# Cost Perspective of a Service Oriented Architecture in Vehicle Design

Boris Schegolev[1], Himesha Wijekoon[2], Vojtěch Merunka[3,4]

[1]Czech University of Life Sciences Prague, Faculty of Economics and Management, Department of Information Engineering, Czech Republic; e-mail: schegolev@pef.czu.cz
[2]Czech University of Life Sciences Prague, Faculty of Economics and Management, Department of Information Engineering, Czech Republic; e-mail: wijekoon@pef.czu.cz
[3]Czech University of Life Sciences Prague, Faculty of Economics and Management, Department of Information Engineering, Czech Republic; e-mail: merunka@pef.czu.cz
[4]Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering, Department of Software Engineering, Czech Republic; e-mail: vojtech.merunka@fjfi.cvut.cz



**Abstract.** This paper looks at organization of micro-controllers in a Service Oriented Architecture in road vehicles. Research is being done on why SOA was originally considered by automotive manufacturers and compares the situation with the one of nowadays. Overview of a common implementation setup is provided as an example to showcase the resulting complexity of the solution. This is followed by a discussion on alternative paths that may be taken to reduce the overall cost of the solution with comparison of key differentiating points between SOA and the alternative solution.

**Keywords:** SOA; Micro Services; design patterns; cost efficiency; Separation of Concerns; Forced Encapsulation.

## 1 Introduction

Vehicle manufacturing industry is over 100 years old. It has experienced numerous changes in approaches to manufacturing, design and market positioning. What's even more important, over that time the industry had the opportunity to degrade some design and manufacturing cues that were deemed ineffective, only to reinvent them decades later in a different context.

One such change took place at the end of the 1980s, when computerization of vehicles became economically viable. Not only it altered the production processes, but it also required the whole product to be regarded in a different manner. No longer was it a set of mechanical elements with given projected reliability, but rather a complete system that can be designed in a way we understand the systems design and development today. (Hula, Alson, Bunker, Bolon, 2014).


Proceedings of the 9th International Conference on Information and Communication Technologies in Agriculture, Food & Environment (HAICTA 2020), Thessaloniki, Greece, September 24-27, 2020.

## 2 Motivation

One of the differences between the mechanical and electronically controlled elements is that the latter can provide feedback. This is crucial from a systematic point of view. This has been widely accepted in automotive production, which resulted in a wave of effort to replace mechanical moving parts with an electronically enhanced unit. (Hula, Alson, Bunker, Bolon, 2014)

Generally speaking, moving parts have to be controlled in modern cars. Ideally, all moving parts should be computerized to provide information on the part's usage, health, errors, etc. Initial automakers' solutions used dedicated controllers and computation units to support them. For example, Antilock Brake System (ABS), Anti Slip System (ASR), Electronic Brakeforce Distribution (EBD), Electronic Stability Programme (ESP) and other active suspension parts (to name a few), all have a dedicated computer chip to support its need for computing power.

This approach largely resembles the Service Oriented Architecture (SOA) or Micro Services, widely used in the software world. Compared to the software, the hardware solution required in production of an individual vehicle is way more sensitive to production, deployment and maintenance costs. The question here is whether the advantages of SOA outweigh the added complexity of the system when discussed in a highly cost-sensitive environment.

In this paper we will look into the organization of micro-controllers in cars. Research will be done on why SOA was favored by manufacturers and comparison will be made with the current situation of R&D. A common implementation setup will be reviewed as an example to showcase the resulting complexity of the solution, followed by a discussion on alternative paths that may be taken to reduce the overall cost of the solution with comparison of key differentiating points between SOA and the alternative solution.

## 3 SOA

Service Oriented Architecture (SOA) is at the heart of a revolutionary computing platform that is being adopted world-wide and has earned the support of every major software provider. (Erl, 2005) The micro service-based architecture implements several concepts (Singh, Huhns, 2005) (Marks, Bell, 2006) (Udantha, 2019):

- Scalability
- Availability
- Resiliency
- Flexibility
- Independent, autonomous
- Decentralized governance
- Failure isolation
- Auto-Provisioning
- Continuous delivery through DevOps
- Separation of Concerns

• Forced Encapsulation

### 3.1 Original implementation arguments

Even though all of the concepts are deemed beneficial to a vehicle's controller network design, it's probably the failure isolation that tipped the scale. It's easy to understand that at the point in time when these decisions were made, the production quality particularly in the electronics segment was different from today's standards. The automakers had to adopt not only the new functionality, but also the complete technology that supported it.

Another factor that played a role in architectural decisions is the power output of individual chips. At that historic moment the processing units were largely under-powered. Individual chips had to be chosen carefully for a given task to prevent failures, overheating, etc.

### 3.2 Implementation details

Individual chips, properly scaled for the task, had to interact further to give the benefit of a responsive system. This was covered by a Controller Area Network (CAN bus). Its aim was to provide a common protocol to integrate individual micro-controllers into a network of units that can communicate information in a unified manner. (Johansson, Törngren, Nielsen, 2005)

Later on, CAN bus was replaced by a LID bus. It served the same purpose and its main goal was to provide a lower-cost alternative for the proprietary CAN bus. Many sources still reference it a CAN bus and do not make a difference between the two as they are conceptually same.

The later results can be illustrated in an official schematic provided by Volvo:

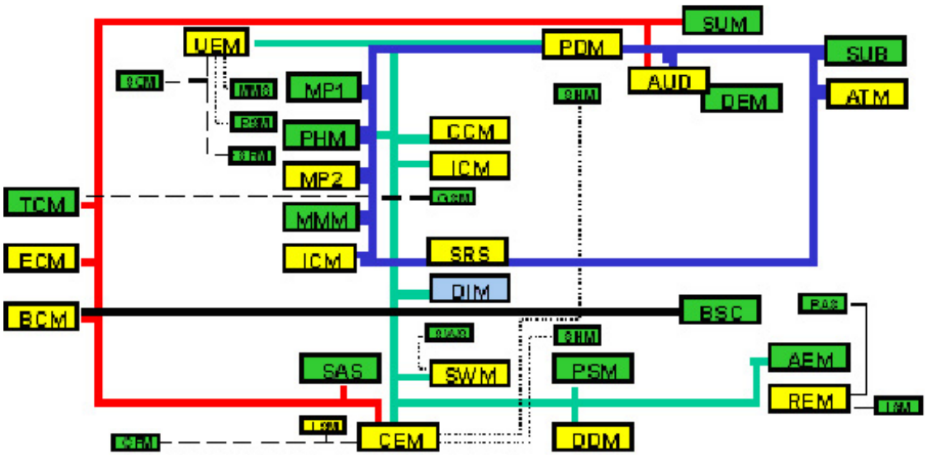


**Fig. 1.** Distributed control architecture for the Volvo XC90. Two CAN buses and some other networks connect up to about 40 ECUs. (Courtesy of Volvo Car Corporation.) (Johansson, Törngren, Nielsen, 2005).

## 4 Centralization

As a part of simplification process, some automakers are considering a partial or even a complete removal of a common bus between controllers. This has large implications on the overall solution. In particular, computational power requirements, manufacturing processes, pattern support and, of course, cost efficiency.

### 4.1 Computational power

Current state of the processing units is very different from the situation of the end of the 20th century. Controllers and chips are easy to produce, and the demanded production quality standard may be high.

Also, there is a huge performance overhang above what is required for almost any individual function. This said, there is no reason to believe that computational power may be a limiting factor preventing unification of multiple controllers into one unit.

### 4.2 Manufacturing processes

Reduction in a number of controllers used to cover the same functionality results in wider use of analog communication. For example, when a vehicle's door controllers are replaced with a single unit, this unit will be farther from each individual door. This will require more analog wiring (controller-to-servo) then in the case when the controller is incorporated in the door itself.

It was stated that Tesla Model S (as an example of a highly centralized architecture) uses about 3km of wiring. The number can be allegedly reduced multiple times. (D'Angelo, 2017)

### 4.3 Pattern support

Service Oriented Architectural patterns (as described in section 3) are suspected of being generally applicable to any distributed system. Here individual patterns will be confronted with the possible implementation options in automotive environment and effort will be made to validate its benefits in given context.

**Scalability** (horizontal) is naturally reduced. As stated earlier though, the requirement of horizontal scalability is minimal, as the performance overhang of individual chips may be enormous.

**Availability and Resiliency** is only a concern in case of critical systems. These, of course, can use separate controllers. Yet again, manufacturers like Tesla have shown that even critical components like Self Driving Computers (SDC) may be implemented with multiple CPUs on the same motherboard.

**Flexibility** is a matter of maintenance and development point of view. This is not really degraded by unification of similar controllers or equal redundant controllers.

**Independent and autonomous** services are welcome, but not required. This largely covers very specific situations.

**Decentralized governance** is not beneficial in automotive environment at all. A vehicle is an isolated system with only a few input-output options, which provide all the governance possible.

**Failure isolation** is a big issue. It also is probably the main argument why manufacturers focus on the SOA instead of following the obvious simplification strategy.

**Auto-Provisioning** is not beneficial in automotive environment. The services are closely incorporated with the hardware, so there can be no automated provisioning of new services.

**Continuous delivery through DevOps** is yet again non-beneficial in automotive environment, as the services are closely incorporated with the hardware.

**Separation of Concerns** is a development pattern that has a great use in vehicle manufacturing. This is because it allows to purchase some of required functionality. However, if the functionality is completely developed in-house by the company itself, the benefits are much smaller.

**Forced Encapsulation** is another development pattern. This one is exactly the reason why a common bus had to be introduced in every modern vehicle. It also causes the obvious costs attached to the development of bus-compatible controllers and production of the bus itself during every vehicle's manufacture.

### 4.4 Cost efficiency

Development and maintenance of a large system is costly. The further the development deviates from a monolithic design, the more costly the development becomes. Micro Services have their great advantages, but cost efficiency is not one of them.

Service Oriented Architecture introduces borders between individual services. These have to be maintained, documented, and in case of vehicle context they also have to be manufactured.

### 4.5 Example of a centralized model

Electric vehicles have fewer moving parts in general. There are no driveshafts, no intake and exhaust systems, etc. In case of the aforementioned Tesla, the solution seems to come down to a body control unit, Self-Driving Computer (SDC) and a central computer with infotainment. Currently Tesla also use in-door universal micro controllers to reduce wiring in between the components.

Here is a comparison of a similarly sized vehicles. One uses a micro service setup (Chevrolet Bolt, left) and one uses a centralized setup (Tesla Model 3, right).

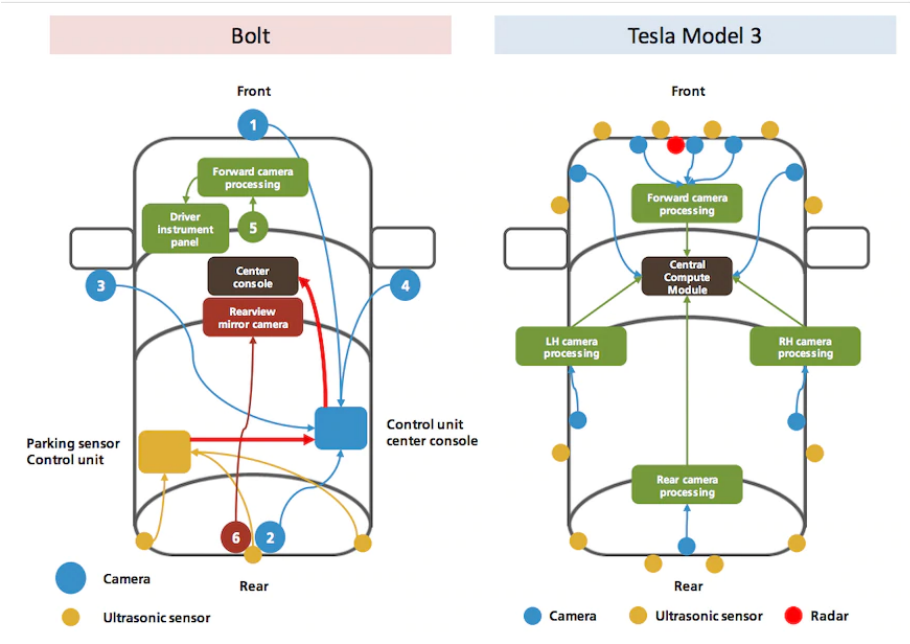


**Fig. 2.** Source: UBS (Rapier, 2018).

## 5 Discussion

Utilization of a common chip to complete multiple tasks is an option. Vehicle manufacturers historically used a Service Oriented Architecture, but some are targeting simplification through:

- Computation power sharing. This makes possible to use a single HW unit to provide multiple functions, provisioning individual applications with unused computational power.
- Workspace sharing between applications. This is important during both development and maintenance, as applications may utilize common logging, same outer-world interfaces, unified storage, etc.
- Great reduction in HW production costs. Due to less produced elements the HW design and production may be largely simplified.
- Some reduction in SW production costs. As developers are not required to overcome strict restrictions imposed on them by forced encapsulation, the SW development processes may become less costly.

## 6 Conclusion

It has been shown that there is a historic reason why vehicle manufacturers maintain the Service Oriented Architecture. SOA has undeniable benefits in both SW development world and in automotive field. Redundancy and failure isolation are hugely beneficial, same as Separation of Concerns during the development process.

Yet the costs of maintaining this design are large and may use some reduction. There are companies that try to achieve cost efficiency through removal of SOA from their micro-controller network. It has been shown that some of the concerns are of little value, while others may be mitigated by proper design even within a reduced number of separate computational units.

## References


1. Hula, A., Alson, J., Bunker, A. and Bolon, K. (2014). Analysis of Technology Adoption Rates in New Vehicles. SAE Technical Paper 2014-01-0781, 2014, doi:10.4271/2014-01-0781.
2. Erl, T. (2005). Service-Oriented Architecture (SOA): Concepts, Technology, and Design. Prentice Hall, Englewood Cliffs, ISBN-13: 978-0131858589
3. Johansson K.H., Törngren M., Nielsen L. (2005) Vehicle Applications of Controller Area Network. In: Hristu-Varsakelis D., Levine W.S. (eds) Handbook of Networked and Embedded Control Systems. Control Engineering. Birkhäuser Boston. DOI: 10.1007/0-8176-4404-0_32
4. Rapier, G. (2018). Tesla's electronics are miles ahead of the competition — but Wall Street analysts who tore down a whole car worry they could be "too centralized" (TSLA). [online] markets.businessinsider.com. Available at: https://markets.businessinsider.com/news/stocks/tesla-stock-price-electronics-miles-ahead-but-could-be-too-centralized-ubs-warns-2018-8-1027473918 [Accessed 29 Mar. 2020].
5. Singh, M.P. and Huhns, M.N. (2005). Service-oriented computing: semantics, processes, agents. Hoboken N.J: J. Wiley. ISBN: 978-0470091487
6. Marks, E. A., & Bell, M. (2006). Service-oriented architecture: a planning and implementation guide for business and technology. Wiley. ISBN: 978-0471768944
7. Udantha, M. (2019). Microservice Architecture and Design Patterns for Microservices. [online] Medium. Available at: https://medium.com/@madhukaudantha/microservice-architecture-and-design-patterns-for-microservices-e0e5013fd58a [Accessed 29 Mar. 2020].
8. D'Angelo, M. (2017). Tesla's "flexible circuit" technology could spark wiring changes in EVs. [online] TESLARATI. Available at: https://www.teslarati.com/teslas-flex-circuit-spark-changes-evs/ [Accessed 29 Mar. 2020].