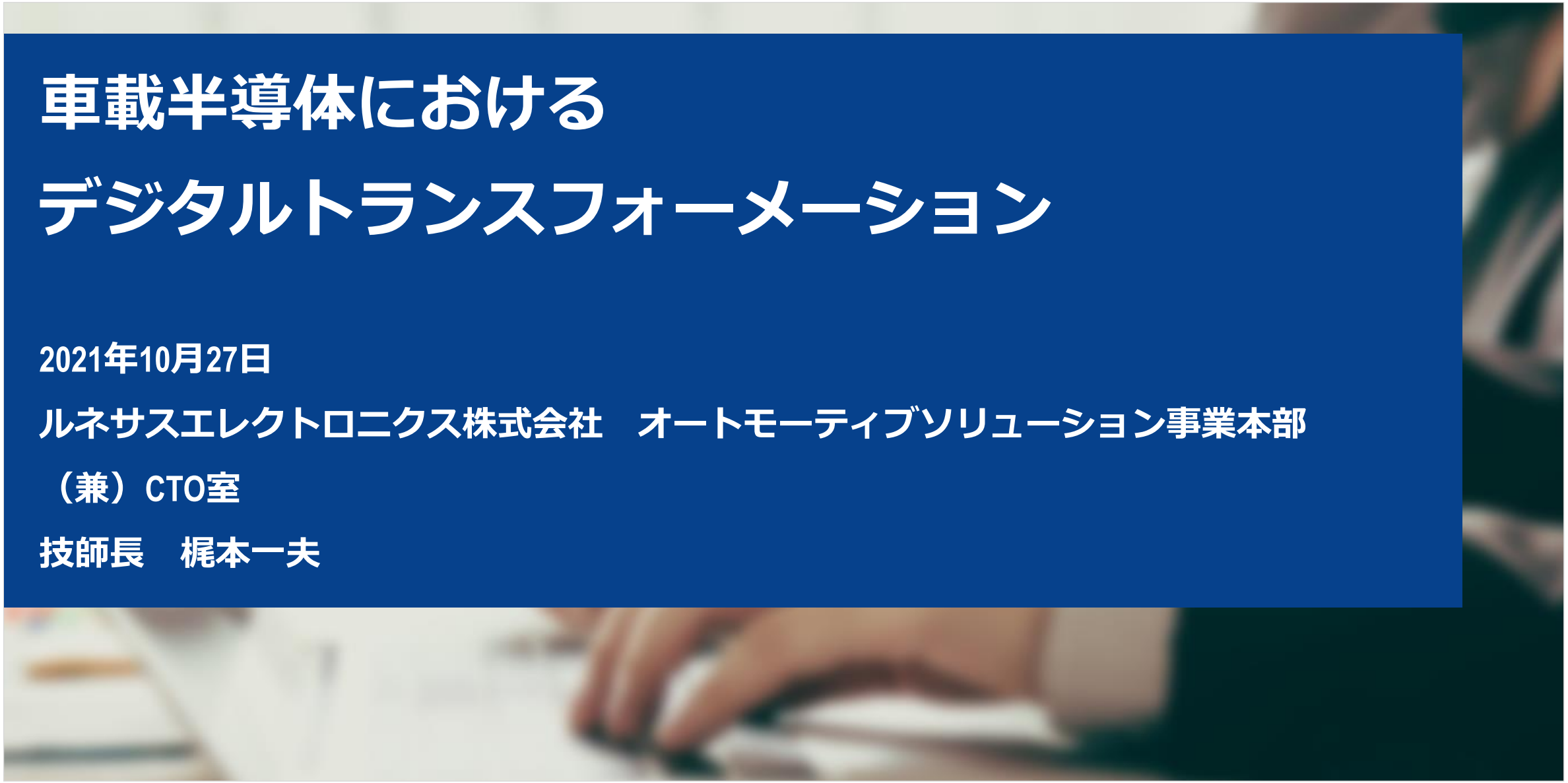
車載半導体における
デジタルトランスフォーメーション

2021年10月27日

ルネサスエレクトロニクス株式会社　オートモーティブソリューション事業本部
（兼）CTO室
技師長　梶本一夫



ルネサスとは

ルネサス エレクトロニクスは、グローバルな半導体会社です。
人々が安心・安全に暮らせる社会を実現するために、あらゆるモノとモノをつなぎインテリジェント化することを通して、組み込み機器に進化をもたらしています。

そして、無限の未来をカタチづくるために、自動車、産業、インフラ、IoT分野に対して、世界的に高いシェアを誇るマイコンに加え、アナログ&パワーデバイス、SoCなどの各種半導体と幅広いソリューションを提供しています。

SoC: System-on-a-chip

本社所在地
東京都江東区

連結従業員数
～ 19,000人

関連会社所在国数
28カ国

2020年売上収益
7,157億円

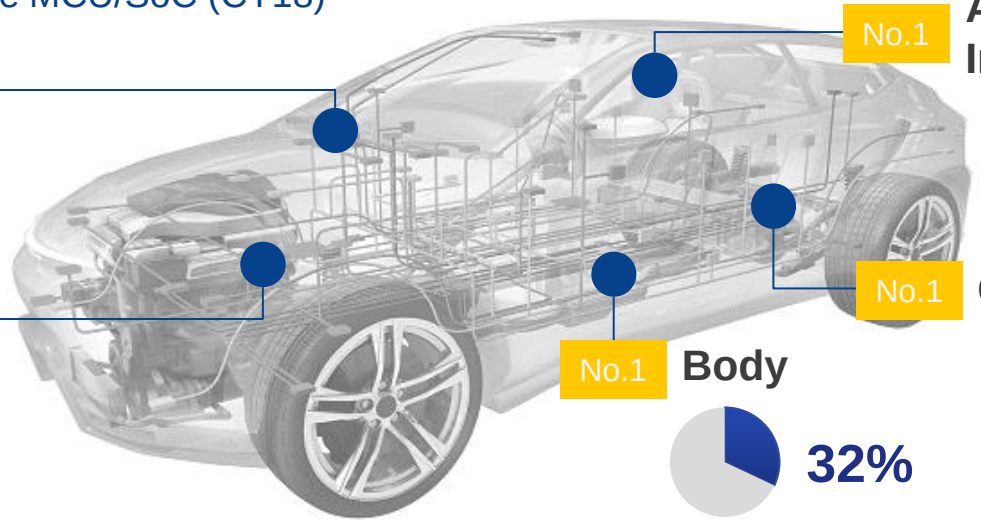
特許取得数および出願中件数
約20,000件

AUTOMOTIVE

ACHIEVE GROWTH IN AUTOMOTIVE BUSINESS WITH MARKET-LEADING MCU/SOC

- **Leading MCU/SoC supplier** – W/W shipments reached aprox.1.3B units in 2018
- **Best in quality** – Extremely low failure rate at 0.1ppm
- **Advanced process** – 16/14nm FinFET for SoC, 40nm to 28nm cutting edge process for MCU

Market Share in Automotive MCU/SoC (CY18)*1



Powertrain No.1

33%

xEV No.1

28%

No.1 **ADAS + Infotainment & Instruments** *2

32%

No.1 **Chassis & Safety** *3

21%

No.1 **Body**

32%

RENESAS R-Car

RENESAS RH850

RENESAS RL78

*1: Renesas' revenue estimate in each segment is based on the market analyses by Strategy Analytics 2019. / *2: Including infotainment and instrument according to the definition of Strategy Analytics. / *3: Excluding ADAS. / MCU: Microcontrollers / SoC: System-on-a-chip

自動車におけるデジタルビジネス

- デジタルトランスフォーメーション(DX) -

TECHNOLOGY AND MARKET TREND

PLAYER CHANGE INNOVATION

- Trends: DX, Carbon free, PACE for IoT and Automotive market

  – DX : Connected, Automated and Big data management
  – PACE: Personalized, Autonomous, Connected and Electrified
    (Former CASE: Connected, Autonomous, Social and Electrified)

Targets	Core Technologies
Personalized	AI, Security
Autonomous	AI, Safety, Cloud native
Connected	Security, Cloud native
Electrified	System (Digital, Analog and Power) control
Big data management	AI, Cloud native
Carbon free	System (Digital, Analog and Power) control

今、自動車業界で起きていること – DX前後で主役が変わる恐怖 –

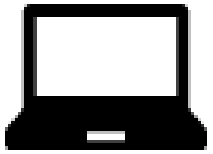
デジタルトランスフォーメーション前

デジタルトランスフォーメーション後

単品・セットメーカの天下

垂直型
プラットフォーマ

水平型
プラットフォーマ



IBM NEC

NOKIA

多くの自動車メーカ

淘汰

Microsoft intel

android Qualcomm

WAYMO NVIDIA

android Qualcomm

産業機器、スマートビル、スマートホーム、ひいてはスマートシティも続く

DX勝ち組のビジネス戦略＝Software First（梶本の理解）

Software First = お客様とのエンゲージメント（長期にわたる信頼関係）を、ビジネスの勝ちポイントに置く

実は、この勝ちポイントは古来から変わっていない。実現する手段が変わった。

Software First＝お客様とのエンゲージメントをSoftwareにより実現する
- Softwareにより顧客とダイレクトにつながり続ける（customer ID）
- Softwareにより顧客の嗜好に合わせてサービスを常に変化させる（CI/CD）
- 一瞬の高額対価ではなく低額で長期に渡る対価を得る（Subscription）

旧来の材料・部品・完成品・流通と言う売り手側のエコシステムではなく、Softwareにより顧客を巻き込んだエコシステムを形成する：そのための手段がSoftware Platformとなる

Software Platformに最適なモノづくりに発想が変わると、モノはSoftware Definedになる

自動車業界のGoogle（Platformer）になりたいメーカ

自動車メーカ	SWプラットフォーム	推進母体	備考
トヨタ	Arene	Woven Holdings.	・GoogleよりJames Kuffner氏を招聘 ・ROS2**に倣ったSOA***による車内外制御
フォルクスワーゲン	VW.os	Car.Software.org →CARIAD (Car, I am digital)	・クラウド連携部のVW.AC****をMicrosoftと提携し企画
ルノー・日産・三菱	FACE	Renault Software Lab	・Autosar AP*****を使ったSOAによる車内制御

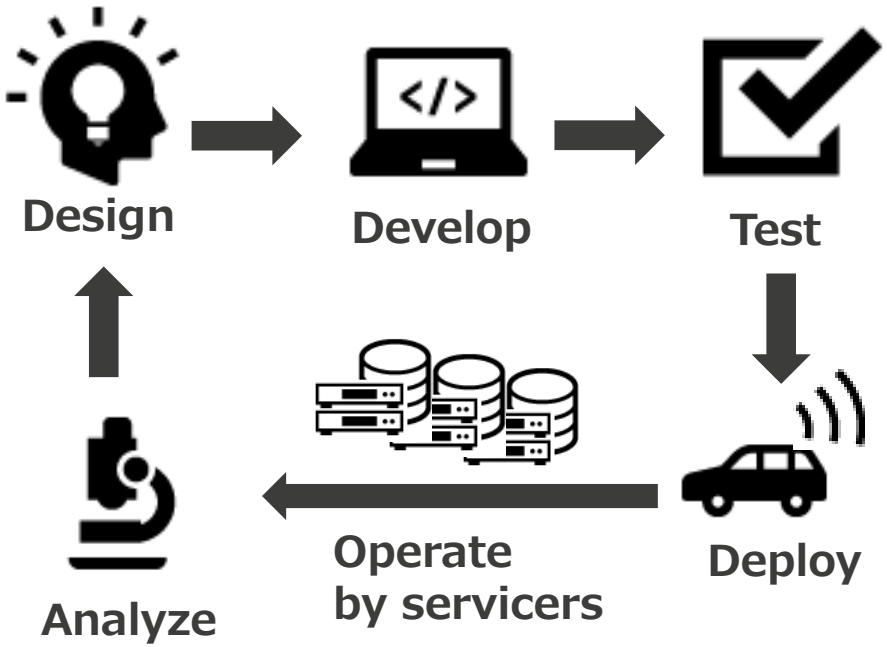
VS

ITベンダ	SWプラットフォーム	推進母体	備考
Alphabet	Android Automotive OS	Google	・スマートフォンでのアプリシェアを優位に自動車の全エンタメ系PFを独占の勢い ・SoCを性能余裕で選別始める
Alphabet	Waymo	Waymo	・自動運転のSWプラットフォーム

*TRI-AD: Toyota Research Institute Autonomous Driving
**ROS2: Robot OS version 2
***SOA: Service Oriented Architecture
****VW.AC: Volks Wagen Automotive Cloud
*****Autosar AP: Autosar Adaptive Profile

トヨタ様の大戦略 – Software Firstでも自動車の安全を担保 –

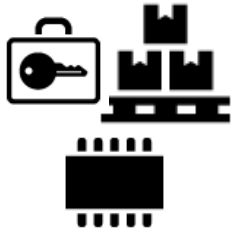
スマートカー SWライフサイクル



スマートカー SWプラットフォーム

従来は、半導体メーカは自社半導体のデバドラ、OSを用意、Tier1、OEMはその上でのソフトを開発すれば良かった（ランタイムのみ）

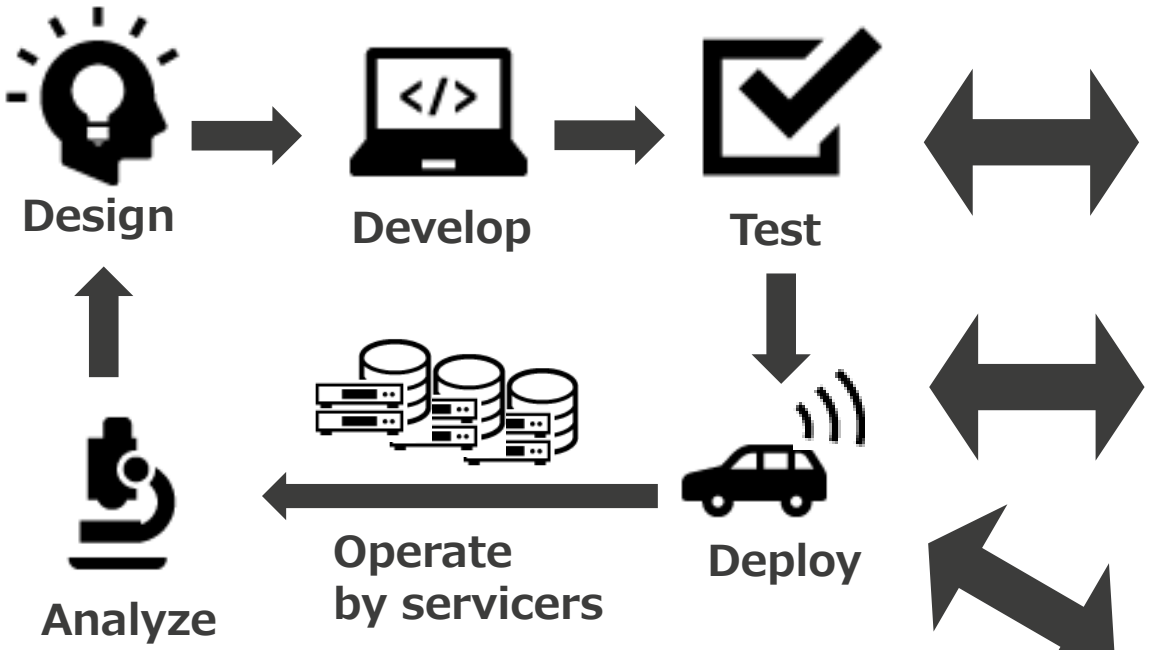
3. ランタイム



AUTOMOTIVE GRADE LINUX

トヨタ様の大戦略 – Software Firstでも自動車の安全を担保 –

スマートカー SWライフサイクル



スマートカー SWプラットフォーム

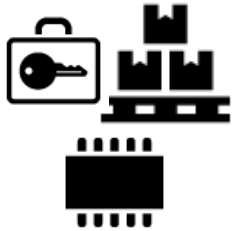
1. 仮想開発環境（MILS, SILS, HILS*）

*MILS: Model in the Loop Simulation
SILS: Software in the Loop Simulation
HILS: Hardware in the Loop Simulation

2. CI/CD**環境

**CI/CD: Continuous Integration,
Continuous Delivery

3. 機能安全・セキュアランタイム



arene

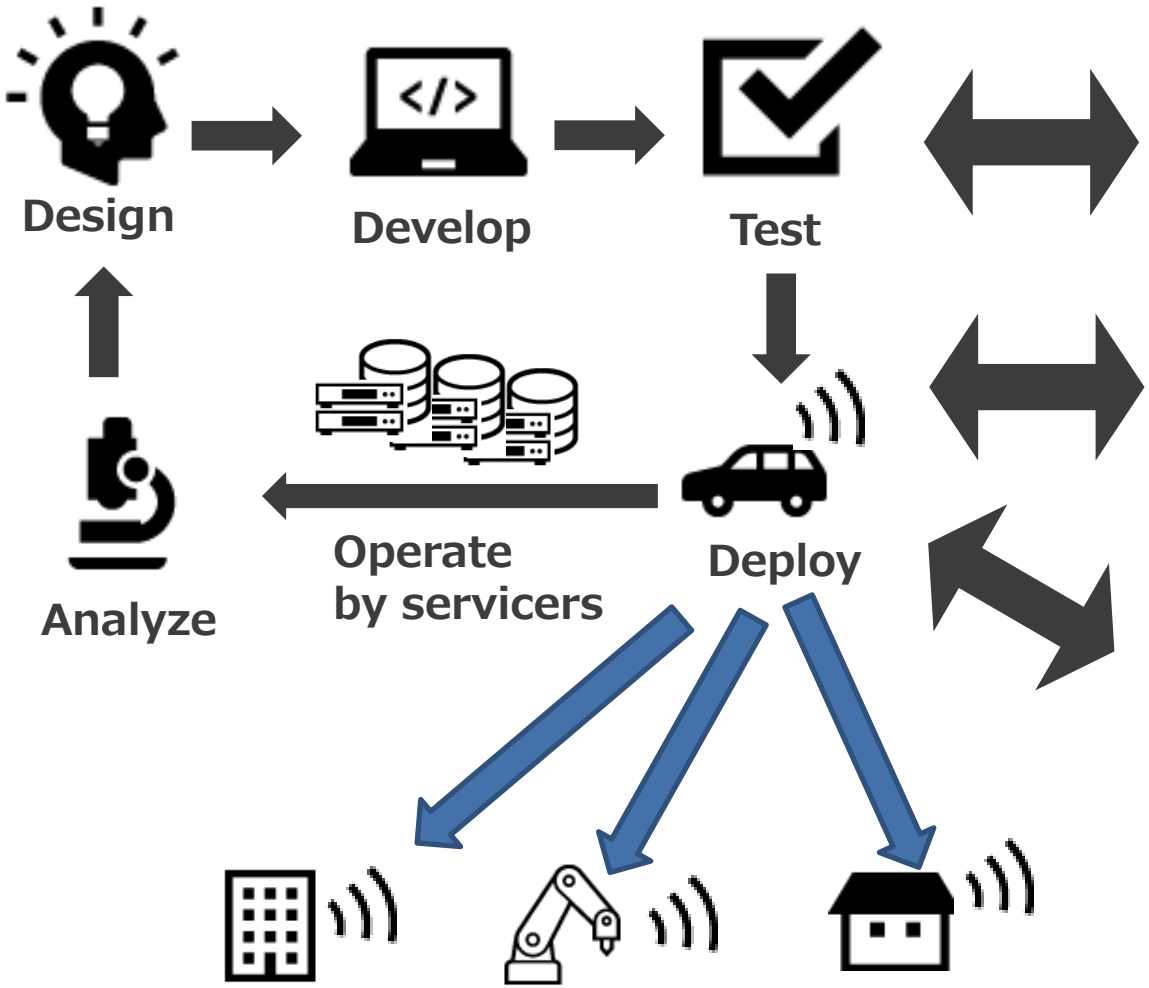
自動車では特に安全性の証明が必須

↓

安全で常に改善されるソフト環境を提供

トヨタ様の大戦略 – Software Firstでも自動車の安全を担保 –

スマートカー SWライフサイクル



Arene SWプラットフォーム

OSS（オープンソース）の進化を享受

↓

機能安全認証済のOSS由来のソフトを利用しプラットフォーム構築（商用版）

↓

自動車向けを工場向け、インフラ向けと安全認証が必要な分野に展開
（Googleでは参入が難しい分野）

↓

都市全体をトヨタのSWプラットフォームで覆い、安心・安全な生活を支える

arene

OSSの破壊力

- Software Firstの主役 -

なぜOSSを使うのか？

企業がOSSを使う理由

スピード

イノベーション

業務上のメリット

低コスト

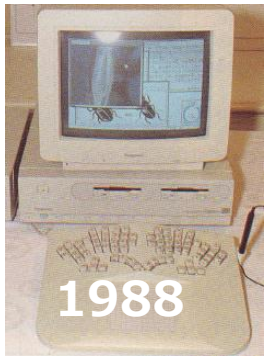
柔軟性

Open Source Sense

これからはOSSでなければならない理由

・多くの支持を集めたソフトのみ生き残り、成長を継続する

・一社だけの尖ったソフトはガラパゴス化し、やがて廃れる
　＃過去の日本はガラパゴスの宝庫



BTRON
ビデオ機能で
世界初
(1993退場)

ノンリニア編集機
HDD編集世界初
(1999退場)

音楽配信サービス
携帯向け世界初
(2004退場)

2000

ガラケー
カメラ搭載世界初
(2018退場)

2000

デジタルTV
データ放送世界初
(2021製造退場)

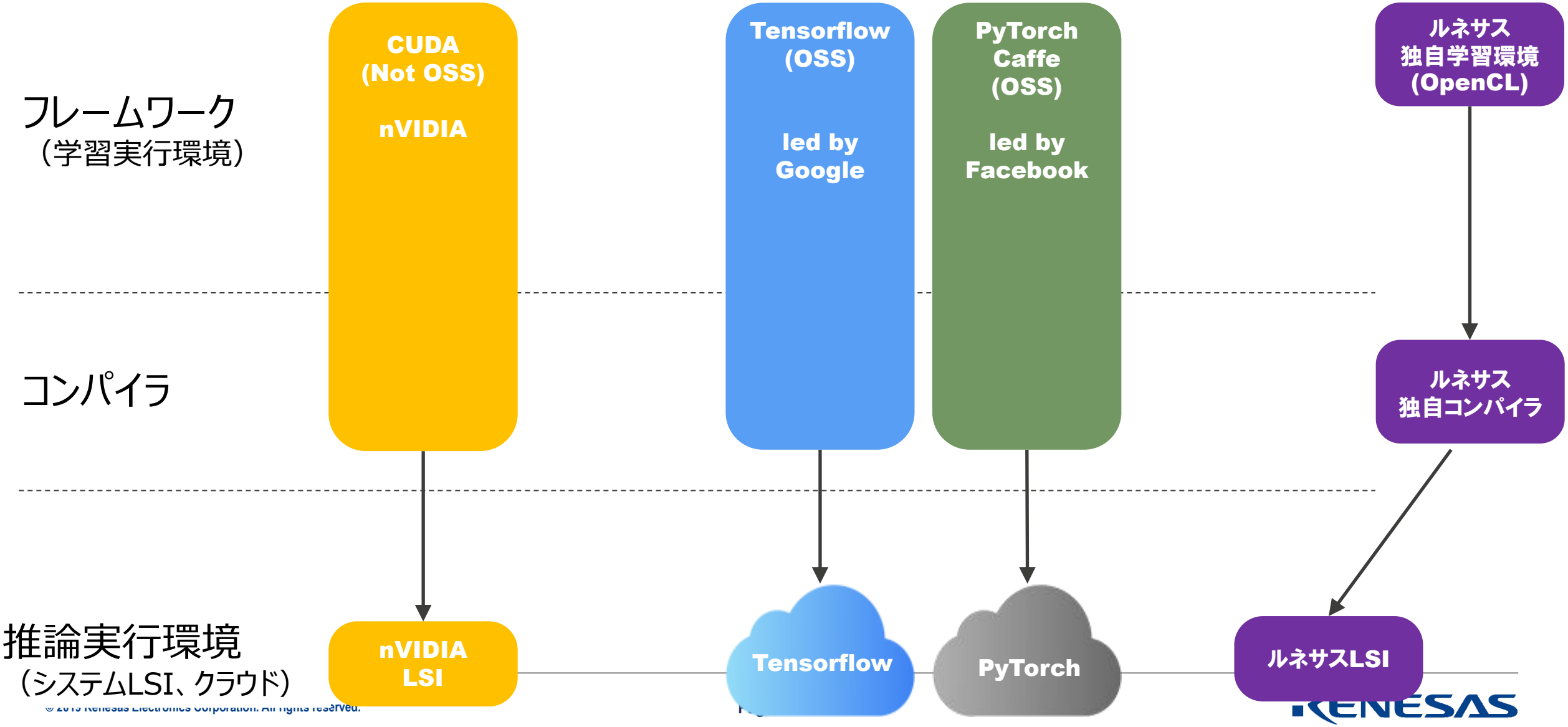
退場年は
Panasonic

・多くの支持を得るためにOSSへの投資をした会社が最後に勝つ

RENESAS

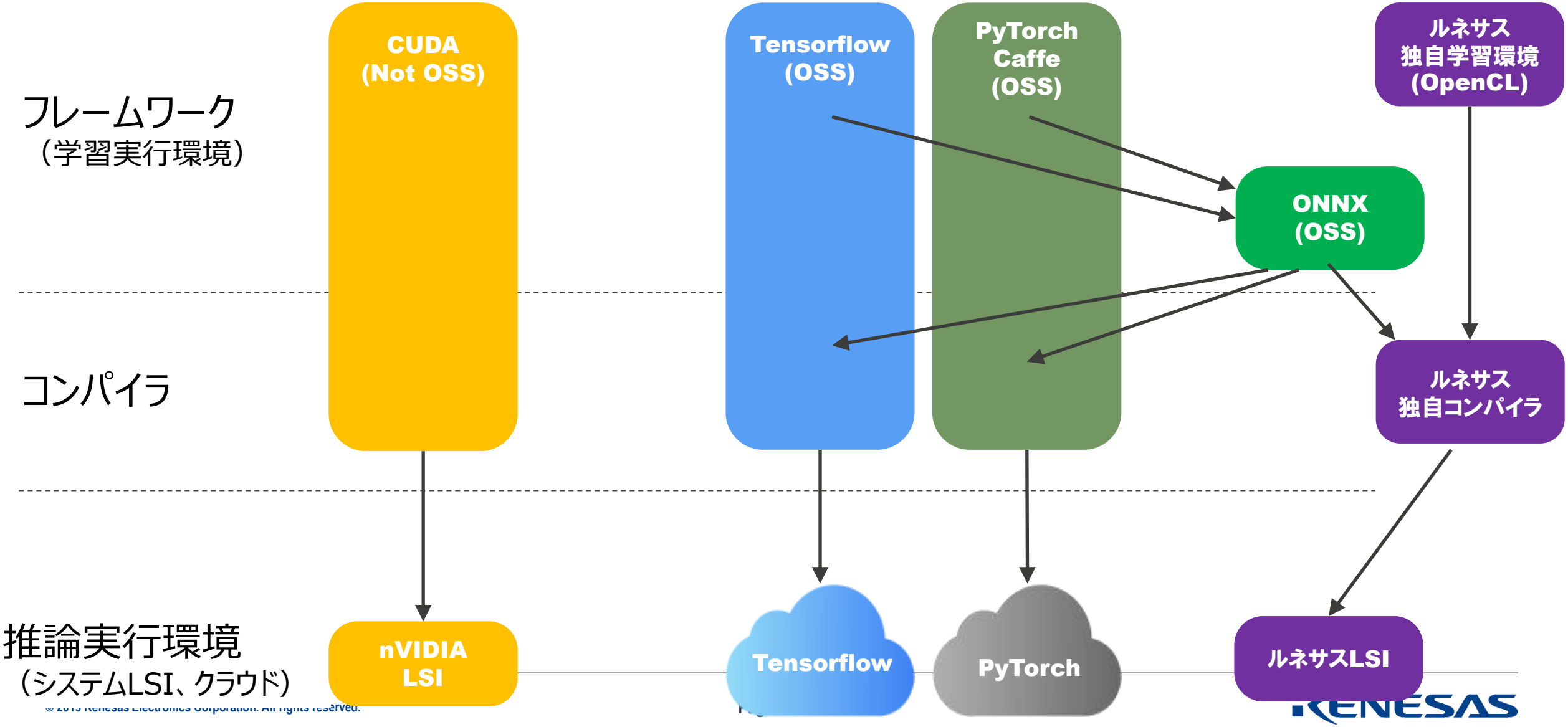
AIを取り巻くOSS化による主導権争い（１）

2017年までの業界構図（垂直統合型フレームワーク全盛）



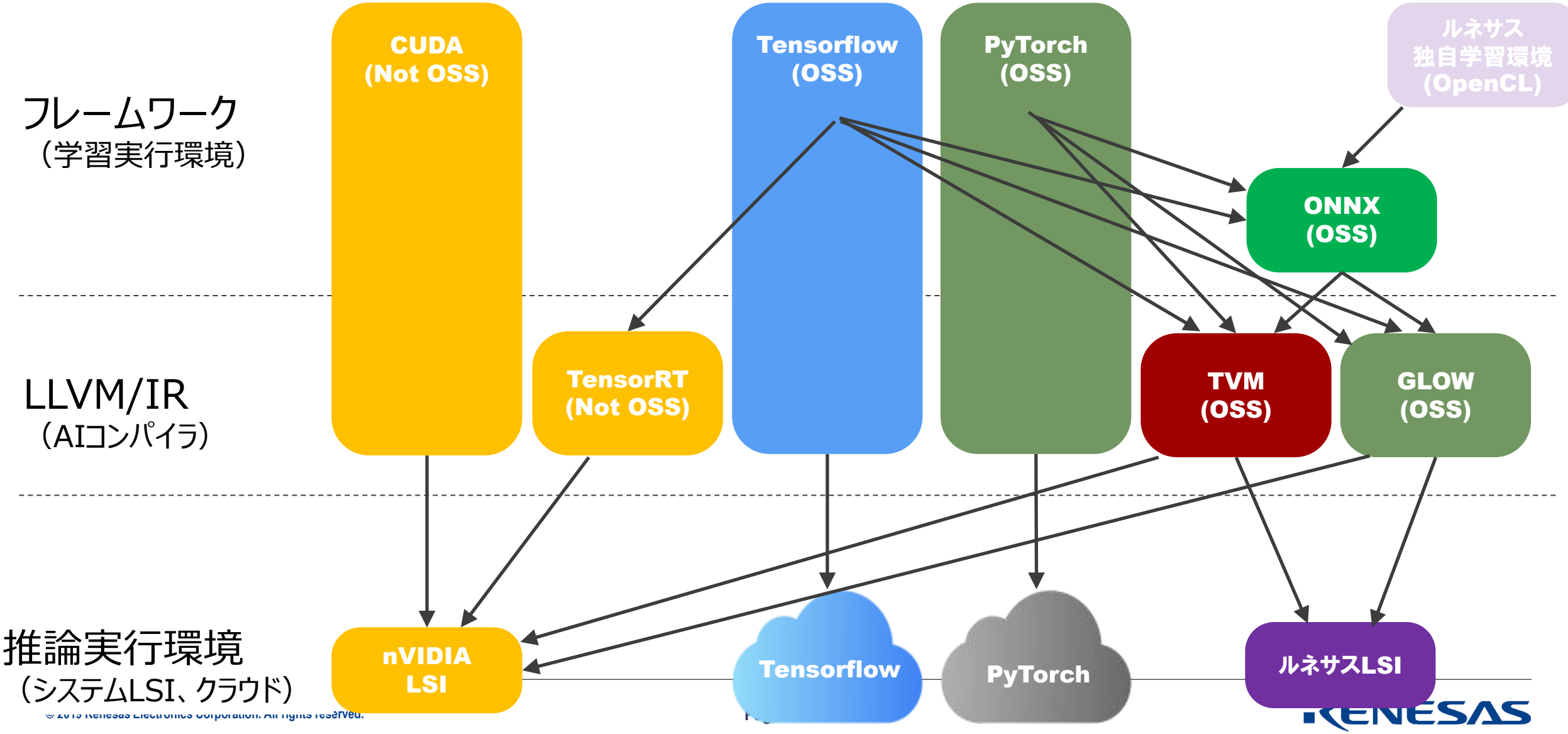
AIを取り巻くOSS化による主導権争い（２）

2018年までの業界構図（垂直統合型フレームワーク全盛で、一部はONNXで相互変換）



AIを取り巻くOSS化による主導権争い（３）

2019年以降の業界構図（各社が相互乗り入れによる多数派工作を仕掛けている）



組込みシステムでもOSS化が主流になる

OSSの特性	組込みシステム特性	スマホによる変革	スマホ以降の組込み	OSS側の歩み寄り
高速進化	信頼性・長期保証		OSSによる高速進化を享受	コミュニティによるLTSのサポート６年へ
ベストエフォート	実時間保証		ベストエフォートでも実時間で動作	
同一ハード環境	多様なハード環境	・PC並ハイスペック ・同一ハード環境をソフトが規定		

元々、OSSと組込みの特性は水と油

OSSでのLTS (Long Time Support)の動きは組込みを意識

ソフトにより規定されたハード（Software Defined）

P906iではリッジレーサがサクサク動く！
NやFでは無理。

ガラケー時代はセットのスペックが競争力

XperiaでLINEは動くがGalaxyではダメと言うことはあり得ない！

スマホ時代はセットスペック競争はご法度
アプリ互換性が命（同じものがSCALE）

PFの進化を決定
同じものがSCALE

OS=HOWソフト — **Google (Android)**

Set — Samsung / Sony / LG / Huawei / ------- / Oppo

SoC — **Qualcomm, MediaTek**

カメラ以外
差別化禁止

自動車制御にもOSS化の波が来ている

AUTOSAR Classic Platform	AUTOSAR Adaptive Platform
Same address space for all applications (MPU support for safety)	Each application has its own (virtual) address space (MMU support)
Optimized for signal-based communication (CAN, FlexRay)	Service-oriented communication
Based on OSEK = **RTOS**	Based on POSIX (PSE51) = **Linux vs QNX, eMCOS**
Execution of code directly from ROM	Application is loaded from persistent memory into RAM
Fixed task configuration	Support of multiple (dynamic) scheduling strategies
Specification	Specification as binding Standard Code as Demonstrator

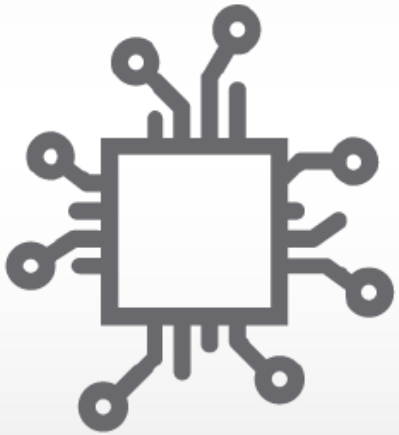
お客様の変化への適応

- ルネサスにおけるDX -

Our Purpose

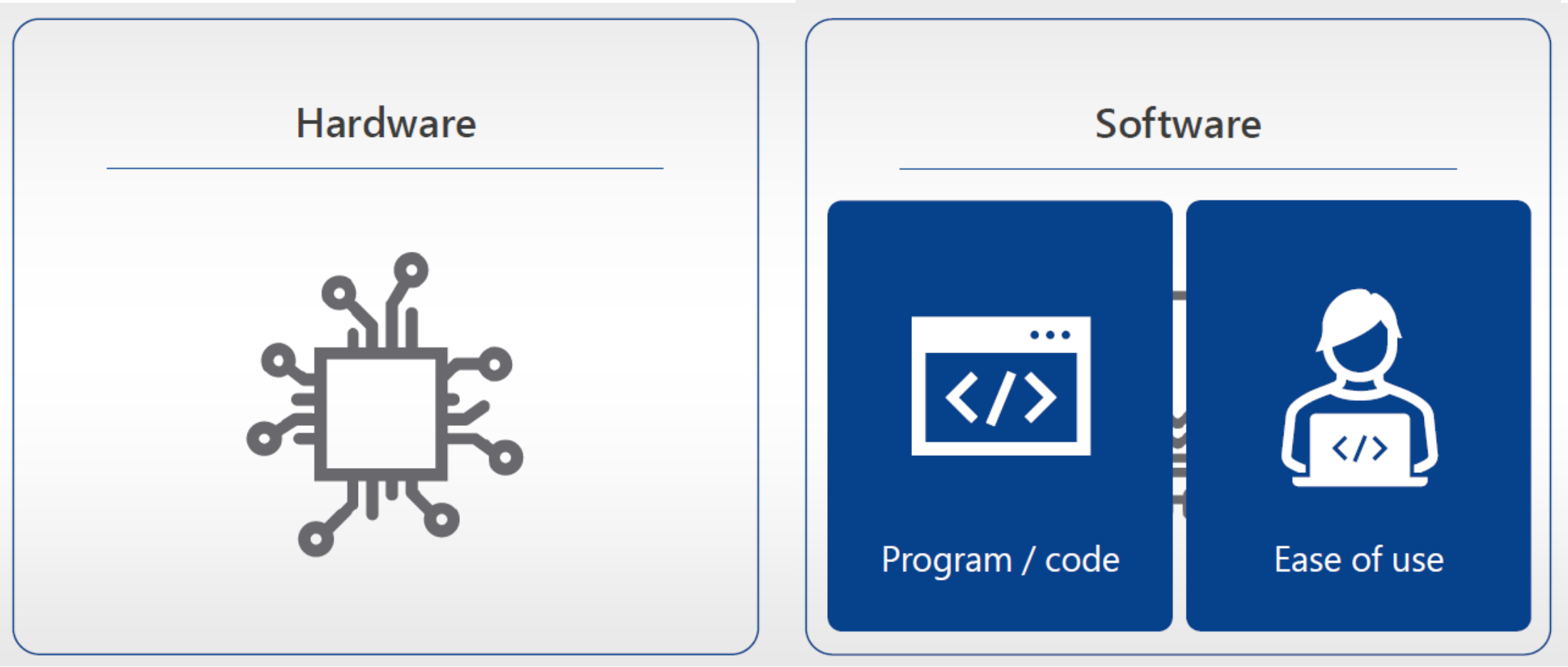
How do we make it easier?

Hardware



Software

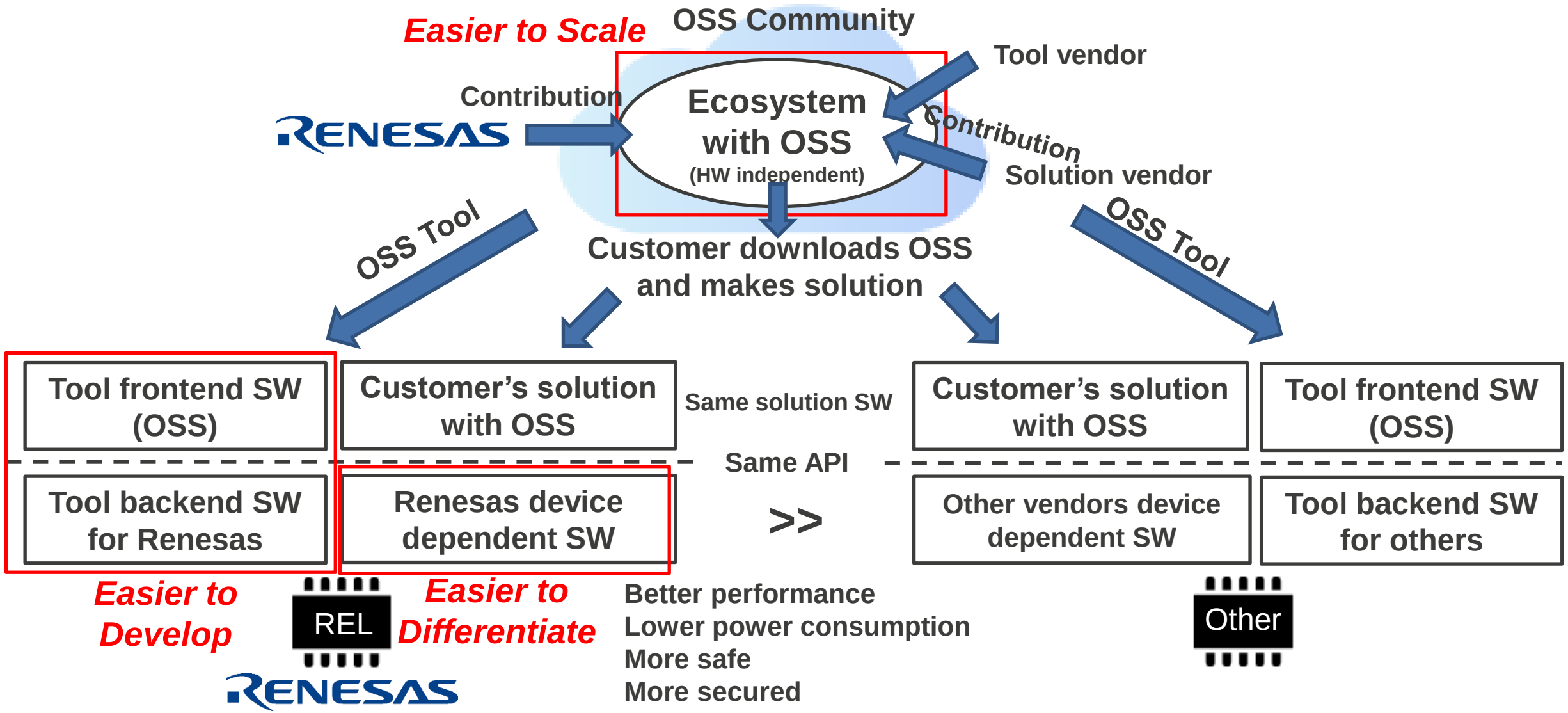
How do we make it easier?



Renesas Core Technologies

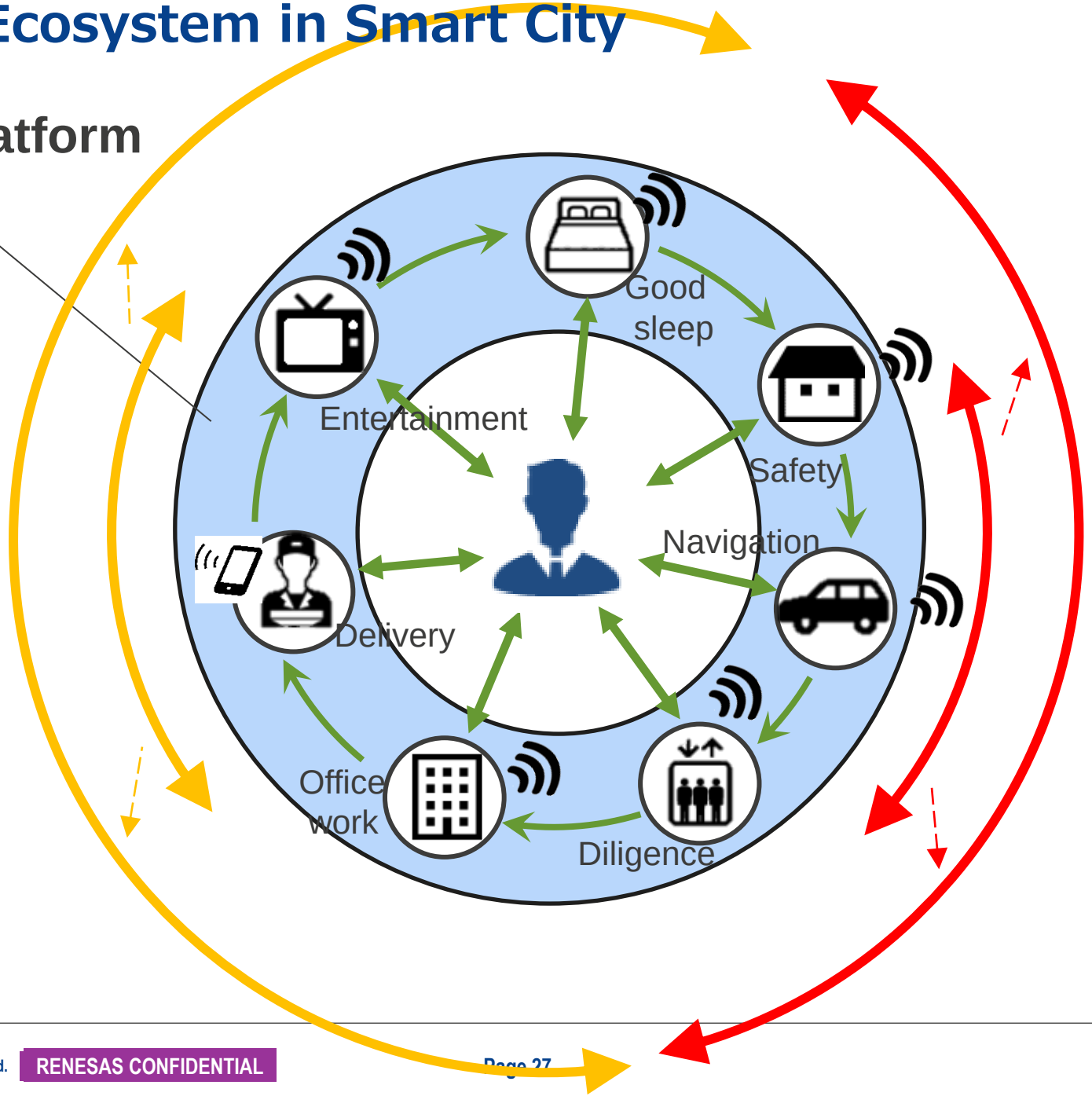
AI	Safety & security	Digital-analog-power synergy	Cloud native
			
for intelligence	for robustness	for system first	for scalability

SW development with OSS becomes more important



RENESAS

Human Centric Ecosystem in Smart City



FuSa needless services spreads from smartphone.

IIBU main market

FuSa required services spreads from automotive.

ABU main market
IIBU will follow.

ビジネスモデルを支える
ソフトウェアアーキテクチャ

- Microservice -

Webアプリ、クラウドサービスはOSSが主役（１）

ロゴ	名称	ライセンス	メンテ団体	コメント
	Linux	GPLv2 only	Linux foundation/ kernel.org	ITシステムから組込みまで幅広く利用されるOS
	Docker	Apache2.0	Docker.com	Linux前提で、OSやMWをパッケージ化したコンテナを扱うフレームワーク
	Kubernetes (k8s)	Apache2.0	Cloud Native Foundation	コンテナ群を管理するフレームワークでGoogleが開発
	Jenkins	MIT	Jenklns.io	CI/CD (Continuous Integration, Continuous Delivery)のコマンド群を自動実行
	Ansible	GPLv3	Red Hat	システム構成をセットアップする定型手順を管理、RPA (Robot Process Automation)で使われる

コンテナがもてはやされる理由(１)

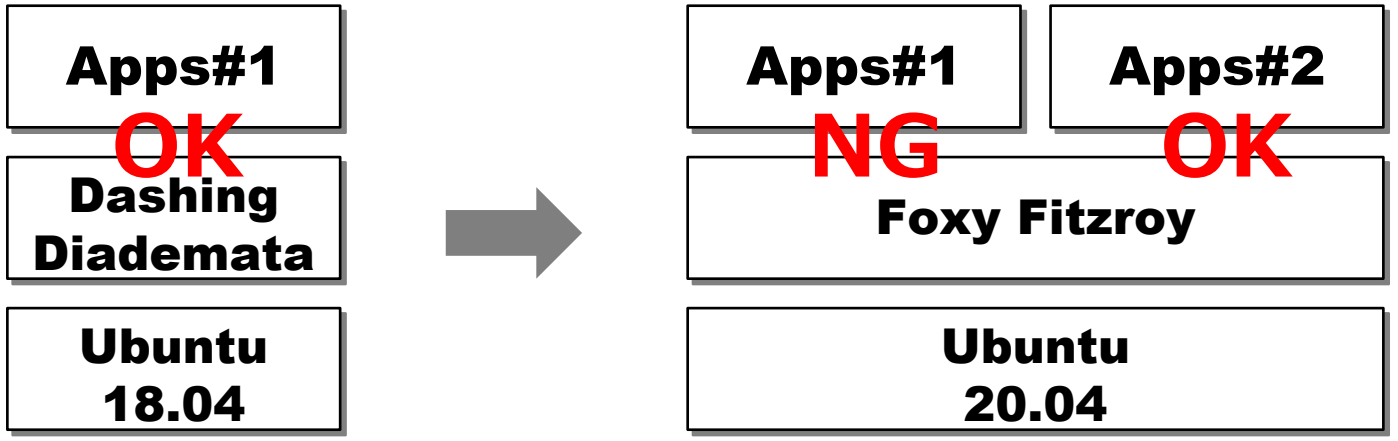
みんな悩まされる「**バージョン相性問題**」

ROS2			Ubuntu (LTS)		
バージョン	リリース時期	サポート期限	バージョン	リリース時期	サポート期限
Ardent Apalone	2017/12	2018/12	16.04	2016/4	2021/4
Bouncy Bolson	2018/7	2019/7	16.04	2016/4	2021/4
			18.04	2018/4	2023/4
Crystal Clemmys	2018/12	2019/12	16.04	2016/4	2021/4
			18.04	2018/4	2023/4
Dashing Diademata	2019/5	2021/5	18.04	2018/4	2023/4
Eloquent Elusor	2019/11	2020/11	18.04	2018/4	2023/4
Foxy Fitzroy	2020/6	2023/5	20.04	2020/4	2025/4

ROS2のDashing Diadematataでサービスを作り運用しているのに、それをサポートしながら、Foxy Fitzroyで追加された機能を提供して欲しいと言われたというケースを想像してください。

しかしUbuntsを20.04に上げると、Dashing Diadematataは正常に動作する保証がありません。（次ページ）

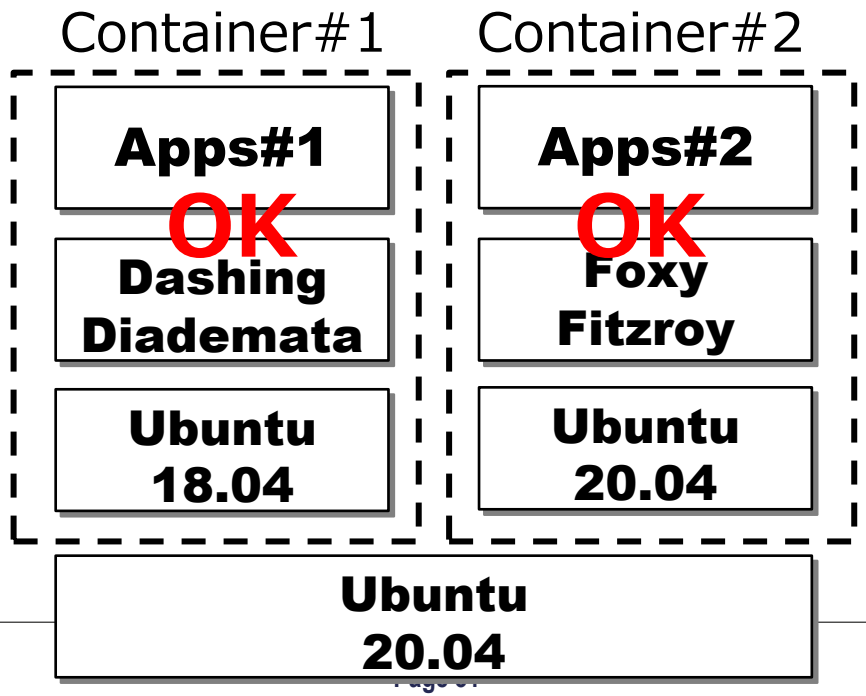
コンテナがもてはやされる理由（２）



Apps#2 runs on
ROS2 version Foxy Fitzroy
on Ubuntu 20.04 (Linux v5.4)

ROS2 version Dashing Diamata
cannot run on Ubuntu 20.04.
So, Apps#1 cannot run.

Apps#1 runs on
ROS2 version Dashing
Diademata on Ubuntu
18.04 (Linux v5.3)



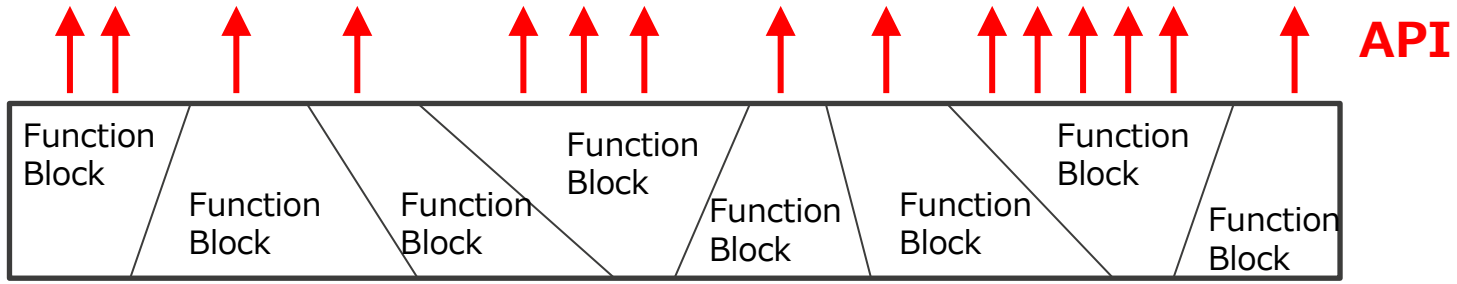
Thanks to Container, Ubuntu 18.04
and Ubuntu 20.04 can coexist.

So, Apps#1 and Apps#2 can coexist

コンテナがもてはやされる理由(３)

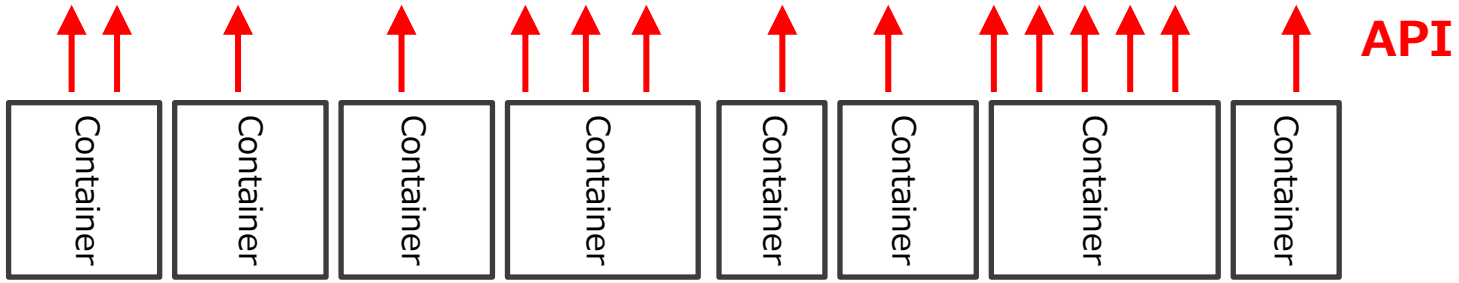
Microservice Architecture

Conventional Monolithic Architecture



Even if an API is changed, all software should be tested again.

Microservice Architecture



When an API is changed, only one container should be tested.
Customizing cost will be reduced drastically.

ビジネスモデルを支える
ソフトウェアアーキテクチャ

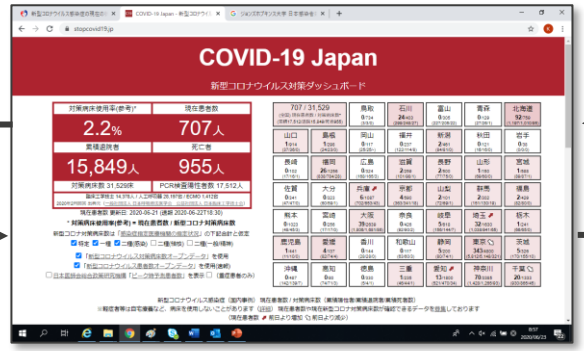
- Service Oriented Architecture -
(SOA)

Web SiteのSOA ≒Mush Up（参考）

Non-SOA world ⟵ | ⟶ *SOA world*

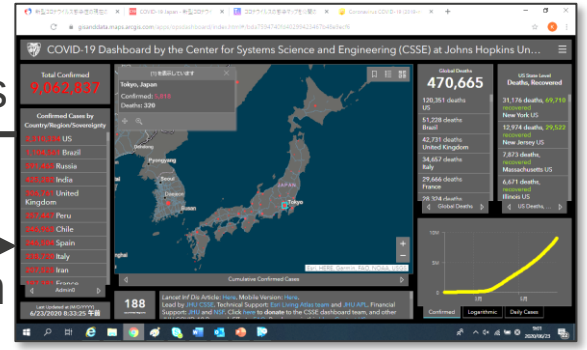
Crawl

Get text data



Access REST APIs

Get data on COVID-19



Web site of “Ministry of Health, Labor and Welfare (MHLW)” in Japan

Independent site without REST APIs

Web site of “COVID-19 Japan”

Crawling MHLW site and tagging data of COVID-19 infected persons
Providing REST APIs to access tagged data

Web site of “COVID-19” hosted by Johns Hopkins Univ.

Accessing REST APIs provided by each country service site on COVID-19 and summarizing data

車内でのSOA

Non-SOA world ⟸ | ⟹ *SOA world*

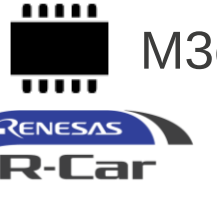
ABS system app (In vehicle closed service)

ABS worked notification (thru CAN)

ABS event wrapper app

Access REST API as notification

(publisher) (subscriber)



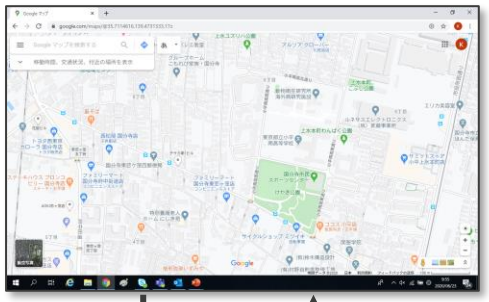
ABS system is designed in vehicle closed service.
But ABS related information runs thru CAN.

ABS event wrapper app hooks ABS related information and recognizes ABS worked.
Then this app publishes ABS worked information.

IVI app is notified ABS worked by subscribing events list.
Then IVI shows panel on navigation display.

車内外でのSOA

Web site of “Slip spots”



This site access many cars’ ABS history and shows the latest slip spots on the map.

Out of vehicle

In vehicle

Get ABS history

S4

Access REST APIs

M3e

S4 works as router within vehicle.

IVI app stores ABS worked time and place history and provide REST API to access the history.

Reference Architecture Example

ARM SOAFEE (Scalable Open Architecture for Embedded Edge) 2021/9

https://www.arm.com/ja/company/news/2021/09/new-arm-technologies-to-transform-the-software-defined-future-for-the-automotive-industry

・個々の機能単位はMicroservice Architecture

・アプリはあちこちに分散している機能をつまみ食いして作るSOA

・OSSで基本的に作成

・ASIL定義をContainer Orchestrationのレシピで記述する方法を提案
その記述に従いコンテナ（Microservice）をデプロイするOrchestratorをARMが試作

・ただし、RTOSでのコンテナ機能は未サポートなので試作は全てUbuntu
従って機能安全認証は取れず、機能安全課題はは実装者側で考える

・トヨタのWoven (Arene)、VWのCARIAD (VW.os)がリファレンスソフトウェアアーキテクチャ
として支持

Cloud Native Service
先行試作 (2019)

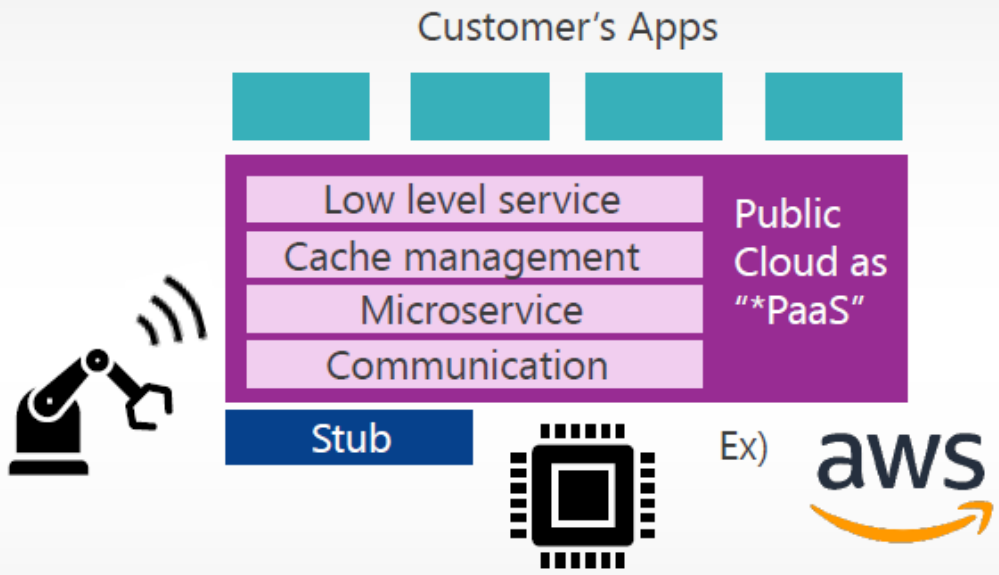
AOS

Customer's expectation to Renesas on cloud native as of today

CLOUD NATIVE
DEPLOYMENT OF INVOLVING CLOUD TECHNOLOGIES

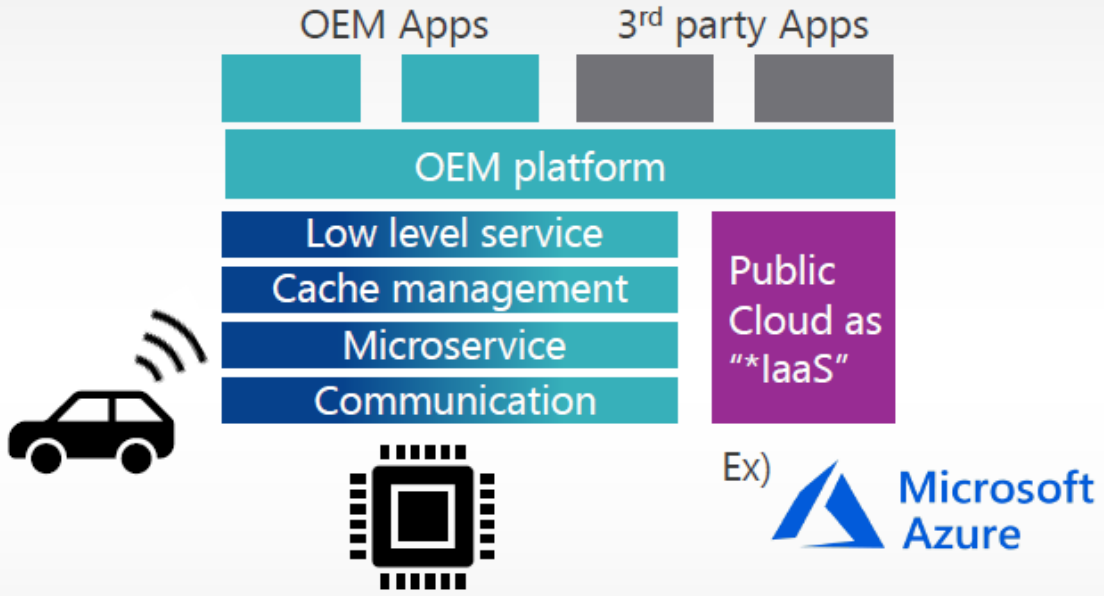
Others
Cloud vendor
Customer
Renesas

Industrials (MCU Based)



Customer: Want to "**ADAPT" DX**.
Renesas : Deploy Cloud Native SW of IT Vendors.

Automotive (SoC Based)



Customer: Want to "**BE" SW platformers**.
Renesas : Support Development of Cloud Native SW.

AOSを使ったIVIアップデートのデモ事例

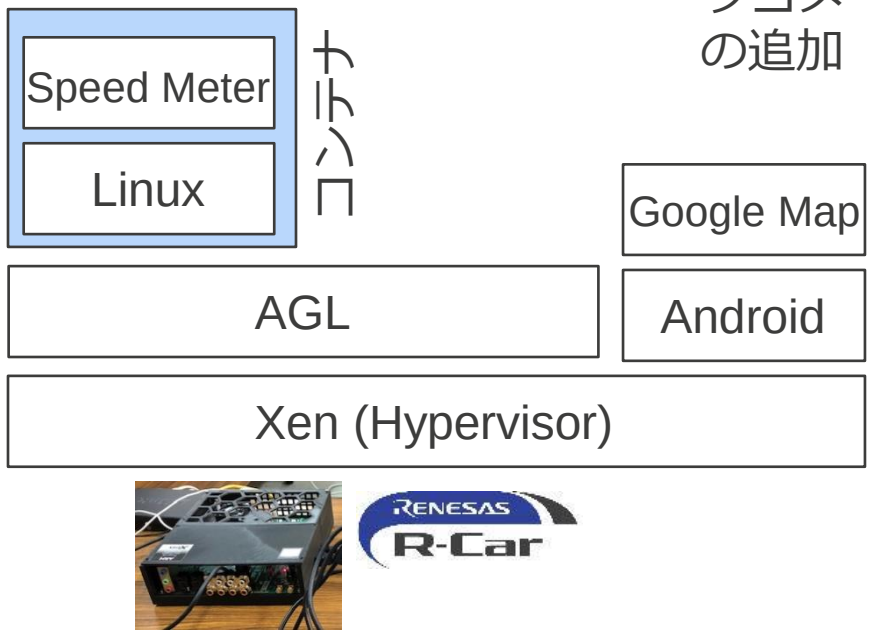
Google Map

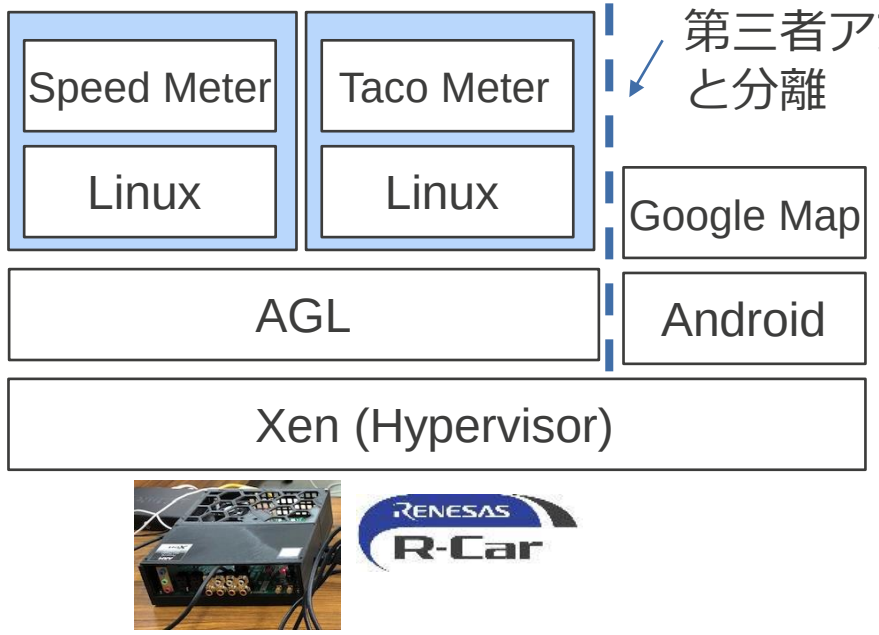
スピードメータ表示

Renesas-EPAM Cloud (AoS Cloud)

タコメータの追加

タコメータ追加





Androidがさらされる第三者アプリ脆弱性と分離

IPMMU, Bandwidth Reserve Mechanism　（半導体における差別化）

IPMMU:
CPUが利用するメモリを、ソフトウェアごとに独立性を担保して相互不可侵にする技術が
MMU（Memory Management Unit）
CPU以外の半導体IP向けに、メモリをソフトウェアごとに独立性を担保して相互不可侵に
技術をIPMMUと呼ぶ

Bandwidth Reserve Mechanism：
ソフトウェアのタスクスイッチに対して、バスやネットワークでの通信帯域をハードウェア的に
予約して、同時動作のソフトウェアが増えても、通信帯域を保証する技術

Software Firstになりソフトがハードのくびきから解放されることを指向すると、上記のハードウェアによる支援と、それをOS層で使えるようにする基本ソフトウェアが重要となる。

自動車業界のビジネスは100年に一度の大変革を迎えています。

既にSoftware Firstを実践しているTeslaの存在も、トヨタやVWに取り、大きな脅威です。

車載半導体をビジネスの柱として持つルネサスも、このSoftware Firstへのお客様の変化に乗らないと、クラウド・AIで先行するQualcomm、nVIDIAの後塵を拝し、業界から撤退する危機感を持っています。（だから私の転職を受け入れてくれました）

日本全体のビジネス強化のため、今後も技術と事業の両面でルネサスは、皆さまに貢献します。

To make our lives easier

by complementing human capabilities