Proceedings of the 56th Hawaii International Conference on System Sciences | 2023

A Total Cost of Ownership Model for Cloud Computing Infrastructure

Simon Heinrich
Pforzheim University
simon.heinrich@lb.hs-pforzheim.de

Nils Kreft
Pforzheim University
nils.kreft@lb.hs-pforzheim.de

Thomas Schuster
Pforzheim University
thomas.schuster@hs-pforzheim.de

Raphael Volz
Pforzheim University
raphael.volz@hs-pforzheim.de

Abstract

A holistic cost assessment of cloud computing architectures is currently hampered by the lack of assessment methods and the absence of a standardized and comprehensive total cost model. This issue creates uncertainty about the cost developments of concrete scenarios and architectural changes. This article proposes a total cost of ownership model for cloud computing, covering the cost of adoption, procurement, migration, operation, usage, and exit. We evaluated our model in multiple application scenarios and against other models. Our model has shown to be substantially more comprehensive and applicable than other available models for cloud computing. Thus, our model can be valuable both in practice and in research. We demonstrate that our model can increase cost transparency and improve decision support.

Keywords: Cloud Computing, Total Cost of Ownership, Model, Analytics, Accounting

1. Introduction

Cloud computing (CC) has changed the way companies conduct business by offering agility, scalability, a short time to market, high availability, and access to complex technologies (Sujeet Kumar et al., 2016). Companies successfully leveraging CC already attain competitive advantages by achieving business model innovation (Berman et al., 2012). Moreover, many authors have reported savings potential due to pay-per-use concepts (Goasduff, 2015; Kokilavani & Saravanan, 2015; Mangiuc, 2017; Snyder, 2017).

Comprehensive CC adoption, however, often falls short of expectations (Raza et al., 2015). Solanki holds the uncertainty in cost development partly responsible for it (Goasduff, 2015). An EMA study revealed that most organizations fear exploding costs instead of expecting savings (2018). Other studies emphasized that ICT controllers, auditors, and administrators often lack cost transparency and fail to make informed decisions about CC (Mahlindayu et al., 2014; Mangiuc, 2017). Responsible decision-makers can lose track due to the wide variety of CC services and the diversity in billing models. This leads to unnecessary costs, up to 50 % increase (EMA, 2017, 2018; Rightscale, 2018).

Raza et al. revealed that cost transparency and potential cost savings work as an accelerator for cloud adoption and help companies enable CC advantages. Cost transparency is also a key step in eliminating unnecessary costs (Armbrust et al., 2010). In addition, KPMG examined this relationship, finding that increasing knowledge of CC leads to higher savings (2017). Kokilavani and Saravanan suggest more sophisticated cost calculation models to identify the cost mechanisms of CC (2015). An extensive cost model for CC is crucial to support further developments but is still absent today.

Our research aims to offer a total cost model, providing financial decision support to leverage CC. Previous research in costing ICT infrastructure has revealed that indirect costs make up to 50 % (Carr & Ittner, 1992; Engates, 2017; Höllwarth, 2012); hence, total cost methods have become the focus of our research. Strebel and Stage consider the total cost of ownership (TCO) approach to be the most significant method both in practice and in research (2010). Alkhanak et al. points out that TCO is the most reliable prognostic tool for ICT infrastructure to ensure long-term success (2016). Our own comparison in this paper supports these findings, and we consider the TCO approach the ideal fit for our cost model.

2. Research design

To comprehensively identify the CC cost components and proper evaluation methods, we chose a multimethod research approach. We conducted both a systematic literature review and expert interviews. The findings were analyzed with a concept-based framework. We extracted requirements for developing a sound TCO model in a separate literature review. Our research design is outlined below and illustrated in Figure 1.

URI: https://hdl.handle.net/10125/103337
978-0-9981331-6-4

Literature reviews were conducted in October 2018 and reevaluated in 2021 and 2022. The literature selection was based on scientific research journals with a rating from A+ to D according to the JOURQUAL3 rating. To ensure a sound body of knowledge the keyword list (Table 1) was constantly adjusted by scanning for relevant literature not identified by the current keywords in the references of the previous findings, as Webster and Watson suggests (2002). For each keyword combination, both the top 100 most recent and most relevant articles were scanned. Journals were not only selected from the primary research fields of economics, computer science, and business administration but also form the adjacent research areas of law, tax, logistics, operation research, and marketing, as suggested by Malone and Crowston (1994) and Robey et al. (2000). Thereby the relevant findings increased from 117 to 216. Strebe and Stage point out that a significant amount of fundamental research for CC is still missing (2010). Therefore, we also reviewed grid-computing literature as the predecessor of CC.

Table 1. Keyword list

Primary Keywords	Secondary Keywords
Cloud Computing, as a Service, Distributed Systems, TCO, Total Cost of Ownership, Grid Computing	Cost, Accounting, Economic, ROI/Return on Investment, ICT, Adoption, Migration, Demand, Requirement, Standard, Model, Request, Criteria, Application

For the expert interviews, ICT architects and application managers of a car manufacturer, as well as cloud service providers (CSPs) were interviewed. All experts had an exclusive position in the CC environment. All expert interviews were transcribed word for word.

For data analysis, we followed the procedure for analyzing qualitative content described by Mayring (2016). We thereby combined both data sources and analyzed them in a concept-based framework, as suggested by Schlütz and Möhring (2013). The aggregation of the findings resulted in the models' cost components and evaluation methods. We added the concept of analytical units, introduced by Téeni (Webster & Watson, 2002, S. xvi), to organize the cost components and evaluation methods and group them into cost categories.

Data	Scientific Journals (JQ3)	Expert Interviews
Collection Method	Systematic Literature Research	Qualitative Content Analysis
Analysis Method	Concept-Based Literature Review	Concept-Based Coding
Result	TCO-Model Requirements	TCO-Model Components and Evaluation Methods

Figure 1. Research design.

3. Related research

Our research for CC cost models only revealed one mathematical TCO approach, one cost-benefit model, and several case studies, which were mostly restricted to operating costs. The cost-benefit model from Marešová et al. (2016) partially covers fundamental cost components but is insufficient because it neglects lifecycle costs. Martens et al. (2012) provide a mathematical approach for CC cost analysis. Their model covers fundamental cost components with suitable evaluation methods and is primarily based on case studies. For a generic TCO model, however, their findings are insufficient because their model is restricted to external operation costs. They neglect the costs of cloud adoption, provider certification, end-user operations, building a cloud platform, capital costs, opportunity costs, and most internal operating costs. In addition, they do not provide a descriptive TCO model. Our research identified several additional cost components, marked italics in Table 2. In this study, we evaluated the significance of these costs.

Table 2. CC TCO model comprehensiveness

Adoption: Strategy, Training, *Organization, Security, GRC, Data, Network, Platform*
Procurement: Strategy, Training, *Provider Certification, Sourcing (Licenses, ICT Components), Cost of Capital, Opportunity Cost*
Migration: Infrastructure Shift, Testing, Implement., Configuration, *Planning (Requirement-, Feasibility-, Application-, Data-Analysis-, Migration-Strategy)*
External: CSP Cost (IaaS, PaaS, and SaaS)
Internal: Maintenance Support, Testing, Training, Downtime, *Management, ICT Components Development, Update/Patch/Release-Management, Platform, Generalization, Standardization, Automation, Disaster Recovery Disaster Management, Change Management*
Usage: Training, Downtime, *End User Operations (Futzing, Self-Support, Peer-Support, Data-Management, Self-Development)*
Exit: Data Export and Lock-In

4. Cloud computing TCO model

Before revealing both our descriptive CC TCO model and its mathematical methodology, we discuss the requirements and quality criteria to develop a sound model. The model is based on our findings from the conducted research studies. We outlined where these findings are consistent with or complemented by the current research.

4.1. Requirements and quality criteria

As requirements for our TCO model, we considered the standards defined by Ellram & Siferd (1994), (1998), who developed cornerstones of the TCO movement. According to these sources, we can summarize requirements in terms of transparency, decision support, usability, relevance, modularity, and comparability. We took the following actions to fulfill these criteria.

Transparency is provided in our model through its foundational premises and structure. *Decision Support* is granted by disclosing the financial burdens of CC, supporting decision-making. *Usability* is ensured by adding evaluation methods to each cost component. Studies have concluded that most TCO models are held back by the absence of evaluation methods. By applying standard cost metrics, we are also able to provide a mathematical approach, enhancing our model's usability to a greater extent. We further evaluated our model in multiple case studies supported by a car manufacturer. The *Actuality* of our model was achieved by reviewing the latest literature and conducting interviews with experts in exclusive positions in the field of CC. The *modularity* of the model is granted by its structure, allowing a use-case-specific selection of cost components, as well as individual expansions. This enables our model to act as a base model for future development and research. The resulting limitation in *comparability* is acknowledged.

We also considered quality criteria as defined by Geißdörfer et al. (2009). In this way, the ability of a cost model to act generically and in a standardized manner is measured, and compliance with our study objectives is ensured. Gartner's TCO model was ranked highest in Geißdörfer's study; therefore, we took inspiration from its foundational structure. All quality criteria and our model's capabilities are presented in Table 3. We took the following actions to accomplish them.

We applied the concepts of *net present value, cost categories, transaction costs, activity-based costing*, and *revenue effects* as fundamental pillars for the TCO approach. These concepts support the complex cost structure of CC with its variety in pricing models and payment intervals. They further allow for a cost analysis using time- or activity-based cost aggregation, leveraging our cost categories. *Cost dependencies* are also considered in our model and are discussed in the mathematical section. The *usability of the model in procurement* is enacted by supporting CSP selection and overall investment decisions. For *development,* the model can be used to identify cost drivers. In *sales,* the model provides cost transparency to customers. The *accuracy* of our model was evaluated in eight case studies.

Qualitative cost components that function as multiples could not be identified for our model. In some models, they can be used as proxies for nonquantifiable cost components. The *overall equipment efficiency*, describing constraints due to wear and tear, does not apply; hence, most physical equipment is outsourced to the CSP. Our TCO model currently fulfills 10 of the 11 remaining criteria. Although we are still testing the model's accuracy, by introducing it to different customers, we are convinced that the current finding supports the relevance of our TCO model. This process enables our model to function as a standardizable base model, with the opportunity for individual model adjustment.

Table 3. Compliance with Geißdörfer's quality criteria

Fulfilled	Further Testing
net present value, cost categories, activity-based costing, transaction costs, revenue effects, cost dependencies, usability in procurement, development, and sales	model accuracy

4.2. Mathematical approach

To enhance the usability of our CC TCO model, we provide a *mathematical approach* using three layers. These layers are also used in our descriptive model. The first layer represents the cost categories (C^{cat}). They illustrate cost distribution through each investment stage and between functional areas. A cost category (C^{cat}) represents an aggregation of all associated cost components (C^{comp}). Cost components (C^{comp}) form the second layer. They disclose cost-driving activities, thus indicating the potential for cost minimization. Furthermore, the distribution of indirect and direct costs is revealed. Costs on both layers are aggregated based on the time interval (i), revealing the cost distribution through the investment period.

The third layer represents the foundation for all cost calculations and is based on the usage of cost drivers (C^{driver}). They are affected by the region (r), payment date (d), and billing interval (u) of the services in use. For example, cost drivers (C^{driver}) for storage can be described as the storage size used and usage (query pattern). Storage pricing differs per region (r). The payment date (d) and billing interval (u) affect the calculation and mean capital commitment used for the cost of capital and opportunity cost calculation. The relation between the layers is shown in Equation I.

$$TCO = \sum_{i}^{n} C^{cat} = \sum_{i}^{n} C^{comp} = \sum_{i}^{n} C_{r,d,u}^{driver}. \quad \text{(I)}$$

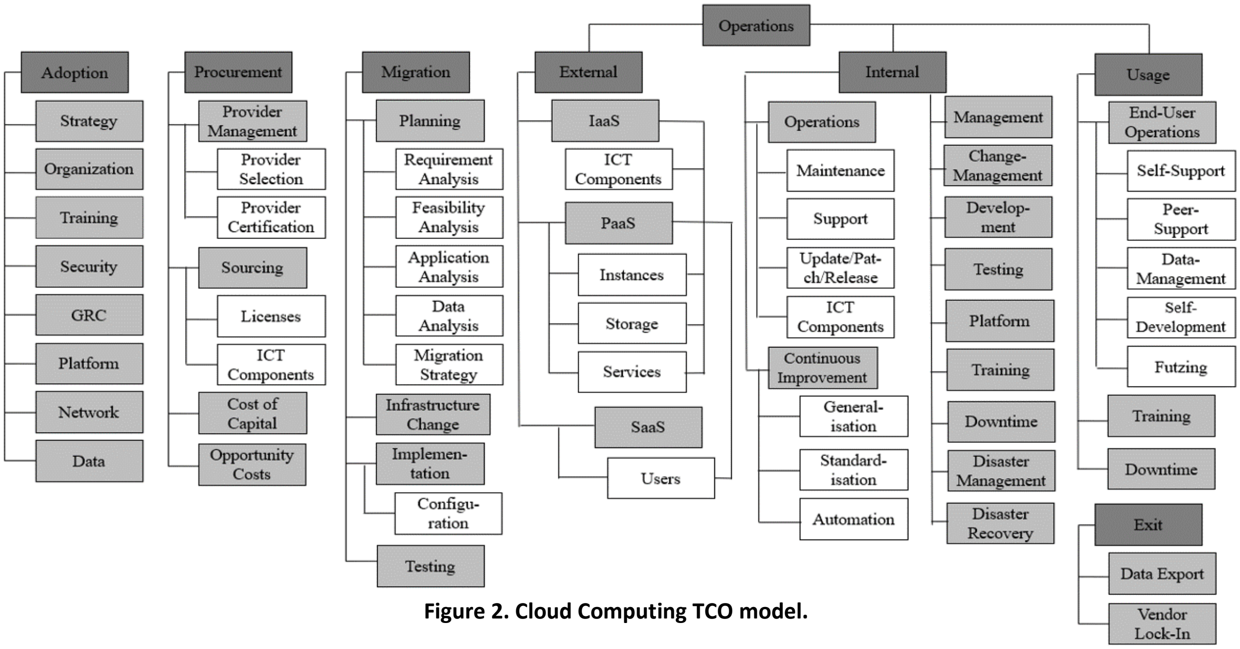

Figure 2. Cloud Computing TCO model.

To calculate the opportunity costs, our model includes an internal calculated yield factor (q_i) (e. g., average ROI). To evaluate the net present value, compound interest must be considered. resulting in q_i^n. The yield factor is expected to grow over time; thus, a growth factor (g^{n-1}) is used. Opportunity costs need to be calculated separately for each cost component due to different billing intervals (Eq. II):

$$\sum\nolimits_i^n C_{d(u),c,r}^{driver} * \prod\nolimits_1^n q_i * g^{i-1}. \qquad \text{(II)}$$

For succeeding equations, the opportunity costs are not listed to avoid illegibility and reduce complexity of the terms.

Along with our descriptive TCO model, we provide generic calculation methods for each cost category using the evaluated cost components and cost drivers. The application of our provided evaluation methods is further demonstrated in our conducted case studies.

4.3. Descriptive CC TCO model

Our descriptive CC TCO model is illustrated in Figure 2. The model covers the cost categories for adoption, procurement, migration, internal and external operations, usage, and exit. We consider the cost categories of adoption and migration to be relevant for CC; thus, we introduce them in extension to the typical TCO categorization.

For CC adoption, we noted that those costs are frequently mentioned in literature but have not yet been examined in TCO models. Therefore, we inquired nine CSPs, 12 research institutes, and the German Federal Office for Security and Information Technology (BSI) for quantifiable research results. Two of the nine CSPs consented to personal interviews. None of the research institutes could deliver any public information. One research institute delivered a list of service partners with better end-customer contact, with one of them willing to conduct an interview. The BSI also reported no information. In addition to the expert interviews, we also examined published CC adoption models to identify cost components. Furthermore, we reevaluated our findings with a car manufacturer as an early adopter of CC. In our descriptive model, we do not label costs as direct or indirect because their relationship depends on the business case.

Subsequently, we describe the relevance, cost components (**bold**), and evaluation methods (*italics*) of each cost category. We provide a table summarizing the associated cost components and drivers for each cost category, followed by a corresponding equation that details our mathematical approach. Each equation contains all associated cost drivers that can be assigned for each cost component during the iterations.

The first cost category of CC adoption covers the costs associated with the shift from on-premises to CC. The CC adoption includes initial planning, development of operational/technical capabilities, modification of processes, and enablement of employees. The CC adoption not only affects the ICT departments, but financial departments are also affected, due to the shift from capital expenditures (CapEx) to operating expenditures (OpEx) and the new pay-per-use pricing models (Abdula et al., 2018; AWS, 2017). Operational departments must gain awareness of CC opportunities to identify valuable business cases.

We aggregate the costs of CC adoption at the cost components of strategy, organization, training, security, GRC, platform, network, and data. The cloud **strategy** covers expenses for evaluating use cases, cloud readiness, the target goal, and an action plan (Abdula et al., 2018; Aggarwal & McCabe, 2009; AWS, 2017; Google, 2019; Mangiuc, 2017; Martens et al., 2012; Minkiewicz, 2014; Shimba, 2010). **Organizational** costs primarily occur for enabling employees and process changes (Abdula et al., 2018; AWS, 2017; Google, 2019). The required knowledge transfer should be performed through systematic **training** to prevent indirect costs of inefficient self-support or peer support (Abdula et al., 2018; Aggarwal & McCabe, 2009; David et al., 2002; Mangiuc, 2017). Expenses are incurred for *internal or external trainers*, *training material*, and the *loss of working hours* for the trained employees (Google, 2019). In the field of **security**, cost accrue for the development of good practices for incident management, infrastructure security, data protection, reporting, and recovery (Abdula et al., 2018; Singh & Jangwal, 2012). Companies ready to use DevSecOps must implement security directly into the toolchain and the services they use.

In the fields of **governance, risk**, and **compliance**, organizational and legal regulatory frameworks must be evaluated, covering IAM, risk management, cost management, reporting, service management, license management, naming conventions, and naming standards (Abdula et al., 2018, 2018; AWS, 2017; David et al., 2002; Google, 2019; Microsoft, 2019; Singh & Jangwal, 2012). In the field of **data**, data technologies, data management, and data governance must be evaluated. For networking, primarily security topics must be addressed.

The development of an **internal cloud platform** provides a technical implementation for most of the mentioned requirements in cloud adoption (Abdula et al., 2018; Aggarwal & McCabe, 2009; Assunção et al., 2012; AWS, 2017; Google, 2019; Minkiewicz, 2014). The key goal is to provide centralized, standardized, and automated cloud services for secure, efficient, and seamless use of CC (Assunção et al., 2012). These services often provide ITSM, IAM, monitoring, logging, data management, and people management. Building an internal cloud platform and design patterns to automate the refactoring of legacy applications should be evaluated to reduce migration time and cost, as well as ensure efficient operations. For CC adoption, most costs are incurred by *working hours spent* and evaluating and implementing new concepts (Martens et al., 2012). *External consulting* is often required in the adoption phase. The best way to predict costs in this phase is to create an action plan and estimate costs for each step with experts, CSPs, and external consultants.

Table 4. Cloud-Adoption cost components/drivers

Cost Components	Cost Drivers
Strategy, Organization, Security, GRC, Data, Network, Platform	time, consulting
Training	time, consulting, material

$$C^{Adopt} = \sum_{i}^{n} \left(p_i^h +* t_i\right) + c_i^{consult} + c_i^{material} \quad \text{(III)}$$

Costs in procurement include **provider selection** and **provider certification**, especially for reviewing the service level agreements (SLAs) (Aggarwal & McCabe, 2009; Assunção et al., 2012; Martens et al., 2012; Minkiewicz, 2014). Costs are determined by the *working hours spent*. A significant amount of cost is incurred for sourcing **licenses**, including difficulties in the field of highly scalable environments (Armbrust et al., 2010; Connor et al., 2014; Lahey & Macdonald, 2010; Marešová et al., 2017; Singh & Jangwal, 2012). *On-off costs* for an **ICT infrastructure** in the form of facility and hardware (server, storage, and network) are barely incurred because only a portion of the hardware is needed for own service consumption and the internal cloud platform (Singh & Jangwal, 2012). The **cost of capital** and **opportunity costs** are assigned to procurement in this model. Their calculation is based on all other cost components, the payment day, and the payment interval. Both costs can be evaluated based on the company's *financial figures*. The calculation is further examined in the mathematical model.

Table 5. Procurement cost components/drivers

Cost Components	Cost Drivers
Vendor Selection Vendor Certification	time
Licenses	fee
Network, Server, Storage	on-off-costs, time
Cost of Capital	interest-rate
Opportunity Costs	financial figures

$$C^{Proc} = \sum_{i}^{n} (p_i * t_i) + c_i^{licence} + c_i^{hw} + \sum_{i}^{n} C_{d(u),c,r}^{driver} * \left(r_i + \prod_{i=1}^{n} q_{\,i} * g^{i-1}\right) \quad \text{(IV)}$$

The cost of cloud migration involves the initial **requirements**- and **feasibility-analysis**. If met, an **application-** and **data-analyses** should be carried out to identify the needed changes (Aggarwal & McCabe, 2009; Connor et al., 2014). For legacy systems, the whole data structure must often be adjusted (Minkiewicz, 2014). The definition of a **migration**

strategy helps prepare and seamlessly fulfill the **application** **migration** to the CC infrastructure (Aggarwal & McCabe, 2009; Connor et al., 2014). Cost fluctuates for lift and shift, refactoring, or a complete redesign. The cost of refactoring can be reduced with automation and preselected design patterns. The cost for the process of lifting the application to a CC infrastructure includes the preparation of the environment and the fee for the *data transfer* (Connor et al., 2014; Kokilavani & Saravanan, 2015; Marešová et al., 2017; Martens et al., 2012; Singh & Jangwal, 2012). Afterward, the costs for the final **implementation**, **configuration**, and testing are incurred (Aggarwal & McCabe, 2009; David et al., 2002; Mangiuc, 2017; Martens et al., 2012; Minkiewicz, 2014). All costs except the data transfer are related to *working hours spent* and the required *consulting* (Martens et al., 2012).

For external operation cost, no evaluation methods are required because only direct cost incurred, and the pricing models of CSP are publicly available. The current cost components are based on Adobe-Cloud, AWS, Azure, Google App Engine, Rack Space, and Salesforce. Costs are viewed individually for infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS).

Table 6. Cloud migration cost components/drivers

Cost Components	Cost Drivers
Requirement-, Feasibility-, Application-, Data-Analysis, Migration-Strategy, Testing, App.-Migration, Implementation, Configure	time, consulting
Data-Transfer	transfer

$$C^{Mig} = \sum\nolimits_i^n \left(p_i^h * t_i\right) + c_i^{consult} + \left(p_i^{dt} * GB_i\right) \quad \text{(V)}$$

For IaaS, costs are incurred for the operation of instances, storage, and other ICT components such as load balancers and CC services. The cost of **instances** depends on the requested server type, categorized by *CPU*, *RAM*, *SLA*, *storage*, *networking*, *operating system*, and the *scalability type*. Categorizing CC resources to server types leads to overprovisioning because the instance size is determined by the most needed characteristics, leaving others at lower untrue to the pay-per-use idea. In addition, unlike the on-demand idea, more inexpensive, non-scalable subscription-based reserved instances were introduced. These instances can cover static load and save costs, whereas on-demand instances can cover peak load.

Further, AWS introduced spot instances, which are currently the most inexpensive and can be withdrawn in a given warning period to handle peak load. A cost optimal provisioning of these instance types is challenging (Seenivasan et al., 2012). For an optimal allocation, computerized algorithms use predictive models, such as the RCRP (Chaisiri et al., 2012).

The cost of storage as for instances depends on predefined **storage** categories. Costs are incurred for *read, write,* and *delete queries*, *data transfer,* and the chosen *backup solution*. Additionally, **ICT components**, such as load balancer, firewall, or bastion host, can be sourced, in addition to supporting **cloud services**.

Table 7. IaaS cost components/drivers

Cost Components	Cost Drivers
Instance	CPU, RAM, SLA, storage, network, OS, scalability, initialization
Storage	capacity, query (read, write, delete), transfer (in/out), back-up
ICT-Comp., Services	CSP

$$C^{IaaS} = \sum\nolimits_i^n \left(p_i^{inst} * t_i\right) + \prod\nolimits_{i,k}^n p_i^{qry} * GB_k + c_i^{service} \quad (VI)$$

The cost of PaaS resemble IaaS, including **instances**, **storage**, **services**, and **usage**. The cost for instances depends on fewer variables, and no additional ICT components can be sourced because the infrastructure is managed by the CSP. Usage costs can be incurred for *users* or *API queries*.

Table 8. PaaS cost-components/-drivers

Cost Components	Cost Drivers
Instance	CPU, RAM, scaling-capabilities,
Storage	capacity, query (read, write, delete), transfer (in/out), back-up
Services	CSP
Usage	users, API

$$C^{PaaS} = \sum\nolimits_i^n \left(p_i^{inst} * t_i\right) + \prod\nolimits_{i,k}^n p_i^{qry} * GB_k + p_i^{user} * x_i^u + p_i^{api} * x_i^a + c_i^{service} \quad (VII)$$

Costs of SaaS are usually **subscription-**based and evaluated with the count of monthly *active users*. Some providers started using consumption-based costing based on the active usage of their services.

Table 9. SaaS cost-components/-drivers

Cost Components	Cost Drivers
Subscription	user
Service Consumption	time, data, runs

$$C^{SaaS} = \sum\nolimits_i^n p_i^{user} * x_i^u + p_i^{session} * t_i \quad (VIII)$$

Internal operation costs are primarily incurred from the *working force* of the operation teams, maintaining applications, the central cloud platform, and internal hardware (Martens et al., 2012). Costs include

management, **maintenance**, **support**, **development**, **testing**, **change** and **release management**, as well as **preventing** and **recovering** from incidences (Aggarwal & McCabe, 2009; Kokilavani & Saravanan, 2015; Martens et al., 2012; Opitz et al., 2008). Management includes administration and auditing (Connor et al., 2014; Singh & Jangwal, 2012). Support levels must be first-, second-, and third-level (Singh & Jangwal, 2012). Costs for the **central cloud platform** include management and the development of foundation services (Marešová et al., 2017; Marešová & Soběslav, 2017). The operation teams further constantly work on achieving higher levels of **generalization**, **standardization**, and **automation** (Connor et al., 2014; David et al., 2002; Google, 2019).

Table 10. Internal operations cost-components/-drivers

Cost Components	Cost Drivers
Management, Maintenance, Support, Development, Change-Mgmt., Testing, Release- Mgmt., Disaster-Mgmt., Disaster-Recovery	time, services
Cloud-Platform	time, ict-comp.
Training	time, consulting, material
Generalization, Standardization, Automation	time
ICT-Comp.	time, ict-comp.

$$C^{Ops} = \sum_{i=1}^{n} \left(p_i^h * t_i\right) + c_i^{consult} + c_i^{material} + c_i^{service} + \left(p_i^e * t_i\right) \quad \text{(IX)}$$

Usage costs are incurred based on efficiency loss caused by the infrastructure change due to a lack of the required capabilities or other circumstances, e. g., CC releases becoming more frequent, constantly causing small obstacles for users due to application changes. As bugs are fixed quicker and new features are implemented faster, an overall increase in efficiency is assumed. If obstacles occur, the user must rely on **self**-support, peer support, or **self-development** to regain working efficiency (Mieritz & Kirwin, 2005). Training must be held to prevent barriers (David et al., 2002). Furthermore, users take time for **data management** or **futzing** (David et al., 2002; Strebel & Stage, 2010). Moreover, **downtime** causes costs by inhibiting the efficiency of users. All costs except training are measured on the *loss of working force* due to barriers expressed in *time waste*, calculated using the *opportunity cost* for the user group.

Table 11. Usage cost-components/-drivers

Cost Components	Cost Drivers
Training	time, consulting, material
Data-Mgmt., Self-Sup.,Peer-Sup., Futzing, Self-Dev., Down-Time	time (opportunity-costs)

$$C^{Use} = \sum_{i=1}^{n} \left(p_i^h * t_i\right) + c_i^{consult} + c_i^{material} \quad \text{(X)}$$

Exit costs include the final **data export**, measured in the required *bandwidth/hours*, and the impacts of **vendor lock-in**, such as the necessary refactoring due to CSP-specific implementation (EMA, 2018; Mangiuc, 2017; Martens et al., 2011, 2012).

Table 12. Exit cost-components/-drivers

Cost Components	Cost Drivers
Data-Export	transfer
Vendor Lock-In	time

$$C^{Exit} = \sum_{i=1}^{n} \left(p_i^h * t_i\right) + \left(p_i^{dt} * GB_i\right) \quad \text{(XI)}$$

All described cost categories with associated cost components and evaluation methods build the foundation of our CC TCO model. The descriptive model discloses the cost structures of cloud usage, thus being of value to anyone assessing the total cost of cloud usage.

5. Case study

To validate our CC TCO model, we conducted eight case studies in cooperation with a car manufacturer. We reviewed five cloud-born and three applications migrated from on-premises into the cloud. First, we introduced the application owners to our model and interviewed them regarding the comprehensiveness of our model. All eight application owners found the model conclusive and could not identify additional cost components.

Exclusive cost components of our model were responsible for 45.3 % of the total costs. The categories of cloud adoption are responsible for 22.3 % and cloud migration for 5.5 % of the total cost. The cost of procurement, an essential part of classical TCO models, but absent in other common cloud assessment methods, is responsible for 16.2 %. Costs for usage and exit barely occurred in our case studies since none of the applications were self-consumed or had not yet left the cloud. In summary, only 41.8 % of the total cost is directly charged by the CSP, emphasizing the importance of the TCO approach.

We outline two of these eight case studies. We selected a use case comparing the cost of a three-tier web application to demonstrate the applicability and usability of our CC TCO model. We illustrate two use-case implementations: a cloud migration (lift and shift approach) and a modern cloud-native serverless architecture. Both scenarios are highly relevant and hosted on a public cloud environment. The lift and shift approach allows a fast and inexpensive way to partly leverage CC for existing applications. In contrast, modern cloud-native infrastructures, such as serverless, fully enable CC advantages and meet modern application lifecycle requirements.

As lift and shift, we consider only applying minor changes to an existing application, still allowing for seamless integration and operation in a public cloud scenario. Contradictory, refactoring existing applications into modern cloud-native infrastructures requires up-front costs. In this scenario, the cost of cloud adoption and the cost of continuous improvement are considered basic investments necessary to procure business in the future. Thus, the investment is not allocated to the lifecycle costs in this scenario. Moreover, no cost for procurement emerged because the CSP was already preselected and certified for three-tier web application usage, and no additional internal hardware or licenses were required. No usage cost emerged because the application is used by external customers and not internally. Exit costs, determined by the data export rate, are insignificant in this scenario.

We outline both case scenarios below in detail. We evaluated the cost categories not considered in this specific use case using the remaining six case studies. Overall, the case studies highlight the modularity of the CC TCO model.

5.1 Serverless scenario

In the serverless scenario, a preexisting application was refactored to become a serverless architecture. Each core function was rewritten as an executable lambda function. The costs of migration (C^{Mig}) are dominated by the refactoring (C^{app}), comprising 71 % of the total migration cost. The remaining 29 % is spent on planning ($C^{plan=req;feas;app;data;strat}$), implementation ($C^{imp}$), configuration ($C^{conf}$), and testing ($C^{test}$). Platform migration costs (C^{mig}), determined by the data transfer rate, were insignificant in this use case.

$$\sum_i^n C^{Mig} = \sum_i^n C^{plan} + C^{app} + C^{imp} + C^{conf} + C^{test} \quad (XII)$$

For external operation costs ($C^{Ex_{Opp}}$), the costs of the environment (C^{Inst}) are determined by the memory used (ram) and the runtime for each lambda function (t_i). Depending on the memory size, there is a free runtime contingent on $\mathrm{tt}_i(ram)$. In the case study, the runtime of the lambda functions did not surpass the threshold, and no costs occurred.

$$\sum_i^n C^{inst} = \sum_1^{12} \mathrm{p}_i(\mathrm{ram}) * (\mathrm{t}_i - \mathrm{t}_i(ram)) \ , with\ (t_i - t_i(ram)) > 0 \quad (XIII)$$

For storage (C^{stor}), S3 buckets are used as an object store in conjunction with Mongo DB as a NoSQL database. The cost of both storage solutions is determined by storage size ($C^{stor_{size}}$), the query count (per 1000) ($C^{stor_{qry}}$), the amount of data transferred ($C^{stor_{traf}}$) and scanned in ($C^{stor_{scan}}$):

$$\sum_i^n C_{i(m)}^{stor} = \sum_m^n (p * x)_i^{size} + \left(p * \frac{x}{1000}\right)_i^{qry} + (p * x)_i^{traf} + (p * x)_i^{scan} . \quad (XIV)$$

For cloud services (C^{serv}), CloudFront was used as a content delivery network, Route53 as a DNS service, Cognito as IAM, and VPC for private hosting and an API gateway. As the pricing of cloud services constantly changes, we do not elaborate on the service costs further in this article. However, in this concrete scenario, all services must be calculated as demonstrated above. Overall, external costs for operation ($C^{Ex_{Opp}}$) consist of 39 % spending on storage (C^{stor}) and 61 % spending on cloud services (C^{serv}). In terms of internal operation costs ($C^{Int_{Opp}}$), professionals are currently in CC adoption. We can assume that administrative tasks ($C^{admin=manage;main;sup}$) become less time-consuming and thus less expensive over time. The cost of DevOps ($C^{devops=dev;test;release}$), with short innovation cycles and the training cost (C^{train}) are assumed to stay similar in the future:

$$\sum_i^n C^{Int_{Opp}} = \sum_i^n C^{admin} + C^{devops} + C^{train}. \quad (XV)$$

Compared to an on-premises architecture, total operating costs are reduced by approximately one-third with the serverless architecture. The total cost, however, is nearly three times higher within four years and is still 50 % higher after 15 years due to extensive efforts spent on refactoring. Table 13 illustrates the overall cost comparison.

Table 13. Cost comparison serverless to on-premises

	4 years	10 years	15 years
Operation Cost	-28 %	-34 %	-34 %
Total Cost	274 %	83 %	47 %

These results indicate that operating costs can be significantly reduced for applications or single business functions with a low-to-medium or infrequent load. For existing applications, however, the costs of refactoring can significantly surpass operational savings. However, for new application development, estimates are similar to those of on-premises applications; thus, overall savings can be expected for serverless architectures. With ongoing training, the migration cost can also be reduced.

5.2 Lift and shift scenario

For the lift and shift approach, C^{Mig} divides into C^{plan} one-fifth of the cost, $C^{app_{mig}}$ at one-sixth and C^{imp}, C^{conf} and C^{test} at one-third of the cost, as only minor adjustments needed to be executed. Efforts spent on platform migration (C^{mig}) are insignificant. In total, C^{Mig} is one-fifth of the cost of the serverless approach.

For $C^{Ex_{Opp}}$, the costs of C^{Inst} vary based on instance specifications and the type of instance, scalability, and availability. We selected different instances for the production, staging, and test environment to optimize the results. The application was not adapted for horizontal scalability; thus, a combination of on-demand instances for peak load and reserved instances for the base load was not considered. In the next step, this could achieve lower cost figures. In that case, reserved instances would be sized to bear peak load. Even at low utilization, they are still slightly less expensive than on-demand instances. For the test environment with infrequent usage the on-demand instances are suited better.

Like other IaaS components (C^{comp}), application and network-load balancers were used. C^{stor} is the same as for the serverless scenario. Moreover, for C^{serv}, the same cloud services are used, except for Cognito. In total, $C^{Ex_{Opp}}$ is 10 % higher than for the serverless scenario. For $C^{Int_{Opp}}$, the same costs as for on-premises can be assumed.

Compared to on-premises, operation costs for lift and shift result in a cost reduction of 25 % (see Table 14). Including the one-time cost of migration, the lift and shift approach is 44 % more expensive within four years, breaks even after 10 years, and is 8 % more cost-efficient after 15 years.

Table 14. Cost transition of lift & shift to on-premises

	4 years	10 years	15 years
Operation Cost	-21 %	-25 %	-25 %
Total Cost	+44 %	0 %	-8 %

The results indicate, for new application development with low or infrequent load, lower overall cost can be achieved with both approaches. Modern application infrastructure can achieve further savings while truly enabling CC advantages. For existing applications, the cost of migration can surpass the savings potential. The less expensive migration, with the lift and shift approach, is amortized faster. For our research partner, cost control is crucial, but their primary interest in CC is to rapidly enhance global customer and employee experience.

6. Discussion and conclusion

In this article, we introduced a holistic TCO model for CC. The discovery and collection of relevant cost components and their evaluation methods contribute to the key findings of our research. This findings led us to the development of a generic and standardizable total cost model for CC. We applied our model to real-world scenarios at a car manufacturer.

In cloud adoption and migration, we revealed key cost components with the help of expert interviews. Our study revealed that, besides the operational advantages of CC, significant financial savings can be achieved with the knowledge of total costing. In addition, CC TCO knowledge can also reduce uncertainty about cost trends and help to overcome hesitation or fear of out-of-control costs explosions. This finding should encourage decision-makers to explore the TCO for CC.

We consider our CC TCO model the first descriptive and most exhaustive total cost model for CC, currently publicly available in practice and research. The model is sufficient to serve as a standardizable base model for further development and customization.

The practical applicability of our model was demonstrated in the case studies we conducted. This outcome also indicates that our model is exhaustive, and the calculation using the mathematical approach is conclusive under the given conditions. The results of our case studies support the current findings. Thus, a significant cost reduction is possible for modern application designs and applications with an irregular or infrequent load deployed on scalable CC infrastructure.

Further research demand arises from the conducted studies. We are currently working on further testing our model and its assumptions. One interest is to identify more efficient evaluation methods to simplify and reduce the efforts involved in the overall calculations. Furthermore, procedure models for the evaluation methods should be introduced to ease the evaluation.

The usability of our model can also be improved through tool support. Therefore, we are working on a prototype that facilitates simulation-based analyses of architectures and multi cloud scenarios. In this context, provider comparisons are also of interest.

References

Abdula, M., Averdunk, I., Barcia, R., Brown, K., & Emuchay, N. (2018). *The Cloud Adoption Playbook*. Wiley.

Aggarwal, S., & McCabe, L. (2009). The Compelling TCO Case for Cloud Computing in SMB and Mid-Market Enterprises. *Hurwitz & Assoicates*.

Alkhanak, E. N., Lee, S. P., Rezaei, R., & Parizi, R. M. (2016). Cost optimization approaches for scientific workflow scheduling in cloud and grid computing. *Journal of Systems and Software*, *113*

Armbrust, M., et al. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50.

Assunção, M. D. de, Netto, M. A. S., Peterson, B., Renganarayana, L., Rofrano, J. J., Ward, C., & Yo-ung, C. (2012). Cloud Affinity: A framework for matching servers to cloudmates. *IEEE*, 213–220.

AWS. (2017). *An Overview of the AWS Cloud Adoption Framework—Version 2*.

Berman, S. J., Kesterson-Townes, L., Marshall, A., & Srivathsa, R. (2012). How cloud computing enables process and business model innovation. *Strategy & Leadership*, *40*(4), 27–35.

Carr, I., & Ittner, C. (1992). Measuring the Cost of Ownership. *Journal of Cost Management*, *6*(3), 42–51.

Chaisiri, S., Lee, B., & Niyato, D. (2012). Optimization of Resource Provisioning Cost in Cloud Computing. *IEEE Transactions on Services Computing*, *5*(2).

Connor, K., Cook, I. P., Porche, I. R. I., & Gonzales, D. (2014). *Cost Considerations in Cloud Computing*. RAND Corporation.

David, S., Schuff, D., Louis, R. (2002). Managing Your Total IT Cost of Ownership. *Commun. ACM*, *45*(1).

Ellram, L. (1994). A taxonomy of total cost of ownership models. *Journal of Business Logistics*, *13*(1).

Ellram, L., Siferd, S. (1998). TCO: a key concept in strategic decision making *Journal of Business Logistics*.

EMA. (2017). *TCO, ROI, and Customer Pain Points of Hybrid Cloud and the Software-defined Data Center*.

EMA. (2018). *Optimize Cloud Costs Through Governance and Collaboration*.

Engates, J. (2017). *Adding up cost before u sign*. Rackspace

Geißdörfer, K. (2009). *Total Cost of Ownership (TCO) und Life Cycle Costing (LCC)*. LIT.

Goasduff, L. (2015). *The Financial Case for Moving to the Cloud*. Gartner.

Google. (2019). *The Google Cloud Adoption Framework*.

Höllwarth, T. (2012). *Cloud Migration*. MITP.

Kokilavani, K., & Saravanan, K. S. (2015). Cost analysis using efficient methods and tools in cloud computing. *International Journal of Multidisciplinary Research and Development*, *2*(6), 282–286.

Lahey, D., & Macdonald, T. (2010). The true TCO of the cloud. *Accounting Today*, *24*(12), 21.

Mahlindayu, T., Rasid, S. Z. A., Alrazi, B., & Roni, R. A. (2014). Cloud Computing Awareness and Adoption among Accounting Practitioners in Malaysia. *Procedia Social and Behavioral Sciences*, *164*.

Malone, T. W., & Crowston, K. (1994). The Interdisciplinary Study of Coordination. *ACM Comput. 26*(1).

Mangiuc, D. (2017). Accountants and the Cloud – Involving the Professionals. *Journal of Accounting and Management Information Systems*, *16*(1), 179–198.

Marešová, P. (2016). Cost Benefit Analysis Approach for Cloud Computing.

Marešová, P., & Soběslav, V. (2017). Effective evaluation of cloud computing investment – application of cost benefit method analysis. *E+M Ekonomie a management*, *20*(2), 134–145.

Marešová, P., Soběslav, V., & Krejcar, O. (2017). Cost–benefit analysis – evaluation model of cloud computing deployment for use in companies. *Applied Economics*, *49*(6), 521–533.

Martens, B., Poeppelbuss, J., & Teuteberg, F. (2011). Understanding the Cloud Computing Ecosystem. *Proceedings of the 10th International Conference on Wirtschaftsinformatik*, 65.

Martens, B., Walterbusch, M., & Teuteberg, F. (2012). Costing of Cloud Computing Services. *45th Hawaii International Conference on System Sciences*.

Mayring, P. (2016). *Einführung in die qualitative Sozialforschung.* Belz.

Microsoft. (2019). *Cloud Adoption Framework for Azure*.

Mieritz, L., & Kirwin, B. (2005). Defining Gartner Total Cost of Ownership. *Gartner*.

Minkiewicz, A. (2014). *Costs of Migration and Operation in the Cloud, PRICE Systems,*.

Möhring, W., & Schlütz, D. (2013). Standardisierte Befragung: Grundprinzipien, Einsatz und Anwendung.

Opitz, A., König, H., & Szamlewska, S. (2008). What Does Grid Computing Cost? *Journal of Grid Computing*, *6*(4), 385–397.

Raza, H., Adenola, F., Nafarieh, A, & Robertson, W. (2015). The Slow Adoption of Cloud Computing and IT Workforce. *Procedia Computer Science*, *52*.

Rightscale. (2018). *Cloud Computing Trends: 2018 State of the Cloud Survey*.

Robey, D., Boudreau, M., & Rose, G. M. (2000). Information technology and organizational learning. *Accounting, Management and Information Technologies*, *10*, 125–155.

Seenivasan, D., Gayathri, P., Anitha, R., & Dhivya, P. (2012). Optimization of Resource Provisioning Cost in Cloud Computing. *International Journal of Computer Applications*

Shimba, F. (2010). *Cloud Computing: Strategies for Cloud Computing Adoption*. DIT.

Singh, S., & Jangwal, T. (2012). Cost breakdown of Public Cloud Computing and Private Cloud Computing and Security Issues. *IJCSIT, Vol 4*(No 2), 17–31.

Snyder, L. H., Marc E. (2017). *Navigating Uncertainty, Harvey Nash / KPMG CIO Survey*. KPMG.

Strebel, J., & Stage, A. (2010). *An economic decision model for business software application deployment on hybrid Cloud environments*. MKWI.

Sujeet Kumar, S., Al-Badi, A. H., Govindaluri, S. M., & Al-Kharusi, M. H. (2016). Predicting motivators of cloud computing adoption: A developing country perspective. *Computers in Human Behavior*, *62*.

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a literature Review. *MSI Quarterly*, *26*(2), xiii–xxiii.