Automotive SDN Architecture for Enabling SDV

Japan Automotive Software Platform and Architecture

2023 IEEE SA Ethernet & IP @ Automotive Technology Day

JASPAR Next Generation High-Speed Network WG

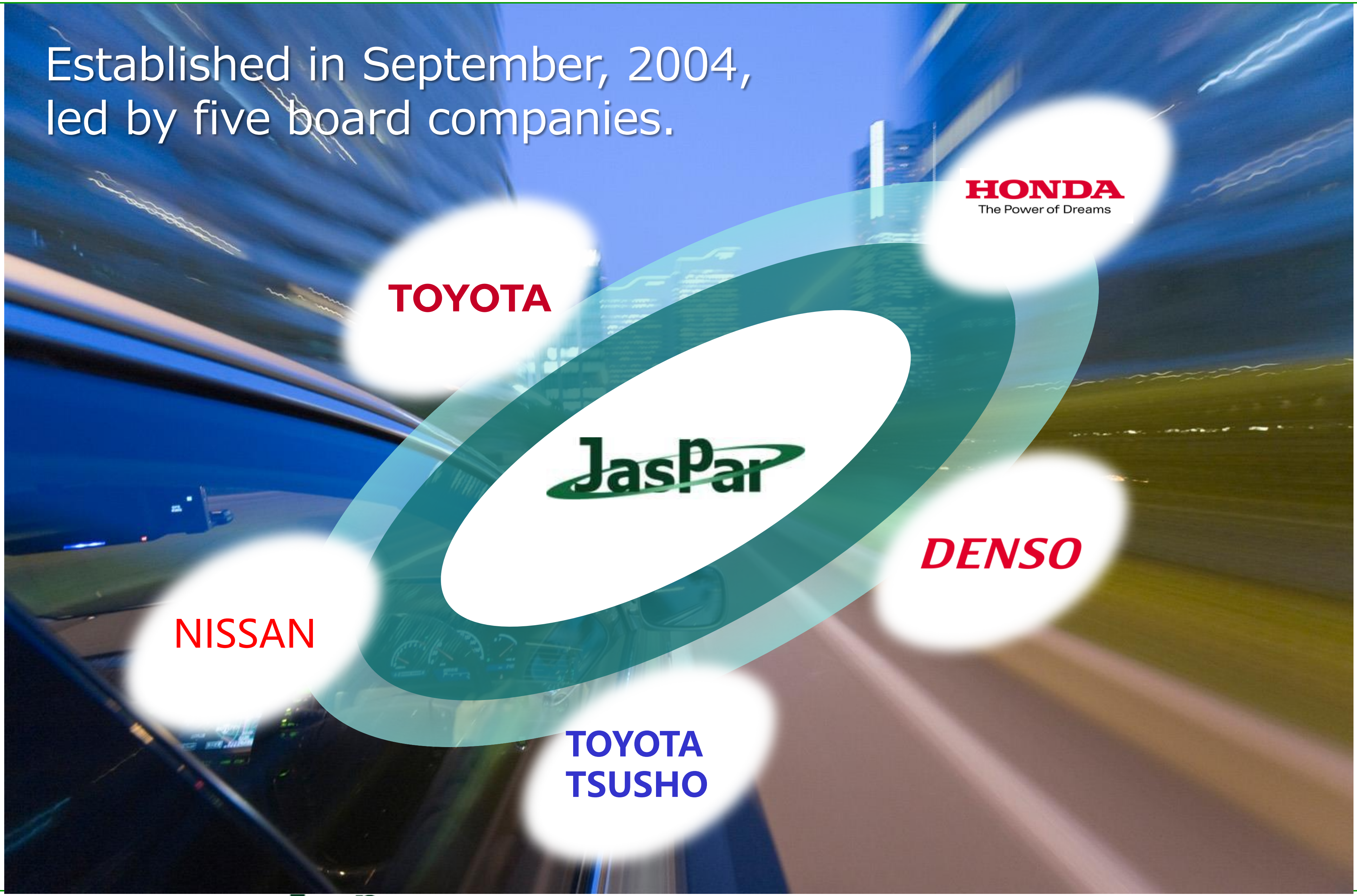
Tatsuya Izumi, Sumitomo Electric
Yoshihiro Ito, Nagoya Institute of Technology
Takumi Nomura, Honda
Yasuhiro Yamasaki, Toyota
Hideki Goto, Toyota

JASPAR, General incorporated association

Agenda

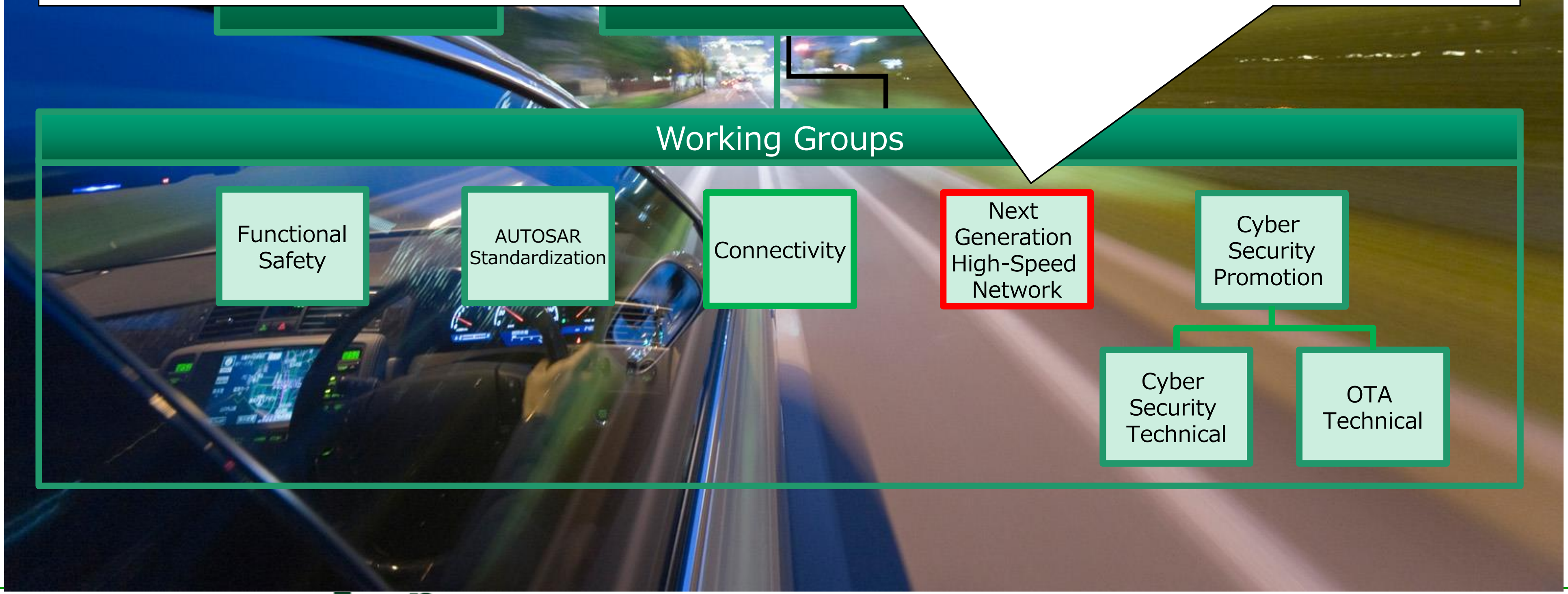
1. Introduction
2. Background & Objectives
3. A study of Automotive SDN
    - JASPAR's automotive use case & roadmap
    - Functional Requirements for Automotive SDN
    - Architecture (Logical & Physical)
4. Future works (Verification by PoC)
5. Conclusions

Introduction: About JASPAR



Next Generation High-Speed Network Working Group

To define standard specification of high reliability technology of in-vehicle high-speed networks with an eye focused on control system applications, and to define vehicle requirements/problem extraction and solution method of **Automotive SDN (Software Defined Networking)**, Automotive TSN, 10Gb/s class Ethernet and SerDes.



Introduction: 2 presentations from JASPAR

All layers are covered

OSI REFERENCE MODEL LAYERS	
7	APPLICATION
6	PRESENTATION
5	SESSION
4	TRANSPORT
3	NETWORK
2	DATA LINK
1	PHYSICAL

Next Generation High-Speed Network Working Group

Next Generation Technology Study Team

WG is composed by 4 teams and 4 sub-teams

- ***SDN Study & Verification Team***
- Next Generation Technology Study Team ***TSN Sub-Team***
- ***Time Synchronization Study & Verification Team***
- ***Hardware Team (Circuit Sub-Team, W/H Sub-Team)***
- Next Generation Technology Study Team ***Next Gen. Physical Layer Sub-Team***

2 presentations from JASPAR

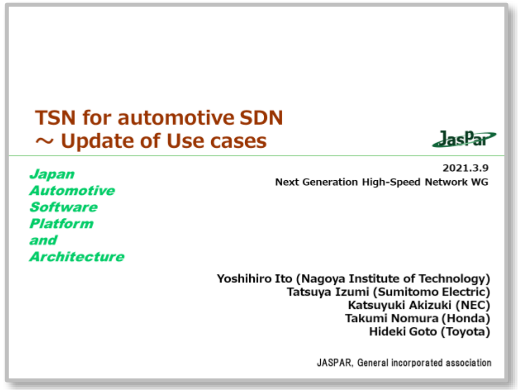
Automotive SDN
Automotive SDN Architecture for enabling SDV

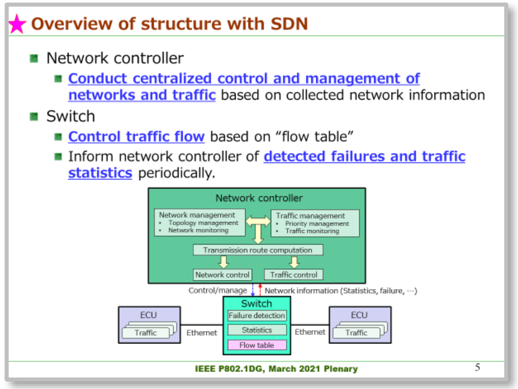
Ethernet TSN
Requirement of Harmonization among Automotive Ethernet Standardizations

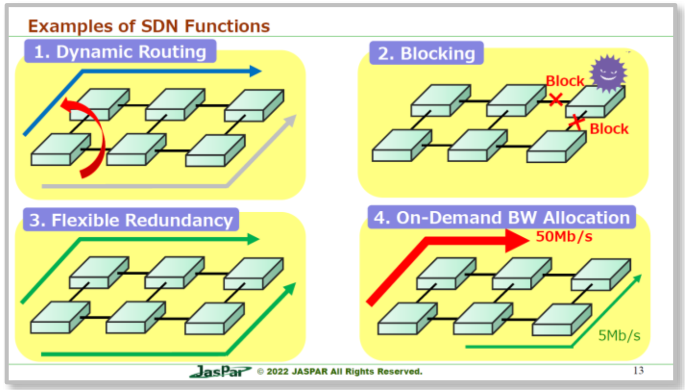
Team Composition of Next Gen. High-Speed Network WG

Background: JASPAR's efforts to date for Automotive SDN.

- To realize **the dynamic configuration of in-vehicle Ethernet**, JASPAR has advocated **the concept of in-vehicle SDN** since 2017.
  - Ex. 1) Change to a communication path that avoids the failed one
  - Ex. 2) Network reconstruction that can support functional extension through OTA
- A study of **JASPAR's use cases is proceeding** under the leadership of OEM.
  - Proposed Automotive SDN to IEEE P802.1DG as JASPAR's new use case (2021)
  - Presented Automotive SDN at Ethernet & IP @Automotive Technology Day (2022)







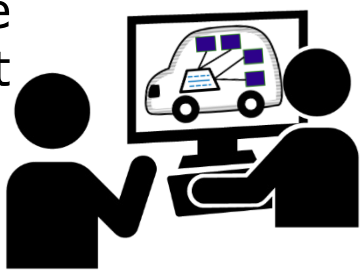
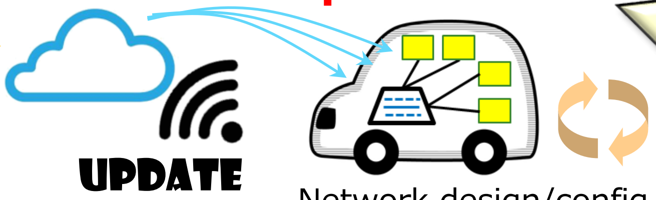
Y. Ito, et al. "TSN for automotive SDN Update of Use cases," IEEE P802.1DG contribution, Mar. 2021.
T. Nomura, K. Akizuki, et al. "Proposal of Dynamically Configurable In Vehicle Network as an Enabler of Software Defined Vehicle," IEEE-SA EIP ATD 2022.

JASPAR was the first in the world to raise the concept of Automotive SDN, which it had envisioned for some time.

► **JASPAR wants to start considering the architecture with it taking the lead.**

Background: SDV (Software Defined Vehicle)

SDV : Vehicles that can continue to evolve with a software update

	Before（not SDV)	After（SDV)
Usage of vehicle	No function updates after shipping. 	**Functions are updated daily.** 
Network	Implement a fully equipped design in the development stage 	**Redesign and reconfiguration are required in conjunction with function updates.** 

The software will play a leading role in car making.

The hardware should be ready for future updates.

Enable high-frequency redesign and reconfiguration

To realize SDV, in addition to a flexible software platform such as OTA, hardware and networks must have a mechanism that enables functional updates.

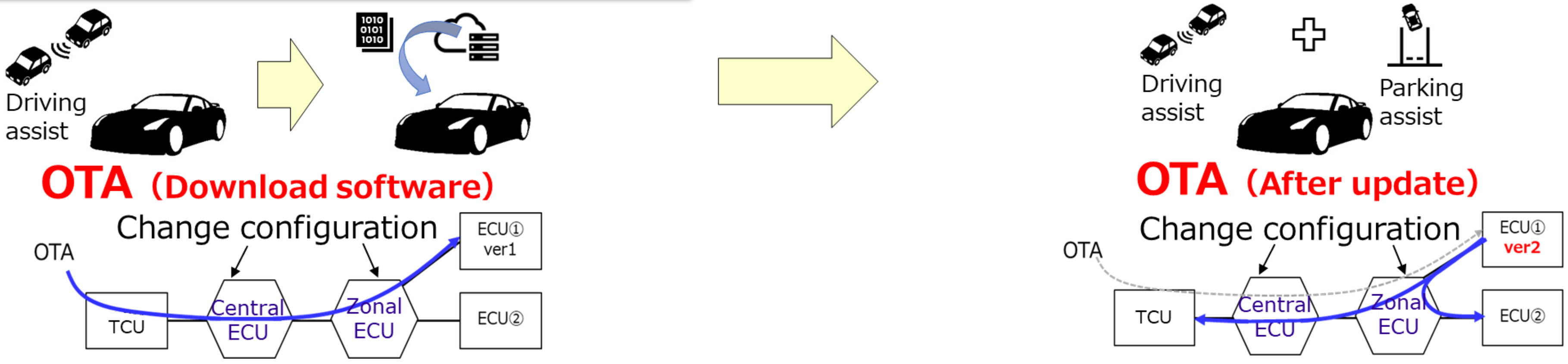
Target of this presentation

Agenda

1. Introduction
2. Background & Objectives
3. A study of Automotive SDN
   - JASPAR's automotive use case & roadmap
   - Functional Requirements for Automotive SDN
   - Architecture (Logical & Physical)
4. Future works (Verification by PoC)
5. Conclusions

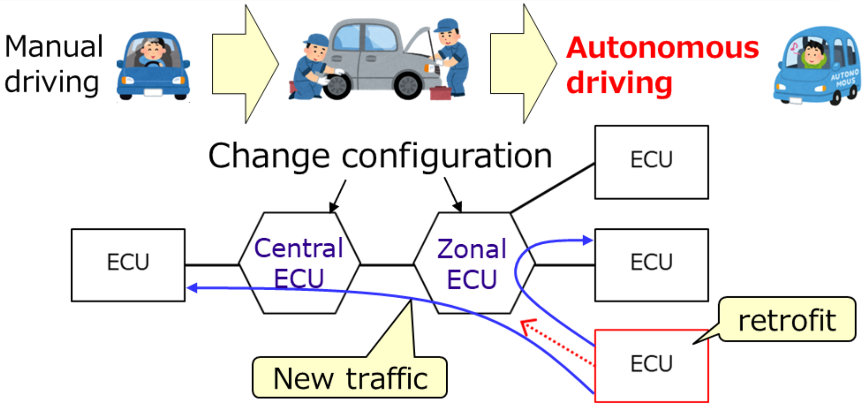
Use cases where SDN can be expected to apply

Software update（OTA）



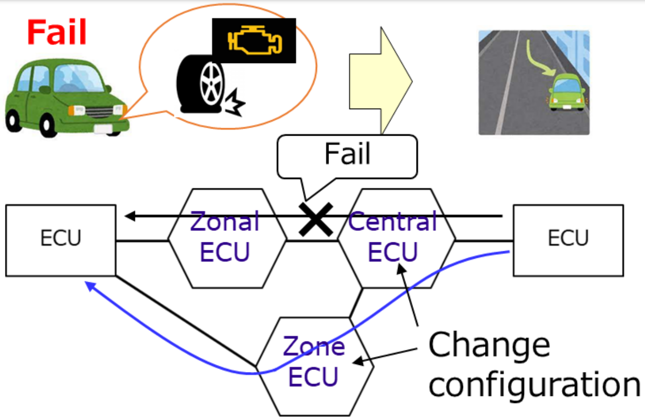
Shorten download time by temporarily changing configuration. **Optimize configuration as traffic amount changes.**

Hardware update（Plug & Play）



Allow new traffic by updating flow table.

Fail-operational system

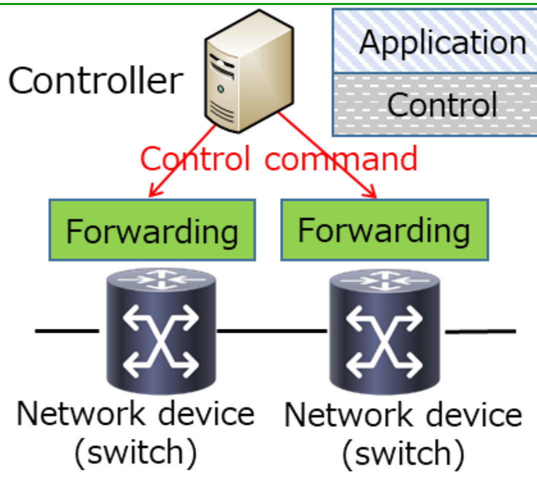


Avoid communication failure by rerouting.

Since SDN has a high affinity with the evolution of SDV, it will significantly increase the added value of cars.

Trends in existing technology (ICT field)

- Until now, setting configuration and updating have been carried out for each network device (each switch).
- SDN that can **dynamically update configuration** through centralized control is gradually being put to practical use.



*Current main applications: data centers and telecommunications carriers

- Advantage
  - Reduced workload (controller change only)
  - Quick configuration change
- Disadvantage
  - High processing load on the controller
  - Communication between the controller and switches is mandatory.
- Example of SDN protocol implementation
  - OpenFlow (not an L2 switch)

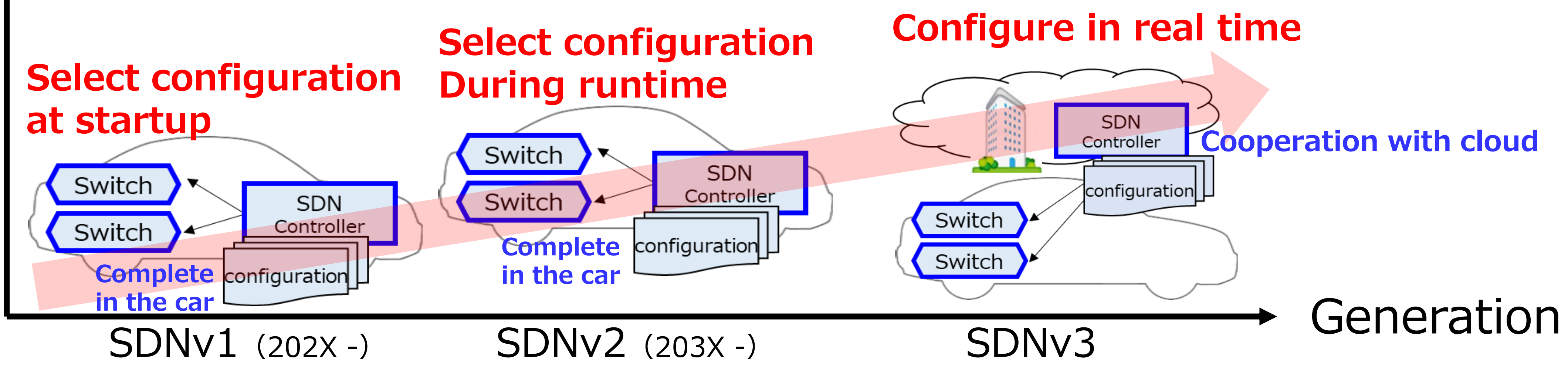
Reliability	ProvideL2/L3 connectivity service Guarantee of Quality of Service (QoS)
Scalability	Network virtualization • Separation of logical and physical configuration • Separation of addresses • Isolation of traffic • Virtual Machine (VM) Mobility
Availability	Automatic provisioning Service detection Operation, Administration, Management

The concept of SDN is useful but it isn't easy to apply the technology of the ICT field directly to vehicles.
► JASPAR is considering an architecture suitable for in-vehicle use.

JASPAR's Automotive SDN Roadmap

Function

Adapt to updates (software/hardware)	+Adapt to change of vehicle status (APP. ON/OFF, Link failure, etc.)	+Cooperates with environment outside vehicle (Smart City、V2X, etc.)
Select from preset configurations		Configure by dynamic calculation
Stopped (at startup)	During runtime (vehicle speed is zero)	



In the future, it would be ideal for updating dynamically and in real-time, like in the ICT field, but operations will be as static as possible for the time being.

How to proceed with architecture study

Requirements analysis, requirements definition

1. Analysis of upper requirements
   (SDV as conceived by JASPAR)

⇩

2. SDV ▶ **Requirement analysis for in-vehicle SDN**

⇩

3. Determine the assumed scenario from typical U/C

⇩

4. Define basic requirements (common items)
   for in-vehicle SDN from multiple assumed scenarios.

⇩

Architecture study

5. Analysis of current architecture
   (In-vehicle or ICT field)

⇩

6. **Study on in-vehicle architecture** (JASPAR proposal)

⇩

7. **Architecture embodiment** (SDNv1 as an example)
   Requirements study ▶Logical Arch. ▶Physical Arch.

Define basic requirements of in-vehicle SDN from the upper-level requirements (SDV), and study the in-vehicle architecture suitable for the basic requirements.

Image of SDV defined by JASPAR

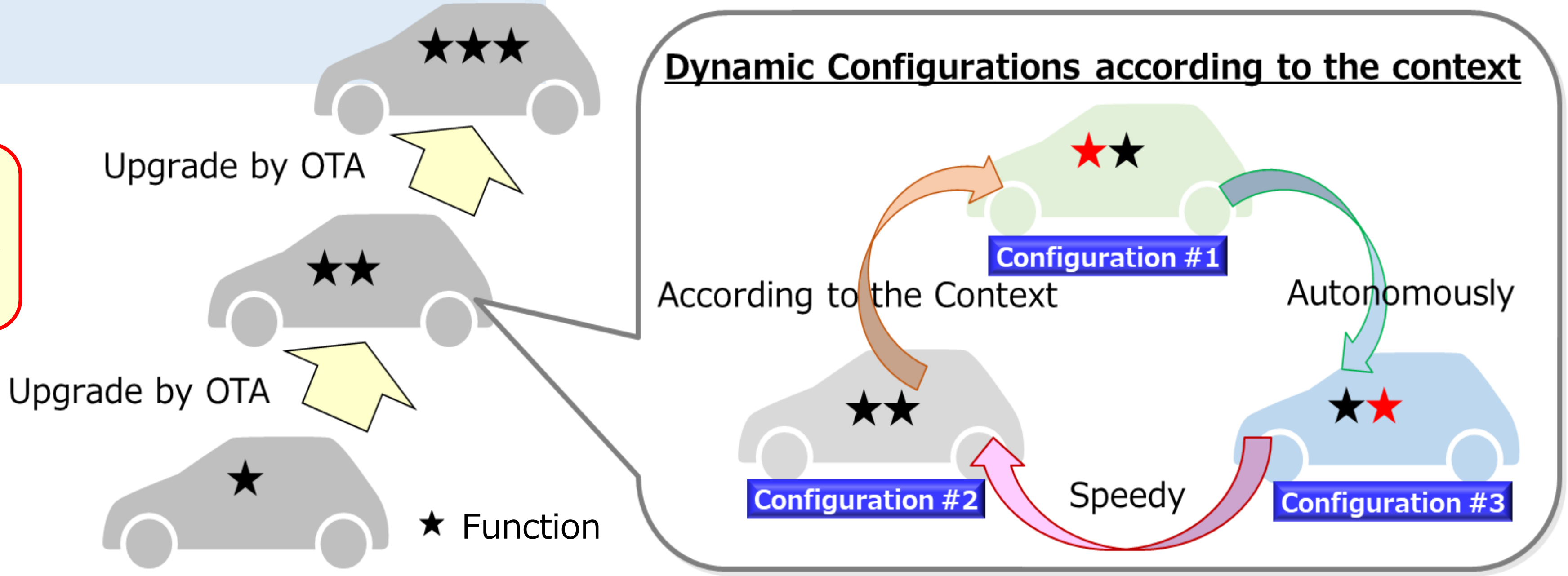
- **In addition to being reprogrammable by OTA, the car can**
  - automatically and instantly
- **switch configurations**
  - according to the user's request and context
  - without software reprogramming nor necessarily being connected to the cloud

For networks, it is equivalent to "**switch network configuration.**"

Here, context means the state where the vehicle, driver, etc., are placed; it can change frequently and reversibly.

Examples:
- Enable/turn off the functions depending on the driver,
- Activate/deactivate the functions according to location and time

  etc.

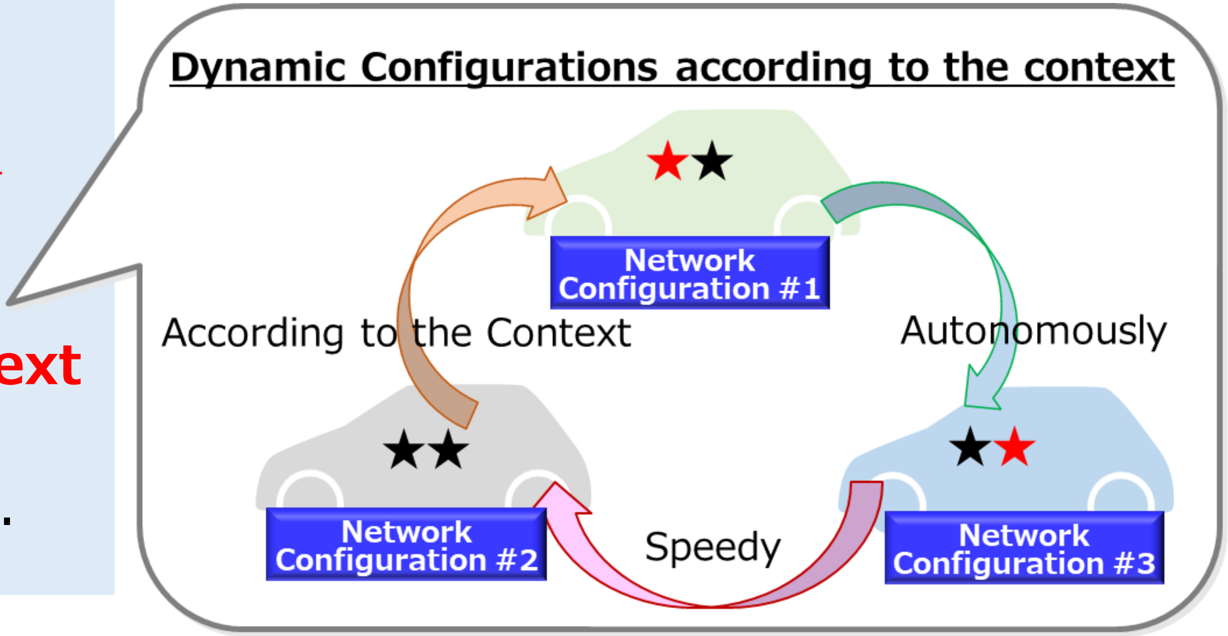


Requirement analysis: Requirement analysis for automotive SDN

Requirements for automotive SDN
(from the JASPAR's SDV image)

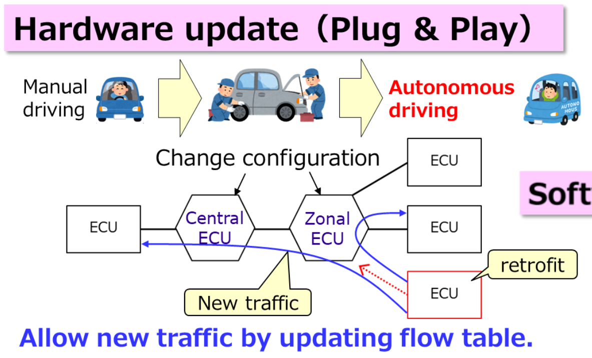
A mechanism that can switch (network) configurations

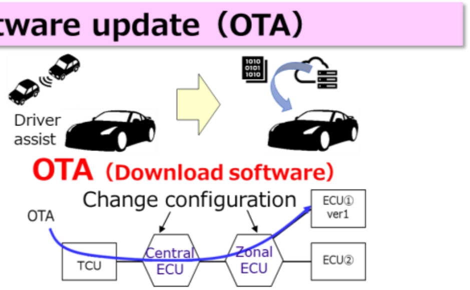
- automatically and immediately
- according to **user requests** and **context**
- without reprogramming software
  or necessarily connecting to the cloud.



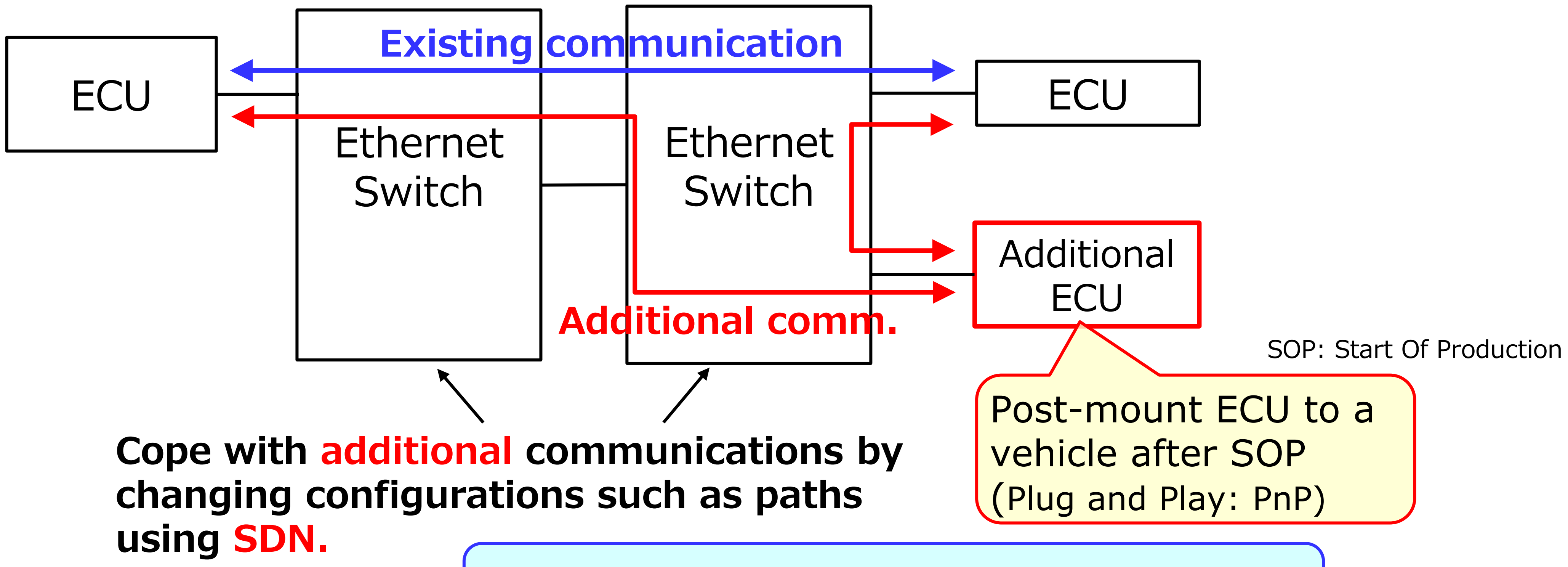
Study requirements by applying them to concrete use cases (scenarios)

- Plug and play (change configuration after adding ECU)
- OTA (change configuration after software update)
- OTA (temporary change configuration during software download)





Example of assumed use case (Configuration change after Plug and Play)



Detail of "Switch configurations according to user requests and context"

<Assumed scenario>

I. Connect an additional ECU to a free port **(Trigger for the start of SDN)**

II. **Collect** topology/application **information** of the whole network.

III. Based on the information, **derive** an appropriate network **configuration**.

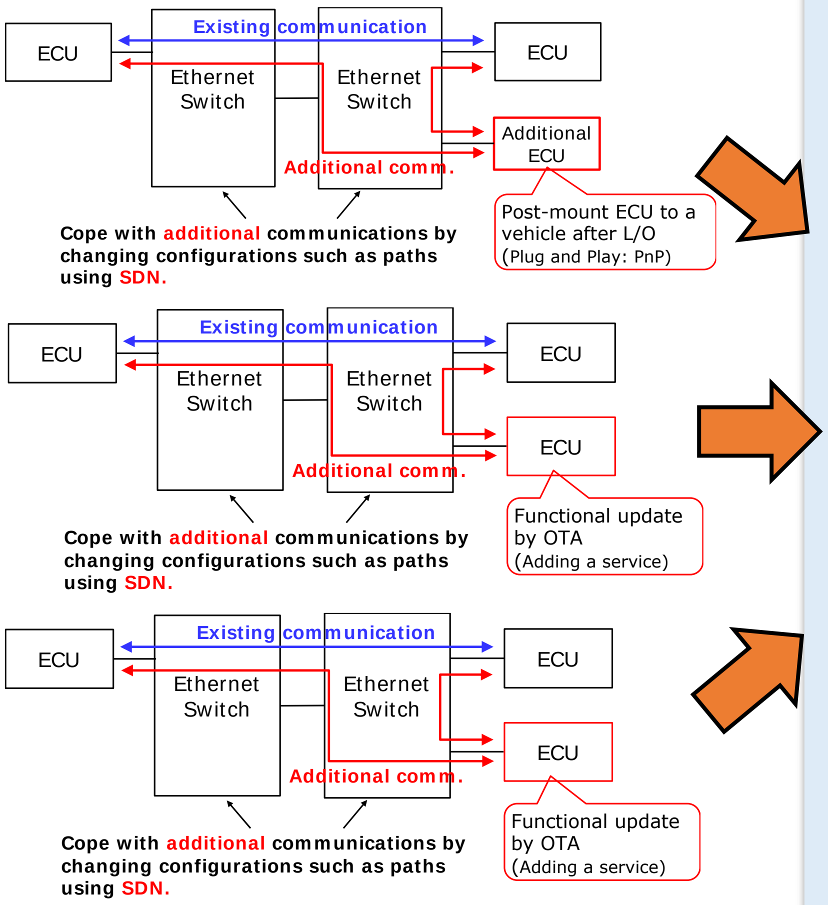
IV. **Apply** the new **configuration** to each Ethernet switch.

V. A new communication path is established between the existing ECUs and the additional one.

Note that the blue parts are common and do not depend on the use case.

Definition of Requirement: Study on SDN requirements

Assume use cases



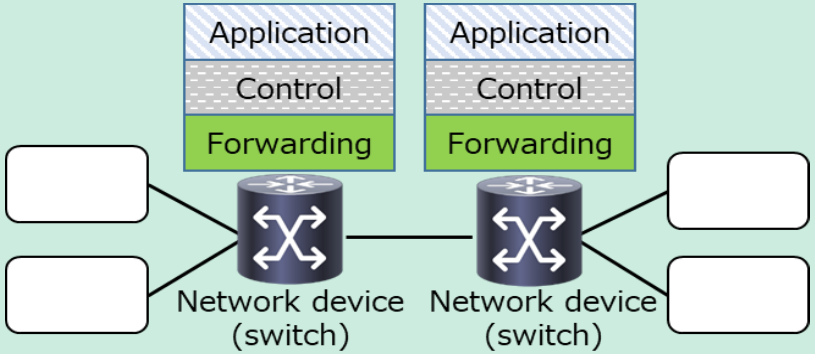
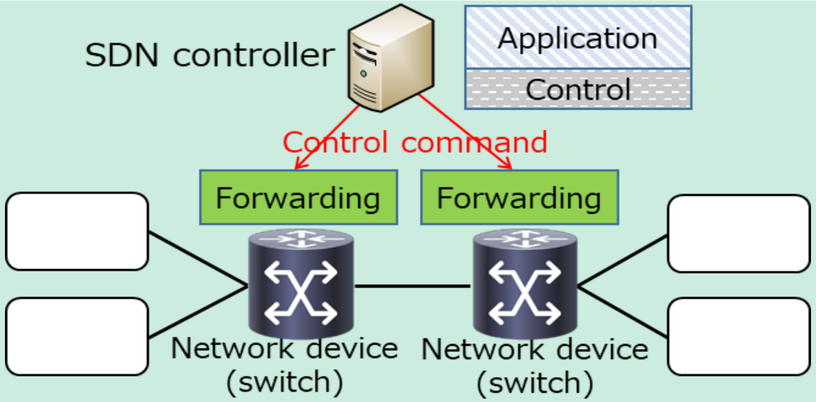
Basic requirements for automotive SDN

(... requirements)

- **Trigger acceptance**
  - Function to detect trigger of configuration change
- **Information collection/Configuration derivation**
  - Function to derive appropriate network configuration according to network topology (Connection status)
  - Function to derive appropriate network configuration according to application demands
- **Reflection of configuration**
  - Function to reflect the derived network configuration to the Ethernet switches.

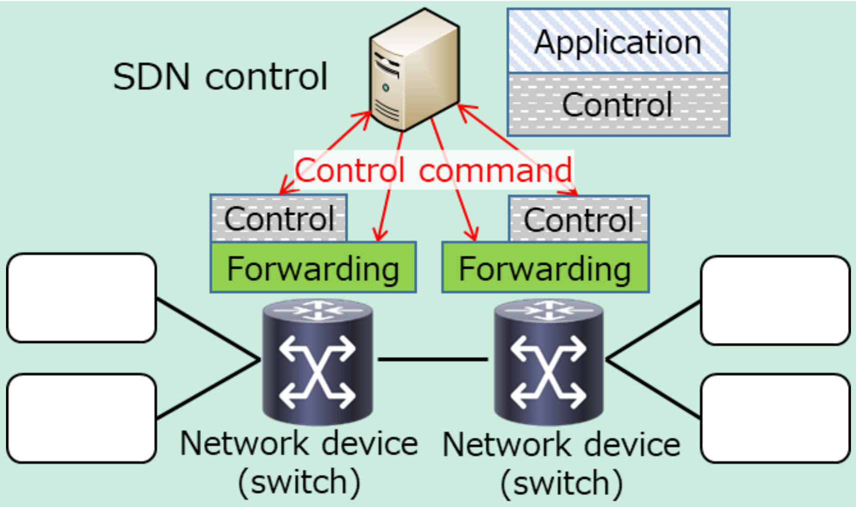
Define basic requirements for automotive SDN based on the common requirements of the assumed use cases

Analysis of existing architecture (Automotive/ICT field)

	Conventional in-vehicle network devices (Non-SDN switches)	SDN-supported network devices in the ICT field (e.g., OpenFlow)
Architecture		
Features	**Autonomous distributed network control** • Complete controls within each switch(SW)	**Centralized network control** • Separate control function and data transfer function • The SDN controller collects information from the whole network and **centrally controls** SWs.
Basic actions	• Operates based on **fixed network(NW) configuration** input during SOP	• At receiving an unknown flow, SW asks the controller. • The controller **derives new NW configurations (flow table) by dynamic calculation**. • SW sets the configuration at the controller's request.
Issues	• Works with no controller (brain) and fixed configuration (multiple SWs work individually) • All switches need to be **reprogrammed** every time the NW configuration is changed.	• Since the **processing load on the controller is large**, it is unsuitable for automotive use. • **No automotive SW-IC** supports OpenFlow. (HW processing is mandatory for satisfying performance.)

It is necessary to consider an architecture suitable for automotive use to deal with the above issues.

Study on automotive architecture (JASPAR's proposal)

	Hybrid architecture (assuming in-vehicle SDNv1)
Architecture	
Feature	**Combine distributed and centralized network control**
Basic actions	・Hold **multiple NW configurations** during SOP. [Distributed] ・Normally work with preset **fixed configuration**. [Distributed] ・Controller **selects a new configuration** based on an external trigger and collected information. [Centralized] ・SWs set the config. by controller's instruction. [Centralized]
Delight	・Reduction of processing load on the controller ・Reduction of reprogramming for configuration change

An architecture that keeps the advantages of centralized control and compensates for its disadvantages by combining distributed one

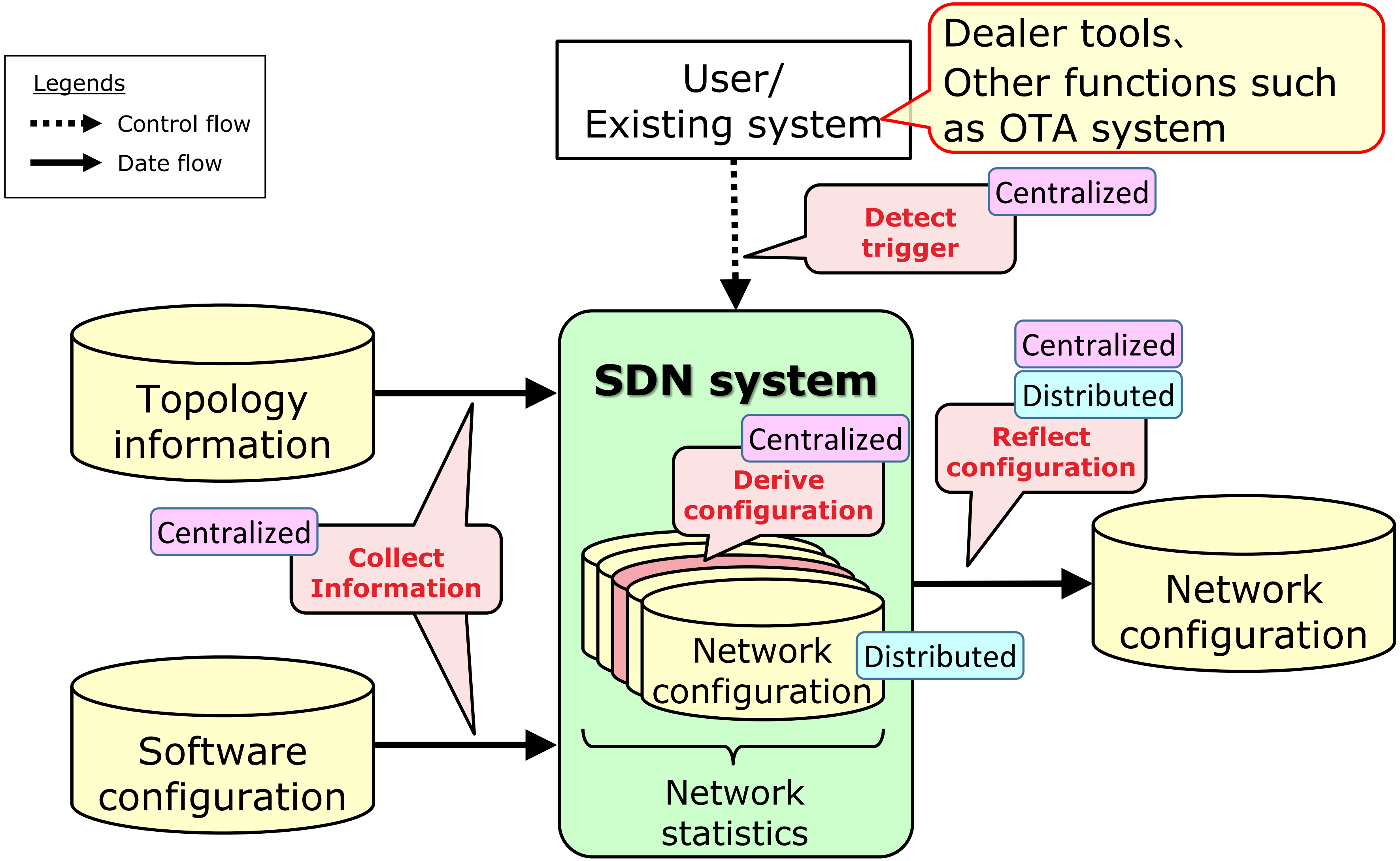
Realization of Architecture: Study on Requirements for SDNv1

Software: Reprogramming/OTA
Hardware: Plug and Play

Basic requirements	Demand	Function requirements	Corresponding architecture: Distributed	Corresponding architecture: Centralized
Trigger acceptance	**Ready for updates (Software/Hardware)**	• Function to detect SDN execution triggers or receive SDN execution requests from other functions		○
Information collection	**Ready for updates (Software/Hardware)**	• Function to collect **topology information** and **software configuration**, or receive them from other functions		○
Configuration derivation	**Select from preset configurations**	• Function to store **verified configurations** in advance	○	
		• Function to select **appropriate configuration** based on the combination of collected topology information and software configuration		○
Configuration reflection	**Stopped (at startup)**	• Function to request configuration changes to switches		○
		• Function to save the selected configuration in a non-volatile area	○	
		• Function that can read the new configuration at the next startup (IG: OFF to ON)	○	

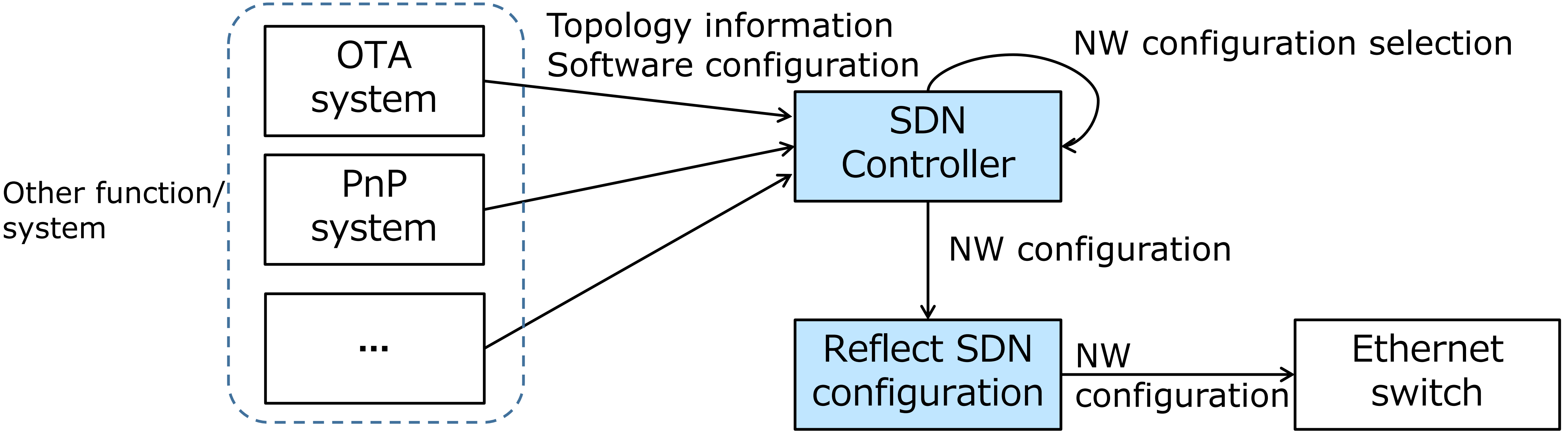
Subdivide functional requirements into more detailed ones based on requirements for SDNv1

Realization of Architecture: Role of SDN System (an Example)



Configure the SDN system based on the functional requirements on the previous slide (Concretion of roles)

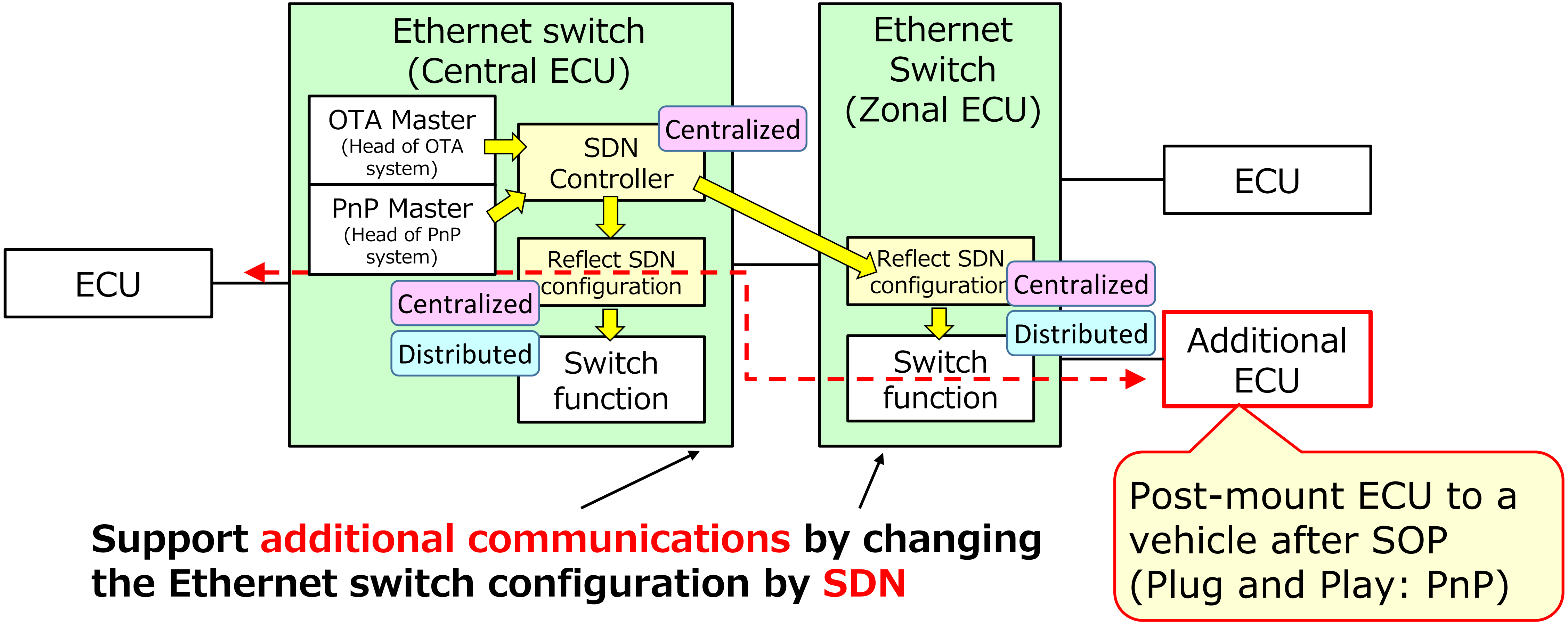
Materialization of Architecture: Logical Architecture of SDNv1



Function	Overview	Distributed	Centralized
SDN controller	Select an appropriate configuration according to the network status and instruct the SDN configuration reflection function to change the switch-related parameters.		O
SDN configuration reflection	Change switch-related parameters according to the requests from the SDN controller function.	O	O

SDN controller function and SDN configuration reflection function are defined as inevitable functions.

Materialization of Architecture: Physical architecture for SDNv1 (an Example)



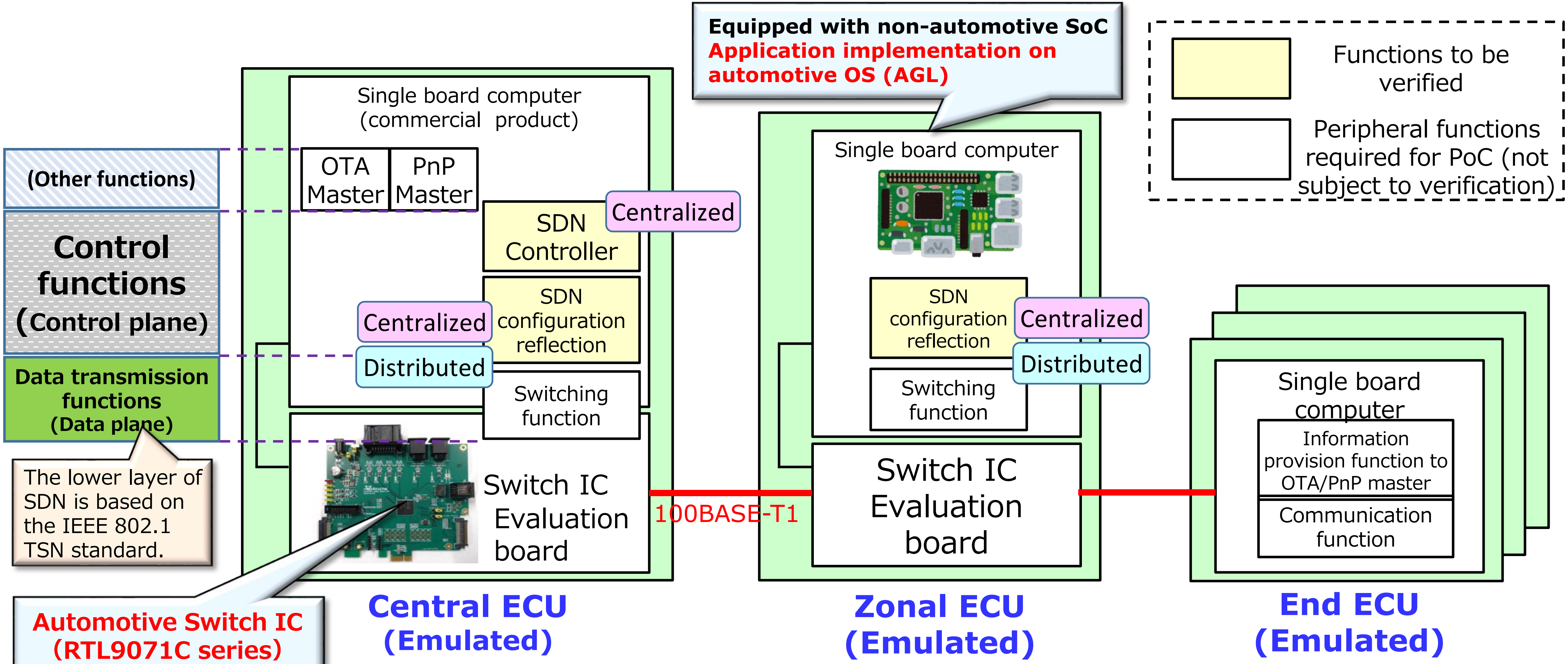
Support additional communications by changing the Ethernet switch configuration by SDN

In the case of central & zonal architecture, the controller function is placed in the central ECU, and the network configuration of the zonal ECU is centrally managed. (The switching function of each zonal ECU works autonomously.)

Agenda

1. Introduction
2. Background & Objectives
3. A study of Automotive SDN
   - JASPAR's automotive use case & roadmap
   - Functional Requirements for Automotive SDN
   - Architecture (Logical & Physical)
4. Future works (Verification by PoC)
5. Conclusions

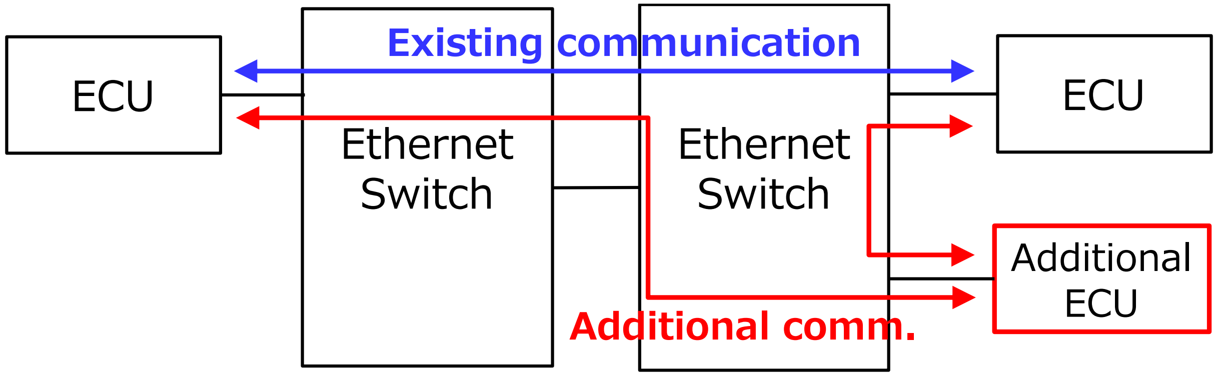
Configuration of PoC system (planned)

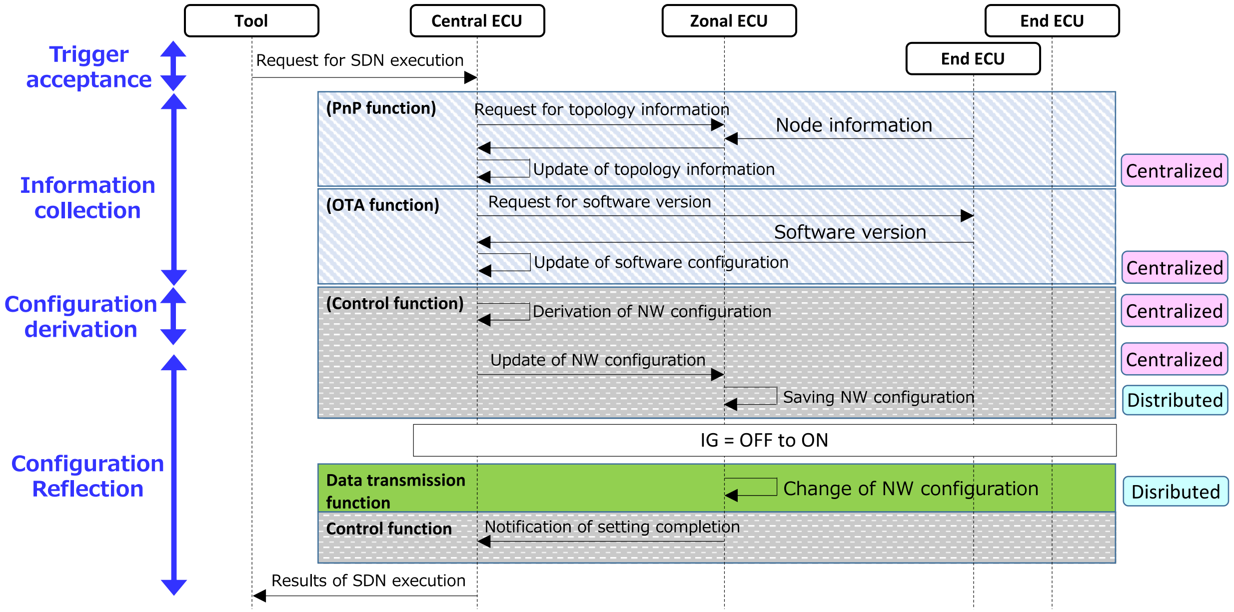


To quickly verify the feasibility of the architecture, it consists of a combination of commercial boards.
(Automotive products are selected for the vital switch IC.)

Evaluation plan

- Evaluation items (draft)
  - Verification of a flow of functions
    - From Trigger acceptance to Configuration reflection
    - Static/Dynamic configuration of switches
  - Processing time
    - Latency, configuration time, etc.
  - Amount of traffic for SDN control
    - Bandwidth usage of the traffic
  - Resources
    - CPU load
    - Amount of memory
  - ...





The results of our verification will be fed back to the functional requirements and architecture for SDNv1 and will be published as a JASPAR standard document (March 2024).

Conclusions

- To realize SDV, dealing with the trinity of software, hardware, and network is essential.
- Automotive SDN is a general term for technologies that can dynamically change the configuration and functions of networks; it is highly compatible with the evolution of SDV and can coexist with existing technologies (OTA reprogramming, etc.).
- It isn't easy to directly apply SDN technology in the ICT field to vehicles.
  ► JASPAR is studying architecture suitable for automotive use.
- This presentation proposes a novel hybrid architecture that combines the advantages of distributed network control and centralized network control.
- After proof by PoC, the results will be published as a JASPAR standard document (March 2024).

Thank you for your kind attention.