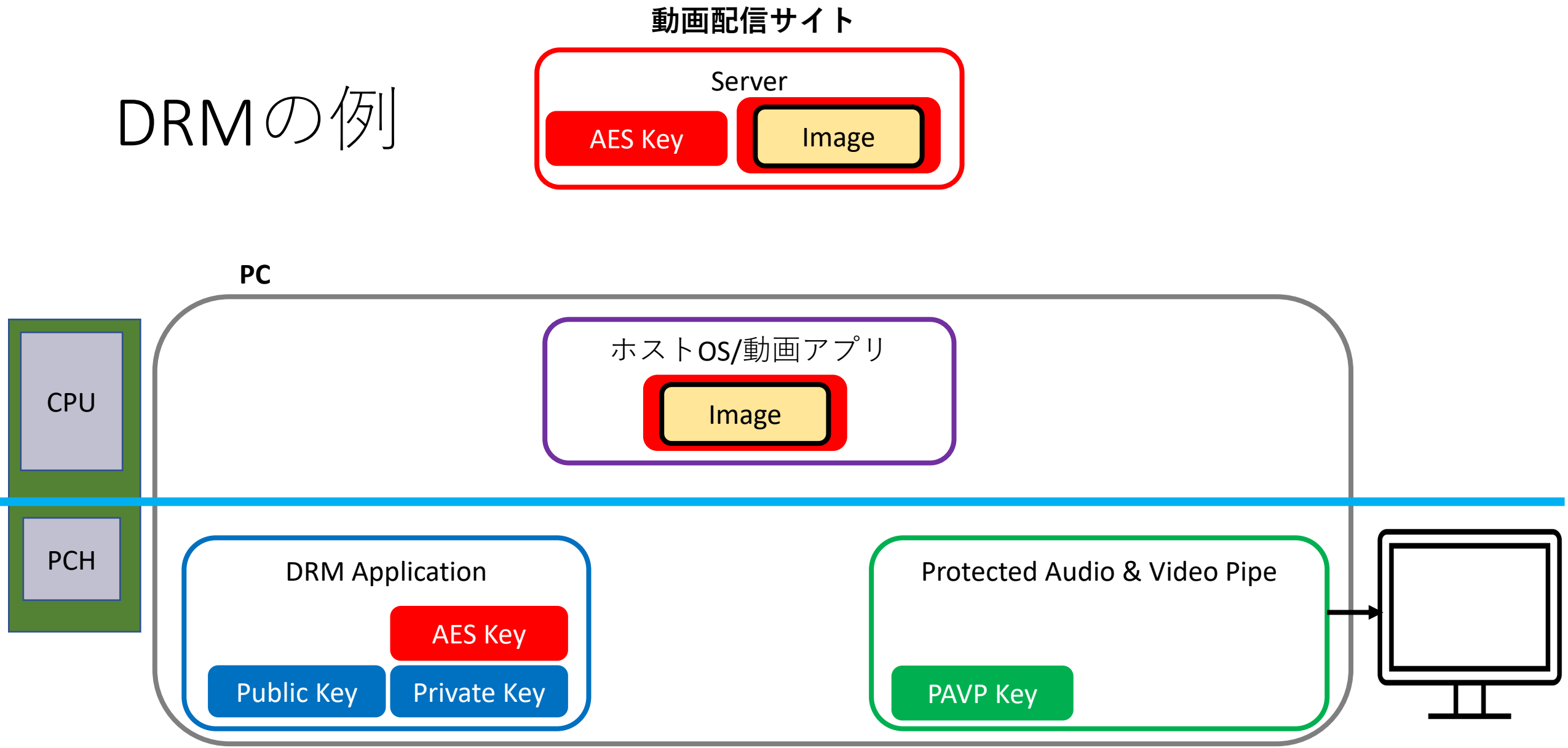
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



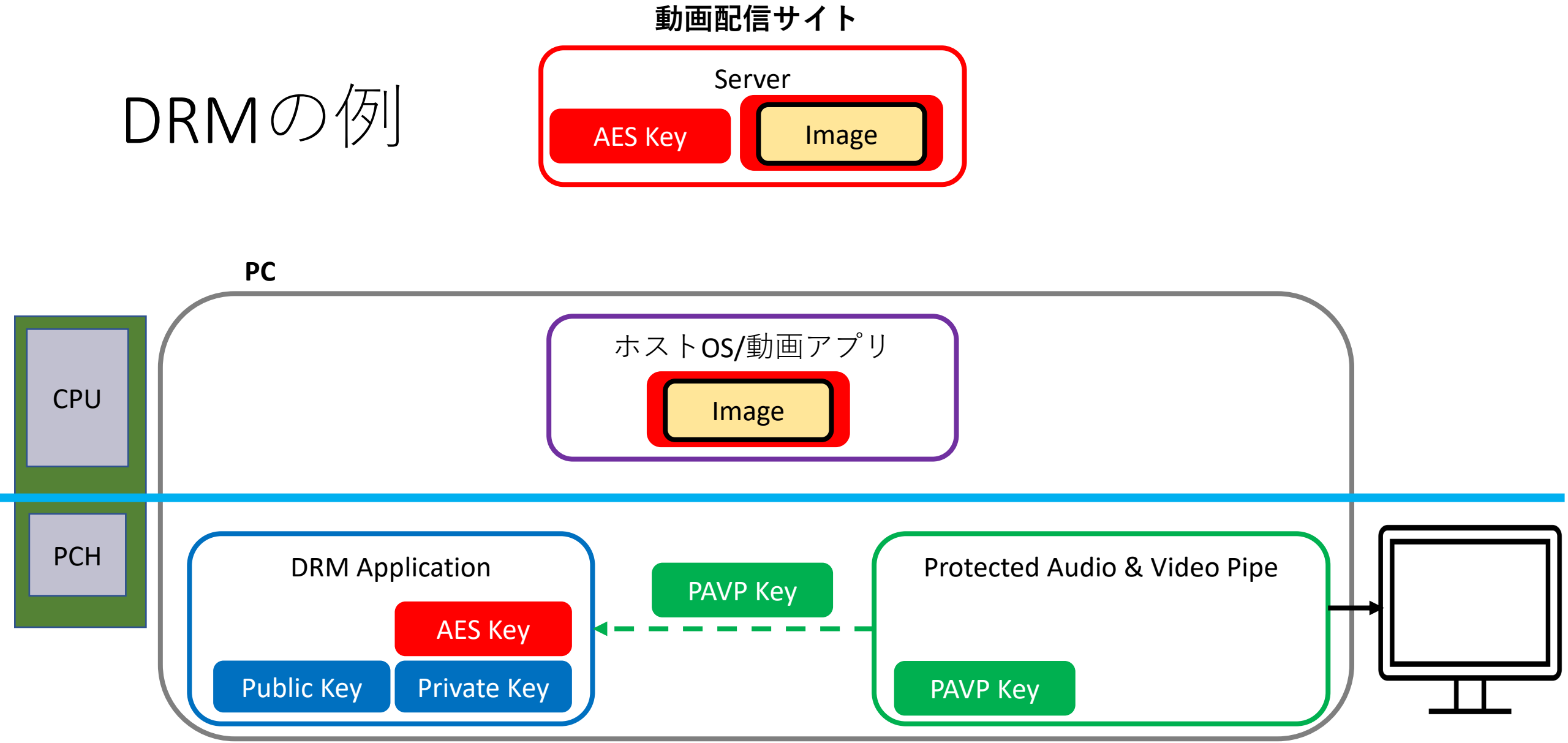
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



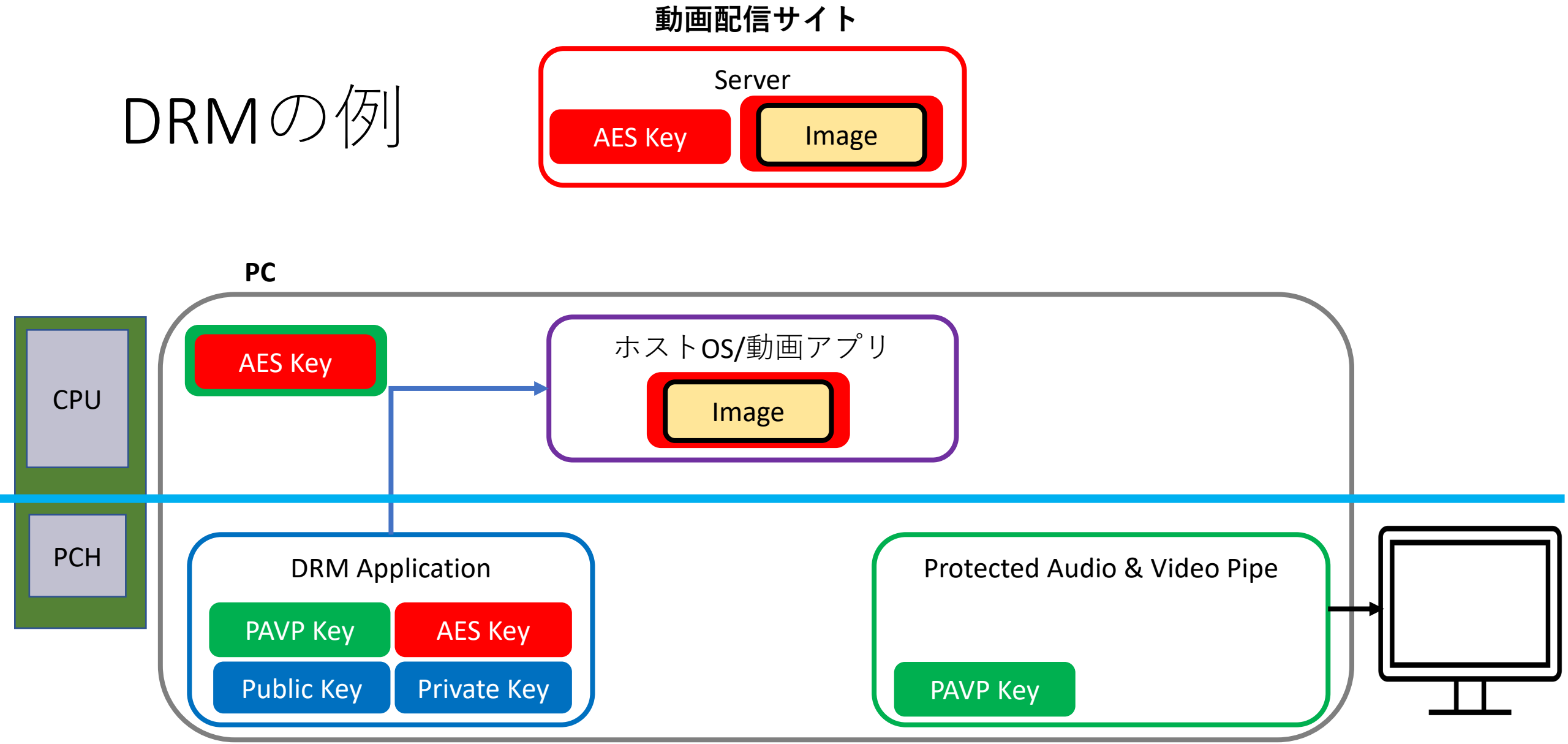
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



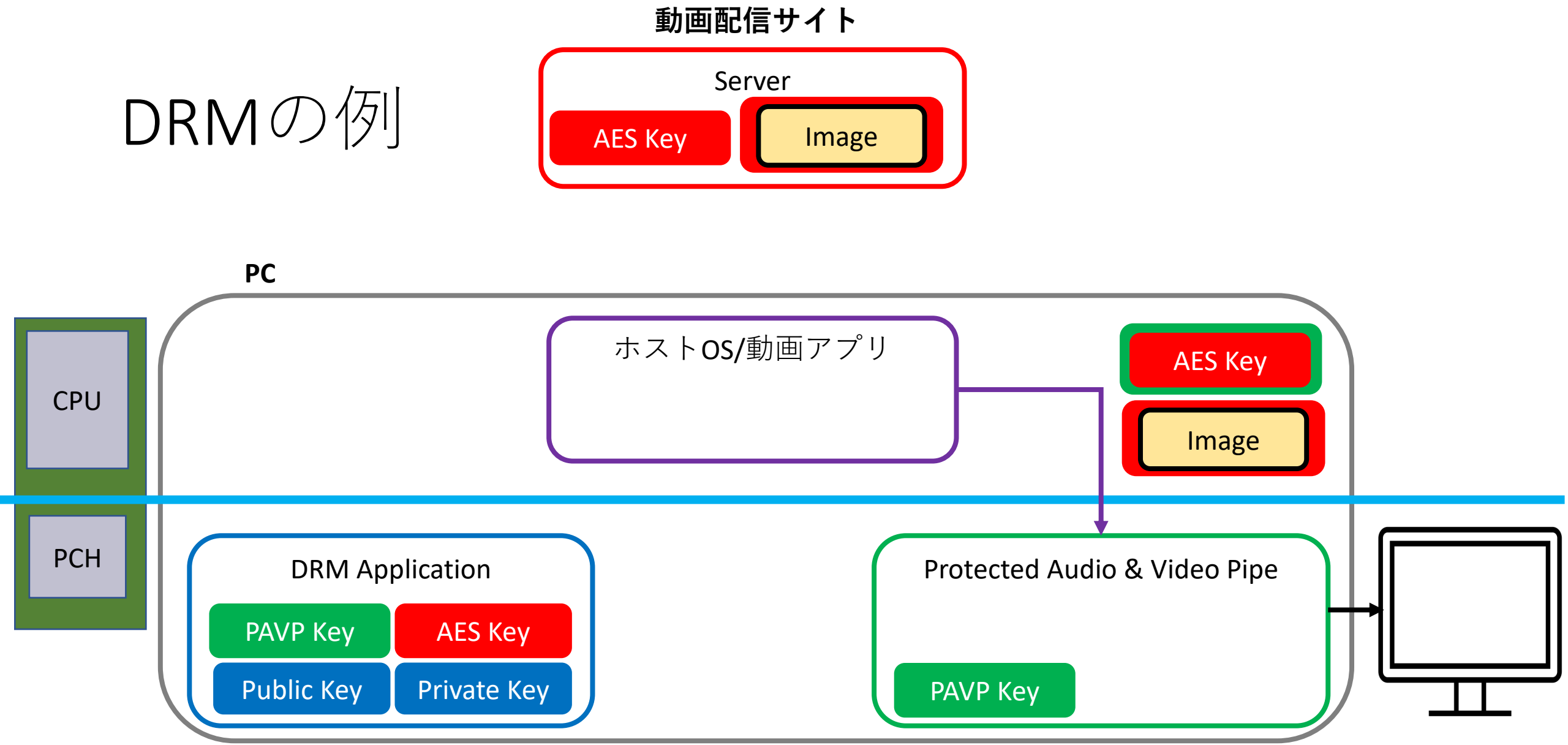
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



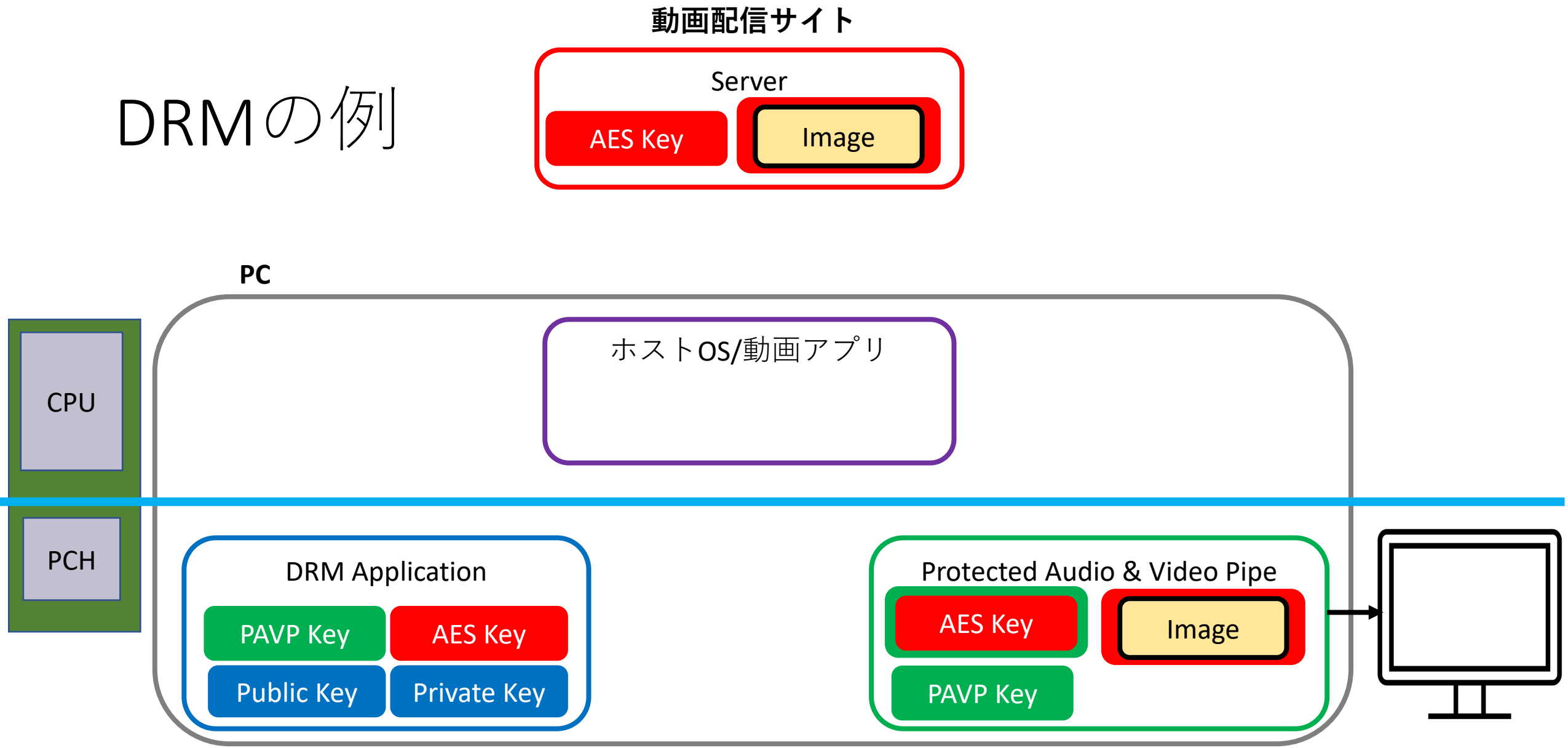
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



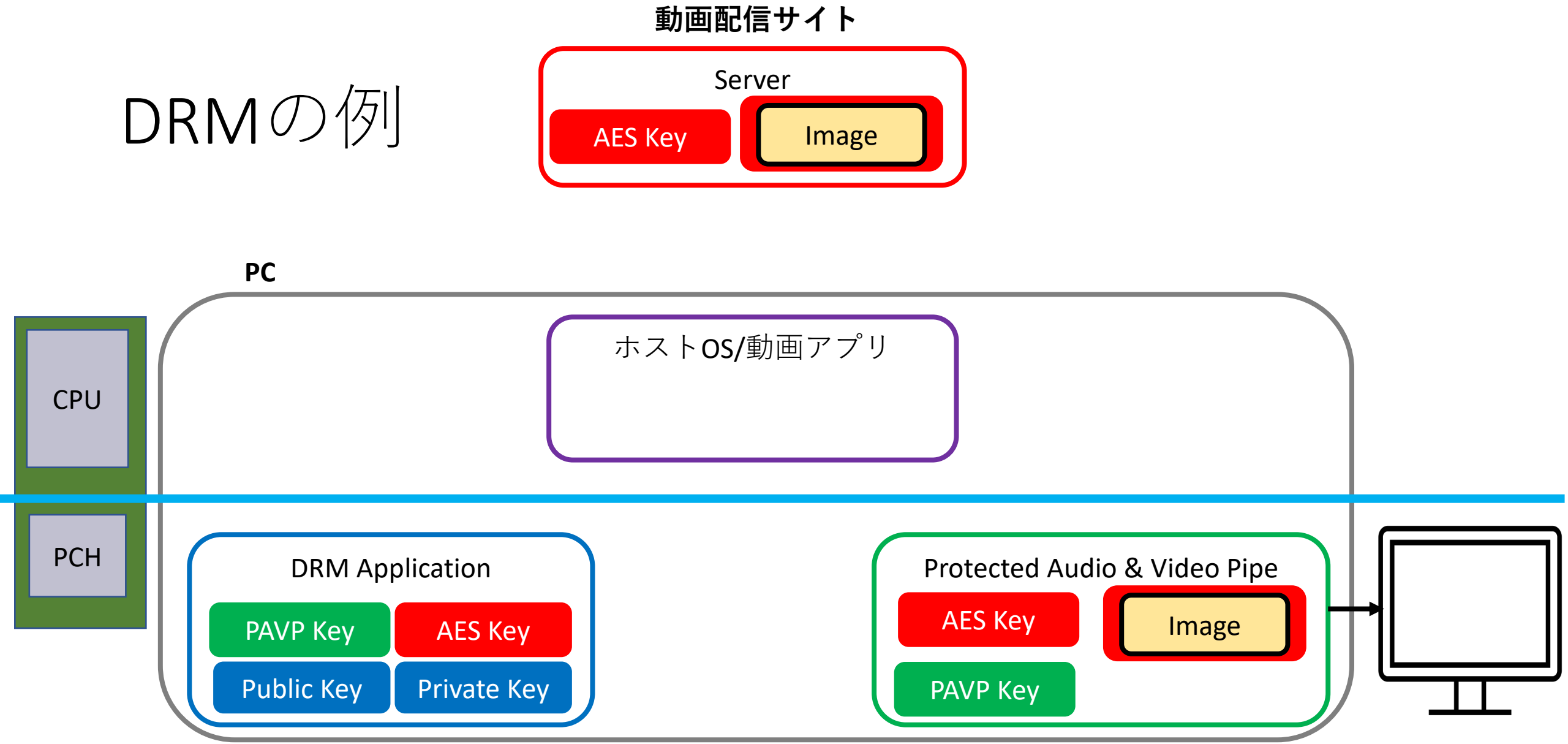
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



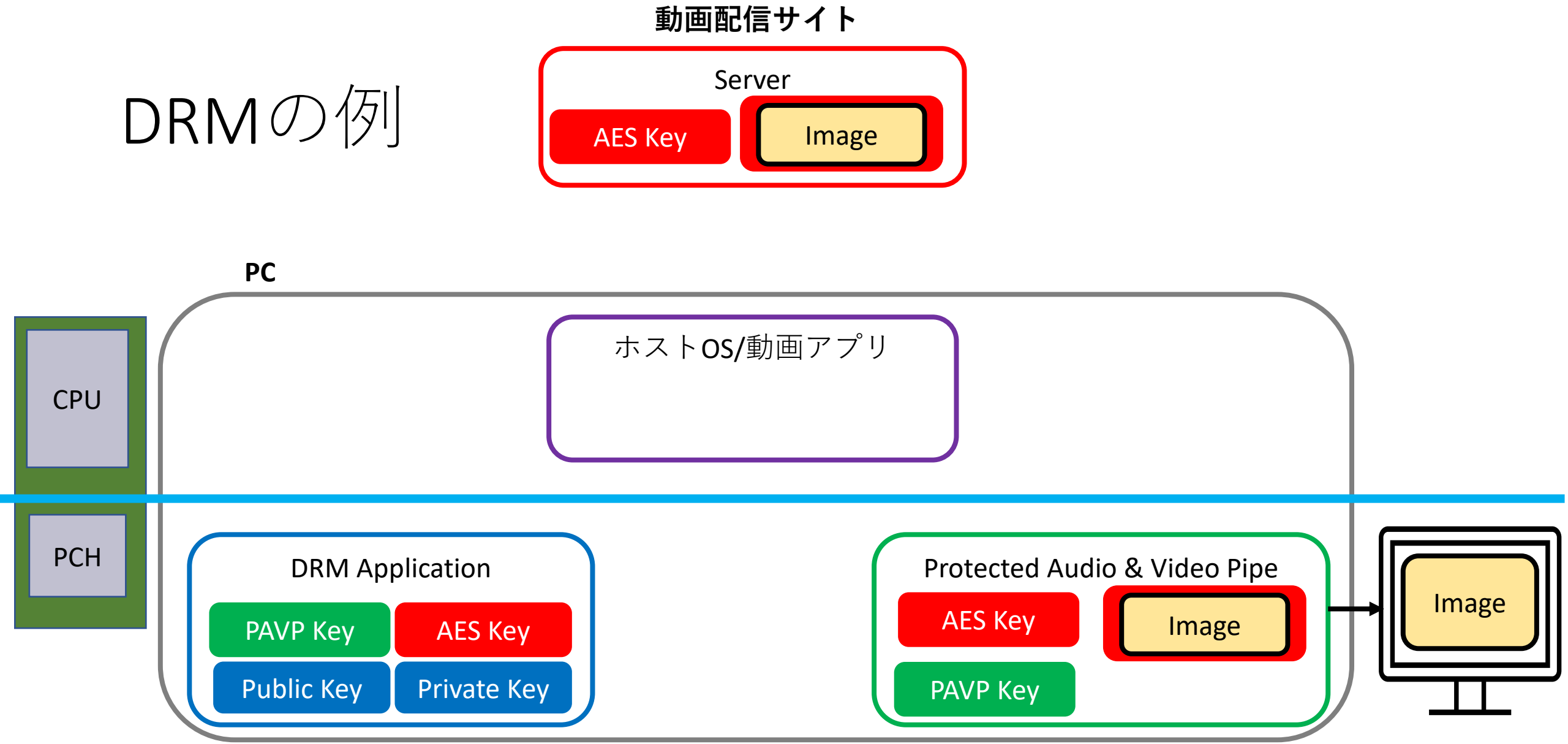
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



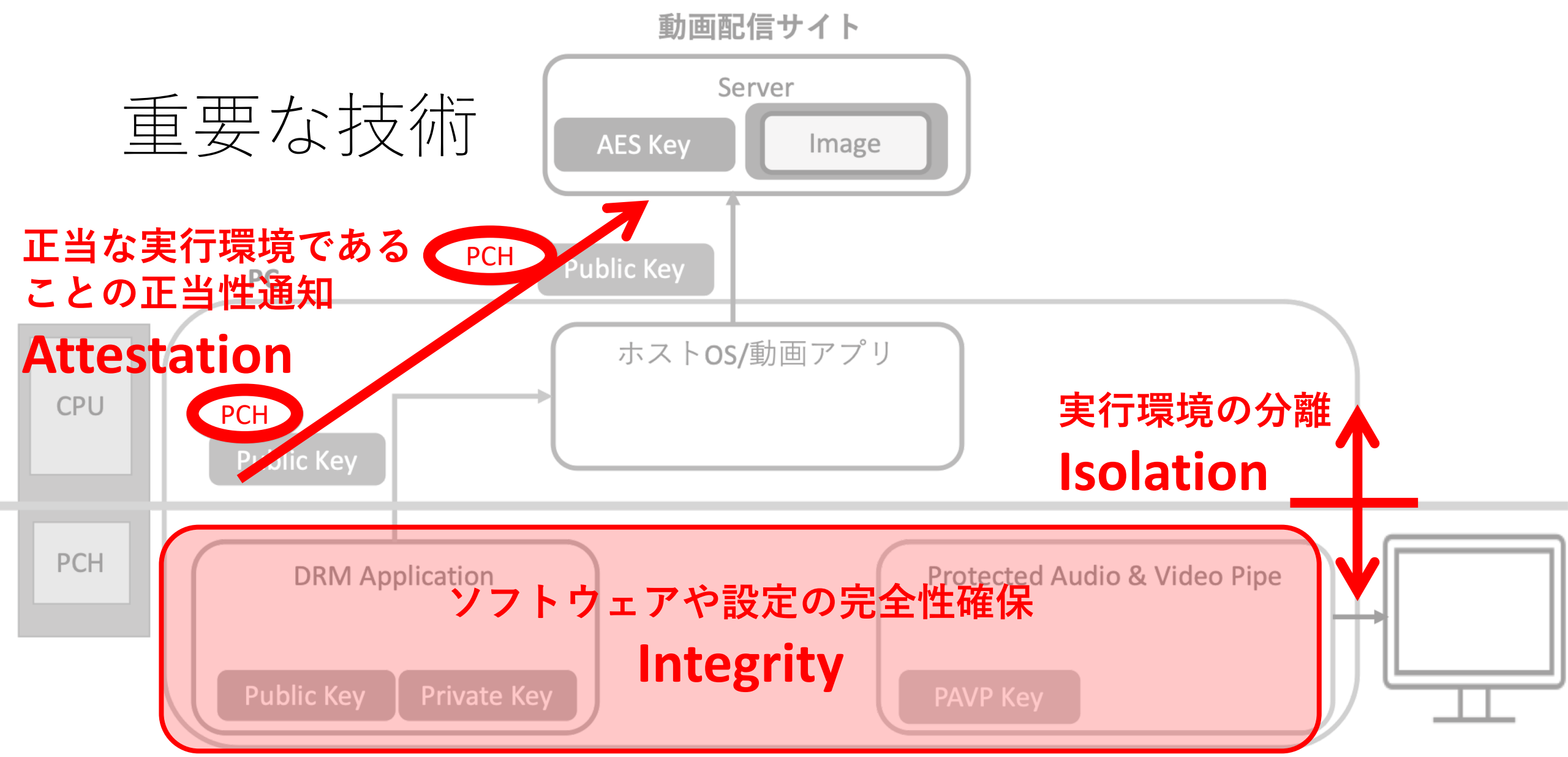
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

DRMの例



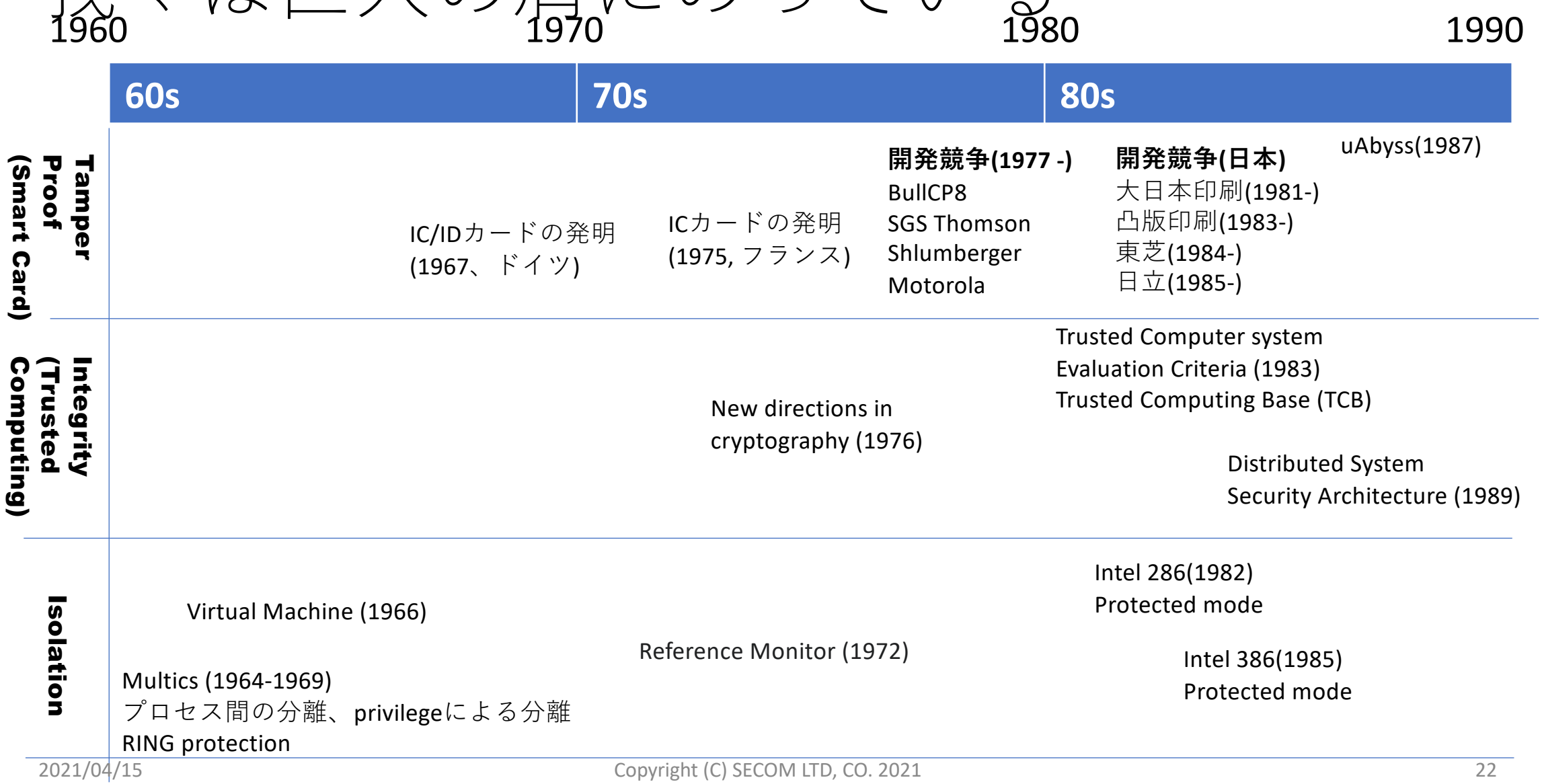
https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

重要な技術

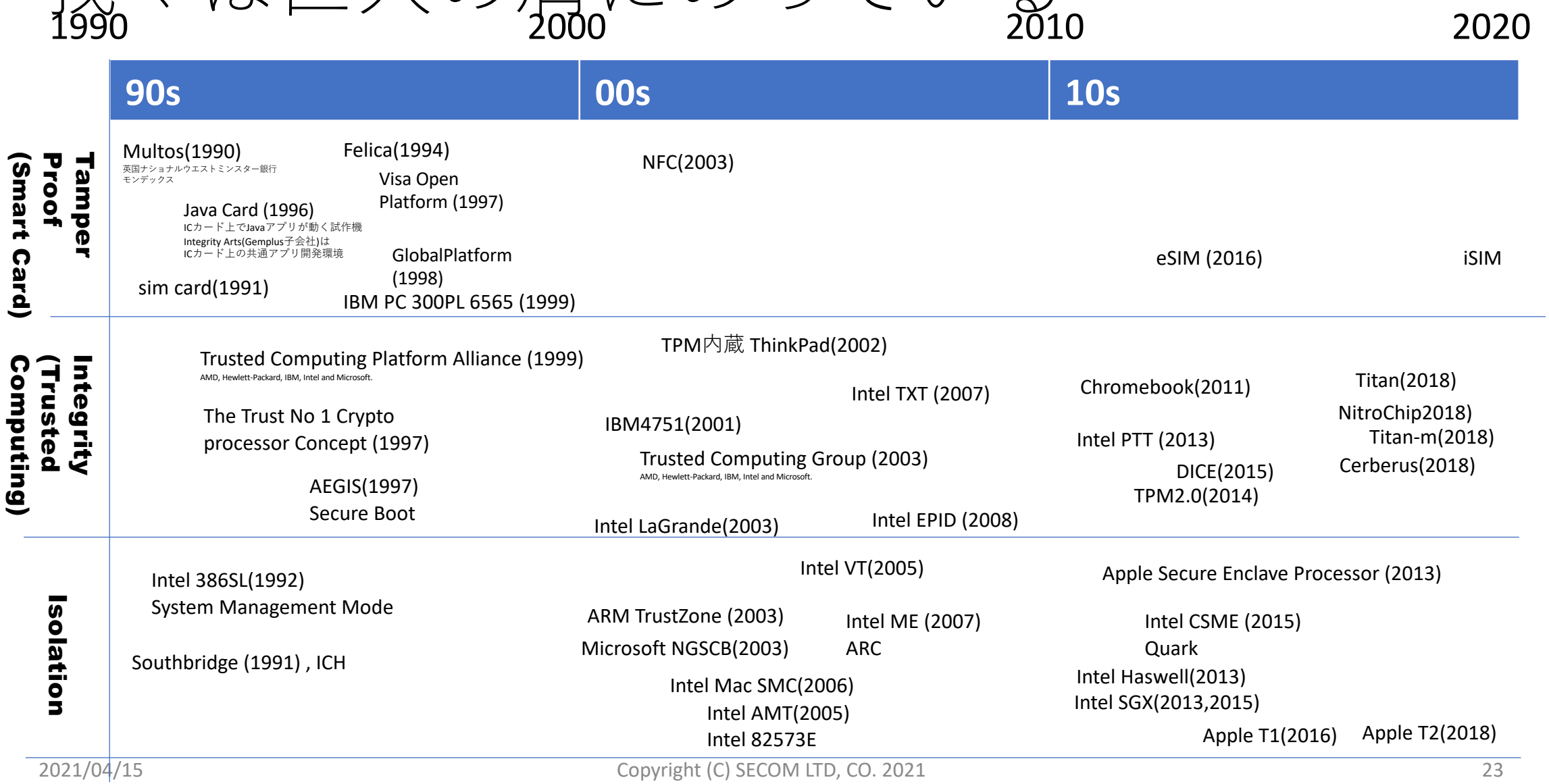


https://software.intel.com/content/www/us/en/develop/documentation/dal-developer-guide/top/sdk-contents/sdk-contents-samples/sdk-contents-samples-applets-with-host-applications/sdk-contents-drm-solution-sample.html
で掲載されている例をさらに簡略したもの。

我々は巨人の肩にのっている



我々は巨人の肩にのっている



本日の流れ

- トラストの技術を活用した具体例：DRM
- 歴史
  - **ICカード（SmartCard）の歴史**
  - モバイルの歴史
  - PC/サーバの歴史
- Intel SGXの特殊性
- まとめ

本講演でのTrust

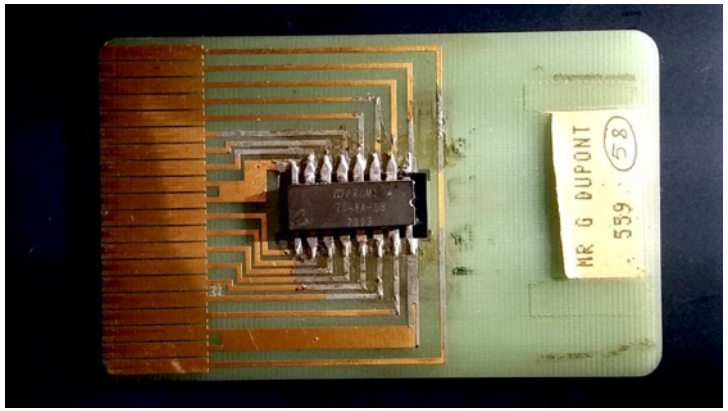
Trust ＝想定したプログラム（データ）だけが動く

Trustを確立する技術

- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

Smart Card (ICカード)

- 1959年アメリカでICが発明される
- 1967年ドイツで発明される
  - IDカード用途
- クレジットカード、SIMカード、交通系カード、NFC、SE、基盤埋込型、等々
- 暗号処理、署名処理、検証処理、秘密鍵、公開鍵、その他秘密情報
- 不正な読み書き、回路分析の対策が施された、**対タンパ性**のある回路



初期のプロトタイプ(1975)

One of the first prototype of smart card, realized by its inventor Roland Moreno circa 1975. It is based on a 2 kilobit programmable read-only memory device.
ByB, Creative Commons Attribution-Share Alike 4.0 International
https://commons.wikimedia.org/wiki/File:Prototype_moreno2.jpg

現代の様々なインターフェイス

耐タンパ性

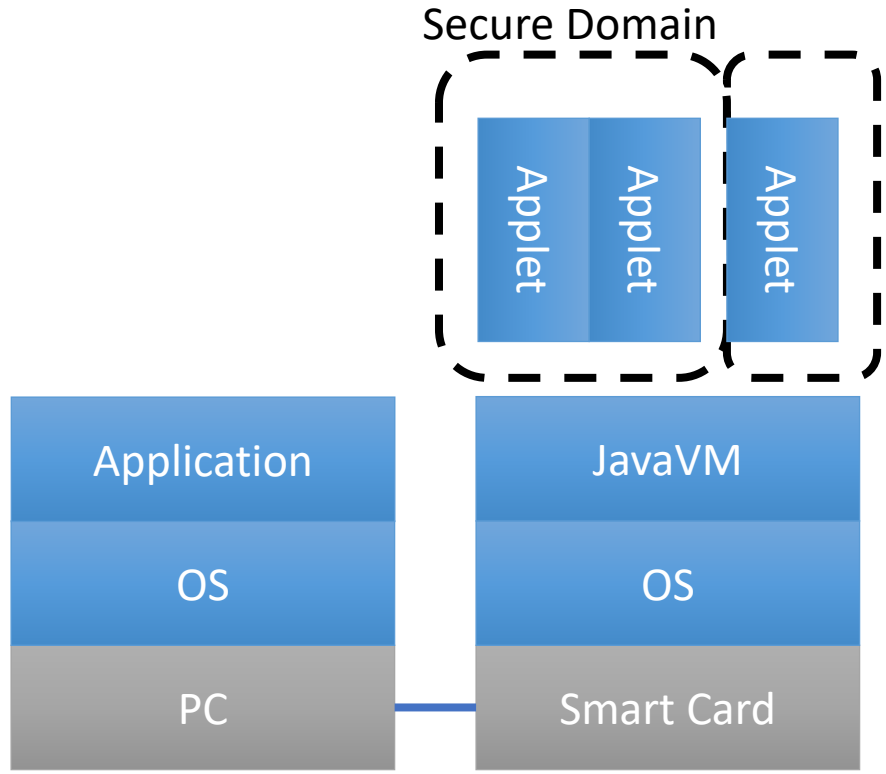
- Tamper
  - （許可なく勝手に）変更する
- Tamper Proof Tape
  - 貼った後、剥がされた跡がわかるテープ
- 様々な表現
  - Anti-Tamper
  - Tamper Proof
  - Tamper Resilient
  - Tamper Detection
- ICへの応用
  - 回路を製造する物理的な技術
  - チップ内部について、
    - 後から変更できない
    - 中身がどうなってるかわからない
    - 見ようとするとゼロ化される、壊れる

Permanent tamper evident numbered label showing the label applied to the surface, the label voiding and the permanent void message left on the surface of the container.

TamperTechTeam

Java Cardの登場とGlobalPlatform

- 1996
  - Schlumberger社がSmart Card内部でJavaバイトコードを実行するJava Cardを発表
  - Visa, Integrity Arts (Gemplus子会社)がSmart Card用のオープンなOSを開発中
  - SunによるIntegrity Arts買収
- 1997
  - Java Cardフォーラム結成（Gemplus, Schlumbergerその他）Java CardのAPI策定
- 1998
  - Visa Open Platform発表
  - （Smart Card内のアプリケーションの管理手法の策定）
- 1999
  - Visa から Open Platformの仕様をGlobalPlatformコミュニティへ移譲



Hendry, M. (2007). Multi-application smart cards: technology and applications.

本日の流れ

- トラストの技術を活用した具体例：DRM
- 歴史
  - ICカード（SmartCard）の歴史
  - **モバイルの歴史**
  - PC/サーバの歴史
- Intel SGXの特殊性
- まとめ

本講演でのTrust

Trust ＝想定したプログラム（データ）だけが動く

Trustを確立する技術

- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

モバイルの歴史

- 1980年代
  - 各国で異なるアナログ方式（第１世代）
- 1983年
  - 欧州で第２世代仕様(GSM)策定開始（デジタル化の波）
  - SIM (Subscriber Identity Module) カードの仕様もGSMの一部
- 1989年
  - GSMの仕様策定作業がCEPTからETSIに引き継がれた
- 1992年
  - 欧州で、GSM仕様の携帯の普及が始まる
- 1990年代半ば
  - 携帯用Javaアプリの登場
- 2007年初頭
  - iPhone発表

https://commons.wikimedia.org/wiki/File:Nokia_C1-02-91938.jpg
Creative-Commons-Licence CC BY-SA 4.0.
Raimond Spekking, “Nokia C1-02”

モバイル端末へのセキュリティ要求

- モバイル向けアプリケーションのオープン化が進んだ
  - さまざまなアプリケーションを携帯電話に後から追加できようになった。
- 各所からのアプリ/データ保護の要求
  - 規制当局
    - 盗難携帯かどうか見分けたい（機器IDの保護）
    - 使用電波の帯域などの設定の保護（設定の保護）
  - キャリア
    - SIMロック（設定やプログラムの保護）
    - コピープロテクション（プログラムやデータの保護）
  - 携帯利用者
    - 盗難携帯かどうか見分けたい（機器IDの保護）
    - 秘密情報の保護（データの保護）
- 実現方法
  - 完全性の確保と実行環境の分離
- 実装案
  - 通常処理のチップと重要な処理のチップとの 2 チップによる実装
    - 開発コストの問題で却下
  - 1つのチップを通常モードとセキュアモードで切り替えて、あたかも 2 チップのように振る舞う実装

Asokan, N. (2019, November). Hardware-assisted trusted execution environments: Look back, look ahead.

ARM TrustZone

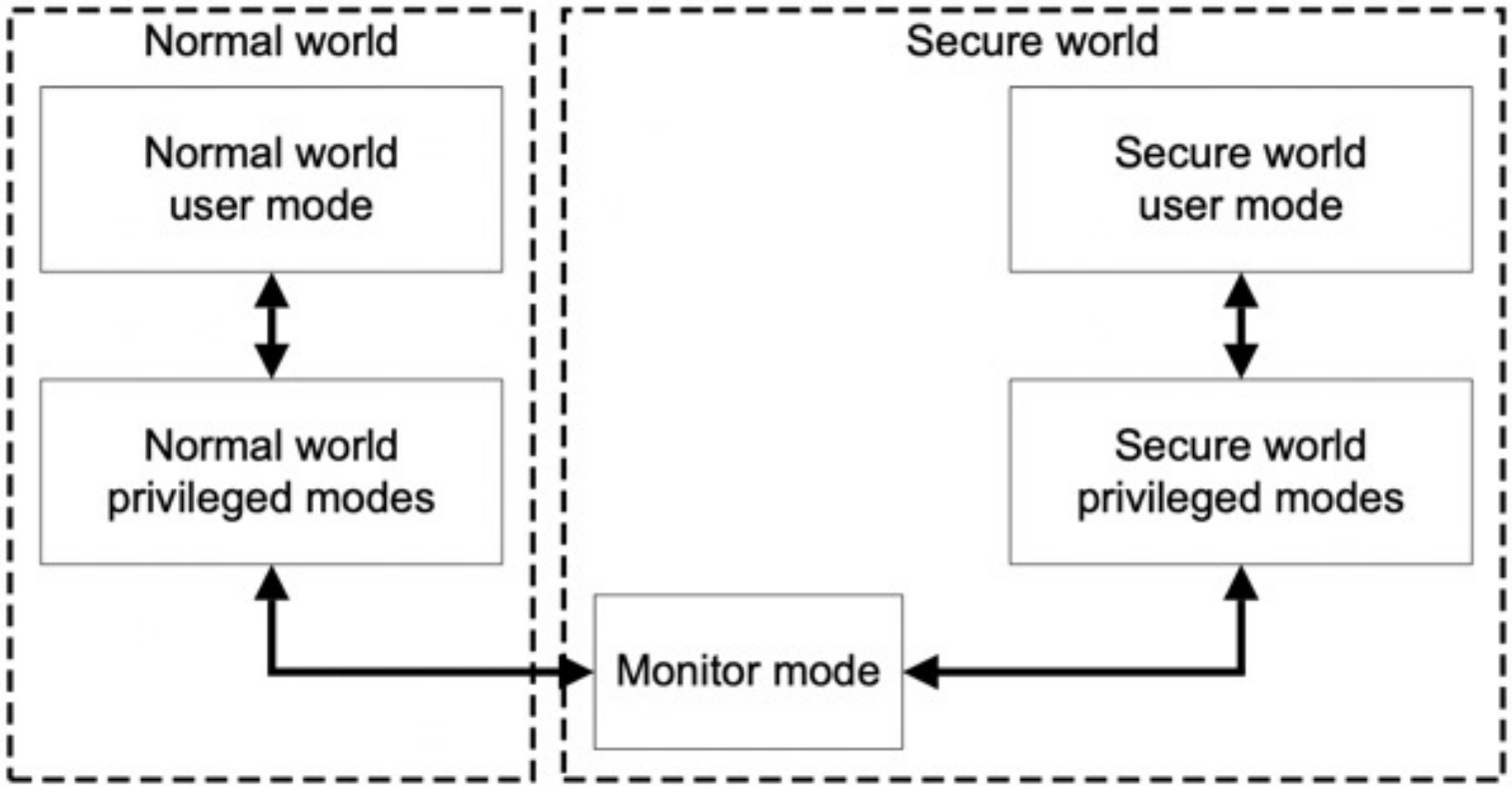
携帯ユーザーにある程度
解放されている実行環境

携帯ベンダーやキャリア側で
管理されている実行環境

ユーザーアプリの階層

カーネル/OSの階層

仮想マシンの階層

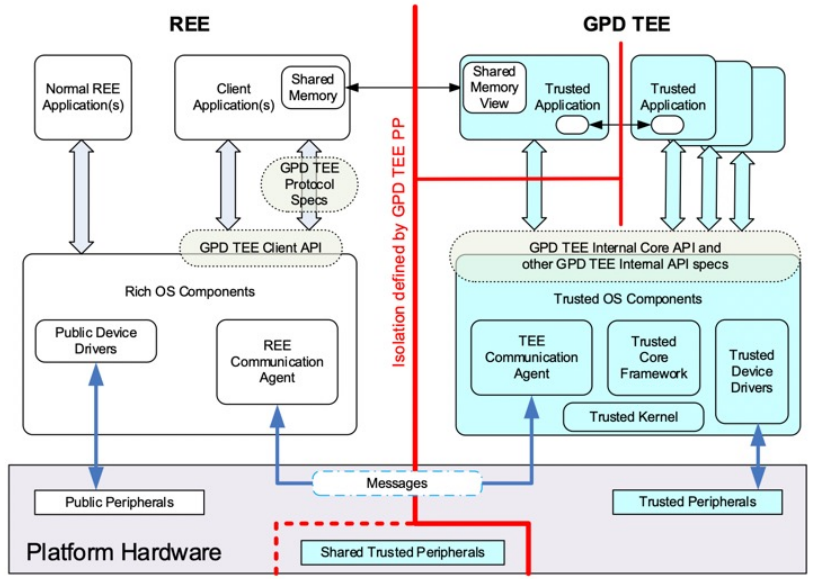


重要な
周辺機器への
I/O

Holdings, A. R. M. (2009). ARM Security Technology: Building a Secure System using TrustZone Technology.

TrustZone登場とその後の道のり

- 1982
  - 特許[Texas Instruments] Security bit for designating the security status of information stored in a nonvolatile memory
  - 特許[Texas Instruments] Secure microprocessor/microcomputer with secured memory
- 1996
  - 特許[Intertrust] Systems and methods for secure transaction management and electronic rights protection
- 2002
  - 特許[Nokia] Secure execution architecture
- 2003
  - 製品[Texas Instruments] OMAP
- 2004
  - 製品[ARM] Trust Zone
- 2006
  - 製品[ARM, Discretix] CryptoCell （Root of Trustとなる暗号モジュール）
- 2009
  - Open Mobile Terminal Platformによる標準化： Trusted Execution Environmentという言葉の誕生
- 2010
  - Global Platform から TEE API 1.0の発表



GlobalPlatform Technology TEE System Architecture Version 1.2
https://globalplatform.org/wp-content/uploads/2017/01/GPD_TEE_SystemArch_v1.2_PublicRelease.pdf

Asokan, N. (2019, November). Hardware-assisted trusted execution environments: Look back, look ahead.

本日の流れ

- トラストの技術を活用した具体例：DRM
- 歴史
  - ICカード（SmartCard）の歴史
  - モバイルの歴史
  - **PC/サーバの歴史**
- Intel SGXの特殊性
- まとめ

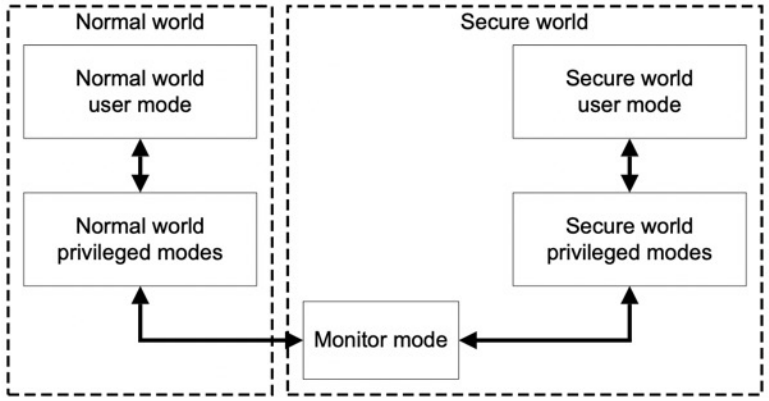
本講演でのTrust

Trust ＝想定したプログラム（データ）だけが動く

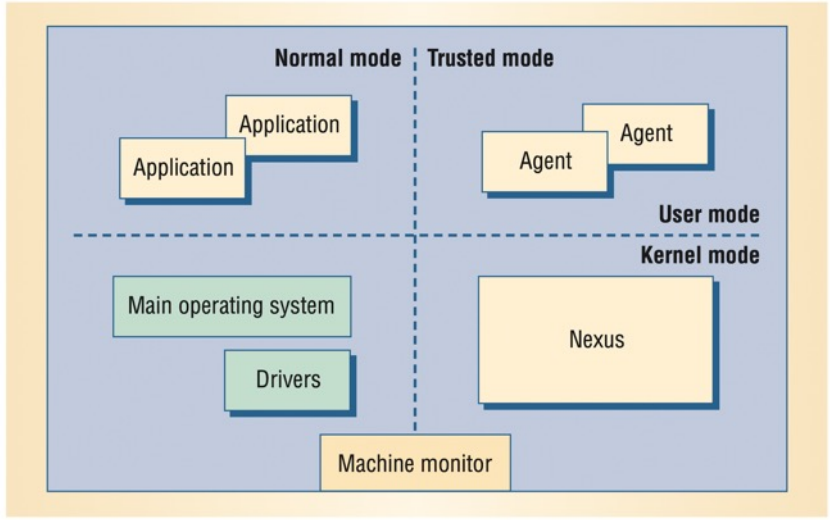
Trustを確立する技術

- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

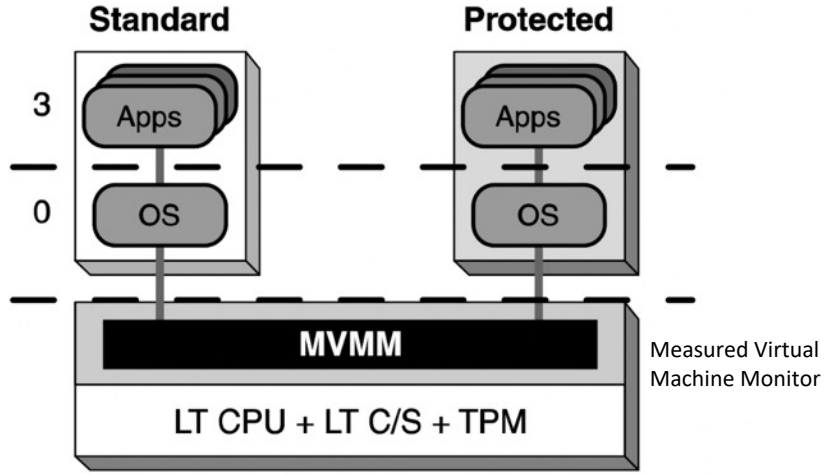
PC業界もモバイル端末と同じ目標があった



ARM TrustZone (2003)



Microsoft NGSCB(2002)
旧称・Palladium



Intel LaGrande(2002)
のちのIntel TXT

Reference Monitor (1972)のアイディアを参考に
・ベンダーのための完全な実行環境：出荷時（アップデート）から完全性を保証できる実行環境
・ユーザーのための自由な実行環境：どのようなOSでもアプリでもユーザーが好きなものを実行できる環境
の両立という目標
Wintelは実用化に遅れたが、Windows10のセキュリティで実現（詳しくは後半で！）

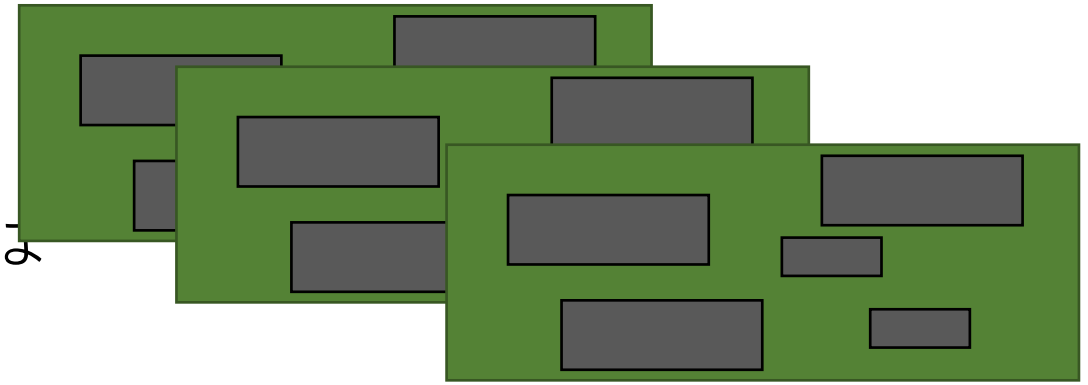
Intel HT/CT/VT/LT (HT, 64bit, VM, DRTM)

Anderson, J. P. (1972). Computer security technology planning study.
Holdings, A. R. M. (2009). ARM Security Technology: Building a Secure System using TrustZone Technology.
England, P., Lampson, B., Manferdelli, J., & Willman, B. (2003). A trusted open platform. *Computer*, *36*(7), 55-62.
Grawrock, D. (2005). The Intel Safer Computing Initiative.

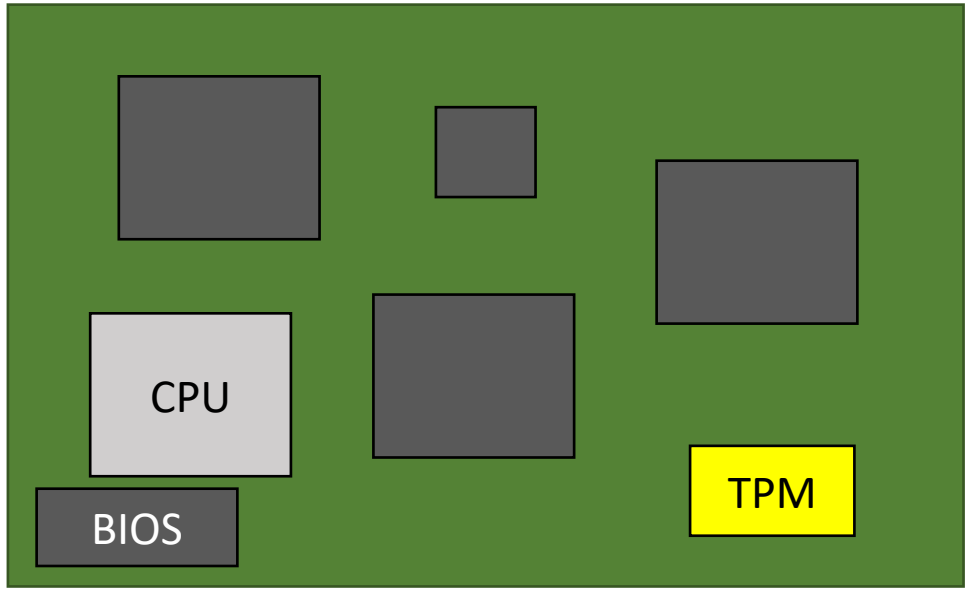
Trusted Computing Group

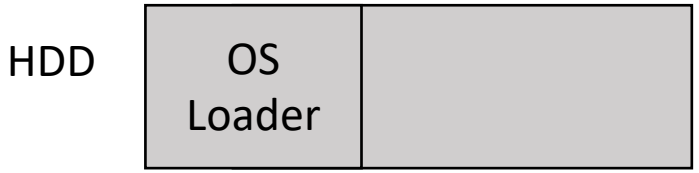
- Trusted Computing Platform Alliance (TCPA)
  - 1999/10/11設立
  - メンバー
    - Intel, Microsoft, IBM, Compaq, Hewlett-Packard
  - 目的
    - パソコンによる電子商取引の信頼性向上のための技術の標準化
- Trusted Computing Group
  - 2003年TCPAを引き継ぎ
    - Intel, Microsoft, IBM, AMD, Hewlett-Packard
  - 成果物：Trusted Platform Module　(TPM)
    - 乱数機能、ハッシュ機能、暗号機能、鍵の安全な保存場所などを備えたチップの仕様

TPM (Trusted Platform Module)

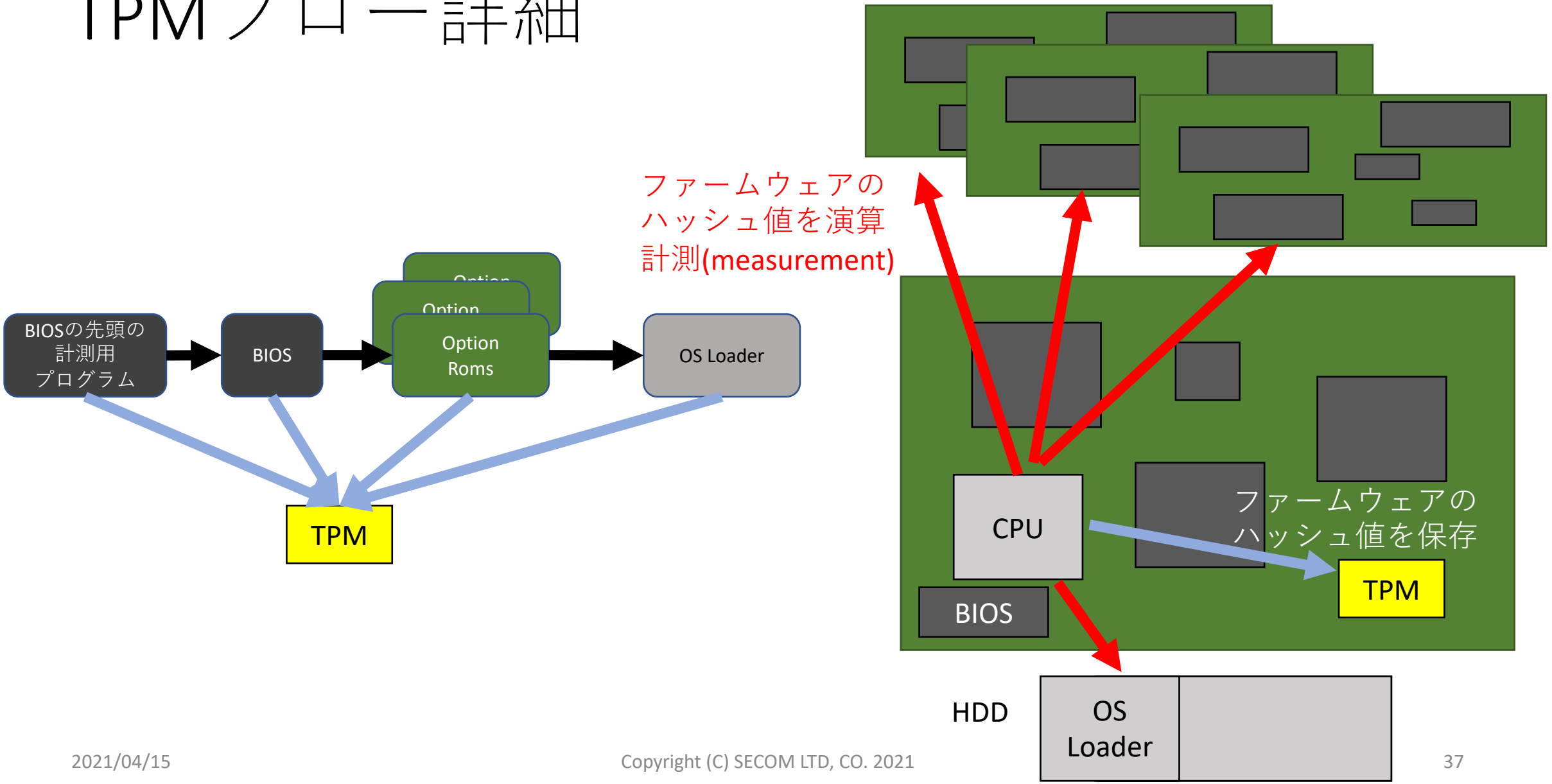


- ICカードは、一つのチップを保護すれはよかった
- PCの世界は？
  - マザーボード、拡張ボード、ハードディスの組み合わせ
  - これらの完全性を満たす方法は？
- 耐タンパ性のあるチップを一つ利用して、基盤全体（プラットフォーム）のハードウェアとファームウェアの完全性を実現

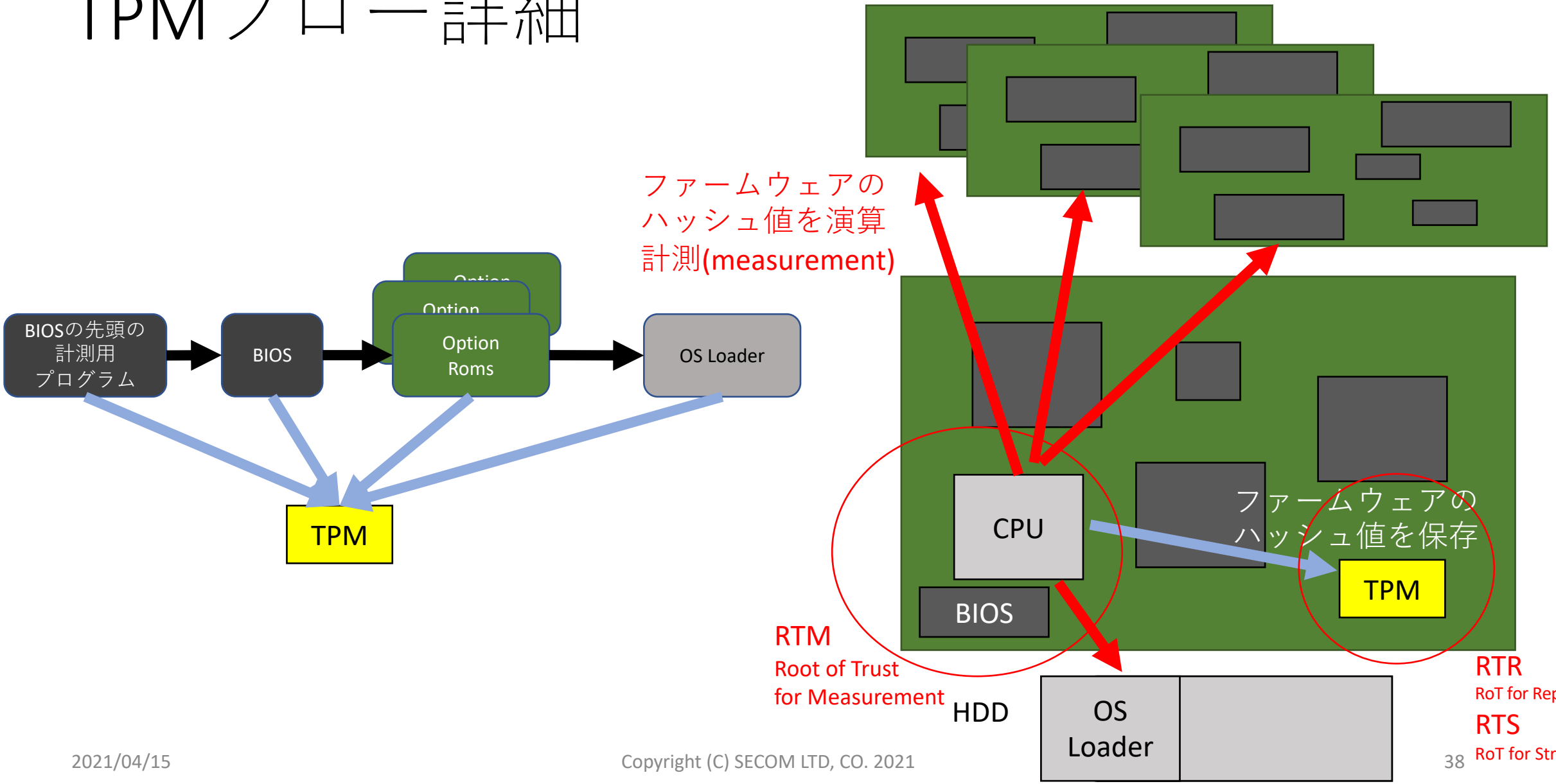




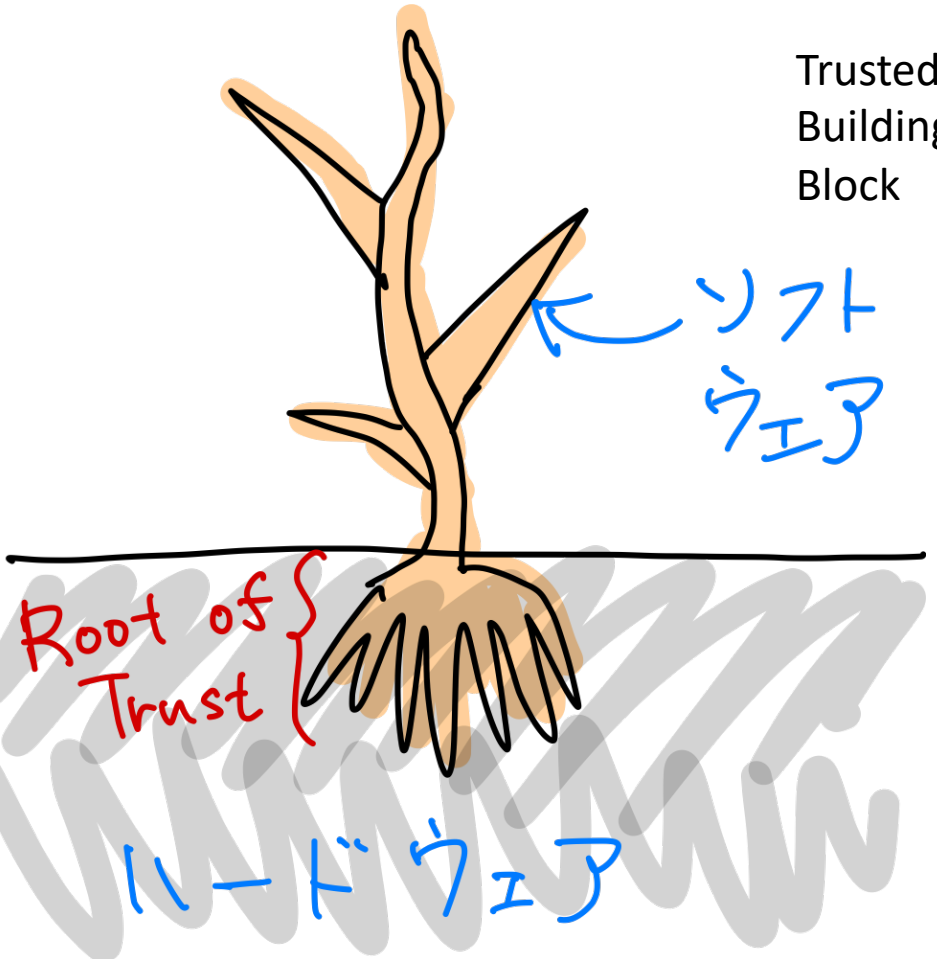
TPMフロー詳細



TPMフロー詳細



Root Of Trust



宮澤のイメージ
本来は「信頼の根幹」という意味

Trusted Building Block

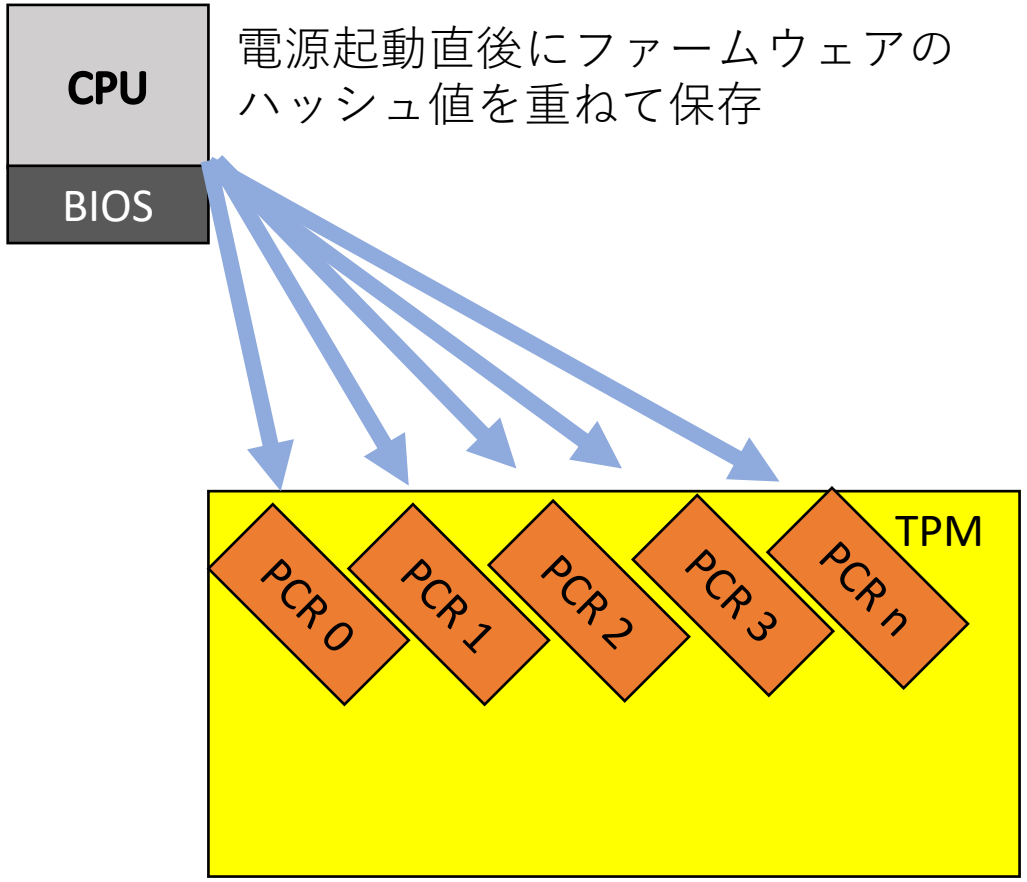
- Root Of Trust for Measurement
  - **完全性**を確認する計測プログラム
  - BIOSのROMやCPUのマイクロコード
- Root Of Trust for Report
  - **機器認証**できる形での**完全性報告**のための証明書
  - TPM
- Root Of Trust for Storage
  - 外部ストレージへ**暗号化保存**する暗号鍵
  - TPM
- ソフトウェアの完全性の木を成長させる
  - Chain Of Trust
- 計測のタイミング異なるRTM
  - Static RTM(BIOS + TPM)：電源起動直後
  - Dynamic RTM (Intel TXT)：任意タイミング

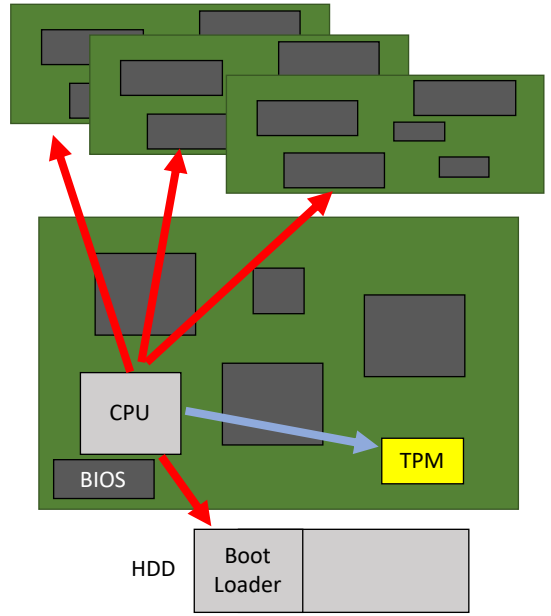
詳しくは後半で！

Secure Boot/Trusted Boot/xxx Boot

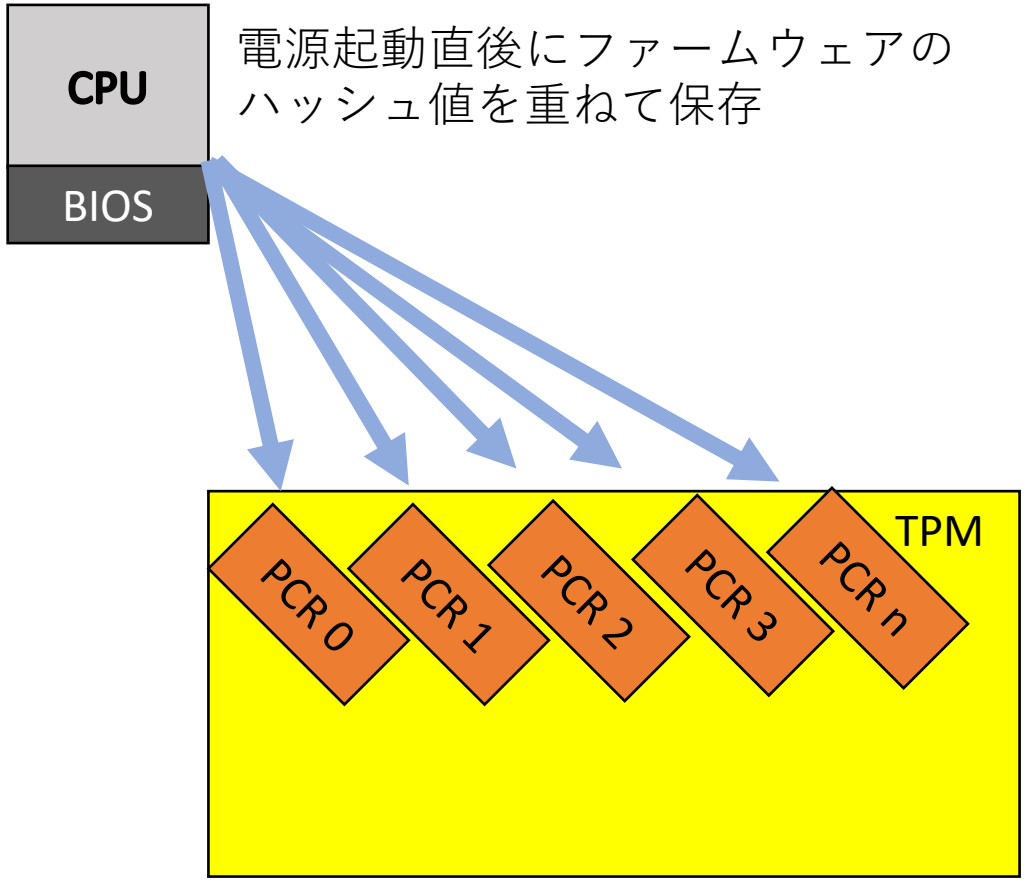
- Secure Boot
  - ブート時にソフトウェアをチェック（端末内で検証）
  - ブート時に想定外のファームウェア・ソフトウェアがあった場合起動を止める
  - 適用例：改竄検知後に起動すると深刻な問題が発生する場合
- Trusted Boot
  - ブート時にソフトウェアを計測 (ハッシュをRoot Of Trust for Reportに保存するだけ)
  - ブートを最後までやり遂げる。
  - ブート時のチェック結果をまとめ、TPM内部の鍵で署名をつける。
  - 第３者にブート時のチェック結果を送信し判断を仰ぐ(Remote Attestation)
  - 適用例：分散システムのノードにSecure Bootを適用すると、故障なのか改竄なのか遠隔から確認できない
- その他ブート(xxx Boot)の呼び名、機能や会社によって異なる
  - Authenticated Boot、Verified Boot、Measured Boot

Remote Attestation



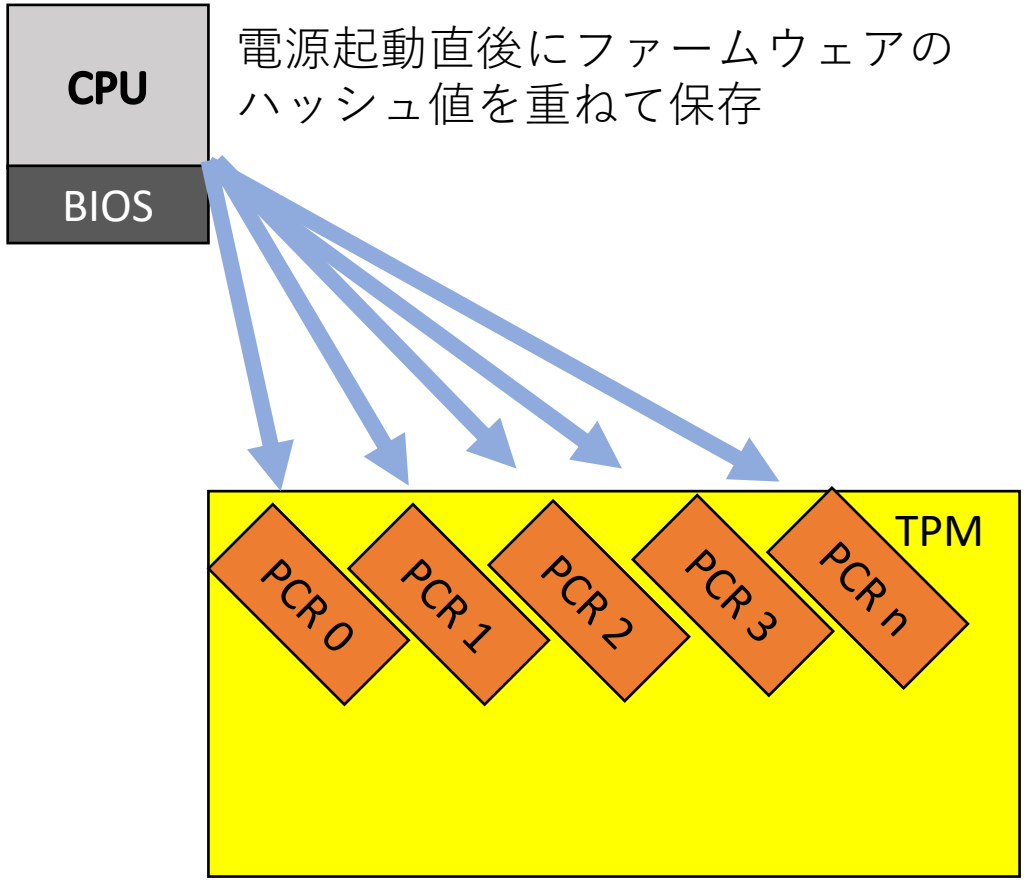


Remote Attestation



OSが起動

Remote Attestation

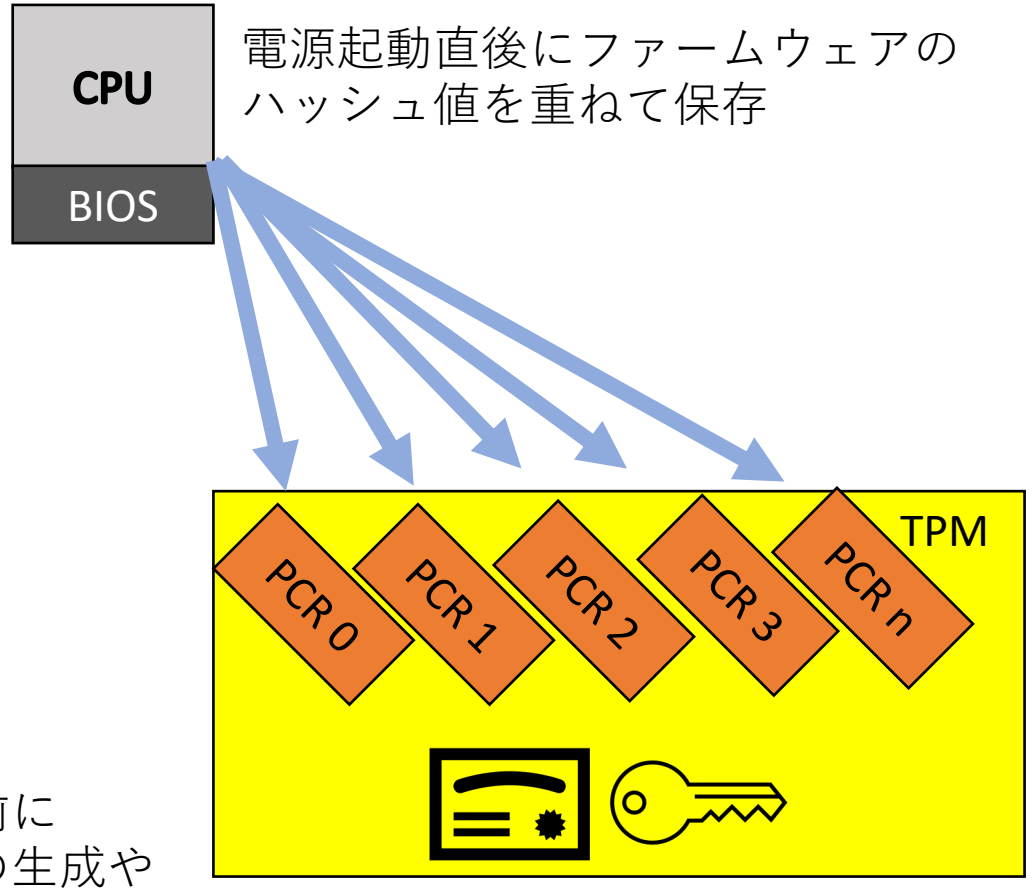


OSが起動

例
・社内ネットワークにノートPCでログイン
・分散コンピューティングの１ノードとして参加

コンピュータ群を
管理するサーバ

Remote Attestation



電源起動直後にファームウェアの
ハッシュ値を重ねて保存

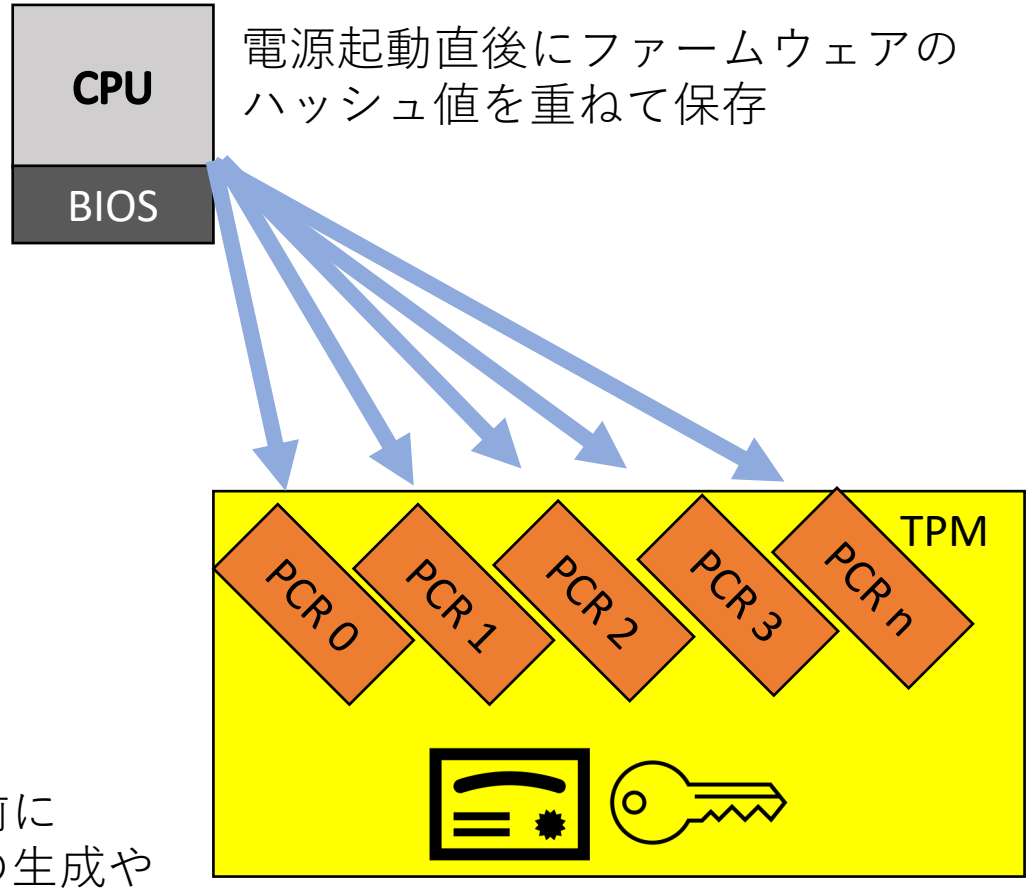
事前に
鍵の生成や
証明書を用意しておく

OSが起動

例
・社内ネットワークにノートPCでログイン
・分散コンピューティングの１ノードとして参加

コンピュータ群を
管理するサーバ

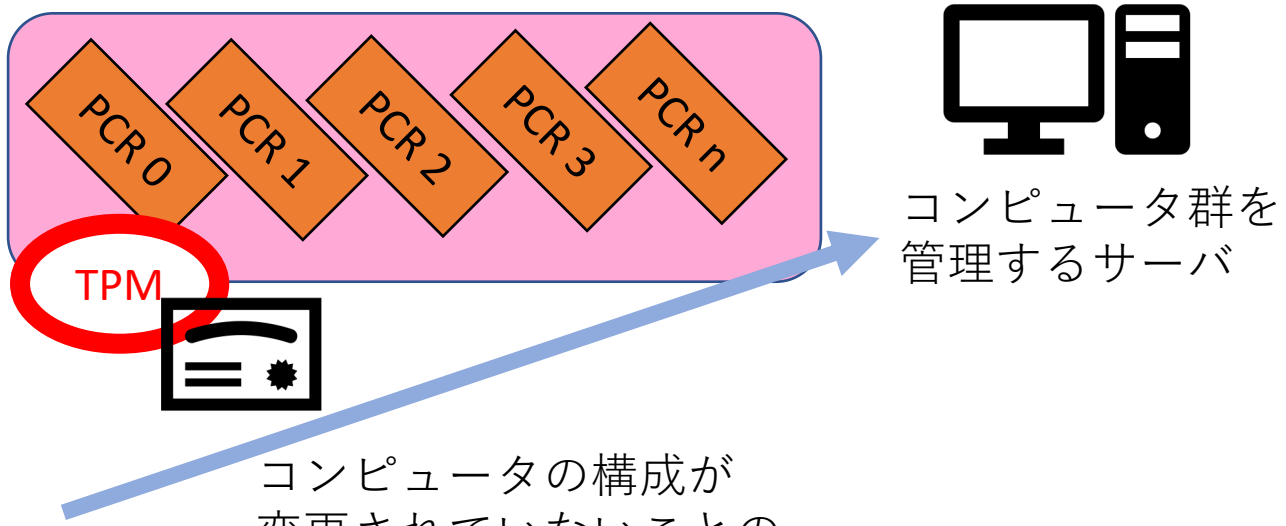
Remote Attestation



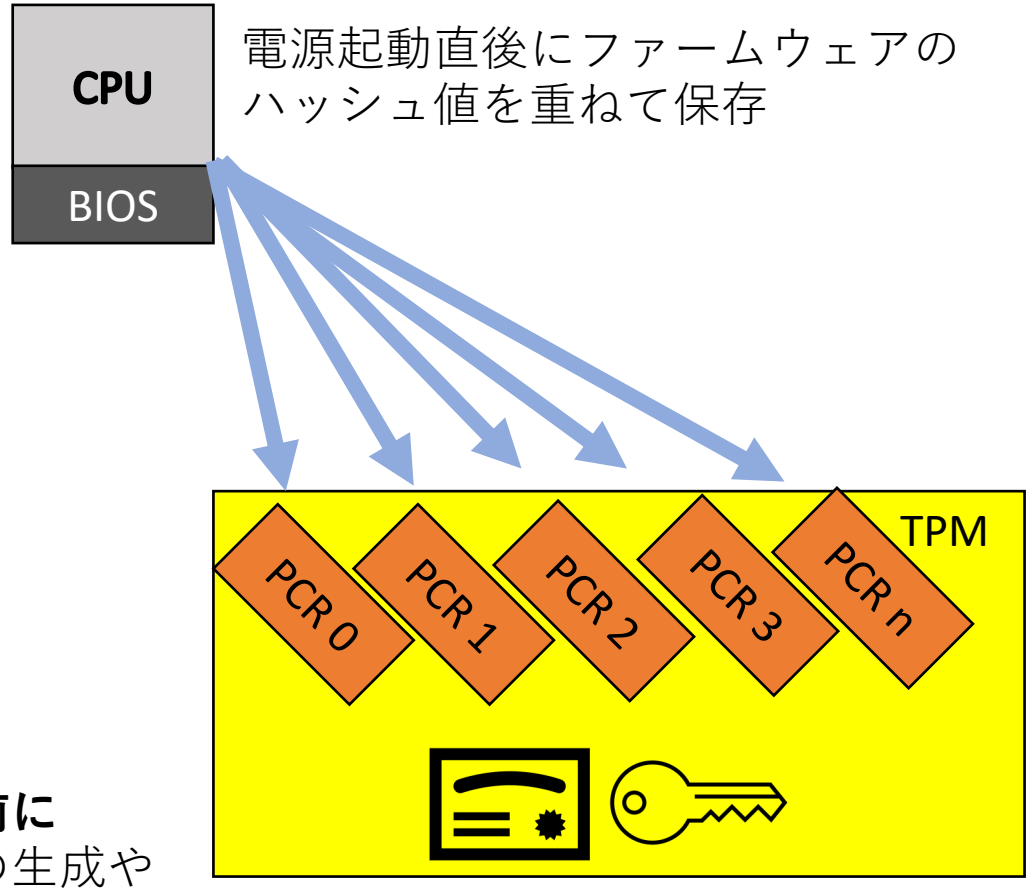
事前に
鍵の生成や
証明書を用意しておく

OSが起動

例
・社内ネットワークにノートPCでログイン
・分散コンピューティングの１ノードとして参加



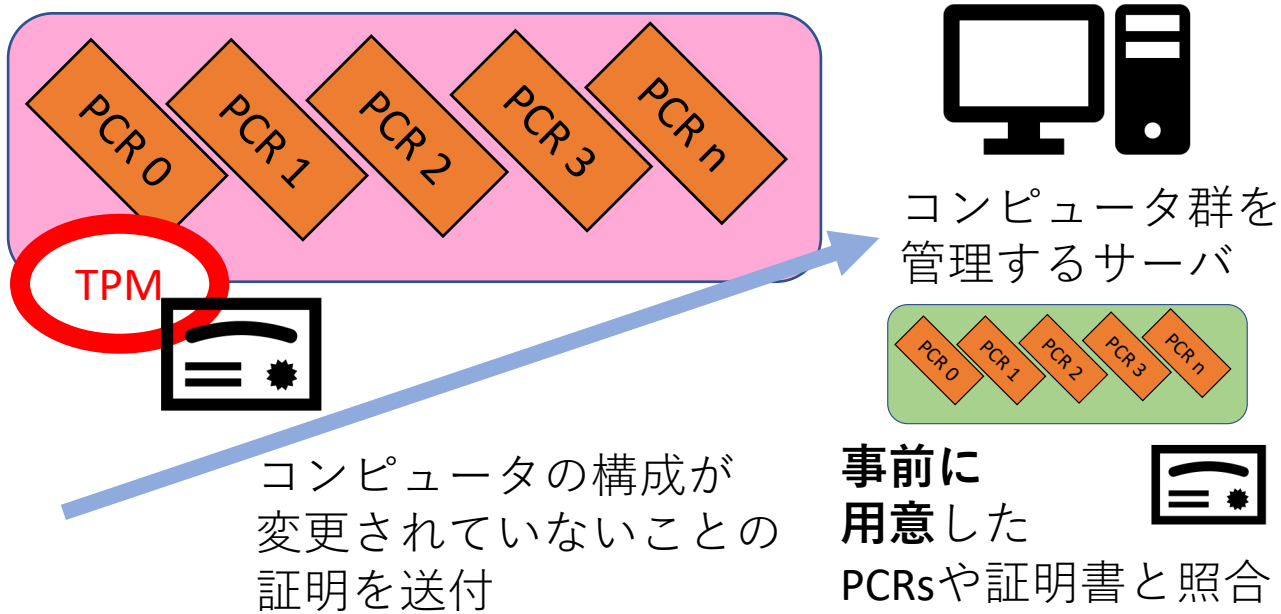
Remote Attestation



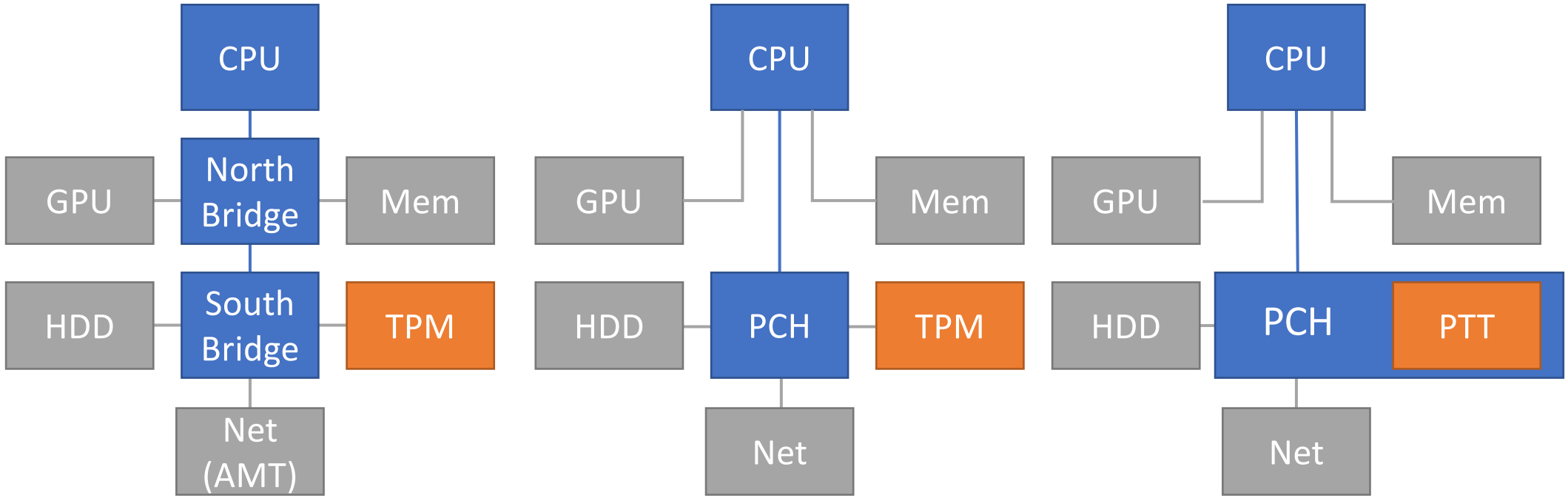
事前に
鍵の生成や
証明書を**用意**しておく

OSが起動

例
・社内ネットワークにノートPCでログイン
・分散コンピューティングの１ノードとして参加



PC内における TPMの変遷

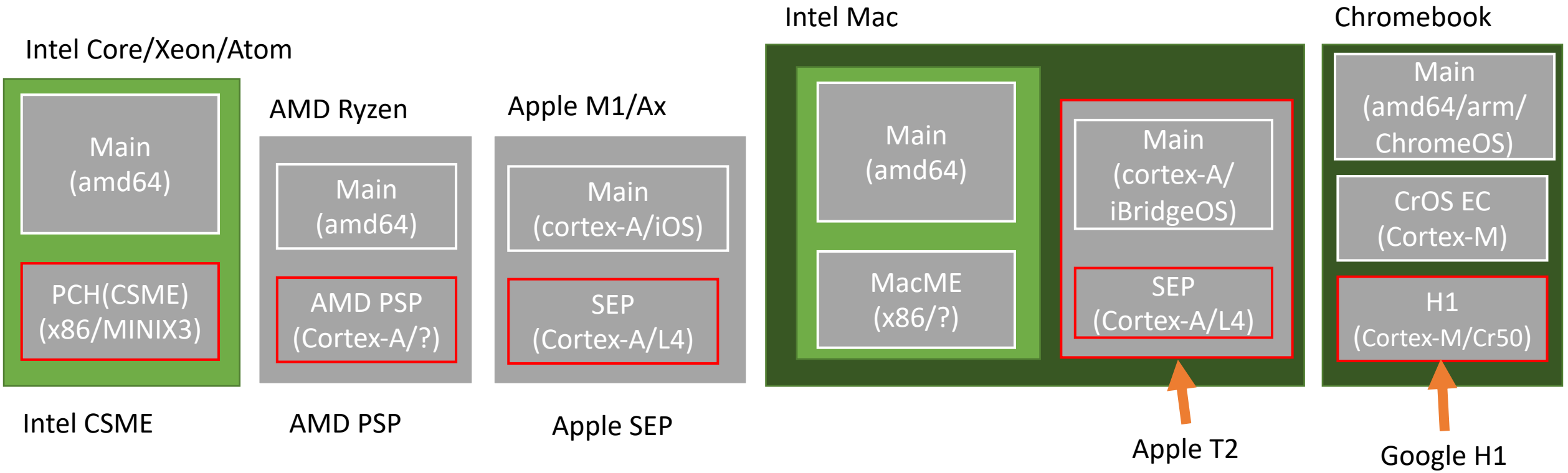


～2008

2008 ～

2013 ～

各社のセキュリティを担当するチップ（赤枠）



Intel CSME

Converged Security Management Engine

AMD PSP

Platform Security Processor

Apple SEP

Secure Enclave Processor

本日の流れ

- トラストの技術を活用した具体例：DRM
- 歴史
  - ICカード（SmartCard）の歴史
  - モバイルの歴史
  - PC/サーバの歴史
- **Intel SGXの特殊性**
- まとめ

本講演でのTrust

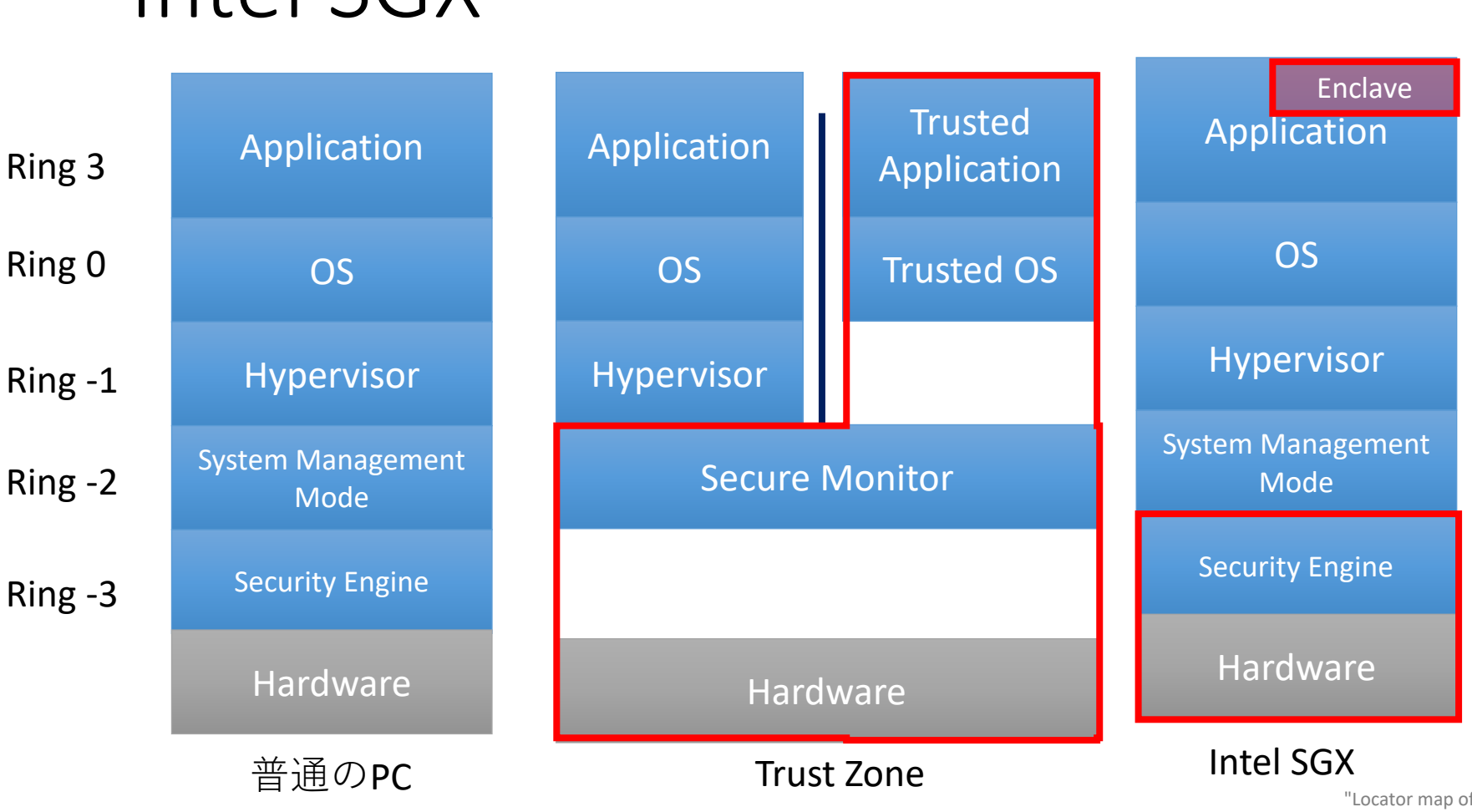
Trust ＝想定したプログラム（データ）だけが動く

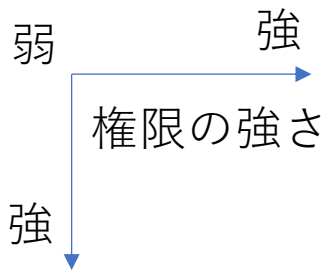
Trustを確立する技術

- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

Intel SGX

詳しくは後半で！





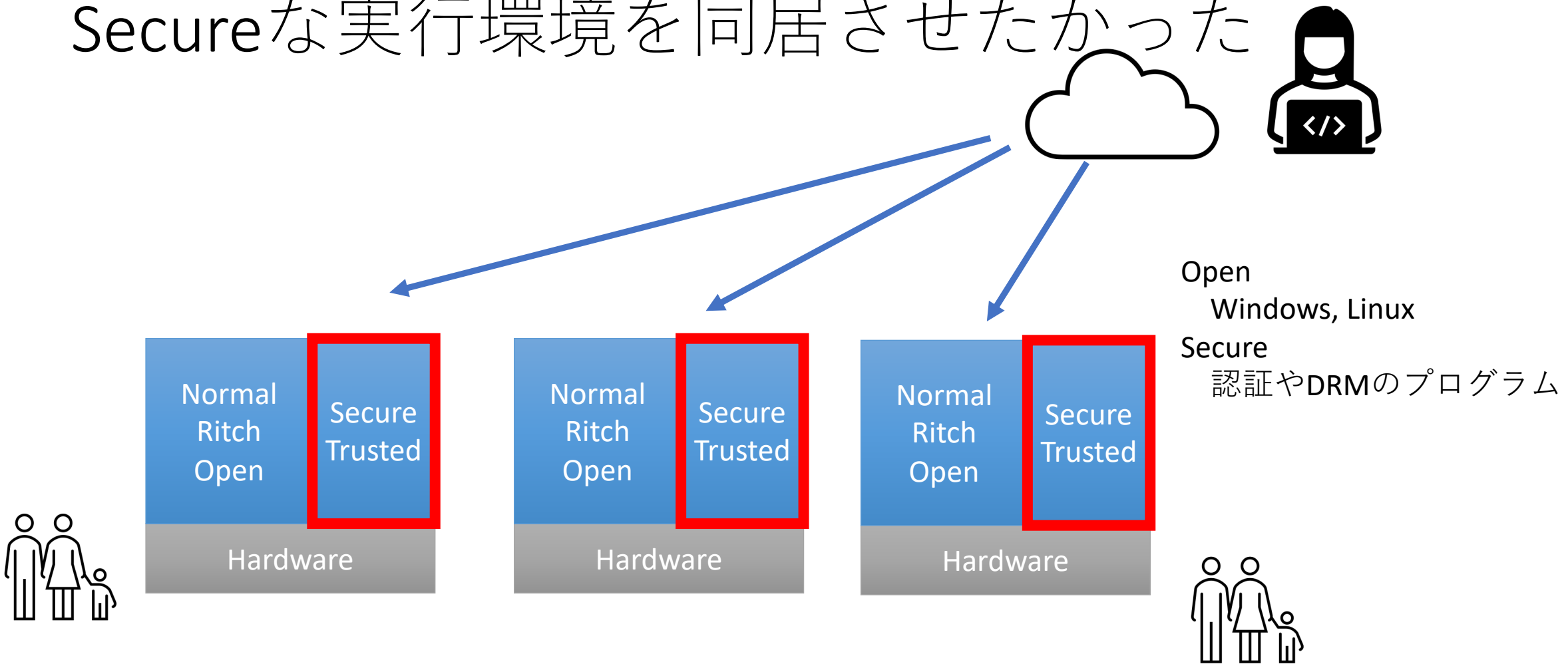
Enclave (飛び地)

"Locator map of municipalities of East Timor"© J. Patrick Fischer (Licensed under CC BY 4.0)

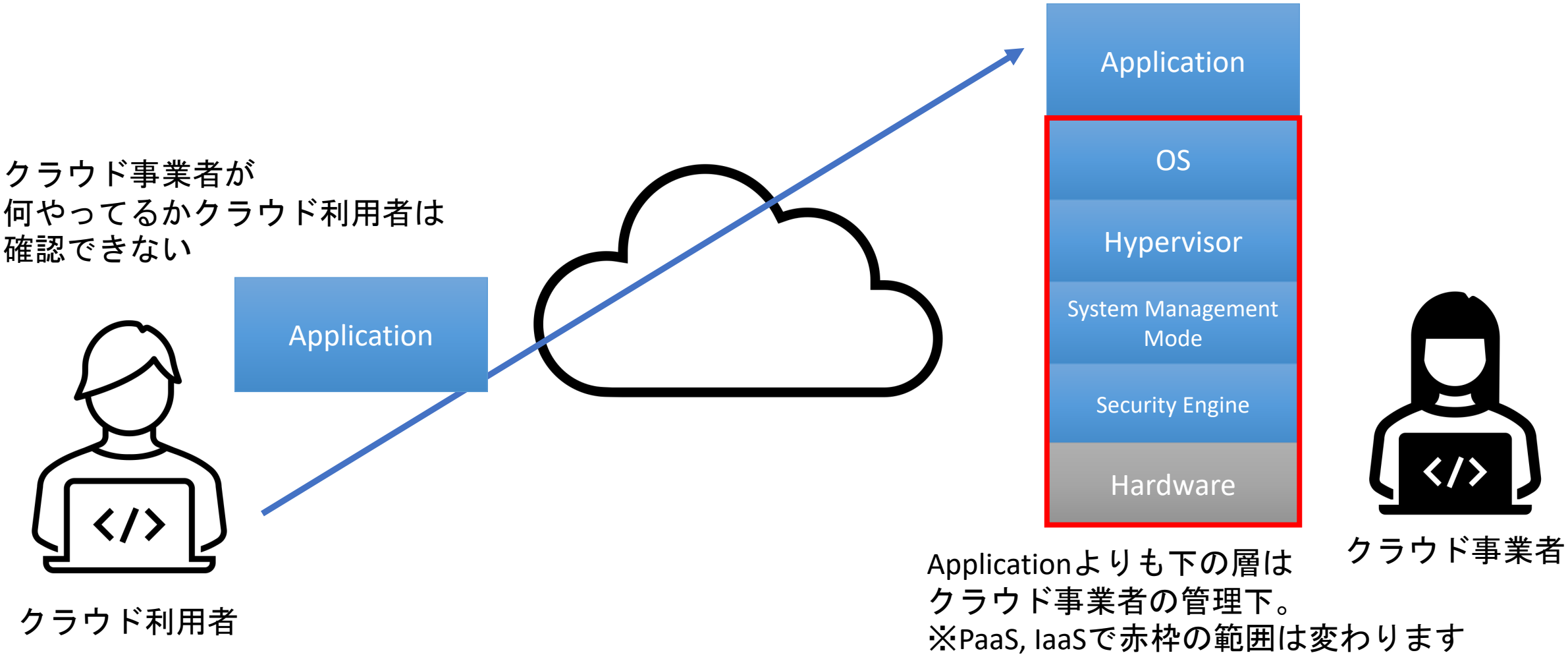
※アプリケーション開発者（利用者）は赤枠を信頼する必要がある。

Szefer, J. (2018). Principles of Secure Processor Architecture Design.

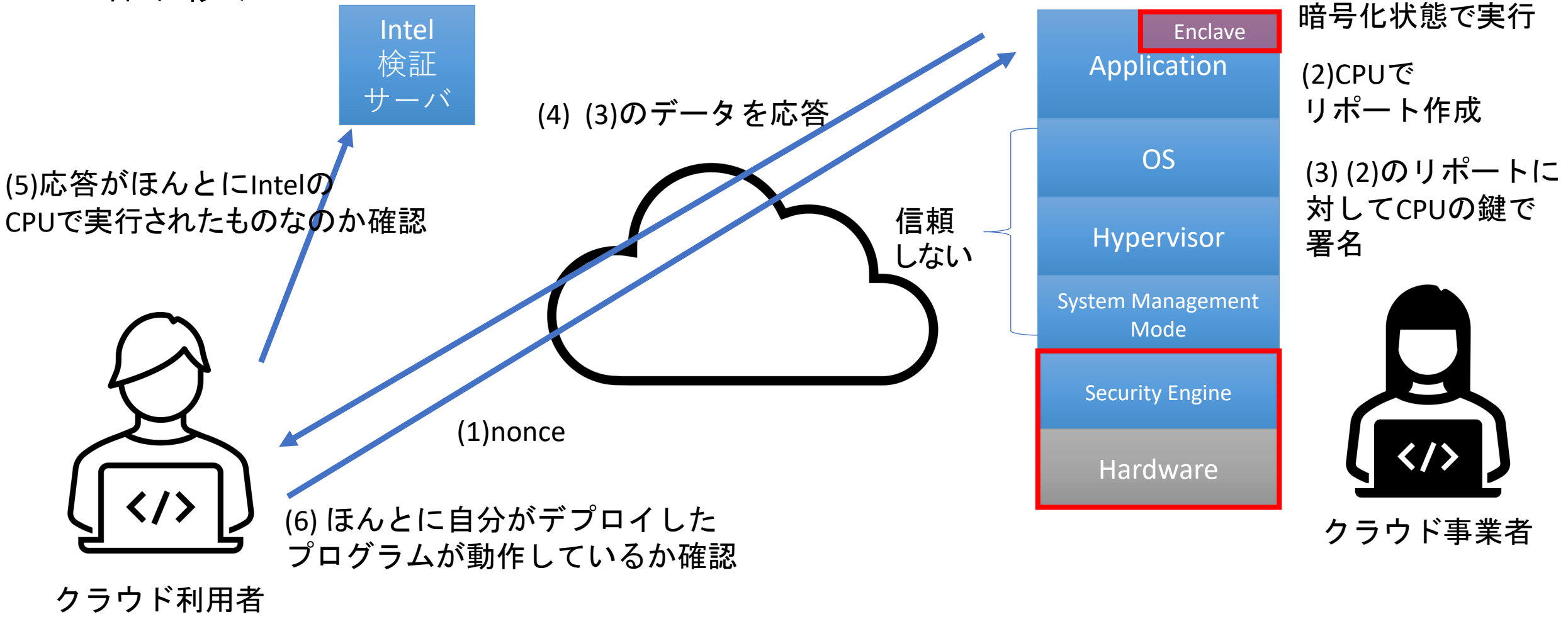
2000年前後はPCやモバイル端末上に Normal（オープン）な実行環境と Secureな実行環境を同居させたかった



2000年後半以降のクラウドの普及
クラウド事業者を信頼してアプリ実行



クラウド事業者よりもチップベンダーを信頼するモデル



本日の流れ

- トラストの技術を活用した具体例：DRM
- 歴史
  - ICカード（SmartCard）の歴史
  - モバイルの歴史
  - PC/サーバの歴史
- Intel SGXの特殊性
- **まとめ**

本講演でのTrust

Trust ＝想定したプログラム（データ）だけが動く

Trustを確立する技術

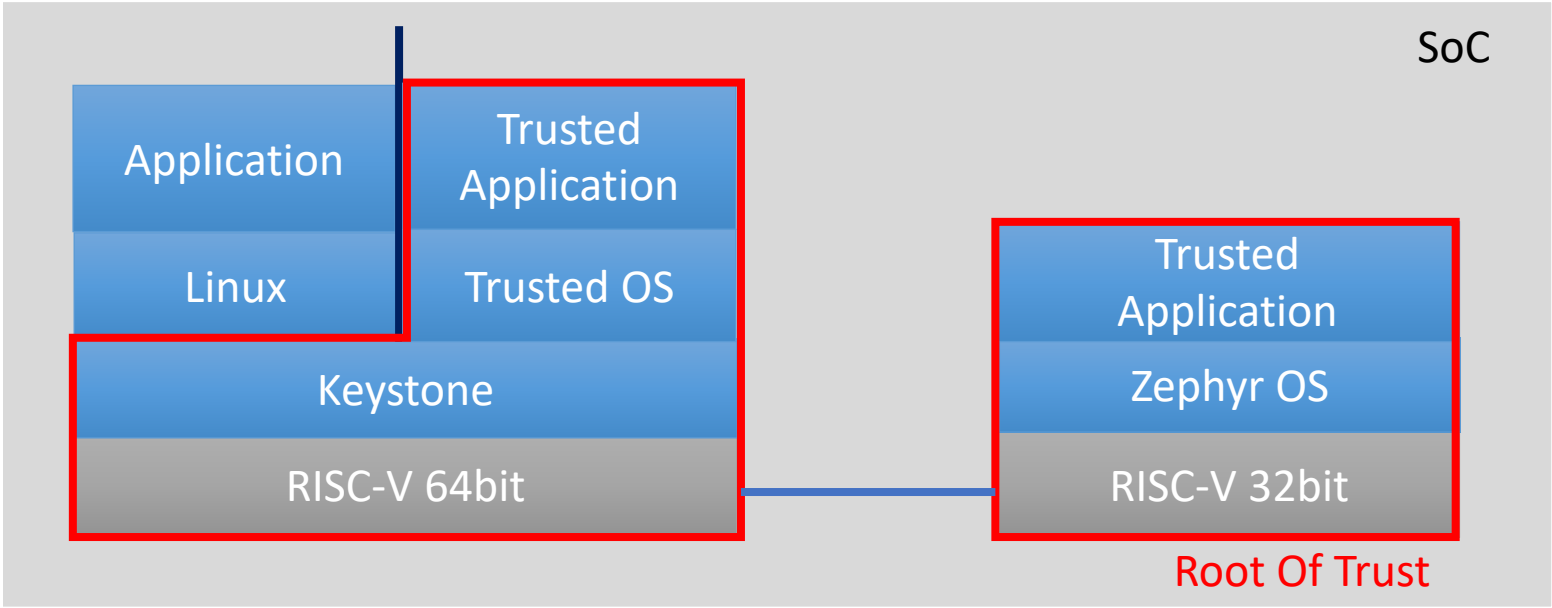
- ある時点から1ビットも変更されていないことを確認してから実行できる技術（完全性）
- 誰が作ったプログラム（データ）なのか確認できる技術
- 想定したプログラムだけが実行される環境を準備する技術

まとめ

- 我々が利用する、スマホ、PC、クラウドにおいてトラストの技術が活用されている
- その歴史は古く、1960年代からの研究や製品開発の積み重ね
- トラストを確立する重要な技術
  - 完全性確保の技術 (Integrity)
    - Secure Boot/Trusted Boot/xxx Boot
    - Root of Trust
    - 耐タンパ性実現の技術
  - 正当な実行環境であることの伝達 (Remote Attestation)
    - Remote Attestation
  - 実行環境の分離 (Isolation)
    - Trusted Execution Environment (実現方法はまちまち：別チップ、仮想化、Enclave）

宣伝

- NEDOプロジェクト
  - 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
  - RISC-Vを使った、TEE対応CPUと、Root of Trustとなりえるセキュアなマイコンの研究開発（鍵管理についてもセットで研究）



鍵管理

セキュアオープンアーキテクチャ・エッジ基盤技術研究組合
http://trasio.org/home/