Total Cost of Ownership: Cloud-based vs. Onboard Vehicle Software Components

Daniel Baumann, Martin Sommer and Eric Sax
Institut fuer Technik der Informationsverarbeitung (ITIV)
Karlsruhe Institute of Technology (KIT)
Engesserstr. 5, 76131 Karlsruhe, Germany
E-Mail: {daniel.baumann, ma.sommer, eric.sax}@kit.edu

Falk Dettinger and Michael Weyrich
Institute of Industrial Automation and Software (IAS)
University of Stuttgart
Pfaffenwaldring 47, 70550 Stuttgart, Germany
E-Mail: {falk.dettinger, michael.weyrich}@ias.uni-stuttgart.de

***Abstract*—The automotive industry is increasingly focusing on connected vehicles that have the opportunity to connect to external platforms, such as the cloud or edge. In this context, the electric/electronic (E/E) architecture is evolving from a signal-oriented to a service-oriented architecture where loosely coupled services, representing functions or the software components (SWCs) they are composed of, can be dynamically connected. At the same time, realization by means of independent services enables the execution both in the vehicle and in the communication network, like the cloud. The costs involved in developing, operating, and maintaining vehicle SWCs have a significant impact on whether it makes sense to execute them in the cloud. In this paper, the authors propose an approach to calculate the Total Cost of Ownership (TCO) with Capital Expenditures (CapEx) and Operating Expenses (OpEx) of SWCs for the two different execution platforms, vehicle and cloud. The TCO model includes the lifecycle of the function from development to usage and maintenance. In a case study with a machine learning SWC for the heating, ventilation and air conditioning (HVAC) function, the model is investigated and break-even periods for the two platforms are calculated.**

***Keywords*-Total Cost of Ownership; Electric/Electronic Architecture; Cloud-based Software Components; Cloud Computing.**

I. INTRODUCTION

The automotive industry is rapidly moving towards connected vehicles, integrating a growing number of software-defined functions [1]. This transition presents a critical decision for upcoming Electric/Electronic (E/E) architectures: where to execute these functions or the Software Component (SWC) they are composed of, aboard the vehicle or in the cloud.

Safety-critical functions may require local processing to minimize latency and ensure robustness, while non-safety critical comfort functions are possibly suitable for cloud execution. Previous work identified cost as a significant factor in determining the suitability of the different execution platforms [2].

Therefore, a comprehensive cost analysis based on Total Cost of Ownership (TCO) is needed to enable high-quality decision-making for the cloud offloading. This paper contributes to the ongoing research on connected vehicle architectures by providing a structured model for evaluating the economic feasibility of cloud-based execution. We anticipate that this model will be valuable for both automotive E/E architects and software developers involved in the design and deployment of software-defined functions for the future of connected vehicles.

The remainder of the paper is structured as follows: Section II will describe the theoretical background necessary for understanding the models utilized in the paper. Section III will cover the related state of work, providing an overview of existing research and literature on existing TCO models. Following in Section IV, these models for the onboard and the cloud-based SWCs are described. The TCO model will then be analyzed with the two execution platforms cloud and onboard in a case study with a fleet size of 1000 and 5000 vehicles in section V. A discussion of the use cases and the cost reduction options will be provided in section VI. Finally, section VII presents a conclusion to the paper and outlines future work.

II. BACKGROUND

A. E/E architecture

The E/E architecture refers to the electrical and electronic system of a vehicle, which includes all electrical components and control units required to operate the vehicle [3].

Historically, specific functions were executed on dedicated Electronic Control Units (ECUs) with limited interconnectivity. This so-called distributed architecture has been replaced by the domain-oriented architecture, where software is abstracted from the hardware and logically subdivided according to functions instead of according to individual control units [4]. This means that more than one SWC runs on a single ECU. In the future, centralized architectures will increasingly be used. Fewer but more powerful control devices designated to High Performance Computer (HPC) run the software components that underlie the functionality [5].

Today's automotive applications use signal-oriented architectures in which the software and hardware components involved are closely coupled with each other. In order to manage the increasing proportion of software in the vehicle and to enable dynamic E/E architectures, so-called service-orientated architectures are being introduced. The various functions are designed as independent services that can interact with each other in a modular fashion. Instead of linking individual components directly with each other as in the signal-orientated architecture, the components are viewed as independent services that can communicate via defined interfaces. [6]

Service-oriented E/E architectures can be used to design vehicle systems in such a way that they can communicate seamlessly with cloud services. This enables the utilization of cloud resources, such as the provision of data-intensive services. [7]

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

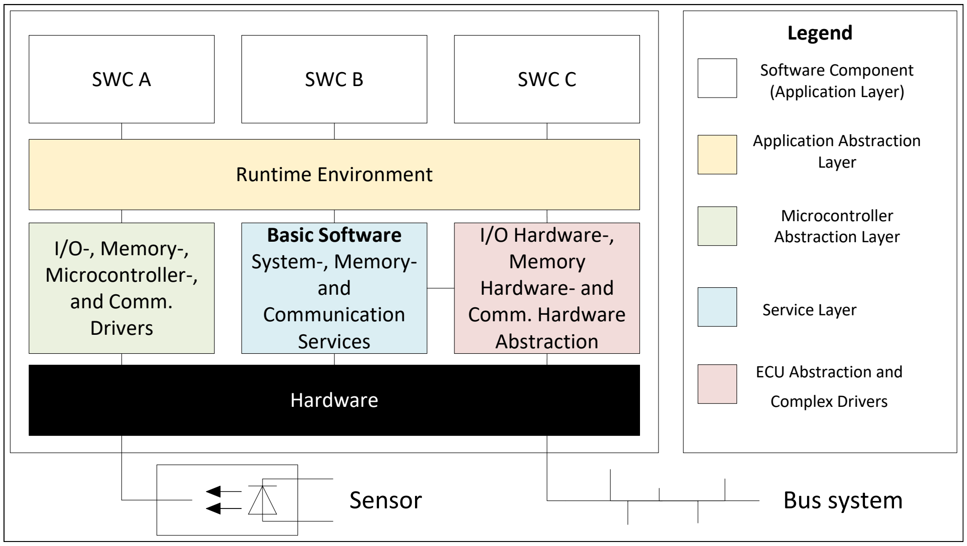

Figure 1. Software components within the AUTOSAR layered software architecture (adopted from [3]).

We define cloud-based SWC as follows, referring to and adapting Milani's definition [8]: "Cloud-based software components are regulation, control, or monitoring tasks that use the computing & storage capacities of the cloud instead of the available computing capacities of the vehicle. They can use both information from the vehicle and data from the cloud as input. The output of the relocated components optimize existing functions in the vehicle, replace them, or create a new function for themselves." The localization of software components can be explained using the layer structure of the AUTomotive Open System ARchitecture (AUTOSAR) architecture and refers to the application layer (s. Figure 1). Whereas a software component itself is defined as an "entity with discrete structure, such as an assembly or software module, within a system considered at a particular level of analysis" [9].

B. Cloud Computing

In general, Cloud Computing refers to the characteristics of a flexible and scalable infrastructure that conveys the illusion of unlimited, on-demand access to IT resources. The most widely adopted definition originates from the U.S. National Institute of Standards and Technology (NIST) [10]: “Cloud computing is a model to enable ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” In cloud computing, three different service models have emerged for accessing cloud resources: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [10].

C. Total cost of ownership

TCO is a financial estimate of the overall cost of a product or service over its entire lifespan, not just the initial purchase price. TCO consider all the direct and indirect costs associated with owning and using the product or service. First introduced by Ellram and Siferd in 1995, the concept of TCO has become widely adopted across industries and academia as a means of evaluating the long-term economic viability of investments. [11] TCO encompasses both the initial Capital Expenditures (CapEx) and the aggregate of Operating Expenses (OpEx) [12]:

$$TCO = CapEx + OpEx \tag{1}$$

CapEx refers to the upfront costs incurred at the time of purchasing the product or service. In contrast, OpEx are the ongoing costs associated with owning and using the product or service over its entire lifecycle.

III. Related Work

In their paper, Martens et al. [13] introduce a TCO approach tailored to cloud computing services. It outlines different pricing structures for cloud computing services and develops a formal mathematical model. The cost categories identified include strategic decision-making, evaluation and selection of service providers, service charges for different cloud models, implementation, support, initial and ongoing training, maintenance and modification, system failures, and backsourcing. A case study is presented as an IaaS example, detailing the cost types and related cost factors.

Kashef et al. [14] provide a detailed specification of cloud computing costs for hybrid clouds. Twenty cost factors are identified in the categories electricity, hardware, software, labor, business premises and cloud service. Costs are broken down into fixed and variable costs over time. The costs of a scenario for running an in-house data center with ten different services is shown.

Walterbuch et al. [15] introduce a TCO model tailored for cloud computing services in a public cloud environment. The example demonstrates the provisioning of a public IaaS Cloud Computing Service.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

The paper by Heinrich et al. [16] proposes a TCO model for cloud computing, covering the cost of adoption, procurement, migration, operation (external and internal), usage, and exit. A case study is presented, comparing two scenarios: a Serverless Scenario and a Lift and Shift Scenario, against an on-premises architecture. Serverless refers to a platform for deploying applications without the user having to care about the underlying infrastructure. In contrast, the term lift and shift is used to describe a function that is migrated to the cloud. The study shows that the operation cost for cloud computing is lower than that on-premises. In terms of total cost, the lift and shift scenario is less expensive than on-premises after 15 years. The serverless scenario is always more expensive than the on-premises solution.

There is a gap in the literature regarding the comparison of traditional onboard functions with cloud computing services, especially within the automotive sector. This paper aims to address this gap by providing a detailed analysis of the TCO of the automotive onboard and cloud functions. Furthermore, we identify and analyze cloud computing as an enabler for further automotive functions and services.

IV. TCO MODEL FOR SOFTWARE COMPONENTS

The development and operation of vehicle SWC in the cloud or on board the vehicle is associated with costs. The TCO of a SWC TCO_{SWC} consist of the development costs C_{dev}, the deployment costs C_{depl}, both capital expenditures, and execution costs C_{exe} which are operating expenses on a monthly basis:

$$TCO_{SWC} = \underbrace{C_{dev} + C_{depl}}_{CapEx} + \underbrace{C_{exe}}_{OpEx} \tag{2}$$

Expert interviews were conducted with a German Original Equipment Manufacturer (OEM) for the creation and evaluation of the TCO model. The cost components of the onboard function and the cloud function are explained in more detail below.

The following cost breakdown relates to the SWC shown in Figure 1, which are assigned to the application layer.

A. Onboard SWC

Similar to equation 2 the costs of an onboard vehicle SWC can be summarized as:

$$TCO_{v,SWC} = C_{v,dev} + C_{v,depl} + C_{v,exe} \tag{3}$$

1) Development: The costs of developing an onboard vehicle SWC $C_{v,dev}$ can be divided into software $C_{v,dev,sw}$ and hardware $C_{v,dev,hw}$. In the following, hardware always refers to ECUs with a microcontroller and the associated peripherals [17]:

$$C_{v,dev} = \sum_{i=0}^{n} C_{v,dev,sw} + \sum_{i=0}^{n} C_{v,dev,hw} \tag{4}$$

Software development costs $C_{v,dev,sw}$ include expenses related to designing, implementing, integrating and testing software functions used in vehicle control systems. The development costs of software increase due to the requirements for reliability and safety. This is because careful validation and verification are necessary to ensure that the software is error-free and robust. Hardware development costs $C_{v,dev,hw}$ include expenses for designing ECUs, sensors, actuators, and other physical components required for the function's functionality. These components must often meet strict requirements for robustness, reliability, and performance to withstand the demanding environmental conditions of vehicle operation.

2) Deployment: The deployment costs of an onboard vehicle function $C_{v,depl}$ consist of the sum of expenses associated with the material, production and logistic of the ECUs $C_{v,depl,ecu}$:

$$C_{v,depl} = \sum_{i=0}^{n} C_{v,depl,ecu} \tag{5}$$

3) Execution: The execution costs $C_{v,exe}$ pertain to ongoing expenses associated with the operation and utilization of onboard vehicle SWC. These costs encompass various aspects $C_{v,exe,n}$, including Over-the-Air (OTA) updates of ECUs, operation costs like the energy consumption costs, maintenance and repair costs and costs for customer support and service.

$$C_{v,exe} = \sum_{i=0}^{n} C_{v,exe,n} \tag{6}$$

B. Cloud-based SWC

As described in Section IV in equation 2, the cost of a cloud function can be calculated as follows:

$$TCO_{c,SWC} = C_{c,dev} + C_{c,depl} + C_{c,exe} \tag{7}$$

1) Development: The development of a cloud-based function $C_{c,dev}$ is associated with various costs, including internal introduction $C_{c,dev,intro}$, purchasing $C_{c,dev,pur}$, migration $C_{c,dev,mig}$ and software development $C_{c,dev,sw}$:

$$\begin{aligned} C_{c,dev} = &\sum_{i=0}^{n} C_{c,dev,intro} + \sum_{i=0}^{n} C_{c,dev,pur} \\ &+ \sum_{i=0}^{n} C_{c,dev,mig} + \sum_{i=0}^{n} C_{c,dev,sw} \end{aligned} \tag{8}$$

The internal introduction costs $C_{c,dev,intro}$ encompass strategic planning, training and ensuring security protocols. Strategic planning involves defining the objectives, scope and methods for transitioning from traditional onboard functions to the cloud. Training programs are essential to introduce employees with the new technology and ensure smooth introduction and use. Furthermore, it is essential to integrate security measures to protect data and systems from potential threats. This often requires investment in cybersecurity infrastructure. Procurement costs $C_{c,dev,pur}$ include the cost of purchasing or licensing the software, as well as any associated infrastructure costs. To run a function in the cloud, it is possible to migrate an existing function or develop a function from scratch. The other cost factor can be set to zero accordingly. Migration costs $C_{c,dev,mig}$ refer to the expenses associated with transitioning

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

from existing systems to cloud-based infrastructure, including infrastructure switching, implementation, testing and configuration. Software development costs, on the other hand, relate to the implementation and testing phases of building cloud-based functions. The development costs $C_{c,dev,sw/hw}$ are divided between software and hardware, as with the onboard function, whereby the hardware here is limited to sensors, actuators and other physical components. In contrast to the onboard function, no execution platform needs to be developed.

2) Deployment: Capital expenditures for the deployment of SWCs in the cloud are regarded as very low and therefore negligible. The costs associated with deployment in the vehicle are costs listed in the execution costs for the cloud case. Shifting a SWC to the cloud, converts initial investments for computing power and storage requirements into OpEx.

$$C_{c,depl} = 0 \tag{9}$$

A touchscreen in the vehicle is required to use the function.

3) Execution: The execution of a cloud-based SWC is associated with various costs $C_{c,exe}$. These include the costs for the cloud service provider $C_{c,exe,CSP}$, the internal operation $C_{c,exe,int}$, the usage $C_{c,exe,use}$ and the update & upgrade of the function $C_{c,exe,u\&u}$:

$$\begin{aligned} C_{c,exe} = &\sum_{i=0}^{n} C_{c,exe,CSP} + \sum_{i=0}^{n} C_{c,exe,int} \\ &+ \sum_{i=0}^{n} C_{c,exe,use} + \sum_{i=0}^{n} C_{c,exe,u\&u} \end{aligned} \tag{10}$$

The primary cost factor for cloud-based SWC is the cost of the Cloud Service Provider (CSP) $C_{c,exe,CSP}$. The cost models vary depending on the model of cloud computing and the cloud service provider. Microsoft Azure provides a comprehensive overview of the different options [18]. One commonly used pricing model is pay-as-you-go, where users are billed based on their actual usage of cloud resources. This flexible payment approach enables organizations to dynamically scale their infrastructure according to their needs. Internal operational costs $C_{c,exe,int}$ include maintenance, support and addressing downtime incidents. The costs associated with usage $C_{c,exe,use}$ include those for end-user operation, primarily data transmission charges for a data contract. Additionally, the cost of electricity used to operate the communication module must also be paid and are part of the usage costs. Costs associated with updates and upgrades $C_{c,exe,u\&u}$ are crucial for maintaining the functionality, security and performance of the cloud-based function over time. These costs may include fees for acquiring new software versions, migrating data, testing compatibility and deploying updates across the cloud environment.

C. TCO reduction options

The reduction of the TCO of a vehicle SWC can be implemented by the following three use cases:

1) Saving of a complete ECU in an existing E/E architecture: Assuming that an existing ECU in the vehicle does not have any design-related hardware proximity (e.g., due to installed I/Os) and only a single offloadable SWC is currently deployed on this ECU, it is conceivable that the ECU can be completely omitted and replaced entirely by cloud resources. In this case, the complete CapEx of the ECU will be saved and replaced by OpEx of the cloud. Due to the increasing integration of software components on an ECU as described in section II, this case is very unlikely.
2) Downsizing of an ECU in an existing E/E architecture: By relocating individual SWCs that are executed on an existing ECU to the cloud, the ECU can be re-dimensioned. This means, for example, that less computing power may be required for the ECU, meaning that a lower-performance model within a series can potentially be installed in the vehicle. Thus, saving potentials only arise through cost savings in smaller models of a ECU series.
3) The cloud as a new execution platform alternative for new E/E architectures: When designing a new E/E architecture that is not restricted to the vehicle but allows the cloud as a possible execution location, a TCO comparison needs to be executed for cloud-suitable functions or SWC.

It is important to note that these examples are not exhaustive, but rather provide an indication of the diverse approaches to TCO reduction in the development and management of SWC.

V. CASE STUDY

The proposed TCO model is assessed with a new function for electric city buses. The function is described in [19] and proposes a cloud-based machine learning model that is able to predict the Heating, Ventilation and Air Conditioning (HVAC) energy consumption with different temperature set values.
As this is a new function, it is assigned to the use case 3 "cloud as new execution platform for new E/E architectures" and the two execution platforms need to be compared with the TCO model.

A. Onboard SWC

1) Development: Software development for this new SWC in the vehicle can be estimated at 150,000 € while hardware development is one third of this amount.
2) Deployment: Adding the new SWC to the vehicle's onboard E/E architecture adds the need for computing resources that are currently not available. Therefore, a new ECU would be needed. The cost of a high-performance ECU with machine learning support sums up to 30 €. Additional hardware savings as proposed in subsection IV-A are not feasible with the function.
3) Execution: The main cost driver is the cost of OTA updates. The cost of executing the onboard SWC is rounded to 2 € per month for four updates in a one-year period [20].

B. Cloud-based SWC

1) Development: The software development costs of the SWC for the cloud can be set lower than for the vehicle. This can be attributed to the fact that it is not necessary to rely

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

on automotive-specific runtime environments, but instead common tools can be used for development on standard hardware. For this reason, $C_{c,dev,sw}$ is set to 90,000 €. Introduction, purchasing and migration cost add up to an equal amount.

2) Deployment: Deployment costs are negligible as they are converted into operating expenses.
3) Execution: The Microsoft Azure "Cloud services" offer a suitable configuration named *A0* for our function [21]. The function we have designed takes approx. 20 seconds to calculate for one call. We designed the function to be called every 300 seconds, such that we assume the *A0* configuration is able to handle 10 calls representing 10 vehicles within the defined time frame of 300 seconds. $C_{c,exe,CSP}$ result in the *A0* configuration monthly costs divided by 10 which is 1,49 € per month. Operation, usage and update/upgrade costs sum up to 1€ per month. Again, this is based on updating 4 times annually.

TABLE I. OPEX AND CAPEX FOR CLOUD AND VEHICLE EXECUTION PLATFORM

Exe Platform	C_{Dev}	C_{Depl}	C_{Exe}
Vehicle	200,000 €	30 €	2 €/month
Cloud	180,000 €	0 €	2.49 €/month

The life cycle of a city bus in Germany is 8.3 years [22]. The TCO for both execution platforms for this timeframe calculate to:

- $TCO_{v,SWC} = 200,229.20$ €
- $TCO_{C,SWC} = 180,247.92$ €

A break-even analysis for both execution platforms, both with and without development costs, shows the following periods after which the onboard SWC is again the cheaper option. In this case, the development costs were divided between a bus fleet of 1,000 respectively 5,000 city buses (see Figure 2):

- Break-even for 1,000 respectively 5,000 buses without C_{dev}: 61 months (5.1 years)
- Break-even for 1,000 buses with C_{dev}: 102 months (8.5 years)
- Break-even for 5,000 buses with C_{dev}: 69 months (5.8 years)

VI. DISCUSSION

The role of cloud computing in the automotive sector is becoming increasingly important as OEMs look for innovative ways to improve the functionality and user experience of their vehicles. In the future, many vehicle functions will be cloud-based, making it possible to process data in real time, perform updates and offer personalized services. Naturally, the issue of costs plays an important role here. The TCO model in this paper offers an approach to gain an overview of the costs to be expected when applying a cloud or an onboard SWC. The model enables a cost comparison and thus the inclusion of costs in the decision-making process as to where the SWC should be implemented. In Section IV-C, three approaches

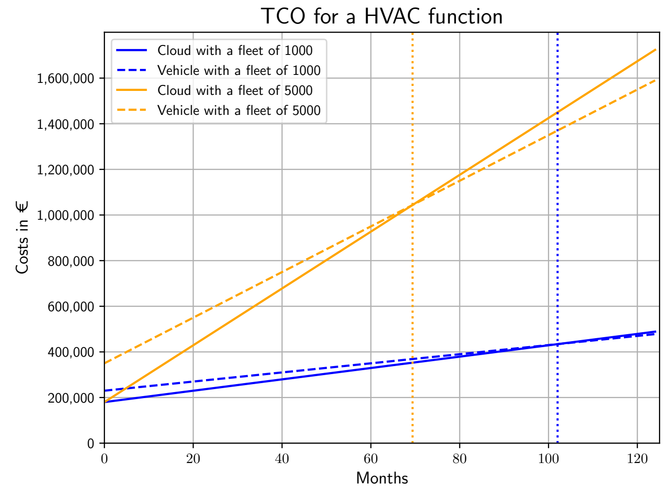

Figure 2. TCO for the HVAC function with a fleet of 1,000 (blue) and 5,000 (orange) city buses. The break-even points are marked with the horizontal dotted line.

for cost savings are presented. Saving a complete ECU is, as already mentioned, unlikely. Downsizing a current ECU by moving SWC to the cloud only has a small financial effect, in the single-digit euro range. This use case is therefore also rather unlikely. The most likely use case: the cloud as the new execution platform is in detail described in Section V. The main costs of a cloud-based SWC for the OEM are the monthly costs for the CSP, which depend on the type of service used. In addition, the OEM has the option of becoming a CSP and hosting its own cloud infrastructure. This can offer strategic advantages as the OEM has more control over its data and services, can better control security and potentially save costs, especially for long-term use. Implementing its own cloud infrastructure, allows the OEM to better meet specific performance and data protection requirements. The OEM also has the data that can be used for the development of new functionalities, for example. Within the use case (Section V) described, a fleet size of 1,000 and 5,000 vehicles is analyzed. Economies of scale must be considered for smaller and larger fleets, as they affect the cost structure. The larger the fleet size, the sooner the break-even point is reached where the aggregated cost of the cloud-based SWC becomes higher than the onboard SWC. Here, the monthly costs for deployment and execution are assumed to be constant, but as the fleet size increases, the costs for ECUs and CSPs shrink [17]. But the scaling factor of cloud costs is higher than that of onboard costs. The described case study focuses on buses, whereas a OEMs fleet of passenger cars is much larger and the economies of scale are therefore of a different order of magnitude.

While comfort functions such as HVAC systems are not considered safety-critical, network failures can still result in a negative user experience. In the scenario where functionality is lost due to network issues, the cost can be seen in terms of user frustration and potential loss of confidence in the product. To address this, manufacturers could implement

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

fallback mechanisms, such as offering offline modes for critical features.

The increasing spread of cloud-based SWC is accompanied by a shift from CapEx to OpEx, which will have an impact on OEMs. This transition reduces upfront costs, allowing OEMs to allocate their financial resources more efficiently. Furthermore, cloud computing enables OEMs to dynamically scale their operations in response to changing demand. Cloud-based services represent also a viable strategy for end customers, who may opt for monthly subscription payments as needed. This approach provides original equipment manufacturers (OEMs) with a continuous cash flow.

VII. Conclusion and Future Work

This paper presents a TCO model that determines the total costs of a vehicle SWC. A distinction is made between two execution platforms, namely cloud and onboard. Three options are presented for reducing the TCO of a SWC. The case of a new execution platform, i.e. the relocation of an SWC to the cloud, was examined using a machine learning function in the city bus sector. Taking development costs into account, the break-even point for executing the function in the cloud is nearly the approximate lifespan of a city bus in Germany.

Although the presented TCO model offers a comprehensive framework for evaluating the total costs associated with vehicle SWCs across different execution platforms, there are several areas that require further investigation. The current model could be enhanced by incorporating additional dynamic cost factors, such as fluctuating energy prices, varying cloud service fees, and changes in hardware costs. A more dynamic model would permit real-time cost assessment and more accurate forecasting. Further research is needed to explore the scalability and adaptability of the TCO model. This includes examining how well the model performs in different organizational contexts and identifying ways to customize the model for specific business needs.

Acknowledgment

The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the project INDU2 - OTRACE within the InnovationsCampus Future Mobility.

References

[1] J. P. Trovao, “Trends in automotive electronics [automotive electronics]”, *IEEE Vehicular Technology Magazine*, vol. 14, no. 4, 2019. DOI: 10.1109/MVT.2019.2939757.

[2] M. Sommer, D. Baumann, T. Rösch, F. Dettinger, E. Sax, and M. Weyrich, “Process for the identification of vehicle functions for cloud offloading”, in *Science and Information Conference*, Springer, 2024.

[3] T. Streichert and M. Traub, *Elektrik/Elektronik-Architekturen im Kraftfahrzeug: Modellierung und Bewertung von Echtzeitsystemen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-25478-9.

[4] A. Bucaioni and P. Pelliccione, “Technical architectures for automotive systems”, in *2020 IEEE International Conference on Software Architecture (ICSA)*, IEEE, 2020. DOI: 10.1109/ICSA47634.2020.00013.

[5] M. Traub, A. Maier, and K. L. Barbehön, “Future Automotive Architecture and the Impact of IT Trends”, *IEEE Software*, vol. 34, no. 3, 2017. DOI: 10.1109/MS.2017.69.

[6] S. Frohn and F. Rees, *From signal to service: Challenges for the development of autosar adaptive applications*, Vector Informatik GmbH, Ed., Mar. 2018.

[7] E. Sax *et al.*, “Analyse der Aktivitäten und Entwicklungsfortschritte im Bereich der Fahrzeugelektronik mit Fokus auf fahrzeugeigene Betriebssysteme”, *Themenpapier Cluster Elektromobilität Süd-West*, 2020.

[8] F. Milani and C. Beidl, “Cloud-based vehicle functions: Motivation, use-cases and classification”, in *2018 IEEE Vehicular Networking Conference (VNC)*, IEEE, 2018.

[9] International Organization for Standardization, *Iso/ iec/ ieee 24765:2017: Systems and software engineering — vocabulary*, 2017.

[10] P. M. Mell and T. Grance, *The NIST definition of cloud computing*, Gaithersburg, MD, 2011. DOI: 10.6028/NIST.SP.800-145.

[11] L. M. Ellram, “Total cost of ownership: An analysis approach for purchasing”, *International Journal of Physical Distribution & Logistics Management*, vol. 25, no. 8, 1995.

[12] P. Rosati, F. Fowley, C. Pahl, D. Taibi, and T. Lynn, “Right scaling for right pricing: A case study on total cost of ownership measurement for cloud migration”, in *Cloud Computing and Services Science: 8th International Conference, CLOSER 2018, Funchal, Madeira, Portugal, March 19-21, 2018, Revised Selected Papers 8*, Springer, 2019.

[13] B. Martens, M. Walterbusch, and F. Teuteberg, “Costing of cloud computing services: A total cost of ownership approach”, in *2012 45th Hawaii International Conference on System Sciences*, IEEE, 2012. DOI: 10.1109/HICSS.2012.186.

[14] M. M. Kashef and J. Altmann, “A cost model for hybrid clouds”, in *Economics of Grids, Clouds, Systems, and Services*, ser. Lecture Notes in Computer Science, K. Vanmechelen, J. Altmann, and O. F. Rana, Eds., vol. 7150, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. DOI: 10.1007/978-3-642-28675-9{\textunderscore}4.

[15] M. Walterbusch, B. Martens, and F. Teuteberg, “Evaluating cloud computing services from a total cost of ownership perspective”, *Management Research Review*, vol. 36, no. 6, 2013. DOI: 10.1108/01409171311325769.

[16] S. Heinrich, N. Kreft, T. Schuster, and R. Volz, “A total cost of ownership model for cloud computing infrastructure”, in *Proceedings of the 56th Annual Hawaii International Conference on System Sciences*, T. X. Bui, Ed., Honolulu, HI: Department of IT Management, Shidler College of Business, University of Hawaii, 2023.

[17] J. Schäuffele, *Automotive Software Engineering: Grundlagen Prozesse Methoden und Werkzeuge effizient einsetzen*, 6. Wiesbaden: Springer Vieweg, 2016. DOI: 10.1007/978-3-658-11815-0.

[18] Microsoft, *Pricing calculator*, https://azure.microsoft.com/en-us/pricing/calculator/, Retrieved: 21.10.2024, 2024.

[19] D. Baumann, M. Sommer, F. Dettinger, T. Rösch, M. Weyrich, and E. Sax, “Connected vehicle: Ontology, taxonomy and use cases”, in *2024 IEEE International Systems Conference (SysCon)*, IEEE, 2024.

[20] S. Abuelsamid, *Automotive over-the-air updates: A cost consideration guide*, Guidehouse Insights, Ed., 2021.

[21] Microsoft, *Cloud services – preise*, https://azure.microsoft.com/de-de/pricing/details/cloud-services/#pricing, Retrieved: 21.10.2024, 2024.

[22] K. Bundesamt, *Fahrzeugzulassungen*, https://www.kba.de/SharedDocs/Downloads/DE/Statistik/Fahrzeuge/FZ15/fz15_2022.pdf?__blob=publicationFile&v=5, Retrieved: 21.10.2024, 2022.

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org