S-CORE Communication Scheme

Combining CommonAPI with mw::com/LoLa

2025

SMART TECHNOLOGY FOR SMARTER MOBILITY

Consumers
Provider
SOA
Enterprise ESB
Service Bus (ESB)
Order Management
Inventory
Payment Processing
Data Integration
Security

Agenda

Motivation

Key message

- Expand the supported use cases for S-CORE communication framework
- Accelerate the development for the communication framework
- Ensuring safety workflow for all future use cases

Executive Summary

Observations

- S-CORE Communication Feature set primarily targets a true Service bus realization
- mw::com is more focused on transport layer (zero-copy IPC)
- The goal is to leverage S-CORE mw::com's modern architectural advantages, which include a **high-performance**, **"zero-copy" design** and **native OS integration** (like QNX), particularly for Intra-ECU Transport
- CommonAPI runtime is still capable of supporting multiple communication architectural patterns (binder / gateway based approaches).
- CommonAPI can be leveraged to support missing features like a **mature IDL & Toolchain** (Franca) and support multiple transport protocols (zero copy, copy, Some/IP...)
- CommonAPI is a best match implementation for S-CORE implementation feature set.

Clarification about CommonAPI

General Points

CommonAPI adds latency overhead due to its copy based nature

- CommonAPI adds minimal latency overhead, the overhead usually comes from the underlying binder.

CommonAPI is built for SomeIP/DBUS

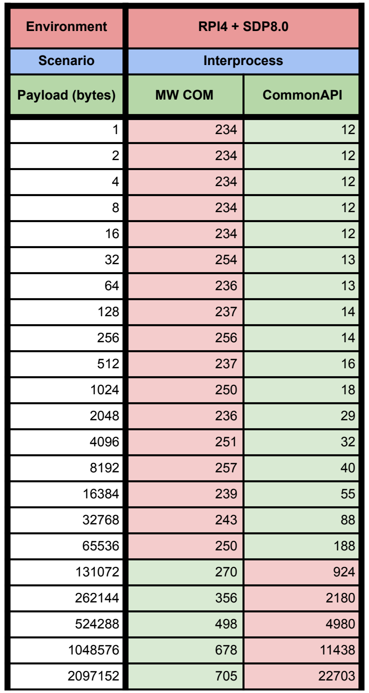
- Valeo Implemented a binder based on QNX Message Passing with as low as three kernel calls for the roundtrip shown in the following measurements
- Valeo is working on the following binders
  - PCIE on Qualcomm
  - Inter core Communication

CommonAPI can not be extended to support zero-copy approaches

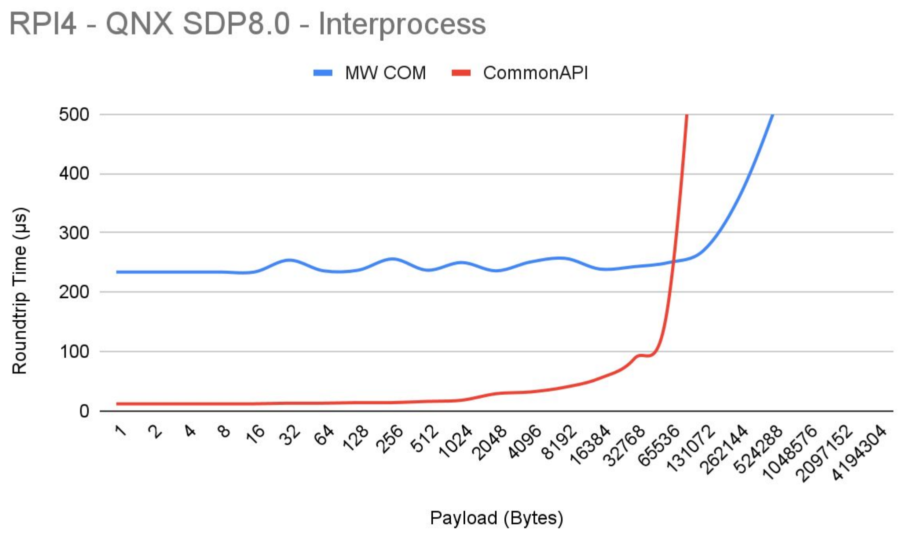
- Valeo is already working on a zero copy extension which would support buffer management through the typical “Allocate”, “Release” APIs
- Valeo is working on a POC for using mw::com/LoLa as a binder under the new CommonAPI extension

Benchmarks

Comparison



Environment	RPI4 + SDP8.0	
Scenario	Interprocess	
Payload (bytes)	MW COM	CommonAPI
1	234	12
2	234	12
4	234	12
8	234	12
16	234	12
32	254	13
64	236	13
128	237	14
256	256	14
512	237	16
1024	250	18
2048	236	29
4096	251	32
8192	257	40
16384	239	55
32768	243	88
65536	250	188
131072	270	924
262144	356	2180
524288	498	4980
1048576	678	11438
2097152	705	22703

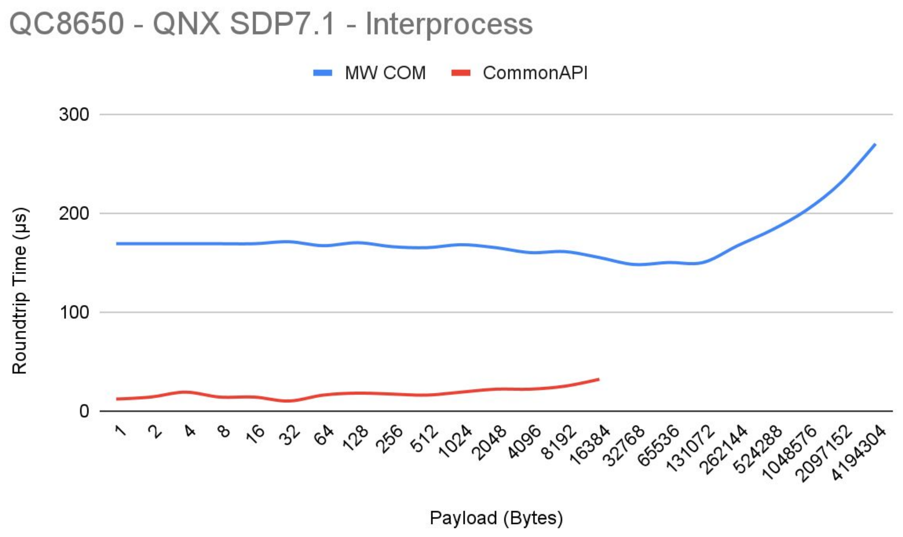


- CommonAPI adds minimal overhead
- Copying approach IPC offers an even better performance metrics in small to medium data transfer (~16K)
- For large data a zero copy binder extension can be implemented giving good performance with the light overhead of CommonAPI

Benchmarks

Comparison

Environment	QC8650 + SDP7.1	
Scenario	Interprocess	
Payload (bytes)	MW COM	CommonAPI
1	169	12
2	169	14
4	169	19
8	169	14
16	169	14
32	171	10
64	167	16
128	170	18
256	166	17
512	165	16
1024	168	19
2048	165	22
4096	160	22
8192	161	25
16384	155	32
32768	148	
65536	150	
131072	150	
262144	167	
524288	183	
1048576	203	
2097152	231	



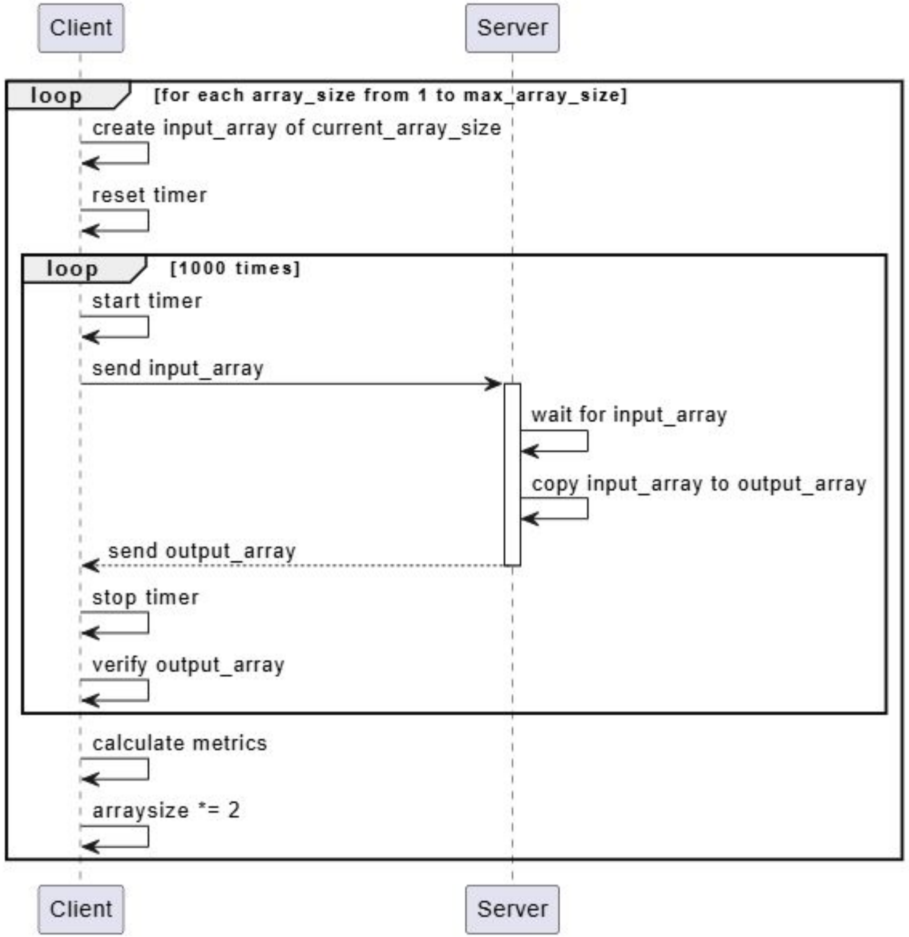
- CommonAPI adds minimal overhead
- Copying approach IPC offers an even better performance metrics in small to medium data transfer (~16K)
- For large data a zero copy binder extension can be implemented giving good performance with the light overhead of CommonAPI

Test Case scenario

Pseudocode

```
current_array_size = 1

WHILE current_array_size <= max_array_size:
   input_array = create_array_of_size(current_array_size)

   FOR i FROM 1 TO 1000:
      output_array = create_empty_array()
      start_timer
      send input_array
      wait_for ouput_array
      stop_timer

   total_time = get_total_time()
   mean_time_per_call =
        total_time / loop_count_per_payload

   current_array_size = current_array_size * 2

END WHILE
```

Performance Test Interaction



Recommended Strategy (based on current progress)

	CommonAPI Custom Binder / SomeIP binder	mw::com shared memory IPC	CommonAPI → mw::Com extension Custom Binder / SomeIP binder / Lola binder
Zero Copy Support	□	✓	✓
ara::com compliance	x	✓	✓
Safety	✓	✓	✓
IDL / ToolChain	✓	x	✓
SOME/IP	✓	□	✓
Methods/ Attributes	✓	□	✓
Typed Data	✓	□	✓

Recommended Strategy

The Hybrid Architecture

- **Conclusion**: Currently SCORE communication feature set is pending the realization of multiple features to be a true service bus most of which can be covered by the CommonAPI current implementation in addition to the support of the zero copy that can come from LoLa IPC complemented by the Valeo extension for zero copy support in CommonAPI

- **Recommendation:** Merge both into one final solution that capitalizes on already available solutions
  - Supports FrancaIDL (score edition)
    - Methods
    - Attributes
    - Events
    - Topics
  - Supports Binder Strategy
    - Support SOME/IP (Binder or Gateway)
    - Support QNX Message Passing
  - Supports Zero Copy Interface
  - Supports Rust
  - Safety certifiable

Valeo

SMART TECHNOLOGY
FOR SMARTER MOBILITY