

End-to-End Connectivity Design with Automotive Ethernet & Service-Oriented Architecture

Wonseon Sim

Hyundai Motor Group

Seung Jun Lee

AirPlug, Inc.

Contents

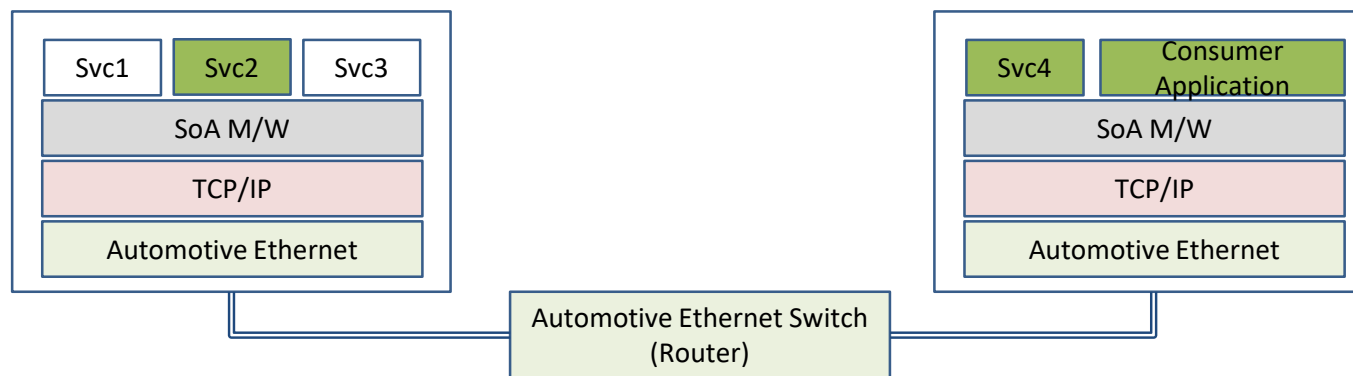
- **Service-Oriented Architecture**
 - Background
 - Benefits
- **End-to-End Service-Oriented Architecture**
 - Challenges & Approach
 - Overall Architecture
 - SoA Adaptor
 - SoA Gateway
 - SD Proxy & Service Router
- **Use Case**
- **Concluding Remarks**

Service-oriented Architecture

- Background
- Benefits

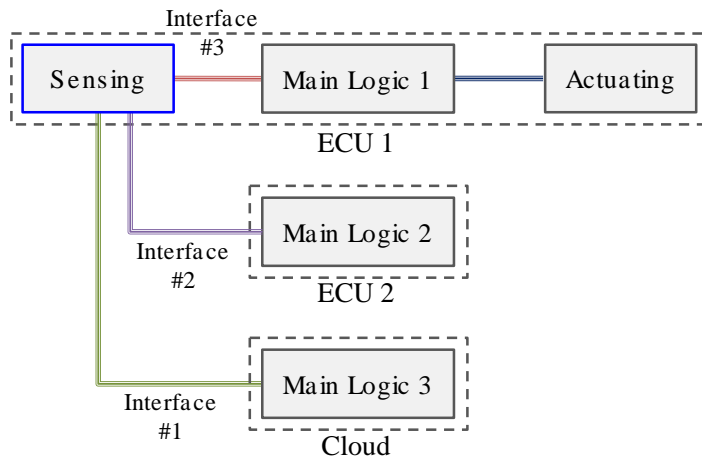
[SoA] Background

- Innovation of Automotive Network : **Automotive Ethernet & IP**
 - Faster & Unicast/Multicast/Broadcast Support
 - Logical & Dynamic Configuration
- Innovation also in System/Software area : **Service-Oriented Architecture (SoA)**
 - “Services” : reusable, remotely accessible, independently maintainable
 - An application can consume services regardless of service location.
 - Services provided through SoA Middleware running on top of Automotive Ethernet & IP.

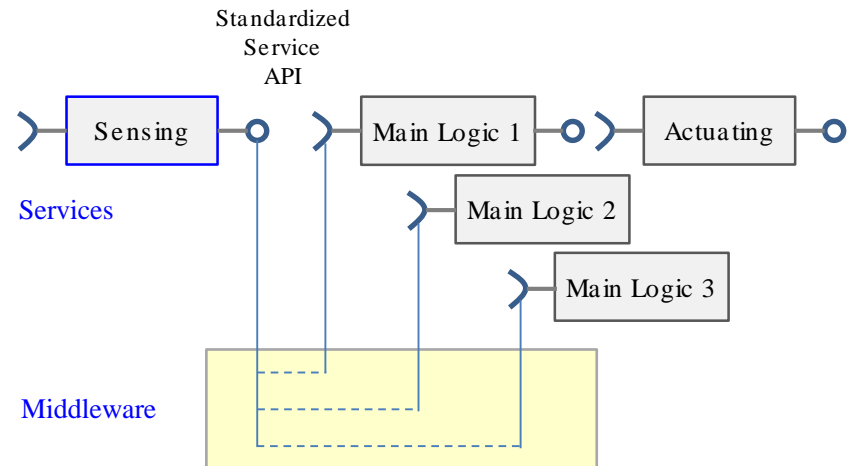


[SoA] Benefit

- Standardized Interface & Location Transparency
 - Reduced Cost for Manufacturing & Test
 - Reduced Time for Development of Vehicles & Connected Car Services
 - Efficient Handling of Models/Options
 - Extensibility



Without SoA



With SoA

End-to-End Service-oriented Architecture

- Challenges & Approach
- Overall Architecture
- SoA Adaptor / SoA Gateway
- SD Proxy / Service Router

[E2E SoA] Challenges

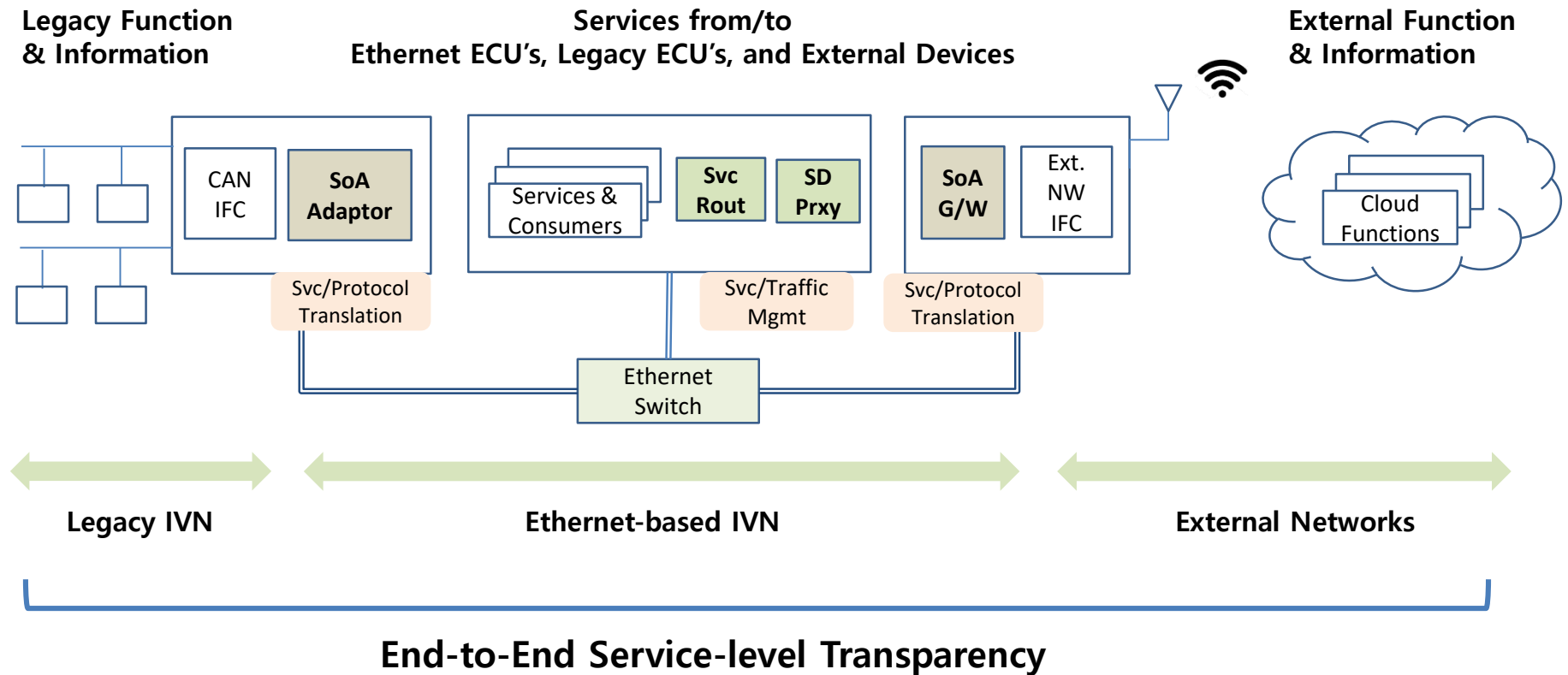
- **Legacy In-vehicle Network (IVN)**
 - Legacy IVN (e.g., CAN) will co-exists with Ethernet at least for a certain time period
 - SoA not directly applicable to these legacy IVN's
- Needs for Service-level **Interoperation with External Devices**
 - External network characteristic different from that of IVN
 - More Critical Security Issues
- Needs for efficient **Handling of Service-related Information**
 - Added and updated frequently
 - Should be accessible from many interested parties
- Security and Other **Issues** of SoA from the nature of **Distributed System**
 - Services should be found easily but only by allowed ECU's
 - Services should be accessed easily but only by authorized Service Consumers
 - Interoperability & Resource issues might arise.

[E2E SoA] Approach

- Legacy In-vehicle Network Issues
 - Information/Functions from Legacy IVN transformed to “Services” with **SoA Adaptor**
 - Needs for Service-level Interoperation with External Devices
 - Service-level transparency achieved with **SoA Gateway**
 - Security and Other Issues of SoA from the nature of Distributed System
 - Centralized Service Discovery (SD) architecture using **SD Proxy**
 - Security- or Resource-Critical Services made accessible through **Service Router**
- **Efficient and Secure End-to-End Service-oriented Architecture**

[E2E SoA] Overall Architecture

- End-to-End Service-oriented Architecture
 - Extended Service-level Transparency and Integrated Service Design

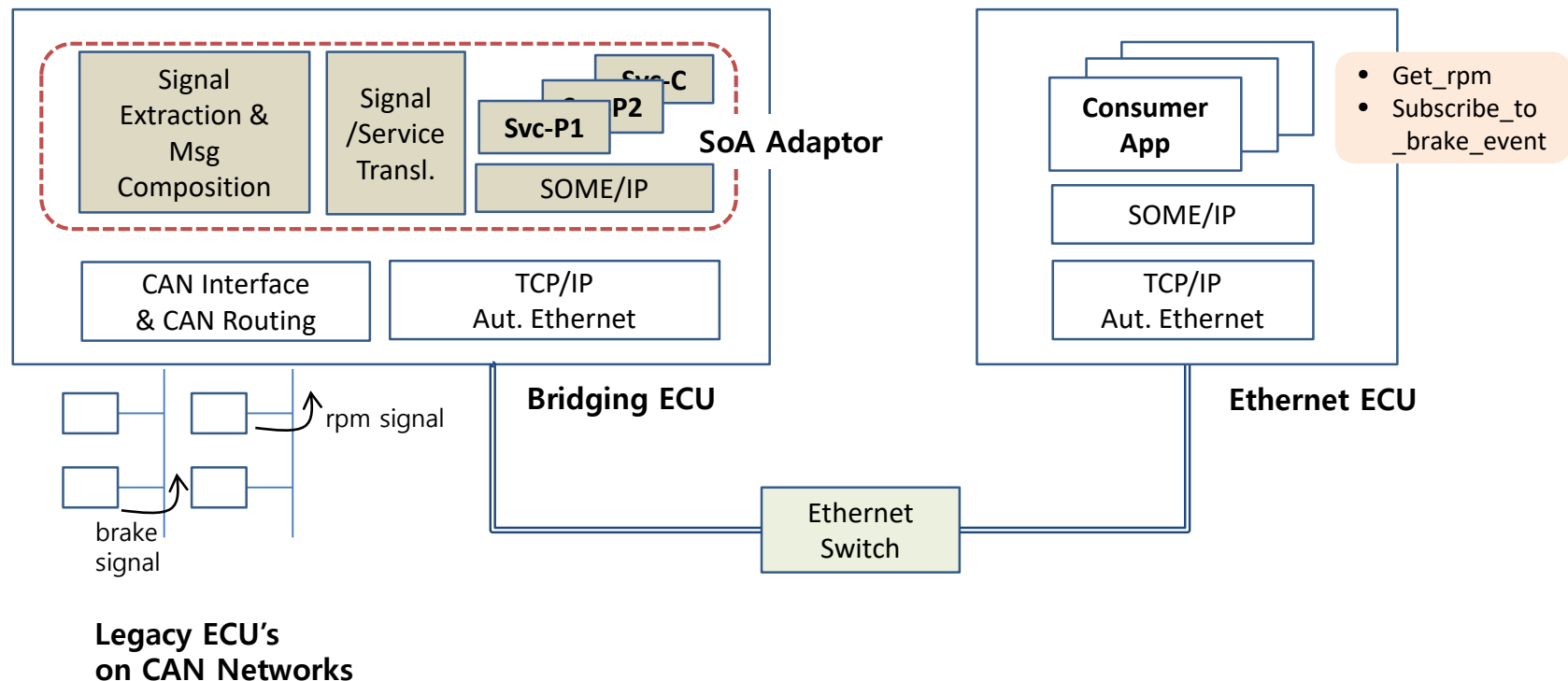


[E2E SoA] SoA Adaptor

- Legacy In-vehicle Network
 - Many ECU's still work based on legacy IVN like CAN
 - Large portion of vehicle information/functions are from legacy IVN
 - These should be made accessible to new applications on Ethernet-based ECU's.
- **SoA Adaptor**
 - Transforms Information/Functions from **Legacy IVN to "Services"**, which applications on any Ethernet-based ECU's can easily access.
 - On Ethernet side, services are provided on top of **SOME/IP protocol**.
 - Can be implemented on **"bridging ECU"** between legacy IVN and Ethernet-based IVN, like Domain Control Unit, Zone Controller, etc.
 - Also can be implemented on non-bridging ECU's only with Ethernet interfaces.
 - Services provided by SoA Adaptor can be changed dynamically.

[E2E SoA] SoA Adaptor

- (Example) SoA Adaptor on Bridging ECU for CAN Networks

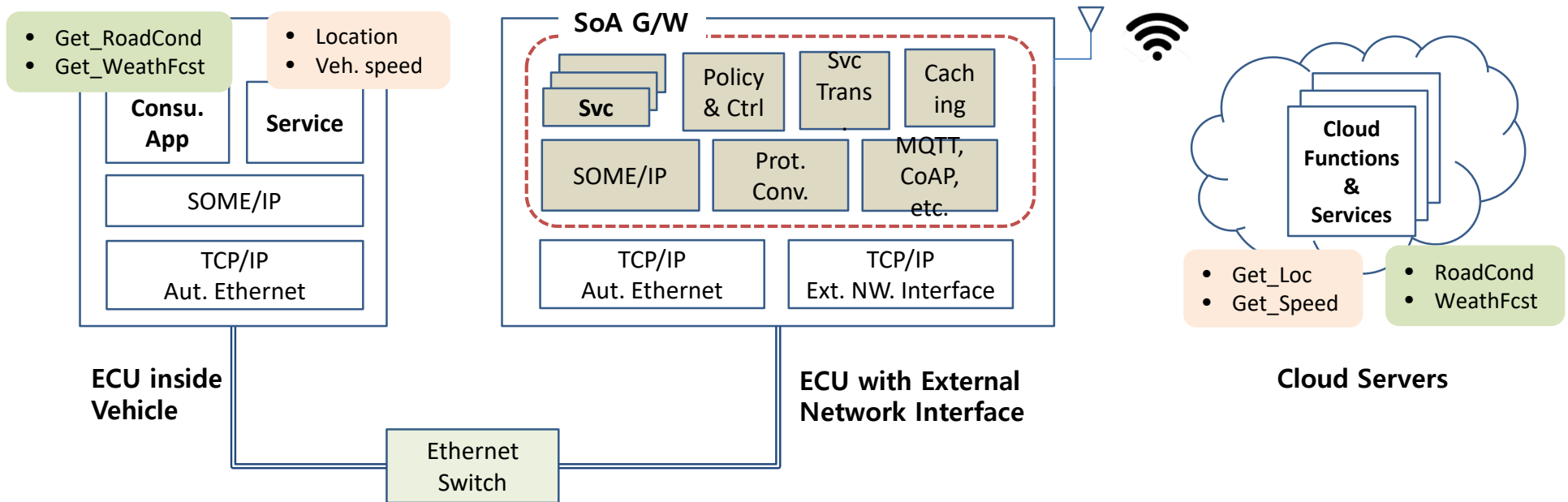


[E2E SoA] SoA Gateway

- External Devices on External Network
 - Vehicle needs to interwork with external devices like cloud servers and smart devices.
 - Interworking need gets much larger for ADAS and other connected car services.
 - External networks has very different characteristics compared to IVN
 - : availability, bandwidth, latency, cost, etc.
 - Protocols for external connectivity are usually different from those for IVN.
 - Higher security issues when interworking through external networks.
- SoA Gateway
 - Handles issues related with external device/network interworking.
 - **Converts Protocols and Translates Services**, when needed.
 - **Caches external information** to deal with availability & cost issues of external networks.
 - Applies **Policy** and Performs Service-level **Access Control**.
 - Should be implemented on ECU's with external connectivity.

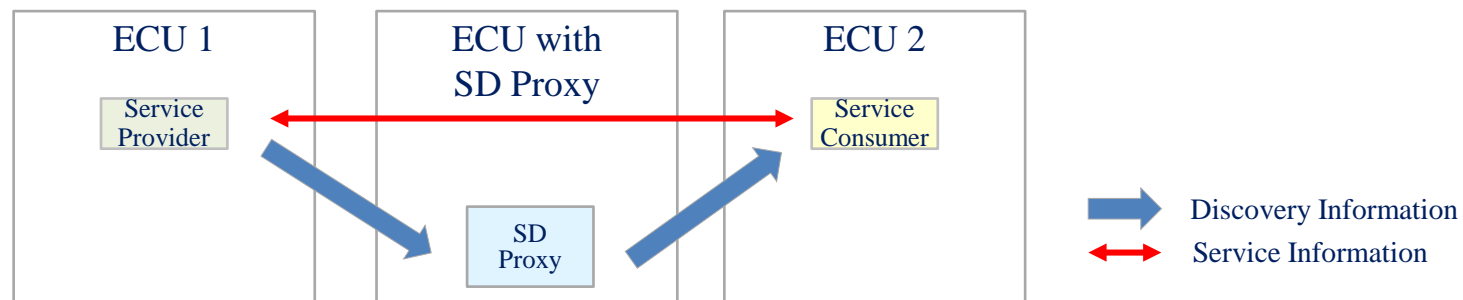
[E2E SoA] SoA Gateway

- (Example) SoA Gateway for Cloud Function Interworking



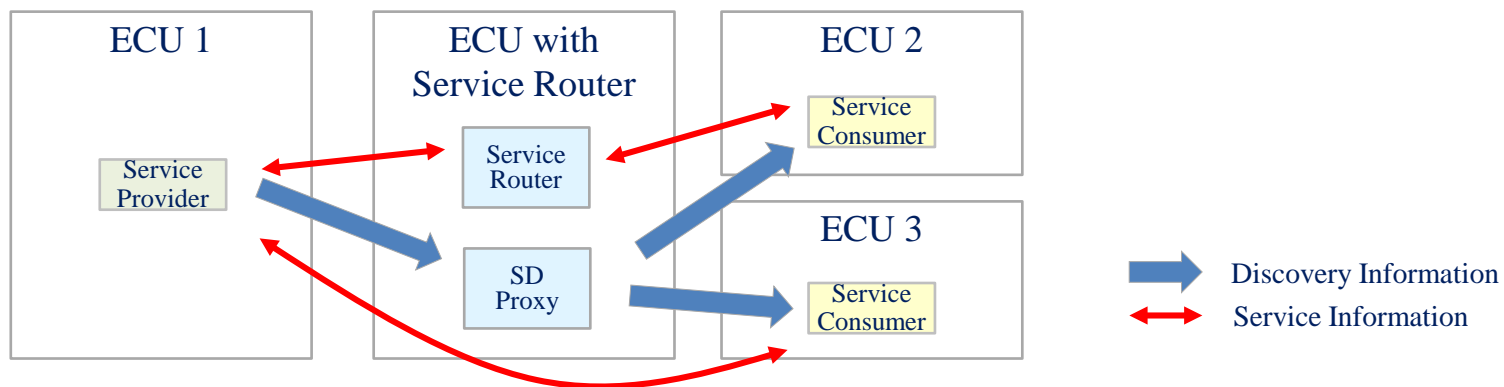
[E2E SoA] SD Proxy

- **Centralized SD** can be achieved using SD Proxy
 - Service discovery messages are exchanged through one central S/W module, called “SD proxy”.
 - **SOME/IP-SD message** can be used also for communication between ECU and SD Proxy
- **Security and Traffic issues** of distributed SD approach can be handled by Centralized SD
 - Each service can be found and subscribed to by only allowed ECU’s.
 - Service availability and search/subscription attempt can be efficiently monitored.



[E2E SoA] Service Router

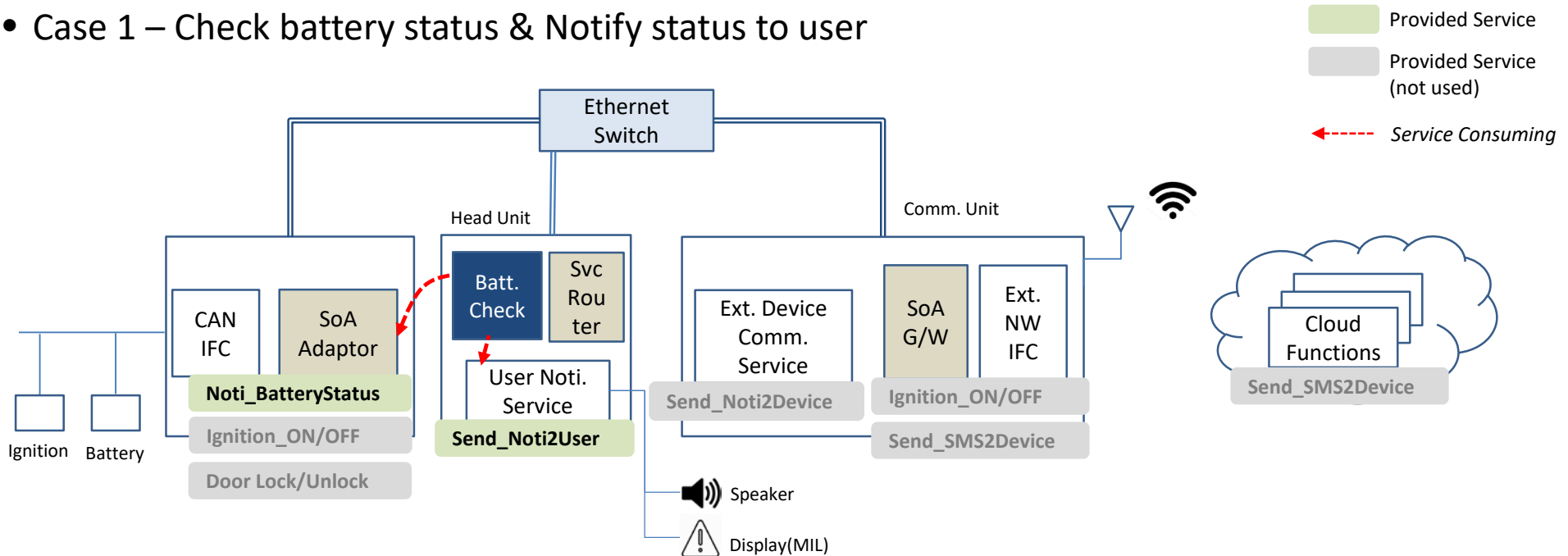
- Issues from distributed nature of SoA can be handled using Service Router
 - Services can be consumed only **through Service Router**.
 - Service Routing can be applied **for selected services**
 - : e.g., services with high security level, non-time-critical services, service use across domain.
 - **SD Proxy** can be used for efficient service routing implementation.
 - **Security and Resource Issues** can be efficiently handled.
 - Service access can be controlled based on domain, ECU, service or even method.
 - Policy can be also applied dynamically, e.g., depending on IDS module.



Use Case

[Use Case] Battery Status Check & Notification

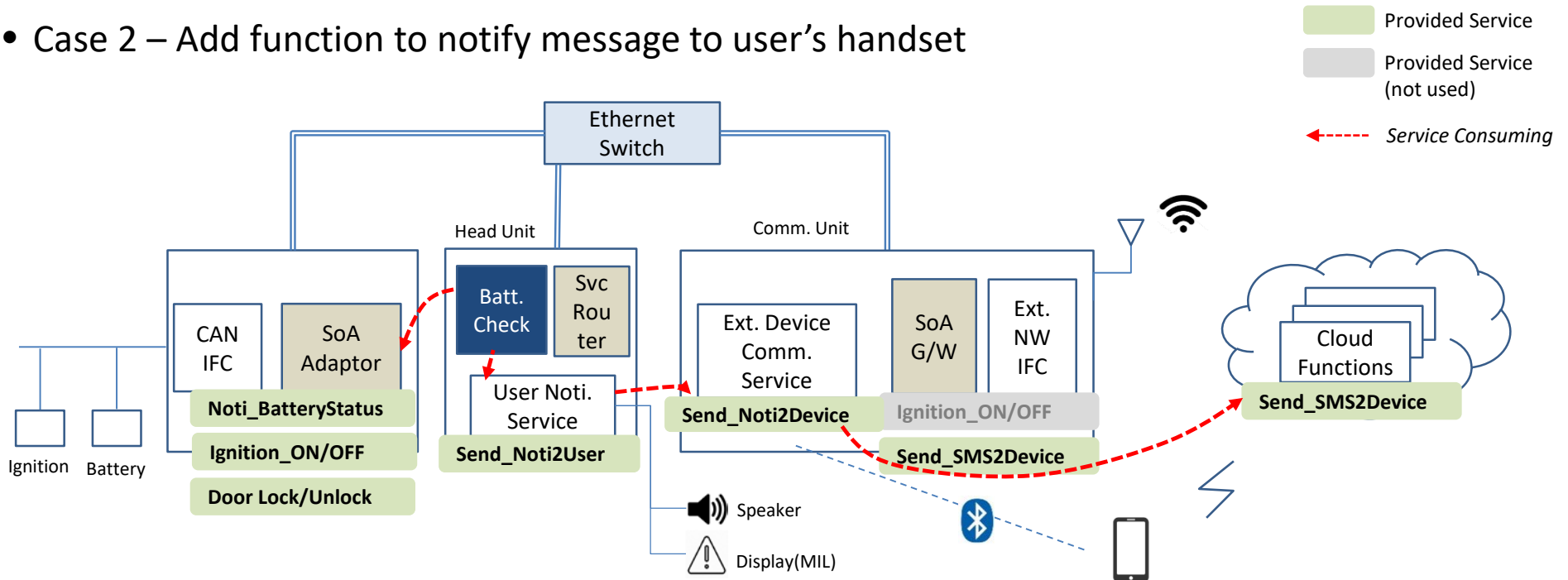
- Case 1 – Check battery status & Notify status to user



- “Battery Check App” consumes two services
 - Gets battery status information : *subscribing to “Noti_BatteryStatus”*
 - Checks the battery status,
 - Notifies to user, if battery low: *invoking “Send_Noti2User”*

[Use Case] Battery Status Check & Notification

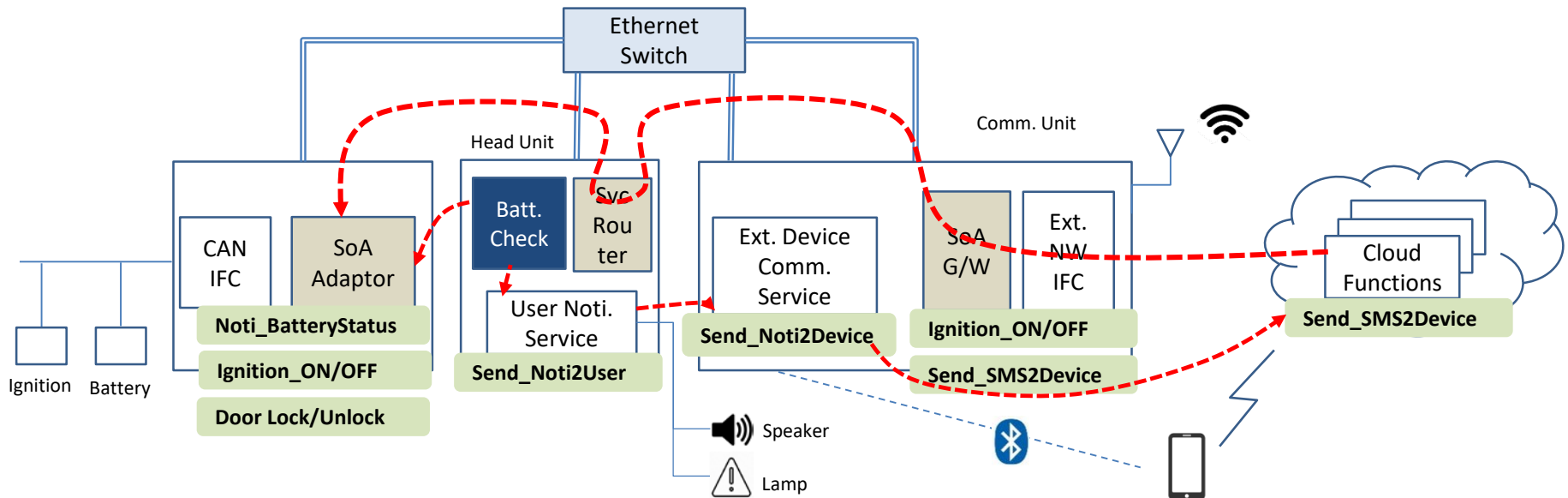
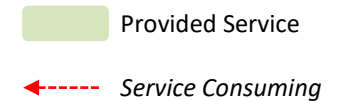
- Case 2 – Add function to notify message to user’s handset



- “User Noti. Service” App
 - Detects that the driver is not in the car : *checking “Ignition On/Off” status and “Door Lock/Unlock” status*
 - Sends notification to user’s handset via external network (Bluetooth or SMS) : *invoking “Send_Noti2Device”*
- “Ext. Device Comm. Service” App
 - Detects user’s handset is not connected through Bluetooth
 - Sends SMS to the handset: *invoking “Send_SMS2Device” (provided by Cloud through SoA G/W)*

[Use Case] Battery Status Check & Notification

- Case 3 – Send command (from external device) to vehicle legacy system



Cloud Function

- Gets user's ignition-on command to re-charge
- Remotely turn on ignition : invoking "Ignition ON" (provided by SoA Adaptor through SoA G/W)
(Note that "Ignition ON/OFF" method invocation is routed and access-controlled by Service Router)

Concluding Remarks

- Automotive Ethernet & IP brought innovation in automotive system/software architecture
: **Service-oriented Architecture (SoA)**
- SoA concept can be **extended to End-to-End** ranging from legacy ECU's to external devices.
- SoA Adaptor and SoA G/W can be used for legacy and external devices, respectively.
- SoA can be efficiently managed by using other SoA Entities like SD proxy and Service Router.
- End-to-End SoA enables fast and efficient deployment of various **connected car services**.
- Other features like **variant handling** or **Plug-and-Play** can also benefit from End-to-End SoA.
- **Joint design of SoA and SDN** is in progress.