IS SOME/IP THE RIGHT SOLUTION FOR THE NEXT 10 YEARS OF VEHICLES?

Dr. Lars Völker, Technica Engineering
Josetxo Villanueva, Ampere, Renault Group
Jan Schäferling, Technica Engineering

AMPERE

technica engineering
Member of KPIT Group

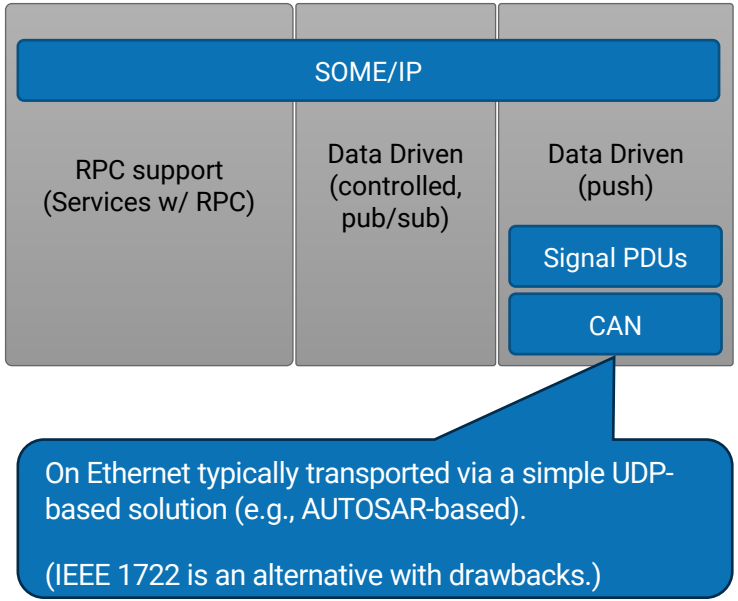
AUTOMOTIVE ETHERNET CONGRESS
10TH ANNIVERSARY
MARCH 05-07, 2024 | MUNICH, GERMANY

AGENDA

- SOME/IP History
- Challenges, Benefits, and Future Needs
- Deep dive into selected Challenges and Future Solutions
- Summary

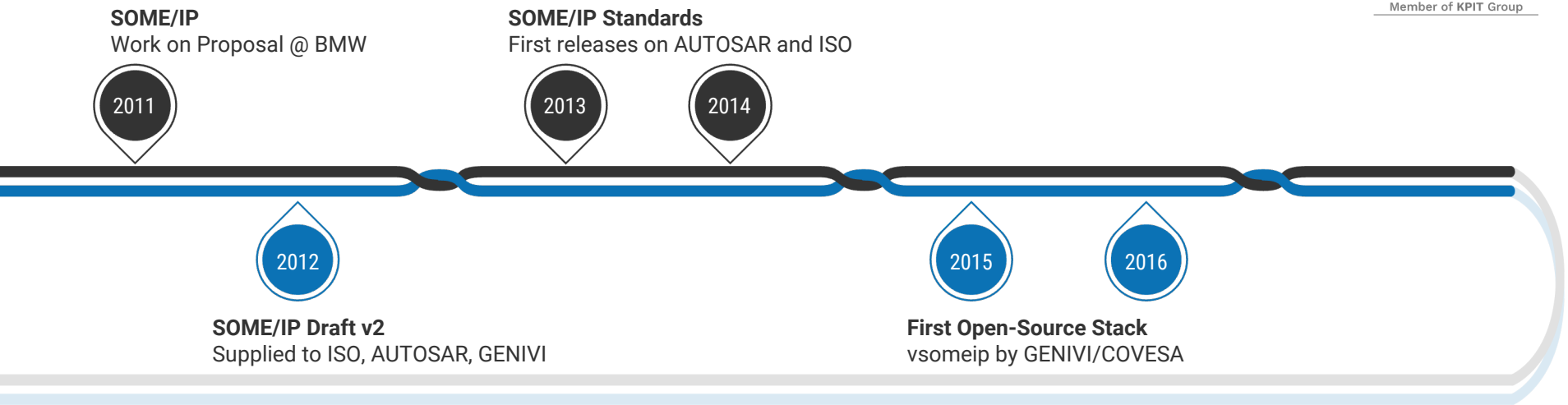
SOME/IP TODAY

- SOME/IP covers the important paradigms:
  - Data Driven & Pub/Sub & RPC & Service Orientation
  - Most alternatives are limited to just a subset
- SOME/IP is simple, efficient, and fast
  - Designed for automotive and embedded first
  - Very low CPU and memory usage
  - Serialization avoids mistakes like TLV and is much faster ([1] did not fully cover this).
- Ethernet vehicles are powered by SOME/IP
  - Many OEMs are using or introducing it
  - 10s of millions of vehicle on the street right now
- **How did we get here?**
- **What can we learn for the next 10 years?**



[1] B. Petersen: "The Promise and Pitfalls of TLV Serialization", 2023 Ethernet & IP @ Automotive Technology Day

SOME/IP HISTORY



- SOME/IP was designed after an extensive market research with 50+ solutions considered

- SOME/IP design goals:
  - Support for service-orientation and common automotive use cases
  - Solution for automotive, embedded, and AUTOSAR
  - An open standard covering different stacks to build a common middleware (scalability)
  - In vehicles as quickly as possible as Automotive Ethernet was about to be rolled out

SOME/IP HISTORY @ RENAULT GROUP

technica engineering GROUPE RENAULT

2016

SOME/IP POC

IEEE EIPTD 2017, San Jose
Testing Expo 2017, Stuttgart

2019

SOP First SOME/IP Service

Command & Control
Control RTP Video Streams
Control TFTP File Transfers
Service Discovery for fast-startup
Multiplatform (POSIX & AUTOSAR)

Cockpit Domain

2022

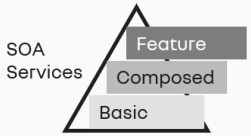
Cockpit, ADAS,
Body & Infotainment Domains

SOP + 50 SOME/IP Services

Re-use of 1st Service
Multi-Domain Command & Control
Serialization of complex structures
Multicast communication
QoS, safety & security

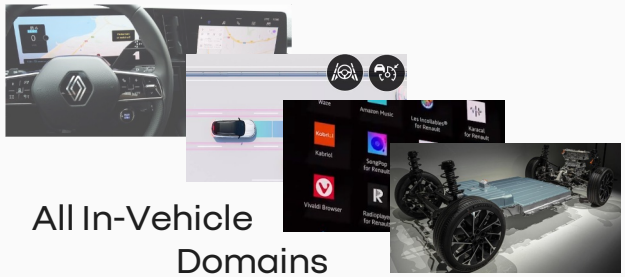
SOME/IP ROADMAP @ RENAULT GROUP

Introduction of SOME/IP –TP
Service Oriented Architecture
Hierarchical Definition Of Services
Decoupling features, logics, sensors & actuators
Adaptive , Classic Autosar & POSIX



SOP + 150 SOME/IP Services towards SDV

2026



All In-Vehicle Domains

After 2030

Software Defined Networking?

Current challenges
Future needs

SOME/IP BENEFITS, CHALENGES & FUTURE NEEDS

AMPERE

BENEFITS

- Service Discovery
- Fast & Effective serialization
- Multiplatform Standard Communication
- Standard Safety, Security & QoS
- Broad Support of automotive industry

FUTURE REQUIREMENTS

- RUST implementation for SOME/IP
- Unique Interface Description Language (IDL) for all platforms
- SOME/IP for everything?
  - Observability of parameters ?
  - Video & Audio Stream Transfer?
  - Diagnostics & reprogramming?

CHALLENGES

1. Start-up performance
2. Service Interface Design
3. Signal to Service
4. QoS, Safety, Security in SOME/IP
5. Improve toolchain

- Services need to start up fast
- Guidelines for Service Design
- Right balance between Signals & Services
- QoS, Safety, and Security supported
- Faster & easier toolchain SDV

SOME/IP BENEFITS, CHALENGES & FUTURE NEEDS

BENEFITS

- Service Discovery
- Fast & Effective serialization
- Multiplatform Standard Communication
- Standard Safety, Security & QoS
- Broad Support of automotive industry

FUTURE REQUIREMENTS

- RUST implementation for SOME/IP
- Unique Interface Description Language (IDL) for all platforms
- SOME/IP for everything?
  - Observability of parameters ?
  - Video & Audio Stream Transfer?
  - Diagnostics & reprogramming?

CHALLENGES

1.	Start-up performance	• Services need to start up fast
2.	Service Interface Design	• Guidelines for Service Design
3.	Signal to Service	• Right balance between Signals & Services
4.	QoS, Safety, Security in SOME/IP	• QoS, Safety, and Security supported
5.	Improve toolchain	• Faster & easier toolchain SDV

CHALLENGE 1: START-UP PERFORMANCE

AMPERE

ECU1 START-UP In CAN BUS

- No sync between Sender & Receivers
- If Sender is not available before global start-up time-out expires, Receivers might raise errors.

As a result,

- Problems arise very late in V-Cycle
- Slower Start-up

ECU 1 START-UP in Ethernet network with SOME/IP

- Sync between Server & Clients thanks to SOME/IP-SD
- Clients only consume a Service when is available.

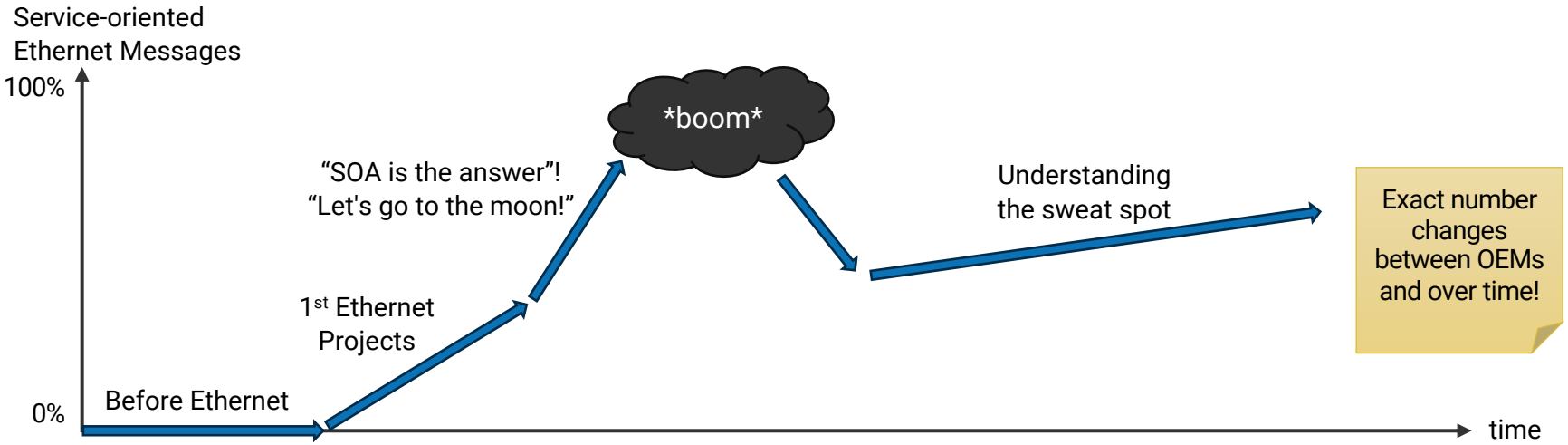
As a result,

- Systems are more dynamic & independent
- Faster Start-up times

→ SOME/IP-SD IMPROVES START-UP BEHAVIOR COMPARED TO CAN

CHALLENGE 2: **SERVICE INTERFACE DESIGN**

- Design starts with understanding what should be a service
  - Adoption of Services and Service-Oriented Architecture allows for fresh ideas
  - Risk: Services and SOA are initially overused or used on the wrong way
  - Not all messages on Ethernet should be service-oriented!
  - Many OEMs experience a similar adoption cycle



CHALLENGE 2: **SERVICE INTERFACE DESIGN (2)**

technica engineering
Member of KPIT Group

Useful Service Interfaces (good)

- Consider all possible operations
  - Methods, events, and fields
- Bidirectional communication considered
- Designed for the use cases
  - Purpose design
- Leads to good “API structure”
- Leads to good understanding

- Fulfills the promise of Service-Orientation

“Pale” Service Interfaces (bad)

- Uses only a limited subset of operations
  - e.g., only events or only fields.
- Mostly unidirectional (like CAN PDUs)
- Possibly automatically generated
  - e.g., for Signal-to-Service
- Might only have the “API structure” of CAN
- Little or no benefit for understanding

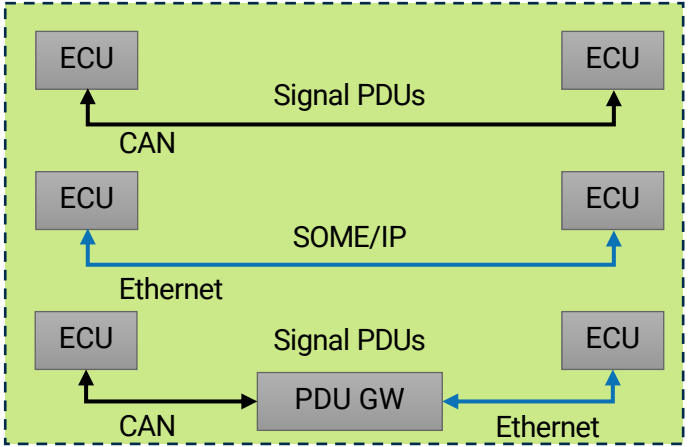
- No benefit compared to regular PDUs

Designing useful service interfaces requires experience!

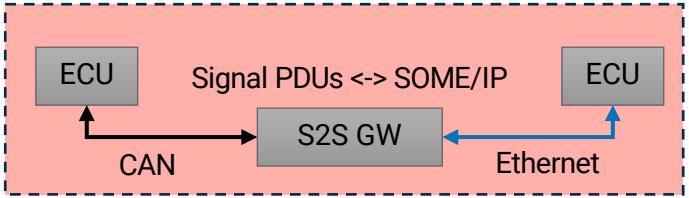
CHALLENGE 3: **SIGNAL-TO-SERVICE (S2S)**

- S2S is a communication paradigm adaption
  - Data-driven vs. service-orientation
- Avoid adaption of communication paradigm!
  - If data-driven on CAN, keep it like that on Ethernet
  - PDU Transport keeps end-to-end principle
- Be careful with Signal-to-Service (S2S):
  - Generated S2S is a scalability and safety nightmare
  - S2S inside an ECU can be fine: think API adaption!
  - If your service adds value or intelligence, S2S is ok
- Excellent communication designs combine:
  - Service-orientation (e.g., based on SOME/IP)
  - Data-Driven Signal PDUs (e.g., based on AUTOSAR PDUs)

Recommended designs



Danger Zone

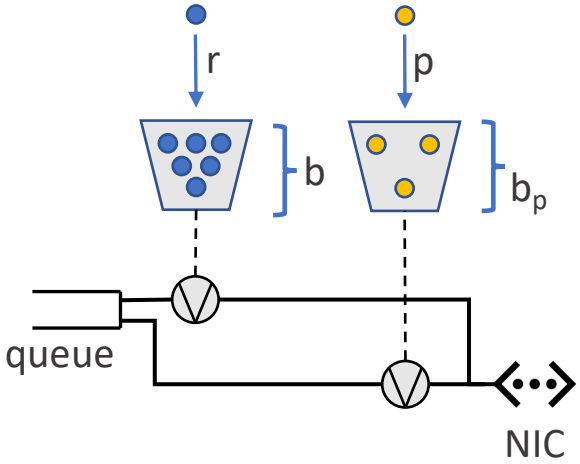


CHALLENGE 4: QOS, SAFETY, SECURITY IN SOME/IP

- What Quality of Service means, differs by person:
  - Reliability of message delivery: 0, 1, >1 [protocol]
  - Frequency adaption of cyclic messages [local stack]
  - Real network QoS [whole system]

- SOME/IP:
  - Static reliability based on “Reliable Flag” in IDL
  - Frequency adaption by local stack possible but not required
  - Mapping to different priorities to achieve network QoS

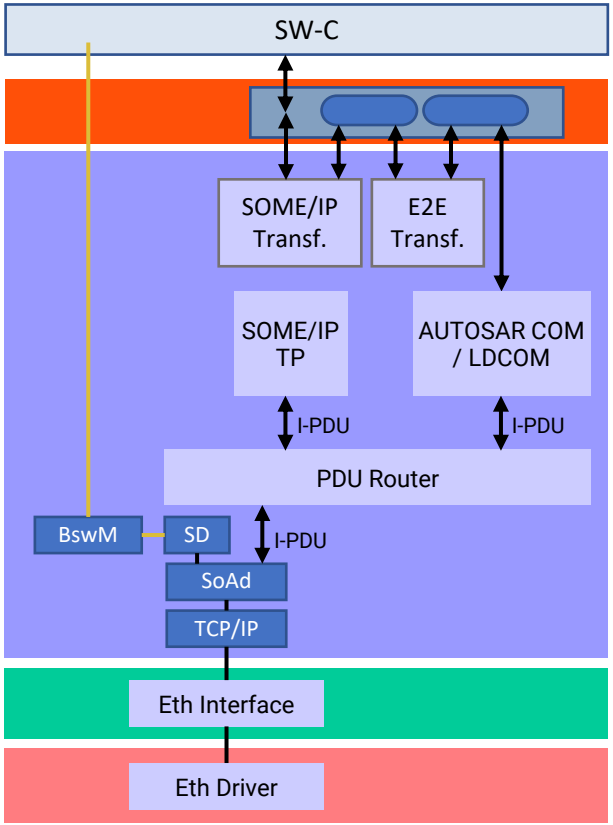
- SOME/IP choices are best practice in Automotive:
  - QoS is “by contract” and not chosen “by implementation”
  - Predictability is more important than flexibility of implementer
  - Communication system is engineered as one



Token Bucket Shaper

CHALLENGE 4: QOS, SAFETY, SECURITY IN SOME/IP

- SOME/IP supports safety, many alternatives do not
- Example: AUTOSAR Classic and the Transformer concept
  - Transformers were created to allow high efficiency serialization
  - E2E safety transformer allows safe SOME/IP implementations
- This allows SOME/IP to be used for safety critical use cases
  - At the same time, the amount of ASIL code was minimized (cost!)
- SOME/IP implements safety “by contract”
  - And not “by implementation” to ensure that Safety is enforced
  - This allows stringent requirement enforcement and tracing



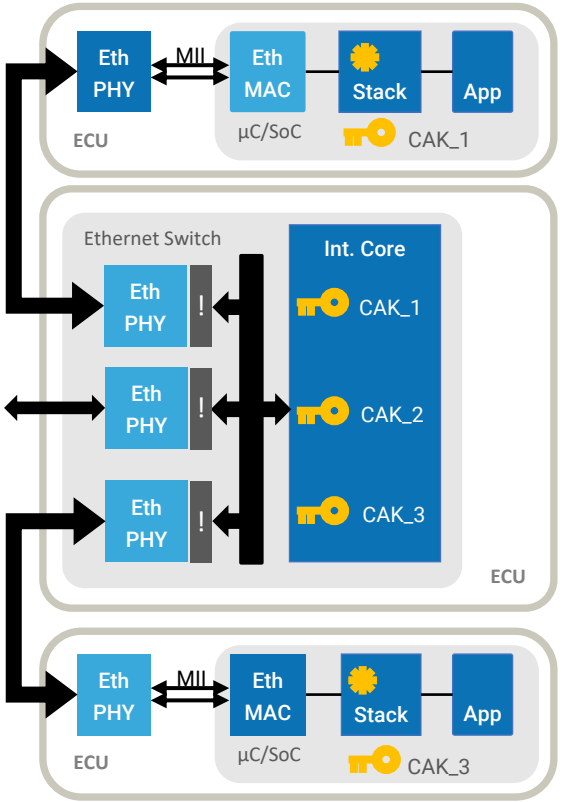
based on AUTOSAR Classic architecture

CHALLENGE 4: **QOS, SAFETY, SECURITY IN SOME/IP**

- SOME/IP does not define its own security protocol
  - Holistic stack design present in contrast to other industries
  - You need to protect a local network, and this includes helper protocols

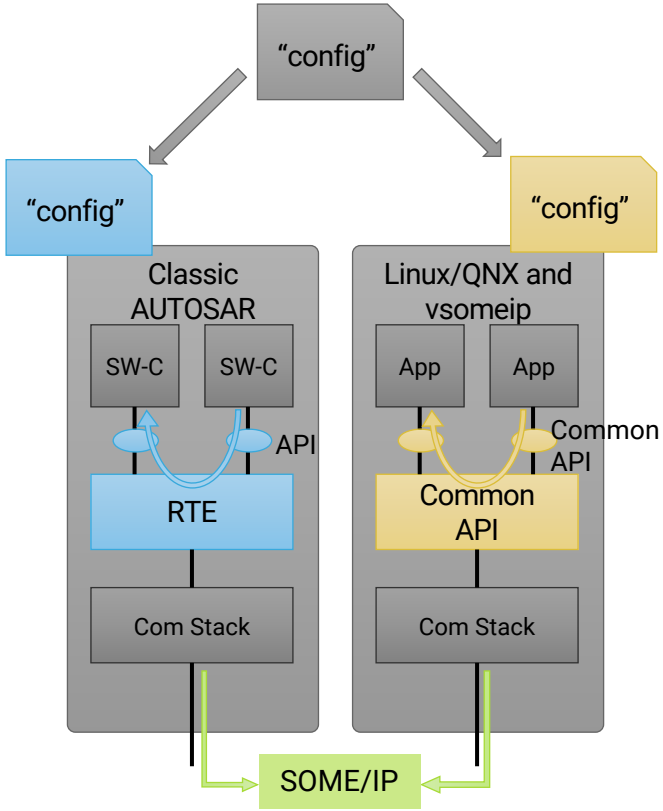
- Security in SOME/IP
  - AUTOSAR allows to use SecOC inside SOME/IP
  - Many stacks support (D)TLS or IPsec
  - Most stacks allow ACLs
  - Some more advanced policy solutions exist

- Recommendation “simple and holistic”:
  - Network security for communication stack instead of SOME/IP
  - Use MACsec to protect layer 2 and stop address spoofing
  - Use (simple) ACLs for access control in SOME/IP and SOME/IP-SD

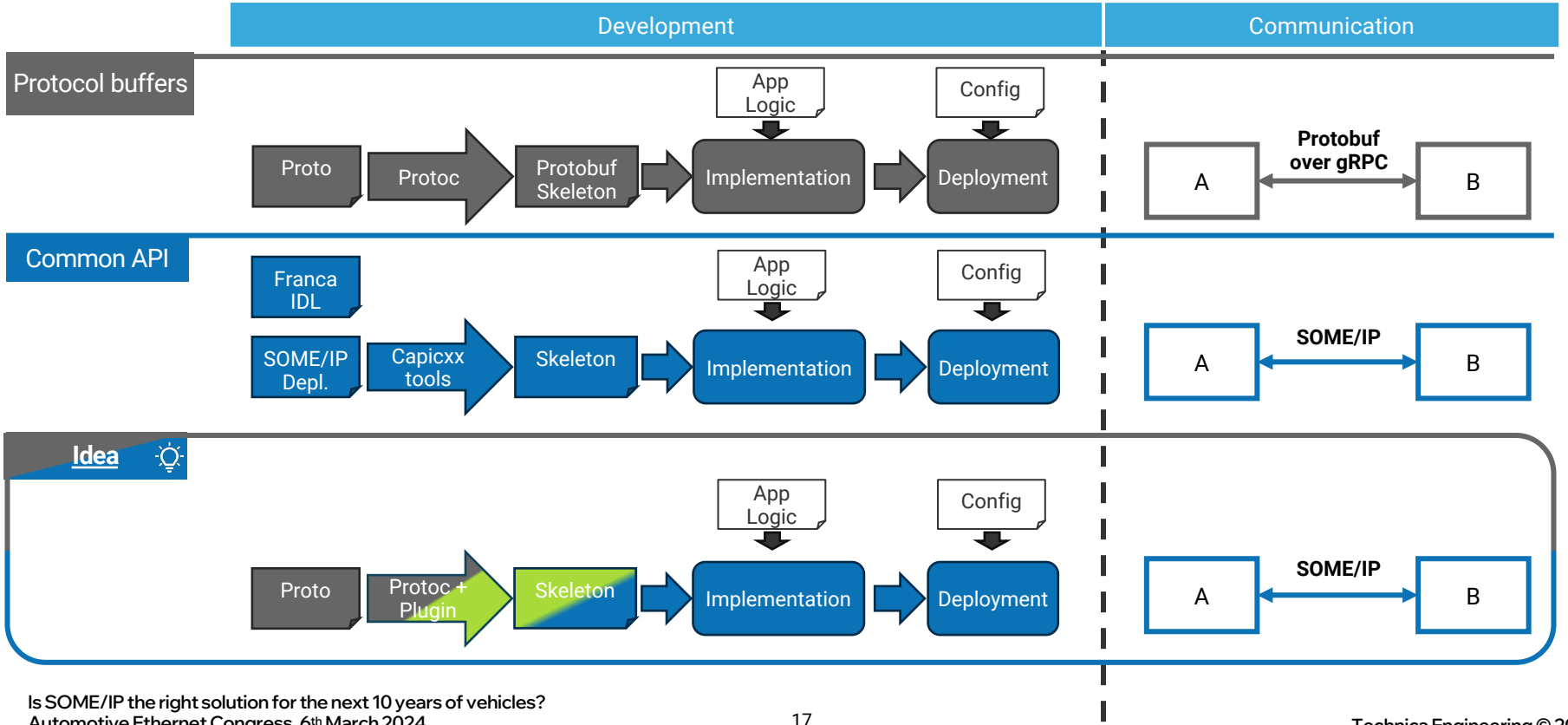


CHALLENGE 5: **TOOLCHAINS**

- ARXML is not the best IDL for SDV
  - Too complicated, not repository compatible, implementation dependent
  - But SOME/IP can be used with ARXML
- Very common misconception:
  - “IPC/API, RPC, and IDL are a fixed combination”
  - e.g., if you want Binder or “proto”, you DO NOT need to use GPB
  - Typically, only API and IPC have a strong interaction
- Example: SOME/IP runs on all platforms
  - Different IPC, APIs, and IDL
  - But still compatible on wire
- Recommendations:
  - Do not limit your thinking based on known combinations



IDEA: USE PROTO FOR SOME/IP?



EXTENDING PROTOC

Protoc is the compiler used to convert IDL to Source Code in multiple languages.

Compilers are translators from one to another language. They usually perform the following generic main steps:

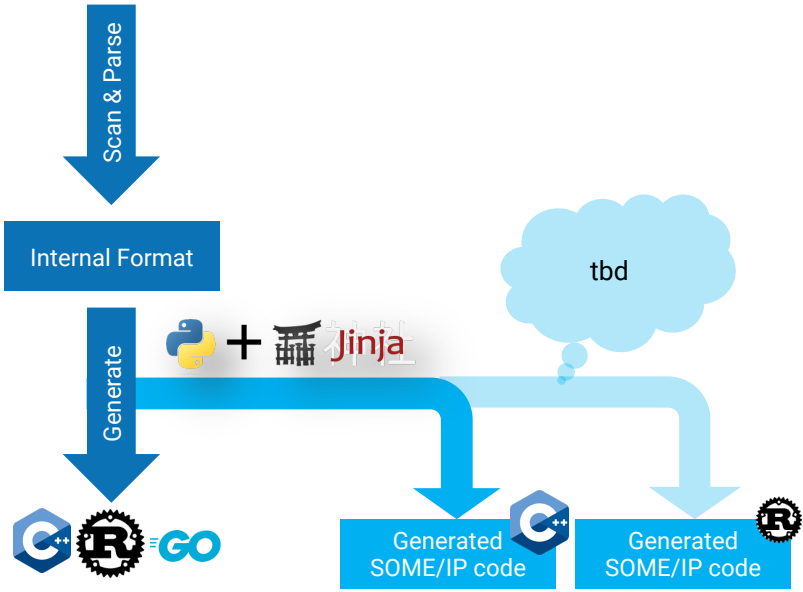
- Scanning
- Parsing
- Generation

Protoc can be extended using (e.g., python-based) plugins to easily support new languages

➔ **Allows extension of Code Generation**

In our toolchain a python-based plugin:

- Accesses the parsing results of protoc
- Renders the information into templates using Jinja2
  PoC: templates are based on CommonAPI and vsomeip



ADDING MISSING FEATURES FOR SOME/IP

```
syntax = "proto3";
import "someip.proto";
message Empty {} // like "null"
message DummyMessage {
  int32 val = 1 [(someip_options).int_type = UINT16];
}

service TestService {
  option (service_id) = 1234;
  rpc testMethod(DummyMessage) returns (DummyMessage) {
    option (cfg) = {
      method: {
        method_id: 0x123;
      }
    };
  }
  rpc testEvent(Empty) returns (DummyMessage){
    option (cfg) = {
      event: {
        event_id: 0x124;
      }
    };
  }
  option (eventgroups) = {
    name: "test_eg_1";
    eventgroup_id: 0x1;
    event_ids: [ 0x124 ]; // [ testEvent ]
  };
}
```

SOME/IP Dataformat
SOME/IP specific attributes via Field-Options
(e.g., finer grained UINT definition)

Methods
Based on native **rpc** keyword.
Supports Request & Response natively, Fire & Forget via **Empty** datatype
➔ Not responding in case of Fire & Forget solved via *protoc-plugin (=code generator)*
SOME/IP metadata like *method_id* added via Method-Options

Events
Based on native **rpc** keyword.
➔ Parameterless method which „returns“ the event + ‚event‘-option

Eventgroups
Custom Service-Option

SUMMARY

- SOME/IP is used in millions of vehicles on the street
  - And there is no reason for this success story to change!

- Success factors:
  - Early entry into automotive standards and markets (first solution in production)
  - Open standards, Open-Source stacks, and Open-Source tools
  - Designed and optimized for Automotive and embedded
  - SOME/IP has excellent performance (serialization and startup)

- Lessons learned
  - Designing Services is hard
  - Do not misunderstand Signal-to-Service
  - Understand design principles for QoS, Safety, and Security
  - Do not judge a protocol by an IDL, API, or Stack (e.g., AUTOSAR)

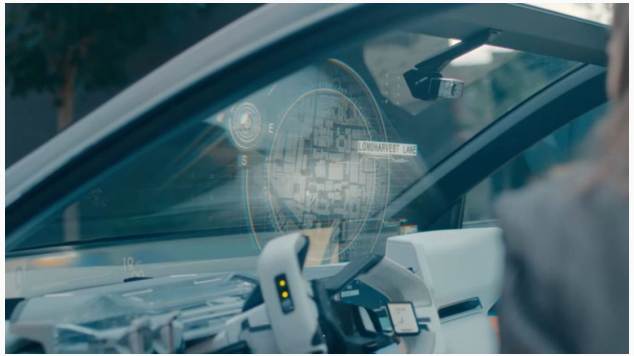
Questions?

BACK TO THE FUTURE : **SOME/IP IN 2053!?**

Renault's Scenic Vision and Megane E-Tech star in Netflix's "Bodies" series, following detectives across time solving a case.

Based on Si Spencer's graphic novel, set in present and future with Renault vehicles aiding investigations in East London's Whitechapel.

The present & future Renault vehicles communicate with SOME/IP.



AMPERE

technica engineering
Member of KPIT Group

THANK **YOU**