

車載組込みシステム開発の現状・動向と mrubyへの取り組み

2017年9月1日

高田 広章

名古屋大学 未来社会創造機構 教授

名古屋大学 大学院情報学研究科 教授

附属組込みシステム研究センター長

APTJ株式会社 代表取締役会長・CTO

Email: hiro@ertl.jp URL: <http://www.ertl.jp/~hiro/>

自己紹介

本務

- ▶ 名古屋大学 未来社会創造機構 教授
- ▶ 名古屋大学 大学院情報学研究科 情報システム学専攻 教授 / 附属組込みシステム研究センター長

その他の役職(主なもの)

- ▶ APTJ株式会社 代表取締役会長・CTO
- ▶ TOPPERSプロジェクト 会長
- ▶ 車載組込みシステムフォーラム(ASIF) 会長
- ▶ 情報処理学会 組込みシステム研究会 元(初代) 主査

研究分野

- ▶ (組込みシステム向け)リアルタイムOS
- ▶ リアルタイム性保証技術, 消費エネルギー最適化技術
- ▶ 機能安全技術, 組込みシステムのサイバーセキュリティ
- ▶ 車載組込みシステムと車載ネットワーク, ダイナミックマップ

TOPPERSプロジェクト



- ▶ ITRON仕様の技術開発成果を出発点として、組込みシステム構築の基盤となる各種の高品質なオープンソースソフトウェアを開発するとともに、その利用技術を提供

組込みシステム分野において、Linuxのように広く使われるオープンソースOSの構築を目指す!

プロジェクトの狙い

- ▶ 決定版のITRON仕様OSの開発 ← **ほぼ完了**
- ▶ 次世代のリアルタイムOS技術の開発
- ▶ 組込みシステム開発技術と開発支援ツールの開発
- ▶ 組込みシステム技術者の育成への貢献



プロジェクトの推進主体

- ▶ 産学官の団体と個人が参加する産学官民連携プロジェクト
- ▶ 2003年9月にNPO法人として組織化

開発成果物の主な利用事例



エスクード (スズキ)



スカイラインハイブリッド (日産)



IPSiO GX e3300 (リコー)



H-IIIB (JAXA)



Cell³iMager duos
(SCREEN
ホールディングス)



OSP-P300
(オークマ)



SoftBank
945SH
(シャープ)



UA-101 (Roland)



PM-A970(エプソン)

名古屋大学 組込みシステム研究センター (NCES)

設立目的

☞ <http://www.nces.is.nagoya-u.ac.jp/>

- ▶ 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

活動領域(スコープ)

- ▶ 大学の技術シーズを実現/実用化することを指向した研究
- ▶ 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

研究プロジェクトの例

- ▶ 車載制御システム向けSPF (AP / A2Pコンソーシアム)
- ▶ ダイナミックマップ (DM2.0コンソーシアム)
- ▶ 宇宙機向けソフトウェアプラットフォーム (スペースワイヤOS)
- ▶ 車載組込みシステムのセキュリティ強化技術

APTJ株式会社



APTJの必要性

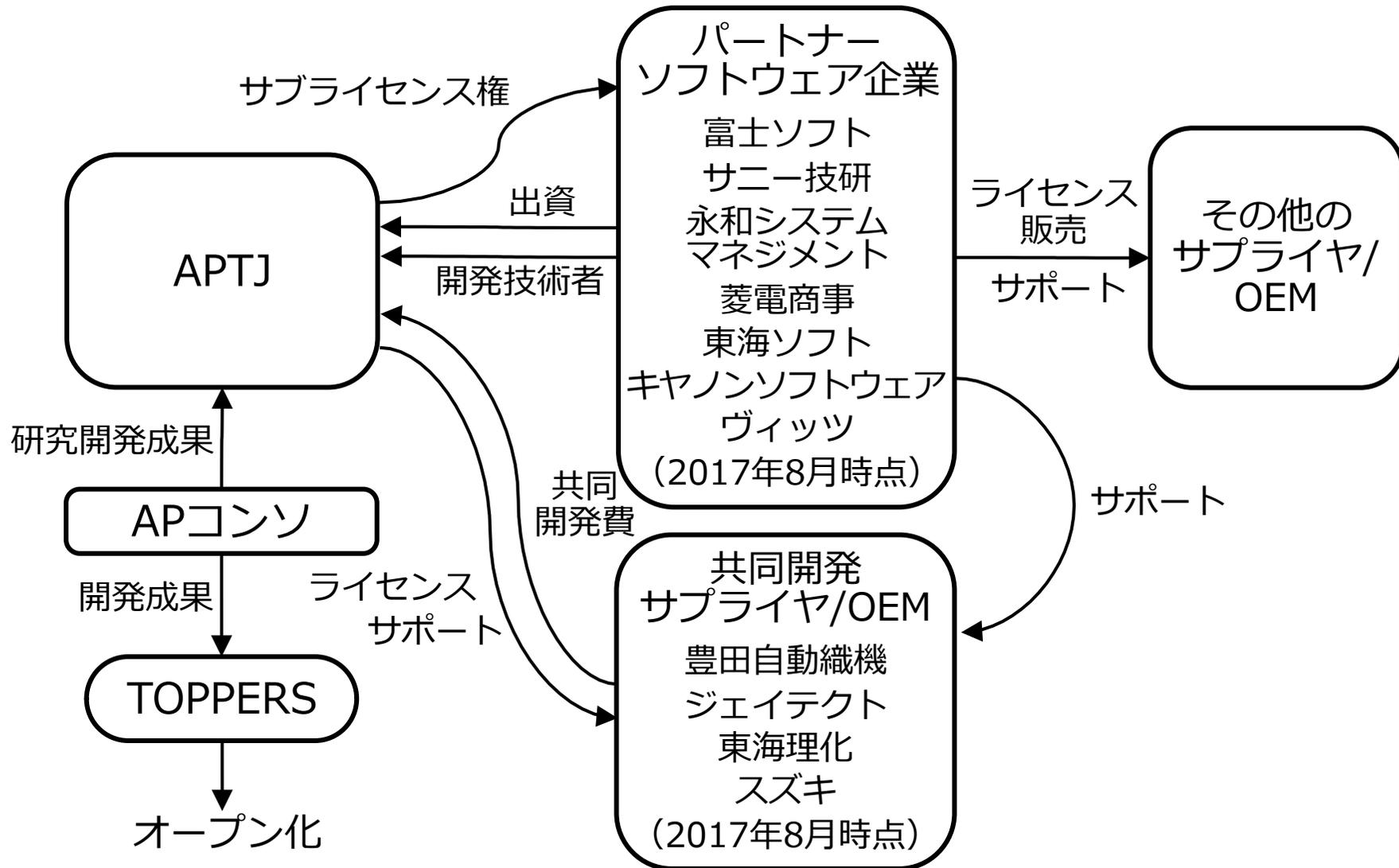
- ▶ AUTOSARが車載制御システム向けのSPF(広い意味でのOS)の国際標準になる流れの中で、車載制御システム向けのSPFが海外製だけになるおそれ
- ▶ 名古屋大学における研究開発だけでは不十分と判断

APTJの活動と位置付け

- ▶ 名古屋大学における産学連携の研究開発成果を活用して、車載制御システム向けのSPFを開発・販売
 - ▶ 技術的な強みは、RTOS, 機能安全, サイバーセキュリティ, マルチコアプロセッサに関する要素技術と知験
- ▶ 自動車メーカー/自動車部品メーカーと共同でSPFを開発
- ▶ パートナーソフトウェア企業との協力
 - ▶ “コンソーシアム型のベンチャ企業”
- ▶ 名古屋大学発ベンチャー企業として設立



APTJを取り巻く組織間の関係(エコシステム)



目次

車載制御システム開発の現状と課題

- ▶ 車載組込みシステムの現状(自動運転登場前)
- ▶ 車載組込みシステム開発の課題, 課題に対するアプローチ
- ▶ 自動運転に向けての組込みシステム技術の変化

自動運転/ADAS技術の動向とプラットフォーム

- ▶ 自動車に起こりつつある変化
- ▶ 自動運転システムの技術構成
- ▶ AUTOSARの新しい動き

TOPPERSプロジェクトにおけるmruby/Rubyへの取り組み

- ▶ TOPPERSプロジェクトのmrubyへの取り組み
- ▶ ツール開発へのRubyの適用

おわりに

- ▶ 車載組込みシステムとmruby

車載制御システム開発の現状と課題

車載組込みシステムの現状（自動運転登場前）

車載制御システムの高度化

- ▶ 自動車に対する新しい付加価値の多くが、電子制御/コンピュータ/ソフトウェアで実現
 - ▶ 省エネルギー，低排気ガス(エンジン制御，…)
 - ▶ 安全性の向上 (ABS, エアバッグ, …)
 - ▶ 利便性の向上, 娯楽性の向上 (カーナビ, …)
 - ▶ コンピュータを活用することで，軽量化やコストダウンが可能になる場合も
 - 例) 車載ネットワークによるワイヤハーネス(自動車内の配線)の軽量化
 - ▶ 統合制御システム/サービス(複数のECUの協調/連携により提供するサービス)が増加
- 車載組込みシステム/ソフトウェアが著しく複雑化**

車載組込みシステム開発の課題

ディペンダビリティの確保/向上

- ▶ システムの大規模化により、設計品質、特にディペンダビリティ(信頼性, 安全性, セキュリティ, ...)の確保が困難に
- ▶ ディペンダビリティに対する説明/証明力の向上が重要に
- ▶ 機能安全規格 (ISO 26262) への対応
- ▶ サイバーセキュリティの確保/強化が大きな課題に

設計生産性の向上(開発の効率化)

- ▶ システムの複雑化や品質要求により、設計生産性が低下

ECUの数の増加, ネットワーク構成の複雑化

- ▶ コストの増大や設置スペースの不足
- ▶ 分散システム/ネットワークの最適設計が困難

その背景にある深刻な問題

組込みシステム技術者 (人財) 不足

課題に対するアプローチ

ディペンダビリティの確保/向上

- ▶ 機能安全規格 (ISO 26262, もうすぐ第2版に) やセキュリティガイドライン (SAE J3061) への対応
- ▶ トップダウンな設計コンセプト (安全コンセプト, セキュリティコンセプト) を持つこと

設計生産性の向上 (開発の効率化)

- ▶ ソフトウェア開発プロセスの地道な改善
- ▶ 設計抽象度を向上させるためのモデルベース開発
- ▶ 設計資産の“良い”再利用を可能にするためのプロダクトライン開発とコンポーネントベース開発
- ▶ アプリケーション開発底上げのためのプラットフォームと、その共通化・標準化 (プラットフォームベース開発)
- ▶ 仮想環境 (シミュレータ) による検証の効率化

ECUの数の増加への対応

- ▶ ECU統合 … 徐々に進行中
 - ▶ プラットフォームの共通化・標準化が前提に
 - ▶ パーティショニング技術が重要に

ネットワーク構成の複雑化への対応

- ▶ CANでは転送レートが不足しており、ネットワーク構造が複雑化する結果に
- ▶ 新しい車載ネットワーク技術の導入を検討すべき時期に
 - ▶ 短期的なソリューションとしてはCAN FDが有力
 - ▶ 車載Ethernetの導入検討が進行中

サイバーセキュリティの強化

- ▶ セキュリティ要求分析手法の確立
- ▶ 車載組込みシステムに向けたセキュリティ強化技術の開発/導入

情報通信技術と組込み技術による自動車の進化

先進運転支援(ADAS)

- ▶ コンピュータにより高度な運転支援を行う
 - ▶ 自動ブレーキ, 前車追従 … 急速に普及期へ

つながるクルマ(connected vehicle)

- ▶ ネットワーク等により(情報的に)外部とつながる自動車
 - ▶ インフラ(道路, クラウド)とつながる
 - ▶ 他車とつながる, 人(ドライバ, 歩行者)とつながる

自動運転(automated/autonomous driving)

- ▶ コンピュータにより運転を行う … 急速に注目を集める
- ▶ 技術的には, ADASの延長線上にある(完全な自動運転になると, 必ずしもADASの発展型ではない要素も)
- ▶ つながるクルマの延長線上にあるという考え方が主流
- ▶ 技術の進歩は著しいが, 完全な自動運転には法的な課題

自動運転に向けての組込みシステム技術の変化

- ! 組込みシステム技術に対する要求は、自動運転システムの構成や実現方式によって大きく異なってくる。よって、今後、要求が大きく変わってくる可能性もある

機能の急速な高度化

- ▶ 手足の電子化／コンピュータ化から、頭脳のコンピュータ化へ
- ▶ ソフトウェアのますますの大規模化・複雑化

外部接続の拡大 ⇒ ダイナミックマップ

- ▶ クラウドとの連携
- ▶ 車車間通信, 路車間通信

ソフトウェアの柔軟な更新

- ▶ ソフトウェアの更新による機能アップ
- ▶ (S)OTA (Software update Over-The-Air)

新しい技術の導入

- ▶ AI, 機械学習, Deep Learning

急拡大する性能要件

- ▶ 新しいプロセッサ技術 (GPUやGPGPU, FPGA, メニーコアプロセッサ) の採用

扱うデータ(種類, 量)の増加

- ▶ 新しい車載ネットワーク技術の必要性 (車載Ethernetが有力)
- ▶ ソフトウェアプラットフォームの高度化 (現状のAUTOSAR仕様では力不足. AUTOSARはAdaptive Platformへ)

安全性の考え方の変化

- ▶ フェールセーフからフェールオペラブルへ

重要性が増すサーバーセキュリティ

- ▶ セキュリティを原因とするはじめてのリコール事例として, クライスラーが140万台のリコールを発表 (2015年7月)

自動運転／ADAS技術の動向と プラットフォーム

自動車に起こりつつある変化

! 相互に関連

電動化

- ▶ 電気自動車
 - ▶ 自動車の構造を大きく変える可能性

自動化, 知能化

- ▶ 自動運転, 無人運転
- ▶ IoT化?

サービス化, シェアリング

- ▶ “mobility as a service”
- ▶ 所有から利用へ
 - ▶ 車の台数が半分になるという予測も

自動運転の社会的価値

安全性の向上

- ▶ 自動運転により事故を減らす

時間を有効活用

- ▶ 自動車で移動中に他の仕事ができる

高齢者／地方の移動手段

- ▶ 公共交通がない地方における移動手段の確保

人手不足の解消

- ▶ 隊列走行や無人配送で運送業における人手不足の解消

その他(いろいろある)

- ▶ 自動駐車による時間の有効活用や運転ストレスの低減
- ▶ カーシェアリングの有効利用(自動運転による回送)
- ▶ 渋滞緩和

自動運転システムの技術構成

認知

- ▶ 周辺認識
 - ▶ 種々のセンサー(カメラ, LIDAR, ミリ波レーダー(長距離, 中距離), 超音波センサー等)の組み合わせ
- ▶ 自己位置推定
 - ▶ GNSS(衛星による測位), ジャイロ, 地図(広い意味でのマップマッチング)などの組み合わせ

判断

- ▶ 行動計画
 - ▶ 目標走行軌跡・速度, 車線変更・合流

制御

- ▶ 車両制御(パワトレ, ステアリング, ブレーキ)
- ▶ HMI(対ドライバ, 対歩行者)

自動運転システムのシステム構成例

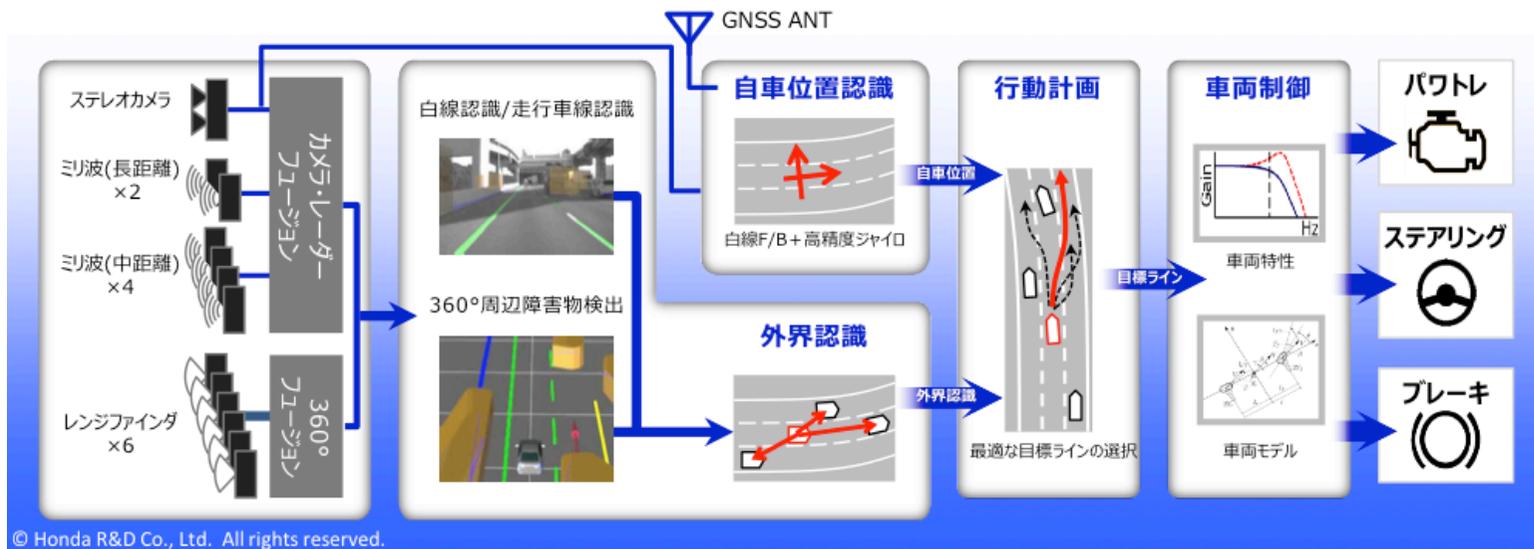
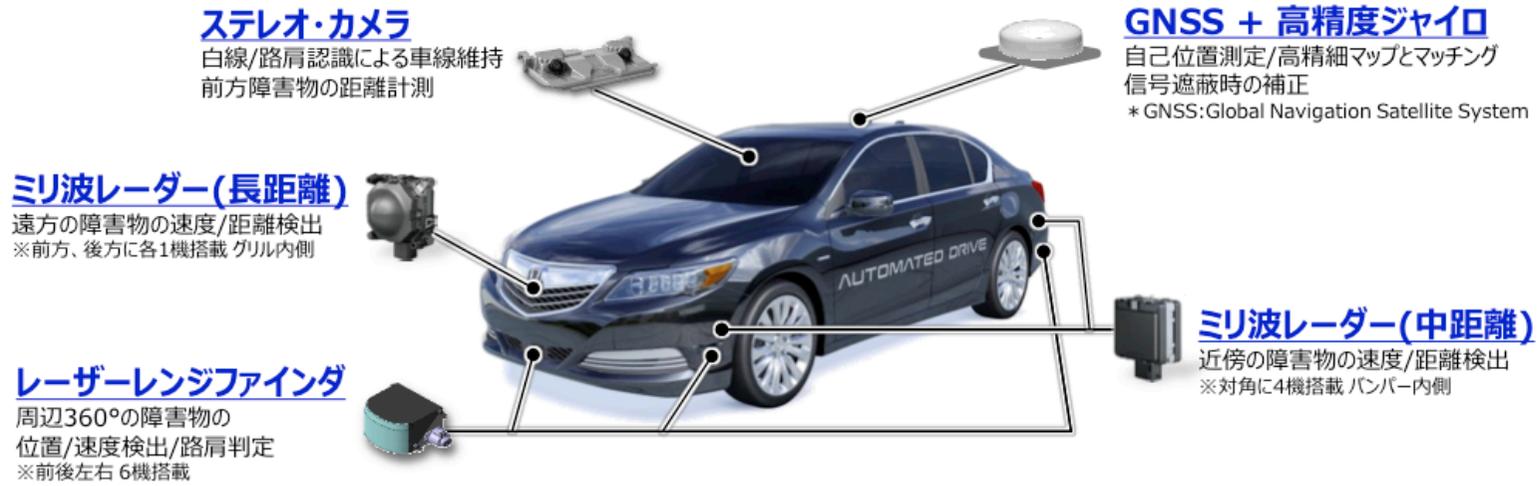
トヨタ自動車の自動車専用道向け実験車のセンサ構成

早期市販化を目指した構成



トヨタ自動車(株) 奥地弘章氏の講演資料より

ホンダの高速道路向け実験車のシステム構成



(株)本田技術研究所 横山利夫氏の講演資料より

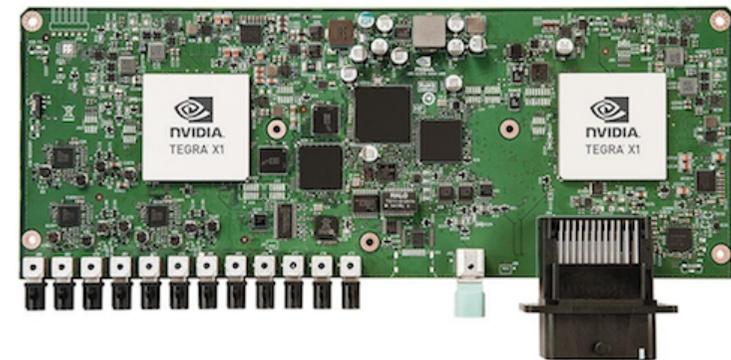
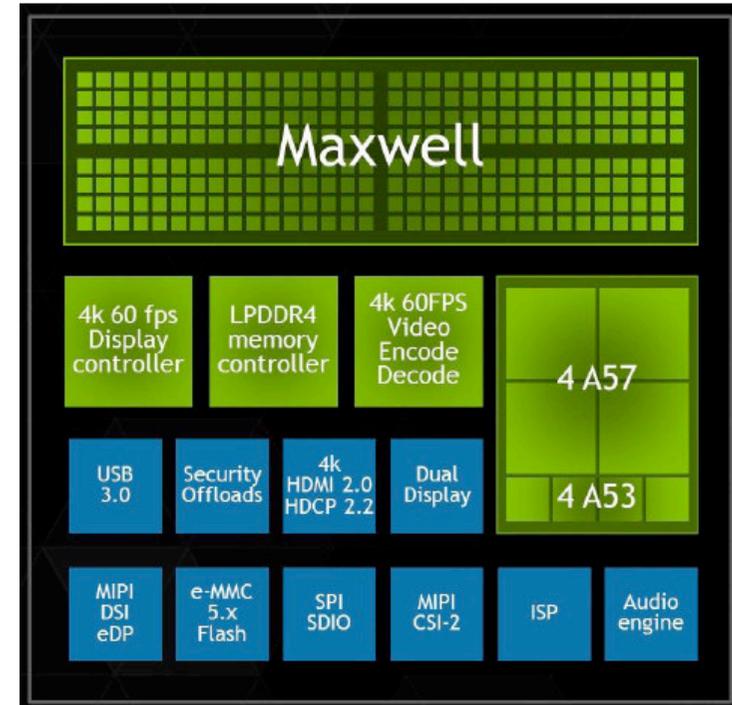
Audi A8のレベル3自動運転システム

- ▶ センサー構成
 - ▶ 超音波センサ × 12 (フロント, サイド, リア)
 - ▶ 360°カメラ × 4 (フロント, リア, ドアミラー)
 - ▶ フロントカメラ × 1 (フロントウィンドウ上端)
 - ▶ ミッドレンジレーダー × 4 (車両四隅)
 - ▶ ロングレンジレーダー × 1 (フロント)
 - ▶ レーザスキャナー × 1 (フロント)
 - ▶ 赤外線カメラ (ナイトビジョン) × 1 (フロント)
- ▶ 使用しているプロセッサ/SoC
 - ▶ NVIDIA Tegra K1 (GPU)
 - ▶ ALTERA Cyclon V (FPGA)
 - ▶ Infineon Aurix (車載マイコン)
 - ▶ Mobileye EyeQ3 (画像処理プロセッサ)

ハードウェアプラットフォームの例

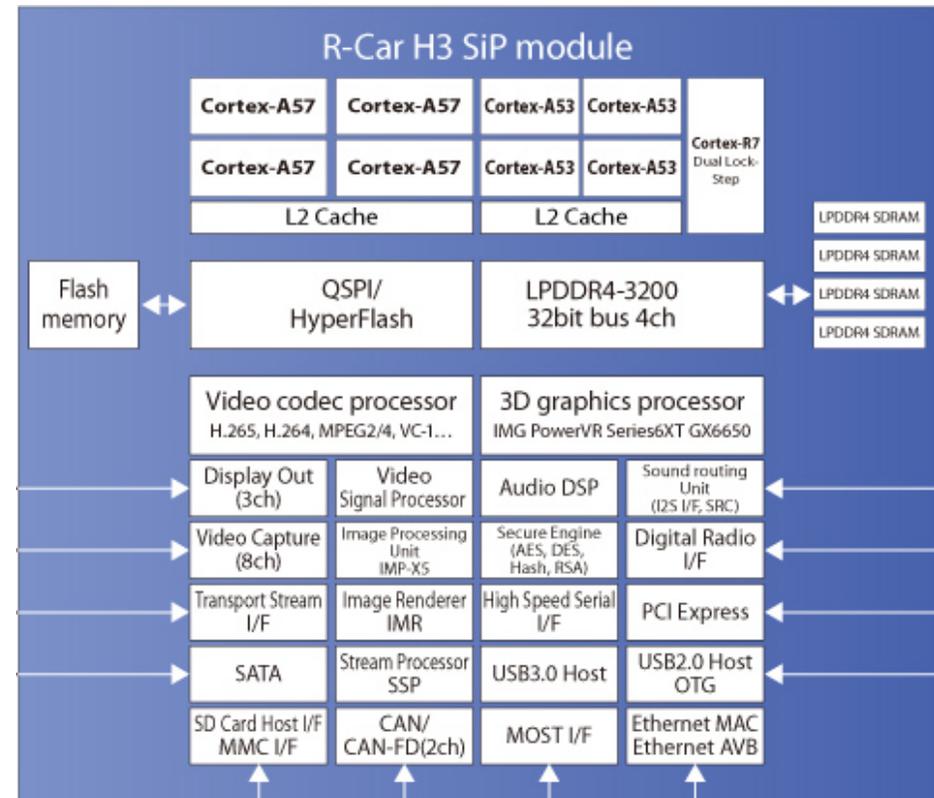
NVIDIA Tegra X1

- ▶ NVIDIAのモバイル向けGPU
 - ▶ 車載もターゲットに (NVIDIA DRIVE PXに2個搭載)
- ▶ Maxwell GPUアーキテクチャ
 - ▶ 256 CUDA Cores
 - ▶ ピーク性能: 512GFLOPS (半精度だとその2倍)
- ▶ ARM Cortex A57×4 + A53×4
- ▶ 最大メモリ帯域: 25.6GB/s



ルネサス R-Car H3

- ▶ ルネサスの自動運转向けSoC
 - ▶ 「自動運転レベル2やレベル3を超えることのできるSoC」
- ▶ ARM Cortex A57×4
+ A53×4
- ▶ ARM Cortex R7 (デュアルコアロックステップ)
- ▶ GPU: PowerVR GX6650
- ▶ 画像認識ハードウェア: IMP-X5
- ▶ 最大メモリ帯域: 50GB/s
- ▶ ASIL B対応



車載ネットワーク／プロトコルに対する要求

従来からの要求

- ▶ 信頼性, リアルタイム性
- ▶ 低コスト, 軽量, 省スペース
- ▶ EMC (EMI, EMS)

大量のデータ転送

- ▶ 広帯域のネットワーク
 - ▶ 100Mbps～1Gbps (システム構成に大きく依存)
 - ▶ LIDAR 1つで数Mbps～数十Mbps (無圧縮)
 - ▶ カメラ画像を無圧縮で流すとさらに帯域が必要
- ▶ オーバヘッドの小さいプロトコル処理

時間同期

- ▶ センサー間の同期
 - ▶ どの程度の同期精度が必要？

サービス指向通信？

- ▶ サービス指向アーキテクチャ(SOA)の考え方を車載システムに導入するための通信機構
 - ▶ メカニズムは, client/serverモデルと差がない
- ▶ それに対応する従来の車載システムの通信パラダイム
 - ▶ sender/receiverモデル(AUTOSARの用語)
 - ▶ publisher/subscriberモデル(動的に接続する意味合い)
- ▶ サーバーが動いているかを知りたいという要求

複雑なデータ構造

- ▶ 自車周辺の障害物情報や地図情報など, 複雑なデータ構造を扱うことが必要に

セキュリティの確保

- ▶ まずは, メッセージ認証コード(MAC)の導入から

ソフトウェアプラットフォーム

当面は戦国時代？

- ▶ まだ要求が明確とは言えない
 - ▶ 自動走行のシステムアーキテクチャやアルゴリズムがまだまだ競争領域で、変化する余地が大きいため
- ▶ これからの競争で標準が決まっていく
 - ▶ 第1世代の製品は、標準を気にしている余裕がない？

現時点の有力候補

- ▶ ROS (Robot Operating System)
 - ▶ 研究開発用としては最有力. 製品への適用は？
 - ▶ ROS2.0 (品質を確保) の開発も進んでいる
- ▶ AUTOSAR Adaptive Platform
- ▶ Googleは？ (Android Automotive ?)

AUTOSARの新しい動き

AUTOSAR (AUTomotive Open System Architecture) とは？

- ▶ 自動車, 自動車部品, エレクトロニクス, 半導体, ソフトウェア企業によるグローバルパートナーシップ (2003年に設立)
 - ▶ ソフトウェアの複雑性を軽減するために, ソフトウェア基盤 (infrastructure) の業界標準を作成
- ▶ コアパートナー (2017年時点)

BMW	Daimler	PSA Peugeot Citroen
Bosch	Ford	トヨタ自動車
Continental	GM	Volkswagen
- ▶ 最新の仕様書 (群) は Release 4.3.0
 - ▶ Release 4.2.2の時点で, 約100のソフトウェア仕様書, 約110の関連ドキュメント等で, 全体で19600ページ
- ▶ “Cooperate on standards, compete on implementation” (標準化で協調し, 実装で競争する) をポリシーに

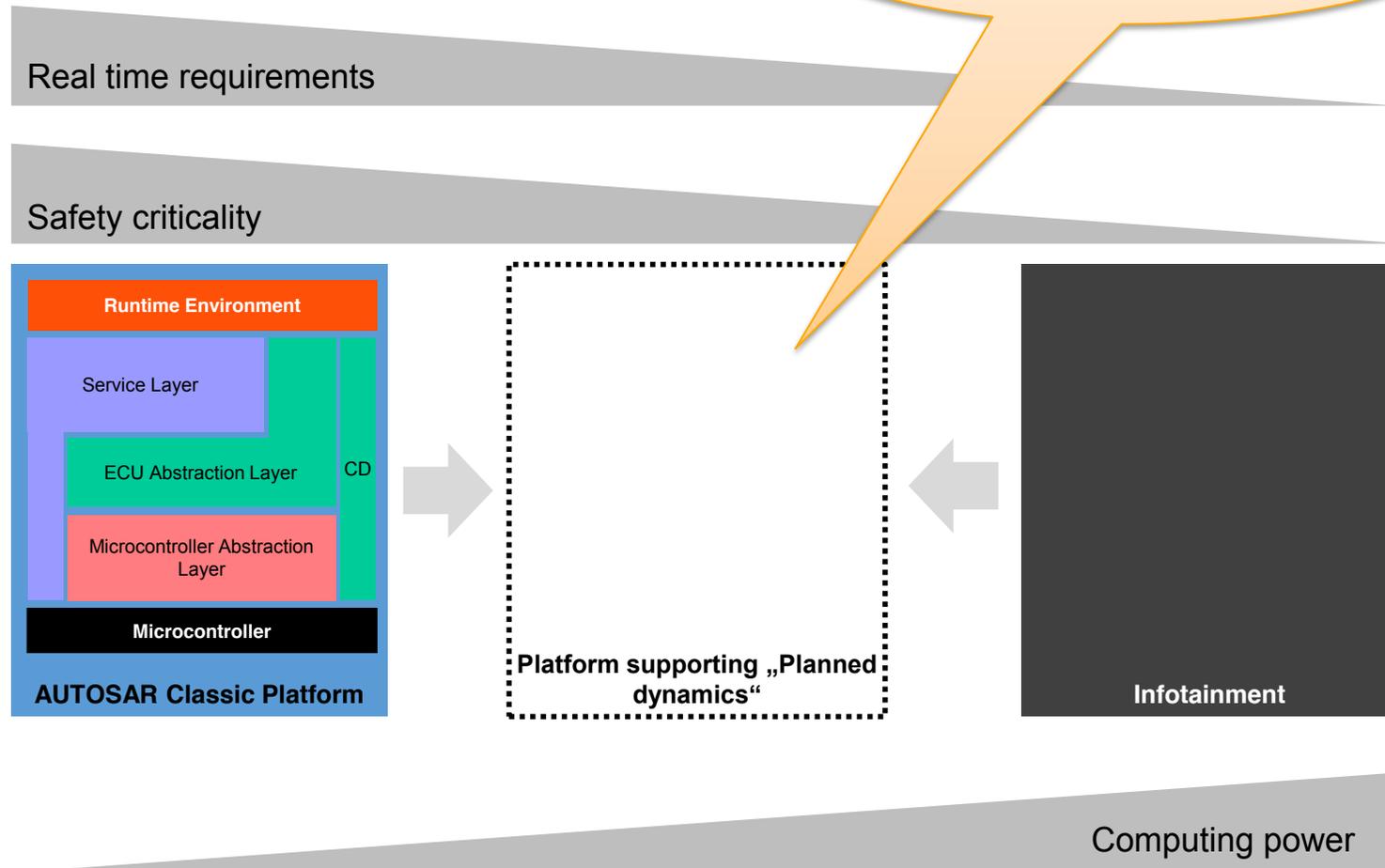
AUTOSAR Adaptive Platform (AP) の策定へ

- ▶ ADAS・自動運転の実現に向けて、ソフトウェアプラットフォームにも新しい要求
 - ▶ 具体的に何が違うのか、まだそれほど明確ではない
- ▶ ADAS・自動運転を狙った新しい仕様として、AUTOSAR Adaptive Platformの策定を開始
 - ▶ 標準化を加速するため、仕様策定とリファレンス実装の開発を並行に進める (AUTOSARの活動モデルの変更)
- ▶ 従来の仕様は、AUTOSAR Classic Platform (CP) と改名
 - ▶ Classicと言っても「古い」という意味ではなく、役割の違いであり、今後も不要になるわけではない
 - ▶ Classic Platform (CP) と Adaptive Platform (AP) の共通事項は、AUTOSAR Foundation (AF) として切り出す

AUTOSAR APの位置付け

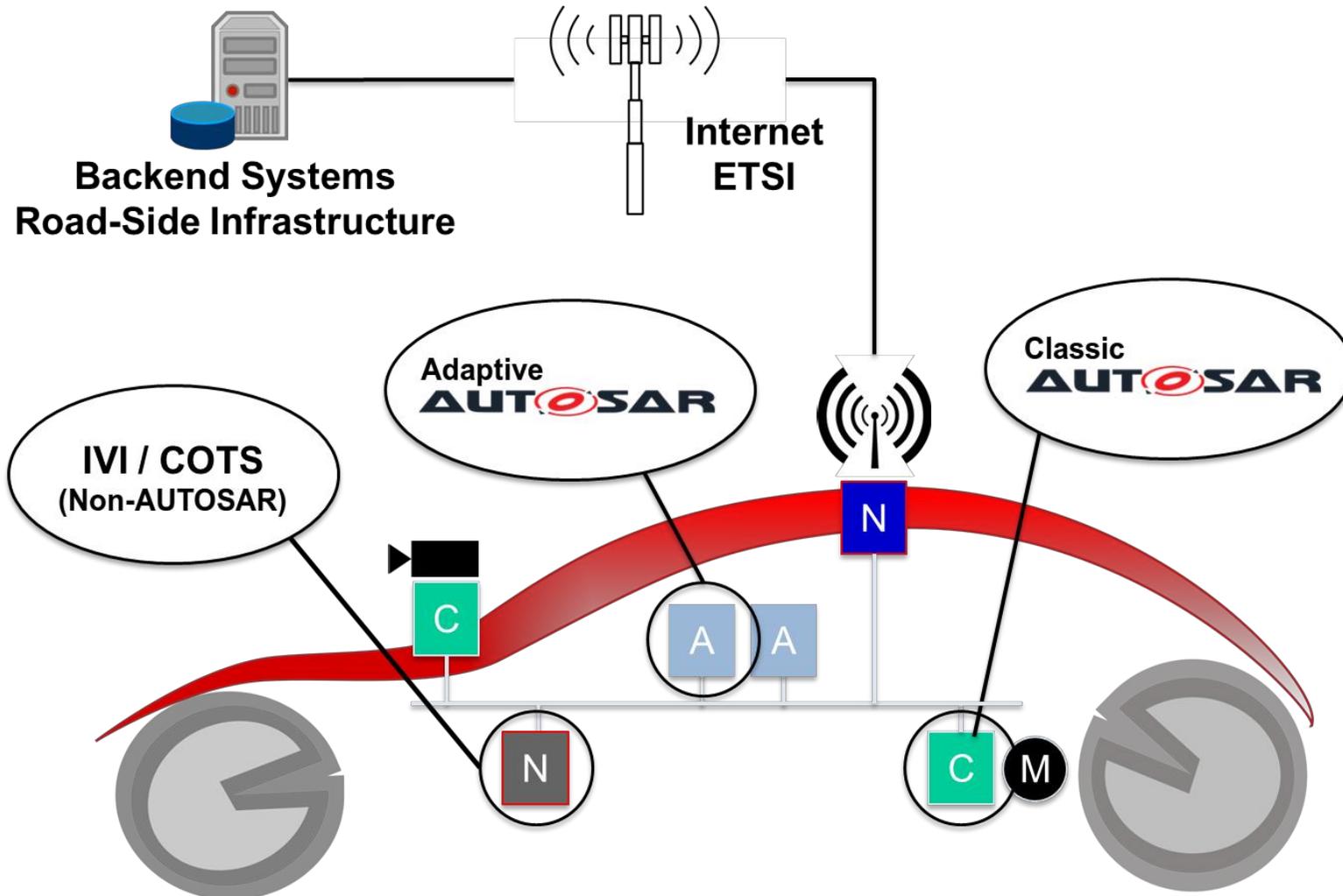
- ▶ CPとの関係 (早い時期の資料)

AUTOSAR CPとIVI向けプラットフォームの間の領域



M. Bechter: “AUTOSAR Adaptive Platform”, 8th AUTOSAR Open Conference 2015より

他のプラットフォームとの使い分け(共存)



AUTOSAR: “Explanations of Adaptive Platform Design”, R17-03より

AUTOSAR CPとAPの技術的な違い

- ▶ AUTOSAR APは、ハードリアルタイムシステム向けのSPFではないと思われる

AUTOSAR Classic Platform	AUTOSAR Adaptive Platform
OSEKベース	POSIX (PSE51) ベース
ROMからコードを直接実行	アプリケーションは不揮発メモリからRAMにロードされる
すべてのアプリケーションが1つのアドレス空間を共有 (安全性のためにMPUを使用)	各アプリケーションが自分の(仮想)アドレス空間を持つ (MMUを使用)
シグナルベースの通信 (CAN, FlexRay) に最適化	サービス指向通信
タスク構成が固定	複数の(動的な)スケジューリング戦略をサポート

M. Bechter: “AUTOSAR Adaptive Platform – Key concepts and development process,” 9th AUTOSAR Open Conference (2016)を独自に日本語訳

AUTOSAR APが活用する技術

- ! APの標準化では、これまで車載システムで活用できていなかったものも含めて、以下のような技術や標準を活用
- ▶ C++(具体的には, C++11/14)
- ▶ サービス指向アーキテクチャ(SOA)
 - ▶ システムをサービスの集合で構成するという考え方
 - ▶ 分散コンピューティングに力点
- ▶ 並列処理
- ▶ セーフティとセキュリティ
- ▶ 動的な振る舞いの計画(Planned dynamics)
 - ▶ ソフトウェアの開発とインテグレーション時には、リソースと通信の動的管理により開発を容易にし、製品提供の時点では、動的な振る舞いを制限する
- ▶ アジャイル開発

最新のC++の活用

- ▶ C++14を活用するためのガイドラインを策定
 - ▶ Guidelines for the use of the C++14 language in critical and safety-related systems
 - ▶ 安全システム向けの既存のC++コーディングガイドライン (代表例:MISRA C++:2008) がC++14に対応していないため, 新たに作成した
 - ▶ このガイドラインでは, 既存の安全システム向けのガイドラインが使用を禁止している以下の機能の利用を許している
 - ▶ 動的メモリ管理, 例外, テンプレート
 - ▶ 継承, 仮想関数
- ⇒ APではこれらを活用

TOPPERSプロジェクトにおける mruby/Rubyへの取り組み

TOPPERSプロジェクトのmrubyへの取り組み

軽量Rubyフォーラムとの協力の合意

- ▶ TOPPERSプロジェクト TECS WGと軽量Rubyフォーラム ライブラリWGで、以下の趣旨とする協業を行うことを合意 (2013年)
 - ▶ 技術を広める活動で相互に協力
 - ▶ 両技術を組み合わせて活用する

TOPPERS OS上へのmrubyのポーティング

- ▶ mrubyをTOPPERS/ASPカーネル上にポーティングし、mrubyのリアルタイム制御への適用性を評価 (2012年)
 - ▶ ポーティング自身は容易
 - ▶ マルチタスクや複数のmrubyを動作させることは (この時点では) サポートしていない

参考)TECSとは？

“*TOPPERS Embedded Component System*”

- ▶ 各種のソフトウェアモジュールを部品化し、必要な部品を組み合わせることによって大規模な組込みソフトウェアを効率的に構築するための技術
- ▶ マルチコアプロセッサシステム/分散システムにおける遠隔呼出し(RPC; Remote Procedure Call)にも対応
- ▶ TECSの特徴
 - ▶ コンポーネントの実装コードはC言語で記述(オブジェクト指向でないコンポーネント技術)
 - ▶ コンポーネントを静的に結合する(最適化を行う)
 - ▶ システム内のすべてのソフトウェアをコンポーネントとして扱う
 - ▶ 遠隔呼出しのコンポーネントをツールにより生成

TECS mruby Bridge Plugin

- ▶ mrubyからTECSコンポーネントを呼び出すプログラムを自動生成するプラグイン
 - ▶ RTOSやTECSで記述したデバイスドライバのAPIを、mrubyから簡単に呼び出すことが可能
 - ▶ リアルタイム性が必要な部分はTECS (C言語) で記述し、mrubyと役割分担



- ▶ 2013年に最初のバージョンを一般公開, その後も改良

mruby VMのTECSコンポーネント化(マルチVM対応)

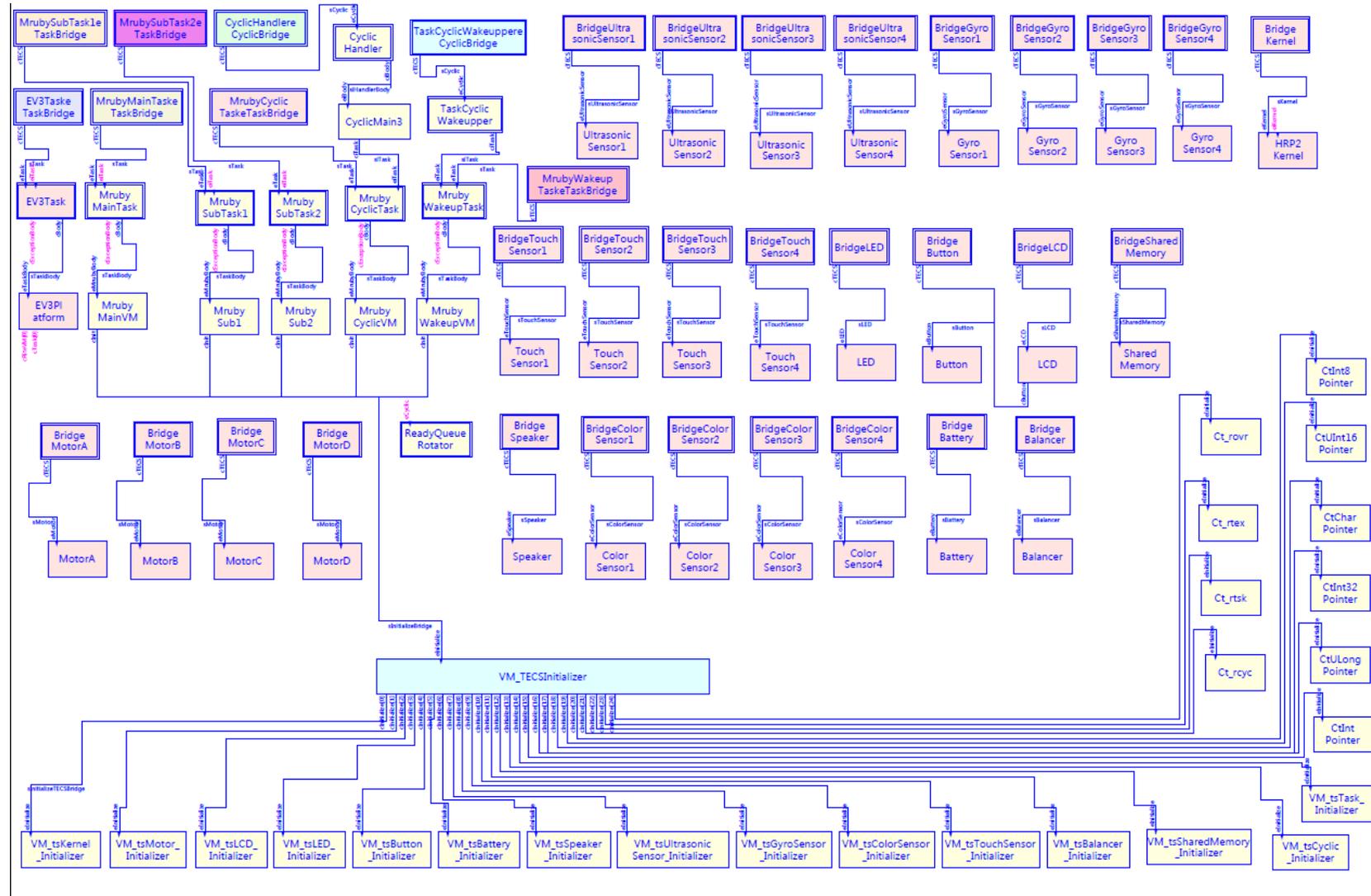
- ▶ 複数のmruby VMを並行動作させることが可能に
- ▶ 異なるmruby VM間で、RTOSのAPIを用いて同期・通信させることが可能
- ▶ mruby VM毎のヒープを分離
 - ▶ ヒープの排他制御が不要に
 - ▶ 1つのVMがヒープを使い切ってもGCが動作しても、他のVMの動作には影響がない
 - ▶ TLSF (リアルタイムシステムに向けたメモリアロケータ)を用いて実装
- ▶ 2013年に最初のバージョンを一般公開, その後も改良

mruby on EV3RT+TECS

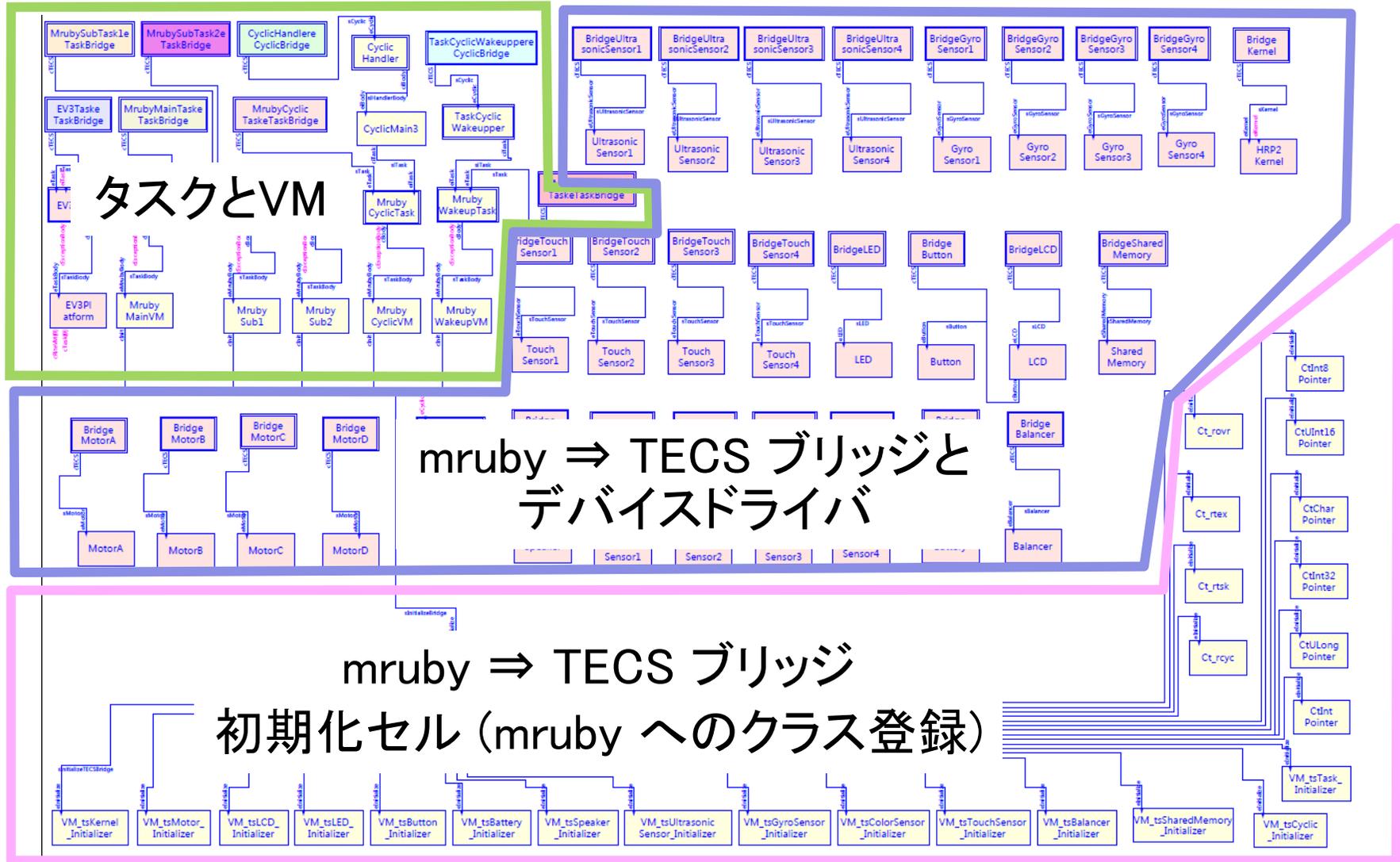
- ▶ EV3RT上に、mruby (軽量Ruby) を載せたSPF
 - ▶ mrubyでEV3の制御が記述できる
- ▶ プラットフォームの構築にTECSを利用
- ▶ 2015年にα版, 2016年にβ版を一般公開, 2017年に改良版を一般公開
- ▶ ETロボコン2016から公式プラットフォームの1つに



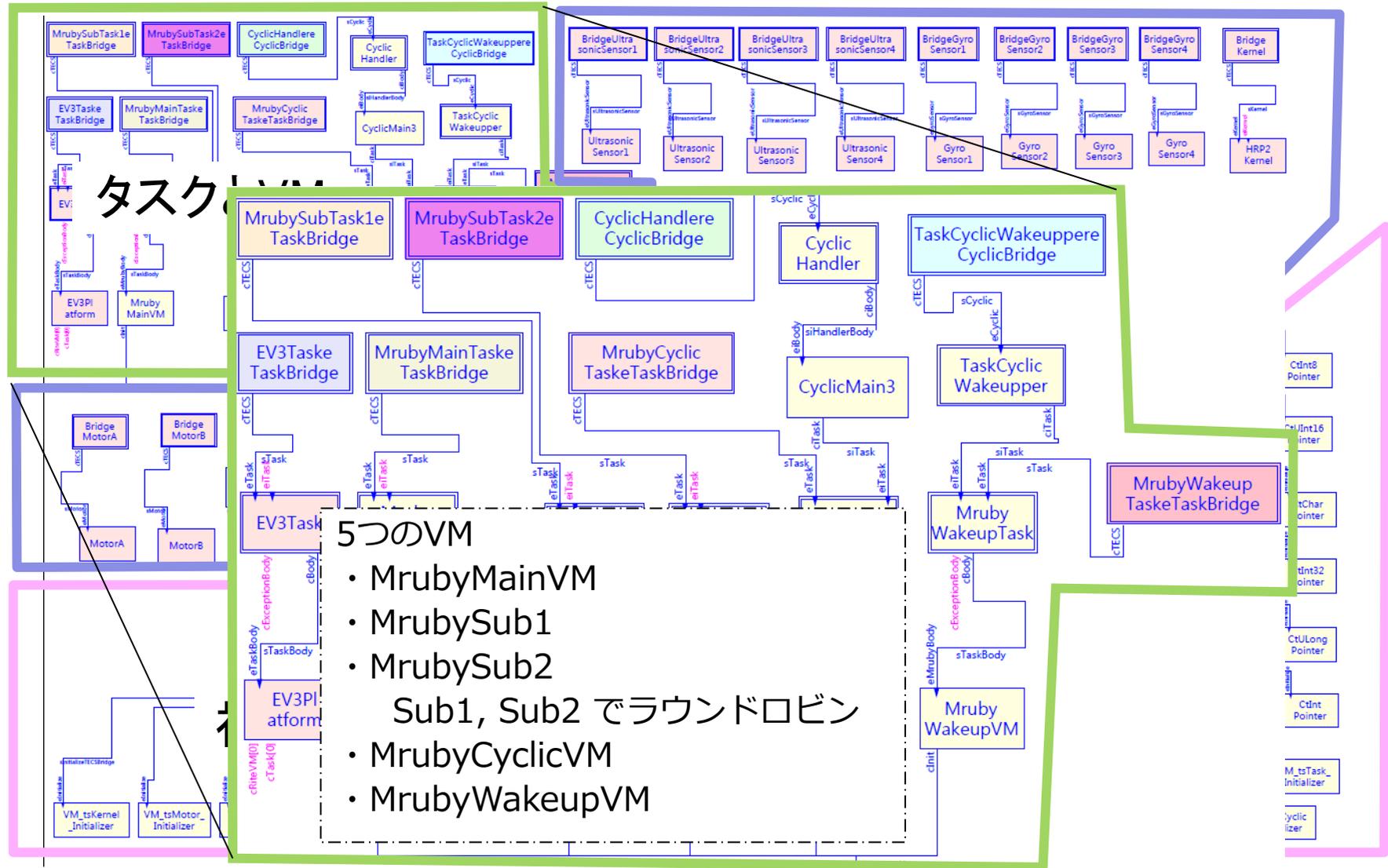
mruby on EV3RT+TECSのコンポーネント構成



mruby on EV3RT+TECSのコンポーネント構成



mruby on EV3RT+TECSのコンポーネント構成



製品化の事例

教育版EV3 mrubyプログラミングセット((株)アフレル)

- ▶ LEGO Mindstorms EV3を使って
mrubyのプログラミングを学ぶ教材
- ▶ プラットフォームとして、
mruby on EV3RT+TECS
を利用
- ▶ 2016年6月29日発売



ツール開発へのRubyの適用

TOPPERSプロジェクトにおけるツール開発へのRubyの適用

- ▶ TECSジェネレータ
 - ▶ 関連ツールやプラグインを含めると、約4万行の規模(コメント等を含む. 以下同じ)
 - ▶ Racc (Rubyのパースジェネレータ)を利用
- ▶ TOPPERS第3世代カーネル向けコンフィギュレータ
 - ▶ コンフィギュレータ本体は2000行程度
 - ▶ ASP3カーネルの生成スクリプトは2000行程度
 - ▶ HRP3カーネルの生成スクリプトは7000行程度
- ▶ TOPPERS第3世代カーネル向けのユーティリティ群

APTJにおけるツール開発へのRubyの適用

- ▶ AUTOSAR OS等のBSWモジュールのジェネレータ (RTEジェネレータは除く)

おわりに

車載組込みシステムとmruby

(やはり)課題は品質

- ▶ 車載制御システム分野は、安全性に関わるものが多いため、特に品質要求が厳しい
- ▶ とは言え、AUTOSAR APのように、使える技術を広げようとする試みがされている

狙うべき領域は？

- ▶ まずは、安全性に関わらない領域
 - ▶ 典型例:IVI(車載インフォテインメント)… 考えやすい
- ▶ mruby(組込みシステム向けスクリプト言語)の特性が活きるのは、ソフトウェアを頻繁に修正したいシステム
- ▶ 例えば,
 - ▶ 車両の個人化(パーソナライズ),おもてなし機能
 - ▶ 自動運転のシナリオ作成