

名古屋大学
組込みシステム研究センター(NCES)の
AUTOSAR に対する取り組み

2013年8月23日(金)

名古屋大学 嶋原 一人

目次

1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

車載ソフトウェアの課題と再利用性の向上

自動車の製造コストに占める電子部品の割合

2007年：20～30%



2015年：40%

ソフトウェアの
再利用性向上が急務

どのOEMへも
提供可能としたい

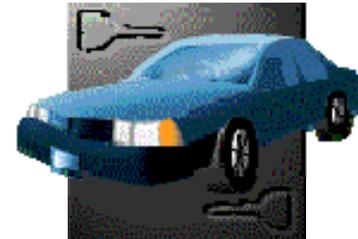
サプライヤ開発
ソフトウェア



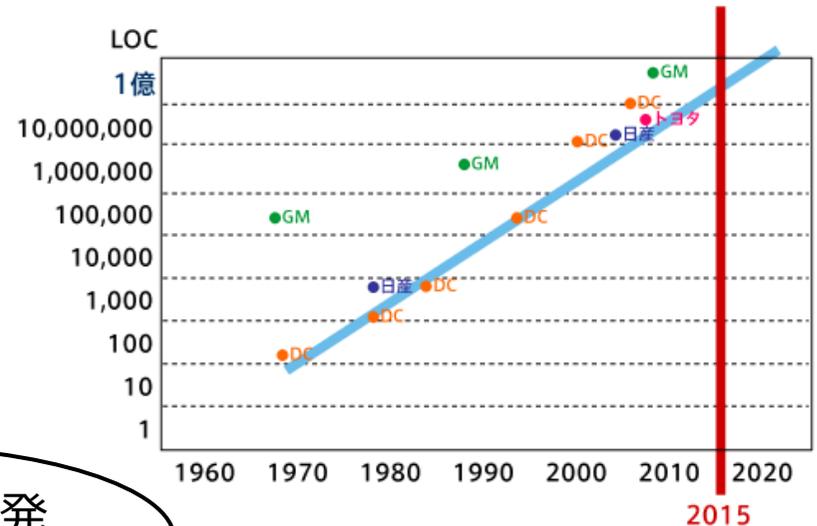
OEM A



OEM B



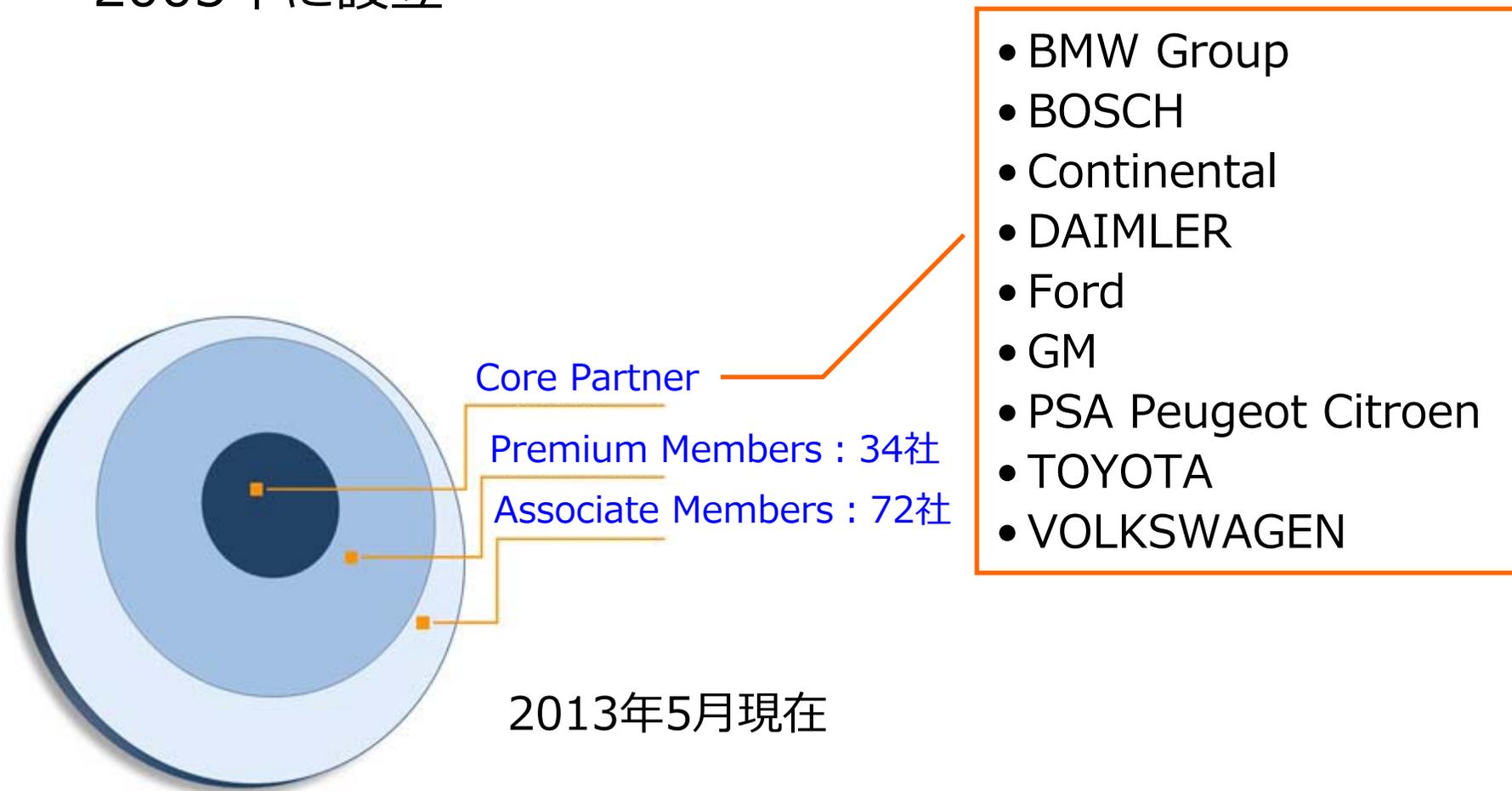
OEM C



車両の現状及び将来技術とエレクトロニクス
May 24th 2007 NISSAN MOTOR CO.,LTD Kazutaka ADACHI

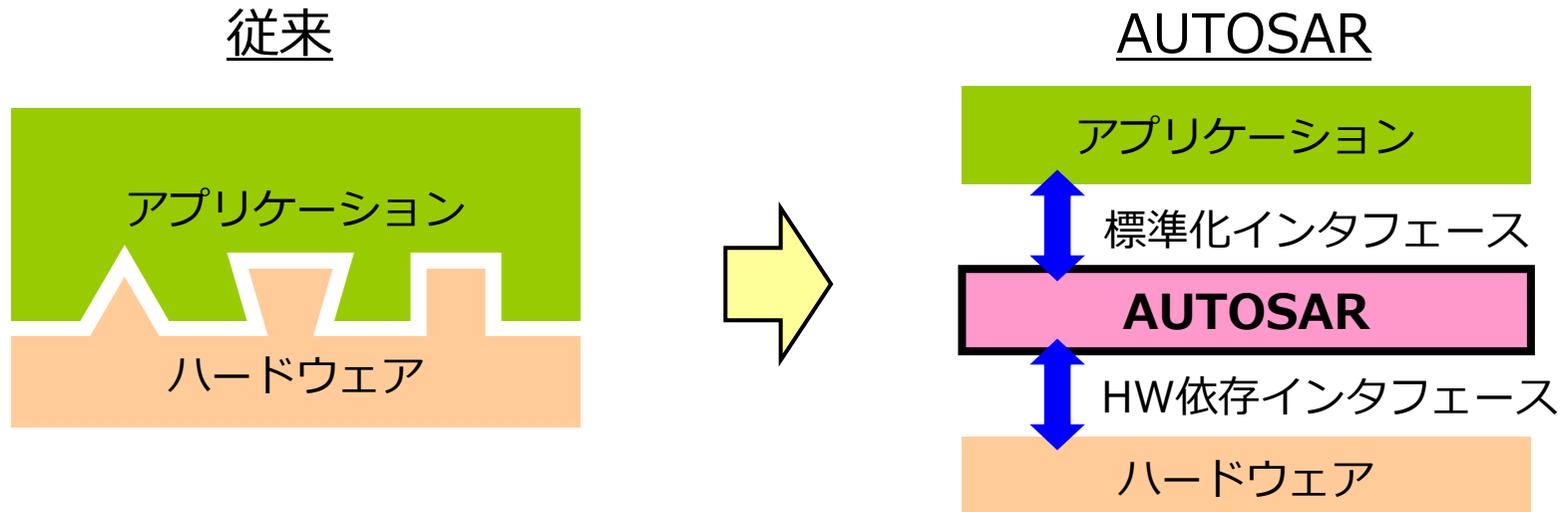
設立

- **AUT**omotive **O**pen **S**ystem **AR**chitecture の略称
- 2003年に設立



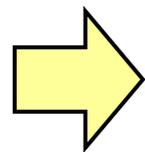
<http://www.autosar.org/>

AUTOSARのコンセプト



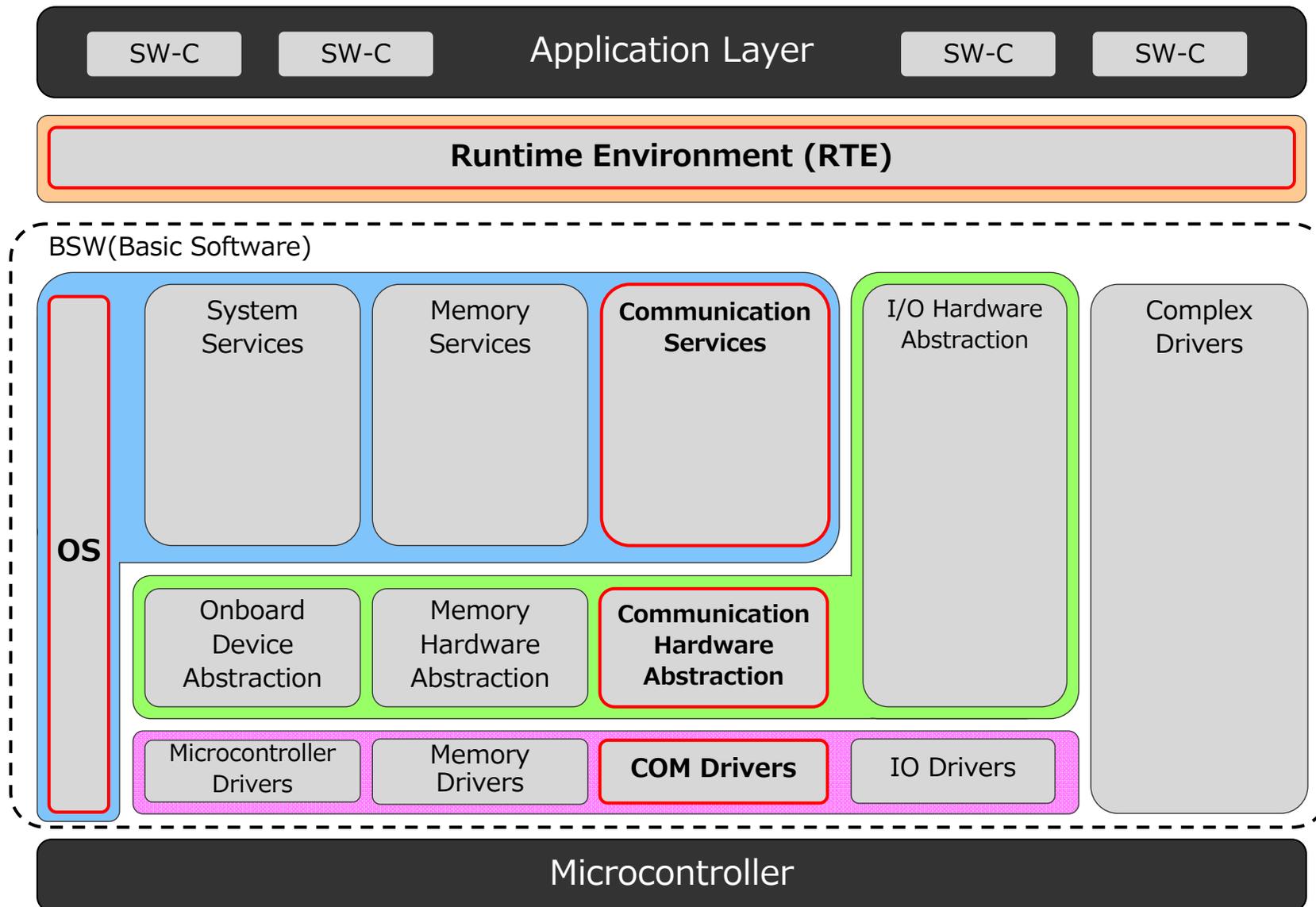
"Cooperate on standards, compete on implementation"

メーカーやハードウェアに依存する
インタフェース, コンフィギュレーションを標準化



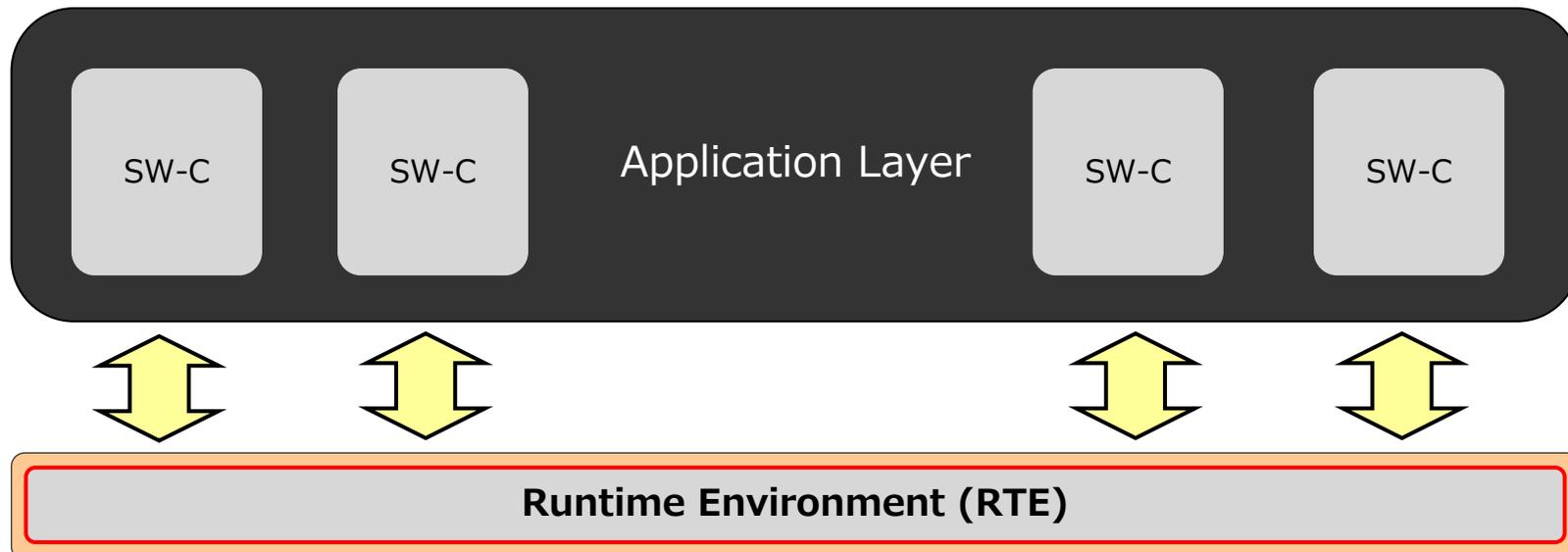
ソフトウェアの再利用性が向上し,
開発コストの低減も実現される

アーキテクチャ



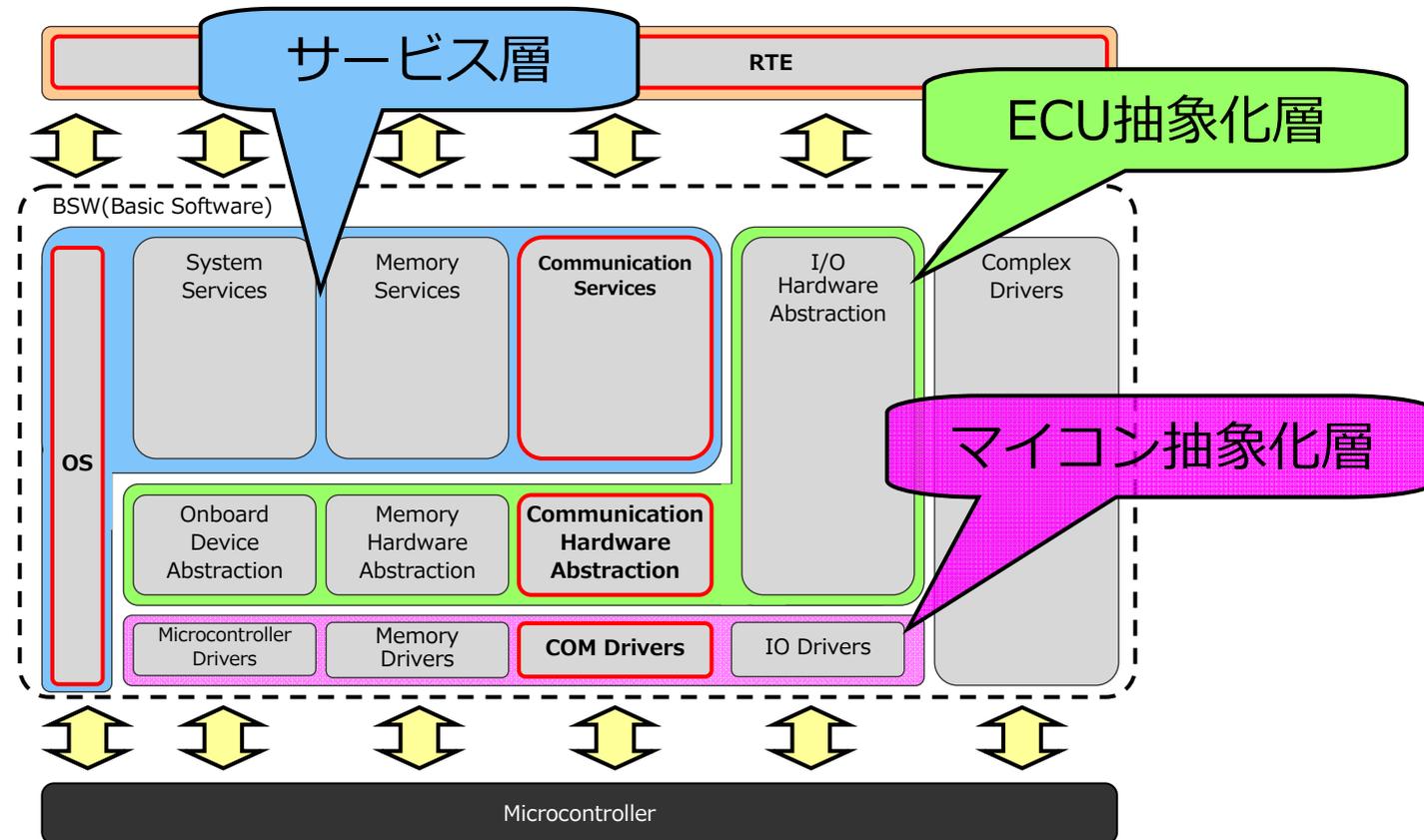
SW-CとRTE

- エンジン制御やセンサ監視など、アプリケーション毎にソフトウェアコンポーネント(SW-C)の開発を行う
- SW-CからはRTEから提供されるAPIを使用して、他のSW-Cとの通信処理や排他制御を行う
- RTEより下位層は、SW-Cからは見えないよう抽象化される
→**どのECUで実行されるかも見えない**
- RTEはSW-Cからの要求にしたがって、BSWを使用して、ECUを制御し、アプリケーションを効率的に実行する
- **SW-Cは、ECUに依存せず開発できるので再利用性が向上する**

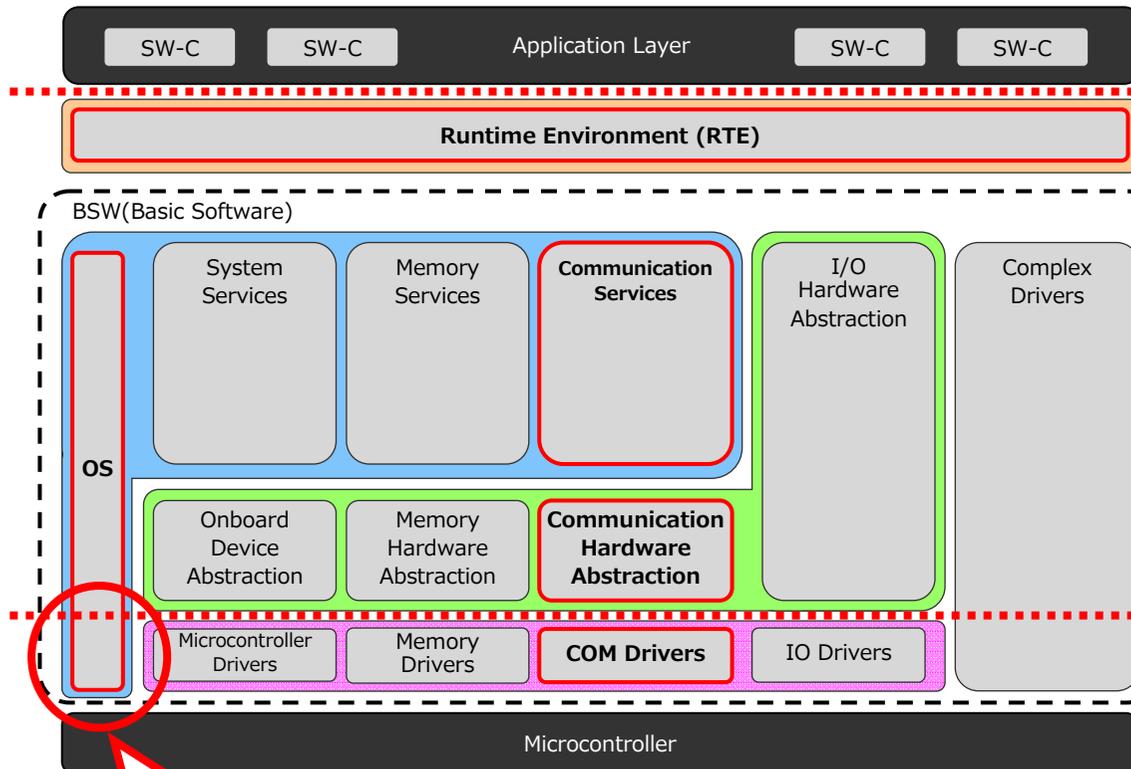


BSW

- RTEからの要求に対して、マイコンを制御するまでの処理を階層構造によって抽象化するためのコンポーネント群
- ほとんどのBSWはOS上で動作するため、SW-C, RTEも最終的にはOS上で動作する



想定される開発対象者



- 自動車メーカー
- 部品メーカー

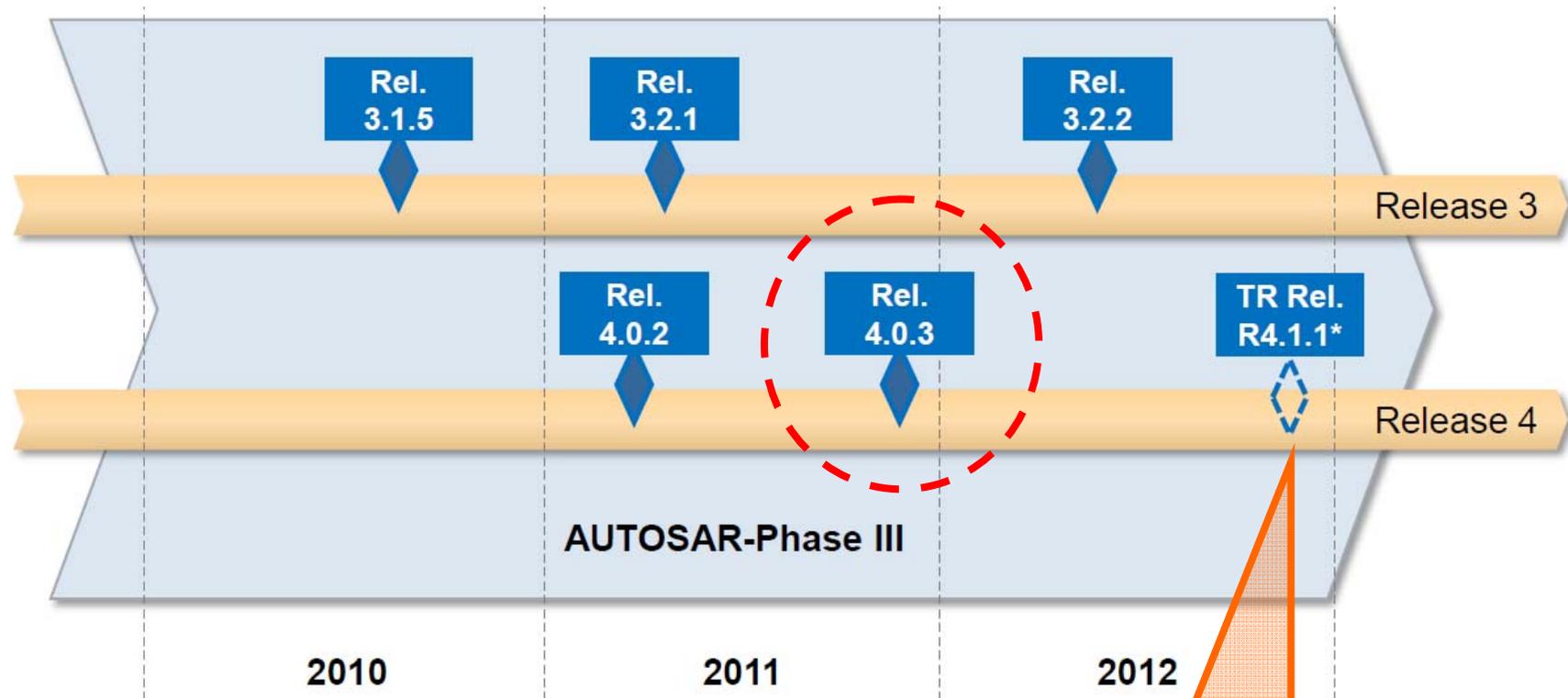
- 部品メーカー
- ツールベンダ

- 半導体メーカー

OSのターゲット依存部は・・・？
(インターフェイスが未規定)

標準化により部品化が可能となり
再利用性が向上する

ロードマップ

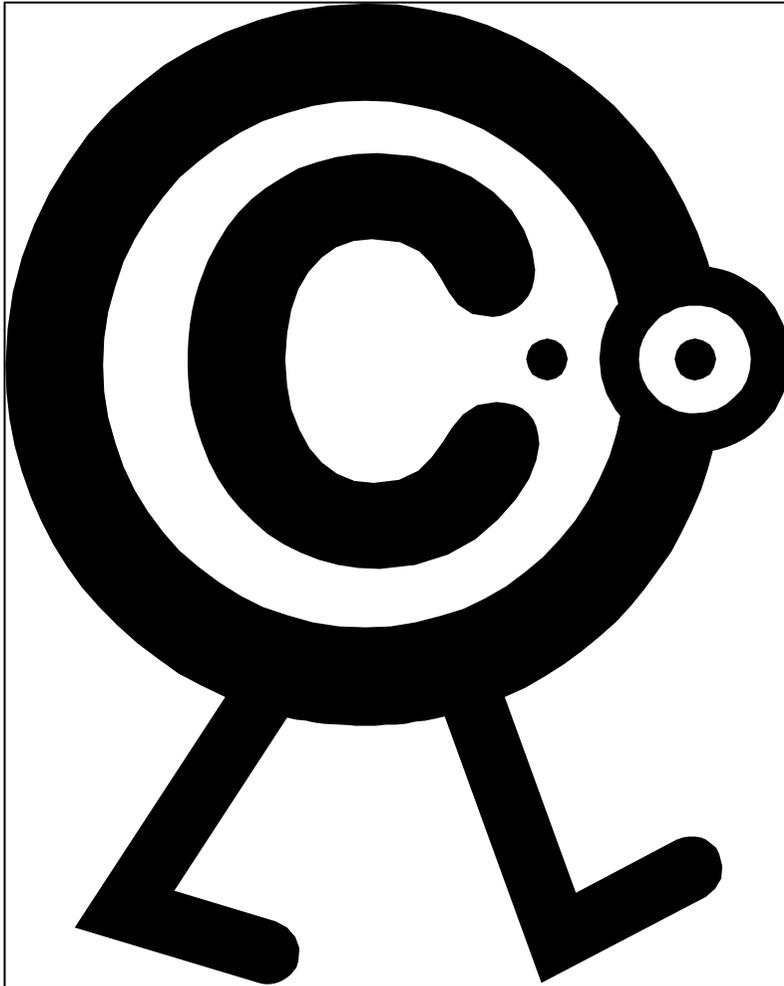


NCESではR4.0.3に
準拠して開発を実施

2013年4月9日に
リリースされた

<5th AUTOSAR Open Conference>

AUTOSAR導入のメリット・デメリット



“全領域”とは？



AUTOSARで規定
されているXML仕様は
十分でない



車載ソフトウェアでは
致命的な問題

<MONOist <http://monoist.atmarkit.co.jp/mn/articles/0910/01/news123.html>>

目次

1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

名古屋大学 組込みシステム研究センター(NCES)

設立目的

- 組込みシステム分野の技術と人材に対する産業界からの要求にこたえるために、**組込みシステム技術に関する研究・教育の拠点**を、名古屋大学に形成
- 産業界が必要とする技術課題を分析・抽出し、大学における基礎研究に反映

活動領域(スコープ)

- 組込みシステムに関する以下の活動に、**産学連携**の枠組みで取り組む
 - 大学の持つ技術シーズを実現／実用化することを指向した研究(第二種基礎研究)
 - プロトタイプとなるソフトウェアの開発
 - 組込みシステム技術者の教育／人材育成

①AUTOSAR OS仕様の実装/評価

プロジェクト概要 (2008年度～2010年度)

- Release 4.0の仕様の明確化と修正
- NCES独自のマルチコア拡張
- 必要性和オーバーヘッドのトレードオフを考慮するため、機能レベル(後述)を定義

プロトタイプ
OSを実装し、
仕様の妥当性を評価

成果物

- **外部仕様書**
 - OSEK/VDX仕様を統合した1つのドキュメント
 - 全仕様を日本語で記載
- **OSソースコード**(SC1, SC2, SC3, SC1-MC, SC3-MC)
 - 未実装の機能あり
 - AUTOSAR仕様の曖昧な点に対する十分な検討は未実施
- 単体評価スイート(機能評価, 性能評価)
 - 十分な検証は未実施

これらの成果を活かして
オープンソースで提供したい

②実用可能なAUTOSAR OSの開発

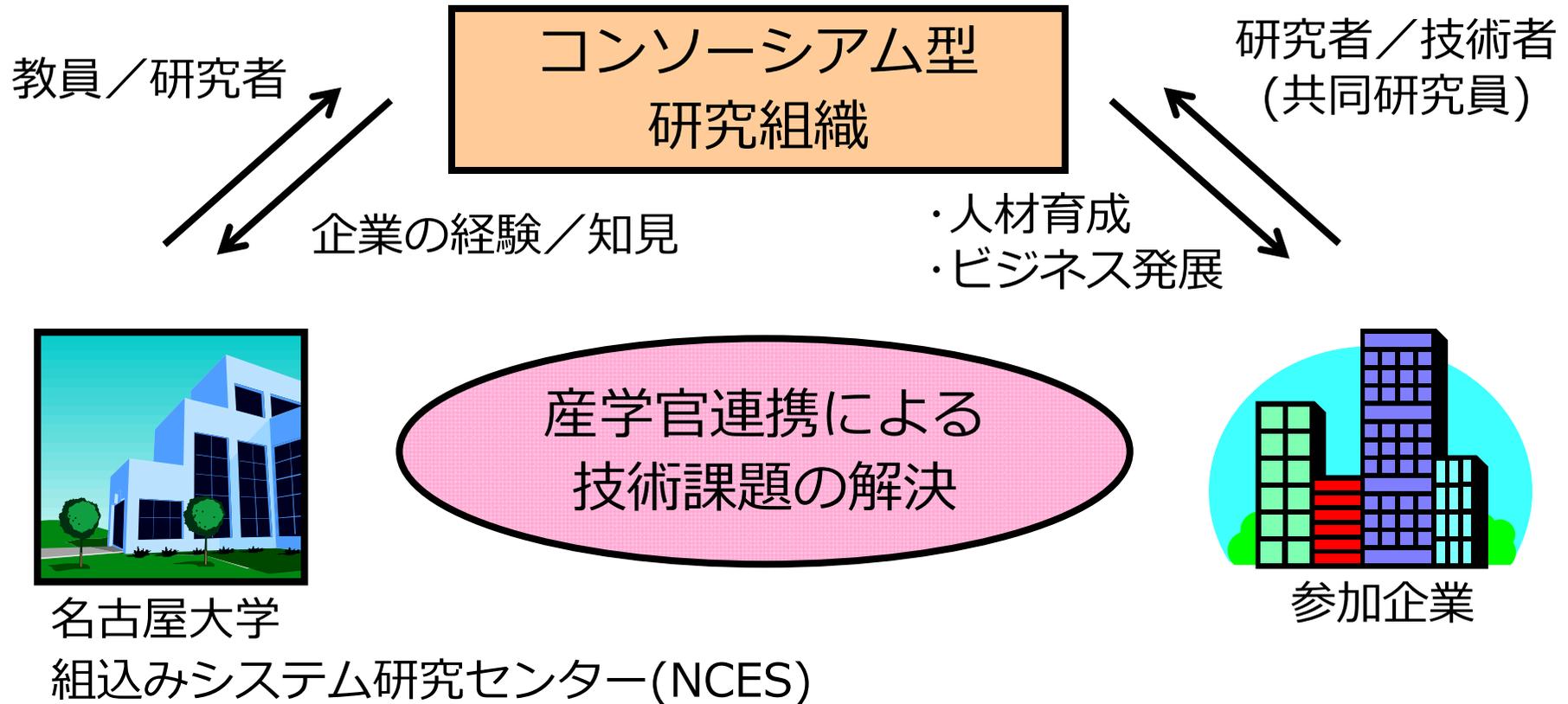
プロジェクト概要 (2011年度～現在)

- 既存の外部仕様書, OSソースコードをベースとする
- AUTOSAR普及に伴い, 国産のAUTOSAR OSの開発を目指す
- TOPPERSプロジェクトからオープンソースとしてリリース
 - **ATK2: AuTomotive Kernel version 2**
 - ATK1はOSEK/VDX仕様(オープンソース)
- 外部仕様書も, さらに検討, 精査を行い公開する
- テストスイートも合わせて開発(公開はしない)
- **段階的にCOMスタック, RTEも開発していく**
- **コンソーシアム型共同研究として実施**



コンソーシアム型研究組織とは

名古屋大学 組込みシステム研究センターが
設定した研究テーマに対して、複数の企業・団体が
参加し共同で研究・開発を行う



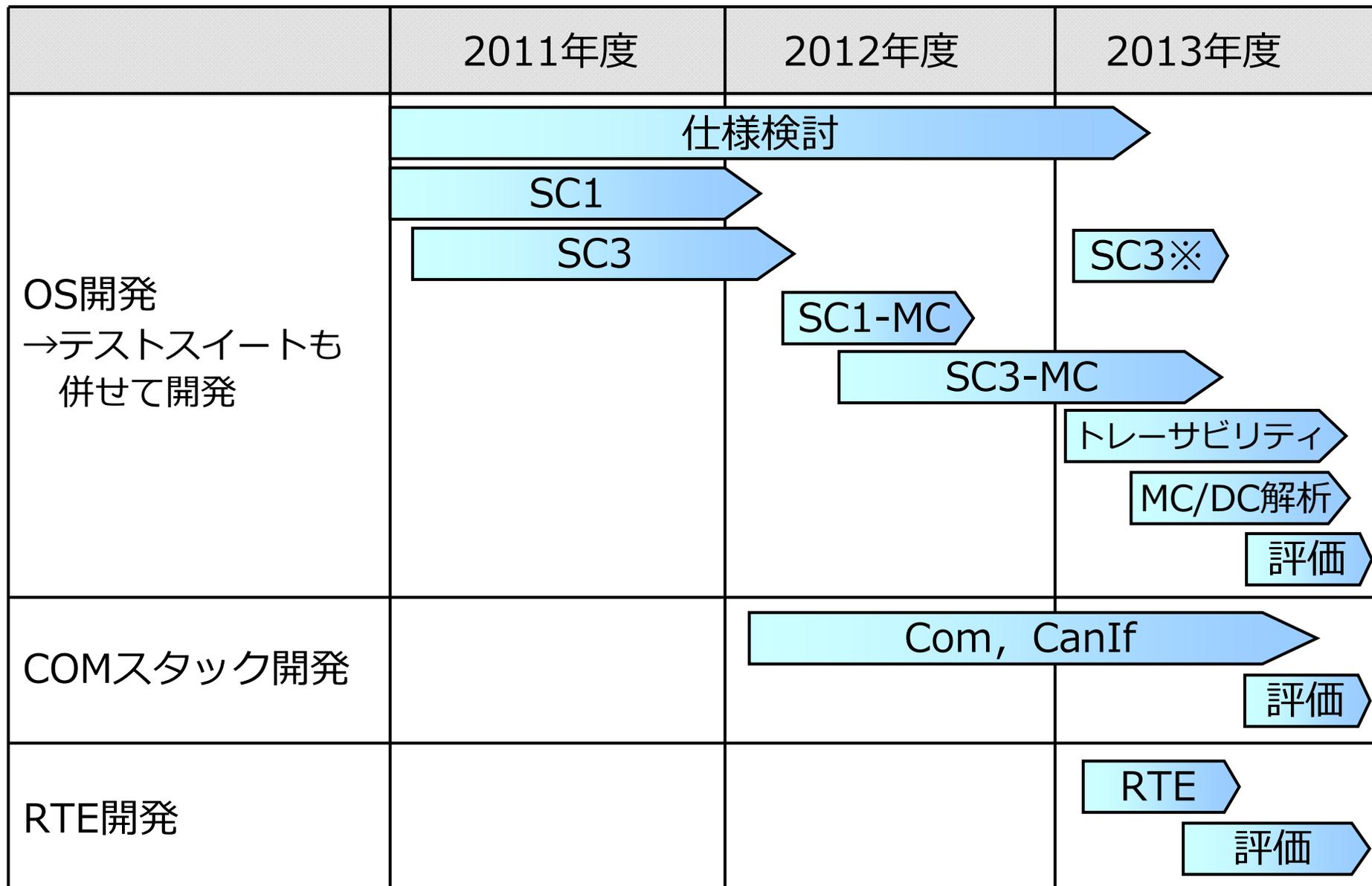
参加企業と人数(2013年5月時点)

| 企業名 | 常駐 | 研究協力 |
|----------------------|----|------|
| (株)ヴィッツ | 1 | 5 |
| (株)永和システムマネジメント | 0 | 3 |
| (株)OTSL | 0 | 2 |
| (株)サニー技研 | 1 | 5 |
| (株)デンソー | 0 | 10 |
| (株)東芝 | 1 | 3 |
| トヨタ自動車(株) | 0 | 3 |
| (株)豊田自動織機 | 0 | 6 |
| パナソニックアドバンステクノロジー(株) | 1 | 3 |
| 富士ソフト(株) | 3 | 7 |
| ルネサスエレクトロニクス(株) | 1 | 6 |
| 合計 | 8 | 53 |

上記の他、名大教員・研究員が3～4名実作業に参加

ロードマップ

※機能拡張



目次

1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

スケーラビリティクラス

- OSが提供する機能に応じてスケーラビリティクラス(SC)が定義されている
- 4つのSCが存在する
 - **SC1** → **TOPPERS/ATK2-SC1として公開**
 - 基本機能セット
 - OSEK/VDX仕様の上位互換
 - SC2
 - SC1 + タイミング保護機能
 - **SC3** → **TOPPERS/ATK2-SC3として公開**
 - SC1 + メモリ保護機能
 - SC4
 - SC1 + タイミング保護機能 + メモリ保護機能
- すべてのSCをマルチコアへ拡張可能
 - 現在, SC1-MC, SC3-MCを開発中
 - **2013年6月末に公開予定**



<http://www.toppers.jp/atk2.html>

OSEK/VDX OS仕様

- 車載システム向けOSの国際標準(ISO17356)
- OS以外にも, 通信ソフトウェア(OSEK/COM)の仕様なども策定
- コンフィギュレーションはOILという独自記法を使用する
- **TOPPERS/ATK1**はOSEK/VDX Version2.2.1に準拠している

サポートする機能

- タスク管理
- アプリケーションモード
- 割り込み
- イベント
- アラーム
- メッセージ
- フックルーチン

OILの記述例(タスク)

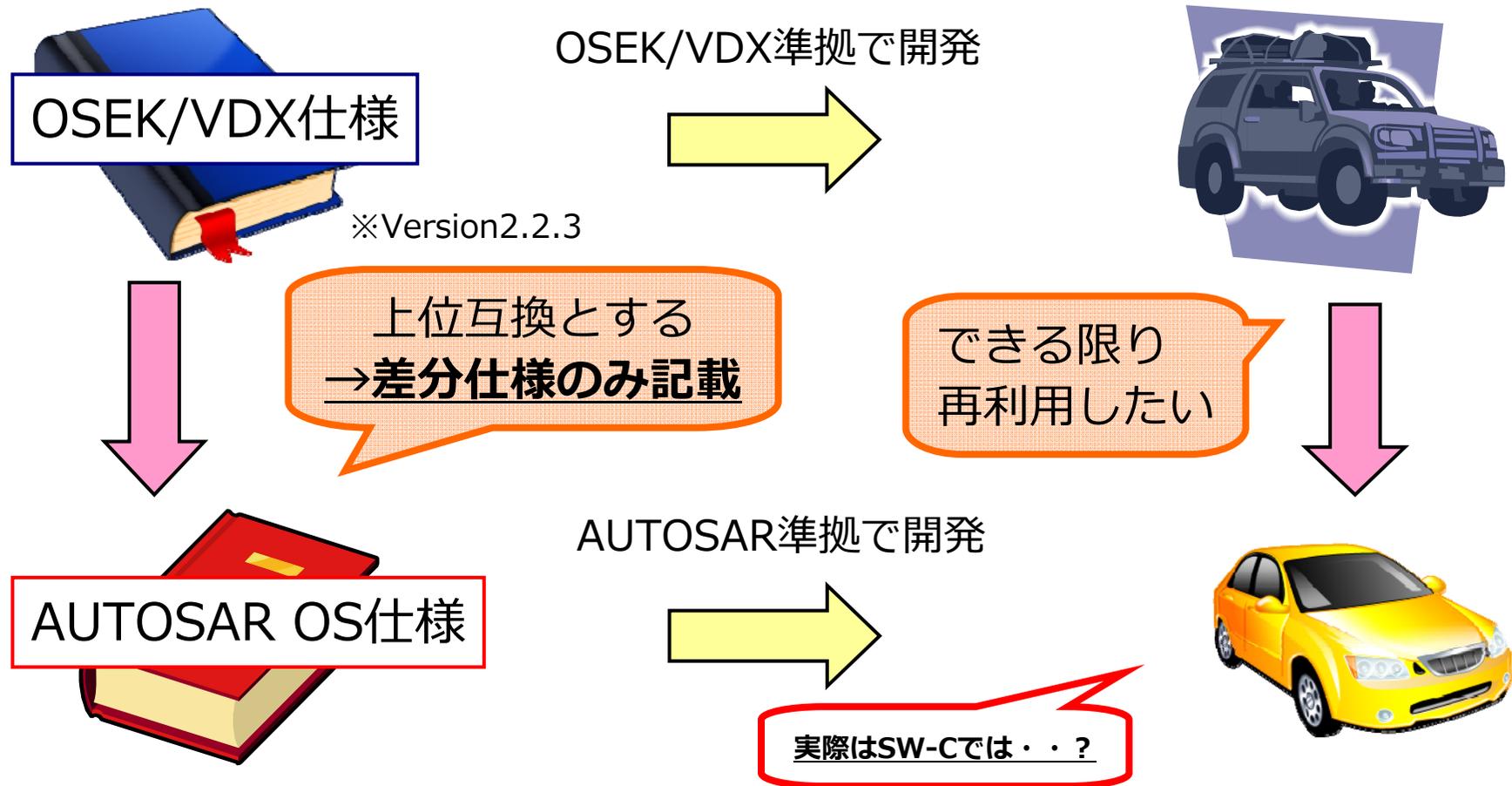
```
TASK {  
    PRIORITY = 1;  
    SCHEDULE = FULL;  
    ACTIVATION = 4;  
    AUTOSTART = TRUE {  
        APPMODE = App1;  
        APPMODE = App3;  
    };  
    RESOURCE = Res1;  
    EVENT = Evt1;  
}
```

AUTOSARとOSEK/VDX

7.1 Core OS

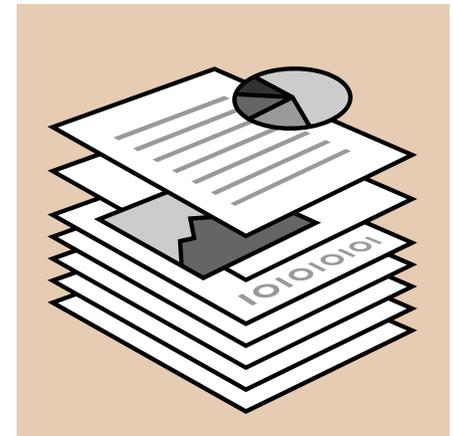
7.1.1 Background & Rationale

Basing AUTOSAR OS on OSEK OS means that legacy applications will be backward compatible – i.e. applications written for OSEK OS will run on AUTOSAR OS.



OSEK/VDX仕様との主な差分

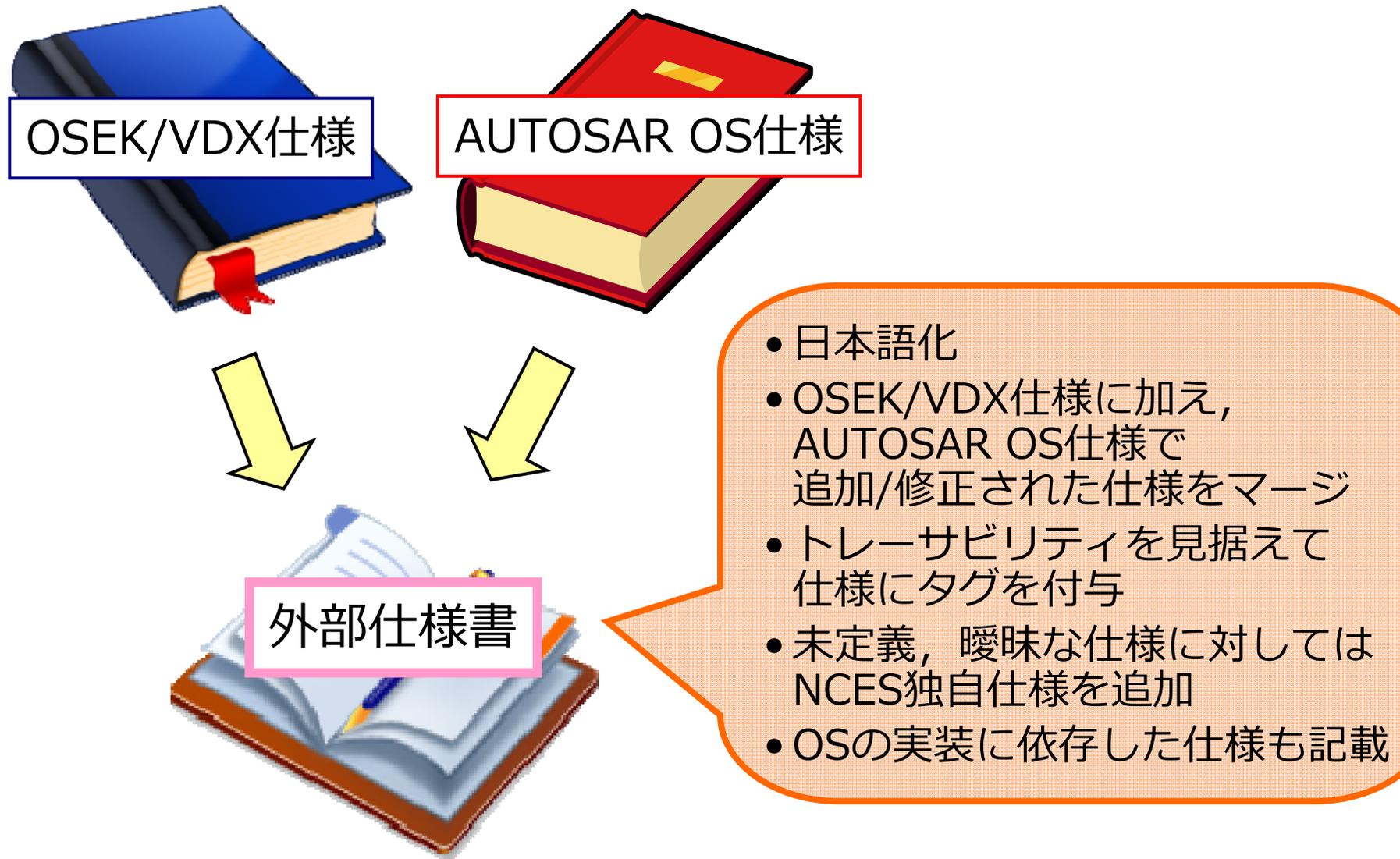
- スケーラビリティクラスの導入
 - タイミング保護, **メモリ保護**の導入
- カウンタをOSオブジェクトとして定義
- スケジュールテーブル
- **OSアプリケーション(メモリ保護のパーティション単位)**
- プロテクションフック(保護違反時処理)
- スタックモニタリング
- コンフィギュレーション方法の変更(XML)
- IOC(Inter OS-Application Communicator)
- **マルチコアのサポート**
 - コアを跨いだシステムサービス呼出し
 - スピンロック



AUTOSAR OS仕様の問題点

- OSEK/VDXとの差分しか記載されていない
 - 仕様に追加や修正を行うと、**差分の差分**を管理する必要がある
 - OSEK/VDX仕様との矛盾/不整合が存在する
 - OSEK/VDX仕様には仕様タグが無い
- **曖昧な仕様や検討不足である仕様が多い**
- 英語で記述されている
 - 日本ではまだ抵抗がある
 - 曖昧な英文も少なくない
- 仕様タグの付与基準が曖昧
 - 要求事項であるのに、仕様タグが付与されていないことがある
 - 複数の要求事項を1つの仕様タグでまとめていることがある

次世代車載システム向けRTOS外部仕様書の作成



トレーサビリティを見据えた仕様タグ

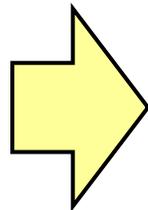
仕様タグ凡例

- 【COSxxxx】 : OSEK仕様(独自にタグを付番)
- 【OSxxx】 : AUTOSAR仕様(原文に付与されているタグ)
- 【OSaxxx】 : AUTOSAR仕様(仕様タグが付与されていない)
- 【NOSxxxx】 : NCES独自仕様
- 【IOSxxx】 : 実装依存仕様
- 【DOSxxx】 : 設計仕様

※〔仕様タグ〕は【仕様タグ】の参照と定義

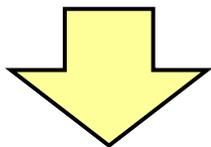
次世代車載システム向け
RTOS外部仕様書

○○○【COS123】



テストデータ
テスト対象
仕様タグ
- COSxxx
- OSxxx

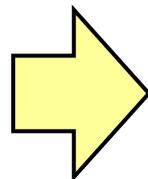
どの仕様に対する
テストデータかを
管理



外部仕様書のどの仕様を基に
設計・実装したかを管理可能

ATK2設計書

△△△【DOS789】〔COS123〕



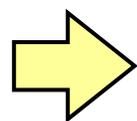
OSソースコード
/* DOS789 */

仕様タグの件数

| | 仕様種別 | 件数 |
|-------------------|------------|------------|
| 外部仕様書 (514ページ) | COS | 427 |
| | OS | 620 |
| | OSa | 177 |
| | NOS | 803 |
| | IOS | 223 |
| | 合計 | 2,264 |

(2013年6月末公開版)

外部仕様書の2,264件の内、NOSとIOSを合わせて1,026件



OSEK, AUTOSARに、数多くある曖昧な仕様や実装依存となっている仕様を明確に規定した

追加した仕様の内訳

| | 種別 | 件数 |
|-------------|-----------------------|-----|
| NOS 803件 | 曖昧, 未規定のために追加した仕様 | 364 |
| | 高オーバーヘッドのために追加/改変した仕様 | 219 |
| | 品質を向上するために追加した仕様 | 220 |

➡ **AUTOSAR仕様の品質は
良いとは言えない・・・**



NCES仕様に改変したAUTOSAR仕様

返り値のデータ型がStatusTypeでないシステムサービス

[OS367] 「Operating System module's services which do not return a StatusType shall not raise the error hook(s). 」 ()

OSEK仕様

- 実行中のタスクIDを取得する
StatusType GetTaskID(TaskRefType TaskID);

タスクIDを受け取る変数のポインタを引数に渡し、実行結果が戻り値となる

AUTOSAR仕様

- 実行中のISR IDを取得する
ISRType GetISRID(void);

実行結果がISR IDとなるため、実行状態のISRが存在しないかエラーが発生したか区別不可
(しかもエラーフックは呼ばれない)

➡ 返り値が無効値となる場合など、エラーフックが呼び出されるべき状況があるため、返り値の型がStatusTypeでないシステムサービスでも、エラーフックが呼び出される場合があることを規定

NCES独自仕様

曖昧な仕様

SC1でのプロテクションフックの扱い

[OS542] [Availability of `ProtectionHook()`: Available in Scalability Classes 2, 3 and 4.] ()

[OS245] [If an instruction exception occurs (e.g. division by zero) the Operating System module shall call the protection hook with `E_OS_PROTECTION_EXCEPTION`.]

SC1でCPU例外が発生したらどうなる？

➡ SC1でもCPU例外やスタックオーバーフローが発生した際にはプロテクションフックを呼び出すように規定

未定義である仕様

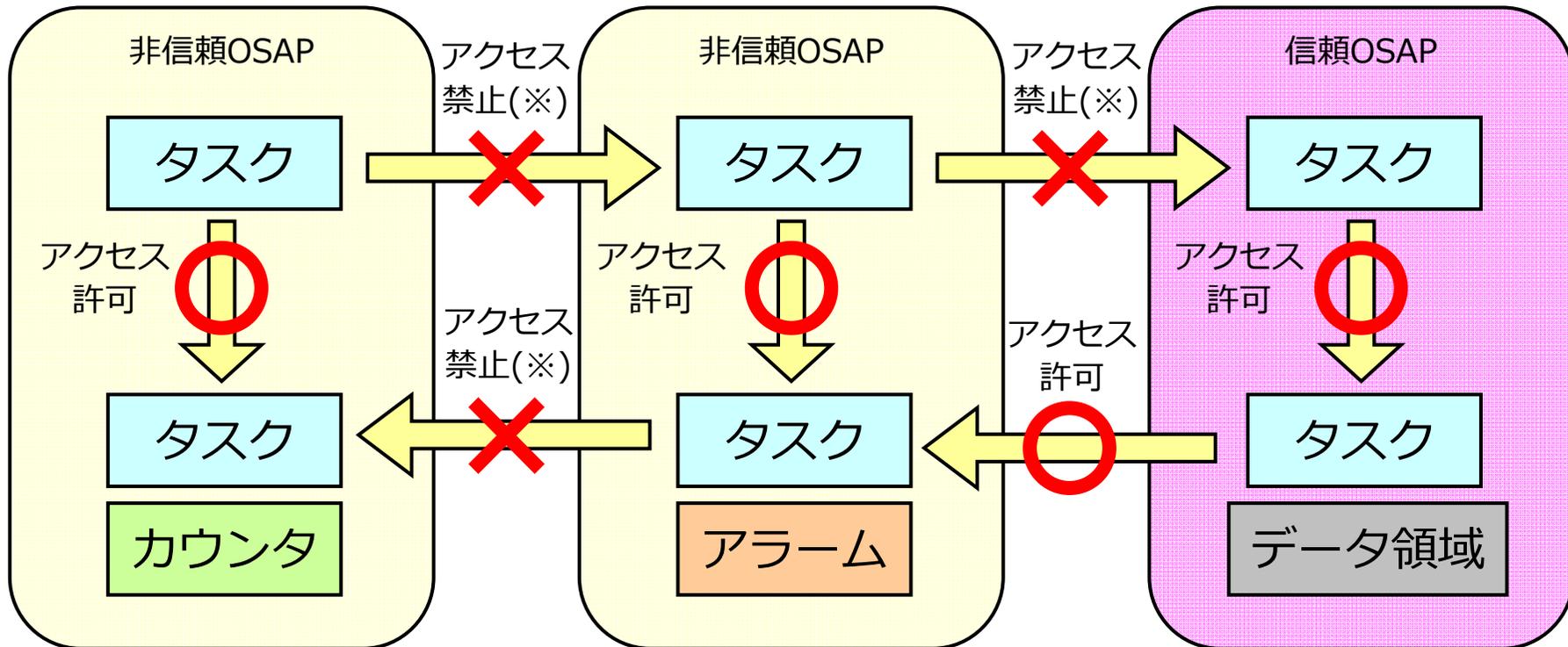
OSが管理するスタック設定

OSEK, AUTOSAR共に, スタック設定に関する記述がない

➡ OSで管理するスタックの種類と, スタックのコンフィギュレーションについて規定

OSアプリケーション(OSAP)とは

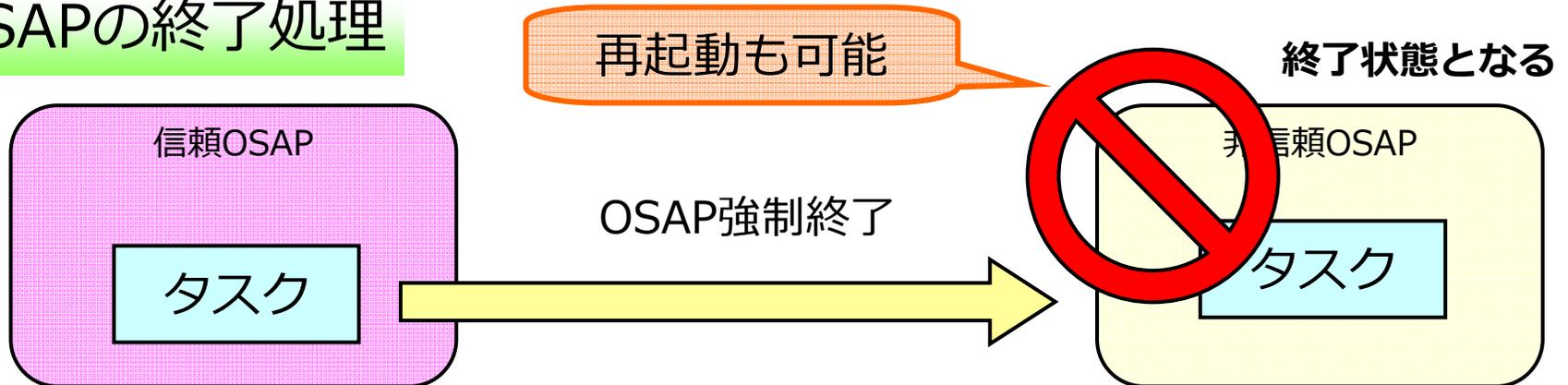
『アプリケーション毎に分割した複数のOSオブジェクトの集合』



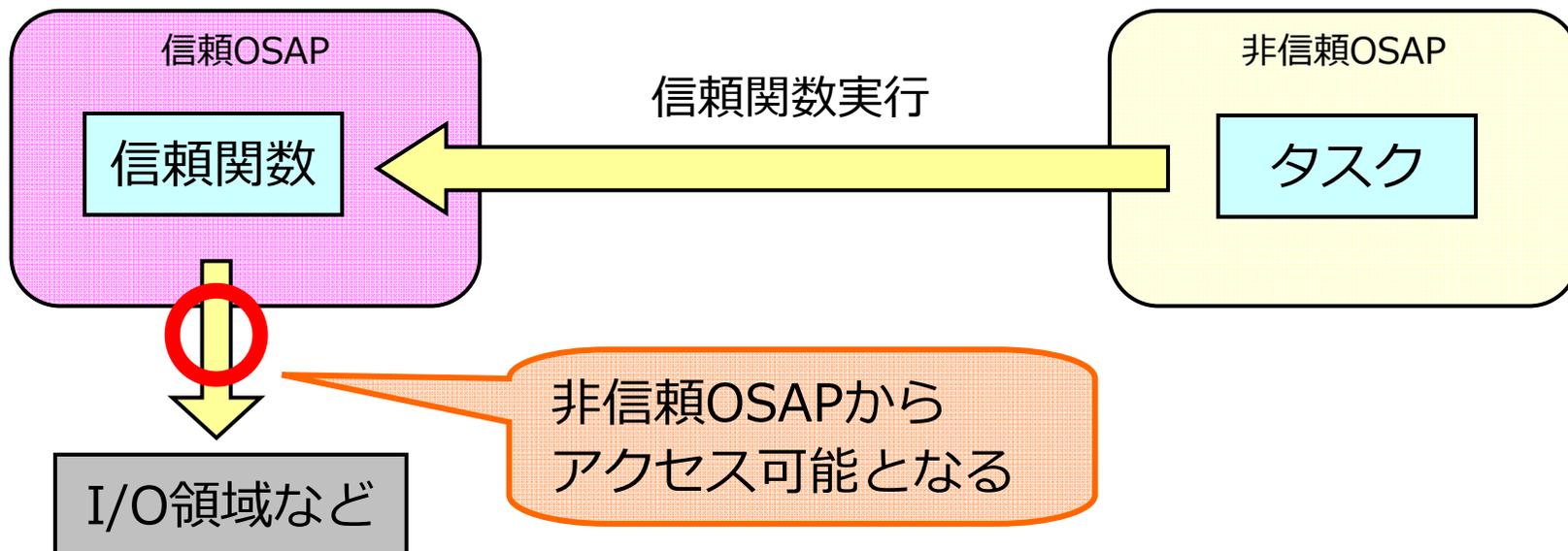
※アクセス権を付与することでアクセス可能となる

OSAPに対して提供する機能

OSAPの終了処理



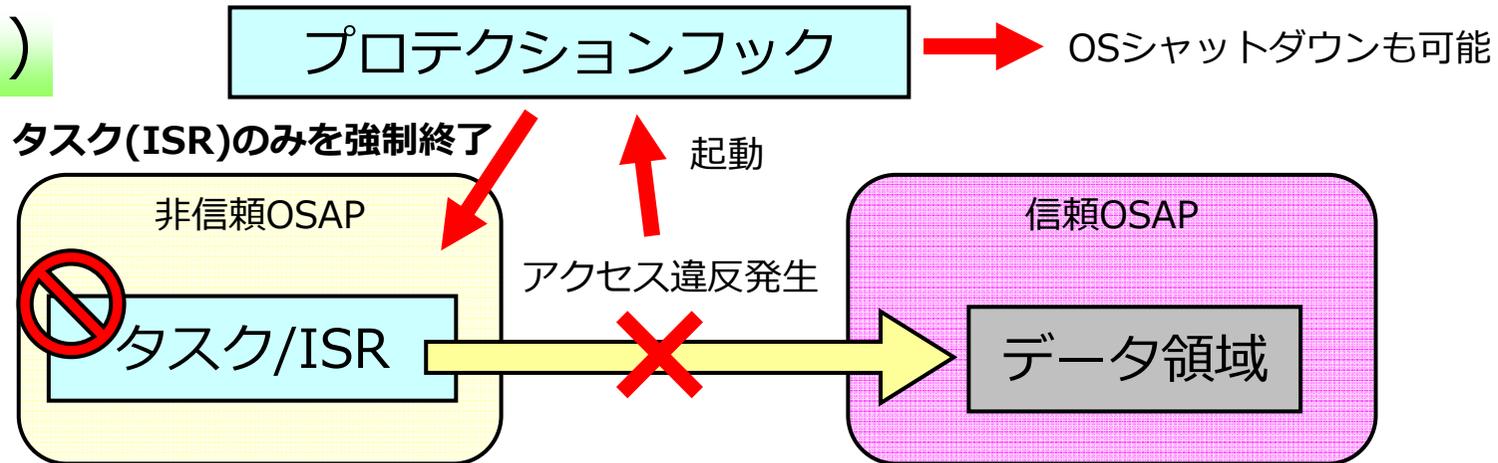
信頼関数



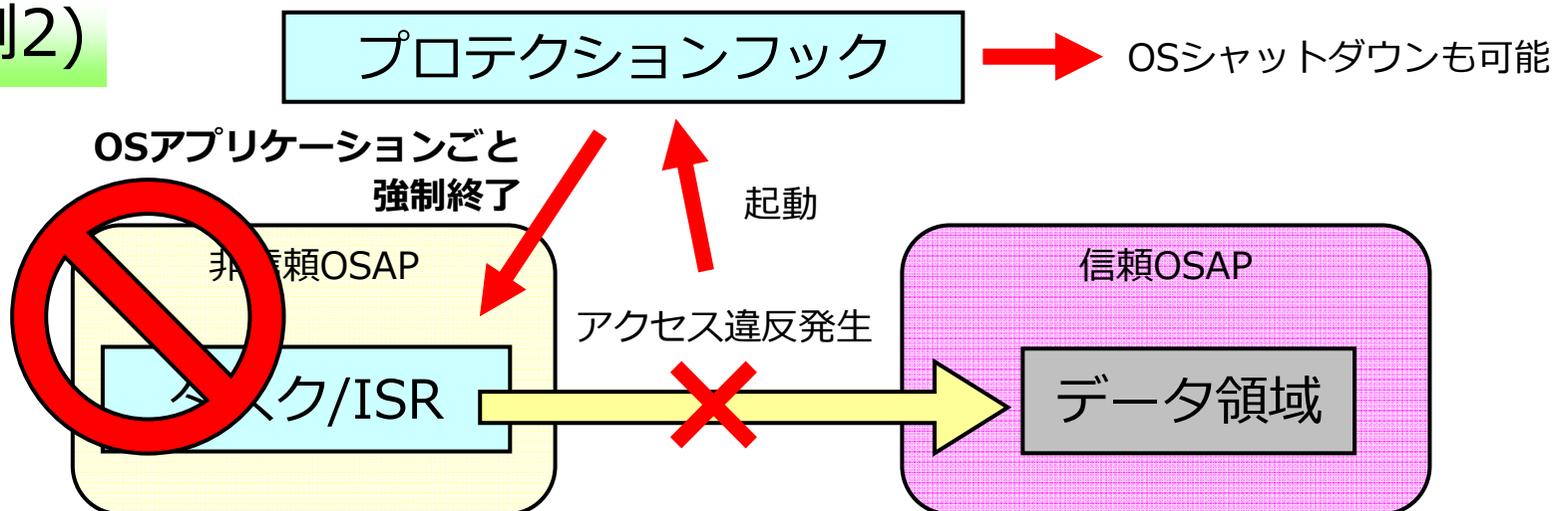
メモリ領域に対する保護と保護違反時の機能

保護違反からの復旧処理をユーザが選択可能

例1)



例2)



メモリ保護機能の問題点①

リソース獲得中のOSAP強制終了

他のOSAPに含まれるリソースを取得していた場合
そのOSAPが強制終了されるとリソースはどうなるのか？

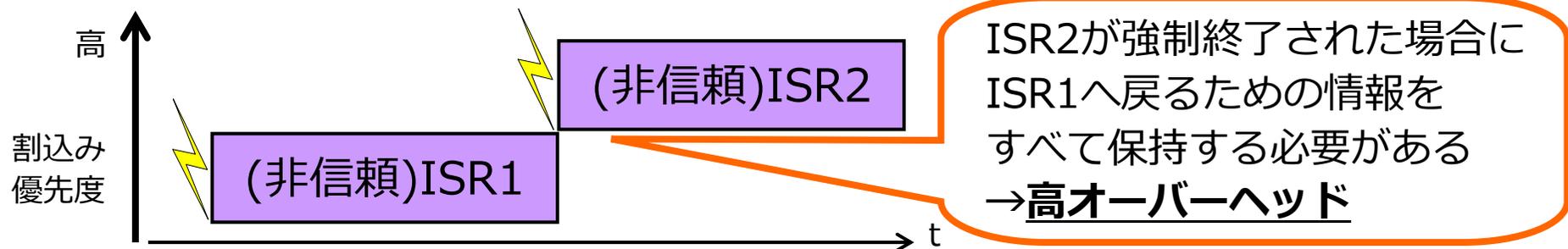


AUTOSAR R4.0.2 → R4.0.3で解決
→ リソースはOSAPに所属しなくなった

メモリ保護機能の問題点②

非信頼OSAPのISR

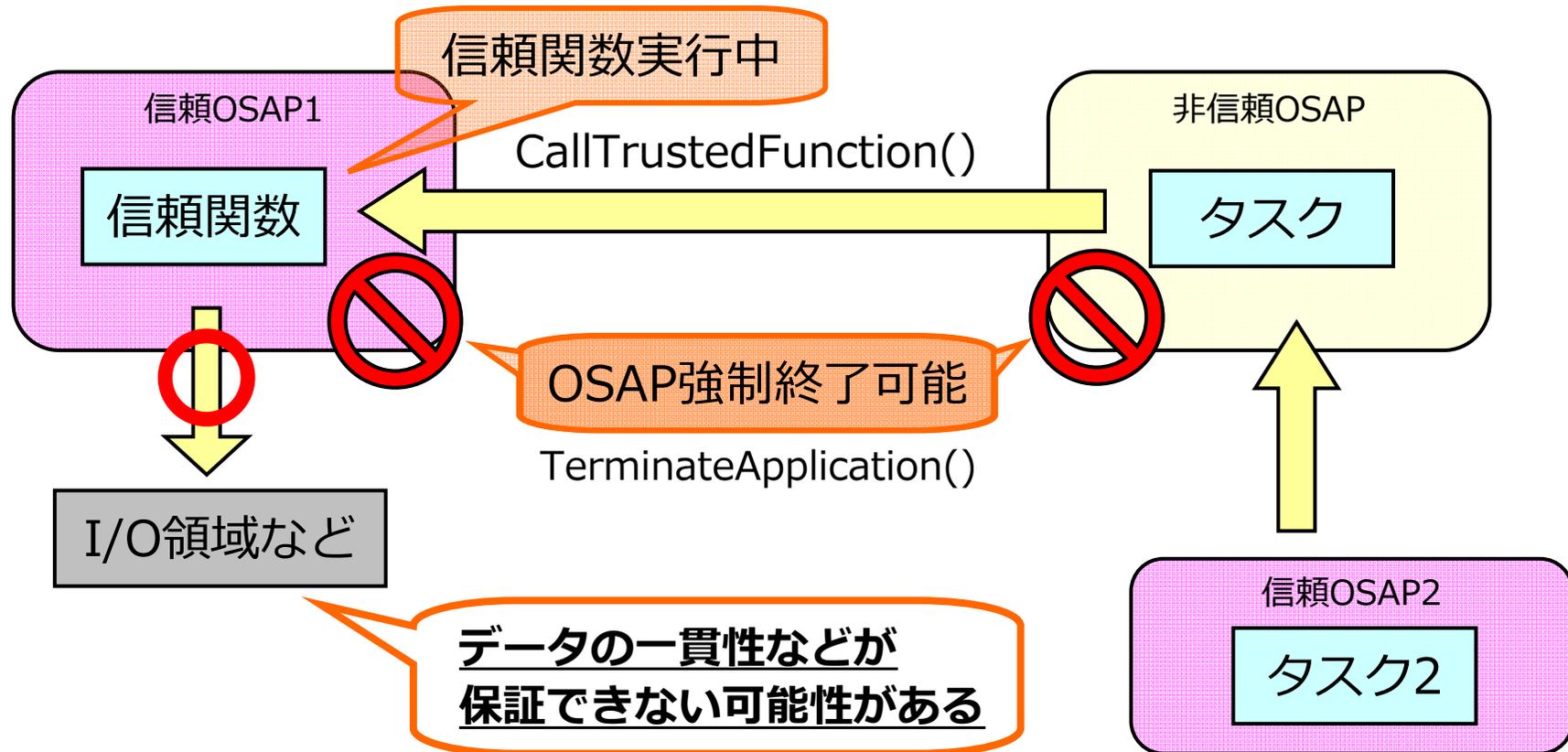
- 非信頼OSAPに所属するISRを実現する場合、割込み発生後、MPUを非特権モードに切り換えメモリ保護属性を設定する必要がある
- 非信頼ISRからシステムサービスを呼び出す場合、ソフトウェア割込み等で特権に切り換える必要がある
- ISRが強制終了される場合に備え、多重割込み発生時は、元のISRに関する情報を保持しておく必要がある
- ISRの強制終了を無効にしても、OSAP全体を強制終了される場合に、同様の問題が発生する



メモリ保護機能の問題点③

信頼関数, 信頼OSAPの終了

- 信頼関数実行中に終了してよいか？
- 信頼OSAPを強制終了する意義はあるか？



機能レベルの導入

メモリ保護の機能レベル

- 機能レベル1
 - 非信頼OSAPに所属するタスクをサポート
- 機能レベル2
 - 非信頼OSAPに所属するフックルーチンをサポート
- 機能レベル3
 - 非信頼OSAPに所属するISRをサポート

ATK2では
機能レベル2を採用！

保護違反白処理の機能レベル

- 機能レベル1
 - 保護違反時, OSシャットダウンをサポート
- 機能レベル2
 - 保護違反時, タスクの強制終了をサポート
- 機能レベル3
 - 保護違反時, ISRの強制終了, OSAPの強制終了/再起動をサポート

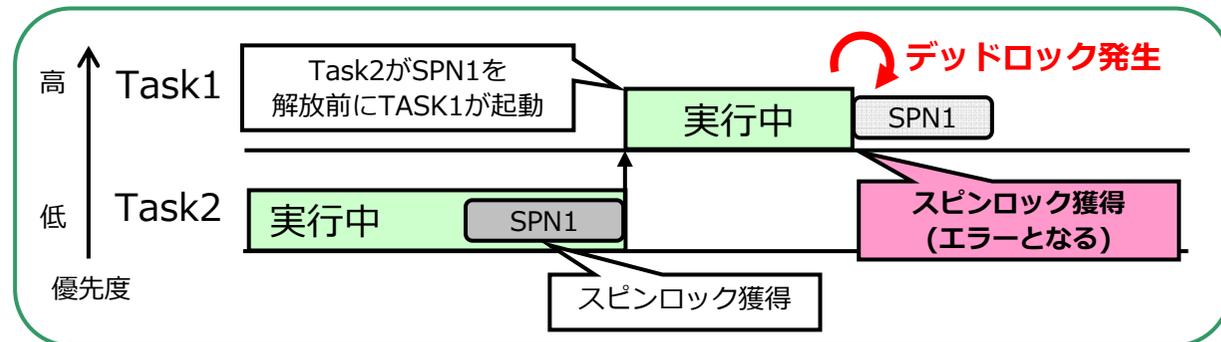
非信頼OSAPの強制終了,
再起動のみサポートする
(2013年9末予定)

スピンロックとは

- **マルチコア**において、異なるコアのタスク、ISR間の**排他制御**に用いるOSオブジェクト
- 他のコアで獲得されたスピンロックを獲得しようとした場合、スピンロックが解放されるまでビジーウェイトする
- **AUTOSAR仕様ではスピンロック獲得中に割り込み禁止としない**
- デッドロック対策用のエラーが2つ用意されている

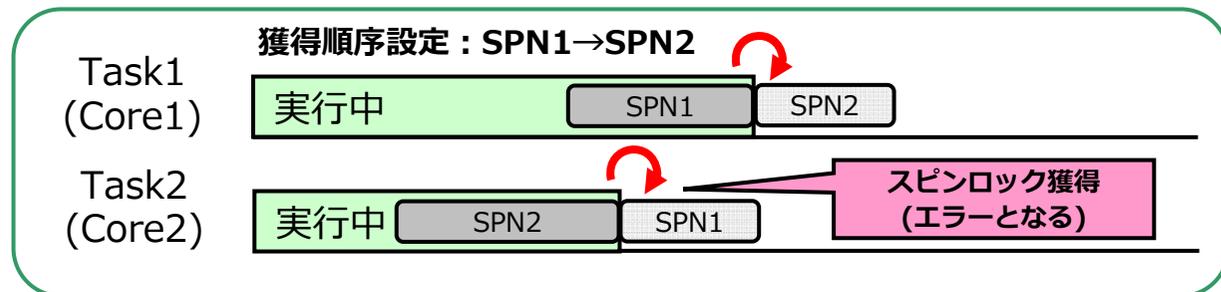
エラー1

同じコアのタスク/ISRが獲得しているスピンロックを獲得しようとした場合

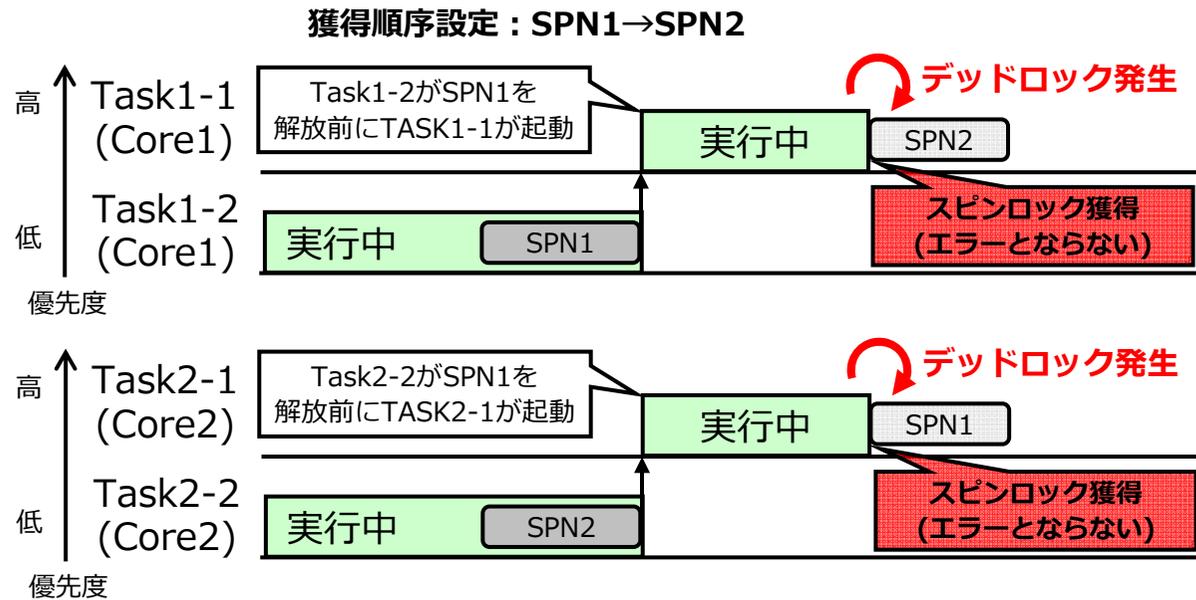


エラー2

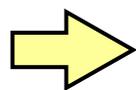
コンフィギュレーション時に指定した獲得順序を守らずにスピンロックを獲得しようとした場合



スピンロックの問題点



- 獲得順序エラーは同じタスク/ISRに対してのみ有効であるので上図のシーケンスで発生するデッドロックは防止できない
- スピンロック獲得時は割り込み禁止とすれば本デッドロックは発生しない
- **NCES独自仕様で割り込み禁止を伴うスピンロック仕様を策定**



R4.1.1で同様のスピンロックが導入された

目次

1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

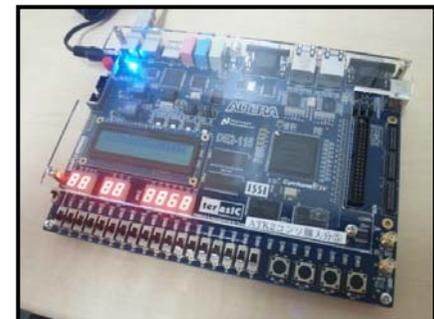
AuTomotive Kernel version2(ATK2)

ATK2概要

- 外部仕様書に準拠したOS
- **SC1, SC3(機能レベル2)**を**2013年1月**に公開
 - Nios2プロセッサ用のターゲット依存部を同梱
 - SC1のみ以下の依存部も同梱
 - SkyEye(ARMシミュレータ)
 - V850E2(ルネサスエレクトロニクス)
- **SC1-MC, SC3-MC(機能レベル2)**を**2013年6月**に公開予定

開発に使用したボード

- Altera DE2-115 Development and Education Board
- Cyclone IV EP4CE
- メモリ保護, マルチコアに対応
- HWイメージのコンフィギュレーション情報ファイルをOSに同梱予定



OSのコンフィギュレーション方法

ATK2では、2つの方法をサポート

ディスクリプションファイル(XML)

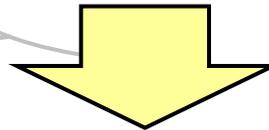
- AUTOSAR準拠のコンフィギュレーション記法
- AUTOSARのメソドロジでは、上位レイヤからディスクリプションファイルを受け取って、各モジュールのコンフィギュレーションを行う
- 不足しているパラメータコンテナを独自に追加
- 人が読み書きするのは困難

静的API(μ ITRON仕様)

- μ ITRONで規定されている静的APIの記法を元に、XMLによるコンフィギュレーションと同等の機能を実現
- 人が読み書きするのも容易

テストスイートによるATK2の検証

過去のコンソーシアム型共同研究で開発した
テストスイートをAUTOSAR OSの検証に活用したい…



TOPPERS新世代カーネルを対象にしたテストスイートである
TTSP(TOPPERS Test Suite Package)をベースに
AKTSP(Automotive Kernel Test Suite Package)を開発

補足：TOPPERS新世代カーネル

μITRON仕様をベースとして、信頼性、安全性、ソフトウェア
ポータビリティを向上させるために改良・拡張

シングルプロセッサ向けRTOS：TOPPERS/ASPカーネル

マルチプロセッサ向けRTOS：TOPPERS/FMPカーネル

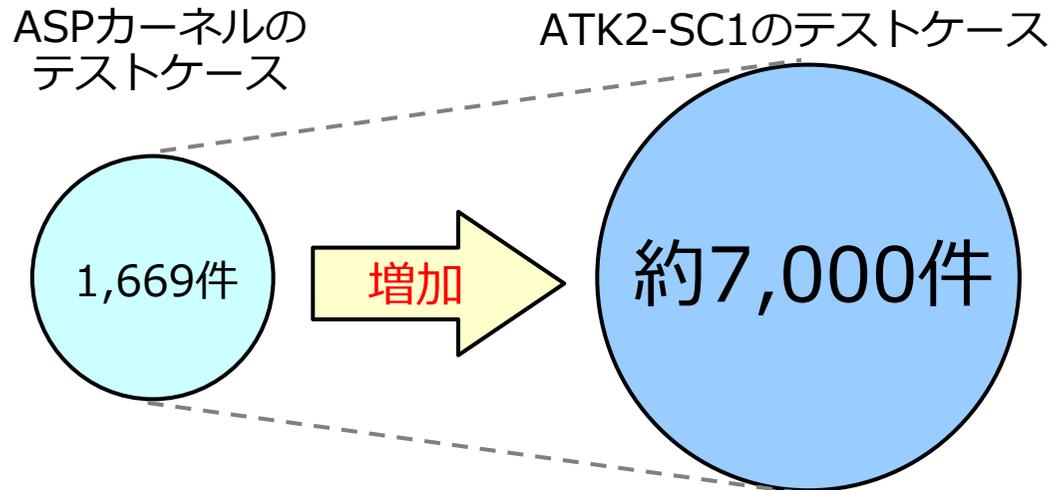


組み合わせテストツール

検査対象OSをμITRON仕様からAUTOSAR仕様に変更

OSが提供するシステムサービスの数は減少したが…

仕様の複雑化によりテストとして考慮すべき組み合わせパターンが増大
→タスクのコンフィギュレーションの組み合わせだけでもASPの**4倍**



従来のテスト開発プロセスで行っていた
組み合わせテストケースの手動作成が困難に

➡ PictMaster(※)を使用して組み合わせを自動生成！

※ <http://sourceforge.jp/projects/pictmaster/>

テストプログラムの自動生成

PictMasterが生成した
CSVファイルから自動生成

形式化したテストシナリオ

前状態

<システムサービス発行前のシステム状態>

処理

<システムサービス発行処理>

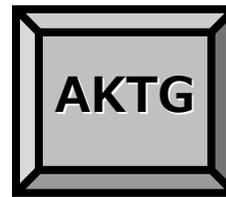
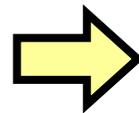
後状態

<システムサービス発行後のシステム状態>

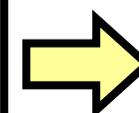
TESRY記法

(TEst Scenario for Rtos by Yaml)

入力



出力



(Automotive Kernel Test
Generator)

- ・Rubyで開発
- ・約23,000行

テストシナリオを
実現するテストプログラム

test.arxml

test.h

test.c

AKTGはTTSPで開発した
TTG(TOPPERS Test Generator)を
ベースに開発

『μITRONベースのRTOS向けテストプログラム生成ツール』
電子情報通信学会論文誌D, Vol.J95-D, No.4, pp.870--884

テストケース数

| | | SC1 | | SC3 | |
|----------------|----------------|------------|--------------|------------|--------------|
| | | シングル コア | マルチコア 追加分 | シングル コア | マルチコア 追加分 |
| システムサービステスト | | 6,841 | 23,303 | 37,717 | 82,862 |
| MODISTARC | | 66 | – | 83 | |
| エラー処理 テスト | 不正終了 | 3,839 | – | 29,093 | – |
| 保護違反時 処理テスト | スタック モニタリング | 240 | – | 790 | – |
| | CPU例外 | 62 | – | 148 | – |
| | メモリ保護(※) | – | – | 42,648 | – |
| | タスク 強制終了 | – | – | 1,655 | – |
| IOCテスト | | – | 3,710 | – | 11,736 |
| 合計 | | 10,982 | 27,013 | 112,051 | 94,598 |

※Nios2の場合

(2013年5月現在)

ATK2設計書およびAKTSPの取り扱い

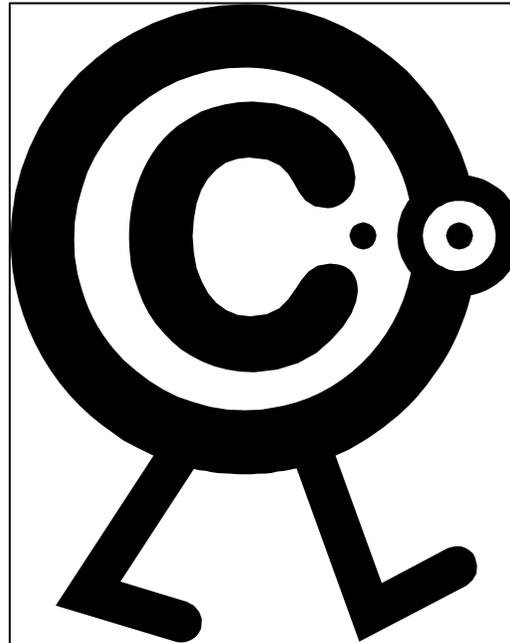
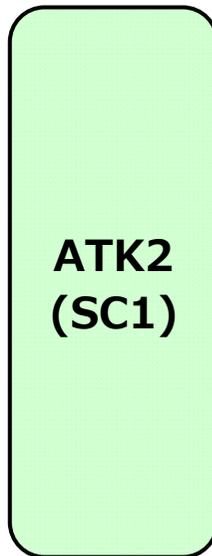
- 外部仕様書, ソースコードのみオープンソースとし,
設計書, テストスイート(AKTSP)は公開しない
- コンソーシアム型共同研究参加企業のみが入手可能
- ただし, 希望する企業には**有償でライセンス提供可能**
 - 既に国内外の複数企業から問い合わせあり
 - 実際にライセンス提供を希望する企業も
- ライセンス料は共同研究費の価格をベースに,
参加企業が不利にならない価格帯を設定
- ライセンス料は年度を跨ぐ毎に減額し,
最終的には無償で公開する予定(5~6年後)

ご興味のある方はお問い合わせください

atk2-conso-staff@nces.is.nagoya-u.ac.jp

NCES

Elektrobit社製 tresos AutoCore上での動作確認



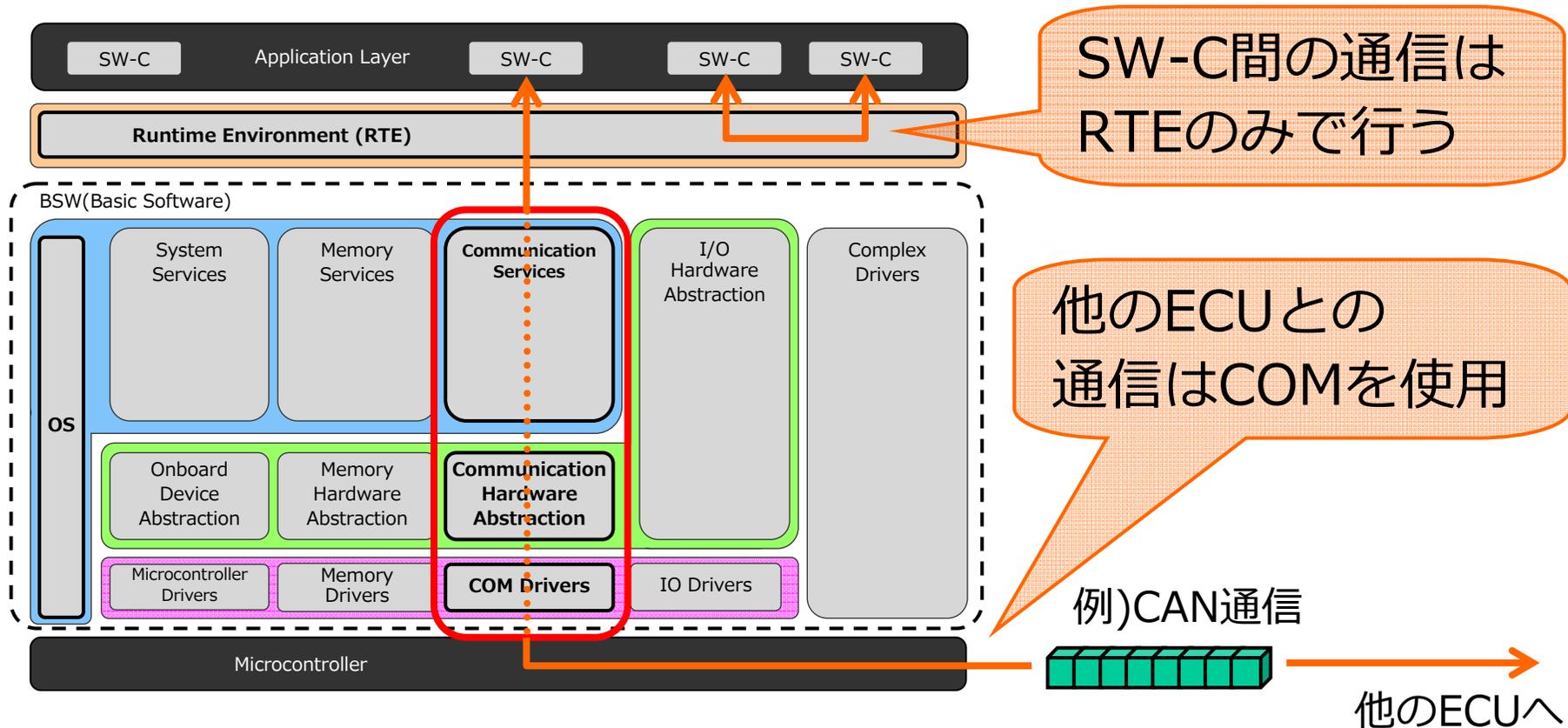
<Elektrobit <http://automotive.elektrobit.com/>>

- tresos上のOSをATK2へ置き換えて動作することを確認
- ターゲットは, V850E2プロセッサを使用

目次

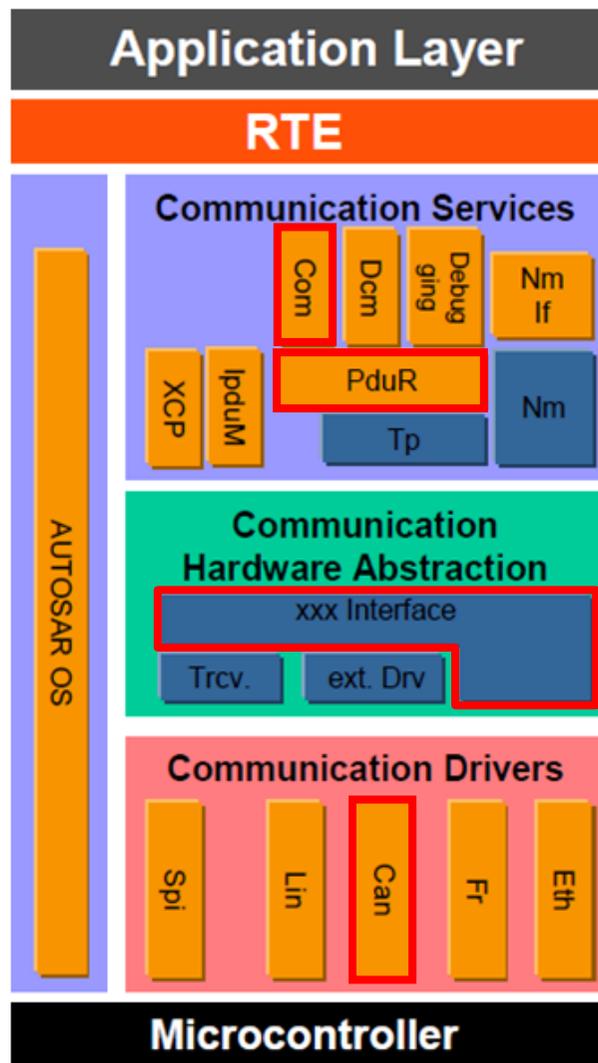
1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

COMスタックとは



- 他のECUとのデータ通信を行う
- SW-Cには通信プロトコルは隠蔽される
- 部分的にOSEK/COM仕様をベースとしている

開発範囲

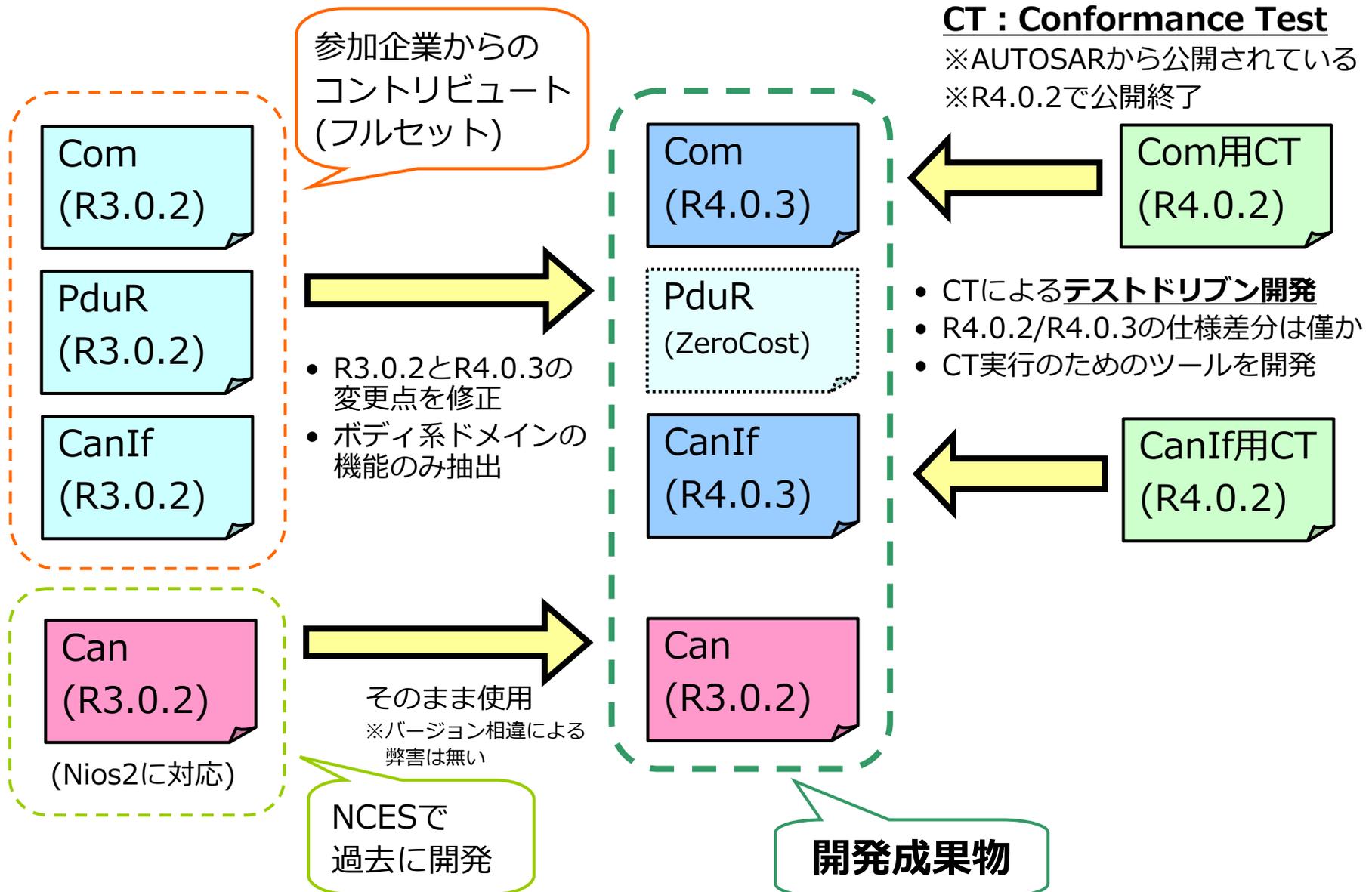


- COMスタックすべての仕様は膨大である
- ボディ系ドメインで使用されるであろう機能に絞ったサブセットを対象とする
- 通信プロトコルはCAN通信のみとする
- 対象バージョンはOS同様, R4.0.3とする
- 工数に余裕があればマルチコアに対応する

| コンポーネント | 機能 |
|-------------|--|
| Com | 信号ゲートウェイ機能 |
| PduR | PDU(Protocol Data Units)のルーティング ※CAN通信のみの場合, 実質不要 |
| CanIf | CANプロトコルコントローラとCANトランシーバの抽象化 |
| Can(Driver) | CANの初期化, CANの入出力 |

<R4.0.3 "AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf" P.53 抜粋>

開発手法



コンフォーマンステスト実行方法

AUTOSARから公開

R4.0.2準拠

テストシーケンス
(* .ttcn)

テストライブラリ
(* .ttcn)

パラメータファイル
(* .par)

ディスクリプション
ファイル(*.arxml)

テスト仕様書
(* .pdf)

- **TTCN-3**というテスト用言語
- TTCNツールで対応するには膨大なコストを要する
- **独自に変換ツールを開発**

TTCNと等価の処理内容となる
C/C++コードへ変換

変換ツール

- 設定値チェック用のデータ定義に使用
- R4.0.3の仕様に合わせたディスクリプションファイルを作成

変換ツール

異なる形式で
等価な情報を記載

R4.0.2のディスクリプション
ファイルは、パラメータが
若干異なるので使用しない

テストシーケンス
(* .c|cpp|h)

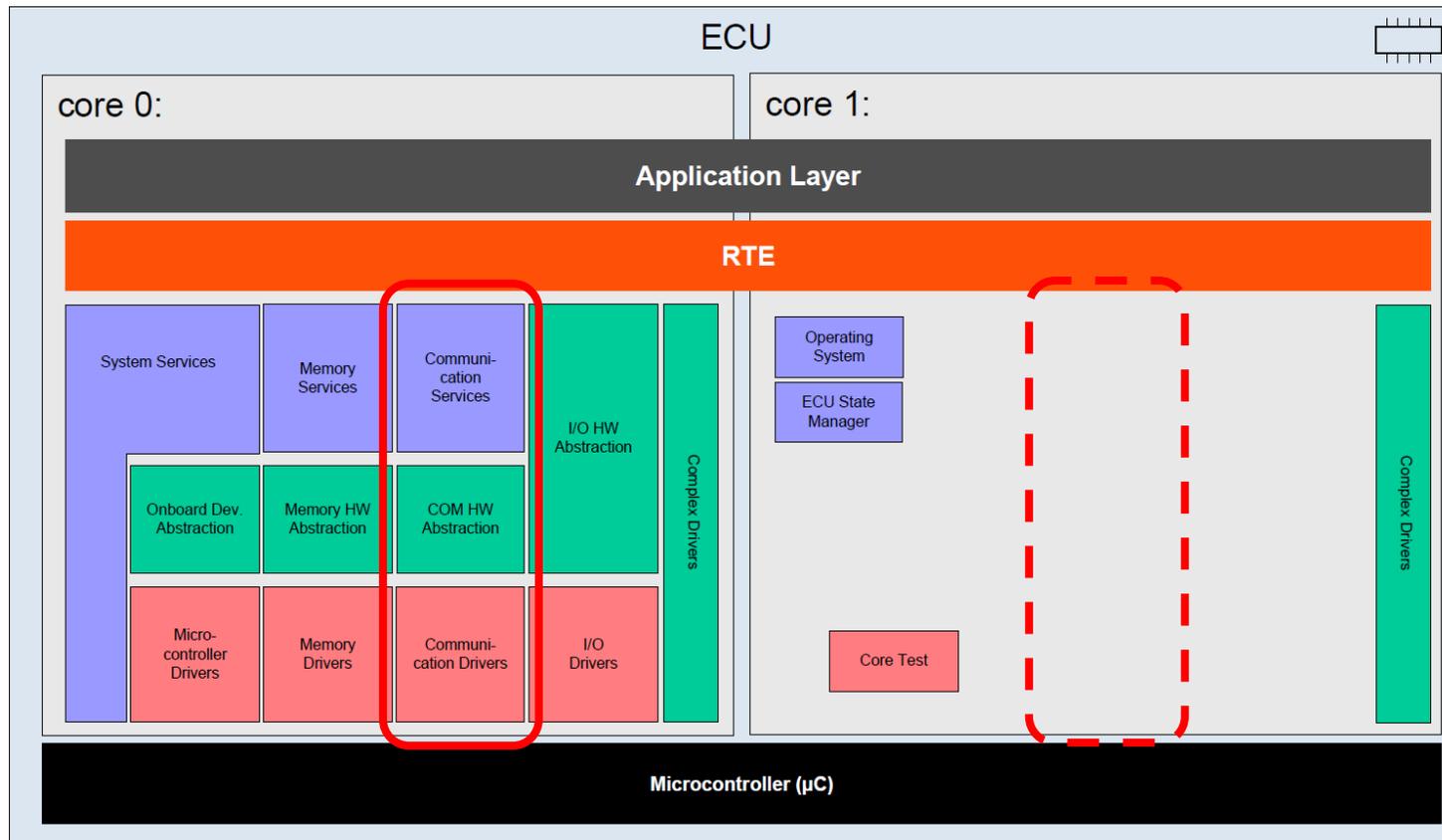
テストライブラリ
(* .c|cpp|h)

データ定義ファイル
(* .c|h)

ディスクリプション
ファイル(*.arxml)

生成したファイルを使用し、
Com/CanIfのテストを実行

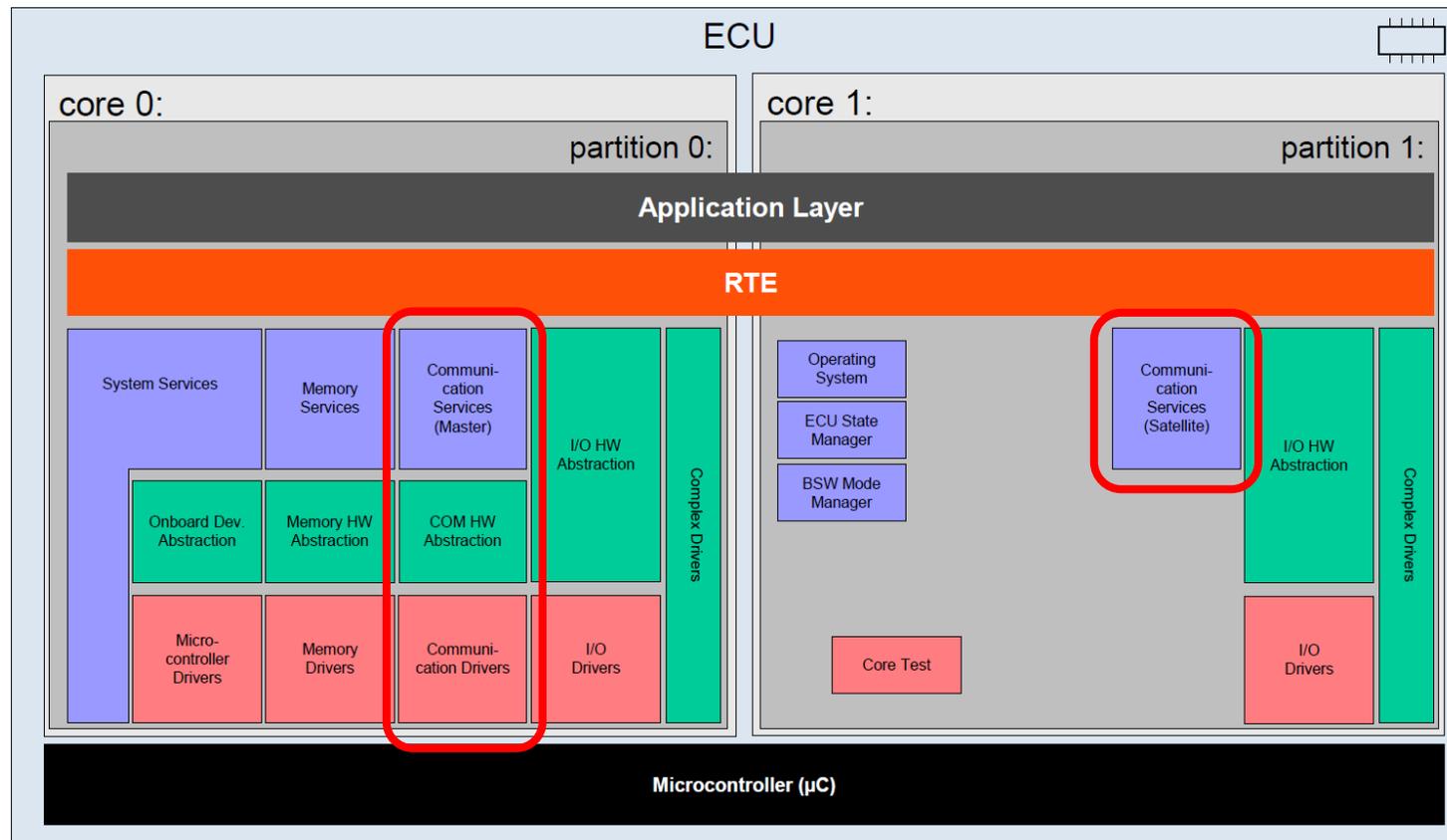
マルチコア仕様(R4.0.3)



<R4.0.3 "AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf" P.53 抜粋>

- マスタコアでしかComによる通信ができない
- スレーブコアからのCom通信が高オーバーヘッドとなる
 - マスタコアを毎回経由する必要がある

マルチコア仕様(R4.1.1)



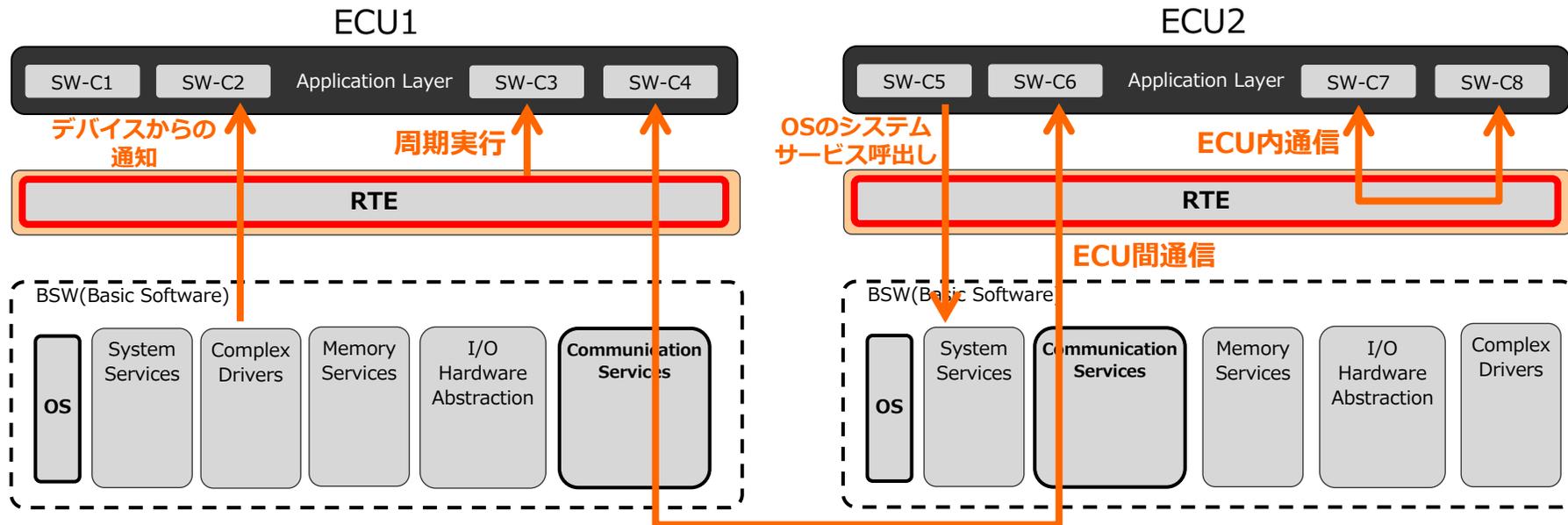
<R4.0.3 "AUTOSAR_EXP_LayeredSoftwareArchitecture.pdf" P.53 抜粋>

- "マスター-サテライト通信"をサポート
- NCESで過去に同様の研究実績がある
 - 工数に余裕があれば適用したい

目次

1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

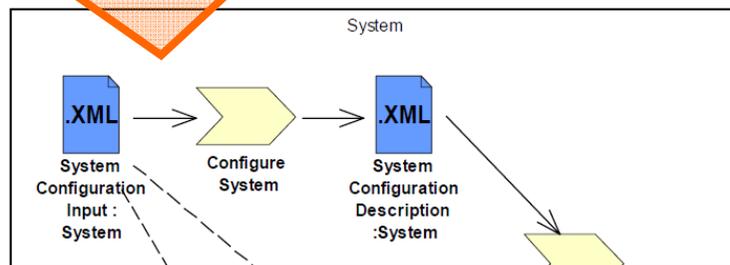
RTEとは



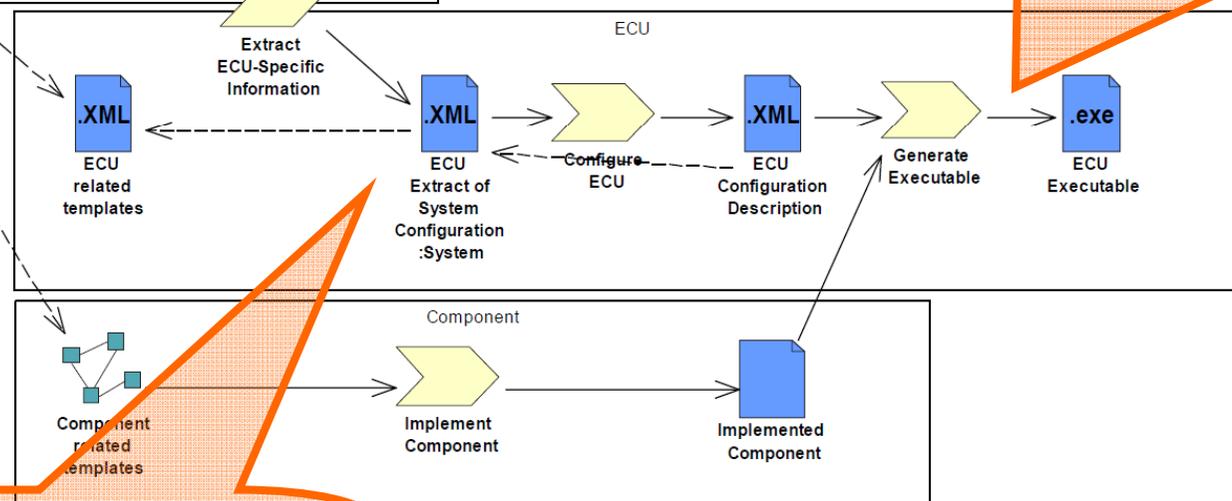
- コンフィギュレーション情報に従って、**SW-CやBSWの実行スケジューリング**を制御する
- コンフィギュレーション情報に従って、効率的な**SW-C間の通信**を実現する
- コンフィギュレーション情報に従って、**RTEジェネレータ**から、必要なAPIが生成される
 - RTEの実体は**自動生成コード**となる

メソドロジ(実行モジュールの生成方法)

ステップ1：
システムコンフィギュレーション



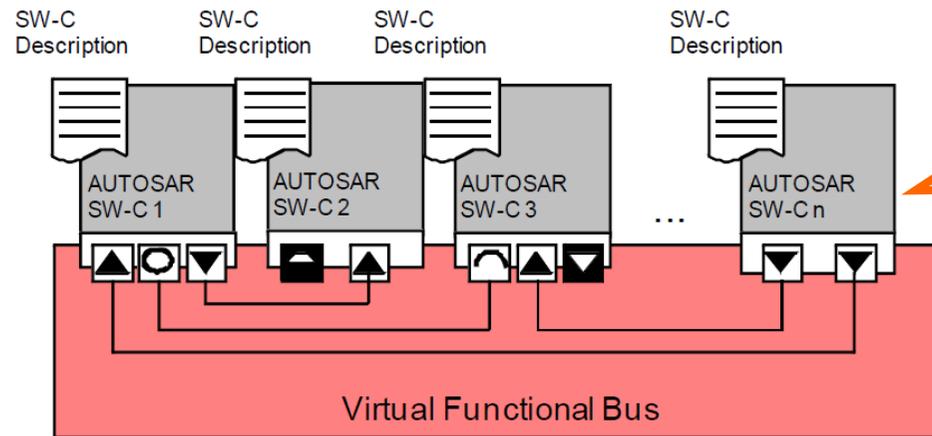
ステップ3：
コンフィギュレーション情報を
元に実行モジュールを作成する



ステップ2：
ECUコンフィギュレーション

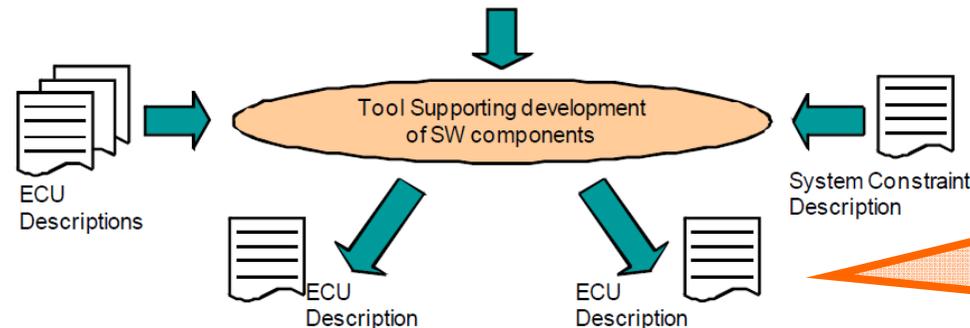
<R4.0.3 "AUTOSAR_EXP_VFB.pdf" Figure 2.1>

システムコンフィギュレーション

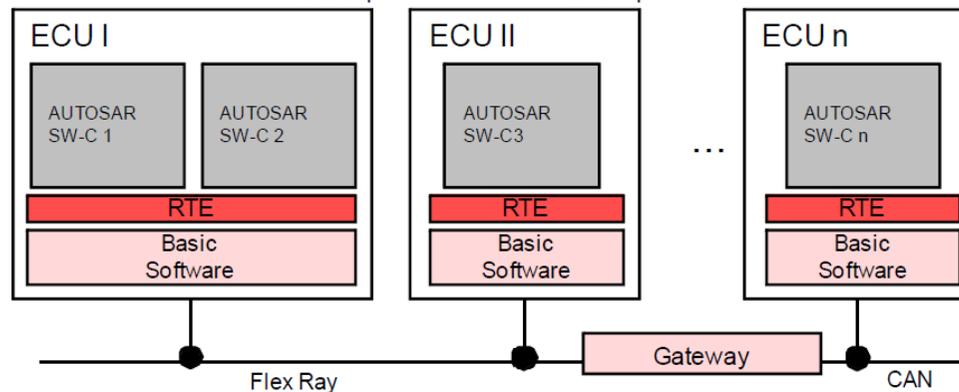


SW-C間で使用するI/F・通信や
どのECUで動作させるかを定義する

VFBによって抽象化され、SW-C
からはRTEより下位層は見えない



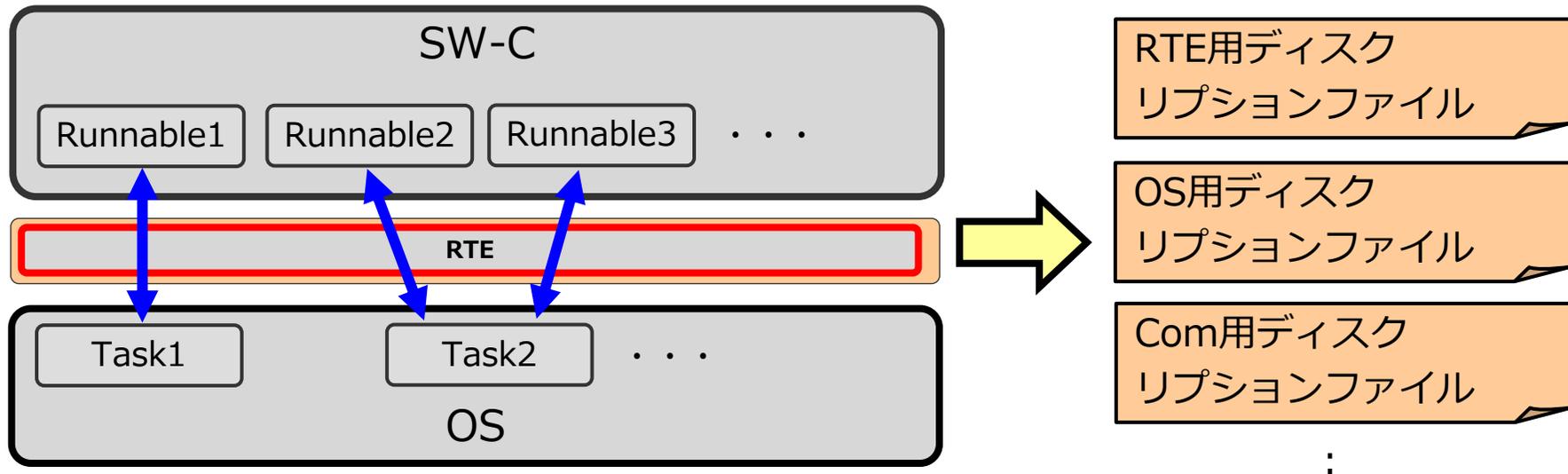
SW-Cを実行するECUの指定、
必要なRTE API情報等をディスクリプションファイルで生成



必要なI/F・通信に応じて、
最適なAPIがRTEジェネレータ
によって生成される

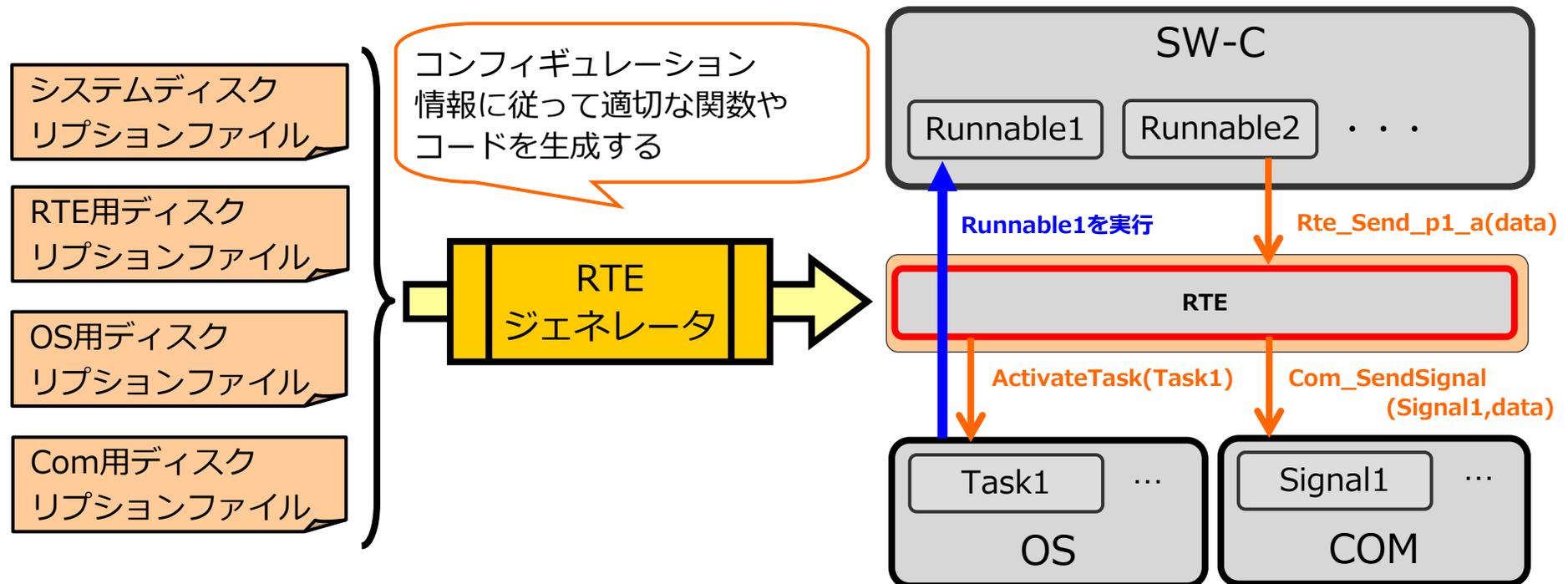
<R4.0.3 "AUTOSAR_EXP_VFB.pdf" Figure 2.2>

ECUコンフィギュレーション



- SW-Cは、複数の**Runnable**と呼ばれるアプリケーション処理の単位で構成される
 - アプリケーション開発者はRunnableを開発する
- どのRunnableがOS上のどのタスクで実行されるか、排他制御には何を使用するか等を設定する
 - RTE, OS用のディスクリプシヨンファイルを作成する
- 同様にCom等のBSW用のディスクリプシヨンファイルを作成する
- **各ディスクリプシヨンファイルをツールで生成するかはユーザ依存**

RTEの生成



- OSの機能を使用して, Runnableのスケジューリングを行う
- SW-Cから, どのECU・通信路を使用しているかを見えないように隠蔽する(ラッパー関数を生成する)
- OS用, COM用ディスクリプションファイルは, OS, COMのジェネレータの入力としても使用する

開発状況

- JasParにて開発実績のある企業への委託開発を予定
 - 対象バージョンはOS, COM同様, **R4.0.3**とする
 - 対象とする仕様はCOMに合わせてボディ系ドメインのみ
 - OSがマルチコアに対応することから,
マルチコアに関する機能は追加する予定
- 2013年10月末に開発を完了し, OS・COMとの
インテグレーションを実施
- システム設計ツールは**dSPACE社製SystemDesk4.0**を
使用する予定
- 最終的に, OS・COMを合わせてTOPPERSプロジェクト
から**オープンソース化**する

目次

1. AUTOSARとは
2. NCESの取り組み
3. AUTOSAR OS
4. TOPPERS/ATK2の紹介
5. COMスタック
6. RTE
7. まとめ

まとめ

- **AUTOSAR**仕様のプラットフォームを開発
 - まずは, **OS/COM/RTE**を提供
 - OSはメモリ保護, マルチコアに対応
 - タイミング保護のみ未サポート
 - COM/RTEはボディ系ドメインに限定したサブセット
 - 最終的に**オープンソース**で提供する予定
- 今後の展望
 - OSのタイミング保護対応
 - COM/RTEの機能拡充
 - 他のBSWの開発

ご意見・ご質問のある方はお問い合わせください

atk2-conso-staff@nces.is.nagoya-u.ac.jp

NCES ATK2

検索

