DENSO
Crafting the Core

# AUTOSARの基礎

山本健太

株式会社　デンソークリエイト

# アジェンダ



1.自己紹介

2.はじめに

3.開発の流れ

4.ツールチェーン

5.AUTOSAR固有用語説明

6.BSW説明

7.まとめ

# 自己紹介



山本健太

株式会社デンソークリエイト所属

- 2010年度入社
- 2014年～ AUTOSAR関連の仕事に従事

LED-Camp

- 2015年 LED-Camp3に参加
- 2016年～ LED-Camp実行委員

# 1. はじめに

1. 本講座について
2. AUTOSAR概要

# 本講座について



## 本講座の目的

- AUTOSARに関する基礎的な知識を獲得すること
  （AUTOSAR詳細理解への弾み、ECU開発に向けての導入となる）

## 本講座のゴール

- AUTOSARの導入・使用に対する目的やうれしさを知る
- AUTOSARの開発の流れ、ツールに関する情報を知る
- AUTOSARの基本的な機能、構成および用語を理解する

## 対象のAUTOSAR仕様バージョン

- Release 4.3（20.7時点での最新はAR19-11）

## 参考文献

- [1]：経済産業省　自動車新時代戦略会議（第１回）資料
  https://www.meti.go.jp/committee/kenkyukai/seizou/jidousha_shinjidai/pdf/001_01_00.pdf

# AUTOSAR概要



AUTOSAR (The **Aut**omotive **O**pen **S**ystem **A**rchitecture)とは

- 設立：
  - 自動車メーカ・サプライヤ企業(部品メーカ)を中心に作られたコンソーシアム
  - '03年7月、DaimlerChrysler社やBMW、Bosch等欧州メーカが中心
- 目的：
  - 車載電気・電子システムアーキテクチャにおけるオープンな**業界標準の構築**

コアパートナー：組織(WG)の運営、管理を行う。仕様の決定権を持つ。

プレミアムメンバー：WGへの参加、策定中仕様へアクセスする権利がある。

ディベロップメントメンバー：専門知識を持ち、策定された仕様のアクセス、利用が可能。


ストラテジーパートナー：コア、プレミアムと共に仕様を策定する

アソシエイトメンバー：策定された仕様のアクセス、利用する権利が与えられる。

アテンディ：WGへの参加や協力、策定中の情報へのアクセスが可能。

出典：AUTOSAR「AUTOSAR Introduction」

# AUTOSAR概要



## AUTOSARの特長


**3つの“標準化”**

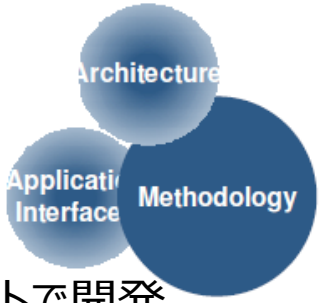
- AUTOSAR方法論　　- AUTOSAR Methodology –
  … 開発方法論の標準化

- アプリケーションインタフェース　- Application Interface –
  … 機能間でやり取りする情報の標準化

- レイヤードアーキテクチャ　- Layered Architecture –
  … ソフトウェアアーキテクチャの標準化

# AUTOSAR概要

CONFIDENTIAL
関係者外秘

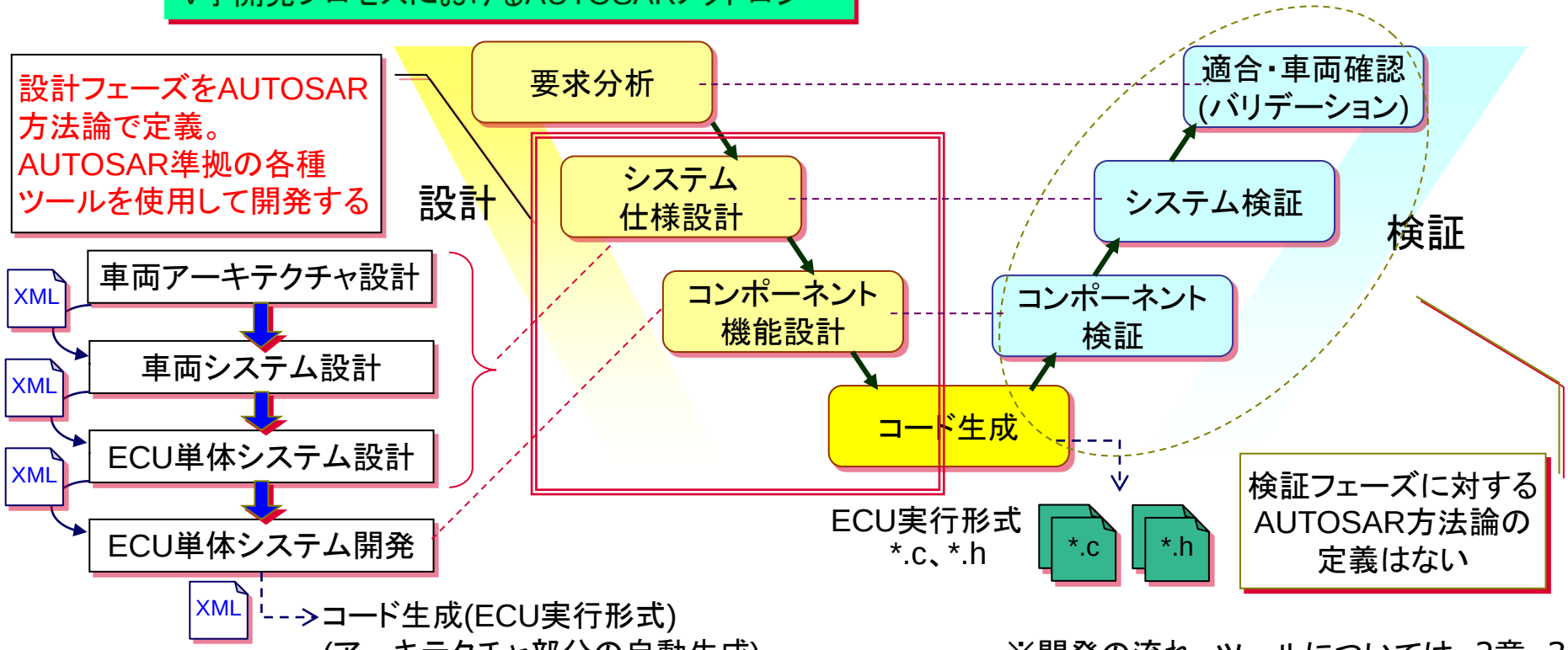



## [特長1] AUTOSAR方法論 - AUTOSAR Methodology -

**開発方法論の標準化**

- AUTOSAR方法論に基づく開発
- 車両全体アーキテクチャから各ECUの設計まで統一の記述フォーマットで開発
  └XMLフォーマット

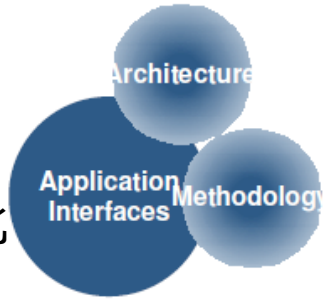
**V字開発プロセスにおけるAUTOSARメソドロジー**




※開発の流れ、ツールについては、2章、3章

# AUTOSAR概要

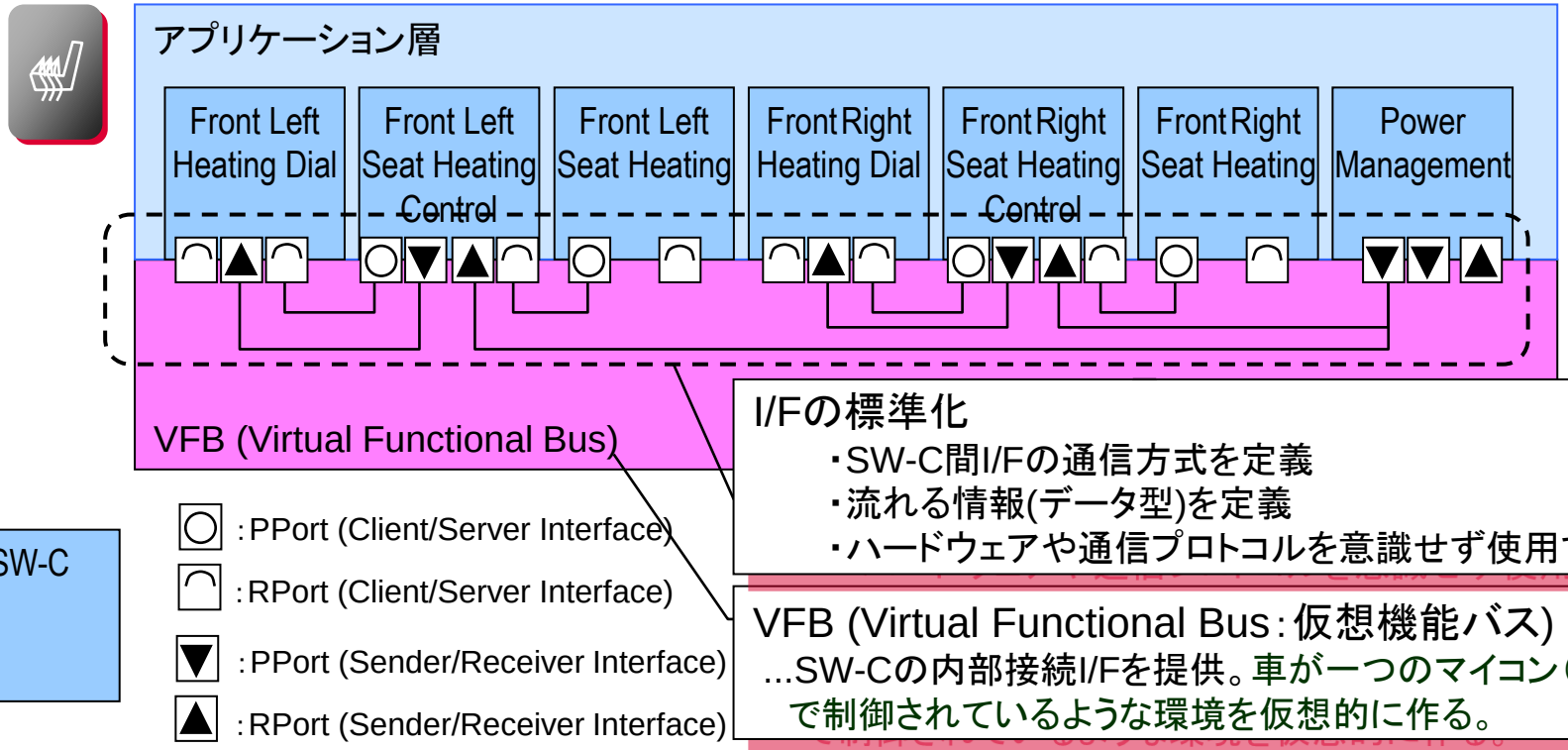
## [特長2]アプリケーションインタフェース - Application Interface -




**機能間でやり取りする情報の標準化**

- アプリケーションを構成する各機能のSW-C間インタフェースを標準化

**アプリケーション例：シートヒーティング制御**




**I/Fの標準化**
- SW-C間I/Fの通信方式を定義
- 流れる情報(データ型)を定義
- ハードウェアや通信プロトコルを意識せず使用できる

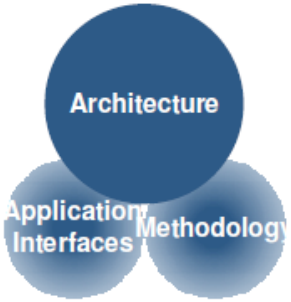
**VFB (Virtual Functional Bus：仮想機能バス)**
…SW-Cの内部接続I/Fを提供。車が一つのマイコン（ECU）で制御されているような環境を仮想的に作る。

※詳細は4章

# AUTOSAR概要

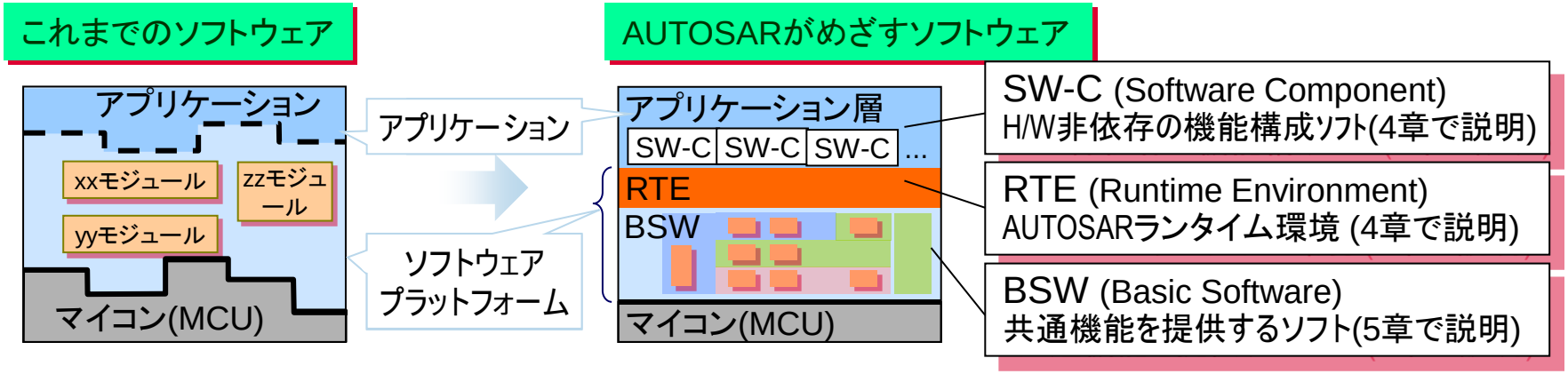


## [特長3]レイヤードアーキテクチャ　- Layered Architecture -




**ソフトウェアアーキテクチャの標準化**

- アプリケーションを再配置、再利用するためのソフト構造を標準化
- 共通ソフトウェアプラットフォームの提供




●メーカ個別のアーキテクチャ

アプリケーション
- ソフトウェアプラットフォーム上に配置
- 個別ソフトウェアプラットフォームとのI/Fを使用(他環境の再利用性が低い)

ソフトウェアプラットフォーム
- 製品独自の構成(モジュール、I/F)
- ハードウェア依存度が高い

●標準化されたアーキテクチャ

アプリケーション(SW-C)
- RTE上に配置
  RTEとのI/Fは共通のため、再配置・再利用が容易

ソフトウェアプラットフォーム(RTE、BSW)
- 部品化、標準化されたソフトウェア
- マイコンの違い、用途の違いに柔軟に対応

# AUTOSAR概要



AUTOSARを導入するメリット

- ソフトウェアの再利用、再配置が容易になる
  前述の「3つの標準化」により…
  - ECUの違いを意識せずにSW-Cを開発でき、汎用性や再利用性が向上
  - BSWを独自に開発しなくとも、標準化された部品として購入可能

  → **品質の向上、コスト削減に貢献できる**

標準化された規格の制約により、個々の部品開発の難易度は上がる。
再利用の向上により、開発トータルとしてのコストは削減可能。

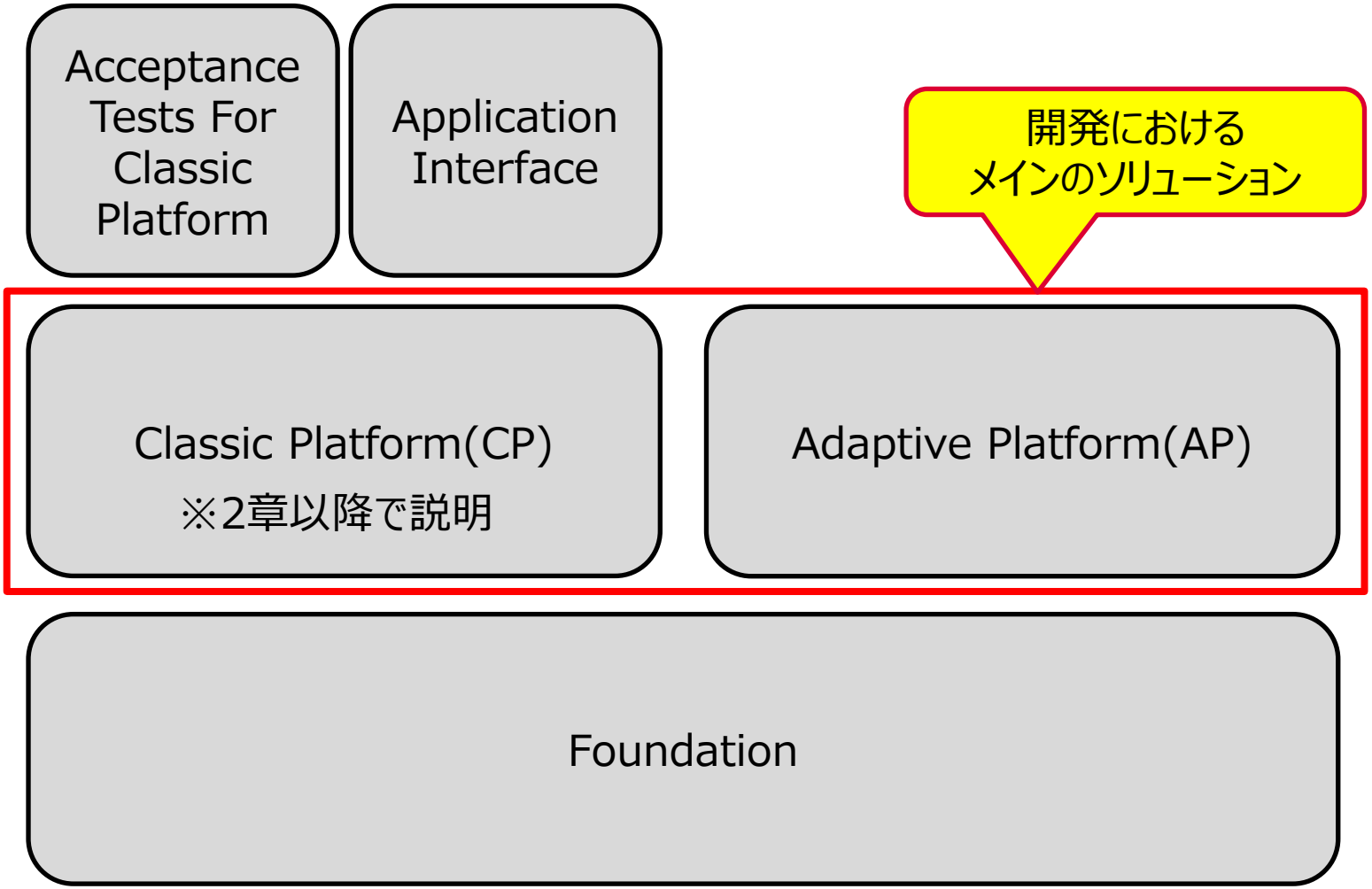
- 欧州を始めとした、OEMメーカ、サプライヤと対等に話ができる
  - AUTOSARの用語や技術について共通認識の下で開発ができる
    → 顧客の要求に対して幅広く対応が可能となる
    競争力の強化に貢献できる

自動車ソフトウェア開発における**標準化**の恩恵を享受できる

# AUTOSAR概要



AUTOSARで提供する標準化のソリューションは以下の5種類



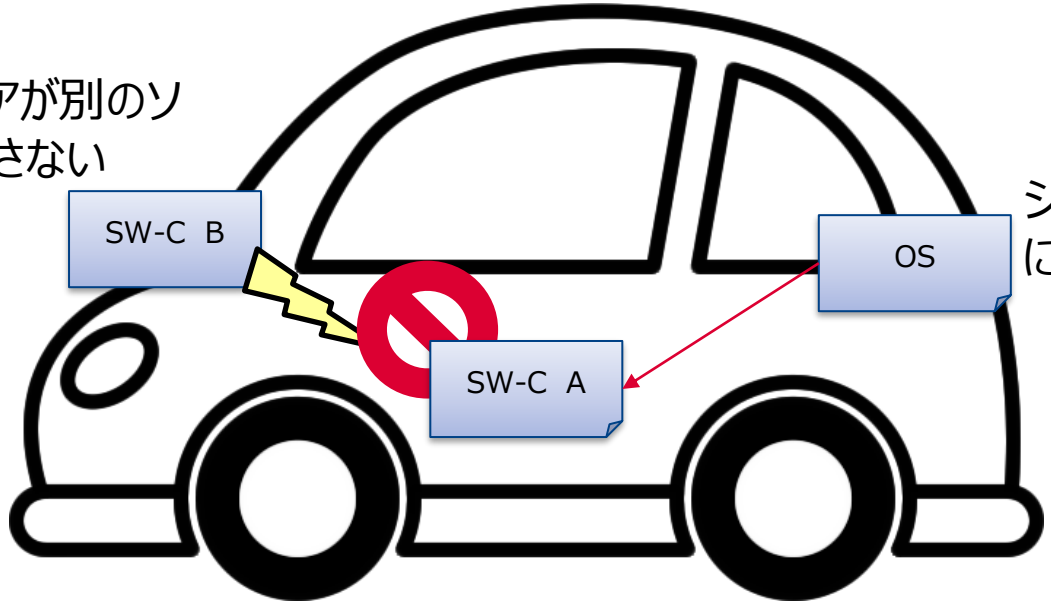
# AUTOSAR概要



## Classic Platform

- 車両内のソフトウェアのリアルタイム性や信頼性を満たしつつ、開発コストや再利用性を高めるために、開発プロセスや仕様の標準化したもの

あるソフトウェアが別のソフトウェアを壊さない




システムは規定時間内に応答を返す

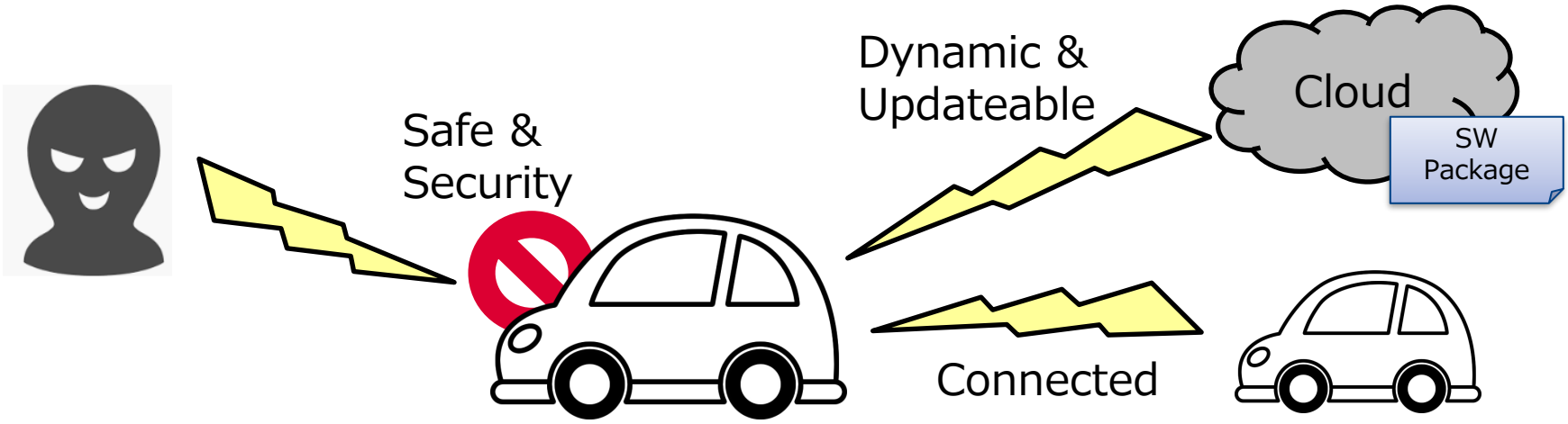
AUTOSARの考えを実現するためのソリューションで、
現在の世界標準となっている

# AUTOSAR概要



## Adaptive Platform

- 運転支援技術や自動運転技術が高まるにつれて、車載内だけに留まらない新しい課題が増えてきた




Classic Platformでは、変化し続けるニーズに対応していくことは難しい

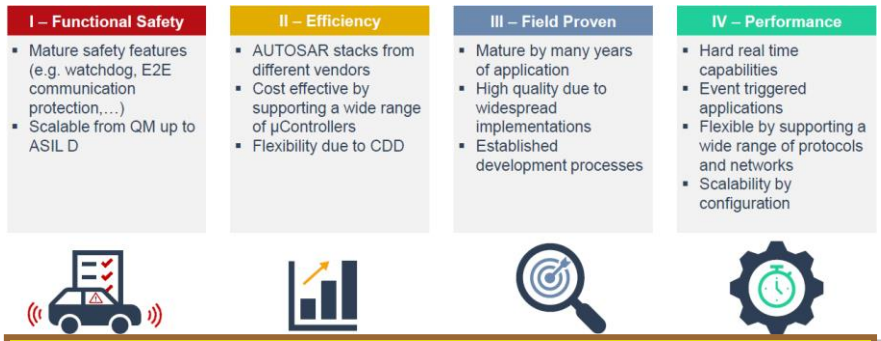
新しいAUTOSARのソリューションとして、
Adaptive Platformを立ち上げ

# AUTOSAR概要



## Classic PlatformとAdaptive Platformの比較

| Classic Platform | Adaptive Platform |
|---|---|
| OSEK OSベース | POSIX OSベース |
| ROMから直接コード実行 | アプリケーションをRAMにロードして使用 |
| すべてのアプリケーションに対してして同じアドレス空間 | 各アプリケーションが自身の(仮想)アドレス空間を持つ |
| シグナルベースの通信 | サービス指向の通信 |
| 固定のタスクコンフィグレーション | マルチで動的なスケジューリング |




リアルタイム性と安全性がある組み込みシステム向けソリューション


Fail-operationを実現するような高性能ECU向けソリューション

出典：AUTOSAR「AUTOSAR Introduction」

# 2. 開発の流れ（CP）

1. AUTOSARを利用したECU開発の流れ
2. 従来開発とAUTOSARを利用した開発の違い
3. まとめ

# 開発の流れ



本章の概要

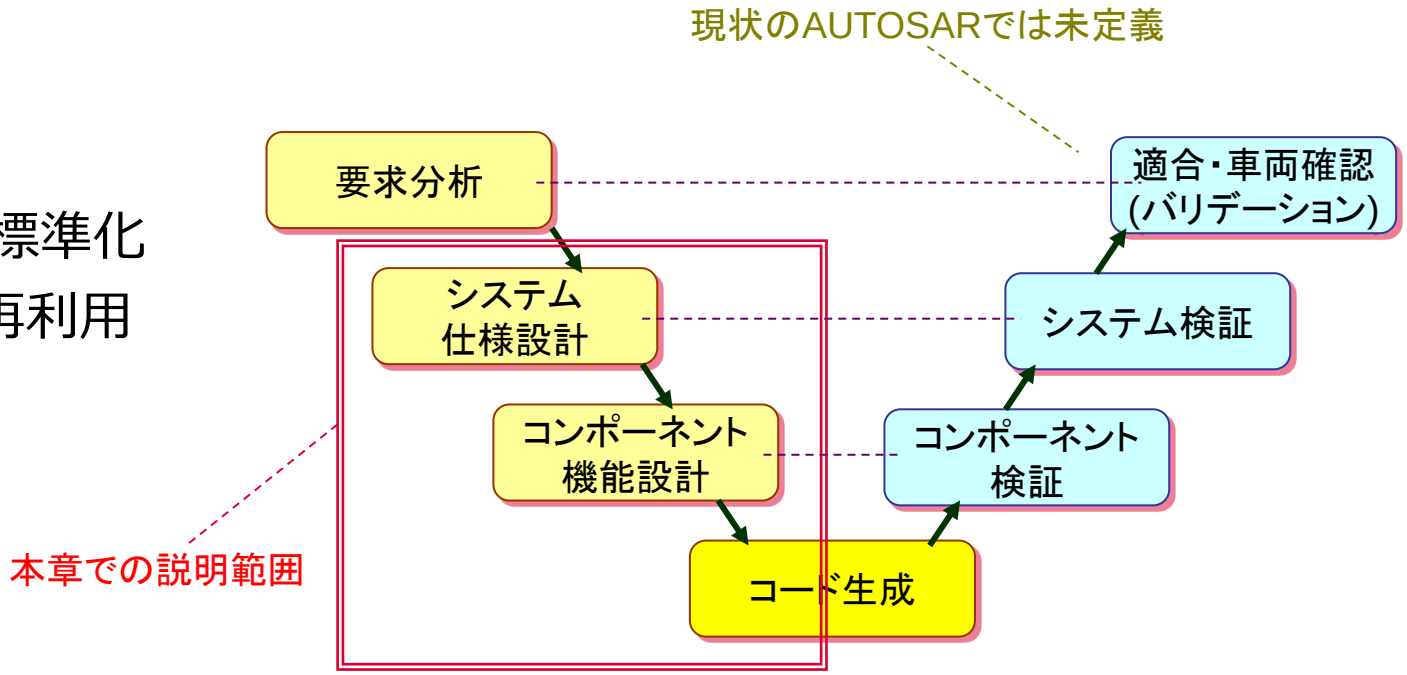
- AUTOSAR方法論で規定されたECU開発の流れを説明する。

ゴール

- AUTOSARを利用したECU開発と、従来のECU開発の違いを説明できること。

キーワード

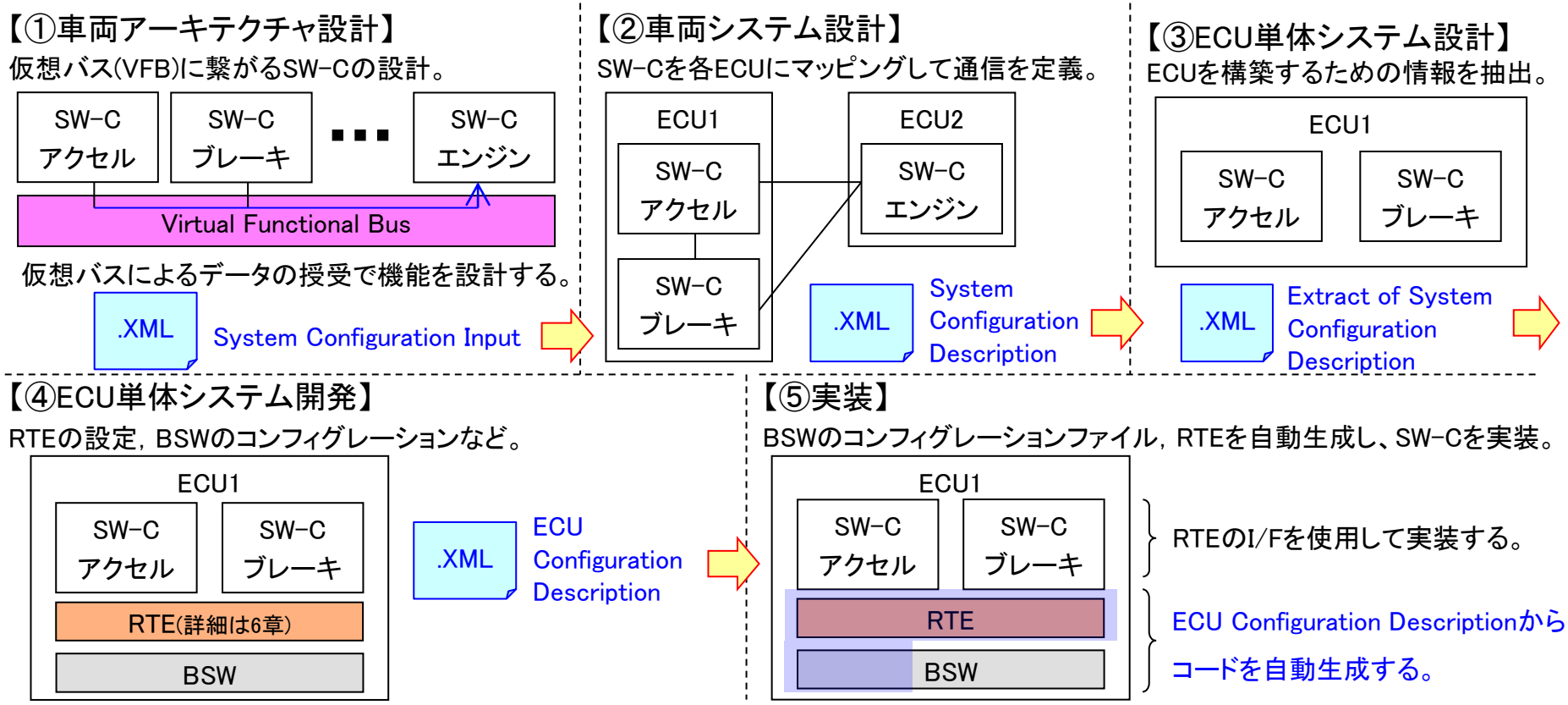
- 成果物の標準化
- SW-Cの再利用



# AUTOSARを利用したECU開発の流れ



上流設計から詳細設計，実装まで一連の流れを標準化し、通信ネットワーク，ECU，マイコンの設定を順序立てて行う
⇒ AUTOSAR方法論に基づく開発の流れ




**規定された形式でXMLファイルを作成し、次工程の入力とする**

# 従来開発とAUTOSARを利用した開発の違い



各設計工程について、従来の開発とAUTOSARを適用した開発の違いに着目して説明する。

- 車両アーキテクチャ設計/車両システム設計
- ECU単体システム設計
- ECU単体システム開発
- 実装

# 車両アーキテクチャ設計/車両システム設計



## 車両全体を設計する

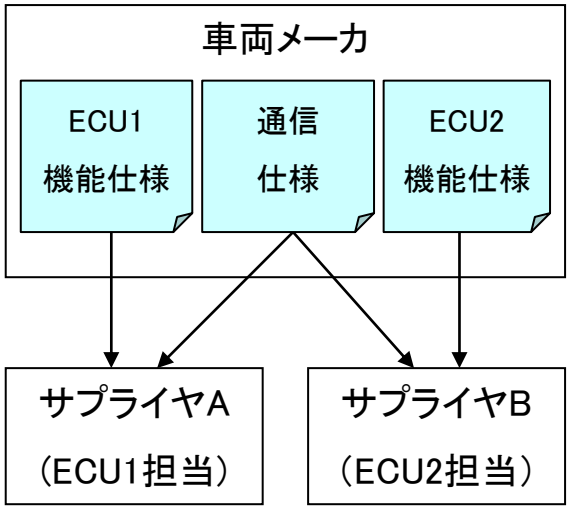
- 車両を構成する機能(SW-C)と機能間I/Fを設計し、SW-CをECUに配置する
  - 配置時は、ECUのハード制約等を加味する。

【従来開発】

成果物：車両メーカ独自形式で定義

WordやExcel，CANdbなど、車両メーカが

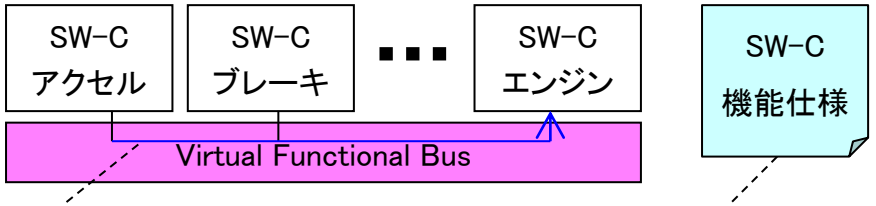
独自のフォーマットでECUの仕様書を提供。




【AUTOSARを利用した開発】

成果物：XMLファイル（System Configuration Input）

<車両アーキテクチャ設計>

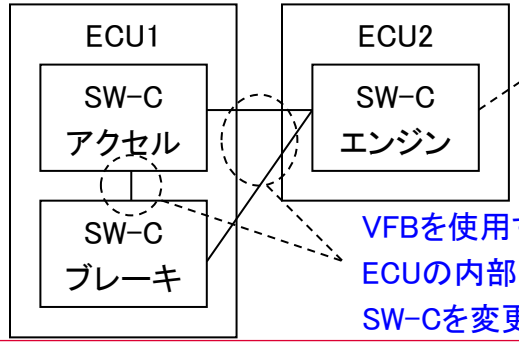



[変化点①]
機能を構成する要素としてSW-Cを定義する。
SW-C間のデータ形式はAUTOSAR仕様に従う。

振る舞いは別文書。
従来開発と同じ。
(ツールで繋ぐなら、
Simulink等を使用可)

<車両システム設計>

成果物：XMLファイル（System Configuration Description）




[変化点②]
SW-CをECUに配置することで、
ECUの持つ機能を定義する。

VFBを使用することで、SW-Cの配置が
ECUの内部⇔外部で変更されても、
SW-Cを変更する必要はない(I/F共通)

# ECU単体システム設計



## ECU単体の情報を抽出する

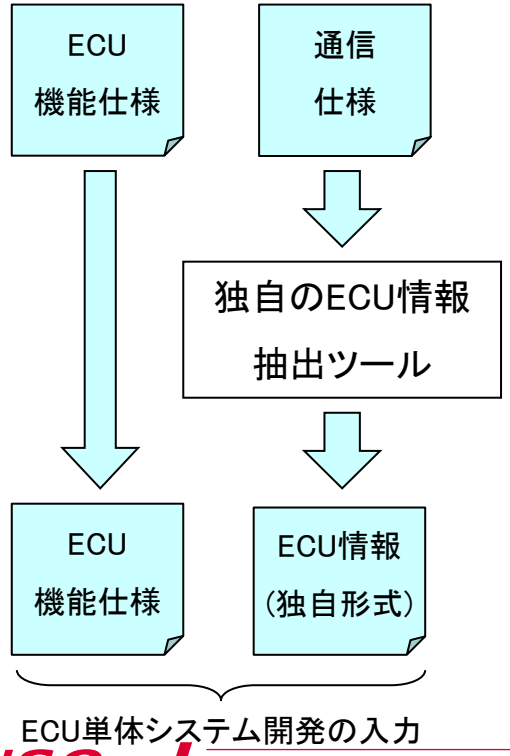
- 車両全体の機能群から開発対象のECUの持つ機能を抽出する。

【従来開発】

成果物：サプライヤ独自の形式で定義

独自開発したツールで通信仕様を抽出。

機能仕様書はそのままECU開発の入力に。

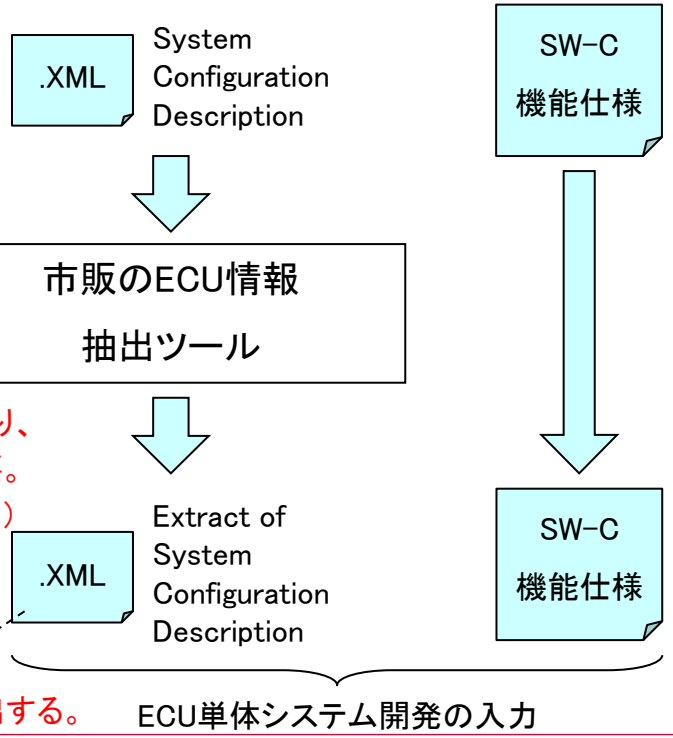



【AUTOSARを利用した開発】

成果物：XMLファイル（Extract of System Configuration Description）

車両システム設計の成果物（車両全体の設計情報）から、

開発対象のECUの情報を専用のツールで抽出する。




[メリット]
XMLの仕様が決まっており、
独自ツールの開発は不要。
（市販ツールを使用できる）

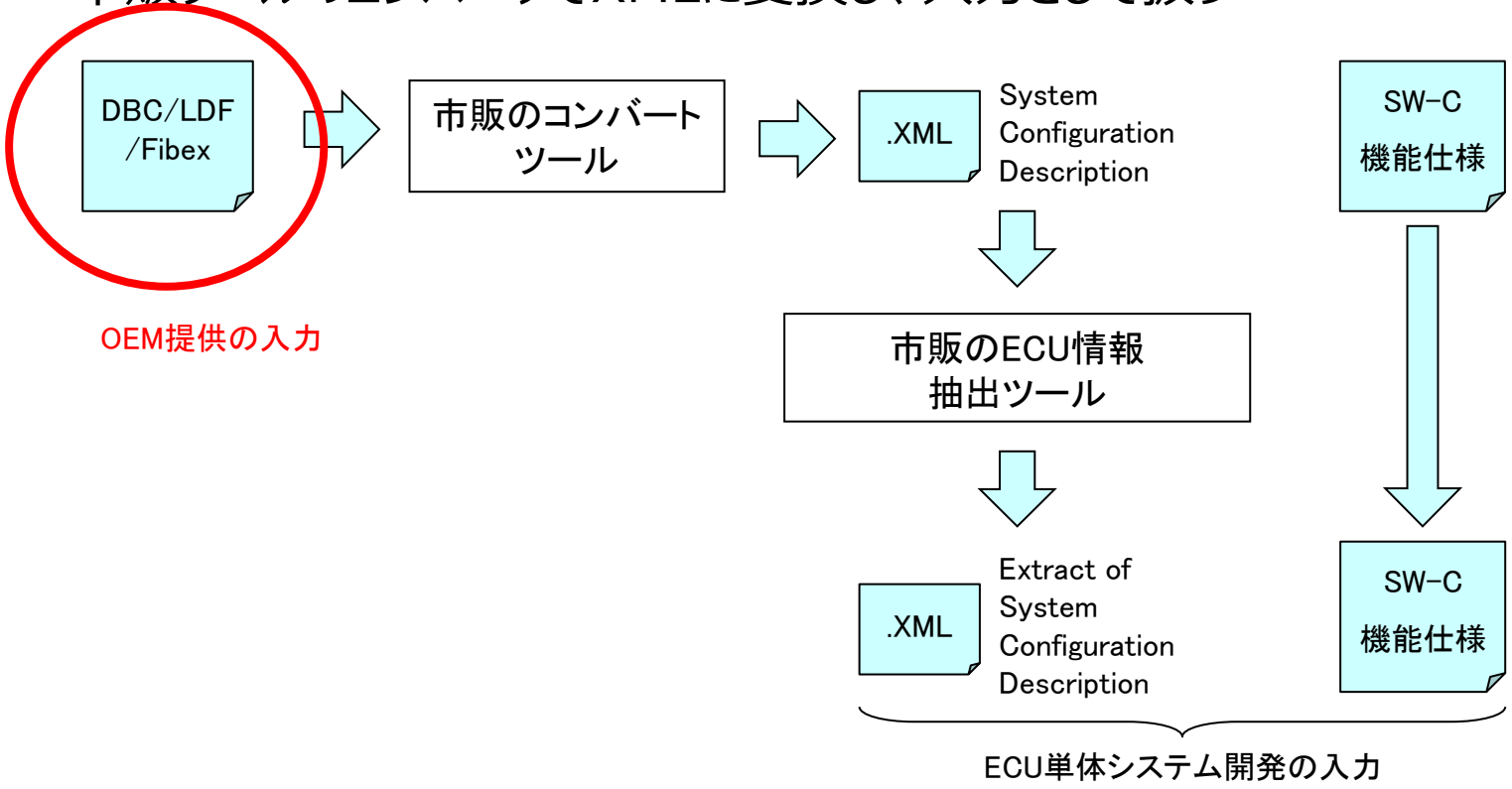
[変化点]
機能(SW-C)の情報も抽出する。

# ECU単体システム設計



OEMによっては従来開発の入力を利用するケースもある

- 従来開発の入力（DBC/LDF/Fibex）がOEMから提供されるため市販ツールのコンバータでXMLに変換し、入力として扱う



# ECU単体システム開発

ECU単体のソフトウェアを設計する

- ECUの機能を分析・設計し、実装できるレベルに落とす。

【従来開発】

成果物：サプライヤ独自の形式で定義

①COM以下のコンフィグレーション

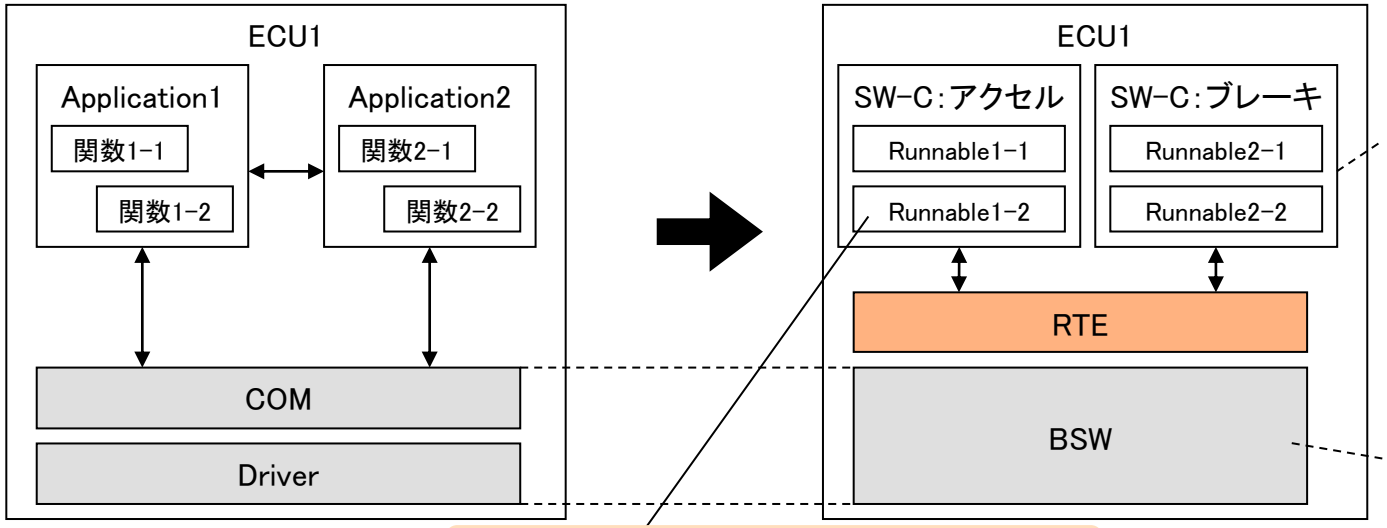
②アプリケーションの構造設計/関数設計

【AUTOSARを利用した開発】

成果物：XMLファイル（ECU Configuration Description）

①BSWのコンフィグレーション

②SW-Cを詳細化、実行条件・タイミング等を設計




[メリット①]
I/Fが標準化されることで、SW-Cを再利用可能になる。
（SW-C同士のI/Fは、ECUの内外を問わず、RTEを介して規定の形式で行うため）

[メリット②]
BSWのコンフィグレーションも市販ツールで可能になる。

Runnable：トリガ発生時に呼び出される関数

（主な設定要素）
- ・Runnableの入出力定義
- ・実行条件の定義
- ・データの定義

[変化点①]
市販のツールを使ってRunnableを設定する。
RTEはこの設定内容から自動生成する。

# 実装

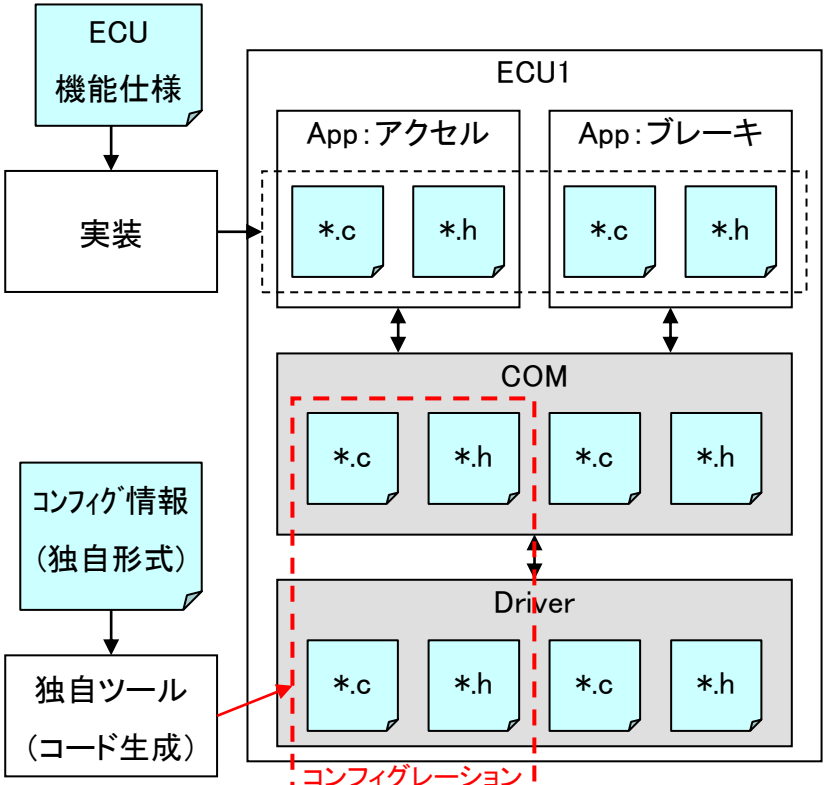


関数をコーディングする

- 上位文書に従って関数内のロジックを実装する。

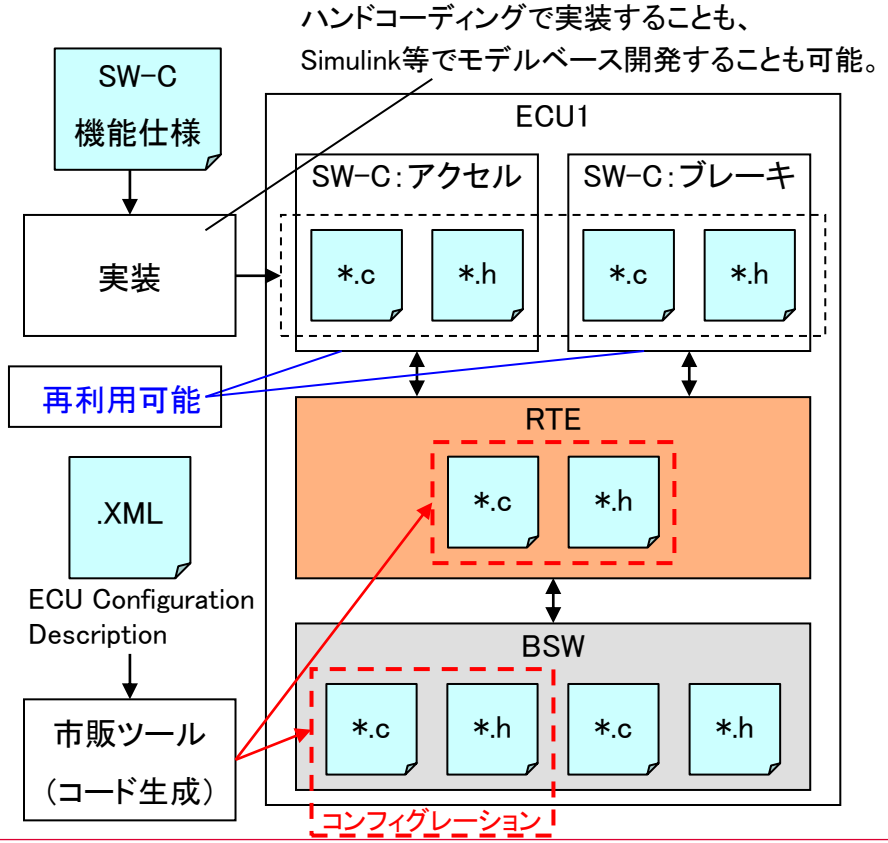
【従来開発】

COMの差異をアプリケーションで一部吸収して実装。




【AUTOSARを利用した開発】

COMの差異はRTEで吸収し、SW-CはCOMに左右されずに実装。



# まとめ



開発の流れに大きな違いはない

- 標準化された形式に従って成果物を作成することが主な変化点。

【従来開発との違い】

| 工程 | 従来開発 | AUTOSAR |
|---|---|---|
| 車両アーキテクチャ設計<br>車両システム設計 | ①ECU単位で機能仕様を作成する。<br>②ECUに配置された機能から通信仕様を作成する。 | ①SW-Cに分解して機能仕様を定義する。<br>②SW-CをECUに配置して通信仕様を作成する。 |
| ECU単体システム設計 | ①独自ツールで情報を抽出する。<br>②主に通信仕様の情報を抽出する。 | ①市販ツールで情報を抽出する。<br>②通信仕様だけでなく、機能を含めて抽出する。 |
| ECU単体システム開発 | ①独自ツールでCOM，Driverをコンフィグする。<br>②通信仕様の差異を意識して設計する。 | ①市販ツールでBSWをコンフィグする。<br>②RTEを使用することで、内外問わずSW-C間のやり取りは規定された形式で行う。 |
| 実装 | ①COMの差異をアプリーケーションで一部吸収して実装する。 | ①COMの差異はRTEで吸収し、SW-CはCOMに左右されずに実装する。 |

→ 成果物を標準化することでツールの開発工数が削減され、

インターフェースを標準化することでSW-Cの再利用可能となる。

# 3. AUTOSAR開発ツールチェーン

1. AUTOSAR方法論で規定されているツール
2. ツールチェーン
3. ツールの嬉しさ・問題点

# AUTOSAR開発ツールチェーン



本章の概要

- 前章で説明したAUTOSAR開発で使用するツールと、入出力によるツール間の連携について説明する。

ゴール

- AUTOSAR方法論で登場するツールの役割と、ツールチェーンについて説明できること。

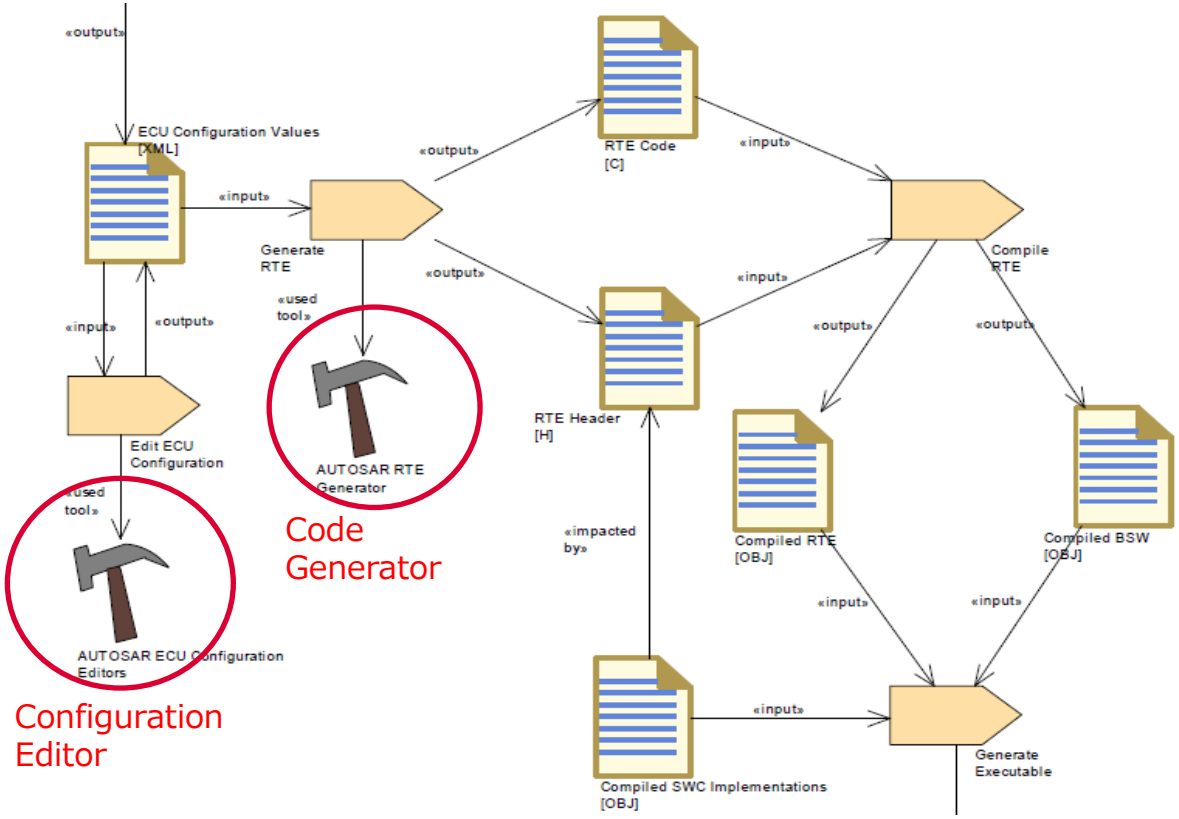
キーワード

- AUTOSARで規定されたXML形式
- 入出力による連携

# AUTOSAR方法論で規定されているツール



AUTOSAR方法論では、各アクティビティに対して使用するツールが規定されている
例：System Build Methodology（抜粋）




出典：Specification of RTE

# AUTOSAR方法論で規定されているツール

CONFIDENTIAL
関係者外秘

| 番号 | ツール名称 | 使用アクティビティ | 機能 |
|---|---|---|---|
| ① | System Configuration Generator | System Configuration | 車両アーキテクチャ設計/車両システム設計 |
| ② | ECU Configuration Extractor | After System Configuration | ECU単体システム設計 |
| ③ | SW Composition Generator | ECU Service Configuration | ECU毎に情報を抽出したシステム設計情報から、ECU内の全てのSW Compositionを保持するファイルを生成する。 |
| ④ | Service Component Configurator | ECU Service Configuration | サービスコンポーネントをコンフィグレーションする |
| ⑤ | Base ECU Config Generator | ECU Configuration | ECUのコンフィグ情報を生成する |
| ⑥ | RTE Configuration Editor | ECU Configuration | ECU単体システム開発 |
| ⑦ | COM Configuration Editor | ECU Configuration | Comをコンフィグレーションする |
| ⑧ | OS Configuration Editor | ECU Configuration | OSをコンフィグレーションする |
| ⑨ | BSW Module Configuration Editor | ECU Configuration | その他のBSWをコンフィグレーションする |
| ⑩ | RTE Generator | Basic Software Generation | RTEソースコードを生成する |
| ⑪ | COM Generator | Basic Software Generation | Comソースコードを生成する |
| ⑫ | OS Generator | Basic Software Generation | OSソースコードを生成する |
| ⑬ | Other BSW Generator | Basic Software Generation | その他のBSWソースコードを生成する |
| ⑭ | Flattener | Calibration –A2L Generation | SW-Cのもつ計測、適合情報の中間ファイルを生成する。 |
| ⑮ | A2L Generator | Calibration –A2L Generation | ECUの計測変数、適合変数を保持するA2Lファイルを生成する |

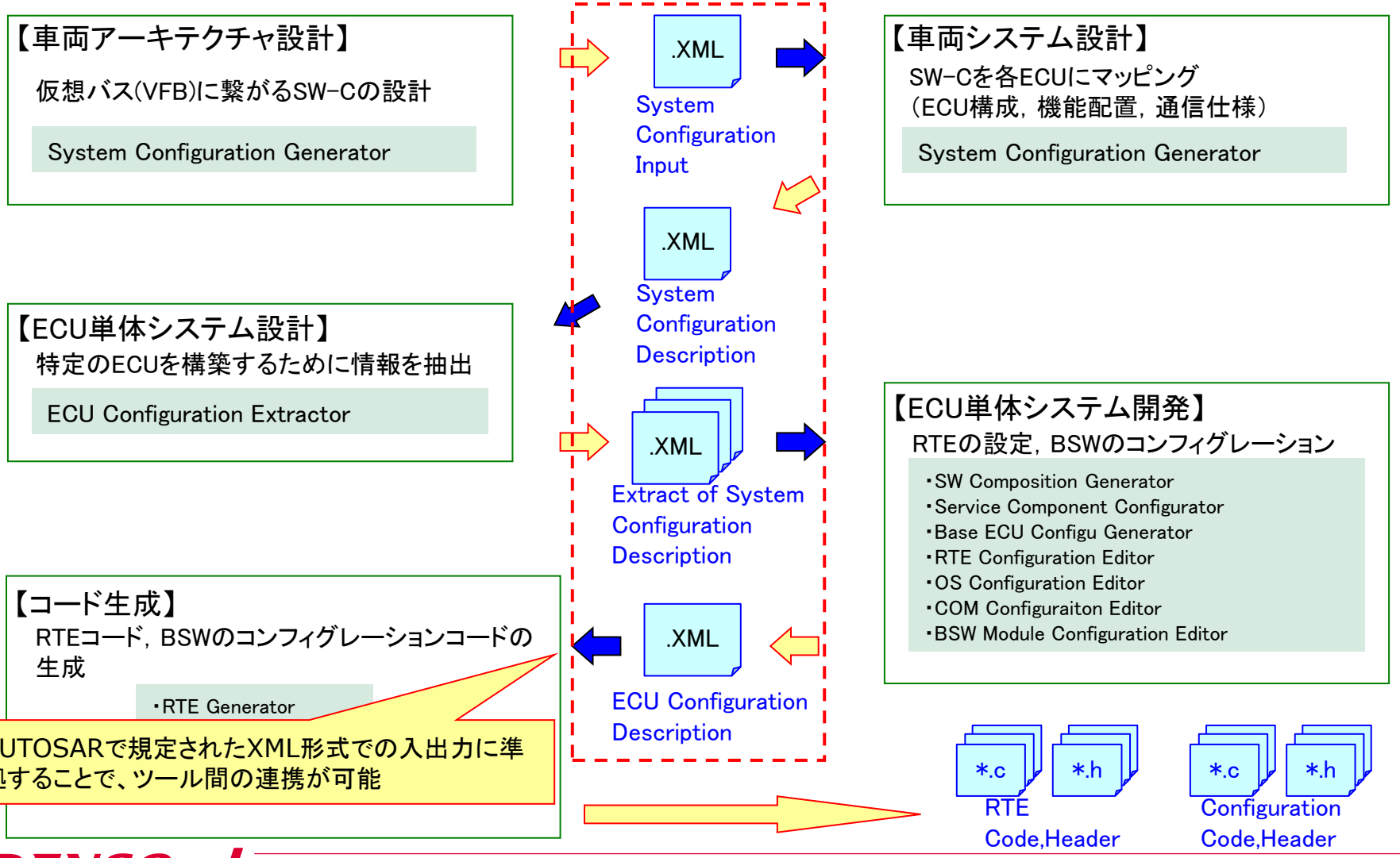
コード生成

アクティビティやモジュール毎に多くのツールが定義されている

# ツールチェーン



## 各設計工程で使用されるツールと、入出力の繋がり



# AUTOSARツールチェーンの嬉しさ・課題



AUTOSARツールチェーンの嬉しさ

- ツールの接続によるやりたいことに適合したツール環境の構築
  - 使いたいツールを選択して組み合わせることで、コストや機能範囲など、やりたいことに適したツールの組み合わせを選択できる

- 自動化による手間削減、品質向上
  - 上流工程の設計情報をそのまま入力として取り込める
  - 項目の自動補間、入力支援（設定項目の選択肢化、選択肢の絞込み等）
  - 入力したデータの整合性、値域チェック
  - 出力ファイルの再現性確保
  - 成果品質の均一化

# AUTOSARツールチェーンの嬉しさ・課題



AUTOSARツールチェーンの嬉しさ

- RTEの自動生成によるVFBの実現

- 設計情報の管理
  - 全ての設計情報がXML形式にまとまることで、関連するデータを一元管理することができる

- シミュレーションツールとの連携による早期テスト実施

# AUTOSARツールチェーンの嬉しさ・課題



AUTOSARツールチェーンの課題

- ツールチェーンの不成立
  - ツールの対象範囲やAUTOSAR解釈の違いなどにより、ツール間の連携がうまくいかない
    - カバー範囲が広く、途中のXMLが出力できない
    - 空タグの有無等、データ構造に関連する細かい点でベンダー独自の仕様が入り込んでしまう余地がある
  - ツールベンダーは他社のツールについて情報がないため、連携可否の情報は使用している側（車両メーカー）しかわからず、改善しにくい

# AUTOSARツールチェーンの嬉しさ・課題



AUTOSARツールチェーンの課題

- 変更の影響範囲をコントロールしにくい
  - 自動化により隠蔽されており、思いもよらないところに影響する可能性あり
- 作業にツールが必須となる
  - ツールがないと設計内容が確認できない
  - ライセンスがあるため複数人での同時作業がしにくい
- 設計内容が確認しにくい
  - プロパティや別ダイアログなど、設定内容が一覧視できず確認しにくい

# 4. AUTOSAR固有用語説明

1. Software ComponentとVFB
    1. Software Component説明
    2. Port説明
2. コネクタ説明

本章の概要

- AUTOSARを適用したECU開発で利用する固有用語について説明する

ゴール

- ECU開発で利用する固有用語について理解し、説明できること。

キーワード

- SW-C
- VFB
- Port/Port Interface
- コネクタ

# Software ComponentとVFB

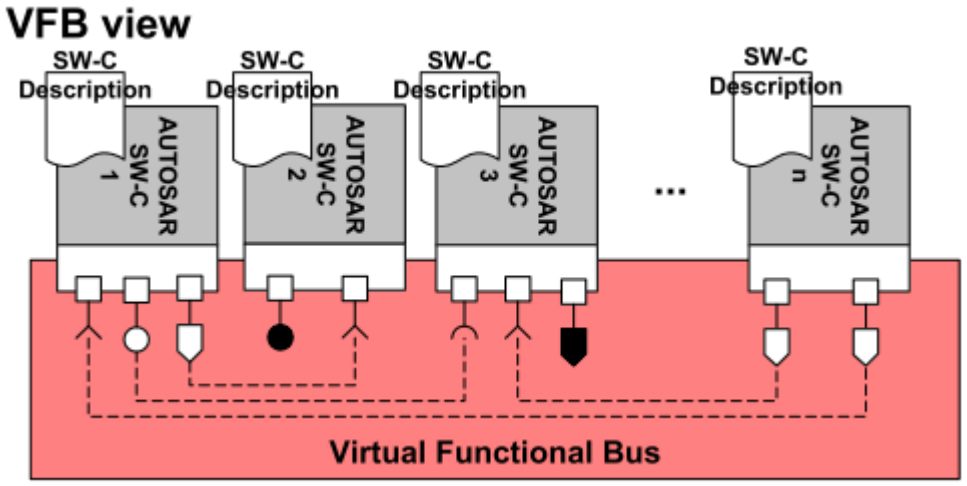


Software Component (SW-C)

- 車両の機能を構成する要素
- ハードに依存しないソフトウェアモジュール
  - 機能の容易な再配置、再利用が可能

Virtual Functional Bus (VFB)

- ハード構成を意識せずに機能設計するための仮想バス
  - SW-C間の通信プロトコル、ハードウェア、タイミング設計等は考慮しない




出典：AUTOSAR「AUTOSAR_SWS_VFB」

# Software ComponentとVFB



| Software Component種別 | 内容 |
|---|---|
| Application software component | アプリケーションを実装するためのSW-C |
| Sensor-actuator software component | センサーとアクチュエータを制御するSW-C |
| Parameter software component | Calibration parameterを他のSW-Cに提供する |
| Composition software component | 複数のSW-Cをカプセル化したより抽象度高いSW-C |
| Service Proxy software component | 外部ECUのBSW Serviceを提供する機能をプロキシとして実現するSW-C |
| Service software component | BSW Servicesの提供する機能をコンポーネント化したSW-C |
| ECU abstraction software component | BSW ECU Abstractionの提供する機能をコンポーネント化したSW-C |
| NvBlock software component | 不揮発性のデータを定義し、SW-C間でそのデータを共有するSW-C |
| Complex device driver component | BSW標準でないデバイスドライバの提供する機能をコンポーネント化したSW-C |

出典：AUTOSAR「AUTOSAR_SWS_VFB」

# Software Component説明



Internal Behavior

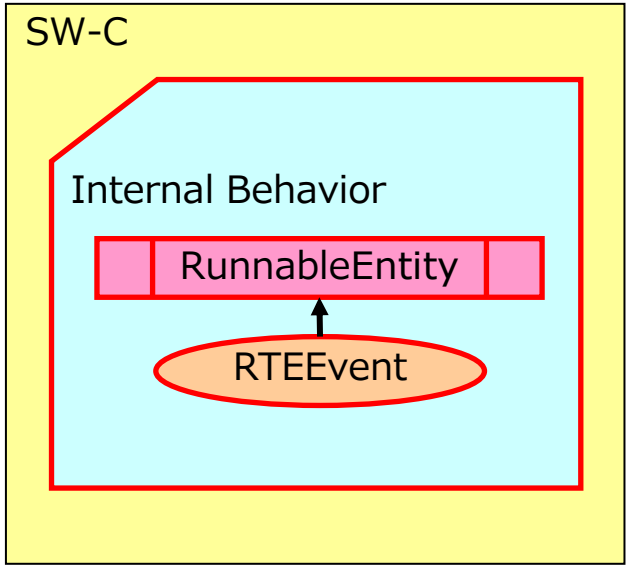
- SW-Cの内部動作
  - RunnableEntity
  - RTEEvent

RunnableEntity

- RTEから駆動される実行単位
  - 処理の名称
  - 参照、更新するデータ
  - 内部の振る舞い

RTEEvent

- RunnableEntityを起動するための要因



# Port説明

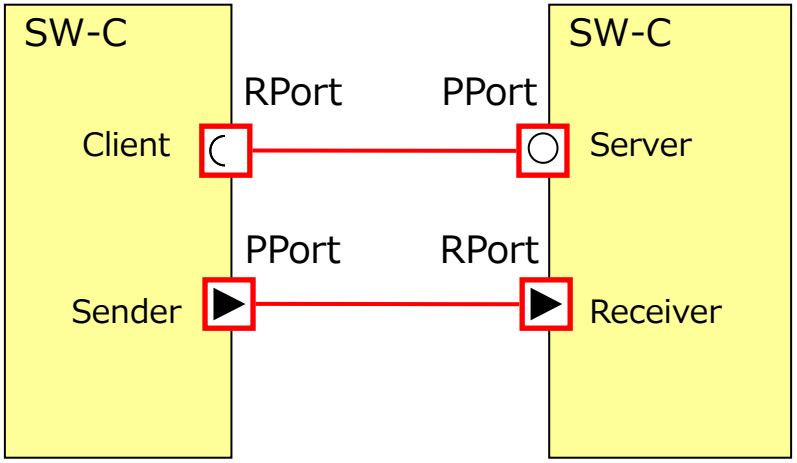


Port

- SW-C間でやり取りするための口

Port Interface

- Port間で流れるデータ、制御の呼び出し関係を定義する
- ECU内/ECU外通信、使用するプロトコルはここでは考慮しない
  - SW-CをECUに割り付ける際に定義する
- 提供側のPortをPPort、要求側のPortをRPortという
  - PPort
    - Server
    - Sender
  - RPort
    - Client
    - Receiver



# Port説明



| ポートインタフェース種別 | 内容 |
|---|---|
| SenderReceiverInterface | SenderからReceiverに流れるデータを定義する |
| NvDataInterface | NvBlock SW-Cとそれ以外のSW-C間で流れる不揮発性データを定義する |
| ParameterInterface | Parameter SW-Cとそれ以外のSW-C間で流れるパラメータデータを定義する |
| ClientServerInterface | ClientからServerに要求する処理を定義する |
| ModeSwitchInterface | ModeDeclarationGroup（システムが扱う状態データ）を定義する |
| TriggerInterface | RTEEventを起動するためのトリガソースを定義する |

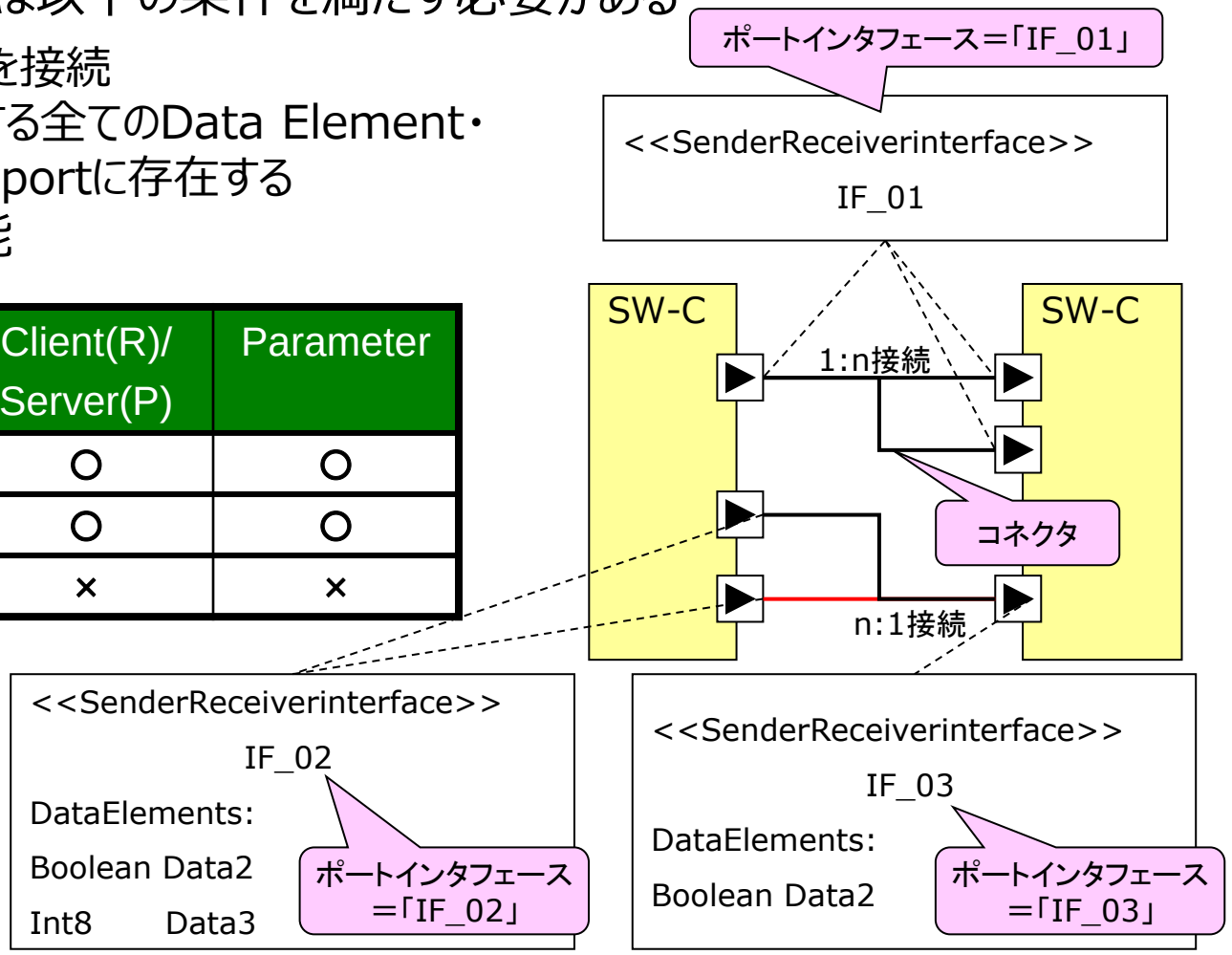
出典：AUTOSAR「AUTOSAR_SWS_VFB」

# コネクタ説明



- SW-Cのポート同士をつなぐ存在
- 接続可能なポートは以下の条件を満たす必要がある
  - PPortとRPortを接続
  - Rportに存在する全てのData Element・OperationがPportに存在する
- 以下の接続が可能

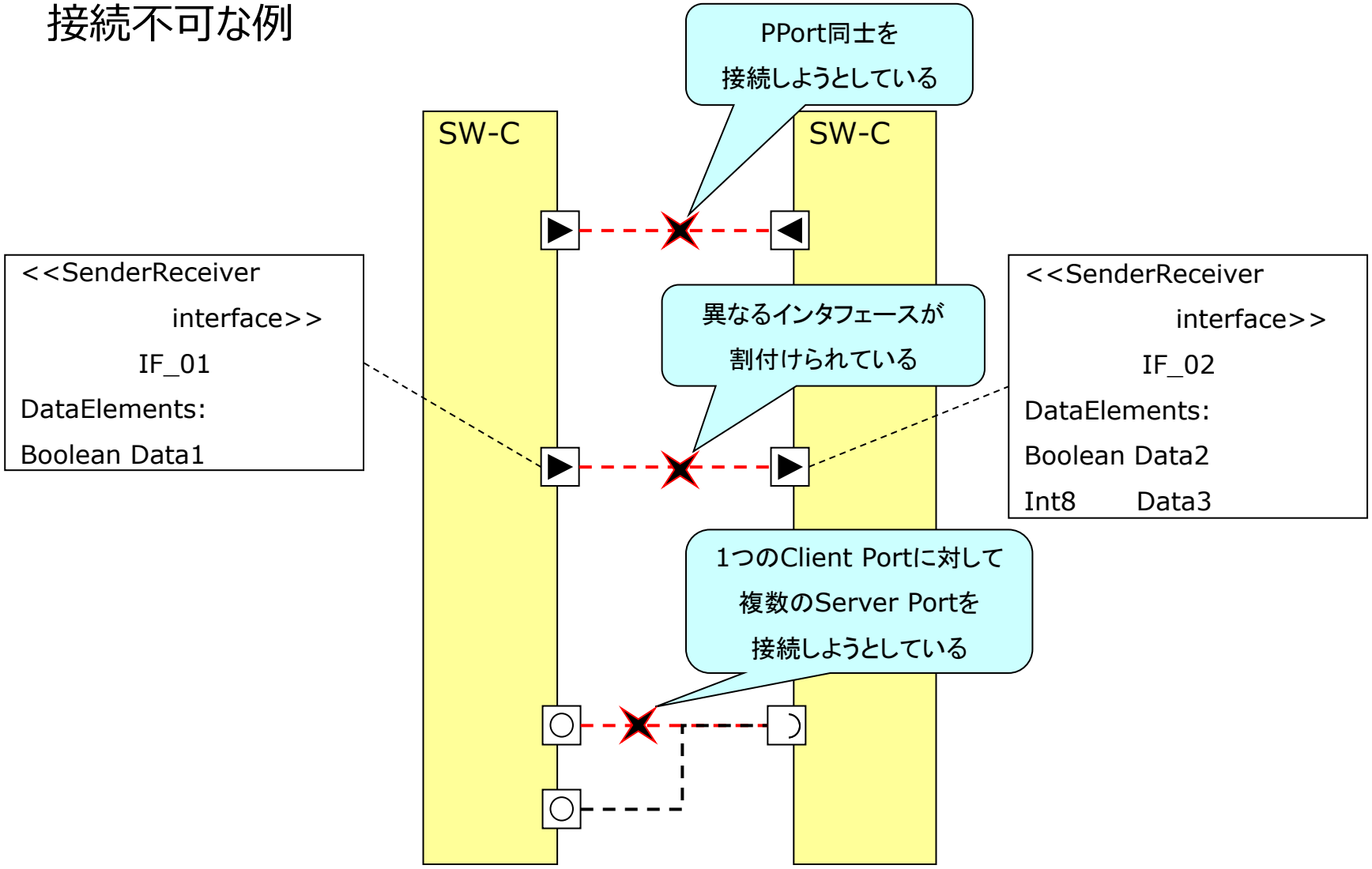
| 接続種別<br>P : R | Sender(P)/<br>Receiver(R) | Client(R)/<br>Server(P) | Parameter |
|---|---|---|---|
| 1 : 1 | ○ | ○ | ○ |
| 1 : n | ○ | ○ | ○ |
| n : 1 | ○ | × | × |



# コネクタ説明



接続不可な例



# 5. BSW説明

1. BSWとは？
2. BSWの構成
3. BSW Stack概要

# BSW説明



本章の概要

BSWの概要と構成について説明する。

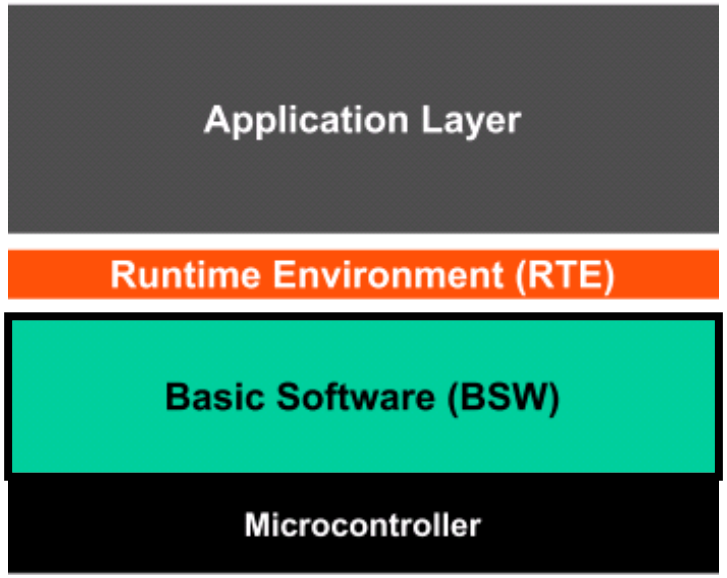
ゴール

- AUTOSARの各Layerの概要を説明できること。

# BSWとは？ ～概要～



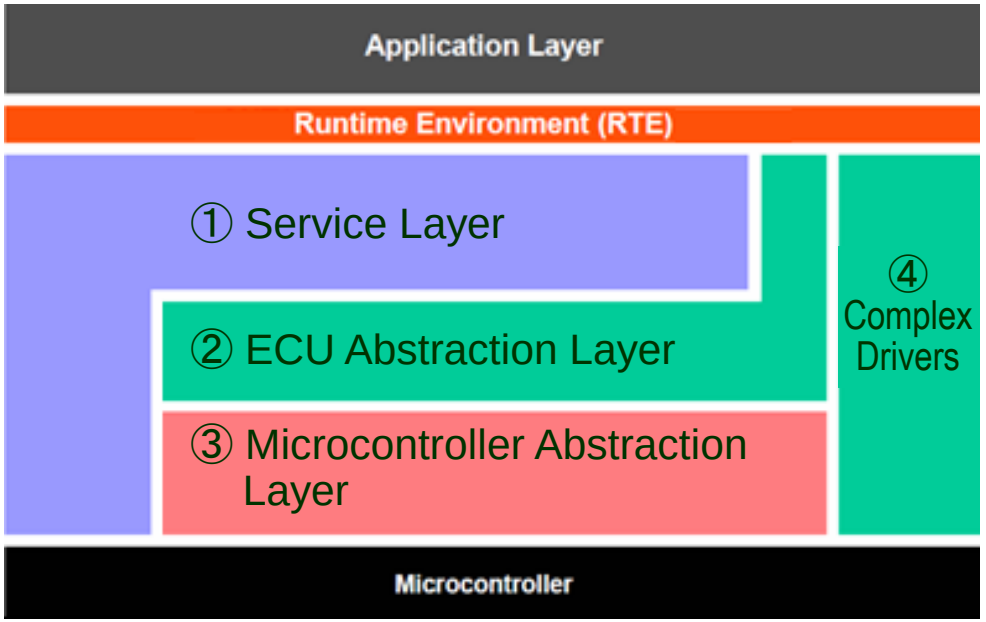
BSW＝**B**asic **S**oft**w**areの略
RTEとMicrocontrollerを繋ぐ層
役割
実装におけるハードウェアの
　差異を吸収する
アプリが必要とする共通機能を
　提供する




出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSWの構成






出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

① Service Layer
- アプリケーションが共通に使用する機能を提供

② ECU Abstraction Layer
- MCALが提供するデータ値をアプリケーションが使用できるように抽象化

③ Microcontroller Abstraction Layer　※MCALとも表記する
- デバイスドライバを標準化し、ハードウェアに依存するデータ値を抽象化

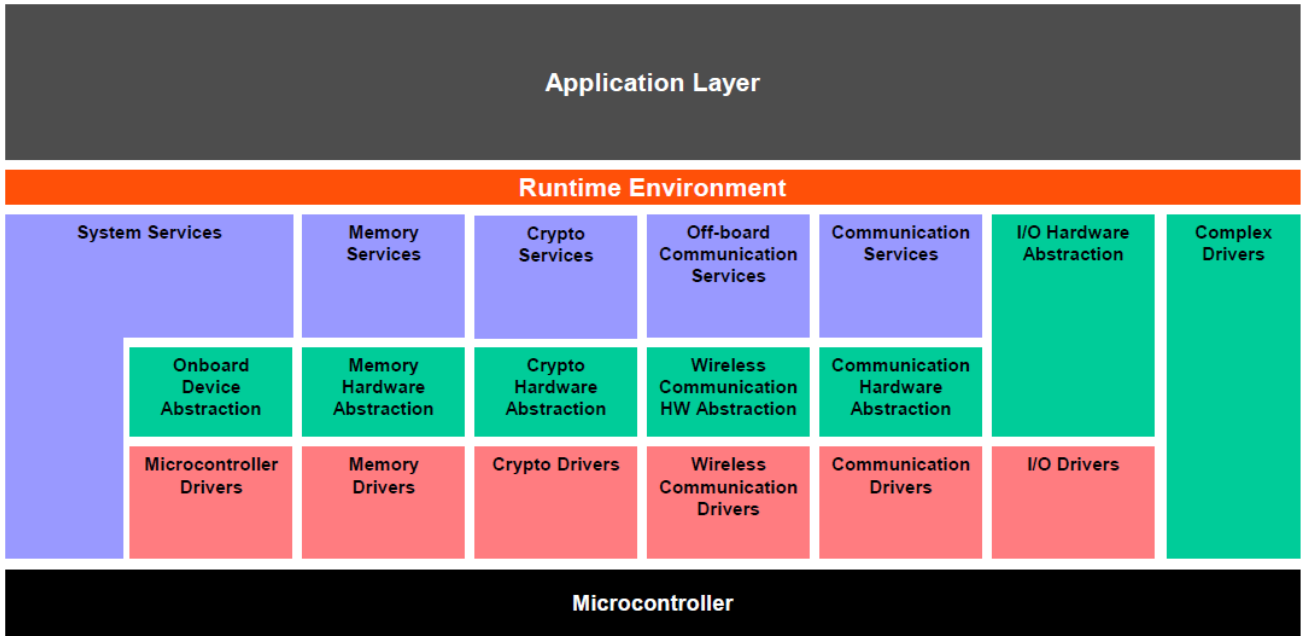
④ Complex Drivers
- AUTOSARで定義されていない機能を提供
  実用例：サーボモータードライバ、LCDドライバ、AUTOSAR非対応の既存資産を本モジュールとして扱う、等

# BSWの構成



## Service Layer

- System Services
- Memory Services
- Cypto Services
- Off-board Communication Services
- Communication Services



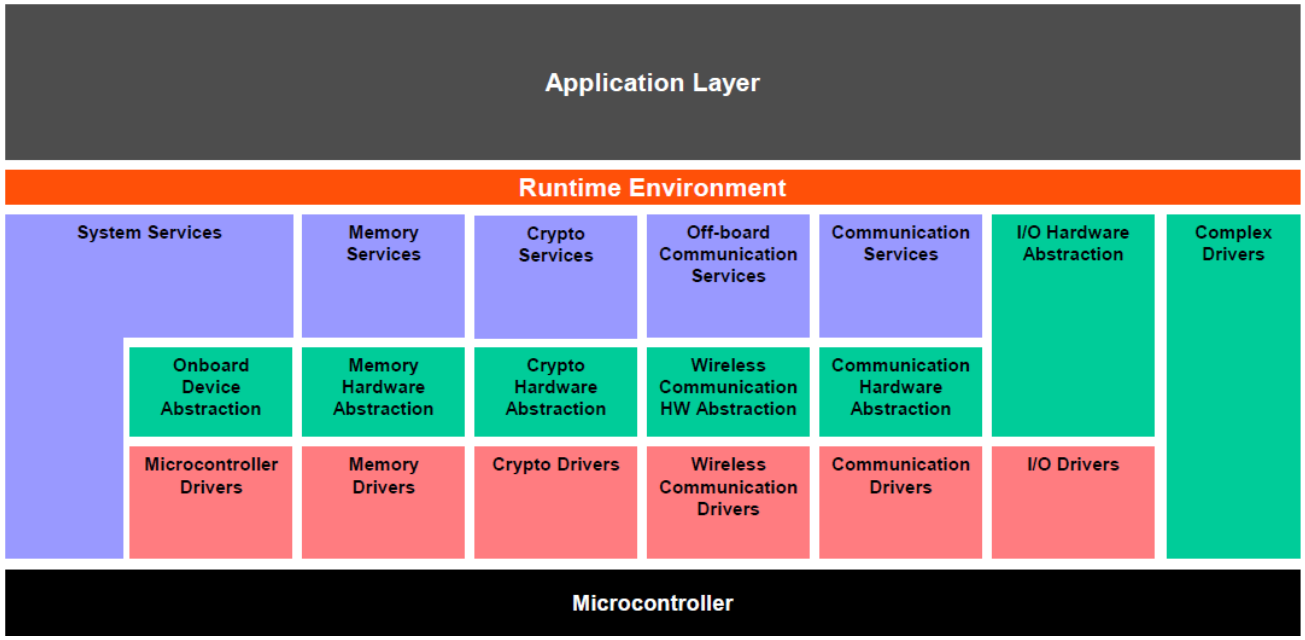

出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSWの構成



## ECU Abstraction Layer

- Onboard Device Abstraction
- Memory Hardware Abstraction
- Crypto Hardware Abstraction
- Wireless Communication HW Abstraction
- Communication Hardware Abstraction
- I/O Hardware Abstraction



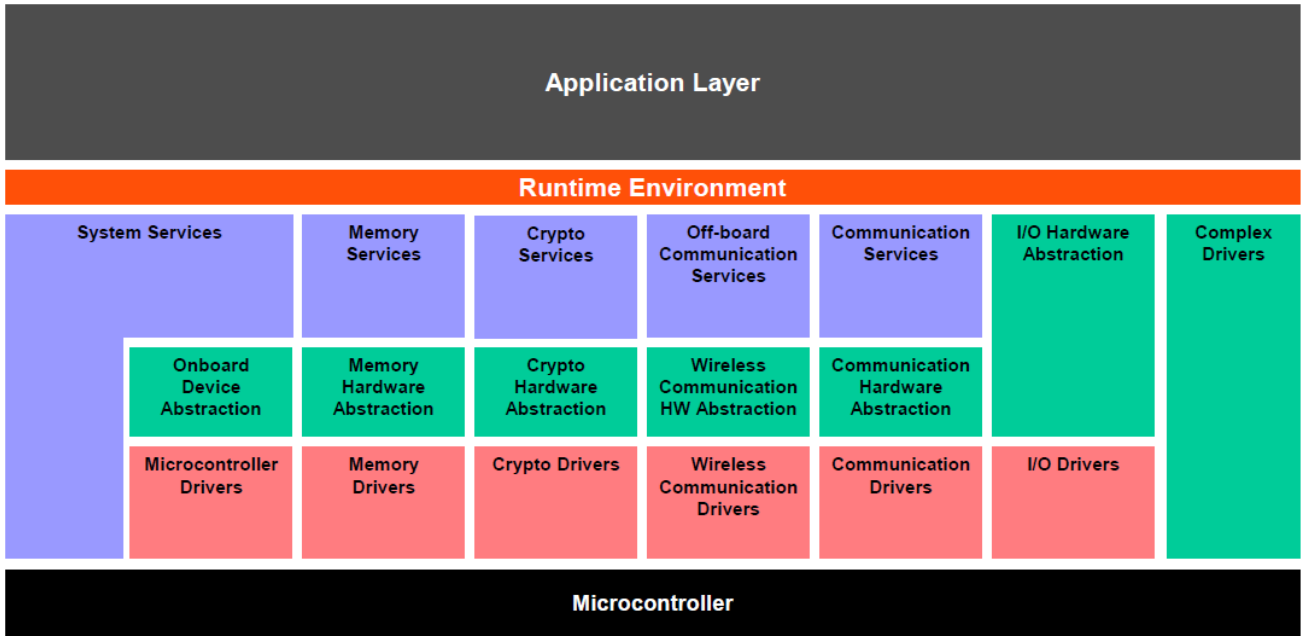

出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSWの構成



## Microcontroller Abstraction Layer

- Microcontroller Drivers
- Memory Drivers
- Communication Drivers
- I/O Drivers
- Crypto Drivers
- Wireless Communication Drivers



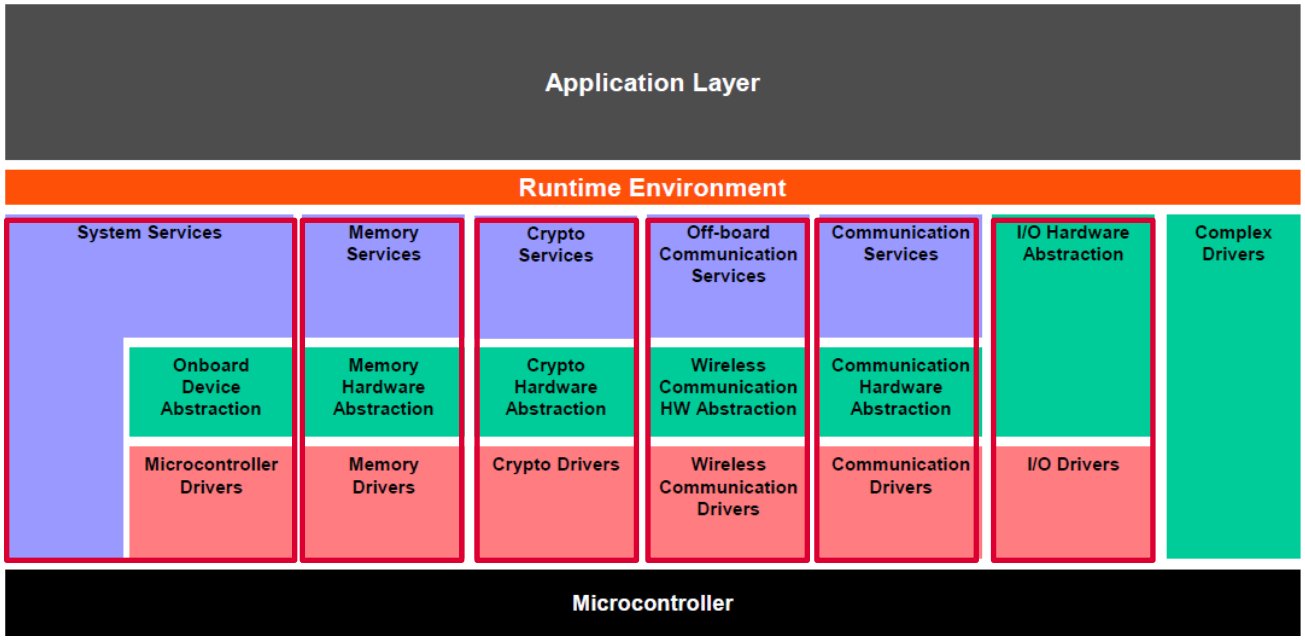

出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSW Stack概要



各LayerはStackと呼ばれるサービスの種別毎に更に分割される

- System
- Memory
- Crypto
- Off-board Communication(AR4.3で追加)
- Communication
- Input/Output(I/O)




出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

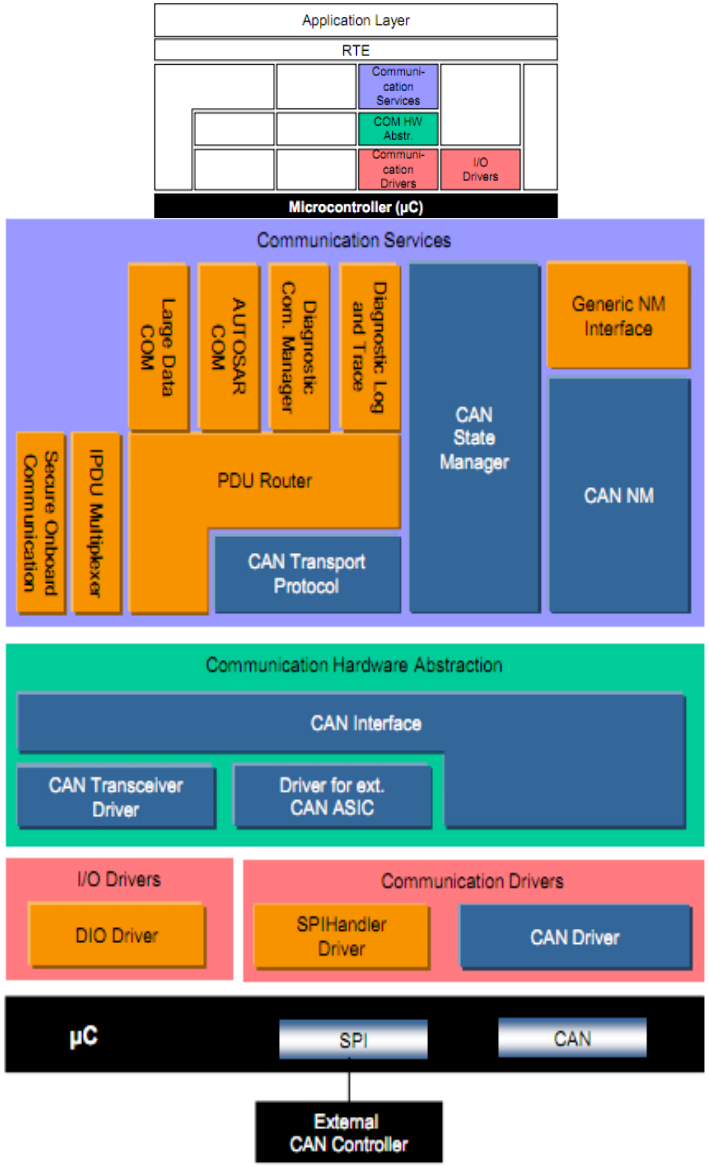
# BSW Stack概要



Communication

- 車載ネットワークシステム、ECUオンボード通信システム、ECU内SW間通信へのアクセスを標準化する
- 対応している通信プロトコル
  - CAN(CANFD)
  - TTCAN
  - J1939
  - LIN(Maste/Slave)
  - TCP/IP(Ethernet)

    ※右図はCANのモジュール構成
- 異なる通信プロトコル間のゲートウェイ




出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSW Stack概要

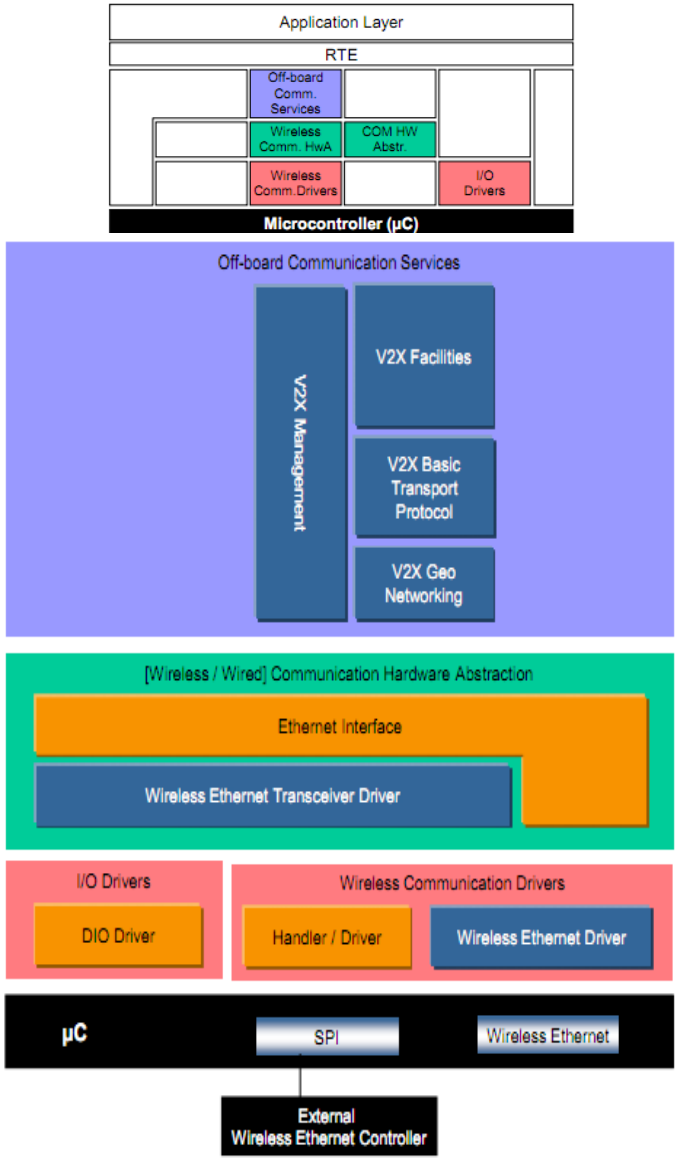


Off-board Communication Services

- 車輌ワイヤレスネットワーク
  システムを用いた通信への
  アクセスを標準化する
- 外部ワイヤレスEthernet
  コントローラの制御にも対応

このStackはV2xと呼ばれることが多い

※Vehicle-2-X



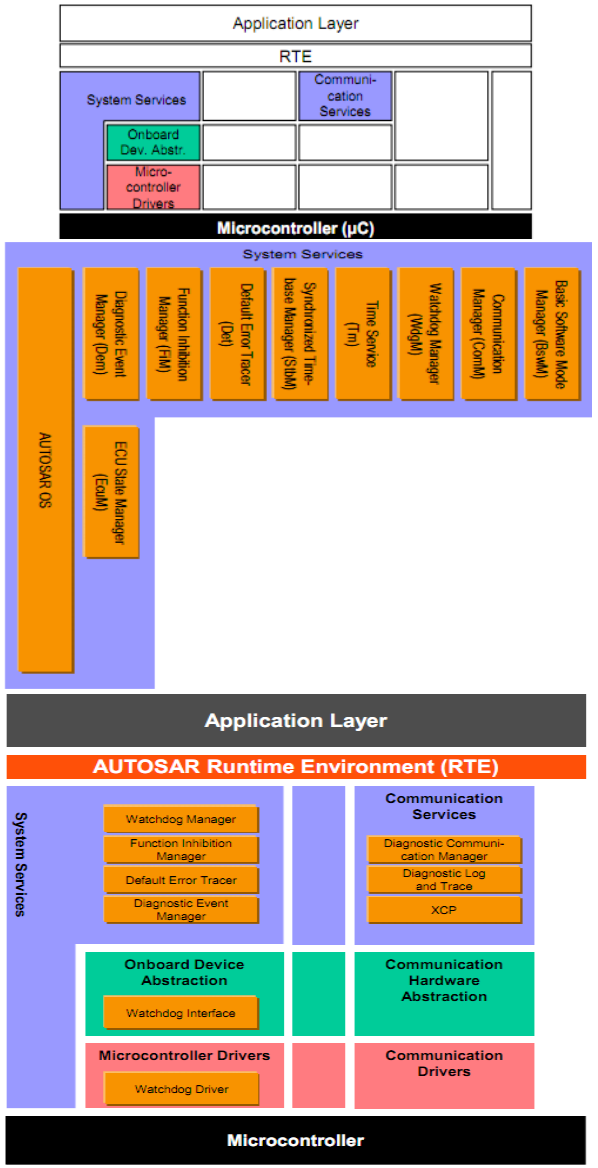

出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSW Stack概要



System

- 標準化可能な機能
  OS,タイマ,エラーメモリ管理
- ECU特有のサービス
  ECU状態管理,ウォッチドッグ管理
- 各種ライブラリ
  固定/浮動小数点演算,E2E
  CRC,Crypto, Bit制御,
  その他拡張機能（ex：64bit演算 等）

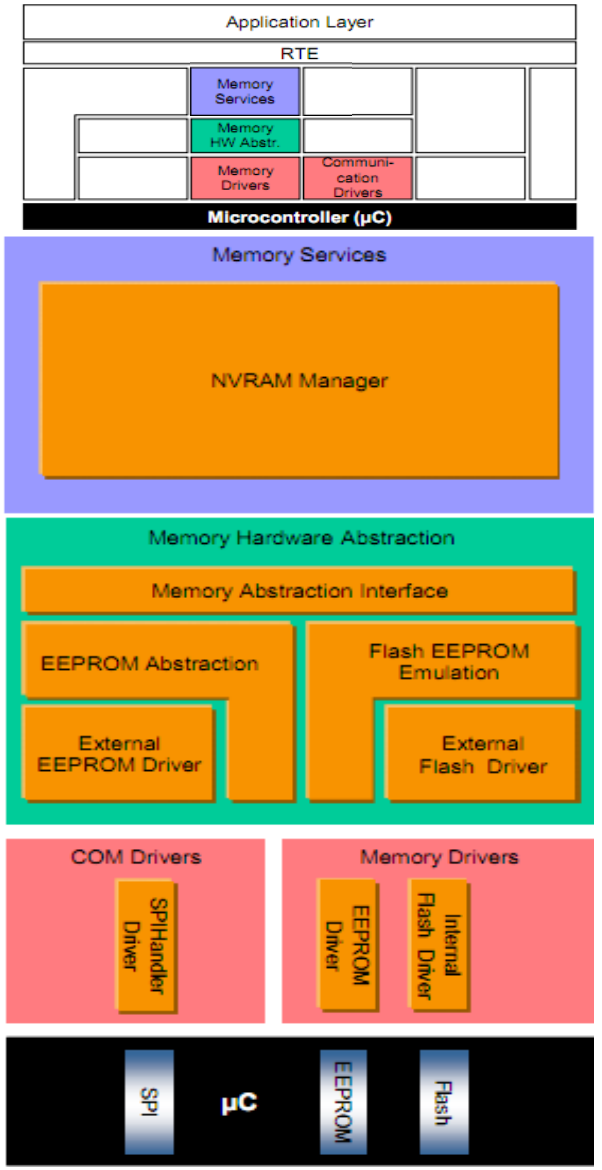



出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSW Stack概要

Memory

- 内部/外部不揮発性メモリへの
  アクセスを標準化する
- 対象デバイスはEEPROM / FlashROM
- データ冗長化や書き込み回数を考慮
  した制御も可能
  ※実装はBSW Vendorに依存する



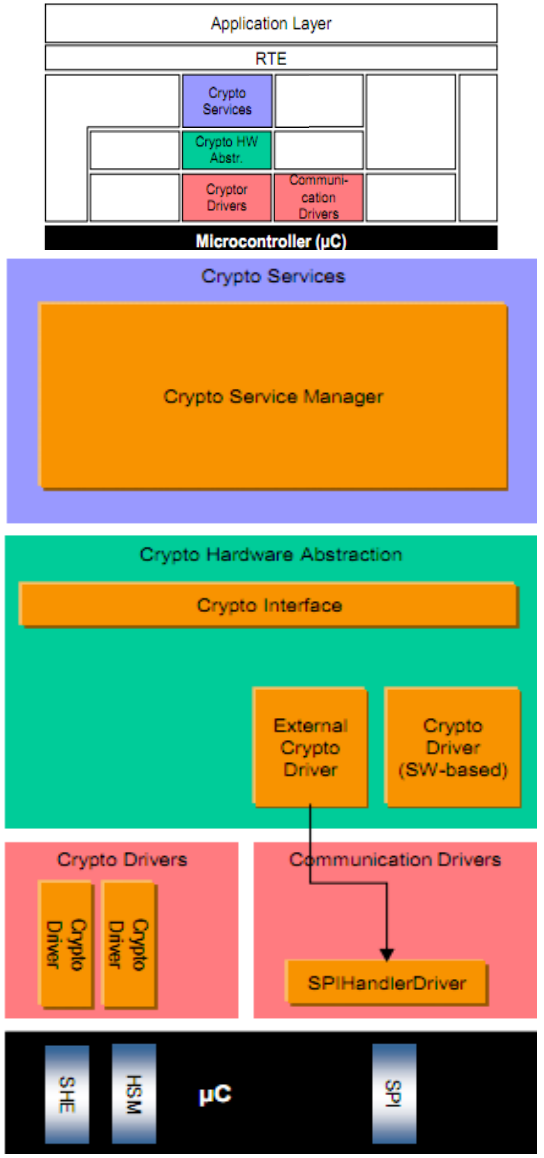

出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# BSW Stack概要



Crypto

- 暗号化機能へのアクセスを標準化する
- ハードウェアによる暗号化、ソフトウェアによる暗号化双方をサポートする
- 外部Cryptコントローラの制御にも対応

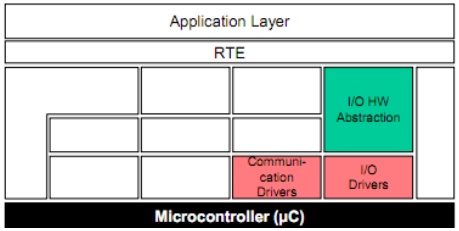


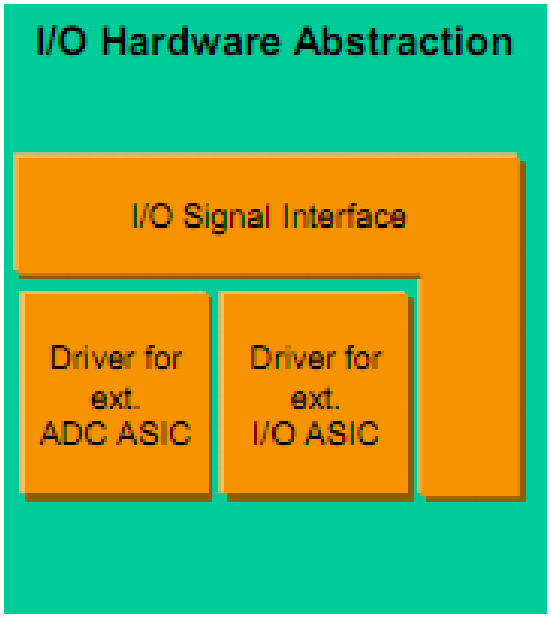

出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

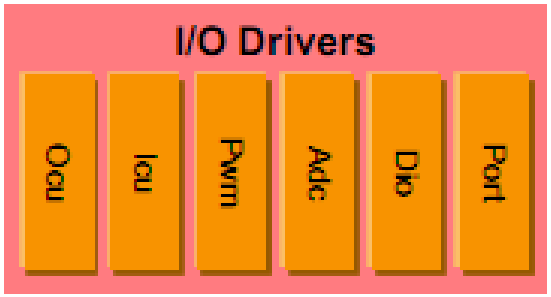
# BSW Stack概要

Input/Output(I/O)

- ECUオンボードの周辺機器へのアクセスを標準化する
- Port：物理ポートの入出力、機能等を設定
- Dio：汎用入出力ポート制御
- Adc：A/D変換制御
- Pwm：Pulse width modulator制御
- Icu：Input Capture制御
- Ocu：OutputCompare制御




出典：AUTOSARAUTOSAR_EXP_LayeredSoftwareArchitecture」

# 実際の開発で感じたこと



## 私たちが開発している工程

【車両アーキテクチャ設計】

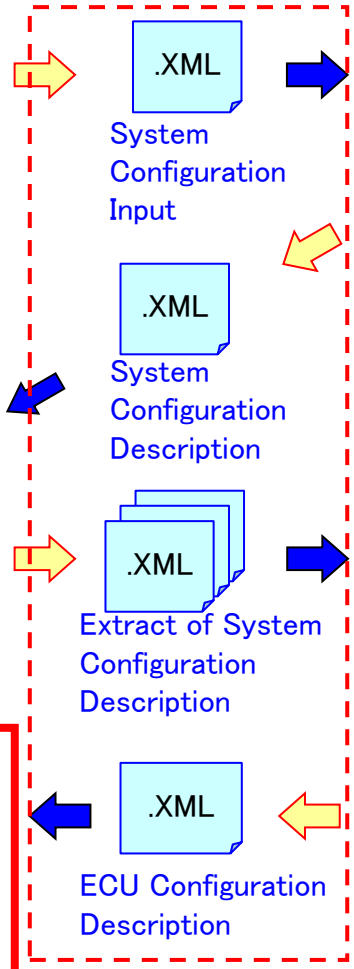
仮想バス(VFB)に繋がるSW-Cの設計

System Configuration Generator

【車両システム設計】

SW-Cを各ECUにマッピング
（ECU構成，機能配置，通信仕様）

System Configuration Generator




【ECU単体システム設計】

特定のECUを構築するために情報を抽出

ECU Configuration Extractor

【ECU単体システム開発】

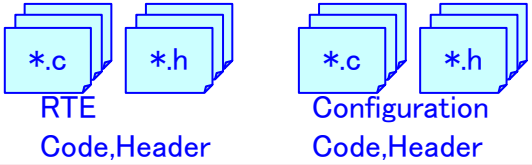
RTEの設定，BSWのコンフィグレーション

- SW Composition Generator
- Service Component Configurator
- Base ECU Configu Generator
- RTE Configuration Editor
- OS Configuration Editor
- COM Configuraiton Editor
- BSW Module Configuration Editor

【コード生成】

RTEコード，BSWのコンフィグレーションコードの生成

- RTE Generator
- COM Generator
- OS Generator
- Other BSW Gnerator



# 実際の開発で感じたこと



うれしいところ

- コンフィグレーションするだけでコードが自動的に生成される
- Communication StackやDiagnosticのように標準規格に従う機能は、製品毎に違いが少ないため、BSWを再利用でしやすい
- OEMとの仕様調整がAUTOSARをベースとするため容易になる

課題

- ベンダー毎の仕様解釈の違い、独自仕様に起因したインテグ問題
  - フォーマットが違うことで異なるベンダのツールが使用できない
  - ツールベンダー独自のI/F、型を持つ場合があり、異なるベンダのソフト同士を結合できない
  - ベンダ毎に仕様準拠状況、制約が異なるため、ベンダの変更に対する影響が大きい　等々
- 購入した3rd Party製ソフトの品質保証
- AUTOSAR仕様ではOEM要件（機能/非機能関わらず）が少なからずある

# まとめ



AUTOSARに関する導入知識獲得のために、以下を説明した。

- AUTOSARの概要(**3つの標準化**) 、導入・使用の目的やうれしさ **[1章]**
  - アプリケーションインタフェース (Application Interface)
  - AUTOSAR方法論 (AUTOSAR Methodology)
  - レイヤードアーキテクチャー (Layered Architecture)

- AUTOSARの開発の流れ、従来のECU開発との違い **[2章]**
- AUTOSARツールチェーンに関する情報 **[3章]**
  - 車両アーキテクチャ設計
  - 車両システム開発
  - ECU単体システム設計
  - ECU単体システム開発

# まとめ



AUTOSARに関する導入知識獲得のために、以下を説明した。

- AUTOSARの固有用語として、主にSW-Cに関連する用語 **[4章]**
  - SW-C
  - VFB
  - Port、コネクタ

- AUTOSAR BSWの役割、機能概要 **[5章]**
  - BSWの階層構造
  - Stack毎のモジュール構成、機能概要

DENSO

Crafting the Core