

# SW defined Vehicle (SDV) : 自動車向け SW 開発の新潮流と AI 技術の適用

公立はこだて未来大 IT アーキテクチャー特論

宗像尚郎

ルネサスエレクトロニクス株式会社  
ハイパフォーマンスコンピューティングプロダクトグループ  
シニアダイレクタ

2026-1-21

## 自己紹介 (Who am I)

日本の半導体メーカーで **自動車向けの SW ソリューション開発** に従事しています

### ■ ルネサスエレクトロニクス株式会社

- 日立、三菱、NEC の半導体事業が統合、**グローバルな半導体専門メーカーになった**
- 「**自動車向け**」大規模ソフトウェア基盤 (OS、プラットフォーム) **開発** に従事
- **社内のオープンソース開発活動** (Linux kernel 開発など) を管掌

### ■ オープンソース (=OSS) 開発プロジェクトへの貢献 (会社公認の社外活動)

- The Linux Foundation 元理事 (ex-Board of Director)
- **AGL (Automotive Grade Linux)** プロジェクト、**yocto** プロジェクト理事
- **COVESA (Connected Vehicle System Alliance)** 理事
- 社外講師、講演、インターフェース誌などへの投稿など多数

**自動車業界でも SW 開発力の強化、特に OSS の利活用が重要になっています**

# 第一部：自動車開発の最新動向

## 自動車は Software Defined を目指す

## SDV (= Software Defined Vehicle) とは何か

スマートフォンのように「SW の更新によってクルマの機能をアップデートさせる」

- 現在は 機能毎に独立した 100 個以上の制御ユニット（ECU）に分割されている
- 各 ECU の SW は自己完結 していて、ユーザーによる SW 更新手段は提供されない
- 新機能の追加 は専用 ECU の更新や追加が必要にあるので 現実的でない
- ユーザーはスマホのように SW 更新で機能が追加できることを期待 している
  - 年数が経過したクルマでも 最新の機能やサービス を利用したい
  - 購入後にも追加機能を有効化 したい（運転支援、遠隔監視等のサブスク化）
- ネットワーク経由で ECU の SW を更新 させる仕組み（OTA）が基盤になる
- SDV 実現には、分散している ECU や SW を集約し連携させる 必要がある

多くの自動車メーカーが今「ソフトウェア ファースト」をスローガンに掲げている

## 「ソフトウェアファースト」にすれば全てが解決できる訳ではない

自動車は **組み込みシステム** であり、**HW 制約** から逃れられない

- **組み込み機器開発では常にプログラムを実行するターゲット** が先に決まっている
  - プログラマーの仕事は **ハードウェア性能を最大限に引き出す制御手順** を考えること
  - 時には **ハードウェアの制約（不具合）回避** もソフトウェアの役割とされた
  - 開発予算配分ではソフトウェア開発費が削減対象コストになることも多い
- **商品の差異化を生み出す要因として SW（と、その更新）** が注目されている
- 「ソフトウェアファースト」になっても、**以下のようなことは起こりません**
  - **どんなハードウェアでも動く（= 移植可能な）ソフトウェア** が開発できる
  - 機器の優位性はソフトウェアによって生み出されるので **ハードは何でもよくなる**
  - 特殊なコーディングが必要な **HW アクセラレータ**ではなく**全て CPU 処理** にしたい
  - **ハードウェア開発よりもソフトウェア開発費用に多くの費用を使うのは当然**

**「SW と HW の疎結合（デカップリング）」を実現することが本来目指すべき方向性**

## 構造がシンプルな 電動車（EV）なら SDV を実現しやすい理屈だが...

日本の報道だけを見ていると 電動車（EV）ブームは失速した印象もあるが、実際には

- 2025 年の全世界自動車販売台数 = 約 8,430 万台
- そのうち EV(=BEV+PHEV) は 2,050 万台 (EV 比率 24%)
  - 中国 1,160 万台 (前年比 +19%)、
  - 欧州 380 万台 (+33%)
  - 北米 170 万台 (-1%)、
  - その他 150 万台 (+48%)
- SDV で成功しているのは 過去 SW 資産を持たない Tesla や中国の新興 EV メーカー
- 既存 SW 資産 のある会社は SDV 向けの SW 開発が全くうまくいっていない
  - VW (CARIAD)、Daimler (MBition)、Toyota (Woven)、Ford (FNV4) など
  - 自動車で確立されてきた「SW 開発手法」や「ECU アーキテクチャ」が全否定された

これまでエンジン車で実績のある伝統のある自動車メーカーはほぼ全て SDV で大苦戦

## 伝統的な自動車メーカーの巻き返し も始まっている（BMW ix3 の例）



# 「ゾーン型アーキテクチャ」が少ない ECU、短いハーネスを実現する

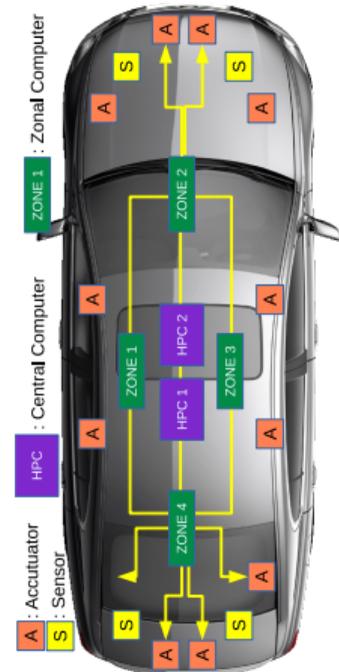
## 100 個もあった ECU を 3 個（中央）+ 8 個（手足）程度に集約

### ■ 物理面 = コストダウン に大きく寄与する

- ECU 間を接続する **ワイヤーハーネス** が大幅に短くなる
- ハーネスの **総延長は 2~3km**、重量は **20~40kg** 強もあり  
車両重量の 2~3% → **短くなれば 燃費/電費が大幅改善**
- 生産ラインでの **ハーネスの取り回しが簡単** → **組み立て工数減**
- **CAN**（低速、単純）→ **Ethernet**（高速、ゾーン分けに対応）
- **部品点数減** → **生産コスト削減**、**故障率** や **保障コスト** も低減

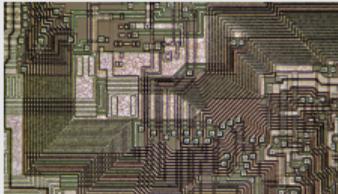
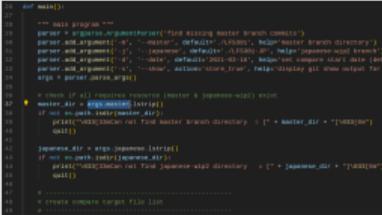
### ■ ソフトウェア面 = 機能アップデートの自由度 が拡大する

- 1 つの ECU が **SW の更新** を受け取って各 ECU に **配信** する
- 利用時間や接続情報の **高精度ロギング** が可能になる
- **仮想化技術** を利用した **SW 的なパーティション分割** を採用



## 自動車制御用 SoC の最先端（仕様、事業）

# 「自動車の電子制御のノウハウ」はどこに集約されている？

	2010 年以前	最近まで	今後
主な 差異化要因	PCB基盤上の回路	SoC	SW
差異化技術の 帰属	最終製品開発メーカー	半導体ベンダー	SW プラットフォーム
主な 技術要件	堅牢性	機能の集積	移植性（再利用性）
イメージ			
解決すべき 課題	機器間ネットワーク 経年劣化対策	処理性能 消費電力（発熱対策）	SW 更新 サイバーセキュリティ

テクノロジー進化に伴いノウハウの形や場所は変化しており、今は SW に集約されている

## ルネサスは新世代 R-Car X5H 開発ボードを一般公開（2025.12.22）



<https://www.renesas.com/en/about/newsroom/renesas-fast-tracks-sdv-innovation-r-car-gen-5-soc-based-end-end-multi-dom>

# R-Car X5H は Intel の Core-i9 14900K よりもはるかに高性能

R-Car X5H		Intel Core i9-14900K (Raptor Lake Refresh)
Arm CA9 x 32 Arm CR52 x 6 (3x2)	CPU コア数	24
1,000k (CA) + 60k (CR)	CPU 性能 (DMIPS)	450k ~ 900k *1 ( PassMark 45,000 )
2,000 (8k x 10 画面对応)	GPU 性能 (GFLOPS)	793
400	NPU 性能 (TOPS)	内蔵していない
2.7GHz	定格クロック周波数	3.2 GHz
3 nm	設計プロセス (線幅)	10 nm
2,916	ピン数	1,700

\*1: 一般的な近似として 1 PassMark ≈ 10~20 DMIPS を用いて性能比較を行っています。

# 半導体メーカーの競争軸は「単体性能」から「ソリューション提案力」へ

## ❁ ウィニング・コンビネーション

アナログ+パワー+組み込みシステム+コネクティビティなど、ルネサスの横断的な製品ポートフォリオを組み合わせたトータルソリューションの数々がご覧になれます。



ワイヤレススマートロック



ユニバーサルNFC充電器



USB IO-Linkマスタ



自動車用ウィンドウコントロール



サウンドビュー& AR-  
HUD用ビデオ出力拡張ソ  
リューション



車載コックピット ハプティ  
クスソリューション



接続型Androidクラスタ



AHLを備えたフルデジタル  
クラスタソリューション

<https://www.renesas.com/jp/ja>

半導体メーカーが SW 開発（ドライバー、ミドル、サンプルアプリ）に注力している

## 半導体メーカーの「HW 開発」と「SW 開発」の最新トレンド

### ハードウェア開発 → 大規模化、超複雑化

- 微細化により **論理規模が巨大化**
  - **検証工数** も肥大化 → 効率化が課題
  - **発熱対策** が大きな課題
  - **チップレット** など実装面での革新
- **最先端プロセスの開発・製造コスト**
  - 適用可能な製品群は限定される
- **性能競争**
  - **CPU 能力**（並列化、高速化）
  - **GPU や NPU (=AI) エンジン** 性能
- **RISC-V** などオープン規格の台頭

### ソフトウェア開発 → エンドレスゲーム化

- **SW の規模が爆発的に拡大中**
  - **メカ制御** → **SW 制御** の適用拡大
  - **ネットワーク**（**クラウド連携** など）
  - **AI による推論機能** の実装
- **組み込み機器も SW 更新** が必要
  - **機能追加** で製品の魅力を維持
  - **脆弱性対策** パッチの適用
  - **HW 収束後も SW 開発は継続必至**
- **OSS の適用範囲** が拡大（Linux など）
  - **開発 コミュニティとの連携** も必要に

**製品出荷後の SW 更新（機能強化、セキュリティ対策）費用の回収スキームが重要**

# SoC ビジネスの特性（線径微細化に伴って参入障壁が拡大＝寡占化）

## SoC ビジネスには 数年の開発期間 と 莫大な開発費用 が必要

### ■ 設計開発期間

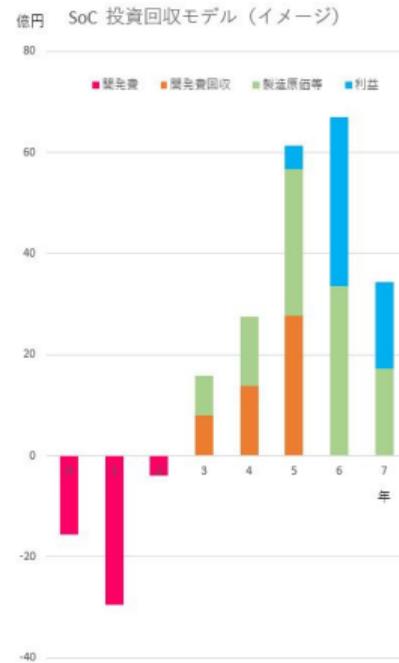
- 回路規模の拡大により **設計検証期間は長期化** の傾向
- 動作周波数高速化により **熱設計（パッケージ）** が課題
- 総開発費に占める **SW の比率が拡大**（現状、約4割）

### ■ 製造期間

- マスク数増大により **ウエハー製造には一か月以上** かかる
- 必要部材が多く、**サプライチェーン問題** が顕在化した
- 微細化により **受託製造メーカ（ファウンダリー）** が台頭

### ■ 投資回収

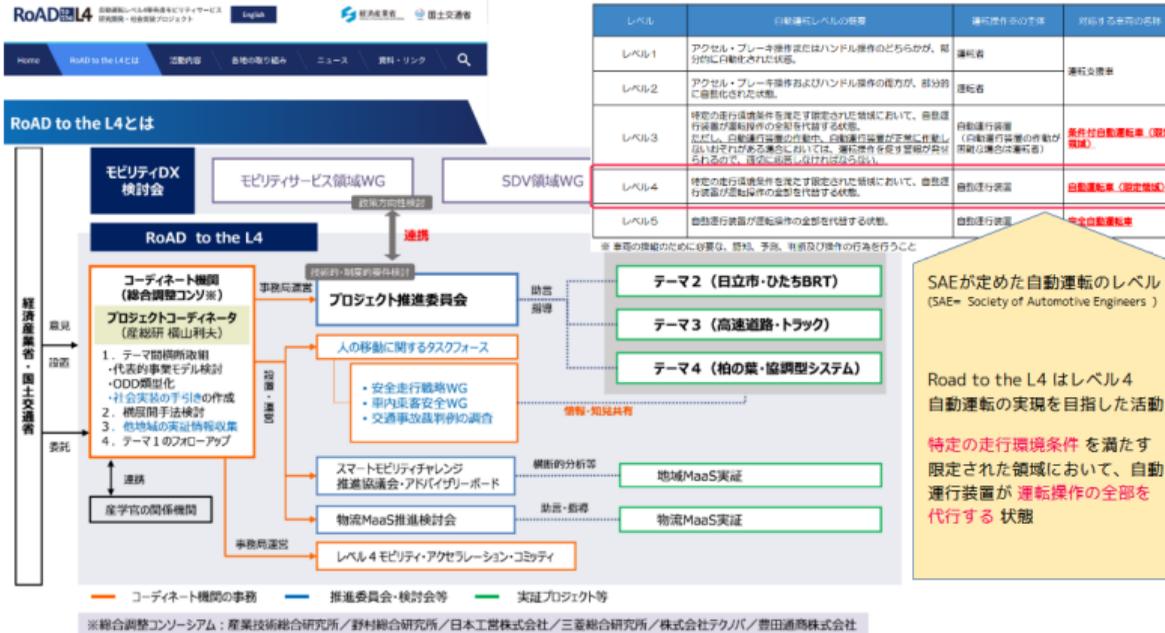
- SoC 開発には **数十億円規模の初期開発投資が必要**
- 量産開始までの期間の投資を支える **資金調達力** が必要
- **初期投資分を回収し利益を出す** には **規模** が必要



## 第二部：自動運転／運転支援の現在位置

## 日本の L4自動運転 実現への取り組み（行政面）

# RoAD to the L4 プロジェクト（経済産業省、国土交通省 共同）



<https://www.road-to-the-l4.go.jp/about/>

# RoADL4 の実証実験：まだまだ L4 実現への道のりは遠いか？

## RoADL4 が目指しているもの

- レベル 4 自動運転 の 社会実装 (特に地方・過疎地域) を実現するための 基盤整備 を検討している
- L4 では「人間が全く介在しない前提」で考える必要がる
- そのため 説明可能性・検証可能性・冗長性 の確保が必須である
- 特に日本では 事故時の説明責任 や 裁判体制 などの 法規・社会受容性 もケアする必要がある

## 5. 「RoAD to the L4」の実証実験地域

### 福井県永平寺町

- ・鉄道廃線跡地の自転車歩行者専用道路を自動運転車両の走路として通行許可承認取得
- ・木々の深い山間の走路のため、電磁誘導線を用いた小型電動カートを活用
- ・国内初の道路運送車両法に基づくレベル4の自動運行装置としての認可（2023年3月30日）、道路交通法に基づく特定自動運行の許可（2023年5月11日）を受け、無人自動運転移動サービスとして事業運行中



### ひたちBRT(茨城県日立市)

- ・鉄道跡地をバス専用道路空間として整備
- ・一般車両や自転車などが存在しない
- ・2025年2月3日より乗務員乗車型レベル4自動運転移動サービスの長期営業運行を開始
- ・遠隔監視型レベル4への移行に向けた活動を実施中



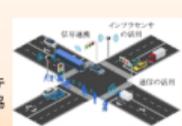
### 新東名高速道路

- ・日本の大都市間（東京～名古屋）を接続する高速道路
- ・従来東名高速道路に並行し、現在、6車線化（片道3車線）の整備
- ・駿河湾沼津SAから浜松SA間で新設された自動運転車優先レーンを活用した実証実験等の活動を実施中



### 柏の葉(千葉県柏市)

- ・柏市柏の葉地域において、2025年度内にレベル4自動運転移動サービスを実装する活動を実施している
- ・他地域展開に向け、協調型システムの開発・評価環境の汎用化、協調型路側機やデータ連携プラットフォームの共通仕様化を回る



## Tesla や 中国の自動車メーカーの最新の運転支援技術

## 海外では既に L4 越え？① Tesla FSD 14.x (Supervised = 監視義務あり)

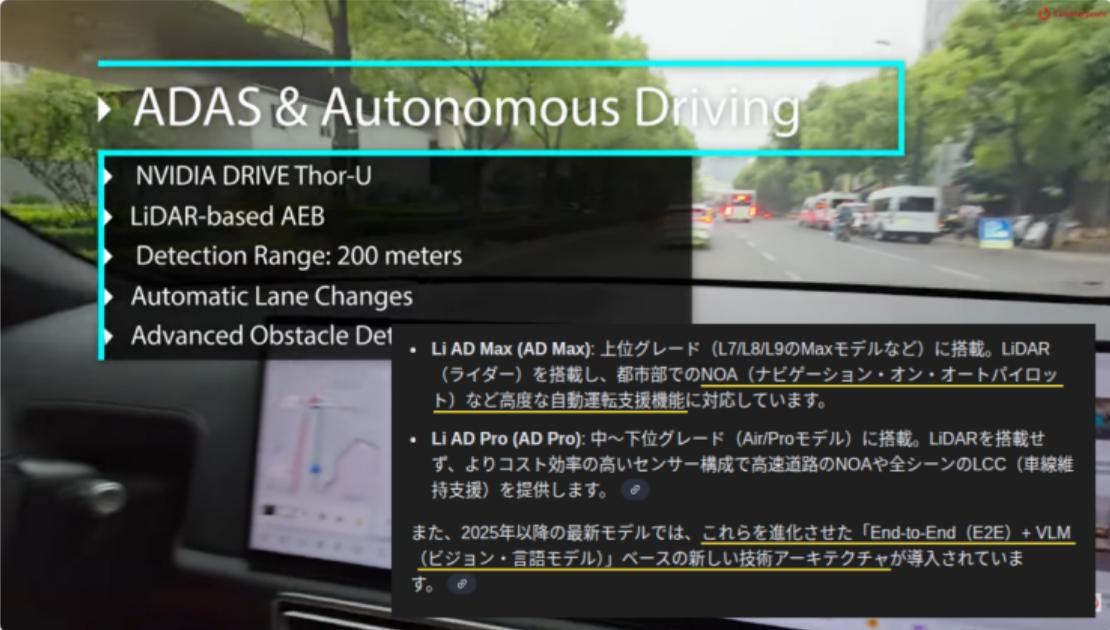


Tesla は SAE の自動運転レベルを  
使わず、**FSD=Full Self Driving**  
と自動運転技術を総称している

**FSD 14.x** は運転者が即座に  
運転を代われる必要があり  
レベル 2 + 相当に分類されるが  
実際には **L4 に極めて近い** 能力

[https://www.youtube.com/watch?v=t1ve\\_ttBEPw](https://www.youtube.com/watch?v=t1ve_ttBEPw)

## 海外では既に L4 越え？② LiAuto（理想汽車）Li AD Max



▶ ADAS & Autonomous Driving

- ▶ NVIDIA DRIVE Thor-U
- ▶ LiDAR-based AEB
- ▶ Detection Range: 200 meters
- ▶ Automatic Lane Changes
- ▶ Advanced Obstacle Det

- Li AD Max (AD Max): 上位グレード (L7/L8/L9のMaxモデルなど) に搭載。LiDAR (ライダー) を搭載し、都市部でのNOA (ナビゲーション・オン・オートパイロット) など高度な自動運転支援機能に対応しています。
- Li AD Pro (AD Pro): 中〜下位グレード (Air/Proモデル) に搭載。LiDARを搭載せず、よりコスト効率の高いセンサー構成で高速道路のNOAや全シーンのLCC (車線維持支援) を提供します。

また、2025年以降の最新モデルでは、これらを進化させた「End-to-End (E2E) + VLM (ビジョン・言語モデル)」ベースの新しい技術アーキテクチャが導入されています。

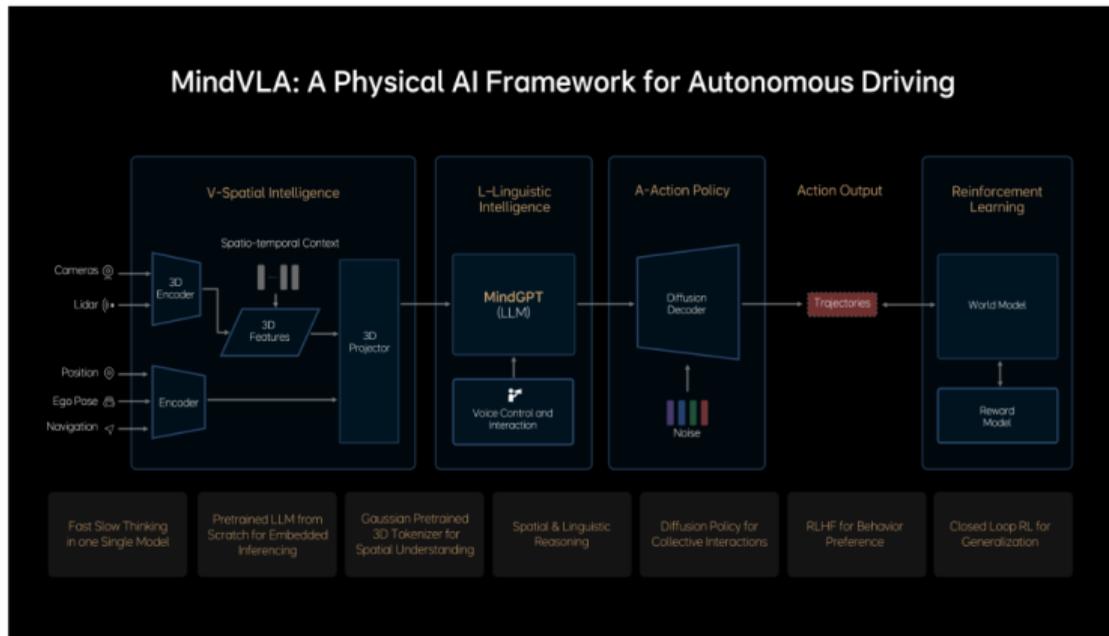
<https://www.youtube.com/watch?v=72l5oTMK-h4>

## 日本の L4 プロジェクトとは 自動運転の基本的な考え方に本質的な差 が

	RoAD to the L4	Tesla FSD など
SAE レベル	Level 4	Level 2 (2+)
ODD (Operational Design Domain：運行設計領域)	意図的に極小化	事実上無制限
運転責任	システム	常に人間
目標信頼性レベル	99.999...%	99% でも成立
事故時責任	事業者・メーカー	ドライバー
先進AI技術活用	消極的（説明責任）	非常に積極的

この2つの取り組みが達成すべき目的・制約・責任構造は根本的に異なっている

# LiAUTO の次世代自動運転：mindVLA = 最先端 AI 技術をフル活用



<https://autonews.gasgoo.com/articles/news/li-auto-unveils-next-gen-autonomous-driving-architecture-mindvla-70036319>

## 第三部：AI 自動運転技術の最前線

## 認識型（Perception）AIの現状

## 現代の AI の二大潮流（認識型・理解型）

### 認識型の AI (Perception AI)

- アルゴリズム：CNN、DNN など
- モデル：ResNet, Yolo, MobileNet
- 処理内容：認識、検出
  - 標識認識、車線認識 → 運転支援
  - 音声認識 → アレクサ、Siri など
- 推論処理は電力効率に優れた組み込みプロセッサ (10TPOS 級) で実行可能
- ほぼリアルタイムで高精度に物体認識ができる (理解系 AI の不得意領域)

### 理解型の AI (Cognitive AI)

- アルゴリズム：Transformer が主流
- モデル：GPT, Llama, Stable Diffusion など
- 処理内容：生成、対話
  - 自然言語による対話
  - 画像生成
- 認識系 AI の 10 倍～100 倍の計算資源 (CPU、メモリー) が必要になるので、現状はクラウドの計算パワーが不可欠

認識型 AI の推論処理は 10TOPS 程度で実行可能なので組み込み機器に搭載可能

# AI の実行には「学習（モデル作成）」と「推論」の2段階が必要

## 学習（Training）

- 大量のデータを使ってモデルのパラメータを最適化すること
  - 教師あり学習（ラベル付きデータ）
  - 教師なし学習（強化学習）
- 学習には **大きな計算資源** が必要
  - CPU 能力、ストレージサイズなど
- AI 認識の信頼度は、**計算精度よりも学習したデータの量に比例** する

## 推論（Inference）

- 学習済みのモデルを利用して、対象データを分類すること
- 推論処理は **相対的に軽量** で、SOC に内蔵された HW 資源を有効活用できる
  - CPU（汎用）
  - GPU（ビデオカード）
  - DSP（積和演算器）
  - NPU（専用 AI エンジン）

推論処理にはクラウドの計算資源は必須でない ⇒ エッジだけで AI 処理完結が可能

# AI 推論をエッジだけで実行する：その「アドバンテージ」と「課題」

積極的に「エッジ内で AI 認識処理を実行させたい場面」が増えている

## ■ アドバンテージ

- リアルタイム応答性：クラウド呼び出しが不要
- 認識精度の高さ
- データをアップロードしないので プライバシー／セキュリティが担保される  
例、独居老人の見守りなど
- インターネット ネットが利用できない環境 でも実行可能  
例、ドローン、ロボットへの適用など

## ■ 課題

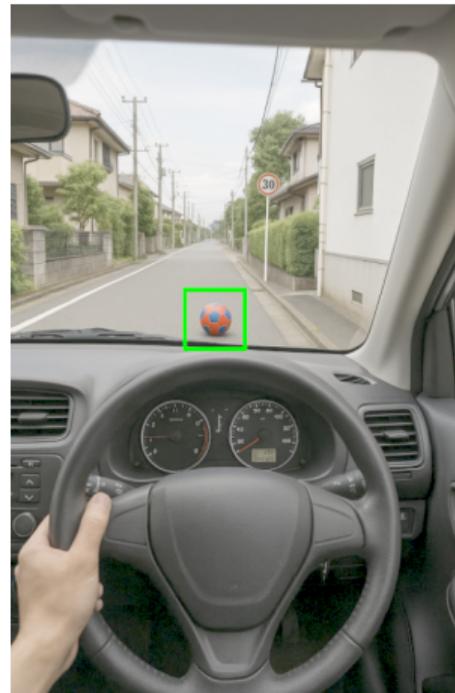
- 組み込みプロセッサの処理能力：⇒ NPU 内蔵の組み込みプロセッサが充実  
⇒ エッジ側で AI 推論を実行できる環境が揃ってきた
- 電力効率 (TOPS/W) の最適化：クラウド AI の電力要求は青天井
- プログラム API：CUDA 以外の AI 実行環境の整備

## 認識系エッジAIの再定義（VLM/VLA/エージェント）

# CNN (Convolutional Neural Network)

## CNN は「画像に何が写っているかを識別」する

- 事前に大量の画像から「検出したい物体」のモデルを用意
- 画像の特徴量を抽出し、視覚情報の中からパターンを認識
- 識別結果は構造化された固定形式で出力される
  - ラベル、ボックス、マスクなど
- 代表的な用途
  - 物体分類 (Cat / Dog / Car)
  - 特徴量抽出 (Edges, Shapes)
  - 位置検出・領域分割
  - 顔認識・特徴比較
  - 道路標識・車線認識
- 学習していない物体は特定できない (例、道路で寝ている人)



# VLM（Vision Language Model）、VLA（Vision Language Action）

## VLM = 視覚情報を意味的に理解

- “画像の意味・関係性・意図”を推論
- マルチモーダル（画像 + テキスト）
- 認識精度や速度は CNN よりも劣る
- 特徴
  - 画像の意味を説明する
  - 画像に関する質問に応答する
  - 誰が何をしているかを推論
  - 画像から意図を推定（例：道路上のボールから子供の飛び出しを予測）
  - 見えていない事象の推測（場面補完）

## VLA = 視覚情報から何をすべきか推定

- VLM（画像と言語の統合理解）をさらに進化させ、視覚・言語・行動（Action）を統合的に扱うモデル体系
- ロボティクスやエージェント AI において、「見る → 理解する → 行動する」を一貫して学習・推論できる 枠組み
- 特徴
  - 言語指示だけで複雑なタスクを遂行
  - 未学習のタスクにも適応できる
  - 環境と目的に応じた柔軟な行動生成

# AI エージェント（Agentic AI）

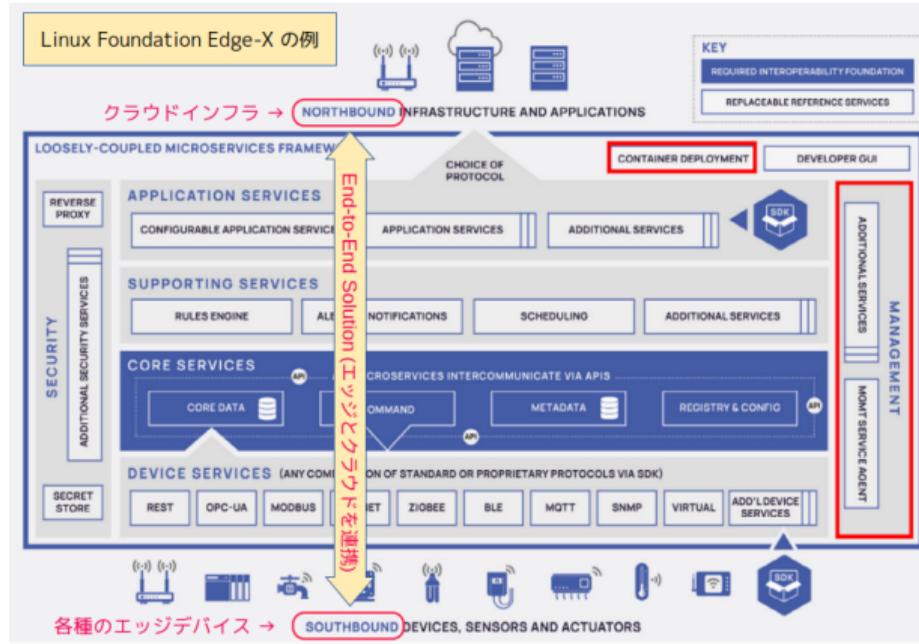
それぞれが別々の役割を持つ AI が協調してタスクを解く最先端のアーキテクチャ

- 今までは **人間が AI に指示（プロンプト）** → 今後は **AI 同士が自律連携** する
- 単一 AI モデルでは実世界の課題解決に役不足なので、役割をエージェントに分割
  - **認識エージェント**：外界を観測し意味的な状態を生成
  - **計画エージェント**：目的に基づき計画立案・推論・判断を行う。
  - **行動エージェント**：指示を実行し、環境に働きかける
  - **記憶エージェント**：長期・短期記憶、コンテキスト保存
- **AI エージェント間を密連携** されるためのプロトコル
  - **MCP (Model Context Protocol)**：エージェントの権限・能力管理
  - **A2A (Agent-to-Agent)**：エージェント間の協調

画像理解系 AI の実装では **認識エージェントの精度や速度** がボトルネックになる

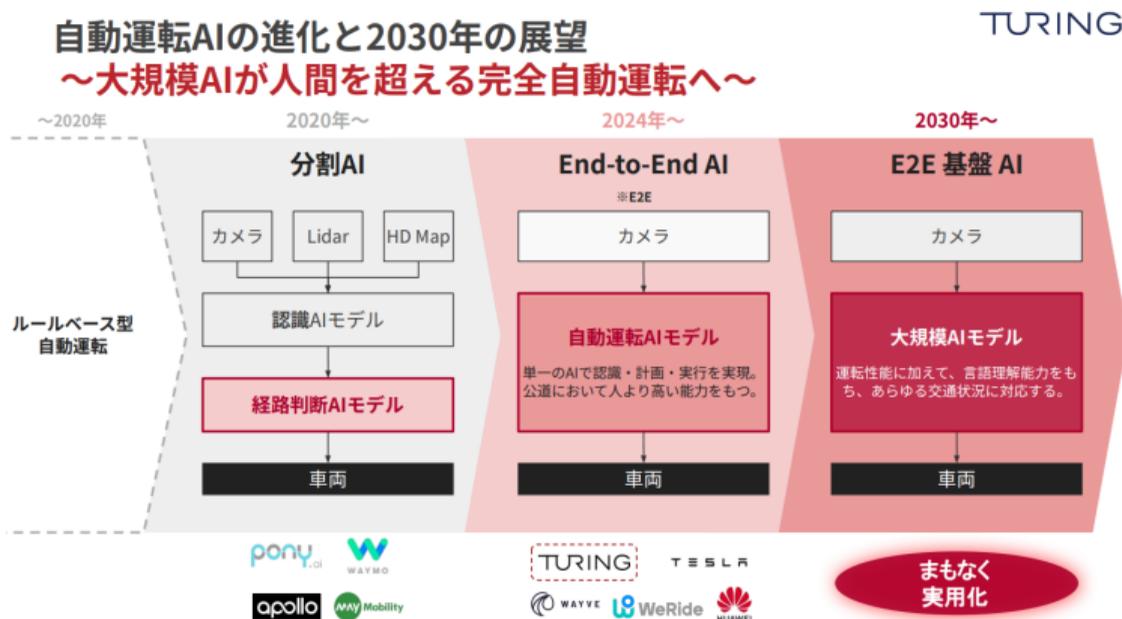
## E2E (End-to-End) 自動運転とは

# 一般的には "E2E" とは「エッジからクラウドを統合するソリューション」



[https://lfdge.org/wp-content/uploads/sites/24/2022/06/LFEdgeTaxonomyWhitepaper\\_062322.pdf](https://lfdge.org/wp-content/uploads/sites/24/2022/06/LFEdgeTaxonomyWhitepaper_062322.pdf)

# E2E（End-to-End）自動運転とは TURING が国交省で説明した資料



<https://www.mlit.go.jp/road/ir/ir-council/jido-infra/pdf03/04.pdf>

## 「ルールベース自動運転」と「E2E 自動運転」の違い

観点	ルールベース	E2E (End-to-End)
基本構造	認識→判断→制御を明示分離	センサー入力→操舵・加減速を一気通貫
設計思想	人がロジックを定義	データから挙動を学習
挙動の根拠	if-thenルール、状態遷移	ニューラルネットの重み
説明可能性	高い	低い (ブラックボックス)
検証・認証	形式検証・網羅試験が可能	網羅検証が困難
失敗モード	想定外条件で停止、保守的	想定外でも動くが暴走リスク
データ依存	比較的低い	極めて高い
開発スピード	遅い	早い
主な用途	L3/L4 (責任がシステム)	L2/L2+ (人が責任)

## E2E 自動運転によってクルマは人間を超えていく？

- LLM 系のAI技術（VLM）により画像の意味を自然言語で説明（Explainability）し、歩行者や他車の意図推定、複雑操作のタスク分解、危険シナリオを生成できるようになった。
- Tesla FSD14 では100msec以下の応答性能が確認されており、**人間の反射時間を超えた**。  
（陸上競技連盟の規定では、合図から100msec以内のスタートはフライングになる）
- 自動運転に使われるAIでは**誤判断が即事故**につながるため、**E2E系の自動運転制御**を外部の**ルールベース判定などで補完する**さまざまな構成が必須であり、実際に検討されている。

比較対象	典型反応時間	特徴
人間の視覚（V1）	180 – 250 msec	多段処理で遅いが「意味理解」は深い
人間の聴覚	140 – 160 msec	ショートカット回路で早い
VLM（AI 視覚モデル）	10 – 50 msec（GPU上）	桁違いに早いの意味誤りが起きやすい
VLA（E2E 自動運転）	5 – 20 msec（Tesla FSD14）	反応速度は人間より100 msec以上早い

## まとめ

- スマートフォンの普及で、自動車でも SW による機能アップデートを求めるようになってきているが、実現には SW アーキテクチャの全面刷新が必要になる。
- ネットワーク経由の SW アップデートへの対応など自動車向けの SoC も高機能化が進んでおり、合わせて SW が大規模化している。自動車開発に関わる企業間での SW 開発スキームの再定義も必要になっている。
- 自動運転技術に対するニーズは高いが、目指すべきゴール（運転者の責任のあり方）によってアプローチには大きな違いがある。Tesla FSD に代表される新世代運転支援では、従来からある単純な物体認識（Perception）に加え画像の意味を理解（=VLM）し行動を喚起（=VLA）する新しい AI モデルが採用されている。さらに認識 → 制御を一つのモデルに集約（=ブラックボックス化）して運転支援の適用領域を大幅に拡大する End-to-End 自動運転が 2026 年以降に本格化。