ラズパイで
”AUTOSAR Adaptive Platform”
を学習しよう

最終更新日：2022/2/21

東海ソフト株式会社　　竹内 健

自己紹介

竹内　健（たけうち たけし）

- **2016年**
  東海ソフト株式会社　エンベデッド技術部　入社
  車載ソフトの開発に従事
- **2017年**
  名古屋大学　大学院情報学研究科付属組込みシステム研究センター　出向
  AUTOSAR DevelopmentパートナーとしてWG-AP-STチームに所属
  リファレンスソフトのセーフティ関連システムテストを担当
  名古屋大学教育プログラムにてAUTOSAR AP入門講座の講師を担当
- **2021年**
  東海ソフト株式会社　組込み技術部　帰任
  AUOTSAR CPなどの車載ソフト開発に従事

本資料について

- Copyright (C) 2018-2020 by
  Center for Embedded Computing Systems
  Graduate School of Informatics,
  Nagoya Univ., JAPAN

- 本ドキュメントは，
  “AUTOSARアダプティブプラットフォームに関するコンソーシアム型共同研究“
  において開発したものである.

- 本ドキュメントの著作権は，上記の著作権者に帰属する.

はじめに

- AUTOASR Adaptive Platform（AP）では,仕様書に加えてリファレンスソフトとして“Demonstrator”もリリースされている.

- この“Demonstrator”を用いてラズパイ上にAPの構築を行い,Adaptiveアプリケーション開発の理解を深める事ができる.

- 本講演では,ラズパイによるAPの構造やアプリケーション開発方法の学習環境をサンプルのデモアプリケーションを通じて紹介する.

アジェンダ

- AUTOSARの概要
- AUTOSAR APの概要
- リファレンスソフト“Demonstrator”について
- ラズパイでの仕様確認

AUTOSARの概要

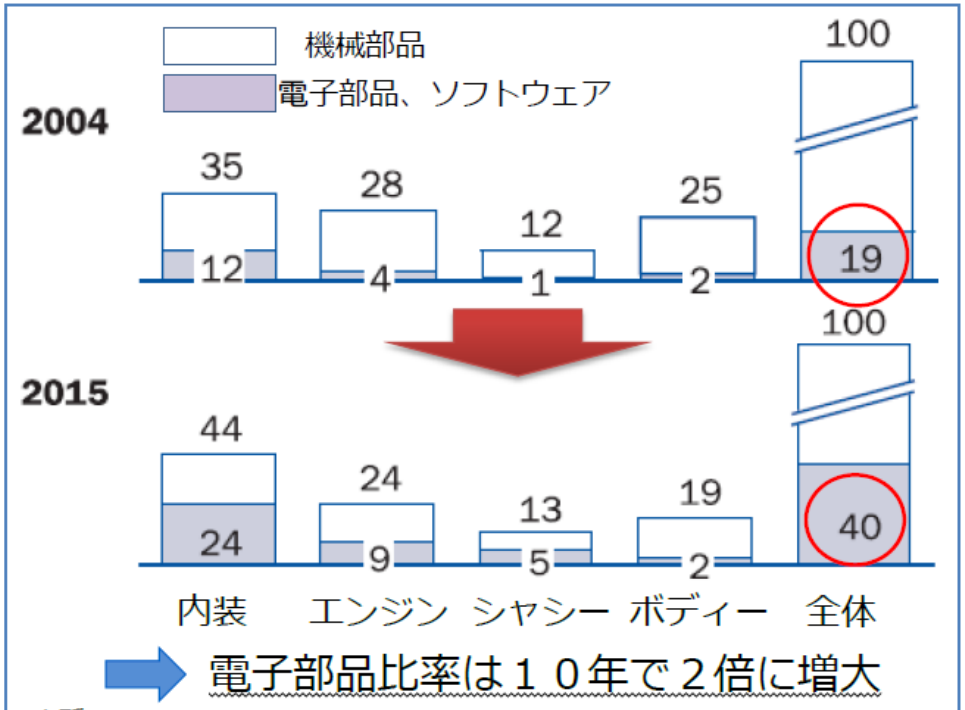
- AUTOSARとは？
  - **AUT**omotive **O**pen **S**ystem **AR**chitecture の略
  - 2003年，ドイツの自動車メーカ(OEM)とTier1サプライヤを中心に設立された，車載ソフトウェアプラットフォームの仕様の標準化を進める団体
  - 現在は，世界各国のOEM，Tier1サプライヤ，半導体メーカ，ソフトウェアベンダ，ツールベンダ等が参加
- メンバ構成（2022年2月現在：３１８社）
  - コア パートナー（９社）：BMW，Bosch，Continental，Daimler，Ford，General Motors，PSA Group，Toyota，Volkswagen
  - ストラテジック パートナー（１社）
  - プレミアム パートナー（５９社）
  - デベロップメント パートナー（６９社）
  - アソシエイト パートナー（１５１社）
  - アテンド（２９社）
- 標準化ソフトウェアプラットフォームの種類
  - Classic Platform（以後CP）2003年～
  - Adaptive Platform（以後AP）2016年～

AUTOSAR
Enabling continuous innovations

[参考]ソースコードの増加

- 車載のソースコードは以下のように年々増加している.
  - 2020年における自動車のソフトウェアコード行数は2億行程度である
- 今後，自動運転が実現されるとさらに増加することが予想される

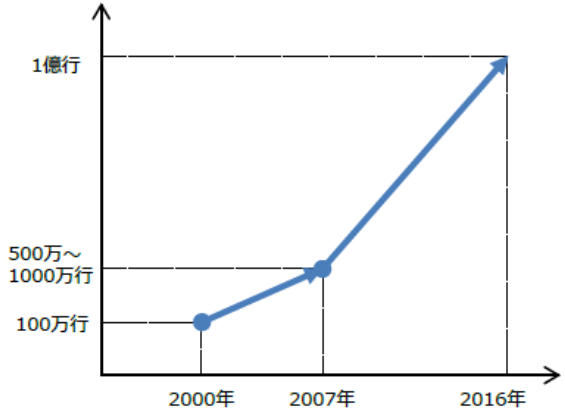
<電子化の進展>



出所：McKinsey “Managing innovations on the road”
※電子部品比率は、車両を構成する部品の総原価に占める電子部品の原価の割合

<ソフトウェアの複雑化>

■自動車ソフトウェアのソースコード行数
➢ 平成12年時点では100万行程度だったものが、現在では１億行以上という規模まで増大。



<参考：他製品のソースコード行数>
- Android OS：1,200万行
- F-35戦闘機：2,400万行
- Microsoft Office 2013：4,400万行

経済産業省 自動車新時代戦略会議（第１回）資料より
経済産業省 「第四次産業革命スキル取得講座認定制度」に関する検討会資料より

APの概要

- **開発背景**
  - CPは，性能が限られ，かつ高い安全性が求められる，従来のECU向けのソフトウェアプラットフォームとしてデファクトスタンダードとなっている
  - 先進運転支援システム（ADAS）や自動運転など，高機能・高性能で，かつ高い安全性が求められる**次世代ECU向け**として，CPよりも，**機能が豊富で柔軟性の高い**ソフトウェアプラットフォームに対する需要が高まってきた
  - AUTOSARでは上記に対応したソフトウェアプラットフォームをAdaptive Platform（AP）と位置付け，開発を開始した

APの特徴

- **C++言語の採用**
  - パフォーマンスが重視される複雑なアプリケーションにおいて，新しいアルゴリズムやアプリケーションソフトウェアの開発に最適な言語であり，APもこれに適応する
- **SOAの採用**
  - Service-Oriented-Architecture：サービス指向アーキテクチャ
  - サービスはアプリケーションが実行されるローカルECU上だけでなく，APが動作するリモートECU上にあってもよい
- **並列処理への対応**
  - 上記のSOAの仕組みは，様々なアプリケーションが異なるサービスを利用でき，並列処理を行うことができる
  - APは並列処理能力を提供する多コアプロセッサ/異種コンピューティング技術の進歩に対応するためのアーキテクチャを備える

APの特徴

- **既存スタンダードの活用**
  - 既存のオープンスタンダードを再利用して適応させる戦略を取り，AP自体の迅速な開発を促進する
- **安全とセキュリティのサポート**
  - APの目標とする多くのシステムは，おそらく最高レベルの安全性とセキュリティを必要とするので，アーキテクチャ，機能，手順の各アプローチを組み合わせて，これらの課題に対処する
- **動的リソースによる開発(Planned dynamics)をサポート**
  - ソフトウェア開発とインテグレーションの労力を削減するために，リソースと通信を動的に管理し，開発サイクルを短くすることでアプリケーションの追加実装をサポートする
  - 実装後はマニフェストに従って動的に動作する
- **アジャイル開発を想定**
  - アジャイル開発を想定し，システムの根本となるアーキテクチャーを段階的に拡張，システム更新をできるようにする

APの全体構造

- Adaptive Platformは,
”Adaptive Platform Foundation”と
”Adaptive Platform Services”で構成され,
それぞれに属する機能毎にブロック分けされた,
複数の**機能クラスタ(FC)**が存在する

- これら二つが持つインタフェースを合わせて
AUTOSAR Runtime for Adaptive Applications(ARA)
と呼ぶ．また，ARA上で動くアプリケーションは,
“Adaptive Application(AA)”と呼ばれ，ARAから
提供されるサービスを用いて動作する

APの全体構造

<Adaptive Platformアーキテクチャ>

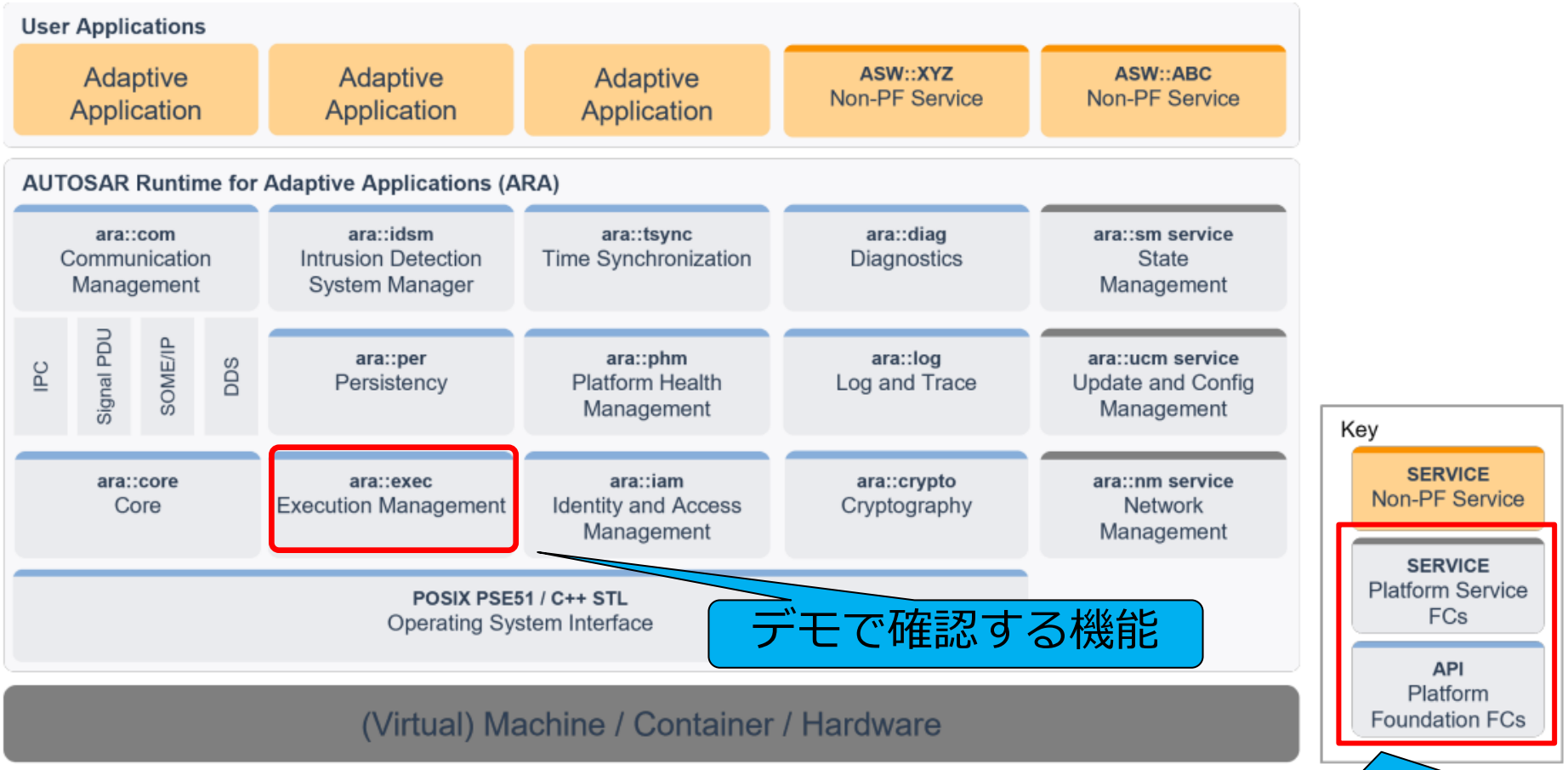
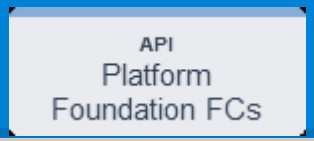


Figure 4.1: AP architecture logical view
(AUTOSAR_EXP_PlatformDesign.pdf)

上の2つをFunctional Cluster
(機能クラスタ)と呼ぶ

各機能クラスタの役割



- 各機能クラスタは，以下の役割を持つ

[Platform Foundation]

[]の中はクラス名を示す

- Communication Management(CM)[ara::com]
  アプリケーション間の通信を担当

- RESTful Communication[ara::rest]
  CMと同様通信を担当するが，CMと異なり以下の特徴を持つ
  - 非AUTOSARとの通信が可能
  - CMのような通信ではなく，車外通信を担当

- Time Synchronization[ara::tsync]
  異なるアプリケーションやECU間の時刻同期を担当

- Persistency[ara::per]
  不揮発性メモリに情報を格納する機能を担当

各機能クラスタの役割

API
Platform
Foundation FCs

- 各機能クラスタは，以下の役割を持つ

[Platform Foundation]

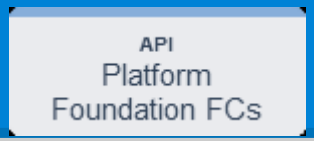
- Platform Health Management[ara::phm]
  プラットフォームの状態監視を担当

- Core Types[ara::core]
  複数の機能クラスタで使用するパブリックインタフェースを定義する

- **Execution Management(EM)[ara::exec]**
  プロセスの起動/停止等の管理を行う

- Identity Access Management(IAM)[ara::iam]
  外部からの攻撃で，AAが侵害された場合に各FCのインタフェースを用いたアクセスを制限するためのフレームワーク

各機能クラスタの役割



- 各機能クラスタは，以下の役割を持つ

[Platform Foundation]

- Operating System Interface(OSI)
  全てのアプリケーションの実行時リソース管理を担当

- Logging & Tracing[ara::log]
  デバック情報を外部のデバイスに送信するための機能を担当

- Cryptography[ara::crypto]
  一般的な暗号化に関する機能を担当
  (鍵の動的な生成 / 鍵の管理)

各機能クラスタの役割

SERVICE
Platfrom Service
FCs

- 各機能クラスタは，以下の役割を持つ

[Platform Service]

- State Management(SM)[ara::sm service]
  AP全体の状態を管理
  複数の状態変更要求等の調整/仲裁を担当

- Diagnostics[ara::diag service]
  ISO 14229-5(UDSonIP)実現を担当
  (インターネットにおける統一診断サービス)

- Signal to Service Mapping[ara::s2s service]
  シグナルでの通信とサービス指向通信のブリッジを担当

- Network Management[ara::nm service]
  ネットワーク管理を担当

各機能クラスタの役割

SERVICE
Platfrom Service
FCs

- 各機能クラスタは，以下の役割を持つ

[Platform Service]

- Update and Configuration Management(UCM)
  [ara::ucm service]
  アップデート機能を担当

OS概要 -機能-

- APとして**OSが指定されているわけではない**
- APではOSI(Operating System Interface)を規定している

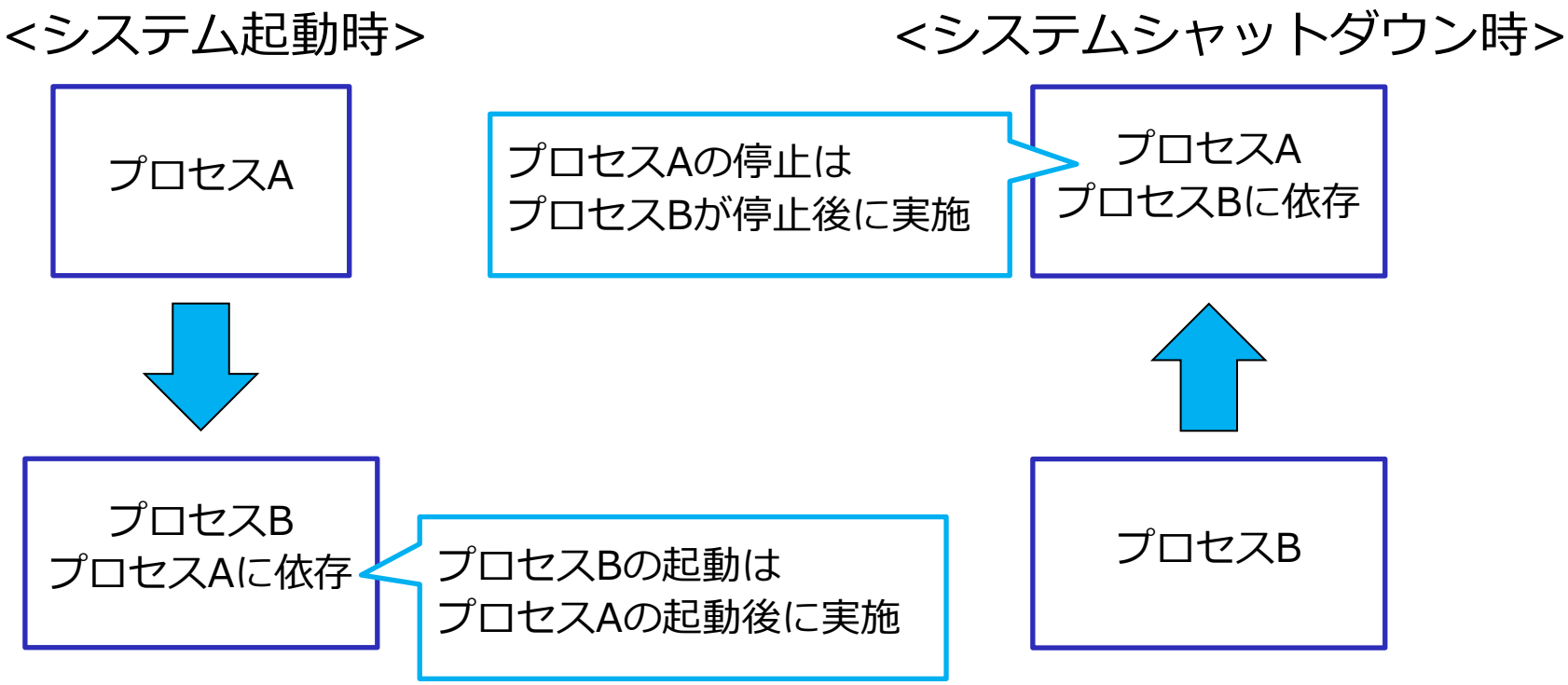
- OSI仕様には以下の内容が含まれる
  - AAが使用するためのインターフェイスについて
    → ARAの一部として，CおよびC++インターフェイスの両方を提供する
  - OSとして必要とされる機能(後述)について
    → これらの機能を提供するインターフェイスは，プラットフォーム実装に依存し，ARAの一部としては利用できない

OS概要 -機能-

- OSとして必要な機能
  - **EMを初期化プロセスとして実行するメカニズムを用いる**こと
  - マルチスレッドをサポートすること
  - OSがプロセスをCPUコアにバインドするメカニズムを提供すること
  - **マルチプロセスをサポートする**こと
  - マルチプロセスで各プロセスが互いに分離されていること(不正アクセスを検知できるように，各プロセスでメモリ空間が分かれていること)
  - 各プロセスが仮想メモリ上で実行されること
  - POSIXで定義されるスケジューリングポリシーをサポートすること
  - タイマ契機での実行をサポートすること(POSIXタイマ，シグナル等をサポートすること)
  - POSIX PSE51で定義されているデバイスのサポートをすること

EM概要 -起動とシャットダウン-

- EMが起動し各プロセスを起動する際，依存関係に基づいて起動とシャットダウンが行われる
- EMはプロセス起動前に依存関係の確認を行う
- **依存関係がない場合は，任意の順序**で行われる



EM概要 -エグゼキューションマニフェスト-

- プロセスの管理は，「**エグゼキューション マニフェスト**」を用いて行われる
- エグゼキューションマニフェストには，**プロセスをどのように動かすのか？**といった内容を記載する

- エグゼキューションマニフェストの記載内容
  - 実行依存関係
    依存するプロセスを指定できる
    ex.) AプロセスがX状態になった時にBプロセスを起動
  - アプリケーションの引数
  - 起動するマシン状態を指定
  - スケジューリングポリシー
  - スケジューリングの優先順位
  - アプリケーションのバイナリ名(実行ファイルの名前)

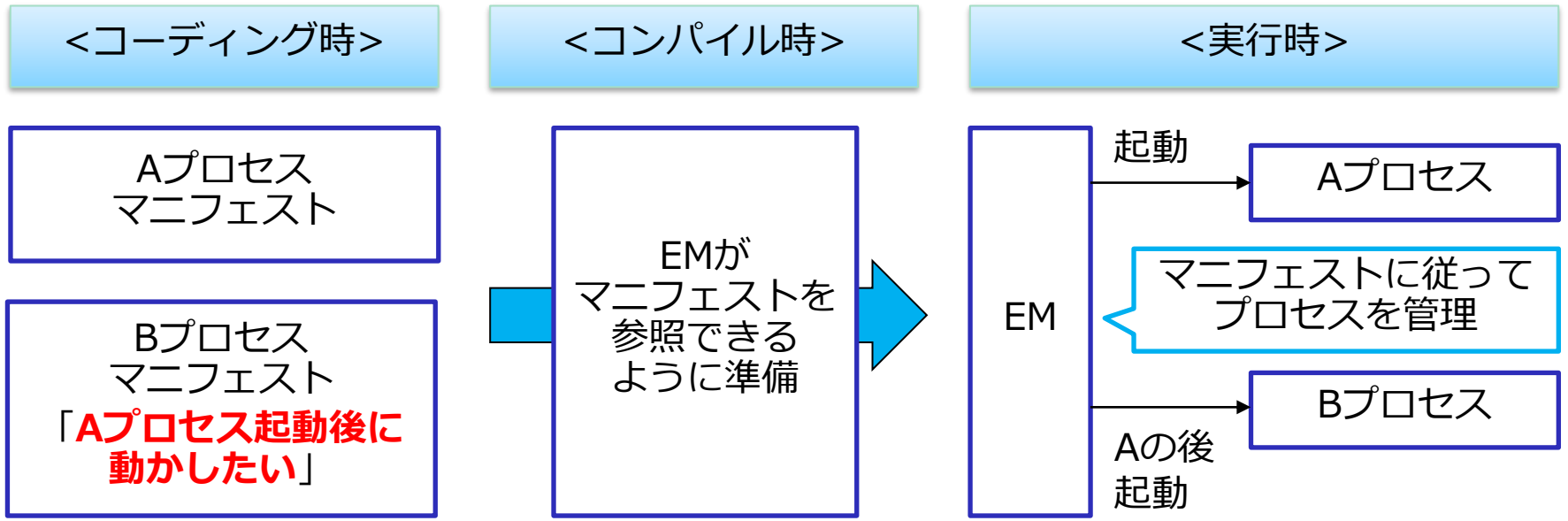
EM概要 -エグゼキューションマニフェスト-

- エグゼキューションマニフェストの記載方法
  - どのように実行依存関係を記載するか？はAPの実装により変化する
  - ex)
    A社のAPでXXXというフォーマットでマニフェストを記載
    →A社と異なるAPでは動かない可能性がある
  - AUTOSARの仕様では，記載内容の指定はあるが記載方法の指定までは存在しない

EM概要 -動作イメージ-

- EMはエグゼキューションマニフェストにより，各プロセスがどのように動作したいかを把握する
- EMが管理するのは，プロセスの起動と終了であり，プロセスが立った後の管理は**プロセス内で自己完結**する必要がある

<EM動作イメージ>



リファレンスソフト“Demonstrator”について

- 複雑な仕様を理解するためのデモアプリ
- AUTOSAR会員にのみリリースされている
- ビルドツールにYoctoを使用
- 今回のデモはWindows上にVirtualBoxで仮想Linux環境を作成し、Yoctoビルドを行った
- “Demonstrator”は様々な機能が動作するため、今回のデモではEMに絞って機能を抽出した

機能抽出のイメージ

<Adaptive Platformアーキテクチャ>

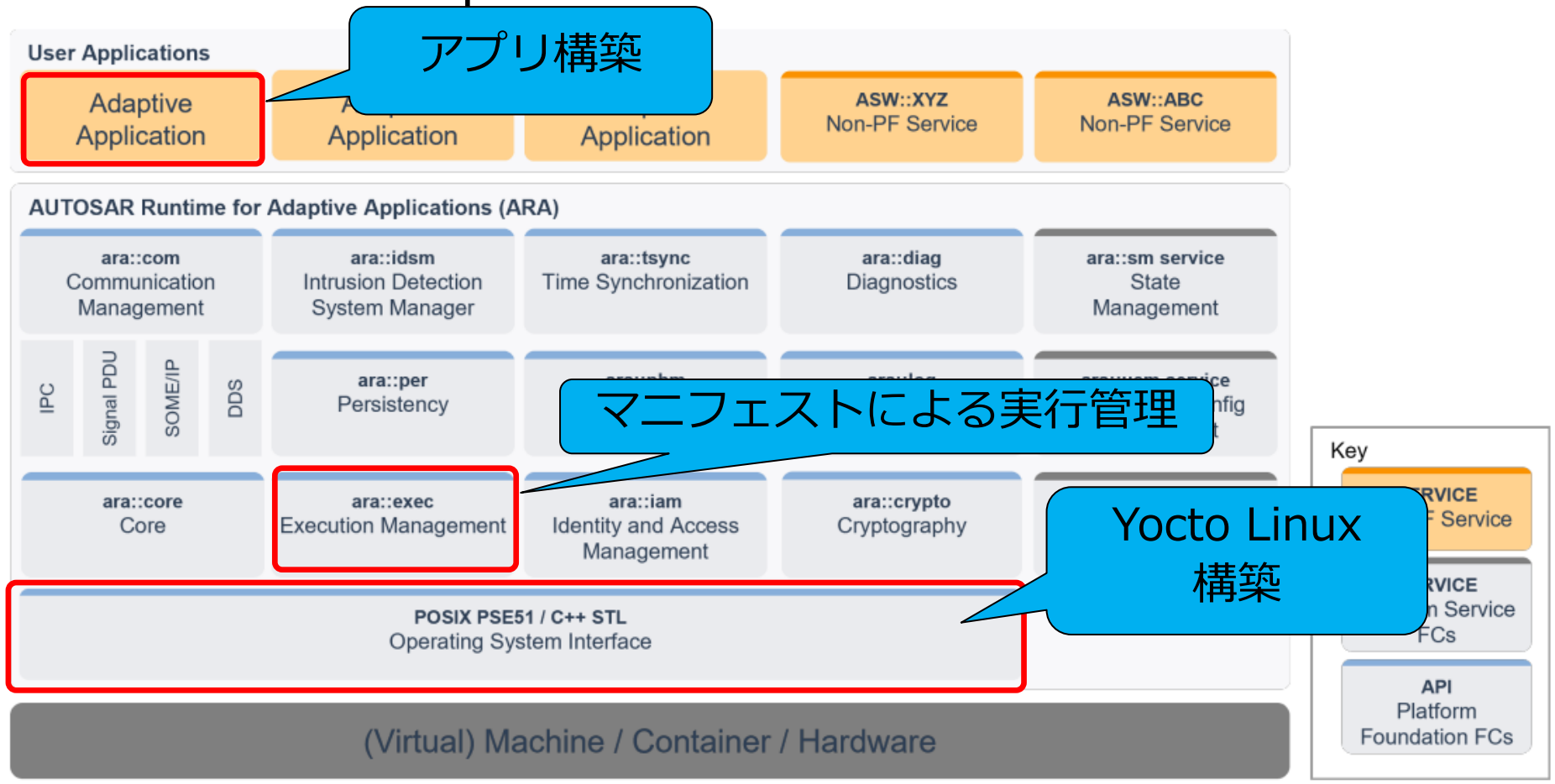
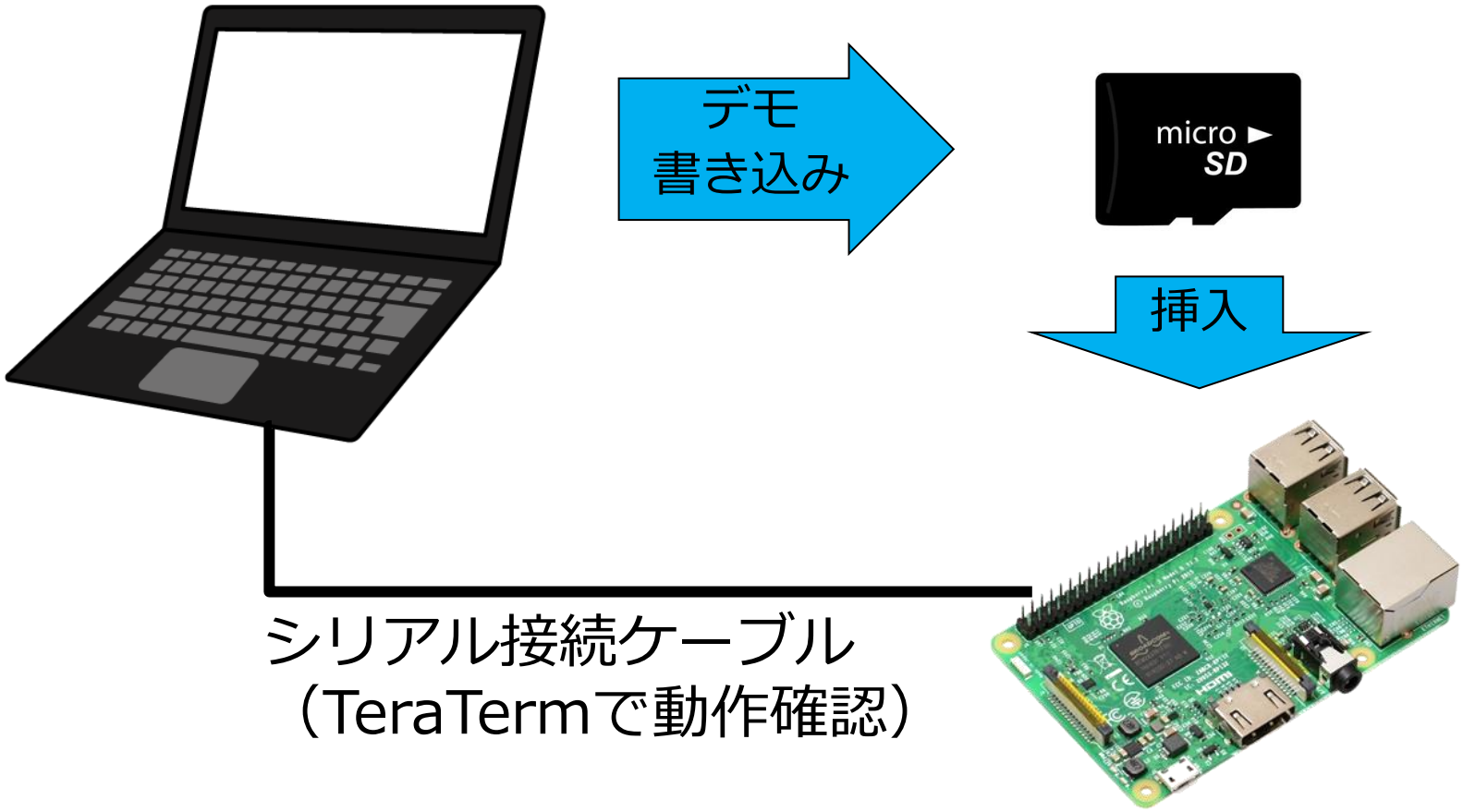


Figure 4.1: AP architecture logical view
(AUTOSAR_EXP_PlatformDesign.pdf)

デモのハード構成



ラズパイでの仕様確認項目

1. ラズパイにイメージをデプロイした後にプロセスの起動順を確認する
2. マニフェストを変更し,再起動を行う
3. 変更したマニフェスト通りにプロセスの起動順が変更されていることを確認する

ラズパイで仕様を確認してみて感じたこと

仕様理解に加えて,AUTOSAR APのアプリケーション開発には、EMに限らず以下が必要

- マニフェストの理解
- C++の概念理解

“Demonstrator”に関して

- Linuxの理解
- Yoctoの理解

AUTOSAR AP 講習 のご案内

AP仕様の詳細を学習したい方
APソフト開発を試してみたい方

①AP概要 講習

- AP仕様の説明・解説を座学で実施

②AP演習 講習

- Raspberry Piを使用した環境構築
- サンプルソフトの実装演習,動作確認
- ニーズに合わせ内容のカスタマイズ対応可能

※受講条件：AUTOSAR会員企業であること

ご清聴いただきありがとうございました

本講演に関するお問合せ
E-mail: info@tokai-soft.co.jp
Tel: 052-300-8331