Document name	Code	Segment	Created by
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > DevOps	. Cloud-native application (CNA) characterizes a distributed, elastic, and horizontal scalable system consisted of (micro) services that segregates state in a minimum of stateful components [14]. CNA may be developed with combination of best languages and managed through the DevOps processes.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > Performance	Intermediate layer or gateways are implemented using multi-agent, SAO, RESTful, or fog computing technologies, whereas backend computing or cloud com-puting layer is responsible for big data analytics and high-performance computing. This section has shed some light to explain the cloud computing trends that will help in planning the cloud deployment.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > Performance	These terms are as follows virtual, service, data, multi, dynamic, mobile, performance, secure, application, and migration consisting of 30, 19, 18, 13, 11, 11, 11, 11, 10, and 10 related terms, respectively. These terms form a cloud migration tuple and are explained in the environment of cloud computing in the following paragraphs.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > Portability	Service: It is also a unique characteristics of cloud computing as it is delivered in the form of services. Some important terms related are service orientation, availabil-ity, innovation, migration, portability, and replication. Service-oriented architecture deserves a mention here as well. And service-level agreements (SLA) are also the most researched topic in this cluster, and some related terms are SLA assurance and monitoring.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > Portability	Data: Data is the primary resource of information technology. Many important terms are mentioned with data such as locality, distribution, portability, migration, security, mining, and deduplication. Data centres (DC) are the basic infrastruc-tural component of cloud computing and related terms were DC networking and management.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > multi-tenant	Multi: Multi-term has many-faceted implication in cloud computing. Most com-mon of all is multi-tenant environment. Multi-agent system (system of multiple interacting intelligent system) [25] has also become significant in the perspective of cloud computing. This term is also associated with replication and redundancy such as multicast, multi-cloud, and multi-gateway system.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > Performance	Performance: This is also an important criterion for the success of cloud comput-ing migration. This term has been given as a precursor to the migration such as per-formance matrix, modeling, prediction, and testing. Similarly, in the post migration stage, performance attributes are analyzed, evaluated and managed.	Ivon Miranda Santos
Ahmad2020-Cloud_Computing_Trends_and_Cloud_Migration_Tuple	CLOUD MANAGEMENT > Performance	These elements are virtual, service, data, multi, dynamic, mobile, performance, secure, application, and migration. These terms have been described in the previous section. In addition to awareness of cloud computing tends, this cloud migration tuple will help in understanding intricacies of migration to the cloud.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > multi-tenant	Multi-Tenancy is referred as resource sharing and associ-ated risks of data confidentiality and integrity. Multi-Tenancy has been called a serious concerning issue for professionals in cloud computing. Professionals' understanding about attack surfaces and attack vectors is most important [16]. In [17], VOLUME 9, 2021 57793B. Alouffi et al.: Systematic Literature Review on Cloud Computing Security an increased number of cloud computing service users resulted in data security and privacy threats	Ivon Miranda Santos

Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Monitoring	Cloud providers should have control over the auditing and maintenance of cloud services. It can be achieved through the 57794 VOLUME 9, 2021B. Alouffi et al.: Systematic Literature Review on Cloud Computing Security consecutive monitoring of users' logs and administrative ses-sions with those parts of cloud services affected by users.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > multi-tenant	Cloud service managers monitor tenants' malicious attacks when they target different services to flood the other tenants' virtual machines [26]. In [27], researchers proposed that the pro-tection of multi-tenancy areas was the major focus of cloud vendors. Cloud services are well-known approaches for product traceability of industrial systems to provide data integration and sharing services [28].	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Performance	The proposed approach's evaluation is performed on a real testbed to show the performance and accuracy in detecting the attacks over the encrypted traffic.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Monitoring	The edge cloud services have their local isolation from the cloud, and security monitoring becomes inappropriate. Malicious users target geographically divided manage-ment [48]. Besides this, log data cannot be stored securely, and forensic information can be removed from the cloud to prevent forensic investigation.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Performance	Data storage, data confidentiality, and data availability in cloud services are among the other identified threats to cloud computing security. For example, outsourcing stored data at the cloud requires an additional security layer to strengthen the data confidentiality. Research [51] introduced a ''cloud storage based on ID-based encryption'' (CS-IBE) with the one file access policy and a user's identity proposed to be used as an encryption key. Although this approach simplifies the key management issues, it shows limited performance for data confidentiality.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Performance	A survey study [53] finds that the top five cloud providers, including Amazon, Azure, Adobe, Google cloud platform, and VMWare, are efficient in their cloud services' data secu-rity feature. Reliability and performance are among other features. To measure the cloud providers' trustworthiness is still an issue for researchers, and a customer cannot judge it without appropriate tools.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > multi-tenant	When multi-tenant run their appli-cations on the same host resource, security, and privacy issues arise from their applications. Existing literature on com-mercial service providers (CSPs) reveals that cloud service models are involved in hampering security concerns [55].	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Monitoring	In a cloud model, policies are con-figured to monitor end-users devices and cloud computing services [32]. Service level agreements are one of the obsta-cles that contain the scope of cloud services.	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Portability	Data unavailability, vendor lock-in and insufficient measures of security create concerns for users. Consumers show their concern over the lack of interoperability and stan-dards. Portability features are provided with limited offers. Therefore, evaluation of SLAs benefits the service providers in terms of legal actions, while minimal assurance of data pro-tection for consumers is specified to reflect the consumers' requirements at the right time [13].	Ivon Miranda Santos
Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Performance	The experimental result showed that better performance concerning throughput, response time, accuracy, and total change security features has been achieved. Security features in the blockchain-based cloud include authentication, network security, access mech-anism, and privacy method	Ivon Miranda Santos

Alouffi2021-A_systematic_literature_review_on_cloud_computing_s	CLOUD MANAGEMENT > Performance	Recent work proposes a novel blockchain scheduler that is better in performance compared with other cloud scheduling models. One of the limitations of the proposed research is its simulation, which could be near a realistic one but cannot be 100 percent applicable to real-world case studies. Therefore, researchers can consider realistic scenarios with different cloud clusters and cloud technologies to develop a multi-cloud system in future works	Ivon Miranda Santos
Ardagna2015-Cloud_and_multi-cloud_computing_Current_challenges_	CLOUD MANAGEMENT > Performance	The choice of the application architec-ture matching and fully exploiting the characteristics of the underlying Cloud environments is also critical [2], [3]. At the infrastructural layer, resource contentions lead to unpredictable performance [4] and additional work for resource management [5], automated VM and service migration [6] is still needed. Also networks are frequently the Cloud bottleneck and data center energy management is very critical [7].	Ivon Miranda Santos
Ardagna2015-Cloud_and_multi-cloud_computing_Current_challenges_	CLOUD MANAGEMENT > Cost	To cope with such challenges the adoption of multi-Clouds [8], has been advocated by many researchers, since deploying software on multiple Clouds overcomes single provider un-availability and allows to build cost efficient follow the sun applications.	Ivon Miranda Santos
Ardagna2015-Cloud_and_multi-cloud_computing_Current_challenges_	CLOUD MANAGEMENT > Performance	Moreover, models can also be used to reason about the QoS properties of an application [2] and to support design-time exploration in order to identify the Cloud deployment configuration of minimum cost, while satisfying QoS constraints [3]. Finally, models can be kept alive also at runtime to trigger dynamic adaptation [10], [5], providing QoS guarantees even under workload fluctuations, virtualized systems performance degradations, or failures. ACKNOWLEDGMENT	Ivon Miranda Santos
Ardagna2015-Cloud_and_multi-cloud_computing_Current_challenges_	CLOUD MANAGEMENT > Cost	Moreover, models can also be used to reason about the QoS properties of an application [2] and to support design-time exploration in order to identify the Cloud deployment configuration of minimum cost, while satisfying QoS constraints [3].	Ivon Miranda Santos
Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Cost	Abstract. Migrating applications to the cloud is rapidly increasing in many orga-nizations as it enables them to take advantages of the cloud, such as the lower costs and accessibility of data.	Ivon Miranda Santos
Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Cost	Moving an application to the cloud enables organization to make use of the advantages of the cloud like elasticity [2], lower costs, and accessibility of data.	Ivon Miranda Santos
Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Cost	Despite to momentum to shift to multi cloud, the cost-benefit analysis models illustrating the business impact of cloud adoption are still a significant risk factor [4].	Ivon Miranda Santos
Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Performance	[7] shows a framework to create a generic reference for process of cloud migration while Iqbal et al. [8] discusses different cloud migration strategies and models, right from evaluating performance to choosing a cloud provider. Iyoob et al.	Ivon Miranda Santos
Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Cost	Perform Text Mining on Data for Each Application. In this step, we first collect meta-data for each application. For each client application that the client wants to migrate to cloud, we screen applications and collect application metadata features like number of users, geography of users, investment on application, security and compliance agree-ment, geography of database where data is stored, software platforms, authentication server, server, hardware, network setup, reliability, scalability, etc. Also, we collect client request for features that should be in the cloud for this application, e.g., security compliance, cost budget for cloud usage.	Ivon Miranda Santos

Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Cost	After creating the weighting function, we can easily prepare a solution comprising of services from different cloud providers with minimum cost and maximum coverage of services.	Ivon Miranda Santos
Asthana2021-Multi-cloud_Solution_Design_for_Migrating_a_Portfol	CLOUD MANAGEMENT > Cost	This is done by preparing a matrix of applications versus different service cloud providers on metrics like cost, QoS, coverage, security etc. as illustrated in Fig.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Performance	Cloud computing will enable universi-ties with limited budgets to benefit from information services without making any new financial investments for informa-tion and communications technology (ICT) resources. With cloud applications in higher education, knowledge can be managed effectively to increase academic performance, effectiveness, and efficiency in universities. Universities have been also affected by the COVID-19 crisis recently.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Cost	Cloud computing will enable universi-ties with limited budgets to benefit from information services without making any new financial investments for informa-tion and communications technology (ICT) resources.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Performance	• On-demand Self-service: The ability to use informa-tion resources automatically in accordance with the usage rates and performances of computing resources. • Broad Network Access:	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Performance	• Measured Service: The ability of users to optimize resource utilization based on their use of cloud resources and their performance. With this feature, cloud services can be charged at the same time.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Cost	Behind the desire to benefit from cloud computing in universi-ties lies mainly the willingness of universities to use their finan-cial resources cost-effectively and efficiently.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Cost	Transition from existing legacy systems to cloud platforms is difficult and costly (Gholami et al.,	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Monitoring	There are different strategies for transition to the cloud in the literature. While some studies indicate that the transition to the cloud consists of the evaluation, design, establishment, imple-mentation and operation, and monitoring steps (Şanlı, 2011), some others emphasize that this transition consists of the steps of learning, organizational evaluation, pilot cloud implemen-tation, cloud preparation assessment, cloud dissemination strategy, and continuous cloud improvement strategy (Wyld, 2009).	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Performance	Private universities, which entered Turkish Higher Education in 1984 for the first time, have taken their places as a public institution, with their increasing numbers and growing student capacities. State and private universities have some structural similarities and dif-ferences in terms of management, finance, education and training structure, quality and qualified graduates, and perfor-mance status. In the current system in Turkey, IT departments of universities are responsible for meeting the ICT needs of the universities.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Cost	The cost-effectiveness of cloud computing is one of its most prominent advantages that causes all organizations to adopt and use cloud computing.	Ivon Miranda Santos
Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Cost	The purpose of the framework is to meet the expectations of universities in a cloud environ-ment in a cost-effective manner.	Ivon Miranda Santos

Aydin2021-A_Study_of_Cloud_Computing_Adoption_in_Universities_a	CLOUD MANAGEMENT > Performance	Cloud computing can also play a very important role in quickly solving the prob-lems faced by universities during this coronavirus period. With the adoption of the cloud, management of knowledge can be done in an effective way to achieve high academic performance and efficiency in universities. The issue of cloud computing use by universities is clearly very important and would continue to be important, especially in the post-COVID-19 world.	Ivon Miranda Santos
Baby2015-Multicloud_architecture_for_augmenting_security_in_clo	CLOUD MANAGEMENT > Performance	The main obstacle that stands as a barrier for opting clouds, despite its benefits are its security challenges. Multicloud architecture is a solution that assures better security and performance at a nominal cost. There are different approaches used for storing and processing data in a multicloud environment, this paper is a study on the different multicloud architectures and their benefits and limitations.	Ivon Miranda Santos
Baby2015-Multicloud_architecture_for_augmenting_security_in_clo	CLOUD MANAGEMENT > Performance	Replication of application and partitioning of application exposes the data to malicious cloud providers thus compromising the confidentiality of data even though they assure the correctness of the computations. Partitioning logic and data however provides reasonable confidentiality but, the performance of the computation is not guaranteed as there is no specific method on how the data is obtained by the cloud from user for calculations and how they need to be integrated with the logic available with them. Business readiness is a measure on how far the approach is ready for real life situations [3], while ease of use refers to the practicality of using these methods in daily life.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Monitoring	Developers and users situated in low-resource settings are faced with unique contextual and infrastructure challenges when accessing and consuming cloud-based services. In low-resource settings, access to cloud services and platforms is usually characterized by low-end computing devices and often unreliable and slow mobile broadband Internet connections. In this paper, we discuss key challenges for developing for and accessing cloud services in resource constrained settings, namely, (1) Frequent Internet partitions and bandwidth constraints, (2) Data jurisdiction restrictions, (3) Vendor lock-in, and (4) Poor quality of service. Inspired by these challenges, we propose a set of important design considerations and properties for a resilient multi-cloud service layer, that includes: (1) Containerization and orchestration of applications, (2) Application placement and replication, (3) Portability and multi-cloud migration, (4) Resilience to network partitions and bandwidth constraints, (5) Automated service discovery and load balancing, (6) Localized image registry, and (7) Support for platform monitoring and management. We present an implementation and validation case study, Crane Cloud, an open source multi-cloud service abstraction layer built on-top of Kubernetes that is designed with inherent support for resilience to network partitions, microservice orchestration (deployment, scaling and management of containerized applications), a localized image	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	Portability and multi-cloud migration	Ivon Miranda Santos

Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	We present an implementation and validation case study, Crane Cloud, an open source multi-cloud service abstraction layer built on-top of Kubernetes that is designed with inherent support for resilience to network partitions, microservice orchestration (deployment, scaling and management of containerized applications), a localized image registry, support for migration of services between private and public clouds to avoid vendor lock-in issues and platform monitoring. We evaluate the performance and user experience of Crane Cloud by implementing and deploying a computational and bandwidth intensive machine learning system. The results show lower response times of the system on Crane Cloud compared with hosting on other public clouds.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	, 2010), private sector and government (Kshetri, 2010; Zhang and Chen, 2010) domains. Major global cloud platforms have their data centers concentrated in countries and regions where there is stable infrastructure and high reliability, performance and low latencies are guaranteed. Such cloud platforms make two broad assumptions:	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	4. Evaluation of performance and user experience of Crane Cloud platform by implementing and deploying a computational and bandwidth intensive machine learning system that shows lower response time compared when hosted on other public clouds. 2.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	Most cloud solutions do not provide con-trols over where data should be stored and in cases where there is no infrastructure presence, users have to make exceptions at the expense of prescribed hosting recommendations. Cloud providers also distribute content over spatial infrastructure located in different regions to maintain the cloud Quality of Service (QoS) along dimensions of performance, availability and reliability. This leads to silos of data spanning different geographical regions that users may have no idea or control of.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	Public clouds offer provider-specific proprietary solutions to meet the market demands and this has resulted in an interoperability, integration and portability downside across the cloud divide.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	The longer the distance, the higher the number of intermediary links which can act as failure points (bottlenecks) and potentially introduce network packet losses. Furthermore, there are applications that are delay-sensitive and these require optimal and stringent quality of service parameter values such as low latency, low jitter and minimal or no packet loss for best performance. Currently, public cloud providers attempt to solve this challenge by moving services closer to the user.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	, 2016; Knoche and Hasselbring, 2019). Driven by application features such as scalability, agility, performance and fault-tolerance, microservice architectures involve autonomous software development teams independently working to build loosely coupled application fea-tures and employing collaborative workflows and automation tools from version control systems to full scale production deployment (Has-selbring and Steinacker, 2017). A number of popular technology com-panies such as Uber, Spotify, Netflix, Amazon and Ebay are now using microservices at the core of their business processes and have achieved differing levels of reliability and scalability in their services (Knoche and Hasselbring, 2019).	Ivon Miranda Santos

Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	This has been spurred by the intro-duction of Software Developers (Dev) and IT Operations (Ops) generally termed as DevOps model for software development: an end-to-end model for fast delivery of reliable applications and services involving cultures(by and for the people), automation (testing, feedback, deployment and performance bench-marks), quality measurement and sharing of ideas, processes and tools (Hüttermann, 2012). In an incremental migration and ar-chitectural refactoring of a commercial mobile backend (mono-lithic application) as a service, Balalaie et al.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > DevOps	This has been spurred by the intro-duction of Software Developers (Dev) and IT Operations (Ops) generally termed as DevOps model for software development: an end-to-end model for fast delivery of reliable applications and services involving cultures(by and for the people), automation (testing, feedback, deployment and performance bench-marks), quality measurement and sharing of ideas, processes and tools (Hüttermann, 2012).	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > DevOps	In an incremental migration and ar-chitectural refactoring of a commercial mobile backend (mono-lithic application) as a service, Balalaie et al. (2016) noted that the microservice architecture is an enabler for use of DevOps. Automation does not imply management overhead on introduction of new applications (Dragoni et al., 2017) but rather increased agility and reliability. In low resource settings, further automation of the containerization and deployment processes can ultimately enhance adoption of cloud computing (Mwotil et al., 2022).	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	The lightweight nature of containerized applications coupled with func-tional and configuration encapsulation facilitates replication and porta-bility, are cost-efficient and have a reduced overhead on the operation and maintenance line. It is for these reasons that containers emerged as the most suitable packaging toolset for microservices	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	, 2017; Knoche and Hasselbring, 2019). Despite its complexity such as in installation, Kubernetes is the most widely adopted and powerful container orchestration tool owing to its immense scalability, performance and advanced automation features. It has inbuilt mon-itoring and logging libraries and processes which are lacking in the other tools (Modak et al.,	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	Static replication strategy where the number of nodes and repli-cas is defined beforehand 2. Dynamic replication strategy where replicas are automatically created or destroyed based on changes such as user density, per-formance, storage utilization, loadbalancing features and band-width consumption. For the rest of this section, we shall consider the four microser-vices for the Automated Plant Disease Diagnosis (APDD) system:	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Cost	User defined and cost-sensitive replication policies The cloud service should operate only within the user-defined repli-cation limits but also ensuring minimal replication costs between the clusters and the target nodes.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	In the deployment of APDD, the user may specify a replication limit of 3: the cloud service should ensure that there are three instances of APDD microservices available at all times. Additionally, the replication approach in cases of downtime should consider the replication costs such as the impact on the network performance whenever provisioning is required. It should also be noted that the cloud service provider may impose restrictions based on, for example resource availability, which the user adheres to but regardless, the user will operate on a higher level of abstraction.	Ivon Miranda Santos

Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	Quality of service (QoS) and high availability Some applications in heterogeneous clouds have higher QoS re-quirements in comparison with others for example a critical medical diagnosis system that should operate under stringent availability and consistency constraints. In distributed cloud environments that sup-port service geo-replication, maintaining consistency and performance consecutively is desirable but not fully achievable according to the CAP's theorem (Brewer, 2000). Consistency may be achieved but at the expense of degraded system performance. A number of research works have been published in the line of QoS and high availability.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	This model also uses an immediate update propagation variant for data access to ensure that users only access a fully replicated service. The success of this approach requires a highly available master site and geographically nearby secondary sites as the updates directly impact network performance. Boru et al.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	At a certain point in time, the APDD consumers may be in South Africa and this requires resource reconfiguration and pro-visioning to serve the new user environment. Location-aware systems should strive to achieve fairness in cost vis-a-vis performance (Shi et al., 2020) in multi-cloud setups.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	3.3. DC 3: Portability and multi-cloud migration Portability in cloud computing can be defined as the ability for movement of applications, workloads, processes and data from one cloud environment to another with least disruption, whether manu-ally or automatically. The least disruption should translate to lowest possible cost, effort and time. The movement of one service, such as the one instance of the prediction microservice for automated plant diagnosis system from Cluster 2 to Cluster 1 as shown in Fig. 5, should cause minimal or no downtime and should not compromise the QoS attributes tagged to overall operation of the system. As noted earlier, cloud computing offers significant benefits such as scalability, disaster recovery, mobility and cost reduction in operation of an organization's IT infrastructure. This is evidenced in the introduction of different cloud computing technologies and deployments to make it easy for organizations to embrace and adopt this new wave of handling com-pute, storage and network workloads. One of the pertinent issues in the adoption of cloud computing is vendor lock-in (lack of portability and interoperability across cloud platforms) where providers work with specific technologies such as tools and programming interfaces. Given the different deployment models and the cloud service models, orga-nizations should be able to move cloud services from one provider to another without worries of complexities and infrastructure	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	Bozman and Chen (2010) identified standardized programming in-terface, abstraction layers and management capabilities as some of the key enablers for portability and service migration between cloud providers.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	However, adopting a standardized approach to portability is a myth as it is extremely difficult for providers to agree and adopt a unified set of standards. This requires major rework of the proprietary APIs and file formats, and this also destroys the competition spirit which has been very effective in delivering high quality cloud services for the market (Gonidis et al., 2012).	Ivon Miranda Santos

Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	Most of the research work geared to support portability across different cloud environments such as mOSAIC16 (Open-Source API and Platform for Multiple Clouds), Open Cloud Computing Interface (OCCI) have focused on abstraction layers and management tools and container-centric solutions.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	Containerization allows an application to be built once, placed inside a container image or series of images for a multi-service application and running it on any host operating system that supports the containerization technology in perspective such as Docker. It should however be noted that achieving full portability out-of-the-box and application storage persistence using containers has some limitations such as no support for cross operating system support - a containerized Linux application requires a Linux host operating system, a windows one requires a Windows operating system.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	a distributed client querying the central registry for location and in-formation of other services either using a client-side discovery (a client queries the service registry, selects an available instance and makes a request) or server-side discovery (a router acting on behalf of the client queries the service registry and forwards the request to an available instance) implementation. The popular centralized ser-vice registry/discovery solutions include Netflix’s Eureka,18 CoreOS’s highly available etcd19 key–value distributed datastore, consul20 and Apache ZooKeeper.21 The major drawbacks of centralizing the service registry/discovery are the introduction of points of failure, performance bottlenecks and possible network congestion. Distributing the nodes providing these services and ensuring there are multiple instances in a consistent way usually suffices.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > DevOps	To fully effect containerization, an image of the application is created and pushed to a local or remote image registry through which a user, such as a DevOps engineer, can now pull and create an instance of it in a container host. A container image registry provides a storage location and distribution portal for images some with multiple versions identified by tags. Docker Hub,22 Google’s GCR,23 Azure ACR24 and Amazon ECR25 are some examples of popular public/private image repositories.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	Compared to traditional monolithic applications, moni-toring of microservice applications requires intensive service reporting features especially given their distributed nature (services run as in-dependent processes on possibly geographically different hosts) and dynamic behavior. Monitoring aids users in understanding the over-all health of an application, gain insight into the performance of constituent services of an application and to ensure that APIs are available and performing as expected. The monitoring metrics divided into platform/host (CPU, RAM, threads and database connections) and application metrics (service availability, service and API endpoint latency, success of API endpoints, API endpoint response times, API request clients, errors and exceptions) should be collected at each stage of the deployment pipeline.	Ivon Miranda Santos

Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	In addition, a real-time moni-toring component of a production-ready microservices application to detect current and imminent failures due to changes in key metrics is necessary. A number of monitoring tools and frameworks exist but most are either native (Amazon Cloudwatch,26 Azure Monitor,27 Google Cloud's Operations Suite28) or virtualization type specific (such as cAdvisor29) or commercial (such as Datadog30 and Dynatrace.31 Given a plethora of monitoring options available and complexities of monitoring mi-croservice applications, a monitoring framework should be designed to capture, report and alert stakeholders on performance and failures of an application based on critical metric data. Noor et al.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Monitoring	A number of monitoring tools and frameworks exist but most are either native (Amazon Cloudwatch,26 Azure Monitor,27 Google Cloud's Operations Suite28) or virtualization type specific (such as cAdvisor29) or commercial (such as Datadog30 and Dynatrace.31 Given a plethora of monitoring options available and complexities of monitoring mi-croservice applications, a monitoring framework should be designed to capture, report and alert stakeholders on performance and failures of an application based on critical metric data. Noor et al. (2019) presents a framework for monitoring microservice-oriented cloud ap-plications in heterogeneous virtualization environments. It is composed of mainly two components: a monitoring agent (a cloud platform-independent software component that collects information from a microser-vice) and a monitoring manager (a software component that receives monitoring information from agents in heterogeneous cloud environments).	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Monitoring	4.1.4. Platform and service monitoring support Monitoring is integral to the overall operation of Crane Cloud in terms of infrastructure and the distributed services hosted to ensure the QoS attributes are in check. The infrastructure includes the nodes while the services are the client applications and supporting tools. Monitoring coupled with an alert system also ensures that possible failures are averted early on before turning catastrophic. More specifically, the cluster monitoring involves the state of the cluster (collection of nodes) which is a constituent of node resource utilization parameters such as network bandwidth, disk utilization, CPU, and memory utilization while the service metrics include CPU, network, and memory usage irrespective of the nodes they are running on.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Portability	This enables organizations to work with data-driven and legacy applications while leveraging the portability, scalability and highly available features of containers. Traditionally, Kubernetes used to provide support for manual attachment of cloud-backed storage to applications limiting usage outside the cloud provider but cloud native storage solutions have now been advanced.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	Experiment setup The purpose of the experiment was to evaluate the performance and user experience of mcrops when deployed on a public cloud (AWS) compared to the deployment on the Crane Cloud platform. We consider the response time metric as an important metric for the measurement of the quality service and user experience.	Ivon Miranda Santos

Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Performance	Connection/Sampler unet.mcrops.org mcrops.cranecloud.io 2G (45 Kbps) A1,B1 A2,B2 A1,B1 A2,B2 3G (4 Mbps) 4G (8 Mbps) WiFi (10 Mbps) popular performance testing tool. Specifically we used JMeter tool to measure the response times of the two application setups against uploaded images over different mobile/wireless connections for a user situated in a bandwidth constrained setting.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > Monitoring	Based on these challenges, we enumerated a number of design considerations and properties for a resilient multi-cloud service layer that would form the foundation for Crane Cloud. From easing terminal complexities of operating a cloud service, desirable scaling, availabil-ity, migration and loadbalancing to platform monitoring, Crane Cloud tries to provide an all-inclusive solution that best fits the resource constrained compute environment.	Ivon Miranda Santos
Bainomugisha2022-Crane_cloud_A_resilient_multicloud_service_abs	CLOUD MANAGEMENT > DevOps	Hüttermann, M., 2012. DevOps for Developers. A Press.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Microservices Architecture Enables DevOps: An Experience Report on Migration to a Cloud-Native Architecture	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Abstract The microservices architecture is one of the first service-based architectural styles that has been introduced, applied in practice, and become popular when the DevOps practices gained momentum in the software industry. Migrating monolithic architectures to cloud-native architectures like microservices brings in many benefits such as flexibility to adapt to the technological changes and independent resource management for different system components.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Here, we report our experiences and lessons learned during incremental migration and architectural refactoring of a commercial Mobile Backend as a Service to microservices. We provide a detailed explanation of how we adopted DevOps and its practices, and how these practices facilitated a smooth migration process. Furthermore, we have taken a step towards devising microservices migration patterns by wrapping our experiences in different projects into reusable migration practices.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	1 Introduction According to Google Trends, both DevOps and microservices are recognized as growing concepts and interest-ingly with an equal rate after 2014 (cf. Figure 1). Although DevOps practices can also be used for monoliths, but microservices enables an effective implementation of the DevOps through promoting the importance of small teams [1]. Microservices architecture is a cloud-native architecture that aims to realize software sys-tems as a package of small services, each independently deployable on a potentially different platform and technological stack, and running in its own process while communicating through lightweight mechanisms like RESTful or RPC-based APIs (e.g.,	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Here we explain our experiences and lessons learned during incremental migration of the Backtory (http: //www.backtory.com/) platform, a commercial Mobile Backend as a Service (MBaaS), to microservices in the context of the DevOps practices. Microservices helped Backtory in a variety of ways, especially in shipping new features more frequently, and in providing scalability for the collective set of users coming from different mobile application developers.	Ivon Miranda Santos

Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Furthermore, we report on a number of migration patterns based on our observations in different migration projects. These patterns can be used by either practitioners who aim to migrate their current monolithic software systems to microservices, or system consultants who help organizations to prepare migration plans for adopting the DevOps practices in the migration process towards the microservices. To save space, the details of these migration patterns are provided in a supplementary technical report [4].	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Google Trends report for DevOps and Microservices keywords DevOps and Microservices (SIDEBAR) DevOps is a set of practices [1] which not only aims to de-crease the time between applying a change to a system and the change being transferred to the production environment, but also insists on keeping the software quality in terms of both code and the delivery mech-anism as one of the key elements in the development process. Any techniques that enables the mentioned goal is considered as a DevOps practice [1, 5]. Continuous Delivery (CD) [6] is a DevOps practice that enables on-demand deployment of a software to any environment through the automated machinery. CD is an essential counterpart for microservices as the number of deployable units increases. Yet another critical DevOps practice is the Continuous Monitoring (CM) [7] which not only provides developers with performance-related feedbacks but also facilitates detecting any operational anomalies [5]. 2 The Architectural Concerns for Microservices Migration	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > Performance	CD is an essential counterpart for microservices as the number of deployable units increases. Yet another critical DevOps practice is the Continuous Monitoring (CM) [7] which not only provides developers with performance-related feedbacks but also facilitates detecting any operational anomalies [5]. 2 The Architectural Concerns for Microservices Migration	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > Performance	Figure 2: (a) The architecture of Backtory before the migration, (b) Transforming DeveloperData to a Service, (c) Introducing Configuration Server, (d) Introducing Edge Server, (e) Introducing Dynamic Service Collaboration, (f) Introducing ResourceManager, (g) Target architecture of Backtory after the migration, (h) The final delivery pipeline, (i) The monitoring and performance feedback infrastructure, (j) DevOps team formation 4	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Figure 2: (a) The architecture of Backtory before the migration, (b) Transforming DeveloperData to a Service, (c) Introducing Configuration Server, (d) Introducing Edge Server, (e) Introducing Dynamic Service Collaboration, (f) Introducing ResourceManager, (g) Target architecture of Backtory after the migration, (h) The final delivery pipeline, (i) The monitoring and performance feedback infrastructure, (j) DevOps team formation 4	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	3 The Migration Process The changes we made throughout the migration process included (i) architectural refactorings and (ii) some necessary changes to enable DevOps. 3.1 Architectural Refactoring	Ivon Miranda Santos

Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > Portability	By utilizing containers, we could deploy service instances with lower overheads than the virtualization, and with a better isolation. Another major benefit is the portability since we could deploy anywhere that supports containerization without any changes to our source codes or container images. Docker is a tool for the containerization of applications [12].	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	This change minimizes the inter-team coordination which despite of a more complex testing strategy, enables forming smaller teams as a DevOps practice. 3.1.2 Transforming DeveloperData to a Service	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > Performance	3.2.1 Filling the Gap Between the Dev and Ops via Continuous Monitoring In the context of microservices, each service can have its own independent monitoring facility owned by the Ops team and thereby, enables independent flow of per service performance information to the development. Appropriate parametric performance models can be adopted in order to provide a good estimate of the end-to-end system performance or to facilitate what-if analyses. This helps the Dev team to refactor the architecture in order to remove the performance bottlenecks [5]. As shown in Figure 2(i), the microservices monitoring solution we used consists of both of the client and server containers.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	Moreover, in the microservices setting, as each team would be responsible for their own services, they cannot benefit from the higher comprehensibility of the code and easier joining of new team members which is caused by the decomposition of the system. In contrast, DevOps recommends vertical dividing of project members into small cross-functional teams which also fits microservices well. Each team is responsible for a service and consists of people with different skills, like development and operations skills, and they cooperate from the beginning of the project to create more value for the end-users of that particular service through more frequent release of new features to the production, and hence, remove the transition overheads which were existed in the horizontal team formation.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	This can be done via introducing new migration patterns that were lacking before and were used in that particular project. This repository will serve as an extensible source for the DevOps community through which they can reuse patterns for migrating towards microservices, like the one for architectural patterns at http://microservices.io/.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	105.1 How the Microservices Migration Patterns Enable DevOps Traditional methods for software development advocate separated Dev and Ops team in which the Dev team provides the Ops team with deployment artifacts and details.	Ivon Miranda Santos
Balalaie2016-Microservices_architecture_enables_devops_Migratio	CLOUD MANAGEMENT > DevOps	These issues collectively delays the whole development life cycle. DevOps together with microservices are tackling the above mentioned issues via providing the necessary equipments for minimizing the coordination amongst the teams responsible for each component, and remov-ing the barriers for an effective and reciprocal relationship between the Dev and the Ops teams. Indeed, in the DevOps setting, these teams help each other through continuous valuable feedbacks. To elaborate further, in Table 1, we briefly describe the impact of our devised patterns on the DevOps and the problems they tackle. Appendix A Backtory Components Before the Migration	Ivon Miranda Santos

Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > DevOps	Indeed, microservices provide many benefits for Cloud-based applications, such as better scalability, autonomy and deployment times [6]. As a result, such architectures are a good fit for DevOps practices [7]. However, the transition from a monolithic to a microservice2 application is still considered as a long and risky process [9]. Furthermore, even though academic work has been done to provide guides and tools to help companies performing such a modernization, the refactoring of legacy systems remains overall a case-by-case process [10].	Ivon Miranda Santos
Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > DevOps	. Microservices are generally packaged and deployed in the cloud using lightweight container technologies such as Docker15, following the fashion of DevOps practices. These architectures are typically supported by fully automated software integration and delivery machinery such as Gitlab16 and Jenkins17 .	Ivon Miranda Santos
Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > DevOps	Such architectures are currently becoming a standard in soft-ware engineering. In [7], the authors emphasize the benefits that microservices architectures can bring to DevOps practices. For instance, microservices are known for their ability to deal with scalability [6], i.e. the ability to make the size of the system evolve as the project evolves. This ability is crucial in a DevOps-driven development, to quickly adapt a system to its usage. In addition to such technical benefits, microservices also allow for more process-related benefits, such as the distribution of the workforce into small and self-managed teams, each focused on a microservice. This fits very well DevOps practices which recommend to vertically divide the workforce into small cross functional teams [18].	Ivon Miranda Santos
Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > Performance	Indeed, while the transition to microservice can bring significant improvements to the application in the long term, and ease future developments, transition from legacy systems towards microservice architecture is a long and risky process [9]. Thus, companies often establish cost-benefit analysis, based on the potential benefits such as increased performances, better scalability and ease of the development process [6]. b) Modernization Planning:	Ivon Miranda Santos
Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > Performance	These structures have to be accessible by language services, more or less frequently depending on the services, as explained previously. Therefore, the transfer and the serialization of these structures have to be handled properly to avoid any drop in performances. If the entirety of the program and the associated resources were transferred at every call of a service such as auto-completion, this could have a great impact on the performances and make the overall environment feel sluggish. Thus, one has to make choices, both about the structures of the microservices and the data to transfer between them.	Ivon Miranda Santos
Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > Performance	Moreover, since no initialization phase is done with the AQL interpreter, a similar work has to be done at each evaluation. This can result in an important drop in performances, similar to the one observed in [2]. b) Stateful Approach:	Ivon Miranda Santos

Belafia2021-From_monolithic_to_microservice_architecture_The_ca	CLOUD MANAGEMENT > Performance	Based on the specification of the manipulated language and the desired language services, the approach generates a set of modular language microservices and a tool-supported feature model to configure their deployment. This approach brings disparity in the performances of language services after their modularization. If heavy services such as compilation show improvements on their response time from this approach, lightweight quick-feedback services such as auto-completion suffer greatly in reactivity. This drop in performances is even more accentuated when the length of the code increases. The authors of the paper have already identified two factors as responsible for this disparity: the statelessness of the services and the lack of optimization for the deployment.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Cost	The cloud assists to reduce time-to-market and provides on-demand scalability at a low cost for the users.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Performance	There is a need for integrating multiple heterogeneous clouds and to solve the problem of distributing services over several providers [4]. Thus, in a scenario where a complex application is distributed on different cloud service providers, a solution is needed in order to manage and orchestrate the distribution of modules in a sound and adaptive way. Such solution should determine the best cloud provider for each particular module based on client requirements (e.g., performance, cost, availability, scalability, etc.). Once the distribution has been decided, the solution should support operations such as managing the relationships between the different modules, maintaining all the specified properties and requirements, and monitoring and reconfiguring the distribution in case any problem occurs.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Cost	There is a need for integrating multiple heterogeneous clouds and to solve the problem of distributing services over several providers [4]. Thus, in a scenario where a complex application is distributed on different cloud service providers, a solution is needed in order to manage and orchestrate the distribution of modules in a sound and adaptive way. Such solution should determine the best cloud provider for each particular module based on client requirements (e.g., performance, cost, availability, scalability, etc.). Once the distribution has been decided, the solution should support operations such as managing the relationships between the different modules, maintaining all the specified properties and requirements, and monitoring and reconfiguring the distribution in case any problem occurs.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	O2) The monitoring and run-time reconfiguration operations of services dis-tributed over multiple heterogeneous cloud providers. Monitoring will be in charge of detecting the need of redistributing services on several cloud providers. As a consequence of monitoring, dynamic reconfiguration will be used to evolve the orchestration by considering all the changes required (without suspending the execution of services not affected by those changes). Reconfiguration may imply updating a service, dynamically replacing erro-neous services or migrating them to a different cloud provider to leverage its advantages or avoid the shortcomings of another cloud provider.	Ivon Miranda Santos

Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	O3) The offer of unified application management of services distributed over different cloud providers. SeaClouds will be able to deploy, manage, scale and monitor services over technologically diverse clouds providers. Such operations will be performed taking into account the synchronization requirements of the application as a whole, providing developers with support beyond the handling of single services.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	2.2 Monitoring of multi-cloud services The ongoing EU FP7 Cloud4SOA project (http://www.cloud4soa.eu) provides an open source interoperable framework for application developers and PaaS providers. Cloud4SOA facilitates developers in the deployment and lifecycle management and monitoring of their applications on the PaaS offering that best matches their computational needs, and ultimately reduces the risks of a vendor lock-in. The monitoring is based on unified metrics, but Cloud4SOA monitors each application separately and it is not able to aggregate monitoring results of multi-component applications.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	Several commercial and open source initiatives target monitoring of cloud applications. Often these initiatives address only particular platforms, for ex-ample Appsecute (http://www.appsecute.com) monitors only (open-source) CloudFoundry-based platforms. More platform-independent technologies are available for the IaaS level, since the latter has undergone a stronger harmo-nization effort. Deltacloud (http://deltacloud.apache.org/) encapsulates the native API cloud provider to enable management of resources in differents IaaS clouds, such as Amazon EC2. Rightscale (http://www.rightscale.com) sup-ports monitoring several public (e.g. Amazon Web Services, Rackspace) and private IaaS clouds (e.g. CloudStack, Eucalyptus, OpenStack). Truly platform-independent monitoring solutions exist, the most known being NewRelic (http: //www.newrelic.com). NewRelic achieves platform-independency by requiring each provider to implement a monitoring component and integrate it in the offered cloud platform. On the one hand, this approach yields the best results from a monitoring point of view. On the other hand, it forces providers to invest quite some resources in order to implement the monitoring.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	Challenges in monitoring of services on multiple clouds. In order to address O2, SeaClouds' monitoring will use and enhance existing monitoring functionalities for the PaaS and IaaS levels.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	For both the IaaS as the PaaS level, SeaClouds aims at coordinating, monitoring and aggregating monitoring information at the single service level to serve the purposes of orchestrated services. Thus, SeaClouds aims at: – Being able to monitor each of the application components, and – Combining and aggregating the above mentioned data to highlight perfor-mance problems and their impact.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Performance	– Being able to monitor each of the application components, and – Combining and aggregating the above mentioned data to highlight perfor-mance problems and their impact. 2.3 Unified management of multi-cloud applications	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	– SeaClouds management will use the REST harmonized API for the deploy-ment, management and monitoring of simple cloud-based applications across different and heterogeneous cloud PaaS offerings.	Ivon Miranda Santos

Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Portability	The OASIS TOSCA (Topology and Orchestration Specification for Cloud Applications) [6] Technical Committee aims at enhancing the portability of cloud applications and services. The main aim of TOSCA is to enable the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behaviour of these services, independently from the cloud provider. By increasing service and application portability in a vendor-neutral ecosystem, TOSCA aims at enabling portable deployment to any compliant cloud, smoother migration of existing applications to the cloud, as well as dynamic, multi-cloud provider applications.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	Challenges in standards for cloud interoperability. SeaClouds intends to actively contribute to the standardization effort of CAMP [12] both by imple-menting a CAMP-compliant interface towards PaaS providers for management, and by contributing review proposals that will possibly emerge while specifying properties of SeaClouds orchestrations, adaptation and monitoring.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	Figure 2 shows the cloud architecture situation currently before SeaClouds (top), and after SeaClouds (bottom). Without SeaClouds, services can only be deployed, managed and monitored on multiple clouds as standalone applications, and not as part of a composite application.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	SeaClouds aims at homogenizing the management over different providers and support the sound and scalable orchestration of services across them. Moreover, systems developed with SeaClouds will inherently support the evolution of their constituent services, so as to easily cope up with needed changes, even at runtime. The development, monitoring and reconfiguration via SeaClouds includes a unified management service, where services can be deployed, replicated, and administered by means of standard harmonized APIs such as CAMP specification [5] and Cloud4SOA (http://www.cloud4soa.eu/).	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Performance	4. Performance and cost optimization. The framework gives users freedom to distribute application requirements over different cloud offerings, using needed options in a flexible manner.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Performance	The Uses cases definition, implementation and validation represents the definition, development and validation of the SeaClouds case studies proto-types that will operate in the frame of preselected realistic use-cases and scenarios. The performance of the evaluation of the case studies and the generation of lessonslearnt and methodological adoption guidelines for cloud computing will be also considered here. In order to obtain the main goals of SeaClouds as regards the development of the platform and the definition, implementation and validation of the uses cases, SeaClouds provides a foundation to allow “Agility After Deployment” providing necessary tools and a framework for Modelling, Planning and Controlling	Ivon Miranda Santos

Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Monitoring	The Controller module will implement the multi-cloud deployment of the application modules and SeaClouds monitoring policy. In particular, the Multi-Cloud Deployer component will input the orchestration specification generated by the Planner, and it will deploy (by exploiting the Multi-Cloud Deployment API) the application modules on the specified clouds. The Monitor component of the Controller will be in charge of monitoring (by exploiting the Monitoring API) that the QoS properties of the application modules are not violated by the clouds in which they were deployed, and that the whole application satisfies the QoS properties specified for the whole application. If the monitoring component will detect a (non-transient) property violation, it will trigger the Analyzer component which is in charge of generating the reconfiguration suggestions (if needed) to be passed as inputs to the Planner module to trigger the generation of a new adaptive orchestration plan.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Cost	Reconfiguration mechanism. Our proposal allows a real flexible reconfiguration, with lighter services without extra migration logic at the cost of slightly higher time-to-service.	Ivon Miranda Santos
Brogi2014-Seaclouds_Seamless_adaptive_multi-cloud_management_of	CLOUD MANAGEMENT > Portability	Contribute to core concepts and usage patterns of Service Templates of TOSCA, as well as to the standardization effort of TOSCA, by providing feedback that will emerge while trying to devise a TOSCA-compliant instance of the SeaClouds service orchestration model. In TOSCA, issues like portability and service composition might benefit from some of the adaptation techniques in SeaClouds, driving the design of the model for specifying cloud service orchestrations. Also the composition of service templates could be improved with a dynamic reconfiguration mechanism, providing more flexible conditions to substitute service templates.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	O3) Offer unified application management of services distributed over different cloud providers. SeaClouds will be able to deploy, manage, scale and monitor services over technologically diverse clouds providers. Such operations will be performed by taking into account the synchronization requirements of the application as a whole and by providing developers with support beyond the handling of single services.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	O4) Compliance with major standards for cloud interoperability. SeaClouds will manage applications deployed on technologically diverse cloud platforms, unifying operations such as monitoring and lifecycle man-agement, promoting the adoption of OASIS standards for cloud interoperability, in particular TOSCA [7] and CAMP [4].	Ivon Miranda Santos

Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	In order to motivate our proposal, we introduce an example where a multi-component application is going to be deployed on (potentially) different cloud providers, according to a number of requirements on each of the modules composing the application. After the (multi-cloud) deployment is performed, and components are being executed in different cloud platforms, a monitoring process is in charge of detecting possible requirements violations, which could eventually trigger a reconfiguration. Figure 1 shows the architecture of an online retailing application which is offered to clients as a shopping Webpage. It consists of four modules: A main web page to access the system, two databases (one for users and one for products), and a payment module. The functionalities of the four components of the online system are:	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Considering the changing situations listed above, the application manager would need to have some modules of the application being moved from the platforms in which they were originally deployed to other platforms. To this end, the SeaClouds platform will provide monitoring facilities allowing the user to observe a set of standardised and unified metrics of different types, based on underlying cloud providers monitoring systems. This functionality will permit the runtime monitoring of the deployed modules so that the required QoS level can be assured.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Challenges in orchestration and adaptation for the cloud SeaClouds will address the following challenges in order to extend service-oriented approaches to the cloud: • Adaptation contracts need to take into account cloud providers characteristics and Service Level Agreement (SLA). • Violations of Quality of Service (QoS) properties need to be monitored across different cloud platforms. • Dynamic architecture reconfiguration might involve migrating some components of the application to other cloud providers at runtime. The latter two challenges (addressed by O2 and O3) are discussed further in the following sections.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	3.2 Monitoring of multi-cloud services The EU FP7 Cloud4SOA project (http://www.cloud4soa.eu) provides an open source interoperable frame-work for application developers and PaaS providers. Cloud4SOA facilitates developers in the deployment and lifecycle management and monitoring of their applications on the PaaS offering that best matches their computational needs, and ultimately reduces the risks of a vendor lock-in. The monitoring is based on unified metrics, but Cloud4SOA monitors each application separately and it is not able to aggregate monitoring results of multi-component applications.	Ivon Miranda Santos

Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Several commercial and open source initiatives target the monitoring of cloud applications. Often these initiatives address only particular platforms, for example Appsecute (http://www.appsecute.com) monitors only (open-source) CloudFoundry-based platforms. More platform-independent technologies are available for the IaaS level, since the latter has undergone a stronger harmonization effort. Deltacloud (http://deltacloud.apache.org/) encapsulates the native API cloud provider to enable management of resources in differents IaaS clouds, such as Amazon EC2. Rightscale (http://www.rightscale.com) supports monitoring several public (e.g. Amazon Web Services, Rackspace) and private IaaS clouds (e.g. CloudStack, Eucalyptus, OpenStack). Truly platform-independent monitoring solutions exist, the most known being probably NewRelic (http://www.newrelic.com). NewRelic achieves platform-independency by requiring each provider to implement a monitoring component and to integrate it in the offered cloud platform. On the one hand, this approach yields the best results from a monitoring point of view. On the other hand, it forces providers to invest quite some resources in order to implement the monitoring.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Challenges in monitoring of services on multiple clouds. In order to address O2, SeaClouds' monitoring will use and enhance existing monitoring functionalities for the IaaS and PaaS levels. • With respect to the IaaS level, SeaClouds will simply reuse what is available (e.g., Deltacloud). • With respect to the PaaS level, SeaClouds aims at augmenting the set of metrics currently available from Cloud4SOA (response time and up-time).	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	For both the IaaS and the PaaS level, SeaClouds aims at coordinating, monitoring and aggregating monitoring information at the single service level to serve the purposes of orchestrated services. Thus, SeaClouds aims at: • Being able to monitor each of application component, and at • Combining and aggregating the above mentioned data to highlight performance problems and their impact.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Performance	• Being able to monitor each of application component, and at • Combining and aggregating the above mentioned data to highlight performance problems and their impact. 3.3 Unified management of multi-cloud applications	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Challenges in unified application management of services distributed over different cloud providers SeaClouds will use Cloud4SOA's management functionality. Specifically: • SeaClouds' discovery functionality may use and extend existing matchmaking functionalities to match application requirements with PaaS offerings. • SeaClouds management will use the REST harmonized API for the deployment, management and monitoring of simple cloud-based applications across different and heterogeneous cloud PaaS offerings.	Ivon Miranda Santos

Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	CAMP (Cloud Application Management for Platforms) [4] aims at defining a harmonized API, models, mech-anisms and protocols for the self-service management (provisioning, monitoring and control) of applications in a PaaS, independently of the cloud provider. However, CAMP is only a protocol specification, so it needs to be implemented by parties adopting the protocol.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Portability	The OASIS TOSCA (Topology and Orchestration Specification for Cloud Applications) [7] Technical Committee aims at enhancing the portability of cloud applications and services. The main aim of TOSCA is to enable the interoperable description of application and infrastructure cloud services, the relationships between parts of the service, and the operational behaviour of these services, independently from the cloud provider [17]. By increasing service and application portability in a vendor-neutral ecosystem, TOSCA aims at enabling portable deployment to any compliant cloud, smoother migration of existing applications to the cloud, as well as dynamic, multi-cloud provider applications.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Challenges in standards for cloud interoperability. SeaClouds intends to actively contribute to the standardization effort of CAMP [18] both by implementing a CAMP-compliant interface towards PaaS providers for management, and by contributing review proposals that will possibly emerge while specifying properties of SeaClouds orchestrations, adaptation and monitoring.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	The MODAClouds project (http://www.modaclouds.eu/) also aims at providing quality assurance during the application life-cycle, supporting migration from cloud to cloud when needed, and techniques for data mapping and synchronization among multiple clouds. To do so, MODAClouds requires software developers to adopt a Model-Driven Development approach. The monitoring platform developed in MODAClouds overcomes the limitation of the one offered by Cloud4SOA by gathering data of various kinds from components, containers and cloud resources distributed and replicated on multiple clouds. Although this approach has differently from SeaClouds, an impact on the code that needs to be deployed on the cloud, we are considering incorporating some of the MODAClouds functionalities related to the usage of data collectors which we are currently analyzing.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Portability	However, it is centred on IaaS portability and does not address the lock-in at level of PaaS. On the other hand, at the level of PaaS, Cloud Foundry Core project is growing fast. The Cloud Foundry Core defines a baseline of common capabilities to promote Cloud portability across different instances of Cloud Foundry. This introduces a new level of lock-in, where one can migrate over platforms but it is required to implement a set of capabilities defined by the Cloud Foundry Core. The SeaClouds project goes one step further, and expects to allow the migration over all platforms, so as to strongly mitigate the vendor lock-in problem.	Ivon Miranda Santos

Brogi2015-Adaptive_management_of_applicatio ns_across_multiple_c	CLOUD MANAGEMENT > Monitoring	4.1 Overall strategy Figure 4 shows the cloud architecture situation currently before SeaClouds (top), and after SeaClouds (bottom). Without SeaClouds, services can only be deployed, managed and monitored across multiple clouds as standalone applications, and not as part of a composite application. This has the consequence that there is no support for synchronized deployment and unified monitoring, which implies that QoS of the entire application is difficult to monitor. There is also no support for migrating one service and reconfiguring the rest of the application to use the migrated service, in case a provider does not respect its SLA.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applicatio ns_across_multiple_c	CLOUD MANAGEMENT > Monitoring	SeaClouds aims at homogenizing the management over different providers and at supporting the sound and scalable orchestration of services across them. Moreover, systems developed with SeaClouds will inherently support the evolution of their constituent services, so as to easily cope up with needed changes, even at runtime. The development, monitoring and reconfiguration via SeaClouds includes a unified management service, where services can be deployed, replicated, and administered by means of standard harmonized APIs such as CAMP specification and Cloud4SOA.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applicatio ns_across_multiple_c	CLOUD MANAGEMENT > Performance	4. Performance and cost optimization. The framework gives users freedom to distribute application requirements over different cloud offerings by using needed options in a flexible manner.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applicatio ns_across_multiple_c	CLOUD MANAGEMENT > Monitoring	On the other hand, SeaClouds also employs CAMP, which proposes standardised artifacts and APIs that need to be offered by PaaS clouds to manage the building, running, administration, monitoring and patching of applications in the cloud. It is however worth noting that the Deployer does not require cloud providers to be TOSCA or CAMP compliant, and it actually generates concrete deployment plans for non TOSCA/CAMP compliant providers as needed.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applicatio ns_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Monitor. The Monitor component gets the set of SLAs of the services in the selected deployment plan, and is in charge of collecting monitoring information from the targeted cloud platforms, of analysing such information, and of presenting the results of such analysis (through the SeaClouds dashboard) to the Deployment Manager. The Monitor is also in charge of generating replanning triggers that are passed (possibly filtered by the Deployment Manager, depending on the platform configuration) to the Planner in order to start a reconfiguration process.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applicatio ns_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Multi-cloud orientation. As described previously, cloud computing has proven a major commercial success in the last years, with the appearance of many different vendors. What followed is a need for integrating multiple heterogeneous clouds and to solve the problem of distributing the services over several providers. In particular, the need of orchestration is more evident when complex applications move to cloud environments. With the current cloud technologies, services can only be deployed, managed and monitored on multiple clouds as stand-alone applications, and not as part of a composite application.	Ivon Miranda Santos

Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Portability	In line with the main goals of TOSCA [17], the use of this standard will ease automated deployment and management, and will enhance the portability and reusability of multi-cloud applications and their components. In addition, it will also allow the SeaClouds platform to generate TOSCA-compliant orchestration specifications, which will ease the matching and interoperation with TOSCA-compliant PaaS offerings.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Compatibility with CAMP. The SeaClouds platform is compatible with the novel OASIS standard CAMP, which is one of the major standards for cloud interoperability. Its objective is to define standardised artefacts and APIs that a PaaS should offer to allow the management, building, administration, monitoring and patching of cloud-based applications. Obviously, the availability of CAMP results can simplify the development of the Discovery API, Monitoring API, and part of the Multi-Cloud Deployment API of SeaClouds	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Performance	Furthermore, by leveraging CAMP, SeaClouds will attract a significant user base (as this standard has a lot of interest but no reference implementations, so far) and advance the standard, ensuring the long-term viability of the benefits implied in SeaClouds, i.e., management and monitoring of underlying providers, performance optimisation, low impact on the code, formal methods support, flexibility to include new services and react to problems at runtime. On the other hand, SeaClouds can provide valuable feedback and contribute to the standardisation effort of CAMP, both by implementing a CAMP-compliant interface towards PaaS providers for management, and by contributing review proposals that will possibly emerge while specifying properties of the included orchestrations, adaptation and monitoring.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	Compatibility with different target platforms. In addition to the above-mentioned cloud standards as TOSCA and CAMP, SeaClouds also uses several existing platforms and initiatives, such as Brooklyn, Whirr, jClouds, Cloud4SOA, and MODAClouds. The Cloud4SOA project provides an open source interoperable framework for application developers and PaaS providers. It facilitates developers in the deployment and lifecycle management of their applications on the PaaS offering that best matches their computational needs, and ultimately reduces the risks of a vendor lock-in. SeaClouds will leverage and extend Cloud4SOA outcomes, such as multiplatform matchmaking, management, cloud monitoring and migration, to ease and accelerate the implementation.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	If requirements are violated, then the SeaClouds monitor will generate reconfiguration information which leverages the creation of a different orchestration of the application. The proposed architecture can well support this process, and also the exploitation of the best available offering for each application component at any time.	Ivon Miranda Santos
Brogi2015-Adaptive_management_of_applications_across_multiple_c	CLOUD MANAGEMENT > Monitoring	A key ingredient in our proposal is the use of two OASIS standards initiatives for cloud interoperability, namely CAMP and TOSCA, which allow us to describe the topology of user applications independently of cloud providers, provide abstract plans, and discover, deploy/reconfigure, and monitor our applications independently of the particularities of the cloud providers.	Ivon Miranda Santos

Caceres2022-State-of-the-art_architectures_for_interoperability	CLOUD MANAGEMENT > Performance	One aspect of this is scheduling apps at runtime to communicate with micro clouds along with data centers. This includes data partitioning over low power as well as high end processors to achieve the best performance specific to user objectives. In a decentralized way of cloud computing, an application needs to be offloaded into micro clouds both from user devices and data centers.	Ivon Miranda Santos
Caceres2022-State-of-the-art_architectures_for_interoperability	CLOUD MANAGEMENT > Cost	This idea also has downsides, such as still comparably high amount of manual work, lack of out-of-the-box platform-specific authentication and authorization mechanisms, and complicated cloud resource costs calculation.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Cost	In a recent survey released by Rackspace[1], 88% of companies using cloud computing said to have reduced costs by moving services to the cloud.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Performance	Special attention also needs to be paid to the data structures used to design the application and to the architecture it uses internally. Both of them are bound to be extremely dependent on the platform and may have a profound impact on performance if done naively. For example, different Platform As A Service (PaaS) providers such as Google[30] and Heroku[31] impose different design constraints to deployed applications, applications written to perform well on one of them may have a hard time running on the other one.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Performance	On the other hand, the runtime component also provides feedback to the design time IDE. Monitoring information is for example fed back to the IDE to help the developer in improving the design of the application, and consequently its performance. D. Our vision:	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Performance	Clients should be able to configure their subscribed services by means of an administration environment. Clients should be able to supervise the availability and performance of their subscribed services and to adapt the deployment of services to their needs by means of an administration environment. By adapting we mean moving projects from one node to another or even from one cloud provider to another.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Monitoring	The only dependency of this design to the specific cloud provider is the communication between the Administration Service and the cloud provider in order to deploy, monitor and eventually migrate services. The actual code to interact with the cloud provider is however encapsulated in a Web Service usually installed on the Administration Service. This Web Service translates actual requests from the user into specific requests to the cloud provider.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Performance	What is the cost of migrating a given service from one server to another? Which data structures will provide best performance for each service? These questions do not necessarily represent functionalities we intend to implement in our offering, but challenges we, as designers, need to overcome.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Monitoring	First of all, the Administration Service needs to be extended with the support monitoring QoS on multiple clouds. We intend to work on the monitoring resource status information such as the available and used memory and disk space, and the CPU consumption. We consider such pieces of information to be important for allowing our customers to decide upon when migrations need to be performed.	Ivon Miranda Santos

daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Performance	Finally, the design of data structures and deployment architecture is also a very important part of the design of a cloud application. Currently, this activity is only feasible by manually rewriting code using different libraries and deploying the application to test its performance. We expect the MODAClouds project to develop the necessary tools and techniques to support us in designing and testing different data designs.	Ivon Miranda Santos
daSilva2013-From_the_desktop_to_the_multi-clouds_The_case_of_mo	CLOUD MANAGEMENT > Monitoring	Throughout the MODAClouds project, SOFTEAM intends to implement and validate the architecture presented in the present paper. The MODAClouds technologies will hopefully help to provision and monitor all cloud-based services, as well as, if necessary, to migrate them from one cloud provider to another.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	Dealing with ven-dor lock-in in multiple clouds requires addressing two important challenges: interoperability and portability. Some solutions have been proposed to deal with both problems, but most of them fail to provide flexibility. Therefore, we propose PacificClouds, a novel architecture based on microservices for addressing interopera-bility in a multi-cloud environment. PacificClouds differs from previous works by providing greater flexibility due to the microservices architectural pattern.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	One method for treating vendor lock-in is the use of multiple clouds, although a small number of enterprises adopt this approach, their popularity is increasing. One reason for the low adoption of multiple clouds is cloud providers interest lack to promote interoperability and portability (Grozev and Buyya, 2014). According to this context, (Opara-Martins et al., 2015) observes the need for dealing with in-teroperability and portability in order to mitigate the problem of vendor lock-in. Section 2 describes mul-tiple clouds, interoperability and portability.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	For that matter, the applications are built with the focus on integration with the cloud model, to obtain full cloud advantages; it, also ensures other fe-atures labeled as IDEAL (Isolated state, Distribution, Elasticity, Automated management, Loose coupling). In this manner, the native cloud application can faci-litate the application deployment in multiple clouds, hence help treat interoperability and portability (Feh-ling et al., 2014).	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Cost	Microservices can aid in obtaining the native cloud application’s characteristics; therefore, they fo-cus on aspects as componentization of small and lig-htweight services, agile and DevOps practices, in-frastructure automation with continuous delivery fe-atures, decentralized data management, and decen-tralized governance among services. The microser-vices promise more agility, more delivery speed, and more scalability compared with traditional monolithic applications, resulting in less overall cost (Newman, 2015), (RV, 2016). In Section 3, we describe, present challenges and propose a definition for microservices.	Ivon Miranda Santos

deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > DevOps	Microservices can aid in obtaining the native cloud application's characteristics; therefore, they fo-cus on aspects as componentization of small and lig-htweight services, agile and DevOps practices, in-frastructure automation with continuous delivery fe-atures, decentralized data management, and decen-tralized governance among services. The microser-vices promise more agility, more delivery speed, and more scalability compared with traditional monolithic applications, resulting in less overall cost (Newman, 2015), (RV, 2016). In Section 3, we describe, present challenges and propose a definition for microservices.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	However, the multiple clouds bring several challenges, as well, e.g., interoperability and portabi-lity related to mitigating vendor lock-in. We consider portability the ability to allow customers to migrate data and systems from one cloud to another and inter-operability capacity to allow customers to use servi-ces across multiple clouds (Rezaei et al., 2014).	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	According to the PacificClouds goals described in section 1, we adopt the multi-cloud delivery model, for it brings greater flexibility. In relation to porta-bility, we assume the three categories used by (Pe-tcu et al., 2013) and we report the IaaS and PaaS le-vels for the portability requirements. According to interoperability levels, we have considered the follo-wing criteria, (Nogueira et al., 2016): we have adop-ted both syntactic and semantic level interoperability, associated with the agreement level.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Performance	In order for architecture to cover all objectives of PacificClouds, aforementioned in this article, it must include the following functionalities: (i) identify the application microservices as well as its requirements; (ii) identify user requirements; (iii) discover and mo-nitor the capabilities of the available clouds; (iv) ma-nage the application deployment and execution; (v) monitor the deployment and execution of each appli-cation microservice; (vi) monitor user and application requirements, as well as the capabilities of the clouds involved in running the application; (vii) provide the deployment of each application microservice; (viii) manage the cloud resources of each application mi-croservice; (ix) provide application performance in-formation to the user, as that the user can modify the requirements at runtime. Therefore, PacificClouds ar-chitecture consists of three parts:	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Monitoring	In order for architecture to cover all objectives of PacificClouds, aforementioned in this article, it must include the following functionalities: (i) identify the application microservices as well as its requirements; (ii) identify user requirements; (iii) discover and mo-nitor the capabilities of the available clouds; (iv) ma-nage the application deployment and execution; (v) monitor the deployment and execution of each appli-cation microservice; (vi) monitor user and application requirements, as well as the capabilities of the clouds involved in running the application; (vii) provide the deployment of each application microservice; (viii) manage the cloud resources of each application mi-croservice; (ix) provide application performance in-formation to the user, as that the user can modify the requirements at runtime.	Ivon Miranda Santos

deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Monitoring	PacificClouds Core is the central part of the architecture. It is responsible for discovering the application microservices and their respective require-ments; determining the clouds; deploying the micro-services; managing cloud resources used by the ap-plication; monitoring information about the resources used by the application at runtime; monitoring all ca-pabilities of the clouds; and verifying the user requi-rements.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Monitoring	4.2.4 Monitoring Service Monitoring Service, shown in Figure 5, is responsible for monitoring the cloud resources used by the application at runtime through the adapter. Monitoring Service is composed by three services: (i) Cloud Resources Register Service (CRRS) receives from DPGS the deployment plan data. After, it re-gisters it in MsRsData and it sends it to CRDAS; (ii) Cloud Resources Detection Service (CRDetS) detects all capabilities of the clouds, which are deployed by the microservices.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Performance	After, it re-gisters it in MsRsData and it sends it to CRDAS; (ii) Cloud Resources Detection Service (CRDetS) detects all capabilities of the clouds, which are deployed by the microservices. Next, it sends them to CRDAS; (iii) Cloud Resource Data Analysis Service(CRDAS) is responsible for comparing the application perfor-mance received from CRDS with the deployment plan data; next, it sends the cloud resources data used by the application to SLA Service and sends a notification to Notification Service if any violation happens. Figure 5:	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Performance	Next, De-ployment Service identifies the cloud for each micro-service according to deploy plan via Microservice As-sociation Service (MAS); after, it deploys the micro-services via Microservice Deployment Service (MD-plS). We can observe that PacificClouds intends to manage the deployment and execution process of the dis-tributed application in multiple clouds to better meet the needs of the application and consequently im-prove end-user satisfaction with the application per-formance. Thus, the PacificClouds will be able to help the software architecture in the process of de-CLOSER 2018 - 8th International Conference on Cloud Computing and Services Science	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Monitoring	The PacificClouds must also monitor application execution by observing the clouds involved in the application deployment to de-tect any violation of requirements, and monitor chan-ges in application requirements, application services’ updates and capabilities of the clouds to identify the need to re-deploy some application services.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	In Table 1, there is the only aspect that all solutions promote the semantic interoperability. Some so-lutions use the semantic interoperability in the appli-cations portability, while others use it in the interope-rability between application modules distributed over multiple clouds, and it includes solutions that use it to treat vertical interoperability between service models.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	Finally, mOSAIC, SeaClouds, and PacificClouds treat interoperability in the IaaS and PaaS service mo-dels. mOSAIC uses the hybrid delivery model; it de-als with vertical interoperability as it makes applica-tion portability. SeaClouds and PacificClouds address vertical and horizontal interoperability levels for the multi-cloud delivery model as well as promote appli-cation portability.	Ivon Miranda Santos

deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	RASIC, ASMEMA, SeaClouds, and Paci-ficClouds address interoperability for distributed applications in multiple clouds geographically dispersed, while the other solutions address only ap-plication portability between clouds. In relation to the background technologies in clouds, just Cloud4SOA and SeaClouds do not use clouds that have different background technologies, that is, all clouds involved in the portability or interoperability of applications must have the same background technology for Cloud4SOA and SeaClouds.	Ivon Miranda Santos
deCarvalho2018-Pacificclouds_A_flexible_microservices_based_arc	CLOUD MANAGEMENT > Portability	The difference bet-ween ASMEMA and PacificClouds is that the former does not do decentralized governance by not treating either interoperability or portability	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	In our pre-vious work, we performed a systematic review to identify the approaches adopted by organizations to migrate to cloud computing and their per-ception of the cost-benefit of this migration.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	The results in this systematic literature review can help the development of guidelines to support newcomers companies to adopt and migrate to the cloud, how the cost-benefit relationship can be evaluated as well as the selection of providers.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	Cloud computing · Cloud migration · Provider selection · Cost-benefit relationship · Systematic literature review	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	However, the identification of opportunities for migration, the reasoning of an attractive cost-benefit relation-ship and the selection of service providers that best fit their needs are not trivial tasks [16,17].	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	The reason for a Systematic Literature Review (SLR) is the necessity to iden-tify, classify, and compare existing evidence on the strategies used by companies to identify scenarios of migration opportunities to the CC. To justify the adop-tion, a set of factors should be considered for the assessment of the cost-benefit relationship.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	(i) the identification of strategies and issues that companies have considered to migrate to the cloud; (ii) factors that should be considered in the cost-benefits relationship while adopting and migrating to the cloud; (iii) and finally aspects related to the selection of cloud computing service providers.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	(i) Public cloud is available from a third party service provider via web and is a very cost effective option to deploy IT solutions [20]; (ii) Private cloud is managed within an organization and is suitable for large enterprises (managed within the walls of the enterprises).	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	(i) strategies to identify migration opportunities to the cloud, (ii) relevant fac-tors for the assessment of the cost-benefit of this adoption of cloud and finally (iii) the selection of providers according to their needs and profile.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	To this end, it identifies a common set of services offered by cloud providers, including elastic computing, persistent storage, and intra-cloud and wide-area networking. The authors argue that CloudCmp enables predicting application performance without having to first port the application230 A.C.M. de Paula and G.d.F. de Carneiro	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	[S2] describes a cloud Adoption Toolkit that uses Cost Modeling techniques to examine cost of deploying IT system to the cloud.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	This was performed with a proposed compatibility checklist that is used to estimate the cost of application migration to PaaS.	Ivon Miranda Santos

dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	In [S41] two types of CC adoptions were suggested, manufacturing with direct adoption of CC technologies and cloud manufacturing. [S44] used small case study to show that application performance doesn’t deteriorate when migrating applica-tions to the cloud. [S45] abstracts from current market prices and investigates the interaction of cloud provider and clients from an analytical perspective.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	During the analysis of RQ2, we identified a myriad of factors related to the cost-benefit relationship of cloud computing adoption.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	In [S3], the authors argue that CC has been viewed mainly from the cost perspective. The paper proposed a model that helps not just identify the suitability of a company for the cloud by clearly spelling out all the factors that need to be considered for the same, but also gives a certain profitability valuation of the benefits associated with CC. An approach to detect performance anti-patterns before migrating to CC based on static analysis was presented in [S12]. In [S4], the architectural features of CC are explored and classified according to the requirements of end-users, enterprises, and cloud providers themselves to support the cloud adoption.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	The [S2] study described the Cloud Adoption Toolkit that provides a framework and a cost modelling tool to support decision makers.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	In [S16], the authors presented a compatibility checklist that is used to estimate the cost of application migration to PaaS. The migration of legacy applications to CC was discussed in [S17], whose focus was on the application performance analysis and providers characteristics. The authors of [S22] discussed the migration of agile projects to the cloud in terms of cost, time and quality.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	In [S16], the authors presented a compatibility checklist that is used to estimate the cost of application migration to PaaS.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	The authors of [S22] discussed the migration of agile projects to the cloud in terms of cost, time and quality. [S23] discussed potential issues and challenges that organizations may face while considering to migrate workloads to the cloud: efficiency, agility, quality, security, governance and standardization in the delivery, consumption and operation of IT services, all at reduced capital and operational expense.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	The paper [S33] investigated the migration costs of several deployment options using benchmarks and concluded that application characteristics such as workload intensity, growth rate, storage capacity, and S/W licensing costs produce complex combined effects on overall costs.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Portability	S36] proposed the use of a real option model to help companies think and decide when to switch to cloud based on the expected benefits, uncertainties and the value a company puts on money. [S37] investigated different approaches to reduce both cost and task completion time of computations using Amazon EC2’s spot instances for resource provisioning. In the case of [S38], the authors focused on the following factors: availability, portability, integration, migration236 A.C.M. de Paula and G.d.F. de Carneiro complexity, data privacy and security	Ivon Miranda Santos

dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	In [S44], the authors discussed the cloud model in five perspectives: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service. A case study was used to demonstrate that application performance does not deteriorate when migrating applications to the cloud. [S51] proposed a tripod model of SaaS readiness that suggests that for organizational users to adopt SaaS, they need to get ready from technolog-ical, organizational and environmental aspects. [S52] presented a taxonomy to help profile and standardize the details of performance evaluation of commercial Cloud services. In [S53], the authors proposed a set of de facto metrics adopted in the existing Cloud services evaluation work to collect and arrange different Cloud service features to be evaluated.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	The Cost Modeling tool (Cloud Adoption Toolkit) was also classified in the	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	In addition, it also provided a matching and migration cost calcula-tion.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	F. de Carneiro important elements that characterise Cloud computing infrastructures: service type, resource deployment, hardware, runtime tuning, security, business model, middleware, and performance [S57] proposed a framework and a mechanism called SMICloud to measure specific quality attributes and prioritize Cloud ser-vices. The goal was to compares different CSPs and measure QoS attributes defined by Cloud Service Measurement Index Consortium (CSMIC). In [S67], the authors identified five main performance criteria considered relevant to measure QoS for cloud users: Availability, Reliability, Performance, Cost and Security. Under each main criteria, subcritera, which are directly measurable from cloud provider premises, were defined.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Performance	In the following, we presented studies that also presented tools. [S7] presented a simulation tool called CDOSim whose goal is to simulate cost and performance attributes in CDOs. The tool is build upon and significantly extends the cloud simulator CloudSim and integrates into the cloud migration framework Cloud-MIG. [S55] the tool CloudCmp to systematically compare the performance and cost of cloud providers along dimensions that matter to customers. This sys-tematic review provided evidences of strategies used by companies to identify opportunities to migrate and adopt cloud computing, how they assess the cost-benefit relationship and strategies behind the rationale to select providers.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	[S55] the tool CloudCmp to systematically compare the performance and cost of cloud providers along dimensions that matter to customers.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	This sys-tematic review provided evidences of strategies used by companies to identify opportunities to migrate and adopt cloud computing, how they assess the cost-benefit relationship and strategies behind the rationale to select providers.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Portability	A spectrum of techniques and approaches has been identified that cope with vari-ous concerns, i.e., security and trustworthiness, elasticity, portability and inter-operability, and cloud resilience. In addition, many studies look into reference architectures and cloud-based architecture design methods as well.	Ivon Miranda Santos
dePaula2016-A_systematic_literature_review_on_cloud_computing_a	CLOUD MANAGEMENT > Cost	In this scenario, we also focused in the identification of evidences related to the cost-benefit relationship of this migration and selection of cloud service providers.	Ivon Miranda Santos

Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	Unfortunately, such SaaS multi-cloud deployment approach faces many technical obstacles such as clouds heterogeneity and ensur-ing data consistency across different clouds. Cloud heterogeneity could be easily resolved using service adapters, but ensuring data consistency remains a major obstacle, as existing approaches offer a trade-off between correctness and performance. Hence, SaaS providers opt to choose one or more of these approaches at design time, then create their services based on the limitations of the chosen approaches.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Cost	locked-in due to clouds heterogeneity and data migration costs.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	One approach for preventing such risk is to deploy business services on different clouds. SaaS multi-cloud deployment provides better performance and lower costs compared to the usage of a single cloud, as it provides better availability, responsiveness, and resources utilization [1]. However, SaaS multi-cloud deployment approach faces many technical challenges such as cloud heterogeneity, ensuring data correctness, and security.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Cost	In addition to the data migration costs.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	The Cloud Orchestration Approaches: Efforts in this approach (such as the works in [1] [5] [6]) are dedicated to create an orchestration layer between SaaS providers and cloud-vendors that can provide many services for SaaS providers such as portability, migration, and mon-itoring services. However, such approach is still not mature enough, furthermore it uses a centralized global control over WAN connections, which leads to bad per-formance as indicated in [7] [8], as global locks are used to ensure data correctness.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Portability	The Cloud Orchestration Approaches: Efforts in this approach (such as the works in [1] [5] [6]) are dedicated to create an orchestration layer between SaaS providers and cloud-vendors that can provide many services for SaaS providers such as portability, migration, and mon-itoring services. However, such approach is still not mature enough, furthermore it uses a centralized global control over WAN connections, which leads to bad per-formance as indicated in [7] [8], as global locks are used to ensure data correctness.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Monitoring	The Cloud Orchestration Approaches: Efforts in this approach (such as the works in [1] [5] [6]) are dedicated to create an orchestration layer between SaaS providers and cloud-vendors that can provide many services for SaaS providers such as portability, migration, and mon-itoring services. However, such approach is still not mature enough, furthermore it uses a centralized global control over WAN connections, which leads to bad per-formance as indicated in [7] [8], as global locks are used to ensure data correctness.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	For example, the Spanner system [12] uses 2PC and 2PL to provide atomicity and isolation, running on top of a Paxos-replicated log, which provides a fault-tolerant synchronous replication process across datacenters. Such pessimistic approaches suffer from bad performance when the executed workload creates hot spots, as global locks are established over the slow WAN connections. SaaS services adopting these approaches guarantees their data correctness, as all the replicas are eagerly synchronized, however they may suffer from bad performance due to hot spots. • The Escrow-Based Serializable Approach:	Ivon Miranda Santos

Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	This is a novel approach (i.e., NASEEB [18]) that guarantees data correctness as in the pessimistic approaches, and still en-sures good performance as in the optimistic approaches. It uses the the notion of a datacenter escrow, in which every datacenter will have a specific non-overlapping capacity quota (i.e.,	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	25 for each if we distribute equally). Hence, datacenters could process their incoming requests locally without the need to have global locks or consensus protocols, which drastically improves performance. Therefore, data correctness is guaranteed without the need for global locks, as any transaction com-mitted locally is guaranteed to commit globally.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	Then design the SaaS service according to the capabilities of the chosen PaaS services. For example, the spanner system enables business service providers to choose from two options: a pessimistic option with bad performance during contention periods, or an optimistic option with a possibility of data inconsistency. We argue that such approach for realizing SaaS multi-cloud deployment is limiting and inflexible for the following reasons:	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	manner to ensure data correctness. Experimental results show that SULTAN handles consistency requirements' changes in a realistic practical timing, also improves services performance when compared with the existing pessimistic serializable data concurrency approaches. The rest of the paper is organized as follows.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	RollBack option implies undoing all con-flicting transactions and performs the corresponding com-pensating transactions if any. This option is the highest in cost and could degrade the performance. Other options such as LastValue, MaxValue, MinValue, AvgValue, SumValue, MajorityValue, apply basic probabilistic ba-sic functions over the conflicting data to create an approximate new common value.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	, WAN connections) is within the range (83ms-445ms). Hence, performance mainly affected by the inter-datacenter latency. We summarize the experiments setup steps as follows:	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	, 5000, where 1000 comes from every datacenter). We argued that strong-eager consistency should be used only for the objects crucial for the service correctness, Figure 3 confirms our argument, as when having only 10% of the objects as strong-eager, it shows big performance improvement (i.e., about 500%, jumping from 600 to 3700) when compared with with the 100% of the objects are strong-eager. This means adopting a composite consistency approach improves SaaS service performance. Also when comparing this scenario to the scenario of 100% of the objects are eventual, we can see limited performance loss (i.e., 26%, dropping from 5000 to 3700), which we believe is a very acceptable price for ensuring SaaS services correctness. From results, we can see that the strong-escrow consistency is the best option for SaaS services, if they can tolerate data unfreshness, as it provides the best performance with data correctness assurance. Otherwise, a composite data consistency approach should be adopted in SaaS multi-cloud deployments to ensure the correctness of the crucial data.	Ivon Miranda Santos
Elgedawy2015-Sultan_A_composite_data_consistency_approach_for_s	CLOUD MANAGEMENT > Performance	Experimental results show that SULTAN handles consistency requirements' changes in a realistic practical timing, and improves services performance when compared with the existing pessimistic serializable data concurrency approaches. REFERENCES	Ivon Miranda Santos

Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > multi-tenant	Recent advantages in virtualization combined with multi-tenancy enables cloud infrastructure providers to perform large-scale provisioning of compute or data intensive services.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Cost	, flexibility, robustness, cost-efficiency, and sustainability of cloud infrastruc-tures [2,27,28,29].	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Performance	Prominent features of the envisioned infrastructure include seamless integration of local resources and capacity leased from external infrastructure providers, as well as the ability to migrate virtual infrastructures, in parts or as a whole, e.g., for continued provisioning over planned system down-time, fault tolerance, or performance improvements by moving services closer to end-users or service components closer to each other. W. Abramowicz et al.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Performance	2 Predictive Elasticity Control A key feature of cloud infrastructures is elasticity which is the ability of the cloud to automatically and rapidly scale up or down the resources allocated to a service according to the current demand on the service while enforcing the per-formance or capacity based Service Level Agreements (SLAs) specified. It should be possible to scale resources either by changing the number of VMs (horizontal elasticity) or by changing the size of the VMs (vertical elasticity) depending on the application's storage, memory, network bandwidth, and computate power requirements.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Cost	Since changes in placement requires migration of already running VMs, migration costs need to be part of the equa-tion, including overhead due to migration downtime, infrastructure capacity loss and monetary loss, etc.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Cost	size), current placement, and cost functions for VM provisioning and migration as input, suitable allocation of VMs across local servers and remote datacenters can be determined.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Performance	By a hierarchical grouping of VM mappings in scenario (1) and by pruning of distribution networks in (2), we strive for a single approach to manage both scenarios. Early results using integer programming techniques show how to optimize a utility function for service performance with, e.g., service layout (load bal-ancing), budget, VM configuration requirements as constraints, for a problem reduced (pruned) to a few tens of VMs and a handful of VM types and desti-nations with feasible solution time constraints for the placement process [20,37]. Our first prototype also takes cost and performance implications for actual VM migration operations into account and allows for modelling of uncertainties, e.g., due to provider's changing conditions or changes in the set of available destina-tions.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Performance	Self-management Challenges for Multi-cloud Architectures 45 about system load and services performance. This approach allows crisp defini-tion of complex BLOs, as this model makes it possible for the high-level manager to optimize some utility function and simultaneously enforce governance policies.	Ivon Miranda Santos
Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Performance	The governance model can be realized, e.g., through machine learning to allow the system (before training) to be unaware of the effect of individual policies to itself and to service performance [31]. 6 Live Migration of Large-Scale Virtual Machines	Ivon Miranda Santos

Elmroth2011-Self-management_challenges_for_multi-cloud_architec	CLOUD MANAGEMENT > Performance	A first prototype of this live migration algorithm has recently been implemented as a modification to the KVM hypervisor [34]. Performance results show up to a factor of 100 in reduced migration downtime for a synthetic benchmark, reduction of user-experienced service interruption from eight seconds to zero for live migration of a streaming video server, and successful migration of the large and complex SAP application in environments where the standard approaches fail [34]. However, extended migration downtime is not the only issue with the stan-dard live migration algorithms.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	Equally, cloud cost models have been investigated.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	Cloud migration · Cloud cost models · Monetisation Architecture migration · Independent software vendor · Cloud native	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	These all relate to defining a payment model for a quality product and making it viable considering the costs for providing cloud SaaS as well.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	4 Cloud Migration – Joint Architecture and Costing Concerns	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	– Cost model (expenses): the projected expenses in the cloud need to be deter-mine, which includes basic infrastructure and platform costs, but also all additional features for external access and networking, internal quality man-agement and possibly development and testing costs.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > multi-tenant	Following this process results in what is called cloud-native, and can be charac-terised through the following properties: scalable/elastic, clusterable, multi-tenant, pay-per-use, self-service This is assumed to have better scalability characteristics.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	And, furthermore, predicting the cost for a fully cloud-native solution is the challenge.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	6 Architecture Migration and PaaS Deployment Cost Calculation	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Monitoring	fees, where many providers offer monthly subscription fees, but extra fea-tures will incur extra cost: examples here are scalability, access (IP endpoint, network bandwidth) or monitoring and advanced self-management.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	What is needed is a calculation of the cloud costs versus the possible revenue result from the SaaS delivery model.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	– Predict costs based on cloud costs (from basic virtualisation to cloud-native) for an assumed usage (workload) of the SaaS application provided to cus-tomers.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Performance	The earlier pricing samples reflect this storage/processing need already. 8.1 Performance Experimentation and Prediction The first step was to determine performance statistics based on some assumed workloads: – standard operations were scaled up to estimated peak loads for the applica-tion, – response times and CPU/memory consumption were measured on different resource configurations to determine a link between resources needed and QoS provided, – resulting in the identification suitable configurations that would maintain SLA compliance for all customers.	Ivon Miranda Santos

Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Performance	A further collection of input data for the calculation includes: – to determine the number of users, – to determine the expected consumption/load of a typical user (based on the performance experimentation), – to determine licensing model options using different account types. This the definition of a pricing model maps costs per typical user, taken from the experiments, into a licensing model.	Ivon Miranda Santos
Fowley2018-Cloud_migration_architecture_and_pricing_–_Mapping_a	CLOUD MANAGEMENT > Cost	A holistic perspective on costing and architecture within a migration scenario does not exist.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Cost	The complexity of migration is exacerbated by the fact that some legacy applications may have been developed without taking into account the unique requirements attributed to cloud environments such as elasticity, multi-tenancy, interoperability, and refactoring. Such requirements raise new challenges to the migration of applications to the cloud and hence needs improving conventional software development methodologies to address these specific requirements. Various projects and studies in cloud computing community define migration approaches in order to enable legacy applications to take benefit from cloud services (e.g. reducing maintenance costs, economies of scale, and pay-as-you-go).	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	The complexity of migration is exacerbated by the fact that some legacy applications may have been developed without taking into account the unique requirements attributed to cloud environments such as elasticity, multi-tenancy, interoperability, and refactoring. Such requirements raise new challenges to the migration of applications to the cloud and hence needs improving conventional software development methodologies to address these specific requirements. Various projects and studies in cloud computing community define migration approaches in order to enable legacy applications to take benefit from cloud services (e.g. reducing maintenance costs, economies of scale, and pay-as-you-go).	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	Each migration type may raise different concerns. For example, if a migration type II is intended, where a legacy application is re-engineered to SaaS, then multi-tenancy aspects such as application customisability and resource provisioning are needed to be properly addressed by the application owner.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	However, in the case of encapsulating an application into a virtual machine and deploying it in the cloud (migration type V), enabling the feature multi-tenancy might be of less or not concern.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	2009, Brebner, 2012, Rimal et al., 2009, Guo et al., 2007, Nathuji et al., 2010, Toosi et al., 2014, Dalheimer and Pfreundt, 2009, Ristenpart et al., 2009), we identified six cloud intrinsic key concerns as follow: (i) resource elasticity, (ii) multi-tenancy, (iii) interoperability and migration over multiple-clouds, (iv) application licensing, (v) dynamicity and unpredictability, and (vi) legal issues. These concerns trigger considerations that an application owner should consider them in the migration process, though they might have been already automatically supported by cloud providers. The remainder of this section delineates these six concerns.	Ivon Miranda Santos

Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > multi-tenant	(ii) Multi-tenancy. In cloud environments, each service consumer is called a tenant (Rimal et al., 2009, Guo et al., 2007). Multi-tenancy is an ability to use the same instance of a resource at the same time by different tenants. On the side of cloud service provider, this maximises the resource utilisation and profit since only one application instance is required to deploy in the cloud. On the cloud consumer side, each consumer feels that he/she is the only users of the application. Tenants can customise application components, such as user interface appearance, business rules and sequence of workflow execution, and last but not least the application code.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Performance	As tenants are dedicated to the same instance of an application, there is often a concern that tenant's QoS is negatively affected by other tenants. For example, the performance of a tenant that uses one core of a 6 multicore processor may significantly be reduced when another tenant runs an adjacent core and performs a massive workload (Nathuji et al., 2010).	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Performance	This issue can also be re-interpreted from a security perspective where all the tenants may use the shared resources and a malicious tenant with a criminal mind can damage resources and pose a serious threat to all the tenants. Tenant isolation for QoS satisfaction (e.g. performance, security, availability and customizability) should be carefully addressed in a cloud application. (iii) Interoperability and migration over multiple-clouds.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Portability	, 2014). These issues face developers to heterogeneities across the application tiers, which imply a certain level of development effort, specifically in migration types I, II, III, IV, and V. As advancements in the cloud computing is still on on-going track and there is not a common standard for development cloud services, application portability is a challenge when its components are to move from a provider to another provider, but there is an incompatibility between underlying technologies of these providers (e.g. APIs).	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > multi-tenant	A criterion is included in the proposed framework if it had addressed at least one of the concerns stated in Section 2.2 and also sufficiently generic to cover a variety of migration scenarios regardless of a particular cloud platform. This resulted in defining 17 cloud-specific criteria including (1) Analysing Context, (2) Understanding Legacy Application, (3) Analysing Migration Requirements, (4) Planning Migration, (5) Cloud Service/Platform Selection, (6) Training, Re-Architecting Legacy Application (including (7) Incompatibility Resolution, (8) Enabling Multi-Tenancy, (9) Enabling Elasticity, (10) Cloud Architecture Model Definition, (11) Applying Architecture Design Principles), (12) Training, (13) Test and Continuous Integration, (14) Environment Configuration, (15) Continuous Monitoring, (16) Migration Type, (17) Unit of Migration. These criteria helped the study to contrast and compare cloud-centric aspects of existing approaches.	Ivon Miranda Santos

Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Portability	The most applied paradigm is model-driven development with 7 approaches. Its primary goals are portability, interoperability, reusability of applications as well as and increasing development speed. Applied in cloud migration, it meant to transform the legacy application models (e.g. codes and architecture) into platform-independent models, configure them and then generate platform-specific cloud applications using model transformation techniques.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Cost	V Not-specified Supporting decision making during the suitability analysis of migration in terms of operational cost, organisational change, energy consumption, and stakeholder analysis.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Cost	(i) determine the type of the application is to be migrated to the cloud since some applications may not benefit from the cloud such as safety-critical or embedded applications, (ii) effort and cost that required for the migration regarding perceived benefits.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Cost	This avoids from unexpected cost of cloud service usage when bills are issued.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Performance	Andrikopoulos et al. [S35] numerate several factors are taken into account for component selection such as data privacy, expected workload profile, acceptable network latency and performance variability, availability zone of cloud providers, the affinity of components in the cloud, and the geographical location of cloud servers. Among the reviewed approaches, Leymann et al.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Performance	It consists of three activities: (i) describing required computing resources, the life cycle of the application and its offered services, and scaling rules of the application and its services, (ii) automatically provisioning resources from selected clouds for the application, and (iii) monitoring and ensuring application performance. Two approaches rather provide high-level advice:	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > multi-tenant	(iii) Enabling Multi-Tenancy. According to (Bezemer and Zaidman, 2010, Guo et al., 2007), re-architecting of a legacy application to support multi-tenancy includes the following aspects:	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > multi-tenant	As off-the-shelf database management systems might not support multi-tenancy (Jacobs and Aulbach, 2007), securing the data tier of application should be properly addressed.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > Performance	The application should monitor its internal state to detect the faults, prevent its propagation, and repair it in a timely manner. — The fourth aspect is performance isolation which is to guarantee the performance of one tenant from the negatively being affected by the performance usage of other tenants in unforeseen behaviors. If this is satisfied in all situations, then the application is performance-isolated. Given the above aspects of multi-tenancy, only the approach proposed by Maenhaut et. al.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey __evaluation_framew	CLOUD MANAGEMENT > multi-tenant	That is, enabling multi-tenancy requires the following steps: (i) decoupling databases, (ii) adding tenant configuration databases, (iii) providing tenant configuration interface, (iv) dynamic feature selection, (v) managing tenant data, users, and roles, and (vi) mitigating security risks.	Ivon Miranda Santos

Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	Andrikopoulos et. al. [S35] point the concept multi-tenant aware applications and recommend addressing two aspects of multi-tenancy as (i) supporting message exchange isolation per tenant, and (ii) administrating and configuring the application per tenant. Other approaches only recommend considering multi-tenancy activity without any proposal on required activity to perform.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Portability	If the code refactoring, as mentioned in the previous item, is costly, then developing integrators/adaptors can be served as an alternative solution to hide incompatibilities. Adaptors provide an abstraction layer, keeping the application code untouched and facilitating application interoperability and portability. With this regard, the ADM (Architecture-Driven Modernisation) approach of Zhang [S12] suggests developing wrapper layers (e.g.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Performance	—Transient Fault Handling. Performance variability and network latency can have an impact on QoS of migrated application. Developers should implement mechanisms to detect and handle transient faults that occur in cloud environments.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	On the other hand, developers need training on new programming concepts such as asynchronous interaction, distributed state and session management, caching, scale out across data centers and providers (scalability), multi-tenancy [S35].	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Performance	Therefore, a certain level of integration and interoperability testing is required. Important tests that should be taken into account are Data Verification, Test Backup and Recovery Plan before they required [S23], Test Network Connectivity (connection between cloud services and local network), Test Connection Speed as there is network latency to receive a response from the cloud server located in different geographical areas [S33] [S40], test provider performance variability and test application latency due to the network performance variability [S35]. An important aspect of the test is to address continuous integration.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Performance	In the staging environment, developers can test the application within a production-like environment (.e.g. infrastructure configuration). That is, a deployment environment in which all the tests to be performed in order to detect bugs, performance and other issues before the application can be used by users. All releases of application are tested first on the staging environment.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Monitoring	—Create installation scripts and setup different third-party libraries and tools which may be used for monitoring and reporting runtime application behavior, though this needs less effort if cloud provider has already done it	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Monitoring	5.4.12 Continuous Monitoring The dynamic and unpredictable nature of the cloud environments necessitate continuous monitoring of application and cloud resources to assure successful SLAs (e.g. [S1], [S4], [S22], [S23], [S26], [S28], [S33], [S38], [S40], and [S43]) as described in followings	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Performance	— Measuring. Collect critical data about the application health and performance, traffic patterns, vital signs such as CPU and memory usage, network traffic, and disk usage. This data are used to detect deviations from SLA and runtime application adaptation and optimisation.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	Lack of adequate support for multi-tenancy, elasticity, test and continuous integration. One interesting observation from reviewing 43 approaches is that only [S35] covers multi-tenancy merely from the customisation aspect.	Ivon Miranda Santos

Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > multi-tenant	with relevant method fragments to the aspects multi-tenancy, elasticity, test, and continuous integration which have been weekly supported. Fourthly, the definition of new roles specific to cloud application development has not been explored in existing approaches.	Ivon Miranda Santos
Gholami2016-Cloud_migration_process—a_survey__evaluation_framew	CLOUD MANAGEMENT > Performance	In addition, journals, conference, workshop proceedings and technical reports attributed to Cloud Computing and SOA areas were sought. These included IEEE Transaction of Cloud Computing, IEEE Transaction of Service Computing, Software Engineering for Cloud Computing, Cloud Computing International Conference, Cloud and Service Computing International Conference, International Conference of Service-Oriented Computing, Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, International Conference on Cloud, Service-Oriented Computing and Applications, and International Conference on High Performance Computing and Communications. Step 3 Defining Study Inclusion and Exclusion Criteria.	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > multi-tenant	• Multitenancy—In multitenancy, multiple users share a single version of soft-ware. It has economic impacts on an unprecedented scale. Vendors often fall back on multitenancy due to advantages of data aggregation. Information about different customers is stored in one scheme of a database, facilitating mining of data and running queries. However, it is not without its security threats.	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	The back end comprises of servers that run the cloud along with infrastructure pertaining to data storage. Advantages of using cloud database are economic, improved performance and efficiency, software updates, improved document compatibility, improved group collaboration and high storage. 7 Virtualization	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	Xen hyper-visors schedule tasks, manage memory and delegate I/O operations to the privileged drive [22]. VMware supports both full and para-virtualization, the choice based on relative performance [31]. 7.2.3 Hardware-Assisted Virtualization	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	It reduces the requirement and cost of configuring a large number of desktops in small- and medium-sized businesses. Using hardware-based GPU sharing one can access high-performance applications from any device over a secure network. 64 M. K. Gourisaria et al.	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	8.1 Hot Migration Hot migration refers to the movement of applications and the operating system from virtual machines to the hardware of a computer without hampering the operating system or application performance. It is useful as it frees as a particular server without providing any disadvantage to users, thereby providing smooth perfor-mance. It also prevents inefficient servers from causing discontinue in operations.	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	The purpose of using cold migration is to reduce complexity and allow the transfer to be more effective. However, the downtime might affect the performance of business and thereby, cold migration is a challenge faced by companies. 8.3 Migration Risks	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	Risks associated with migration can be broadly classified into the following: • General risks—In general risks, we look into tuning and monitoring of performance. • Security risks—There are a number of legal consents that migration has to abide by, which includes accessing execution logs and preserving the rights to review trails.	Ivon Miranda Santos

Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Performance	As a result, multiple VMs may be mapped to a single server [39]. Makespan (MS) and energy consumption (EC) are two of the most important factors in assessing the perfor-mance of a VM. Makespan of a system is defined as the maximum time consumed by a VM to carry out all the tasks in the queue [40, 41]. Minimizing the makespan optimizes the performance of a load balancer. The response time (RT) of a VM also plays an important role. It is the total time taken to respond to a task, and hence, lower the response time of a VM, better its performance. A low RT is essential for an optimized MS [42].	Ivon Miranda Santos
Gourisaria2020-An_Extensive_Review_on_Cloud_Computing	CLOUD MANAGEMENT > Portability	The CSA guide deals with the following domains in cloud computing— cloud construction framework, legal aspects, business risk management, audit and compliance, portability, information cycle handling, electronic discover, traditional security, disaster, recovery and business continuity, incident response, remediation and notification, data centre operations, access management and identity, key management and encryption, application security, virtualization and storage. 11 Conclusion	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	In this paper, we tackle challenges in migrating enterprise services into hybrid cloud-based deployments, where enterprise operations are partly hosted on-premise and partly in the cloud. Such hy-brid architectures enable enterprises to benefit from cloud-based ar-chitectures, while honoring application performance requirements, and privacy restrictions on what services may be migrated to the cloud. We make several contributions.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	Our model takes into account enterprise-specific constraints, cost savings, and increased transaction delays and wide-area communication costs that may re-sult from the migration.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	Cloud computing promises to reduce the cost of IT organizations through lower capital and operational expense stemming from the cloud’s economies of scale, and by allowing organizations to pur-chase just as much compute and storage resources as needed, only when needed.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	, [11]). On the one hand, enterprise applications are often faced with stringent require-ments in terms of performance, delay, and service uptime. On the other hand, little is known about the performance of applications in the cloud, the response time variation induced by network la-tency, and the scale of applications suited for deployment. Further, industry-specific regulations (e.g.,	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	Enterprise applications are typically composed of multiple compo-nents, and hybrid architectures allow for individual components to be migrated, or kept local. Hybrid architectures offer enterprises flexibility in decision making that can enable them to find the right balance between privacy considerations, performance and cost sav-ings. For instance, sensitive databases (e.g.,	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	Our model takes into account enterprise policies, cost savings from migration, and increased transaction delays and wide-area communication that may result from the migration.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	Multiple factors can motivate such hybrid deployments. From a performance perspective, migrating the entire application to the cloud is likely to result in higher response times to users inter-244R2 R1 ACL1 ACL2 R4	Ivon Miranda Santos

Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	The key factors are (i) honoring pol-icy constraints regarding which components must be migrated; (ii) ensuring application response times continue to meet desired tar-gets; and (iii) ensuring the cost savings from migration are as high as possible.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	InternetCosts(M) is the increased communication costs since traf-fic between the data center and the cloud is now sent over the In-ternet, and DelayIncrease(M) is the increase in transaction delay.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	where CostL,I and CostR,I are respectively the per-unit Internet communication cost of traffic from the local and cloud data centers,	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	3.5 Modeling benefits of migration There are several factors that can enable enterprises to reduce their costs as they migrate to the cloud.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	In this paper, we focus on the recurring costs of migration, such as server and wide-area Internet communication costs.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	Executing the migration process may involve one-time costs, such as the ef-fort in acquiring model parameters, and reengineering applications for cloud deployment.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	Many reachability policies apply to sets of entities. Representing each cell as ACL rules unrolls such a structure and hence could significantly increase the number of rules required to be installed, adding extra processing overhead which could degrade network performance after migration. We instead adopt an alternate approach which preserves infor-mation regarding common entity pairs affected by each ACL.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	We view DCN and DCS respectively as the local and cloud data centers. We picked a host (I) located in geographical proximity to DCN as seeing per-formance representative of an internal enterprise user. We picked about 20 Planetlab hosts (O) scattered around the United States, and assumed they corresponded to external users.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	Modeling migration benefits and communication costs:	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	We as-sume that migrating servers to the cloud can reduce costs by a factor of 7 for compute-class servers, and 5 for storage-class servers, as suggested in [14].	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	We leverage the Amazon EC2 cloud pricing [1] to calculate the cost of running a server in the cloud.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	Finally, we used the same val-ues for migration benefits and communication costs as in §5.1.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	Subset of ACLs inheriting rules from a3 after migration. 5.3.3 Performance and scalability To evaluate the scalability of our ACL migration algorithm, we ran it on the reachability policies of the entire campus network, which is a distinct rule-set from that described in § 5.3.1 corresponding to policies in the campus data-center. The network consists of 700 VLANs and 212 ACLs.	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Cost	net communication costs, if deployed at both the local and cloud	Ivon Miranda Santos
Hajjat2010-Cloudward_bound_Planning_for_beneficial_migration_of	CLOUD MANAGEMENT > Performance	Transaction counters and service response time measurements for each component are widely available on enterprise servers today. Most enterprise software packages provide embedded performance monitoring capabilities that can be enabled on the servers them-selves to track and report these performance numbers [5, 12]. Finally, inaccuracies in estimates of model parameters could be dealt with by running the model with multiple sets of inputs and choosing a conservative plan to ensure application response times are met.	Ivon Miranda Santos

Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	This paper suggests an approach for migrating monoliths to microservice-based Cloud-native SaaS, providing customers with a flexible customization opportunity, while taking advantage of the economies of scale that the Cloud and multi-tenancy provide.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	Reliability is impacted as even simple bug fixes cause the entire application to be updated in every deployment. Following the trend of cloud computing, enterprise software vendors are moving from single-tenant on-premises applications to multi-tenant (Cloud native) SaaS [2]. Customer companies no longer buy a license from the vendor and install software products in their own premises.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	Both microservices architecture and multi-tenancy offer additional benefits to the end-users of the application and the developers.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > DevOps	Migrating to microservices architecture (MSA) is the right way forward for legacy systems to be modernized [3], [4]. There are huge benefits for migrating to MSA such as maintainability and scalability in the long run [5], e.g., by adopting DevOps and benefiting from Cloud-native elasticity [6]. Microservices can be packaged and deployed in isolation from the main product, which is an important requirement for multi-tenant context.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	, by adopting DevOps and benefiting from Cloud-native elasticity [6]. Microservices can be packaged and deployed in isolation from the main product, which is an important requirement for multi-tenant context. Moreover, independent development and deployment of microservices ease the adoption of continuous integration and delivery, and reduce the time to market for each service.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	Our approach focuses on three stages during the migration, analyzing and breaking down the application into small bounded contexts, transforming the existing infrastructure to fit the new architecture and implementing functionality from the contexts as separate microservice, and finally adding the necessary components to support tenant-specific customization in the multi-tenancy context	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	D. Multi-Tenancy A multi-tenant application serves multiple customers or ten-ants through an application shared by all the users [11]. Multi-tenancy is prevalent, particularly in cloud-hosted software. Since the application instance is shared among the different users, the software only solves a common set of problems for the users or a problem that the majority of the different users have [12]. Since the application is shared among multiple tenants, costs associated with the infrastructure and operations of the servers are also shared between the tenants, resulting in lower overhead for the application compared to running individual instances for each customer	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	Retailers could have different business models leading to different requirements for customization. Following the trend of cloud computing [2], software vendor A is migrating their software products such as SportStore to become multi-tenant (Cloud-based) Software as a Service (SaaS). Customer companies such as sporting goods retailers no longer buy a license from software vendor A and install it in their own premises.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	IV. OUR MIGRATION APPROACH In this section, we give a brief overview of our approach (IV-A) and how it relates to multi-tenancy and the ability to provide deep customization for tenants (IV-B).	Ivon Miranda Santos

Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	Note that in this paper we do not address the constraints of migrating currently in-use applications, which the Strangler approach [8] can do best. We rather focus on how to logically migrate a monolithic application to become customizable in a multi-tenant context. Our approach can be adopted to be part of the Strangler approach for migrating currently in-use applications.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	B. Multi-tenancy and Deep Customization Our approach aims at enabling the target architecture to be customizable for multi-tenant context as presented in [15], [16]. The approaches in [15]–[18] offer tenants a way to (deeply) customize the functionality of the multi-tenant ap-plication without interfering with behavior for other tenants. The customization-driven aspect makes our approach differ-ent from other migration approaches. Adding customization support for tenants can be done using the tenant-manager as a lookup table (Fig. 2). Tenants register their customizations with the tenant manager.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	First, we focus on introducing multi-tenancy to the applica-tion. To support multi-tenancy, we need a system for Identity Access Management (IAM), and to support customization of the application for the tenants and to configure the storage to isolate tenant data, we need a tenant manager	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	We then implement the groupings we identified during the first phase as separate micro-services with isolated storage. Once we have extracted the services from the old application we connect them to multi-tenant specific infrastructure. SportStore is implemented in .N	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	4) Tenant Manager: The tenant manager is an essential component of the multi-tenant aspect of the application. We use it to configure the persistency layer for the tenants and as a lookup for customized endpoints.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	C. Implementation In this section, we go through the process following our migration approach and applying it to the migration of the SportsStore application, first to the microservice architecture, and then implementing multi-tenancy for the application. We split the migration up into different phases. Each phase in-cludes the extraction of a single service from the pre-existing system, as well as adding the necessary infrastructure to support the new migrated functionality from the monolith.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	The final phase of the migration introduces more infrastruc-ture to support multi-tenancy.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	RELATED WORK This section discusses related approaches for migrating from single-tenant to multi-tenant, and monoliths to MSA. A. Single-tenant to Multi-tenant One of the primary challenges with multi-tenant applica-tions, according to Kwok et al. [11] is that the application has to deliver a shared product to multiple tenants, resulting in one-size-fits-all solutions even though the different user-groups might have slightly different needs from the application.	Ivon Miranda Santos

Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	[12]. Ad-hoc handling of changes related to one specific tenant can potentially affect all the tenants of the application Multi-tenant applications is another way to take advantage of economies of scale. Multi-tenant applications share resources between multiple different user groups or tenants, keeping the tenant-specific data separate. Migrating an application from single to multi-tenant is a large undertaking. There are certain requirements that need to be in place before the application can be migrated, Furda et al. [19] describe an approach for migrating legacy single-tenant applications to multi-tenant. The approach moves through three different phases during the migration.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	The second phase focuses on changing the design pattern of the application into one that is better suited for multi-tenancy.	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	Once the functionality of the application has been migrated, we add the infrastructure necessary for multi-tenancy	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	When it comes to the multi-tenancy transition, there are	Ivon Miranda Santos
Haugeland2021-Migrating_monoliths_to_microservices-based_custom	CLOUD MANAGEMENT > multi-tenant	The primary concerns with multi-tenancy are avoiding noisy neighbors and ensuring that the tenant data is sufficiently isolated.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	With the promise of providing flexible and elastic comput-ing resources on demand, the cloud computing has been at-tracting enterprises and individuals to migrate workloads in the legacy environment to the public/private/hybrid clouds. Also, cloud customers want to migrate between cloud providers with different requirements such as cost, performance, and manage-ability. However the workload migration is often interpreted as an image migration or re-installation/data copying as the exact snapshot of the source machine.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Cost	Enterprises are increasingly migrating their existing IT infrastructure to the cloud, driven by the promise of low-cost access to elastic resources [1, 2].	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	Also, as the competition becomes stronger, cloud providers increasingly offer more diversified services and they differentiate their catalogs with more advanced service features [3, 4]. A deterrent for enterprise migration to the cloud is a lack of migration planning tools that can scrutinize the discovered on-premise data, provide comprehensive analytical information to reason about why the migration can help reduce opera-tional expenses and increase performance, and finally create a detailed migration plan [5, 6, 7]. Existing tools aim to only provide an one-to-one migration that just copies a source image into a target image, but they do not find themselves as the comprehensive end-to-end migration toolings [8, 9].	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	As the target environment becomes more diversified, the complexity on deciding where to move rather than how to move renders migration analytics a harder problem. The pos-sible choices for target environments include container, (pub-lic/private) virtual machine, baremetal, POD (Performance Op-timized Data Center), datacenter, geo-location, cloud provider, service model. As shown in Figure 1 that illustrates selectable cloud model, this collection can be represented as a hierarchi-cal structure.	Ivon Miranda Santos

Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	TABLE I. TRANSFORMATION METRICS FOR EACH SERVER Finally to help make decisions on server decomposition and consolidation, we need to identify versions such as software/operating system versions, and library versions, and performance characteristics such as cpu/memory/disk/network intensity. Since performance attributes are observed for a pe-riod of time, the performance characteristics of servers also can be determined based on the observed data. Version information resolves the software conflicts for installing softwares that share the same operating system or libaries.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	For instance, if an application uses a python module twisted, it requires to run on python 2.x because python 3.x version does not yet support the twisted module. The performance characteristics also are very useful to determine the server consolidation. For instance, if two servers have cpu intensive characteristics, they should be ruled out from the consolidation candidates because the applications inside the two servers may conflict to get cpu resources to run the processes.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	This is an iterative process to find the best resource matching in order to design the optimum target cloud. The resource planning step scrutinizes various factors including performance metrics, system properties, and affinity data in order to decide the best resource allocation for each server. It iteratively decides whether or not consolidating servers into one server or decomposing a server into multiple servers (or containers).	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	One potential pitfall of the migration is that the migration engineers tend to think the source resources are optimized to run the existing applications. However from the sample performance monitoring data shown in the Figure 3 from the real enterprise datasets, we can easily verify the maximum usage of the resources are not close to the allocated capacity. The cloudish approach is to assign the maximally required resource at any given time, and scale applications elastically when the usage increases and the better performace is required. Therefore after finding the resources, we adjust the resources based on the maximum performance used during the period of time. For example, the souce machine with 4 cores (400%) runs CPU in average 32%, and maximum 120%, we subtract 2 cores, thus assign 2 cores.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	If an administrator can change the default ports manually, this constraint can be passed. • Performance intensity should be different. For example, two applications that have a CPU intensity should not be considered to merge into the one server. I(Si) ▯= I(Sj), where I(·) is the performance intensity of the server (i.e., CPU, memory, disk, or network).	Ivon Miranda Santos

Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	E. Server Decomposition Many applications, if not designed for distributed de-ployment, tend to run on the same server because they do not have any conflicts on the configuration such as network ports, and it is convenient for administrators to have them managed in the same server. Also in the perspective of performance, applications run without network delay, which can incur performance degradation. However this often brings troubles because applications running on the same server can use the same resource simultaneously, and this can result in slowing down each other. Not only the performance issues, the scalability is an another issue because, for example, the database can not easily be scaled out when it is running in the localhost, together with a consumer application. When an administrator wants to redesign application de-ployment to increase performance and support an elastic scalability in the target cloud, a server that runs multiple networked applications needs to be decomposed into separate components. For example, a high-end physical server that runs many applications can be decomposed into many virtual machines, each of which has an application, and the decom-posed components are configured to communicate through the network.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	Also, we use the utilities such as top and Windows Task Manager to gather further information. The application performance is important to de-cide how much resource it needs and what platform it should run on. Obviously, it is a bad practice to run a multi-process application only in one container.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	Particularly, we focus on the following goals to check: • resource planning effectiveness with performance met-rics (§IV-B), • resource matching ratio between the planned resources and target catalogs (§IV-C),	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	The collected data include metrics in Table I at the minimum and more such as server affinity. Performance items are measured for two weeks period. For target catalogs, we use one cloud provider, SoftLayer, in the experiments, but the same experiment can be easily performed to any other cloud providers as well.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	B. Resource Planning When planning resources, after discoverying, an immediate step is to identify whether resources can be adjusted at the target environment. This leads to the question whether or not we can save expenses without sacrificing performance. To an-swer this question, we scrutinize both performance and system properties to find out how much overprovisioned resources are. The main objective is shown in Equation (1) that minimizes the overall costs through the resource planning.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	This is because the baremetal can consolidate more virtual servers in one physical server. Since the resource adjustment is based on the maximum resource usage in its peak time, we do not expect to see any performance degradation due to the resource adjustment. The overall potential resource adjustment ratios and cost savings are significant, yet we still want to look deeper ramification through the distribution function on the resource adjustment.	Ivon Miranda Santos

Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	However the cdf of memory size shows 20% - 30% machines largely underutilize the memory resources. C. Resource Matching Once the resource planning is made by adjusting the resources based on the performance and cost analysis, it is time to verify those resources can fit into the cloud environment us-ing the target catalogs. In the custom virtual environment using a baremetal server and own hypervisor, allocating maximum resources is allowed.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	However baremetal servers usually do not have the limitations on provisioning resources so that almost all of servers can fit. Still, there are some servers that require extremely high performance. We notice some custom-ordered machines can not usually be supported at the target environment.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	One simple example is that a load-balancer can scale out services by distributing service requests. Another example can be adding a memory caching layer in order to compensate the database performance. D. Server Consolidation As explained in Section III-D, applications or services can be consolidated into a server for the sake of reducing operational expenses if their performance does not degrade. To decide the consolidation, conflicts need to be resolved in order not to disrupt services.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	First of all, resource intensities of applications should not be overlapped. For instance, two applications that are CPU intensive should not be consolidated into one server because they will definitely fight for more CPU resources, meaning the performance degradation. Next, the system properties should be disjoint.	Ivon Miranda Santos
Hwang2015-Computing_resource_transformation__consolidation_and_	CLOUD MANAGEMENT > Performance	So the condition for the server decomposition is that applications or services should be able to run as modules through network connections. Through the decomposition, we expect more manageability and improved performance. As depicted in Figure 10 illustrates the ratio of potential decompositions, the ratio of server decom-position is very high, that is 60% servers can be decomposed into multiple servers for virtual servers.	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > Portability	There are lots of focuses on reducing and optimizing resources and hence developing application in a serverless fashion is going to be the key in the industry. Many organizations are working on application modernization and developing distributed application and hence microservice is the key focus area in converting their monolithic into microservices and use containers for easy portability across the platform and makes it more platform neutral. There are high elements of focus on whether to go for serverless or Microservice mode and should we use containers for deployment is the key debate among the people who are working in this area and are still not clear which way to go forward in the given situation.	Ivon Miranda Santos

Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > multi-tenant	Although there are some advantages, it also has few drawbacks like any other technology such as, Vendor control, multitenancy problems, vendor lock-in, and security concerns are some of the problems due to the use of third-party APIs. Developers are dependent on vendors for debugging and monitoring tools. Architecture complexity - It gets cumbersome to manage too many functions, and ignoring granularity will end up creating mini-monoliths. Integration testing serverless apps is tough. The units of integration with Serverless (Functions) are a lot smaller than with other architectures and therefore we rely on integration testing a lot more than we may do with other architectural styles.	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > Performance	In AWS lambda, we can run our code in response to events, such as changes to data in an Amazon S3 bucket or an Amazon DynamoDB table. With these capabilities, we can use Lambda to easily build data processing triggers for AWS services like Amazon S3 and Amazon DynamoDB process streaming data stored in Kinesis, or create your own back end that operates at AWS scale, performance, and security. Azure Functions lets you develop serverless applications on Microsoft Azure.	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > DevOps	There are many containers available but docker is the most accepted one and are open source, have its own ecosystem to manage. Like VM, container also has its own life cycle to manage and every cloud service provider, be it a public cloud like AWS, Azure, Google or private cloud like OpenStack, VMware supports Docker containers. For example – AWS has elastic container service (ECS)[4] and Azure provides Azure container services. They manage with the native services and also provide supports to Dockers. In addition, all these service providers support DevOps services to manage operations and support Orchestration tools like Kubernetes[2], Docker Swarm[6] and Mesos[6].	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > DevOps	Another advantage is that it collaborates well and services written in different languages can interact with each other while addressing the business functions. It also simplifies build and release process and support complete devops functionality. Every cloud service provider supports Microservices architecture and has its own API and services to support this development model and enable developers to build a truly distributed application.	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > Portability	Hence each service can be deployed in a container and can be ported across platform to make it secured and truly distributed in nature. Hence containers portability and light weight behaviour and microservice, scalable, and distributive nature – gel well with each other and helps in modernizing application to meet the business demand. Authorized licensed use limited to:	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > DevOps	As microservices enable smaller, faster releases, they allow new features to be released to only a subset of users initially, and then to the entire user base once the feature meets quality expectations. Even though microservices architecture is more complex, especially at the start, it brings much-needed speed, agility, reliability, and scalability, which are critical to today's DevOps teams	Ivon Miranda Santos

Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > DevOps	Containerisation helps in easy deployment in any cloud services or any PaaS platforms like Pivotal cloud Foundry[8], OpenShift[8] or Bluemix[8]. These PaaS providers have inbuilt kubernetes and container management platform to manage this containers effectively and made it as a self-service platform. It has complete support for build and release process including most of the devops features which enable them to host the containers and manage them completely. Mesos kind of tools provide the support to containers for even data centre servers or even our laptops and make the end to end orchestration services for containers. Hence in our analysis and observation, containers are the key for hosting the applications – whether it is monolithic, microservices or serverless and many framework provides complete support in managing containers – Dockers.	Ivon Miranda Santos
Jambunathan2018-Architecture_decision_on_using_microservices_or	CLOUD MANAGEMENT > Performance	Hence need to use appropriately. d) Although a single microservice can be converted into multiple functions, it cannot be monitored or measured hence; too many functions will impact the performance. e) Containerizing microservices makes sense as it runs in an isolated, independent form, where as it is not always recommended to containerize functions as it is not completely independent.	Ivon Miranda Santos
Jamshidi2013-Cloud_migration_research_A_systematic_review	CLOUD MANAGEMENT > Cost	However, cost savings are immediate when a cloud environment is leveraged.	Ivon Miranda Santos
Jamshidi2013-Cloud_migration_research_A_systematic_review	CLOUD MANAGEMENT > multi-tenant	Process IV. Crosscutting concerns. Some crosscutting tasks such as governance [S8], security analysis [S8] [S17] [S22], training [S6] [S7] [S16], effort estimation [S18], organiza-tional change [21], and multitenancy and elasticity analysis [S8] act as umbrella activities in the framework.	Ivon Miranda Santos
Jamshidi2013-Cloud_migration_research_A_systematic_review	CLOUD MANAGEMENT > Cost	inherently complex and influenced by multiple factors such as cost and benefits through migration [33].	Ivon Miranda Santos
Jamshidi2013-Cloud_migration_research_A_systematic_review	CLOUD MANAGEMENT > multi-tenant	. Since cloud applications are designed to support multitenancy, they face more different contexts in comparison with on-premise applica-tions.	Ivon Miranda Santos
Jamshidi2013-Cloud_migration_research_A_systematic_review	CLOUD MANAGEMENT > Cost	As a result, cloud resource consumption for cost-saving purposes has to be adapted to varying contextual conditions.	Ivon Miranda Santos
Jamshidi2013-Cloud_migration_research_A_systematic_review	CLOUD MANAGEMENT > multi-tenant	Crosscutting concerns: governance, security, training, effort estimation, organizational change, multitenancy	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Performance	To account for the situational context of applications, e.g., security, performance, availability needs, existing approaches [1] suggest a trade-off between flexibility and ease of migration using a fixed set of migration strategies. We propose an assembly-based approach based on our experience in situational method engineering [8] where a method is constructed from reusable method fragments and chunks [9].	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Performance	In the cloud, the deployment of high-usage components can be optimized independently of low-usage ones. Re-architecting into independent components reduces dependencies and enables optimization for scalability and performance. However, challenges remain:	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Cost	The solution is improved though best-in-class cloud services, Re-engineering costs and effort are saved.	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Cost	Applications in a public cloud platform can take advantage of economies of scale and have automated processes for managing.	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Portability	Applications located in the public cloud are available over the Internet, but authentication concerns exist. A third goal is portability, i.e., it can be moved between a cloud and a private data	Ivon Miranda Santos

Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Performance	Step 3. Abandon the own payment application and rent a typically more generic cloud service, which needs to be evaluated regarding security, performance, and usability. MP12, MP13, MP14 suit, but a need to integrate Expense with a Payment service, favors MP13.	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Performance	Step 7. Value-added services from the cloud such as caching can maximize perfor-mance when retrieving data or can cache output, session state and profile information MP3 was selected to accommodate these environmental services of the cloud provider. Migration Path.	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Performance	For example, MP1 step 1 follows a gradual migration by adopting the hosting approach, but uses SQL Server hosted in a VM before moving to an Azure SQL Database. Using MP3 instead would take advantage of storage capabilities (table/blob storage) and caching instead of relational databases to improve performance early rather than late. Migration step Requirement Chosen patterns 1 Minimal code changes to application and familiarity with platform	Ivon Miranda Santos
Jamshidi2015-Cloud_migration_patterns_A_multi-cloud_service_arc	CLOUD MANAGEMENT > Cost	, [14]) is inherently complex and influenced by multiple factors such as cost and benefits through migration [15].	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	V-PAM defines activities to plan and execute cloud migration [7] based on the concept of patterns or templates, here describing the entities involved in the process. To account for the situational context of applications, for example, security, performance, availability needs, existing approaches suggest a trade-off between flexibility, and ease of migration using a fixed set of migration strategies [1]. We propose an assembly-based approach based on our experience in situational method engineering [8] where a method is constructed from reusable method fragments and chunks [9].	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	In the cloud, the deployment of high-usage components can be optimized independently of low-usage ones. Re-architecting into independent components reduces depen-dencies and enables optimization for scalability and performance. However, challenges remain:	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	quality-of-service (QoS) requirement for the application (e.g., performance, availability, reliability, and security requirements)? Configuration.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	,[1, 7, 16–21]), and on our on experience in deploying real-word applications in the cloud [22]. These constraint types include financial, organizational, security, communication, performance, and availability constraints. They should all be defined and evaluated within the same context.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Cost	Financial constraints. An example is the cost to operate the application in the cloud. Calculating this cost may be non-trivial, involving technical (e.g., number and types of cloud resources required) and non-technical (e.g., expected user demand) factors.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	PATTERN-BASED MULTI-CLOUD ARCHITECTURE MIGRATION 1165 Performance constraints. These relate to the capacity of the application to serve its users in a timely manner.	Ivon Miranda Santos

Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	VARIABILITY APPROACH TO MIGRATION DEFINITION In order to build manageable and scalable cloud applications that meet the communication, availability or performance constraints just discussed in previous section, a multi-cloud deployment is often appropriate [23, 24]. The V-PAM method proposed in this work aims at facilitating migration pattern selection and customization for applications that run on multiple independent clouds.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Portability	Applications located in the public cloud are available over the Internet, but authentication concerns exist. Another goal is portability, that is, it can be moved between a public cloud platform and a private data center without modification to application code or operations. Furthermore, a tractable migration plan to the cloud platform is essential.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	3. Abandon the own payment application and rent a typically more generic cloud service, which needs to be evaluated regarding security, performance, and usability. MP12, MP13, MP14 suit, but a need to integrate Expense with a Payment service favors MP13.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	7. Value-added services from the cloud such as caching can maximize performance when retrieving data or can cache output, session state and profile information. MP3 was selected to accommodate these environmental services of the cloud provider.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	For example, MP1 step 1 follows a gradual migration by adopting the hosting approach, but uses SQL Server hosted in a VM before moving to an Azure SQL Database. Using MP3 instead would take advantage of storage capabilities (table/blob storage) and caching instead of relational databases to improve performance early rather than late. The result of applying the steps to the source architecture is the architecture provided in Figure 8.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	These include completed or ongoing migrations‡ of an e-commerce application with high availability and performance needs, a document processing system that needs a multi-cloud integration with ERP system compo-nents,	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > multi-tenant	Applications: from traditional structure data-oriented transactional processing to high-volume, high-speed image processing and multi-tenant mobile applications that can handle billion transactions. Sectors: from software vendors to larger financial services and food sector.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Performance	Here a structured, systematic approach proved to be a valuable solution that has helped to deliver cloud solutions that meet the expectations and to keep the projects on track and avoid unnecessary delays. A common problem during migration is the need to refactor the architecture if the aim is to fully benefit from cloud performance and flexibility promises. For instance, the storage refactor-ing options relating to relational, table and blob storage, that we investigated and documented in [30], are particularly addressed by patterns MP1 and MP3.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Cost	There, we highlighted the re-architecting options that advanced PaaS clouds offer, but also showed that while quality concerns such as scal-ability or availability are covered, their quantification and a trade-off analysis with cost aspects is not covered.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > Cost	, [17, 21, 36, 37]) is inher-ently complex and influenced by multiple factors, such as cost and benefits through migration [38].	Ivon Miranda Santos

Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > DevOps	A system can be evolved in terms of three aspects, including re-architecting the current system, introducing new supporting components, and enabling Continuous Delivery using containeriza-tion in the context of DevOps [48]. Using a Situational Method Engineering migration approach proposed in [49], a monolithic source architecture can be migrated to a target microservices archi-tecture through reusable migration patterns (see our initial catalogue of microservices migration patterns in [49]). This confirms our assumption that the pattern-based migration approach proposed here can be suitable for all kinds of service-oriented architectures.	Ivon Miranda Santos
Jamshidi2017-Pattern-based_multicloud_architecture_migration	CLOUD MANAGEMENT > DevOps	Our implicit assumptions here included the possibility to compo-nentise legacy applications and also to target a cloud native architecture. Our brief discussion of microservices as a recently emerging cloud native architectural style demonstrates the importance of servitisation, but also the need to provide a framework that is generic enough to support the different service flavors in the context of DevOps	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	CNA principles describe recurring principles how CNA prop-erties are achieved and how transferability of a CNA can be realized. According to selected papers, CNAs should be operated on automation platforms. Softwarization of infras-tructures should be strived for to support DevOps principles more consequently. Operation of CNAs in multi- and hybrid clouds should by supported by applying migration and interoperability principles.	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	CNA methods include patterns and design methodologies to create effective CNA architectures and DevOps to automate the process of delivery and infrastructure changes. These methods are applied frequently in CNA context.	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > Cost	Other keywords dealt with special aspects like forensics of cloud-native artifacts and cloud costs and accompanying decision making models. However, it seems that there is no clear focus on these aspects associated with the term "cloud-native" (so far).	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > Cost	However, the reader should be aware that we did this detail research only for the identified main research topics and not for mentioned use cases (not the focus of our research and therefore likely insufficient data) and other topics like cloud forensics or cost aspects (not enough data).	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	CNA methods which are often pattern based. Furthermore, DevOps principles are taken more and more into consideration as well. Each of these CNA topics (principles, architecture, methods and resulting properties) are influencing each other. However, some topics and their interdependencies are reflected more intensively than other topics. We will discuss this in following Section 3.3. Additionally, we will propose a definition which takes all identified literature into consideration systematically and relate it to already existing and well defined terms (see Section 5).	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > Portability	, 2014). But interop-erability or portability in hybrid or multi-cloud scenarios (PRINC-2) seems to be only considered by single case or survey studies so far (Ben Belgacem et al.,	Ivon Miranda Santos

Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	A lot of studies report about trends to use microservices (Stine, 2015; Balalaie et al., 2015; Jamshidi et al., 2015; Brunner et al., 2015; Balalaie et al., 2016; Hasselbring, 2016), to use DevOps (Familiar, 2015; Balalaie et al., 2016) or to make use of soft-wareization (Taleb et al., 2016; Nikaein, 2015; Krieger et al., 2016; Housfater et al., 2014; Moldovan et al., 2014; Wang et al., 2015) to realize and operate CNAs pragmatically. These trends seem to be widely accepted and preferred. However, microservices, DevOps and softwarization shift more realizing autonomy to small development and operation teams. Especially the microservices approach motivates this intentionally (Stine, 2015). But we found no study that dealt with the consequences. Microservice fol-lowers often cite Conways Law.2	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	However, according to this law the organizational (two pizza 3) structures coming along with microservices and DevOps principles should lead to more hetero-geneous service realizations which only focus singular problems. The microservice intent is to solve these singular problems well and in a massively horizontal scalable and elastic way. Large scale systems are composed of these independently replaceable com-ponents. But according to Conway (Conway, Melvin E., 1968) this should result in increased heterogeneity (which most other ap-proaches would like to restrict)	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	One may even ask if microservices and DevOps are “eating” top down engineering methodologies? It would be interesting to investigate what the long term implications of these autonomous engineering approaches are (METH.1). It seems likely that resulting systems might have the tendency to become hardly maintainable on the long term. Most existing microservice based systems are still very young systems nowadays. So, the long term effects of these engineering methodologies might not be observed so far	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	 DevOps DevOps is a practice that emphasizes the collaboration of software developers and IT operators. It aims to build, test, and release software more rapidly, frequently, and more reliably using automated processes for software delivery.	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	Applied microservice and DevOps engineering methodologies are mainly bottom-up approaches shifting software design autonomy to plenty of small teams. All this result in more heterogeneity of components. This is astonishing, because most engineering methodologies are used to reduce heterogeneity for a better manageability. The long term effects of these autonomous engineering approaches are hardly investigated so far. Even the existing systems are quite young and the long term effects might not be even observed	Ivon Miranda Santos
Kratzke2017-Understanding_cloud-native_applications_after_10_ye	CLOUD MANAGEMENT > DevOps	CNA architectures are more and more often microservice based. Proposed CNA development methodologies are often pattern-based (relying on comprehensive cloud computing pattern catalogs (Fehling et al., 2014; Erl et al., 2015)) and take DevOps principles into consideration. We could even derive a definition proposal for the term “cloud-native application” and we explained its relationship to well established terms in the context of distributed systems. We think our proposal contributes to a more precise understanding of the term “cloud-native”. Especially if this definition proposal is compared with the often heard but vacuous sentence: “Cloud-native applications are intentionally designed for the cloud.”	Ivon Miranda Santos

Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Performance	The QoS is also con-sidered as the key factor in Web service composition. It could be classified as dynamic QoS (eg, performance, response time, reliability, and availability) and static QoS (eg, robustness, accuracy, and security). Each one has a dimension and can be measured using 1 or more QoS met-rics.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Performance	However, QoS may also depend on the cloud service model, taking as example the QoS attributes specific to IaaS, such as memory, CPU capacity, accuracy, VM response, throughput, storage, stable time, and failure rate. Quality-of-service attributes for PaaS services include availability, accessibility, security, and reliability, whereas those specific to SaaS include security, reliability, performance, interoperability, scalability, and availability.27 Quality-of-service attributes are generally expressed in the user request, depending on the demanded cloud service model. The user request is the set of requirements expected by the consumer, to be satisfied using a composite service from the multicloud environ-ment.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	If the use of services from 1 single cloud could satisfy the users request, it will be so much better given that the communication cost between several clouds is high.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Performance	This migration offers more autonomy while selecting services, as it gives the customer the possibility of choosing from the set of candidate offers the most performant one, at any given time. Furthermore, using several clouds is also successful in avoiding downtime and data waste, enhancing the enterprise performance and eschewing vendor lock-in.22 The classification of the multiple cloud environments (MCEs) differs from one article to another.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	This is by minimizing the execution time, the number of combined clouds, the intercloud communication cost, etc.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Composing services from several clouds could increase the communication cost and time.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Performance	6.1.4 Approaches based on service partitioning and clustering The service partitioning is a technique used to simplify the services treatment for later reuse. It is recommended in order to improve the scalability and the performance of the selected services by regrouping similar ones. From the used service-partitioning methods, we could mention the default method of MapReduce parallel programming model.56 In Dou et al53 and Mezni and Sellami,50k-means and FCA, respectively, are used as clustering techniques.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Reducing the number of involved clouds, the number of service providers and also the intercloud communication cost are the main objectives defined by the authors in this approach.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Combining simple services from the cloud environment could be time consuming, costly, and error source.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Then, the best ser-vices are selected according to their availability zones, as well as other cloud constraints (eg, communication costs).	Ivon Miranda Santos

Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Zou et al57 All clouds * Short composition time * Optimal solution may not be found * All possible solutions are generated * High number of clouds Base cloud * Optimal solution in all cases * High computation time * Low number of clouds * Lack of load balancing among cloud servers Smart cloud * Near optimal solution * Time consuming * Better time and quality solutions are generated * Lack of load balancing among cloud servers Zemni et al59 * Privacy-aware fragments * Only semantically close activities are grouped * Automated decomposition process * Activities execution order is not respected Gutierrez-Garcia and Sim26 * Incomplete informations about * High communication and processing cost distributed Cloud participants are considered * Efficient only for functional and cost requirements * Changed consumer needs are treated	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Performance	However, when several clouds are used, the workload becomes difficult to manage by a single manager, such as the cloud combiner in Kurdi et al,49 which is responsible for organizing all the computation and communication flows. To remedy with such problem, Ismail and Cardellini74 made use of so-called decentralized or distributed approaches given their multipros such as flexibility, fault tolerance, and performance, taking as example, the adoption of an agent-based paradigm in Gutierrez-Garcia and Sim.26 The latter exploits the contract net protocol, which is a distributed technique used by the consumers and the providers searching for an agreement. The distributed approaches are efficient in reducing the messages exchanged between the cloud servers75 but suffer from a high processing cost.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	The distributed approaches are efficient in reducing the messages exchanged between the cloud servers75 but suffer from a high processing cost.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Monitoring	9 OTHER ISSUES IN MULTICLOUD ENVIRONMENT In addition to service composition, tremendous other issues in multicloud environment have been addressed by researchers, such as intercloud SLA enactment,80 dynamic service selection,73 crosscloud monitoring,81 service deployment,19 and adaptive service management.82	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Portability	Paraiso et al90 have proposed a new multicloud PaaS infrastructure. They tackle some existing barriers, such as portability, interoperability, heterogeneity, and geo-diversity. The latter appeared due to the raising number of the cloud computing applications and services, and the necessity of using several clouds.	Ivon Miranda Santos

Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Monitoring	9.5 Service monitoring The complexity of cloud infrastructure translates into more effort needed for the monitoring and adaptation of services and resources in such envi-ronment. The greater scalability and larger size of multicloud environments, compared to traditional cloud or Web service hosting infrastructures, involve more complex monitoring and adaptation systems, which have therefore to be more scalable, robust, and fast. The applications existing in the multicloud platforms often require monitoring mechanisms to supply information about real-time QoS. Quality-of-service monitoring is important in the multicloud layers, as it controls the failures if they exist and stops their propagation to other components.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Performance	Quality-of-service monitoring is important in the multicloud layers, as it controls the failures if they exist and stops their propagation to other components.92 Given the dynamic nature of the cross-cloud environment, that the global optimization approaches existing in the literature could not deal with, the selected composite services need to be adaptive with later changes. Therefore, Yang et al81 proposed a selection method, which is able to guar-antee that the generated service from the service composition process has a near-optimal performance. The performance of the selected optimal services could decrease during the execution if the changes occurred in the selection phase are not handled. More precisely, both the selection and the execution phases should be realized in the same time to simply deal with the changes appeared in the crosscloud environment.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Portability	The multicloud environment is becoming more and more popular, due to the variety of the appeared service models and its ability to deal with issues in the traditional cloud environment. Despite that, tremendous problems appeared in this colossal environment and need to be addressed such as security, interoperability, and portability. • Security:	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Portability	20 of 24 LAHMAR AND MEZNI • Portability: It concerns the SaaS that could be selected by moving from one IaaS provider to another, searching for the provider that has appropriate properties. The portability issue is addressed given the diversity of PaaSs and IaaSs.90 • Data centers heterogeneity:	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	• the reduction of the number of involved clouds and, thereafter, the reduction of the execution time and inter-cloud communication cost.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Given that the communication between providers in the cloud environments mentioned above is costly and time consuming, minimizing the number of the used clouds is a basic objective during the composition process.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Monitoring	The proliferation of multicloud environments has raised the need for effective multicloud management tools, to allow enterprises spread their services across multiple clouds.Mul-ticloud tools are expected to accelerate services deployment and scaling and to automate monitoring, governance, and configuration tasks.	Ivon Miranda Santos

Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Monitoring	Cloud4SOA95 is a broker-like multicloud platform that, in addition to the deployment, governance and monitoring tasks, it allows to semantically orchestrate heterogeneous PaaS services. SeaClouds project96 aims to provide an open source framework for the adaptive management of complex applications over multiple heterogeneous Clouds (PaaS platforms). SeaClouds relies on a discovery APIs and a planning policy to determine the best offer for each application module and to orchestrate the multicloud deployment of those modules across the selected cloud platforms. The planned orchestration will be substituted with another one if the application requirements are violated.	Ivon Miranda Santos
Lahmar2018-Multicloud_service_composition_A_survey_of_current_a	CLOUD MANAGEMENT > Cost	Effectively reusing arbitrary granularities of BPaaS fragments has not been solved yet.19 Reusing BPaaS fragments rather than reusing atomic services from the multicloud can not only decrease the composition time, but also improve the reliability of the whole composition process.58 Seen that a multicloud deployment is costly, but at the same time it is more secure than a single one, it will be interesting to equilibrate these 2 criteria.	Ivon Miranda Santos
Lichtenthaler2019-Requirements_for_a_model-driven_cloud-native_	CLOUD MANAGEMENT > Performance	We chose the FaaS paradigm, because it is based on fine-grained components, namely functions, and it can be seen as the lat-est trend in cloud computing [10]. It has to be noted, that the goal of the project was not to improve the existing appli-cation considering performance, cost or maintainability, but to explore the migration process of transforming an existing monolithic application into a CNA. For the monolithic appli-cation, we chose the REST version of the Spring Petclinic sample application.2 Although the application is compara-tively simple, it features typical characteristics of web-based applications and is therefore suited as a close to reality but still comprehensible example.	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	Implementation of ERP systems is supposed to enhance cross-functional operations within an enterprise (Ifinedo, 2006). Various reasons for ERP implementation have been identified such as improving business performance, positioning organizations for growth, reducing working capital and serving customers better (Panorama Consulting Solutions, 2018, p. 11). The implementation of ERP systems in any organization may reduce time, cost to carry out different work processes that consequently enhance the efficiency and effectiveness of the organization (Subramoniam et al.,	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	The adoption of ERP systems in public/private organizations is growing rapidly for the past decade. Different aspects related to improving strategic decision-making, technological, operational, organizational and financial performance have been mentioned that persuade adoption of ERP systems in organizations (Gabryelczyk and Roztocki, 2017; Simone et al., 2018). The organizations that need to improve business performance and sustain competitive advantages may change their strategy to go for ERP systems rather than in house development of information systems. So, ERP systems have a wide acceptability all-around the world among the organizations and the importance of such systems has almost been realized (Wingreen et al.,	Ivon Miranda Santos

Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	640performance and efficiency of the new ERP system because of lack of awareness and improper integration strategies of the new system. They further advocated the need for effective strategy regarding integration and alignment of technologies with the existing practices.	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	The issue/challenge concerning business process reengineering (BPR) is ranked #6 based on findings of this research. BPR may be defined as the rethinking and redesign of business processes to achieve improved organizational performance in terms of quality, cost, speed and service (Hammer and Champy, 1993). For a successful ERP implementation organizational willingness is required to change the business processes to fit in the ERP software along with minimizing the customization.	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	On the other hand, Hong et al. (2016) claimed that the level of customization does not influence performance. Haines (2009) also found long term cost implications because of extensive customizations.	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	Erkan (2009, p. 1516) found “[...][...] firm, which directly implemented ERP has lower values in selected performance indicators than the firm implemented first BPR then ERP.” ERP and BPR are complementary, whereas BPR is almost essential for the successful implementation of ERP in any organization.	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	According to Dillon et al. (2010), mentioned different drawbacks in cloud-based ERP relate to data security, performance and availability. Elmonem et al.	Ivon Miranda Santos
Mahmood2020-Erp_issues_and_challenges_a_research_synthesis	CLOUD MANAGEMENT > Performance	(2005), have identified failure factors for different organizations. These factors are scope creep, lack of change management, lack of communication, lack of performance measurement, BPR integration, poor consultant effectiveness, poor project management, poor knowledge transfer, a high turnover rate of project team members and poor top management support. 6.	Ivon Miranda Santos
Maniah2022-A_systematic_literature_review_Risk_analysis_in_clou	CLOUD MANAGEMENT > Performance	Therefore, many companies that have this large data set are looking to migrate to the cloud. The impetus for migrating to the cloud is to reduce busi-ness operating costs and want to improve the performance of existing application systems (Leff and Rayfield, 2015). Cloud computing is a platform that provides communication services and is in the form of Internet-based computing that can share resources, data and applications that run in a cloud envi-ronme@@nt (T.K and B, 2016).	Ivon Miranda Santos
Maniah2022-A_systematic_literature_review_Risk_analysis_in_clou	CLOUD MANAGEMENT > Cost	The impetus for migrating to the cloud is to reduce busi-ness operating costs and want to improve the performance of existing application systems (Leff and Rayfield, 2015).	Ivon Miranda Santos
Maniah2022-A_systematic_literature_review_Risk_analysis_in_clou	CLOUD MANAGEMENT > Cost	Cloud migration can save costs because cloud service providers already provide infrastructure so that it can reduce the provision of their own infrastructure (Ren et al.,	Ivon Miranda Santos

Maniah2022-A_systematic_literature_review_Risk_analysis_in_clou	CLOUD MANAGEMENT > Performance	Several other reasons for companies migrating to the cloud are: (1) because cloud comput-ing services have scalability, which means that they can meet the needs of information technology resources according to company needs; (2) because the cloud provider has provided settings for both hardware configuration and software updates or server set-tings and others, so that companies as cloud service users are more focused on developing better innovative products; (3) because the cloud provider has a data center that provides fast and efficient computing services, so this will have an effect on high performance in the cloud compared to the data center owned by the company. Based on data from cisco.com, it is estimated that in 2020 cloud data centers will process by 93%, while in 2021 the workload of data centers will increase to 94%.	Ivon Miranda Santos
Maniah2022-A_systematic_literature_review_Risk_analysis_in_clou	CLOUD MANAGEMENT > multi-tenant	Multitenancy security and privacy are important challenges for cloud users, because multitenancy allows multiple users to run their application	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Many cloud vendors offer similar features with varying costs, so an appropriate choice will be the key to guaranteeing comparatively low operational costs for an organization. The motivation for this work is the necessity to select an appropriate cloud storage provider offering for the migration of applications with less cost and high performance. However, the selection of a suitable cloud storage provider is a complex problem that entails various technical and organizational aspects.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	The motivation for this work is the necessity to select an appropriate cloud storage provider offering for the migration of applications with less cost and high performance.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	As the cloud business increases, many organizations are joining the market as cloud storage and service providers. The majority of providers offer similar services with different pricing since data processing performance remains limited. The marginal differences leave cloud users with the mystery of estimating costs.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	The marginal differences leave cloud users with the mystery of estimating costs.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	The marginal differences leave cloud users with the mystery of estimating costs.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Organizations are interested in migrating their legacy local data storage to cloud-based storage to get maximum benefits with less cost, sharing, consistency and scalability.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Their research is conceptual and does not provide implementation descriptions to build a prototype. To account for the overall performance in cloud computing, we need to consider many diverse factors. If the consumer considers migration of a legacy application to the cloud, a DSS can be used to assess the cloud computing services. This system considers the reduction of risk and the relationship between computation performance and cost. A service called Service Measurement Index (SMI) [8] is an example of the service from such a system.	Ivon Miranda Santos

Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	The final rating is from the possible combinations of VM and service. Key performance indicators (KPI) are defined in order to measure and compare cloud services. An example is assurances from KPIs, like availability and stability of service [9]. The SMI model summarizes the most significant attributes for the service quality such as assurances of usability, cost, agility, performance, scalability, security, accountability, and privacy. The metamodeling technique [10] was adopted to develop a generic process model.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	The research in [11] has investigated the major issues in developing an efficient cloud storage service for the personal health record (PHR) domain where the patient’s data are stored on the cloud and doctors can access it from the cloud server for diagnoses. The performance of the cloud storage services is discussed with their strengths and weaknesses. A qualitative and quantitative analysis was performed on the PHR data.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	In this study, a comparison of eleven (11) storage services was provided through a case study. Their methodology suggested design choices and performance factors for the organizations. Their proposed work was tested on various workloads and the implemented tool is released on the public website (http:/	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	A comparison tool “CloudCmp” was introduced by Li et al. [13] to select the cloud storage provider on the basis of performance and cost of services. The parameters used in this research were elastic computing, storage, and network services.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	It was the first complete decision support framework and method precisely for the provision of cloud computing situations. SMI-Cloud [17] provides features like selection of services and categorizing of services based on their previous usage and performance. It has three main modules:	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Monitoring	After this process, a list of ranked services is provided to the user. SMI Calculator operates on different KPIs and helps the ranking system to generate a sorted list of cloud services. SMICloud-Monitoring identifies all those cloud services that can accomplish the customer’s essential QoS requests. In order to identify an appropriate cloud service, it validates user requirements based on attributes such as the number of CPU cores, storage size, scalability, memory, bandwidth, and synchronization	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	In the development of this framework, the main problem faced was calculation of KPIs associated with different cloud service providers. In the next segment, a QoS framework is developed for IAAS service providers based on SMI key performance indexes. Besides IAAS, this framework can work as SAAS and PAAS.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	A learning base with data assembled from cloud providers on their services offers a back end that con-tains the vendor choice and cost-figuring rationale.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Besides these aspects, a decision-making system specific to application migration to the cloud needs to consider more dimensions. Along with calculating costs, other tasks such as performance forecast and countermeasures against security threats are critical. At the end of this section, we summarize the steps involved in application migration and the technologies used in previous frameworks as shown in Table 1.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cloud-Genius [15] Full Cost, CPU, RAM AHP	Ivon Miranda Santos

Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	The decision to shift data to the cloud environment is not a simple task as it involves multiple contradictory aspects. Some of these complex factors include performance, pri-vacy, data security, legal concerns, scalability, service availability, quality of service, and mainly the cost [24].	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Sometimes, a very small change in the specifications can result in very large cost differences. The user has to choose a sensible value that decreases the legacy system's expenses with a lower cost and higher performance. The main task of the suggested DSS is to rank cloud storage providers on the basis of services by considering two aspects: cost and performance. Performance is an array of multiple parameters. The important function is a comparison of providers and rank-ings that works based on the user's selected parameters.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	The proposed DSS has a simple, elegant, and user-friendly interface where one can perform different operations while logged in. Re-ranking services based on cost or performance are shown for different cloud storage providers. Moreover, users can select different parameters in order to see the best provider for their requirements.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Re-ranking services based on cost or performance are shown for different cloud storage providers.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Cloud service vendors' offerings are recognized by OfferID and OfferName. The parser offers configuration and parameters that match the requirements of the user, and finally, a priority list of vendors is generated on the basis of performance and cost. The performance attribute is further elaborated as follows: CPU cores, storage, and free storage.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Parameter values are fetched from the website of the cloud provider, subsequently cost is calculated and shown to users.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	3.2.3. Cloud Service Provider Ranking The ranking module is a very important part of the DSS [30,31] that gives two options to the user (rank by cost or performance) because some organizations have a limited budget and want to stay within their budget. On the other hand, some organizations need better performance, and budget is not a big issue, so they prefer a system with a better configuration. So, according to the user's selected ranking type, the required procedure will be called.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Figure 10 presents a flow chart of the cloud services ranking process. The performance ranking calculation is performed on the right side, while the steps for cost ranking are on the left side. The data store is accessed from both sides of the flowchart, which starts with selecting the type of ranking and ends with a list of ranked cloud storage providers.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Ranking is basically a priority list of cloud providers, and is generated by calculating comparative ranking values for different cloud services based on performance and cost.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Cloud service provider ranking is one of the most important parts of the proposed decision-making system. Ranking is basically a priority list of cloud providers, and is generated by calculating comparative ranking values for different cloud services based on performance and cost. Because multiple attributes are involved, ranking cannot be calculated easily.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Ranking is basically a priority list of cloud providers, and is generated by calculating comparative ranking values for different cloud services based on performance and cost.	Ivon Miranda Santos

Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	One of the important tasks in the AHP technique is assigning weights to attributes, which are called relative importance values (RIV). After a study of previous research and its analysis, nine values for cost and nine values for performance attributes were taken for ranking cloud storage providers. Table 3 shows the RIV for each option.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	The DSS helps users to select the appropriate cloud storage for their application(s). Usually, there are two kinds of users; first focuses on the cost, and can compromise on quality as well as performance; while the second type requires excellent quality and performance at any cost. The proposed DSS is designed and implemented for both types, so ranking is calculated based on both cost and performance. A structured technique AHP is used to calculate ranking w.r.t the costs of different cloud services.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	A structured technique AHP is used to calculate ranking w.r.t the costs of different cloud services.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	The goal is to rank cloud provider services on monthly and yearly cost criteria.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Steps in the Ranked cloud services by cost are shown in Figure 14.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	The DSS helps users to select the appropriate cloud storage for their application(s). Usually, there are two kinds of users; first focuses on the cost, and can compromise on quality as well as performance; while the second type requires excellent quality and performance at any cost. The proposed DSS is designed and implemented for both types, so ranking is calculated based on both cost and performance. A structured technique AHP is used to calculate ranking w.r.t the costs of different cloud services.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	A structured technique AHP is used to calculate ranking w.r.t the costs of different cloud services.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	The goal is to rank cloud provider services on monthly and yearly cost criteria.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Steps in the Ranked cloud services by cost are shown in Figure 14.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Cloud Services Ranking by Cost. In rank by performance, cost will have low priority as compared to performance at-tributes like storage and number of CPU cores. This option is for users who prefer	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cloud Services Ranking by Cost.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Cloud Services Ranking by Cost. In rank by performance, cost will have low priority as compared to performance attributes like storage and number of CPU cores. This option is for users who prefer	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cloud Services Ranking by Cost.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	2021, 11, 11296 19 of 32 high-performance systems, i.e., maximum CPU cores, RAM, storage, etc.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Main Page of the Proposed System. After the login page, control shifts to the ranking page where a user can rank cloud storage providers by selecting either minimum cost or maximum performance. It is a simple, effortless and fast process.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Main Page of the Proposed System. After the login page, control shifts to the ranking page where a user can rank cloud storage providers by selecting either minimum cost or maximum performance. It is a simple, effortless and fast process.	Ivon Miranda Santos

Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Sync.com 0.018 0.056 0.01 0.109 0.079 0.0034 0 0.045 0.3204 JustCloud 0.018 0.056 0.0073 0.147 0.079 0.0023 0.0022 0.045 0.3568 Dropbox 0.018 0.056 0.1081 0.109 0.079 0 0.0034 0.045 0.4185 Tresorit 0.018 0.056 0.0073 0.147 0.079 0 0.0022 0.045 0.3545 LiveDrive 0.018 0.056 0.1081 0.109 0.079 0 0.0022 0.045 0.4173 pCloud 0.0927 0.056 0.0083 0.147 0.079 0 0 0.045 0.428 hubiC 0.018 0.056 0.0152 0.147 0.079 0.0070 0.0034 0.045 0.3706 Jumpshare 0.018 0.056 0.0083 0.147 0.079 0.0023 0.012 0.045 0.3676 iCloudDrive 0.018 0.005 0.0093 0.017 0.007 0.0034 0.012 0.005 0.0767 MediaFire 0.018 0.056 0.0073 0.109 0.079 0.0050 0 0.045 0.3193 OpenDrive 0.018 0.005 0.063 0.147 0.079 0.0034 0 0.005 0.3204 SugarSync 0.018 0.056 0.0073 0.109 0.079 0.0034 0.0034 0.045 0.3211 ADrive 0.018 0.056 0.0248 0.109 0.079 0.0151 0.0050 0.045 0.3519 Mega 0.018 0.056 0.0152 0.109 0.079 0.0151 0.0034 0.045 0.3407 Google Drive 0.1704 0.056 0.1081 0.147 0.079 0.0060 0.012 0.045 0.6235 SafeCopy 0.018 0.056 0.0057 0.147 0.079 0.0034 0.0034 0.005 0.3175	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	5.2. Performance Ranking The CPU core is the major contributor in the ranking of cloud vendor's performance. In this process, the maximum CPU core values of all listed cloud vendors are picked up and calculations are made with the corresponding weights from Table 25. Besides the CPU, all maximum values of parameters are selected for performance ranking as follows: If (Param[X] == Param[W] or Param[X]~Param[W])	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	C81) All the values in the ranking columns are less than 1, called a consistency index (CI), so our final values are validated. The vendor that has the largest value is the best vendor with respect to performance. Figure 19 shows a clear picture of the performance ranking of cloud providers, where Google and IBM have almost the same rank because they provide maximum computing resources. Cloud storage providers are shown on the X-axis and their rankings, calculated by a structured technique, on the Y-axis.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	C41……C81) All the values in the ranking columns are less than 1, called a consistency index (CI), so our final values are validated. The vendor that has the largest value is the best vendor with respect to performance. Figure 19 shows a clear picture of the performance ranking of cloud providers, where Google and IBM have almost the same rank because they provide maximum computing resources. Cloud storage providers are shown on the	Ivon Miranda Santos

Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Cloud Storage Providers’ Performance-based Rankings. Cloud Vendors CPU Cores Sharing Rep OS Mobile Access Free Storage Trial SYN Ranking Sync.com 0.018 0.056 0.01 0.109 0.079 0.0034 0 0.045 0.3204 JustCloud 0.018 0.056 0.0073 0.147 0.079 0.0023 0.0022 0.045 0.3568 Dropbox 0.018 0.056 0.1081 0.109 0.079 0 0.0034 0.045 0.4185 Tresorit 0.018 0.056 0.0073 0.147 0.079 0 0.0022 0.045 0.3545 LiveDrive 0.018 0.056 0.1081 0.109 0.079 0 0.0022 0.045 0.4173 pCloud 0.0927 0.056 0.0083 0.147 0.079 0 0 0.045 0.428 hubiC 0.018 0.056 0.0152 0.147 0.079 0.0070 0.0034 0.045 0.3706 Jumpshare 0.018 0.056 0.0083 0.147 0.079 0.0023 0.012 0.045 0.3676 iCloudDrive 0.018 0.005 0.0093 0.017 0.007 0.0034 0.012 0.005 0.0767 MediaFire 0.018 0.056 0.0073 0.109 0.079 0.0050 0 0.045 0.3193 OpenDrive 0.018 0.005 0.063 0.147 0.079 0.0034 0 0.005 0.3204 SugarSync 0.018 0.056 0.0073 0.109 0.079 0.0034 0.0034 0.045 0.3211 ADrive 0.018 0.056 0.0248 0.109 0.079 0.0151 0.0050 0.045 0.3519 Mega 0.018 0.056 0.0152 0.109 0.079 0.0151 0.0034 0.045 0.3407 Google Drive 0.1704 0.056 0.1081 0.147 0.079 0.0060 0.012 0.045 0.6235 SafeCopy 0.018 0.056 0.0057 0.147 0.079 0.0034 0.0034 0.005 0.3175 OneDrive 0.018 0.056 0.0152 0.109 0.079 0.0034 0.012 0.045 0.3376 AmazonDrive 0.1804 0.005 0.0357 0.109 0.079 0.0034 0.010 0.005 0.4275 Atlanic 0.1504 0.005 0.0083 0.109	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	Figure 19. Ranking By Performance. It is clearly shown from the results that Google Drive and IBM Cloud are leading other cloud storage vendors due to their features.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cost–Benefit Analysis of Cloud Vendors.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cost–Benefit Analysis of Cloud Vendors.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	cloud provider with top benefits and less cost.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cost–Benefit Analysis of Cloud Vendors.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	All frameworks use multiple criteria, but the proposed system considers fewer parameters than SMICloud as well as Cloud Genius and focuses on the critical parameters like performance and cost because of major impact on the ranking.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	Cost–Benefit Analysis of Cloud Vendors.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Performance	After getting user requirements, all potential pairs of parameters are calculated. All frameworks use multiple criteria, but the proposed system considers fewer parameters than SMICloud as well as Cloud Genius and focuses on the critical parameters like performance and cost because of major impact on the ranking. Accuracy and Dynamic database functionality are the other edges of the proposed system.	Ivon Miranda Santos
Mateen2021-A_dynamic_decision_support_system_for_selection_of_c	CLOUD MANAGEMENT > Cost	All frameworks use multiple criteria, but the proposed system considers fewer parameters than SMICloud as well as Cloud Genius and focuses on the critical parameters like performance and cost because of major impact on the ranking.	Ivon Miranda Santos

Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	In this paper, a constrained multicriteria multicloud provider selection mathemat-ical model is proposed. Three metaheuristics algorithms (simulated annealing [SA], genetic algorithm [GA], and particle swarm optimization algorithm [PSO]) were implemented to solve the model, and their performance was studied and compared using a hypothetical case study. For the sake of comparison, Taguchi's robust design method was used to select the algorithms' parameters values, an initial feasible solution was generated using analytic hierarchy process (AHP)—as the most used method to solve the cloud provider selection problem in the literature, all three algorithms used that solution and, in order to avoid AHP limitations, another initial solution was generated randomly and used by the three algorithm in a second set of performance experiments. Results showed that SA, GA, PSO improved the AHP solution by 53.75%, 60.41%, and 60.02%, respectively, SA and PSO are robust because of reaching the same best solution in spite of the initial solution.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	) in terms of service level agreement; a contract between customers and cloud service providers where cloud service providers (the latter) guarantee a satisfactory level of quality of service (QoS) requirements.2 Cloud service provider selection is one of the most significant challenges for cloud customers.3,4 Due to the growing number of cloud providers, cloud market is becoming more and more competitive. Prices and performance levels of the similar offered services are varied, and, consequently, selecting an appropriate provider that can fulfill QoS requirements becomes increasingly difficult. In order to compare providers, customers (decision-makers) need to determine measurable criteria on which service providers will be compared and the method that will be used to rank them based on these criteria.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Cost	Although service cost is considered as a main criterion used to compare different cloud providers,5-13 customers had to consider many other criteria due to the increasing number of providers and the diversity of services offerings.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	Conse-quently, some research papers, from customer perspective, tried to answer the question of what are the most important criteria for cloud provider selection?12,14,15 But, measuring the identified criteria was not a simple task because of lack of standard to various providers.16 Therefore, the Cloud Service Measurement Index Consortium has developed service measurement index (SMI),17 a standard measurement framework which helps decision-makers to compare cloud services from multiple providers. SMI includes seven major characteristics with three or more attributes/key performance indicators (KPIs) in each characteristic.17 Based on SMI, Garg et al7 proposed a decision-making tool to measure various SMI KPIs and to rank cloud services based on these KPIs. Despite the fact that measuring SMI KPIs is an important step in provider selection process which enables researchers to build accurate models, the proposed methods to solve the selection problem in the literature depended on user's preferences.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Monitoring	Building a multicloud solutions poses a number of challenges.22 One of these challenges is that each cloud provider has different application programming interfaces (APIs) that makes the provisioning, deployment, monitoring and manage-ment of multicloud systems is complex.	Ivon Miranda Santos

Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > DevOps	Building a multicloud solutions poses a number of challenges.22 One of these challenges is that each cloud provider has different application programming interfaces (APIs) that makes the provisioning, deployment, monitoring and manage-ment of multicloud systems is complex. However, some steps are already taken to solve this issue. The first step is provided by IaaS stacks, such as CloudStack,24 OpenStack,25 OpenNebula,26 and vCloud27 that provides APIs and dashboards to create and manage IaaS cloud services. IaaS stacks do not assist the development and administration of multicloud systems, so the second step provides abstraction layers that support numerous IaaS providers and IaaS stacks through IaaS/PaaS libraries. These libraries such as jclouds,28 libCloud,29 and Deltacloud30 facilitate the provisioning and deploy-ment of multicloud systems through a single interface, but it does not support automatic provisioning and deployment. Consequently, the third step provides PaaS frameworks such as Cloudify,31 CloudFoundry,32 and Scalr.33 Some of these frameworks base on so-called DevOps tools that automate the provisioning, deployment, and management of multicloud systems.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Cost	) or in the case of changing cloud provider conditions (variable cost, availability, etc.)	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Cost	The proposed multicloud provider selection model is compared to the sin-gle cloud provider selection model based on cost criterion.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	The proposed model minimized the cost by 25% when it is compared to the single cloud selection model. In order to study the performance of the proposed algorithms, five steps were applied: (a) a hypothetical case study was generated; (b) Taguchi's robust design method was used to select the algo-rithms' parameters values; (c) an initial feasible solution was generated using AHP; (d) the algorithms were implemented to improve the quality of AHP solution; and (e) in order to avoid AHP limitations, AHP solution is replaced by an ini-tial feasible random solution and the algorithms' performance were compared. The experimental results showed that the proposed algorithms improved the AHP solution by approximately 58.06%.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	Section 2 deals with multicloud provider selection problem; it starts with problem description and proceeds with the model formulation. The details of solution algorithms are presented in Section 3, while Section 4 tests the performance of the proposed solution algorithms. Finally, conclusion and future work are given in Section 5.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	2.2 Problem formulation According to SMI,17 the presented model was formulated based on five criteria: financial, performance, assurance, usability, and accountability. For each criterion, KPIs that are relevant to IaaS were selected.7 For the convenience of formulation, the indices, parameters and variables are defined as follows:	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	While the integration cost is assumed to be fixed and it is incurred regardless of the number of services that are rented from the selected provider. Objective function (2) is to maximize performance. The performance covers the features and functions of the pro-vided services. The proposed model includes two appropriate performance attributes for IaaS services: (a) service average response time and (b) suitability. Thus, the performance PFij can be defined as: PF = w(r)	Ivon Miranda Santos

Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	Consequently, a general model/mechanism is needed in order to facilitate data collection from different providers. Reference50 presents an idea about a system and method for monitoring the performance of cloud computing envi-ronments. It may be implemented as a stand-alone cloud computing utility or integrated into computing system utility. It is based on configuring the multicloud monitor in a cloud consumer computer to retrieve performance data from the provider multicloud monitors that configured in cloud computing environments. In the presented model, cloud broker acts as an intermediary between the customer/consumer and providers, so instead of configuring multicloud provider monitor in a cloud consumer computer, it can be configured in cloud broker computer to be able to collect the parameters' data that needed to evaluate different criteria.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Monitoring	Reference50 presents an idea about a system and method for monitoring the performance of cloud computing envi-ronments. It may be implemented as a stand-alone cloud computing utility or integrated into computing system utility. It is based on configuring the multicloud monitor in a cloud consumer computer to retrieve performance data from the provider multicloud monitors that configured in cloud computing environments.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Monitoring	In the presented model, cloud broker acts as an intermediary between the customer/consumer and providers, so instead of configuring multicloud provider monitor in a cloud consumer computer, it can be configured in cloud broker computer to be able to collect the parameters' data that needed to evaluate different criteria.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	12 MOHAMED and ABDELSALAM 4 EXPERIMENTS AND PERFORMANCE ANALYSIS A hypothetical case study was generated to test the performance of the three algorithms by scanning the most popu-lar cloud providers and collecting the needed information. In order to create the case study, the following steps were applied:	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Cost	(a) how much the cost is improved when using the multi-cloud selection model instead of single cloud selection model? and (2) what is robust solution algorithm for solving the proposed model?	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Cost	In order to answer the first question, the proposed multicloud provider selection model is compared to the single cloud provider selection model based on cost criterion.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	The proposed model minimized the cost by 25% when it is compared to the single cloud selection model. To answer the second question, the performance analysis of the three algorithms was done through three phases. In the first phase, the parameters value of each algorithm were determined using Taguchi's method.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	4.1 Parameters selection Algorithms' parameters have a significant role on algorithms' performance, so determining the optimal values of it is an important issue. In this paper, Taguchi's robust design method is used to determine the optimal param-eters set.68 Taguchi method was applied on the three algorithms through the following steps. In the first step, parameters for each algorithm were investigated to identify the parameters that influence its performance. For each algorithm, three parameters (controlling factors) were selected and three discrete levels were chosen for each parameters.	Ivon Miranda Santos

Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	The initial feasible solution that was generated after applying AHP method for all customer's requirements is shown in Figure 12. 4.3 Algorithms' performance In this section, algorithms' performance was studied through three stages. In the first stage, the performance of the three algorithms was checked based on the selected parameters values that mentioned in Section 4.1 and AHP candidate solution. In the second stage, the effects of different generations on the quality of solutions for the three algorithms was checked, while in the third stage, because of generating initial solution using AHP method needs a lot of efforts and time, the change in algorithms' performance was checked with an initial random solution. Through the three stages and based on the selected parameters values, each algorithm was run 30 times and the analysis was done based on the average weighted objective values of the best solution that obtained through all runs.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	It noticed that the quality of solutions of the three algorithms was improved when the number of generation increased. Thereafter, based on 100 generation, the algorithms' performance was checked. The progress of the best solution for the algorithms and the box-plot for the algorithms are shown in Figures 16 and 17, respectively.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	In the presented problem, generating an initial solution using AHP method will be very difficult when number of requirements and number of cloud providers increases. So, in order to overcome this problem, the performance of the algorithms with a random initial solution was compared through two phases. In the first phase, for each algorithm the effect of different initial solution was studied, while in the second phase, the performance of the algorithms with an initial random solution was checked. From Figure 18, it is noticed that there was a remarkable change in the behavior of GA; in comparison with best solution progress when using AHP initial solution, GA achieved worst solution with initial random solution.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	Second, a candidate solution was generated using AHP method. Third, algorithms' performance was compared with considering AHP candidate solution as an initial solution. It was found that the algorithms achieve better solutions compared with AHP solution; SA, GA and PSO improved AHP solution by 53.75%, 60.41%, and 60.02%, respectively.	Ivon Miranda Santos
Mohamed2020-A_multicriteria_optimization_model_for_cloud_servic	CLOUD MANAGEMENT > Performance	However, GA has better properties than SA and PSO. Finally, to avoid the limitations of AHP method, algo-rithms' performance was checked based on random initial solution. The results showed that SA and PSO are robust as they reached almost the same best solution in spite of the initial solution.	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	In this manner, blockchain can reduce the trust concern by using various consensus procedures. Moreover, it can reduce the server costs (including the development cost and the operation cost) and mitigate the performance bottlenecks at the central server. In contrast, in many cases, blockchain has some trade-offs. For example, PoW cases such as Bitcoin and Ethereum, the server and energy cost are orders of magnitude higher, while the performance are also several orders of magnitude lower. 2) PERSISTENCY	Ivon Miranda Santos

Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	However, with the increasing complexity for the mining process and the flexible access of new nodes to the network, it results in limited throughput and higher latency. However, with fewer valida-tors and elective consensus protocols, private and consor-tium blockchain can facilitate better performance and energy efficiency [46].	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	PBFT assumes that certain nodes are dishonest or faulty and was designed to be a high-performance consensus algorithm that can rely on a set of trusted nodes in the network [71]. The nodes in PBFT are ordered in a sequential manner with one being the leader and the other nodes acting as backups [72].	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	However, similar to other emerg-ing technologies, blockchain has its limitations and is not feasible for many all types of business model. This section describes the issues and challenges of blockchain technology as the following: performance & scalability in Section V-A, privacy in Section V-B, interoper-ability in Section V-C, energy consumptions in Section V-D, selfish mining in Section V-E and current regulation problems in Section V-F. A. PERFORMANCE & SCALABILITY Cryptocurrency and blockchain-based solutions for differ-ent business models are gaining popularity. However, there is a concern regarding whether it could meet up with the increasing demand coming from different business and government based sectors, especially regarding performance and scalability. Recently, researchers are working to address the scalability issues regarding the number of replicas in the network as well the performance concern, such as throughput (number of transactions per second) and latency (required time for adding a block of transactions in the blockchain) [99]. Increasing the number of replicas can have a detrimental effect on the throughput and latency because the network needs to deal with the increased amount of message exchange and processing.	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	Any mainstream platform needs to process hundreds and thousands of transactions per second. Otherwise, the economy could not keep moving on without massive delays for consumers and businesses, which proves that scalability and performance is an important concern for this emerging technology. Meanwhile, as the capacity of blocks is very small, many small transactions might be delayed since miners prefer those transactions with a high transaction fee.	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	In this section, we have briefly discussed different future scopes for the Blockchain technology including standardization, asset protection, big data, and smart contract. Blockchain performance to lure investors by promising a huge profit. It is compulsory to know whether this tech-nology fits the requirements before adopting it into a busi-ness solution.	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	The first phase will verify the claims of developers regarding their blockchain solutions based on some specific criteria. The testing phase is to determine the performance of the blockchain-based solution. For instance, the owner of an online retail business cares about the performance of the blockchain-based solution.	Ivon Miranda Santos

Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	Ethereum is providing the infrastructure to deploy many smart-contract based solutions, such as car auctions, online trading, and so on. Evaluation refers to performance and code analysis. It has been proven that even a small bug in developing smart contracts could cause a disastrous impact.	Ivon Miranda Santos
Monrat2019-A_Survey_of_Blockchain_From_the_Perspectives_of_Appl	CLOUD MANAGEMENT > Performance	Therefore, it is very important to analyze the attacks on the smart contract. On the other hand, the performance of the smart contract could become an important research topic. As the blockchain technology is acquiring immense attention from public and private sectors, more smart contract-based applications would be put into use.	Ivon Miranda Santos
Naik2021-Performance_evaluation_of_distributed_systems_in_multi	CLOUD MANAGEMENT > Performance	Docker container is an OS-level virtualization and it requires fewer resources than a virtual machine, thus, it resolves the speed and performance issues of virtualization for software developers [3]. Docker Swarm is a container-based clustering tool that supports the design of multi-cloud distributed systems in those clouds which are supported by Docker [4].	Ivon Miranda Santos
Naik2021-Performance_evaluation_of_distributed_systems_in_multi	CLOUD MANAGEMENT > Performance	Sy st em Us in g Docker Swarm This section presents the performance evaluation of the Docker Swarm-based distributed system based on some com-mon attributes of distributed systems and it also compares with other container-based distributed systems. A. High Availability and Fault Tolerance Both high availability and fault tolerance are the inbuilt attributes of Docker Swarm that allows a distributed system to benignly manage the failover of the Swarm leader and continues the normal working of the cluster.	Ivon Miranda Santos
Naik2021-Performance_evaluation_of_distributed_systems_in_multi	CLOUD MANAGEMENT > Performance	This experiment demonstrates that the Docker Swarm-based distributed system provides automatic scalability, load balancing and maintainability of services. C. Scalability of Large Clusters The several recent research studies [16], [17], [18], [19] have been conducted to test and compare the scalability performance of Docker Swarm with other Containers, where Docker Swarm and Google Kubernetes were tested and com-pared based on the large cluster size. Initially, the scalability performance test was carried out by both organizations Docker [16] and Google Kubernetes [17] for the large cluster size. Subsequently, a Docker-sponsored study for the comparison of the scalability performance of Docker Swarm and Google Kubernetes was performed by an independent technology consultant Jeff Nickoloff [18], [19]. He designed a Cloud Container Cluster Common Benchmark framework (available on GitHub [20]) to test the performance of both container platforms while running 30,000 containers across 1,000 node in a cluster. This automated test framework mainly compared the scal-ability performance of Docker Swarm and Kubernetes based on two main criteria: (1) container startup time and (2) system responsiveness under the load as the cluster is built.	Ivon Miranda Santos
Naik2021-Performance_evaluation_of_distributed_systems_in_multi	CLOUD MANAGEMENT > Performance	Co n c l u s io n This paper presented the performance evaluation of a distributed system using Docker Swarm. It illustrated the design and simulated development of the distributed system in multiple clouds using Docker Swarm.	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Limited studies exist to analyse and highlight the complexity of vendor lock-in problem in the cloud environment. Consequently, most customers are unaware of proprietary standards which inhibit interoperability and portability of applications when taking services from vendors. This paper provides a critical analysis of the vendor lock-in problem, from a business perspective.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	The analysis of our survey of 114 participants shows that, as computing resources migrate from on-premise to the cloud, the vendor lock-in problem is exacerbated. Furthermore, the findings exemplify the importance of interoperability, portability and standards in cloud computing. A number of strategies are proposed on how to avoid and mitigate lock-in risks when migrating to cloud computing.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	The European Network and Information Security Agency (ENISA) and European Commission (EC) have recognized the vendor lock-in problem as a one of the greatest obstacles to enterprise cloud adoption [5]. The reviews of existing literature [6–12] have shown that previous studies have focused more on interoper-ability and portability issues of cloud computing when lock-in is discussed.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	These issues result in difficulties in integration between services obtained from different cloud providers as well as between cloud resources and internal legacy systems [16]. Conse-quently, this renders the interoperability and portability of data and application services difficult. The emergent difficulty is a direct result of the current differences between individual cloud vendors offerings based on non-compatible underlying technologies and proprietary standards.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Many cloud vendors provide services based on custom-built pol-icies, infrastructure, platforms, and APIs that make the overall cloud landscape heterogeneous. Such vari-ations cause interoperability, portability, and integra-tion very challenging. Following the principle that compatible interfaces are important in a cloud environment, two implementations of the same cloud service may store and process data very differently.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	So, while customers might be able to access and use the services from a variety of clients, the ability to move seamlessly from one vendor to another may be difficult because of other dependencies such as different data formats. Clearly, this problem has an impact on interoperability and data portability between clouds. At the core of all these problems, we can identify con-cerns about consumers' demand to migrate data to and from different clouds (data portability), and interoper-ability between clouds. Research has already addressed movability and migration on a functional level [18, 19].	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	The two main reasons are the lack of world-wide adopted standards or interfaces to leverage the dynamic landscape of cloud related offers [14], and absence of standards for defining parameters for cloud applications and their management. Without an appropriate stan-dardized format, ensuring interoperability, portability, compliance, trust, and security is difficult [12]. Standards continue to rapidly evolve in step with technology.	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	In other words a partially adopted standard would represent a poor solution. Es-sentially, this explicit lack of standards to support port-ability and interoperability among cloud providers stifles the market competition and locks customers to a single cloud provider [21]. To expatiate further, potential diffi-culties (by primarily technological means) in achieving interoperability and portability lead to lock-in – result-ing in customer dependency on the services of a single cloud computing provider [22]. From a legal stance, the dependency can be aggravated by the abusive conduct of a cloud computing provider within the meaning of Art-icle 102 TFEU (Treaty on the Functioning of the Euro-pean Union) [18], where other providers are excluded from competing from the customers of the initial cloud provider. In such situations, limitations to interoperabil-ity and portability could be seen as an abuse by a domin-ant provider using this practice as a technical means to stifle (i.e. monopolize) competition. Such practices dis-tort competition and harm consumers by depriving them of better prices, greater choices and innovation.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Hence, the competition law has the role of ensuring competition is maintained and enforced in the market by regulating anti-competitive conduct by cloud providers. To this end, it can be concluded that cloud interoperability (and data portability) constraints are potential results of anti-competitive environment created by offering services with proprietary standards. Vendor lock-in	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Actually, the lock-in issue arises when a company, for instance, decides to change cloud providers (or perhaps integrate services from different providers), but is unable to move applications or data across different cloud services because the semantics of resources and services of cloud providers do not match with each other. This heterogeneity of cloud semantics [25] and cloud Application Program Interfaces (APIs) creates technical incompatibility which in turn leads to interoperability and portability challenges [26]. This makes interoperation, collaboration, portability and manageability of data and services a very complex and elusive task. For these reasons, it becomes important	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	from the view point of the business to retain the flexibil-ity to change providers according to business concerns or even keep in-house some of the components that are less mission-critical due to security related risks. Inter-operability and portability among cloud providers can avoid the problem of vendor lock-in. It is the way to-ward a more competitive market for cloud providers and customers.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Lock-in affects cloud migration Interoperability and portability are essential qualities that affect the cloud under different perspectives [7, 13], due to the risk of vendor lock-in. While many studies cite vendor lock-in as a major barrier to cloud comput-ing adoption [3, 27–32], yet due to its complexity, a lack of clarity still pervades.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	lock-in concerns. Various standardisation solutions from different industry bodies have been developed for increasing interoperability and portability within diverse cloud computing services [32, 34]. However, initiatives by multiple standard bodies, researchers, and consortiums could indirectly lead to the possibil-ity of multiple standards emerging with possible lack of consensus, thereby deteriorating the lock-in prob-lem even further.	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Presently, for many companies, there is a large amount of sensitive data and IT assets in-house which can deter them to migrate to the cloud due to risks of vendor lock-in, security and privacy issues. For these reasons, it becomes not only critical to consider security and privacy concerns but also related issues such as integration, portability, and interoperability between the software on-premise and in the cloud [35], should be taking into account. Therefore, organisations must be aware of appropriate standards and protocols used by cloud providers to support data/ application movability. Moreover, the ease of moving data across (i.e. portability) cloud providers’ platform mandates data to be in a compatible format [34], and in-cludes the need to securely delete the old storage [36]. In other words, the ability to move data/application about is of crucial importance, as much as the effort in-volved in actually moving – inability to achieve this por-tends large as a management issue for cloud computing.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Seven themes emerged in relation to partici-pants’ perception of vendor lock-in problem and how this affects their migration and adoption decisions. The themes were; (1) standards, (2) interoperability in the cloud environment, (3) the need for portability, (4) inte-gration challenges, (5) contract exit strategy, (6) data ownership (7) security and privacy issues. The analysis of the responses across the seven themes showed the participants’ priority of the themes. As a result, data portability and interoperability concerns were the most discussed theme in relation to vendor lock-in. However, participants were less interested to divulge about the security and contract exit strategies, including data own-ership and privacy risks.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	10). Overall, the results indicate that these challenges closely relate to interoperability and data portability issues prevalent in the cloud environment. Moreover further results show that a significant majority (76.6 %) of participants were unsure of relevant (existing or emerging) standards to support interoperability across clouds and portability of data from one cloud provider to another. To confer from Fig. 10, the main challenges associated with cloud lock-in are integration and incompatibility is-sues, followed by data portability. However, as shown in Fig.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	In regard to the interpretation of this finding, our study suggests that the vetting process for selecting vendors is a critical aspect for effective cloud migration with minimized risk of lock-in. Moreover, such finding exemplify the need for organisations to look beyond the vendor selection phase, and focus on constantly monitoring any develop-ment or changes in the cloud that may impact data se-curity or hinder interoperability and portability – thus facilitating a lock-in situation. However, the findings (in Fig.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Monitoring	Moreover, such finding exemplify the need for organisations to look beyond the vendor selection phase, and focus on constantly monitoring any develop-ment or changes in the cloud that may impact data se-curity or hinder interoperability and portability – thus facilitating a lock-in situation.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Standard initiatives Cloud-specific standards are regu-larly proposed as a way to mitigate vendor lock-in and achieve portability and interoperability [50]. It is expressed in [51] that many providers are concerned with customer churn rate that may come with stand-ardisation.	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Clear exam-ples of such cloud-specific standards are OASIS CAMP [54] for PaaS and TOSCA [55] for IaaS. Both specifica-tions aim at enhancing the portability and interoperability of applications across different clouds. We review the two OASIS cloud-specific standards (TOSCA and CAMP) and their potential for dealing with the lock-in problem. TOSCA The Topology and Orchestration Specification for Cloud Applications (TOSCA) [55], is an emerging standard that enhances service and application portabil-ity in a vendor-neutral ecosystem. TOSCA specification describes a meta-model for defining IT services.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	Managing cloud services requires extensive, mostly manual effort by the customers. Further, important cloud properties (such as self-service and rapid elasti-city) can only be realised if service management is auto-mated. In this aspect, TOSCA allows application developers and operators (DevOps) to model manage-ment best practices and reoccurring tasks explicitly into so-called plans (i.e. Workflows). TOSCA plans use exist-ing workflow languages such as Business Process Model and Notation (BPMN) [57, 58] or the Business Process Execution Language (BPEL) [59]. To increase portability, TOSCA allows service creators to gather into plans those activities necessary to deploy, manage, and termin-ate the described cloud service.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	TOSCA plans use exist-ing workflow languages such as Business Process Model and Notation (BPMN) [57, 58] or the Business Process Execution Language (BPEL) [59]. To increase portability, TOSCA allows service creators to gather into plans those activities necessary to deploy, manage, and termin-ate the described cloud service. TOSCA also enables a cloud service creator to provide the same plan or imple-mentation artefact in different languages (e.g. a plan can include the same functionality twice – in BPEL and BPMN).	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Considering that lock-in is un-desirable, and cannot be eradicated, then how can busi-nesses mitigate its associated risks when migrating to the cloud? From a portability perspective, it becomes critical that organisations' data is sharable between pro-viders, since without the ability to port data or applica-tion, it would become simply impossible to switch cloud service providers at all [60, 61]. Cloud portability is a sa-lient consideration to enable organisations migrate a cloud-deployed asset to a different provider and it is a direct benefit of overcoming vendor lock-in [62]. Gener-ally, reconfiguration of systems and applications to achieve interoperability is time/resource consuming and may require a considerable amount of expertise, which could be challenging for some organisations. Therefore, from a business perspective, portability should be seen as a key aspect to consider when selecting cloud pro-viders as it can both help mitigate lock-in risks, and de-liver business benefits. This means allowing applications, systems and data components to continue to work cor-rectly when moved between cloud providers'	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Suppose these organisations use the SaaS CRM and over time, perhaps, the terms of use or the price of the cloud-based CRM service become less attractive, compared to other SaaS providers or with the use of an in-house CRM solution. If the organisation decides to change providers for whatever reason, data portability aspects must be considered. For SaaS cloud services, data formats and contents are handled by the service provider thereby making data portability a major consideration. The issue of importance in a SaaS-level migration is the compatibility of the functional interface presented to end-users and any API made available to other customer applications.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	If the APIs are not interoperable, any customer application or data using the APIs will need to be changed as part of the migration process. Data portability is usually of most concern in a SaaS, since in these services, the content, data schemas and storage format are under the control of the cloud service provider. The customer will need to understand how the data can be imported into the service and exported from the service.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	This also applies to any applica-tion or system belonging to the cloud service customers that use APIs offered by the SaaS application. Data synchronization is another concern, encountered in cloud interoperability and not in data portability [63]. To further substantiate this argument, we elucidate on the need for a portable hybrid environment by highlight-ing two main categories of portability scenarios encoun-tered in current cloud service market: 1) porting legacy applications or data; and 2) porting cloud native applica-tions or data.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	However, in both scenarios, the main problem is that there must be a capability to retrieve customer data from the source cloud service and also a capability to import customer data into the target cloud service. Thus, data portability is based on import and export functionality from cloud data services for data structures. This is commonly done through the existence of some API (or web interface) as-sociated with the cloud service – it may be a generic API or a specific API, unique to the cloud service. In light of such challenges, [64] claims that ensuring data portability is a major challenge for enterprises due Opara-Martins et al.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Therefore, in response to the question of data movability, it is im-portant to note that the API used for the source service may not be the same as the API used for the target ser-vice and that different tooling may be required in each case. The main aspects of data portability are the syntax and semantics of the transferred data. The syntax of the data should ideally be the same for the source service and the target service.	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	, the source may use JSON syntax, but the tar-get may use XML), it may be possible to map the data using commonly available tools. If the semantics of the transferred data does not match between the source and target services, then data portability is likely to be more difficult or even impossible. However, this might be achieved by the source service supplying the data in exactly the format that is accepted by the target service. Therefore, on a long term, achieving data portability will depend on the standardization of import and export functionality of data and its adoption by the providers. The aim is to minimize the human efforts in re-design and re-deployment of application and data when moving from one cloud to another.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	lock-in risks or low proprietary lock-in risk. Organisa-tions must be cautious of potential areas of lock-in traps and take adequate measures to mitigate their exposure; e.g. choice of operating environment, programming models, API stack, data portability etc. Further, busi-nesses should take heed of other legal, regulatory, or reputational risks that may exist.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Thus an emerging research agenda arises as to investigate: 1) ways to come up with multijurisdictional laws to support interoperability and portability of data across cloud pro-viders platform, along with effective data privacy and security policies; and 2) novel ideas of avoiding vendor dependency on the infrastructure layer, platform, and through to the application layer as lock- cannot be com-pletely eliminated, but can be mitigated. However, these require, not just tools and processes, but also strategic approaches – attitude, confidence, comfort, and en-hanced knowledge of how complex distributed cloud-based services work.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	Potential of DevOps tools for avoiding vendor lock-in Issues with cloud lock-in surpass those of technical in-compatibility and data integration. Mitigating cloud lock-in risks cannot be guaranteed with a selection of in-dividual open (technology-centric) solutions or vendors. Instead, the management and operation of cloud services to avoid lock-in should be addressed at a standardised technology-independent manner. In this respect, we present a concise discussion on the potential of DevOps [65] and of tools (such as Chef, Juju and Puppet) that support interoperable management.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	DevOps is an emerging paradigm [66] to eliminate the split and barrier between developers and operations personnel.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	The philosophy behind DevOps is to bring agile methodologies into IT infrastructure and ser-vice management [65]	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	DevOps approaches can be combined with cloud computing to enable on-demand provisioning of underlying resources (such as virtual servers, database, application middleware and storage) in a self-service manner	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	These resources can be configured and managed using DevOps tools and ar-tifacts. As a result, end-to-end deployment automation is effectively enabled by using the DevOps approaches in cloud computing environments [69]	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	Today, several applications provisioning solution exists that enable developers and administrators to de-claratively specify deployment artefacts and dependen-cies to allow for repeatable and managed resource provisioning [56]. Below, we review some DevOps tools among the currently available ones that may help enter-prises simplify their application release circle.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Monitoring	Juju Juju is a cloud configuration, deployment and mon-itoring environment that deploy services across multiple cloud or physical servers and orchestrate those services [74]. Activities within a service deployed by Juju are or-chestrated by a Juju charm, which is a deployable service or application component [75].	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	In summary, as applications evolve to function in the cloud, organizations must reconsider how they develop, deploy, and manage them. While cloud computing is heavily used to provide the underlying resource, our re-view shows that DevOps tools and artefacts can be used to configure and manage these resources.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > DevOps	As a result, end-to-end deployment automation is efficiently enabled by employing DevOps approaches in cloud environ-ments. But, cloud providers such as Amazon and cloud frameworks such as OpenStack provide cost-effective and fast ways to deploy and run applications. However, there is a large variety of deployment tools and tech-niques available [76].	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	In fact, the study has shown that, while organisations are eager to adopt cloud computing due to its benefits, there is equally an urgent need for avoiding vendor lock-in risks. Moreover, the re-sults of our study have highlighted customers' lack of awareness of proprietary standards which prohibit inter-operability and portability when procuring services from vendors. The complexity and cost of switching providers is often under-appreciated until implementation.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	Our findings offer cloud computing con-sumers, service providers, and industry practitioners a better understanding of the risk of lock-in embedded in the complex, technologically interdependent and hetero-geneous cloud systems. In this respect, our research points to the need for more sophisticated policy approaches that take a system-wide perspective to alleviate the current vendor lock-in problem which affects inter-operability and portability. Furthermore, our findings show that within many organisations in the study, a lack of clarity on the problem space of vendor lock-in still pervades.	Ivon Miranda Santos
Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	This lack of knowledge poses a significant bar-rier to obscure the potential effect the vendor lock-in problem could have on enterprise applications migrated to and operating in cloud platforms. Hence, to be pro-tected against such risks when migrating to the cloud environment, companies require standards, portability, and interoperability to be supported by providers. How-ever, this is currently difficult to achieve as explored in this paper.	Ivon Miranda Santos

Opara-Martins2016-Critical_analysis_of_vendor_lock-in_and_its_i	CLOUD MANAGEMENT > Portability	(i) cre-ate awareness of the complexities and dependencies that exist among cloud-based solutions; (ii) assess providers' technology implementation such as API and contract for potential areas of lock-in; (iii) select vendors, platforms, or services that support more standardised formats and protocols based on standard data structures; and (iv) ensure there is sufficient portability. In our future work, we will explore interoperability and portability con-straints which affect enterprise application migration and adoption of SaaS clouds. Additional files	Ivon Miranda Santos
Perrons2013-Cloud_computing_in_the_upstream_oil_and_gas_industr	CLOUD MANAGEMENT > Cost	For example, insurance companies are seeking more flexible cost structures, and the cloud allows firms to shift costs to an operating expense rather than a capital outlay, thereby giving them a highly flexible ''pay as you go'' resource.	Ivon Miranda Santos
Perrons2013-Cloud_computing_in_the_upstream_oil_and_gas_industr	CLOUD MANAGEMENT > Cost	But hybrid systems do come at a cost: they do not offer the near-infinite scalability, extremely high ''outsourceability,'' and cost efficiency that the totally public cloud does.	Ivon Miranda Santos
Perrons2013-Cloud_computing_in_the_upstream_oil_and_gas_industr	CLOUD MANAGEMENT > Portability	, 2009). Moving these functions to the cloud involved transferring sensitive data that are legally protected by the United States Health Insurance Portability and Accountability Act. To ensure that the data is secure at all times, TC3 encrypts the data before placing it in the cloud.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	c 2014 SCPE PORTABILITY IN CLOUDS: APPROACHES AND RESEARCH OPPORTUNITIES	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Abstract. The migration towards Cloud environments is still hindered by several barriers. One of them is the low portability of the applications that are consuming Cloud services. This paper intends to provide an image of the state of the art in this particular topic and to identify the potential paths to follow in order to solve the problem. The main concerns are the portability reasons, scenarios, taxonomies, measurements, requirements, and current technical solutions (through open standards, open application programming interfaces, semantics and model-driven engineering). A research agenda is following the current analysis of the state of the art.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Key words: Cloud computing, Portability, Multi-Clouds AMS subject classifications.	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Introduction. Portability is know as being the ability to use software components on multiple hardware or software environments. In particular, the portability in Cloud or portability between Clouds, or shortly Cloud portability in this paper, is expected to ensure that an application, service or data works in the same manner regardless of the consumed Cloud services and has a common method of interaction with these services. The Cloud portability problem has been recognized already from the beginning of Cloud computing (e.g. in [1]). More recently, the IDC report from 2012 [2] points that portability is one of the most important obstacle hindering increasing Cloud adoption. However, the work on Cloud portability is gaining now momentum, as demonstrated by recent standardization initiatives for Cloud portability (e.g. TOSCA [3]). Cloud providers are nowadays also concerned about portability (e.g. IBM and HP are supporting to OpenStack initiative, while Amazon and Microsoft are investing in virtual machine portability [4]). Several research papers, proof-of-concept implementations, or commercial products, have dealt until recently with the portability problem. However, Gonidis et al [5] observed that there is no extensive survey to describe in detail the problem space of Cloud application portability and how current solutions are mapped to that space. They underlined that such a study is essential in order to understand the root causes of the problem and the desirable characteristics of	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	A first step has been done by them in [5], and we intend to continue in this paper their survey, as well as our earlier attempts in the same direction. More precisely, Petcu et al [6] proposed also a classification of the portability levels, while earlier Petcu [7] has underline the difference between interoperability and portability. The novelty of this paper is two-fold.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Second, it identifies the missing pieces needed to solve the problem. Therefore, the paper results can be useful to researchers in Cloud computing who need an identification of relevant studies, as well as to practitioners interested in understanding the available methods, techniques and tools and their maturity level in supporting portability in Cloud environments. The structure of the paper is as follows. Section 2 is discussing the need for portability. Section 3 is defining the problem and the requirements.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	2. The need for Cloud portability. Cloud computing is currently recognized as a computing model based on ubiquitous network access to a shared and virtualized pool of computing capabilities (like processing, network, storage, message queuing) that can be rapidly provisioned [8].	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	The current software stacks of Cloud services are heterogeneous and the provided features that are often incompatible between different service providers. This diversity is an obstacle with respect to demands such as promoting portability and preventing vendor lock-in. ∗ West University of Timi¸soara, and Institute e-Austria Timi¸soara, Romania, (petcu@info.uvt.ro).	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	252 D. Petcu and A. Vasilakos The portability is requested by reasons varying from optimal selection regarding utilization, costs or profits, to technology changes, as well as legal issues. We discuss them shortly in what follows.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Economical reasons. Cloud portability is needed from an economical point of view for at least two reasons: 1.	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Market: development of a Cloud eco-system and market. Optimizing the operational cost is often the reason that is behind the interest of the Cloud service customer in Cloud portability. Rewriting the services and applications is usually required to comply with the change of the Cloud service.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Cost	Optimizing the operational cost is often the reason that is behind the interest of the Cloud service customer in Cloud portability.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	The porting process of services is usually triggered by the operational changes. If the Cloud portability is ensured, third parties are able to be developed, acting as intermediaries between multiple customer and multiple Cloud providers, enabling deployments according the customer requirements to the appropriate Cloud and adaptation to the current status and consumption conditions of the multiple Cloud services. The porting process of data, applications or services is usually triggered at customer request or automated by the third party.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Technical reasons. Cloud portability is needed from a technical point of view for at least two reasons: 1.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Continuity: to ensure continuity in application and service functionality. The portability between Private and Public Clouds is essential in realizing the vision of the Hybrid Cloud that handles the peaks in service and resource requests addressed to a Private Cloud using external resources of a Public Cloud. The porting process is usually triggered on demand basis.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Legal reasons. Cloud portability is needed from a legal point of view for at least two reasons: 1.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Changes in a country legislation or changes in the customer location can trigger the need to port the software assets from a Cloud environment to another. The Cloud non-portability of application, data, applications or services relaying upon a certain Cloud provider faulty service can produce a cascade effect on the activities of its service dependent customers. In this context the portability is necessary to reach a good recovery time objective. Moreover, public institutions using auctions for acquiring services are reluctant to be engaged in contracts which are not offering proper services or guarantees in case of incidents.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Problem definition. As stated earlier, portability is the ability to move software assets among different runtime platforms without having to rewrite them partly or fully (enhancement of the definition given by Lenhard and Wirtz [9]). If the Cloud portability is achieved, data, application or service components can be moved and reused regardless of the operating system, storage or application programming interface (API). In what follows we discuss some scenarios and attempts for classifications of various cases as well as the basic terminology.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	3.1. Portability scenarios. The real world scenarios are highlighting two main categories of portability scenarios encountered in current Cloud service market (mentioned first by Ranabahu and Sheth [10]):Portability in Clouds: Approaches and Research Opportunities 253 Fig.	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	However a third party running services on multiple Clouds and offering unique entry points to various service customers is interested to ensure that the porting process is reversible, fast and semi-automated. The most challenging scenario for portability is that in which the Cloud applications are distributed across several administrative domains of different providers simultaneously, and, moreover, at least data (if not even application and service components) are ported from one Cloud environment to another. A less discussed scenario is the reverse portability from the Cloud environment towards the own premises resources (even partially, lets say only the data part). This is critical when the service provider changes the API or imposes API restrictions or depreciate a service, so that all depenedent services fall (domino effect).	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	2. Functional portability or application portability refers to the Cloud service agnostic definition of appli-cation functionality. 3. Service portability or platform portability is the ability to add, reconfigure and remove Cloud resources on the fly, independent on the Cloud provider. Data portability is achieved when application data can be retrieved from one provider and imported into an application hosted by another provider. To reach this type of portability, platform-independent data representation is needed to be done in a short-term, as well as specific target representations and code for the application's data access layer. On long term, data portability depends on standardization of import and export functionality of data and its adoption by the providers that are the market key-players. Functional or application portability requires the definition of the application's functionality details in a vendor-agnostic manner. A particular issue is the application portability between development and operational environments. Cloud computing is bringing development and operations closer together, the two being inte-grated as DevOps.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > DevOps	Functional or application portability requires the definition of the application's functionality details in a vendor-agnostic manner. A particular issue is the application portability between development and operational environments. Cloud computing is bringing development and operations closer together, the two being inte-grated as DevOps. This requires that the same environment is used for development and operation, and the application is portable between development and operation environments. If different environments are used the application portability can become an important issue.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	This requires that the same environment is used for development and operation, and the application is portable between development and operation environments. If different environments are used the application portability can become an important issue. Service or platform portability is based on Cloud-agnostic control APIs allowing Cloud resources to be added, configured, or removed in real time by humans or programmatic, based on factors like current outages. In the case of service or platform portability there are two scenarios: 1. Service component portability: service or platform components are ported between Clouds, without involving the applications and data on top of them; 2. Machine image portability: bundles containing applications and data and their services or platforms are ported between Clouds. The portability expectations are different for each service delivery model: IaaS:	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	A IaaS native application assuming the control over operating system requires extensive changes when moving into a PaaS environment. Note that the split according vertical and horizontal views is not yet commonly accepted, as the same terms are reused in other contexts in solving portability issues. In particular, Jonnalagedda et al [13] distinguish two major cases:	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	1. Vertical portability as the capability of an application intended for a Private Cloud to be portable to a Public Cloud. The restriction is that the application runs on the same technology stack on both platforms.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	2. Horizontal portability as the ability to port an application from one technology stack to another, stayingPortability in Clouds: Approaches and Research Opportunities 255 at the same level of abstraction but changing the technology provider.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	3.3. Measuring the portability. The porting process is build from various tasks:	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	3. re-configure services; 4. transfer data, applications, services or machine images between Clouds. If these tasks are straightforward or automated, the degree of portability is high. If these tasks are requiring timely human effort, the degree of portability is low. In particular, according to the service delivery model the portability is evaluated based on: IaaS portability of the virtual machines (VMs) and of the configurations across different run-time environments. PaaS openness degree of the source programming languages for application development, openness degree of the data formats, coupling degree of the coupled services, platform-dependence degree of the abstraction layers for messaging and queuing services.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	SaaS openness of the source code, openness of standard data formats, integration technologies and application server or operating system. Lenhard and Wirtz [9] have define software quality metrics in order to quantify the degree of portability of an executable service-oriented process. They have map existing metrics for program portability to service and process-oriented programs and they have combined the metrics with empirical data on language support in various runtimes. Only program elements that are supported by a majority, or all, runtimes are considered by Lenhard and Wirtz to be portable. Note that their measure of the portability of executable process definitions take only the runtimes for the analyzed process into account. If all runtimes available support all of the language elements available in the same manner with respect to semantics, then any compilable application of service is supposed be portable to any runtime. In such context, Lenhard and Wirtz stated that the first step towards measuring the portability of a process is to calculate the degree of portability for each language element and its configuration: this degree can be identified by the number of runtimes that support a language element. This measurement procedure can be currently applied only partially to portability in PaaS and IaaS environments. 3.4. Cloud portability requirements. Figure 3.2 enumerates several requirements in order to ensure the portability between Clouds. The	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	3.5. Cloud portability relationship with Cloud interoperability. The interoperability and porta-bility are often used in conjunction when Cloud adoption barriers are mentioned. They are often seen as one and the same concept due to the fact that many of their technical solutions are the same.	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Moreover, the interoperability degree between Clouds is usually measured at runtime. Data synchronization is for example a concern that is encountered in Cloud interoperability and is not a concern for Cloud portability. A recent comprehensive study on Cloud interoperability was provided by Toosi et al [14]. On opposite site, Cloud portability is a concern that appears mainly at design and deployment stages of an application, data or service. The aim is to minimize the human efforts in re-design and re-deployment of application, data and services when moving from one Cloud to another.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Types of solutions. Gonidis et al [5] and Silva et al [15] suggested that there are three types of solutions for the Cloud portability problem: 1. adoption of existing or emerging standards and protocols (like TOSCA, CDMI, OCCI, OVF); 2. usage of intermediary layers (like jClouds or mOSAIC); 3. adoption of semantics and model-based solutions.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Open standards. The classical way to bust the portability is the adoption of standards and open source. Surely they are important mostly for the Cloud providers and service developers, and not in a similar degree, for the consumers. Tables 4.1 and 4.2 are pointing towards the most important standards for Cloud portability, and, respec-tively, Cloud standard initiatives. Loutas et al [16] have done a complex survey of the efforts by the industry, standardization groups, and research community, at the level of 2011.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	These operations can be external services or their implementation can be included in the service template (as an implementation artifact). The portability is dependent on the workflow language and engines that are used. An on-going initiative is the Guide for Cloud Portability and Interoperability Profiles (CPIP): the IEEE project P2301 (from Table 4.2), is developing a portability standard. It proposes to group different interfaces, operation conventions or tile formats of the Cloud elements into logical profiles.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Open-source libraries and deployable services. The use of abstraction layers and adapters has been widely adopted by libraries, tools and platforms aiming to enable Cloud portability. An abstraction layer hides the differences between providers and exposes a uniform semantics and syntax. According Hill and Humphrey [18], the APIs that are allowing Cloud portability are classified in three categories: 1. multiple independent implementations, like Eucalyptus versus EC2, AppEngine versus AppScale; 2. runnable on multiple Clouds, but not through independent implementations, like the several implemen-tations of MapReduce; 3. separators of the application into application-logic layer and Cloud layer (Table 4.3).	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Cost	(1) SAF, refer-ring to capacity, quality of service, and cost; (2) CAMP, Cloud ap-plication management for platforms, defining interfaces for self-service provisioning, monitoring, and control of Cloud platforms; (3) TOSCA OW2 www.ow2.org/view/ Cloud/	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Commercial solutions like RightScale or Kaavo are able to deploy applications in various Clouds, but not yet migrate the running ones. In what concerns storage and data portability, three different approaches were proposed: common names-pace, service registry and uniform APIs. The most significant initiatives are enumerated bellow in chronological order.	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	262 D. Petcu and A. Vasilakos Portability at PaaS layer has attract a lot of attention in the last time period. Some significant proposals in chronological order are the followings.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Cuhna et al [23] proposed a definition of a architecture allowing developers to create and expose services through a particular service delivery platform. It enables the portability of service between PaaS through a common API. Zeginis et al [24] implemented a tool in the frame of Cloud4SOA project that allows the portability of Java applications between multiple PaaS. Kolb and Wirtz [11] defined a model of current PaaS offerings and identify different portability perspectives. Starting from the model, they derive a standardized profile with a common set of capabilities that can be found among PaaS providers and matched with one another to check application portability based on ecosystem capabilities (project homepage: https://github.com/stefan-kolb/paas-profiles).	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Moreover, the semantics are used to annotate applications and services (more precisely, their properties), in order to deploy and execute them efficiently on the selected infrastructure. Several theoretical studies have been issued expressing various aspects of using semantics for Cloud porta-bility. We mention the most significant ones, in chronological order.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Ten IaaS were mapped to the proposed taxonomy and then compared using the taxonomy. Di Martino [35] is underlining that metadata added to specific service APIs through annotations pointing to generic operational models could enable the portability and interoperability among the heterogeneous Cloud environments. Another portability solution that is discussed in [35] is that of the Cloud patterns usage. The patterns are describing common behaviour in Cloud environments and they support the idea of an application re-design, reengineering or redeployment.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	NoSQL services, message queues and memcache). The library supports the portability of applications across Java platforms for Ap-pEngine and Azure, reconciles the difference in the semantics of the queue services offered the two PaaS, and implements two different services called task queue, adopting the AppEngine semantics and message queue adopting the Azure semantics. mOSAIC ontology& semantic engine	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > DevOps	This approach enables the continuous evolution of the system with no strict boundaries between design-time and runtime activities (supporting therefore the DevOps concept). The approach is adopted for Clouds by CloudML, promoted by MODAClouds, ARTIST [46] and PaaSage [47], a follow up of PIM4Cloud from REMICS [48]. CloudML is domain-specific modeling language enhanced with a run-time environment. It facilitates the specification at design-time of provisioning, deployment, and adaptation concerns, as well as their enactment at run-time [49].	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	– Monitoring tools are generate alarms needed to trigger a re-deployment – Rely upon adaptors that need to be build for new services or updated when a service version appears Semantics – Offers an abstraction layer that can support various customers – Not widely adopted – Offers viable mechanisms for common un-derstanding of service terminology and ac-tions – The variety of taxonomies and ontologies makes the portability problem to grow – Allow the annotation of services with quality marks by externals from the provider team – The overhead of semantic processing is not	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Monitoring	New standards are emerging nowadays, like TOSCA or CAMP, and several collaborative groups are work-ing to elaborate other proposals. However there are several gaps in the collections of available standards, like proposals for Cloud metrics and real-time monitoring, interfaces for security(-as-a-)services, accountability associated with transparency and responsibility.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Workload migration and workload management can be the subject of dynamic use cases in which location, negotiation, and provisioning of Cloud resources occur at run-time. Cloud service portability has not been examined until now with regard to management and operational tasks. These tasks are an increasing and significant cost factor.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	The diversity of the APIs is natural, as each providers intends to offer something new or unique compared with other offers, in order to attract customers. The portability issue is therefore an issue for the management and governance levels where automation should be achieved as much as possible. Limitations of abstraction layers include maintenance in response to changes made by a Cloud provider, and the limited coverage of provider functionalities.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Silva et al [4] made a comprehensive survey of Cloud lock-in solutions. They claim that there is no solution dealing with the impact of changes in the business process due to the use of one specific product, or working on issues related to the cross-organizational business in a portability scenario. Furthermore they claim that, from a research point of view, a challenge is to conduct more rigorous studies (e.g. only part of the studies of studies included a form of evaluation).	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Different Cloud services rely on specific rule engines to help enforcing the rules governing the service during its whole life-cycle. This fact reduces the portability chances of any given set of rules. The usage of a standard procedure for specifying the governance rules is needed to ensure the rule portability. According Moran et al [51], the Rule Interchange Format (RIF) is likely a candidate as specification language. Portability degree needs to be detected by experiments. There are only few reports that are exposing such experiments.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	5.4. Research agenda for Cloud portability. Following the comments that were exposed in the pre-vious section, Table 5.2 proposes a research agenda for Cloud portability, with topics split on short and long term (next two years, respective a half decade). 6.	Ivon Miranda Santos

Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	Conclusions. In this paper, we conducted a survey on portability of the applications that are consuming Cloud services. We provided a taxonomy of Cloud portability, the latest solutions, reviewed existing works and identified the missing pieces needed to solve the problem. In the end, we also pointed out potentially future research directions and useful design guidelines for Cloud portability. The focus of the paper was on the methodological and technical tools supporting the portability. However the porting a particular application requires also a decision methodology and an approach for the analysis of the destination Cloud environment, that were not discussed in this paper.	Ivon Miranda Santos
Petcu2014-Portability_in_clouds_Approaches_and_research_opportu	CLOUD MANAGEMENT > Portability	portability – Tools for the full service cycle, including Cloud governance – Automate re-deployments – Open-source platforms ensuring automated portability or encompassing various ap-proaches Acknowledgment.	Ivon Miranda Santos
Rai2015-Exploring_the_factors_influencing_the_cloud_computing_a	CLOUD MANAGEMENT > Cost	In this background, cloud services offer a more agile and cost effective platform, to support business applications and IT infrastructure.	Ivon Miranda Santos
Rai2015-Exploring_the_factors_influencing_the_cloud_computing_a	CLOUD MANAGEMENT > Performance	All outcomes should be specified. Outcomes - Secure framework for migration, improved security aspects, performance, cost benefits, applications, tools and techniques. Search string	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	MODERN IT systems are increasingly adopting cloud computing and moving their workloads on cloud to take advantage of its various benefits such as economies of scale, energy efficiency, scalability, and elasticity. At the same time, the applications that run on cloud require their data to be available in cloud for adequate performance, accurate data processing, and operational efficiency. Healthcare organizations have a major concern about the availability and sharing of information across the continuum of medical care due to the healthcare data being typically dispersed across various medical systems located at multiple sites.	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	(HIPAA CFR §164.308) • Offer backup and disaster recovery to support failover and system recovery target times for data availability during disasters. (HIPAA CFR §164.308) • Collect key performance indicators (KPIs) to monitor op-erations and provide business intelligence. (HIPAA CFR §164.308 Part 4)	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	Big data services, like HDI, depend on solid implementations of non-functional requirements to support the volume, velocity, variety and verac-ity of healthcare data [25]. Given the big data and regulatory requirements, the HDI service must ensure auditing, logging, disaster recovery, high availability, security, and performance. Fig.	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	In our use-case, some of the data, such as patient profiles, device information, and organization information, can be cat-egorized as "low-volume," while others, such as the inhalation events, observations, and patient feedback, can be categorized as "high-volume". The "low-volume" and "high-volume" data can be stored separately, using different storage technologies to optimize performance. The type of operations needed over the data are also critical in deciding the underlying storage.	Ivon Miranda Santos

Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	To support our use-case, we selected a relational database IBM DB2 and distributed storage based on the Hadoop ecosystem—HDFS, HBase, and Kafka. HDFS provides data distribution by replicating data across multiple nodes, which is required for redundancy and performance. HBase is configured to distribute data across multiple regions using a rowkey that is designed to avoid hotspotting, where hashing is used to put data in the same region.	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	In our experience, this should be handled by pre-processing the data through, a terminology service—to normalize healthcare data with consistent terminol-ogy; a master patient index service—to map patient identity across multiple sources. The volume, velocity and variety of healthcare data require special care with regards to performance. For instance, data from devices is sized at 10 KB in batches, while a single image can be over a GB. The data size and volume must be considered when choosing the block size of storage, or deciding	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	between HBase vs. HDFS [25]. Data shape is an important factor for performance. In our experience, the data shape of var-ious operations—study registrations, medication administration events, prescription updates—is related to functional require-ments.	Ivon Miranda Santos
Ranchal2020-Disrupting_healthcare_silos_Addressing_data_volume_	CLOUD MANAGEMENT > Performance	5 and lead to inefficiencies when querying the data. In our solution, data is buffered into optimal data blocks, such that a bulk of data is written to the data storage block (about 50 records or if there is no write to the buffer for about 5 minutes)—balancing storage, performance, analysis. VI.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	With companies moving forward with various cloud initiatives, cloud consumers are discouraged from relinquishing their control of the infrastructure to the cloud providers. The rivalry among cloud providers to stay competitive in the market makes it necessary to lock-in its customers, and there-fore customers cannot migrate easily to another due to non-interoperable APIs (Appli-cation Programming Interface), portability and migration issues. In addition, today's cloud users are mobile devices and consuming a cloud service onto mobile device poses another set of risks.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	So, in writing flexible code for the cloud, there are four key concepts: Portability, Interoperability, Federation, and Multi-Cloud. When two cloud providers can communicate with each other, it is Interoperability. When a cloud app works across heterogeneous cloud providers, it is Portability. When mul-tiple cloud vendors interact coherently to achieve a custom task to achieve either peak performance or cost efficiency, it is Federation and when a third-party tries to ease the communication among multiple cloud providers, it is Multi-Cloud.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Performance	When a cloud app works across heterogeneous cloud providers, it is Portability. When mul-tiple cloud vendors interact coherently to achieve a custom task to achieve either peak performance or cost efficiency, it is Federation and when a third-party tries to ease the communication among multiple cloud providers, it is Multi-Cloud. Gener-ally, to eliminate vendor lock-in, an application needs to be portable and purposely designed to be used across multiple clouds.	Ivon Miranda Santos

Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	Developer centric platform as a service offerings are in need to create a “mapping layer” between their own system APIs and cloud service APIs using a cloud service broker. Hence, there has to be an adapter layer, a Platform as a Service (PaaS) platform to support application portability fronting Software as a 1950012-2	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	There are various solu-tions enforced to handle this problem, but efforts to service mobile clients are relatively low. Hence, to harness the power of cloud, there is a need for an integrated solu-tion that surpasses the technical barriers of data and services mobility, application portability, and vendor lock-in. There needs to be a single point of contact through a middleware for ensuring that a single application works on all cloud provider environments irrespective of the programming base it is relying on.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	To understand the differences in the API set, a comparison is made on the differences in the invocation style of the cloud providers. The basis for defining the APIs for solving the portability issue is identified and the generic overview of the middleware is proposed to achieve the stated problem. Thus, the outline of the paper goes this way.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Performance	6 gives the implementation details of the middleware. Section 7 elaborates on the performance analysis of the middleware, while Sec. 8 concludes the paper with future directions.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	The broker also takes care of the heterogeneity in the programming envi-ronments of the cloud providers while negotiating the terms of service consumption. Thus, through interoperable APIs and standard protocols, interoperability can be achieved in the case of federated clouds, and by ensuring data and service mobility, portability can be achieved for multiple clouds. Below are the key forces that drive the development of multiple clouds technical solution.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	Avoiding vendor lock-in is challenging as each provider differs by proprietary implementations with custom interfaces and APIs. However, it could be possible if interoperability and portability issues are solved. • Disaster recovery	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Cost	• Optimize costs and save energy One of the main advantages of cloud computing is the “pay-as-you-go” feature. This feature grants monetary benefits for customers by removing the cost of acquiring, provisioning, and operating their own infrastructures. It is just similar to hiring a call tax for travelling, than maintaining an own car. You only pay for the service used, that of travelling and do not worry about the car maintenance and oth-ers. Similarly, smaller organizations would require strong collaboration with other providers without investing heavily on its capital.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Cost	Hence, the cooperation among cloud providers is required to lower the overall costs and optimize the energy used during the execution of the tasks.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	Here are those: Application Programming Interface View/Application Portability — It is natural for a provider to offer its own unique set of APIs to stay competitive in the market. APIs generally acts as a glue to connect to the underlying resources.	Ivon Miranda Santos

Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > DevOps	Modeling View — Once the application is deployed and adapted to a certain cloud, in order to move it in another cloud, an inspection of the source code is needed to identify the specific API calls or to build a model or representation of the code. Tools that can do are Cloud4SOA for semantic-based PaaS offering,31 REMICS for combining model-driven approach with agile practices,32 and Multi-Cloud DevOps Alliance (MODAClouds) for model-driven development of multi-cloud operations.33 For this, model-based solutions are becoming increasingly popular in cloud com-puting as they provide domain-specific modeling languages and frameworks that enable architects to describe/select/adapt multi-cloud environments. This strategy is summarized as “Model once, generate anywhere”. Unlike programming libraries, they work at a high level of abstraction by focusing on cloud concerns rather than implementation details. Model-driven engineering is employed in a new technique called Cloud Blueprinting.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Portability	The lack of for-malization hinders the understanding of the models, thus complicates the inter-operability in multiclouds. Reference Architecture for Semantically Interoperable Clouds (RASIC), Cloud Provider Independent Model (CPIM), Open-source API and Platform for Multiple Clouds (mOSAIC), and UCI enable semantic interoper-ability across clouds for ensuring portability. Service-Oriented Computing View — Web services are well-known Service Oriented Architecture (SOA) implementation models that could successfully inte-grate heterogeneous services from various service providers like Facebook, Google, and Yahoo and enable interoperability across them.	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Performance	Figure 7 shows the success rate of simultaneous requests over multiple nodes. With the reception of concurrent requests, the performance drops down. When there are 2, 4, and 6 nodes, there is a noticeable improvement in the success rate. After which, there is not a exponential growth, however, the performance is increas-ing. Upon reaching 18 nodes, there is a 100% success ratio. Moreover, it is also perceived that adding more than 18 worker nodes does not show any significant improvment in the performance and the performance becomes stable after that. To conclude, with the existing FUSION platform, the load balancer can support upto	Ivon Miranda Santos
Ravi2019-Emergence_of_middleware_to_mitigate_the_challenges_of_	CLOUD MANAGEMENT > Performance	2000 simultaneous requests using 18 worker nodes for a 100% success ratio. After which, the addition of more nodes does not necessarily improve the performance. However, the cloud bursting capacity from Eucalyptus to a public cloud such as AWS can help in achieving higher scalability more quickly.	Ivon Miranda Santos
Raza2019-A_review_on_security_issues_and_their_impact_on_hybrid	CLOUD MANAGEMENT > Cost	Today many enterprises for cost savings IT cloud technologies are increasingly turning to hybrid clouds, allowing them to combine the benefits of building private and public clouds as well as to leverage the scale inherent in their existing IT Infra-structure to cut costs and modernize IT operational agility for service delivery requirements.	Ivon Miranda Santos

Raza2019-A_review_on_security_issues_and_their_impact_on_hybrid	CLOUD MANAGEMENT > Portability	B. Interconnectivity The parallel processes in which two coexisting environments communicate and interact facilitate the exchange of data, VMs and applications among individual clouds. C. Portability of Applications Using cloud aware development builds systems from re-usable components that will work the same across cloud environments. D. Monitoring and Management across Cloud Environments In a Hybrid clouds, monitoring and management is essential for the health of the system, visibility into system health across clouds is crucial In spite of such significant benefits, migration of IT cloud technologies from enterprises have important aspect over privacy, integrity, security concerns and compliance considerations due to reliability on multi cloud vendors such as Microsoft, Amazon and Google [7].	Ivon Miranda Santos
Raza2019-A_review_on_security_issues_and_their_impact_on_hybrid	CLOUD MANAGEMENT > Monitoring	D. Monitoring and Management across Cloud Environments In a Hybrid clouds, monitoring and management is essential for the health of the system, visibility into system health across clouds is crucial	Ivon Miranda Santos
Raza2019-A_review_on_security_issues_and_their_impact_on_hybrid	CLOUD MANAGEMENT > Performance	Undale et al. [10] describes comparison and performance review of AES, Blowfish and RC6 Symmetric cryptographic algorithm in hybrid cloud application with standard cryptographic techniques, such as Proxy encryption, ABE (Attribute based encryption)with its types. To make an efficient solution and to make an NP Complete solution of image encryption problem on hybrid cloud environment.	Ivon Miranda Santos
Raza2019-A_review_on_security_issues_and_their_impact_on_hybrid	CLOUD MANAGEMENT > multi-tenant	Gharat et al. [11] introduce several techniques have been developed for secure data and authentication and to maintain trustworthiness. For trust management system using feedback filtering approach will be explained. In this paper addresses different methods for data protection includes Single Encryption, Multitenancy and Multi level Virtualization, Authentication model including several scenarios.	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	Additionally, the adoption of a university was examined by Sarkar and Leslie (2011) who presented a case study of a large Australian university, with a risk-averse IT department, that has begun to engage in Cloud Computing. Luoma and Nyberg (2011) did an exploratory and holistic study on how the adoption of Cloud Computing in China is affected by performance and effort expectancy, social influence and organizational and infrastructural readiness. Cloud management:	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	The third level (evaluation criteria) was defined to cover aspects ("soft criteria") which cannot be measured and compared easily. The fourth level (key performance indicators, KPI) is defined to realize an assessment and controlling basis, e.g. relevant for a Cloud benchmarking.	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Cost	Scope & Performance IT Security & Compliance Reliability and Trustworthiness Flexibility Costs Service & Cloud Management	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	Scheme of the Cloud Requirement Framework (CRF) The target dimension "Scope & Performance" cover the functionality and performance of the Cloud service and consists of four abstract requirements: service characteristics, service optimizing, hardware, and performance. The dimension "Flexibility" describes the ability to respond quickly to changing capacity requirements and competition pressure.	Ivon Miranda Santos

Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Portability	It is divided into four abstract requirements: interoperability, portability, delivery model, and automatization degree. The target dimension "I	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	Regarding to our research most of the requirements of the dimensions Costs, Reliability & Trustworthiness, IT Security & Compliance and Service & Cloud Management are independent of the service model. The target dimensions Flexibility and Scope & Performance consist mostly of abstract requirements and evaluation criteria specific to a service model. Furthermore, we divided the scope of the requirements into criteria associated with the provider or related directly to the Cloud service in particular (see Figure 2).	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Cost	Regarding to our research most of the requirements of the dimensions Costs, Reliability & Trustworthiness, IT Security & Compliance and Service & Cloud Management are independent of the service model.	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Cost	Costs Service & Cloud Service Requirement Management Reliability & Trustworthiness IT Security & Compliance Flexibility Scope & Performance Evaluation criterion	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > multi-tenant	Operations protection includes the access management and role concept related to the used services. Furthermore, it can provide a multi-tenancy and firewall protected infrastructure, including virus protection systems. Reliability & Trustworthiness	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	Reliability & Trustworthiness Trustworthiness characterizes the provider, its infrastructure and its business activities, including performance and service transparency (e.g. reports, service description), market experience, the number of customers or the annual revenue. Disaster recovery describes activities related to regularly backups, snapshots and data mirroring in other locations.	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	contract length) and defined contract automatisms (e.g. cancelation period). Scope & Performance Usability and customizability refer to the usability and adaptability of the surface of the web portal, the user interacts with.	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Cost	The specific requirements cover only four dimensions and are not relevant for the dimensions "Service & Cloud Management" and "Costs".	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Portability	Flexibility Service and data portability contain the aspects relevant for the service and data mobility. This includes the provider support related to the data migration, the data backup and the data format. The portability of data is especially of high relevance on the SaaS level and can help to lower the lock–in-effect of the provider. The service portability means the possibility to migrate existing services to another platform (IaaS or PaaS), e.g. proprietary virtual images (AMI) of Amazon are transferable to a Microsoft Azure platform. Automatic resource booking and usage limits characterize the capability to control and manage Cloud services without the need of manual interaction.	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	These presets will be considered during the operation and automatically be executed by the system, e.g. boot up a virtual instance, installing regularly updates or increase necessary transfer volume. Scope & Performance Service optimizing deals with continuous service development, improvement of service functions and maintenance cycles.	Ivon Miranda Santos

Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	The operating platform relevant for the IaaS and PaaS level describes the operating system and the development environment. Whereas the performance & hardware requirements, associated with the IaaS level, contain information about the processor type (32 or 64 bit), the hardware based functionalities (sleep mode), the server type (dedicated or virtual server) and the performance aspects (CPU, RAM or storage). Functional coverage & scaling is directly related to the service usage and cover the offered functionalities for PaaS or SaaS. 5 Implications, limitations and future work	Ivon Miranda Santos
Repschlaeger2012-Cloud_requirement_framework_Requirements_and_e	CLOUD MANAGEMENT > Performance	On the other hand customers will be guided by means of this framework to adopt and implement Cloud solutions, especially for selection and comparing providers or to advance the comprehension of Cloud Computing. The consequence is a shift from a subjective service assessment to a mostly fact-based performance selection where the realization of service requirements is gaining importance. In this context Cloud integrators and aggregators are becoming more relevant to advice customers and to realize a Cloud ecosystem which allows the combination and communication between several Clouds and services of different providers.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Cloud Migration, Total Cost of Ownership, Monetization, Architecture Migration, Software Producer.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	The analysis of relevant cost types and factors of cloud computing generate relevant information for the software producers when deciding to adopt cloud computing, and defining software pricing.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	We present an integrated framework for informing cloud monetization based on operational cost factors for migrating to the cloud and test it in a real-life case study.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > multi-tenant	SPs typically migrate their software to a third-party platform (Infrastructure-as-a-Service – IaaS – or Platform-as-a-Service – PaaS) and their customers access it from this new multi-tenant architecture. In a cloud environment both SPs and their customers are typically charged on a pay-per-use or subscription basis.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Rationally, SPs should offer their software at a higher price to compensate their migration costs,	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Cost savings are a major factor in cloud adoption (CFO Research, 2012; Bain and Company, 2017), however ex-ante TCO estimation is not straightforward due to the presence of long-term and hidden costs of operating in the cloud which tend to be ignored or underestimated (ISACA, 2012).	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	From an SP perspective, this represents a major concern since properly mapping the costs of the cloud represents the basis for adequate and effective pricing strategies.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Cloud implementation and operating costs were divided into eight different categories that mainly represent fixed costs, such as setting-up and maintenance costs that providers need to bear during the whole lifecycle.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	The extant literature is typically focused on ex-post calculation of costs and profits independently from the wider situational context, and typically considers only cloud operational cost.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	(2015) examine cost-aware cloud metering for scalable services.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	TCO is used to estimate the cost of cloud investments from the initial sourcing through to the end of the cloud usage, whether that is the backsourcing of information, or the client switching to other services or providers.	Ivon Miranda Santos

Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	While the measured nature of the cloud allows for a detailed ex-post cost analysis, ex-ante cost estimation can be complicated due to the uncertainty associated with multi-tenancy and resource pooling.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > multi-tenant	While the measured nature of the cloud allows for a detailed ex-post cost analysis, ex-ante cost estimation can be complicated due to the uncertainty associated with multi-tenancy and resource pooling.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Suddenly, they can leave or radically modify their usage, since switching costs in the cloud are significantly lower than on-premise.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	minimum cost. Therefore, a key question for a decision maker is: how many components can I host on a fixed cloud compute resource with a pre-defined latency performance target for a forecasted number of users of a particular application with a forecasted mix of application operation usage? Changes in usage require changes in the number and/or configuration of cloud resources used, which may result in additional costs.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	SLA-backed service that is not entirely fixed in terms of computational and storage resources allocated. Finally, the actual capacity of the offered cloud service may fluctuate over time affecting potential economies of scale and application performance. Only the cloud service provider, and not the SP, can monitor the underlying service availability thus, the first problem is right-scaling i.e.,	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	From an SP perspective, the relationship between cloud cost and price (P) can represented as follows:	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	TCO in a strict sense, is the sum of the initial investment required to purchase an asset (CapEx) plus the operating costs that the cloud generates (OpEx).	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Migration costs tend to be omitted in cloud TCO estimations even though they can be substantial and change the overall return on investment.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	, usage) IaaS Cost components while CapEx includes migration and implementation costs (e.g. development and testing, project management etc.)	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	4 IaaS COST CALCULATION PROCESS The nature of the cloud makes it difficult to determine the input variables of the TCO model, but, we will see, architecture quality concerns such as performance and availability can drive this process. Cloud architecture qualities, and corresponding costs, can be influenced by compute, storage and network resources.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Cloud architecture qualities, and corresponding costs, can be influenced by compute, storage and network resources.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	An SP requires a clear comparison of costs and revenues resulting from the cloud adoption.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	This will also further clarify the impact of cloud software architecture on costs and revenues.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > multi-tenant	This process results in a so-called cloud-native application, which is scalable/elastic, clusterable, multi-tenancy, pay-per-use, and self-service	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	Estimated implementation costs (CapEx) were classified into seven implementation phases: Business Analysis, Cloud Architecture Design, Data Design, Security Framework Design, Development and Test, Performance and Costs Analysis. It should be noted that the calculations do not include the operational costs of migrating the customers to the new cloud web application.	Ivon Miranda Santos

Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > multi-tenant	However, the functional dependency between these do not need to be considered in the TCO analysis since the image processing worker VM acts completely asynchronously to the web server role web requests which continue regardless of the state of the image processor. Therefore, we have calculated the multi-tenant VM requirements based on a simple linear multiplication of the CPU load per tenant. IaaS usage charges (OpEx) are estimated considering the two most relevant cost components:	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	usage of cloud storage resources, we used actual historical data over an eleven-month period from an existing average-sized tenant with a typical application usage pattern. To estimate the computing resources required, we monitored the usage and performance statistics during a snapshot of the operational use of the application by the same typical user. Tables 5, 6, and 7 summarize the usage profile adopted in the calculation.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	6.3 Experimentation – Usage and Cost Table 8 summarizes the estimated implementation and migration costs for the SP (€168,647).	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	The most significant cost component, which represents 47.83% of the overall migration costs, is by far consultancy costs for design and development, followed by security design (16.15%).	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Such a significant amount of upfront migration costs further highlights the need to include such costs into TCO estimation to inform both adoption and pricing decisions.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Migration and Implementation Costs.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	The use case we present in this paper involves a significant image-processing component resulting in high upload- and download- volumes and the in-cloud processing of images. The most critical challenge at the architectural level was to select the optimal Virtual Machine type from the available types on the Azure platform; we carried out a benchmark study of the performance of the different “flavors” of the role VMs, when running the data layer functions of the new application. The costs presented in Tables 9, 10, and 11 are based on the D2-v2 VM type which represented the best trade-off between TCO and SLA requirements on the basis of the average tenant usage.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Performance	No optimization of the queries to the table service to optimize CPU load over the TCO estimation period. No performance tuning on the application and/or on the platform during the TCO estimation period. There is no smoothing effect of multiple tenants sharing the same application compute resources.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Research has covered costing and migration separately.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	An investigation linking architectural decisions and the impact on costing in cloud migration is therefore important and this paper makes an initial contribution in this context (Li et al.,	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	An approach for cost estimations in cloud migration.	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Cloud adoption, like all IT investments, results in direct tangible costs such as cloud resources but also in intangible costs, e.g.,	Ivon Miranda Santos
Rosati2018-Making_the_cloud_work_for_software_producers_Linking	CLOUD MANAGEMENT > Cost	Our paper highlights the need for collaboration between business, accounting and computer science researchers in order to understand the implications for costing, pricing and software design in the cloud computing context.	Ivon Miranda Santos

Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	The dynamic nature of the cloud environment increases the complexity of manag-ing its resources and the distribution of user workload between the available con-tainers in the data center. However, the workload must be balanced to improve the cloud system's overall performance. Generally, most of the existing load balancing techniques sufer from performance degradation due to the communication over-heads among the containers. Moreover, less attention is given to stabilize the load in a multi-cloud environment. Therefore, to overcome this problem, there is a need to develop an elastic load balancing method to improve the performance of cloud systems. This paper proposed an autonomic CSO-ILB load balancer to ensure the elasticity of the cloud system and balance the user workload among the available containers in a multi-cloud environment.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	Based on the task scheduling, the load in each container is com-puted and then balanced using the proposed load balancer algorithm CSO-ILB. The proposed approach is evaluated in the Container CloudSim platform, and the perfor-mance is compared with the existing meta-heuristic algorithms such as Ant Colony Optimization, Bee Colony Optimization, Shufed Frog Leaping Algorithm and Cat Swarm Optimization (CSO). The simulations proved that the proposed approach outperformed the other approaches in terms of reliability, CPU utilization, make-span, energy utilization, response time, execution cost, idle time, and task migration.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Cost	The simulations proved that the proposed approach outperformed the other approaches in terms of reliability, CPU utilization, make-span, energy utilization, response time, execution cost, idle time, and task migration.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	This can be achieved by utilizing the self-management capabilities and control loop of autonomic computing, where the autonomic manager moni-tors the cloud resources and automatically responds to dynamic changes in work-load [4]. The autonomic capabilities can be utilized to improve the performance of the cloud system in various aspects [5]. Moreover, adding these autonomic capabilities to the load balancer can efciently resolve the communication over-head and address the load redundancy problem, thus reducing the need to apply the load balancing algorithm [6].	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Monitoring	This can be achieved by utilizing the self-management capabilities and control loop of autonomic computing, where the autonomic manager moni-tors the cloud resources and automatically responds to dynamic changes in work-load [4].	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Cost	On the other hand, when the provisioned resources exceed the demand (over-provision-ing), the cloud provider compaction cost increases [1].	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	Moreover, depending on the workload, assigning work-loads to the cloud containers may get overloaded or under-loaded. Thus, apply-ing efcient load balancing methods becomes signifcant in optimizing cloud sys-tem performance [25]. The reliability of cloud services is based on the number of resources available and the number of users requesting resources [26].	Ivon Miranda Santos

Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	When the resources in the cloud cannot process the huge business processes, the cloud elastic-ity is declined, and the solutions become unfeasible [27]. Load balancing in a multi-cloud environment ofers performance improvement and elasticity to process huge business tasks within a minimum time. Also, the auto-scaling mechanisms improve the efciency of the cloud environment, thereby refecting the overall reputation [28, 29].	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	This section briefy describes some recent studies on loading balancing, which aim to ensure elasticity using autonomic computing concepts and auto-scaling strategies. However, in a cloud environment, numerous researchers have formulated load bal-ancing with auto-scaling mechanisms to improve the performance and efciency of the cloud system. To this extent, Ullah et al.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	[30] proposed a biologically inspired auto-scaling approach to address the horizontal elasticity problem from the service providers' perspective. They aim to attain the desired level of performance while minimizing computational costs. Moreover, addressing other issues such as the defciency of adaptability, burden of computational overhead, not considering the uncertainty aspects in designing the auto-scaling strategies, and applying the biolog-ically-inspired feedback switch controller with a fuzzy system.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	To improve the elasticity of the cloud environment, an auto-scaling based load balancing scheme was introduced by Arvindhan and Anand [36]. The approach reduced the response time in load balancing to attain high performance. Proactive auto-scaling was done to improve the overall resource elasticity.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	Therefore, the proposed article attempts to maintain the load balancing and ensure the vertical and horizontal elasticity in containerized multi-cloud through autonomic computing. The existing ACO [48] takes much response time; hence SLA violation occurs, leading to performance loss. The artifcial Bee Colony Optimization (BCO) [49] includes the system's reliability problems.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	1 3Table 1 (continued) Ref Year Objective Virtualization Methods Elasticity Type Scaling Application and Workload Deployment Benchmark [31] 2020 Performance, economy, avail-ability, infrastruc-ture cost, resource	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Cost	The deadline violation leads to a certain penalty, such as cost for the cloud provider.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Monitoring	The main aim is to achieve horizontal and vertical elasticity using load balancing. The cloud load balancer autonomously monitors the load of the available containers in the datacentres, and then it decides the migration of the	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	tasks based on the monitoring data. The overloaded containers are chosen, and the load is migrated to the under-loaded containers to enhance the overall performance. The tasks are migrated to the optimal under-loaded container that fts the require-ments.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Monitoring	The components monitor, analyzer, planner and executor remain decentralized to main-tain a balance between multi-cloud containers during the auto-scaling process.	Ivon Miranda Santos

Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Monitoring	3.2.1.2 Managed sub-system This sub-system is located directly in the cloud envi-ronment with the installation of sensors to monitor the dynamic changes and events. It is located below the managing sub-system to perform the functions based on the adaptation decision taken by the managing element. The managed element incorpo-rates the application logic, which stays as a medium for directing the observed input from the environment to the system. Figure 3 depicts the working of a self-adaptation module with a multi-loop in a multi-cloud environment.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	3.2.3.1 Problem formulation After determining the container load, the load balancer algorithm is utilized to balance the load among the containers through task migration. While migrating the tasks from an over-loaded container to an under-loaded container, the optimal container has to be chosen without any degradation in the performance met-rics. For this purpose, the CSO-ILB algorithm is put forth, which encourages optimal	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	Reliability: Reliability refers to the consistency in the performance of the cloud resources. The reliability is expected to be maximum for the efcient performance of cloud systems. where, Frl and Frg indicates the total successful fnishing of tasks under resource	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	The response time is the time taken between the starting, process-ing and end time of a task on using a resource. Response time should be minimized to enhance the performance of the cloud. It can be given as:	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	4 Simulation analysis This section presents the performance analysis of the proposed approach with the other existing load-balancing techniques based on the meta-heuristic optimization algorithms. The existing meta-heuristic algorithms for load balancing, such as ACO [48], BCO [49], SFLA [50] and CSO [51], are taken for comparison.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	4.1, the considered metrics for evaluation are explained in Sect. 4.2, and the performance analysis is presented in Sect. 4.3.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	4.1 Simulation setup Various experiments are carried out to prove the performance and efciency of the proposed approach. Due to the dynamic behaviour of the cloud environment, the Container CloudSim platform is chosen for implementation and evaluation [53].	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	The traces of business tasks from the GWA-T-12 BitBrains are taken to evaluate the proposed approach [54]. The dataset is constructed based on the performance of 1750 containers in processing a total count of 8260 tasks. This dataset is divided into fast-storage and rnd where the fast-storage dataset is considered for the evalua-tion. It holds the performance output of 1250 containers where the performance val-ues obtained are stored in.CSV fles.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	the model has been implemented in Intel (R) Core (TM) i5-4570S 2.90 GHz proces-sor, the RAM installed is 6 GB, and the type of system is a 64-bit operating system. 4.2 Performance metrics The major performance metrics for evaluation are CPU utilization, reliability, make-span, energy utilization, response time, execution cost and task migration. The for-mulations are as follows:	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Cost	The major performance metrics for evaluation are CPU utilization, reliability, make-span, energy utilization, response time, execution cost and task migration.	Ivon Miranda Santos

Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	Reliability: Reliability ensures the system's performance in processing many tasks for some time. Higher values of reliability are preferred for better performance. The mathematical formulation for reliability is provided in Eq.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	This metric is used to evaluate the load balance and is com-puted to identify the degree of load balance. The model aims to reduce the standard deviation to achieve high performance. The mathematical formulation is as follows:	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	where S(t) indicates the scheduled time of the task and A(t) indicates the active time of the task. 4.3 Performance evaluation This section presents the performance analysis and evaluation of CSO-ILB with the other existing techniques. Numerous iterations were performed with various tasks to ensure the overall performance of the proposed approach. The experiments proved that the CSO-ILB outperforms the other approaches in all the iterations and evalua-tion metrics.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	But the migration of the tasks encoun-ters certain issues, and hence the migration has to be limited. Over migration degrades the performance. Table 6 summarizes the results for the makespan, and Fig.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	51 ms of idle time respectively. The performance values of reliability and CPU utilization obtained through simulations for the proposed and existing schemes are provided in Table 11. The values in the table show that the proposed approach outperformed the com-pared approaches in terms of both metrics.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	The experiments proved that the proposed method could efciently balance the load among the available contain-ers by accurately choosing the optimal under-loaded container. Also, the proposed approach attains high performance in achieving horizontal and vertical elasticity through scaling options. The proposed CSO-ILB technique has been compared with the existing ACO, BCO, ASFLA and CSO algorithms. All the existing algorithms are also imple-mented in the multi-cloud environment to evaluate the performance improvement of the proposed approach. From the conducted simulations, it has been observed that the proposed approach obtained better and optimized results than the other existing techniques.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Monitoring	been built for monitoring the dynamic changes in the multi-cloud system to improve the efciency of the scaling decisions. Based on these decisions, vertical or horizontal scaling is performed to ensure the elasticity of the cloud environ-ment. The simulation is performed in the multi-cloud containerized environment using the ContainerCloudsim toolkit to prove the performance and efciency of the proposed approach. The results demonstrated that the proposed approach out-performed the other approaches regarding CPU utilization, make-span, response time, execution cost, reliability, energy utilization, idle time and several task migrations.	Ivon Miranda Santos
Saif2022-CSO-ILB_chicken_swarm_optimized_inter-cloud_load_balan	CLOUD MANAGEMENT > Performance	Based on these decisions, vertical or horizontal scaling is performed to ensure the elasticity of the cloud environ-ment. The simulation is performed in the multi-cloud containerized environment using the ContainerCloudsim toolkit to prove the performance and efciency of the proposed approach. The results demonstrated that the proposed approach out-performed the other approaches regarding CPU utilization, make-span, response time, execution cost, reliability, energy utilization, idle time and several task migrations.	Ivon Miranda Santos

Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > Portability	However, native PaaS environments may be insufficient for applications with special requirements for deployment on a third party environment. For example, native PaaS environments may not support compliant handling of end-user sensitive data which requires special security policies as prescribed by the Health Insurance Portability and Accountability Act (HIPAA) [1]. Another example is the case of applications that bind sensitive data to local (or in country) data support.	Ivon Miranda Santos
Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > Portability	Section 6 presents the conclusions of our findings and future work. 2 Background on HIPAA and HL7 The Health Insurance Portability and Accountability Act (HIPAA) [1] has two major goals: (1) to enable the transfer and storage of electronic Protected Health Information (e-PHI, called here after simply PHI as we only refer to electronical PHI in this paper) and (2) to enable the insurance portability from one insurer to another. Thus, HIPAA fosters the replacement of medical paperwork with standardized electronic medical records (EMRs) enabling healthcare and insurance providers to reduce cost and to be more efficient.	Ivon Miranda Santos
Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > Portability	The healthcare provider is responsible to adopt the Security Rule and maintain written records of the required activities for six years. 2.2 Standardization and Portability Healthcare standards materialize by bills approved by Congress. HIPAA standards cover both the storage and transmission of PHI (e.g.,	Ivon Miranda Santos
Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > DevOps	We used IBM PaaS Cloud as our dev/test environment for an expedite application prototyping and verification, and our Health Compliant Cloud as the target environ-ment to migrate our application to for being hosted in a HIPAA health compliant production environment. IBM Cloud provides PaaS DevOps services which speed-up the development, building, and deployment of online applications as well as the updating of their subsequent versions. For deploying hosted applications, we have two options: installation and migration.	Ivon Miranda Santos
Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > Performance	• The network access to potential 3rd party dependent libraries could be blocked due to security and complaint concerns, resulting in the installation to fail. In the case of migration, where all deployment artifacts come from a verified appli-cation instance in the Dev/Test environment, the files and programs deployed for the application are stable, with no suspicious files to introduce potential risk of exposure, while the configuration is tuned for optimal performance in this new environment. Table 1 compares the installation vs. migration of five popular (see the popularity rank count) Github applications:	Ivon Miranda Santos
Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > Performance	Initialization of the VMs. To boost the migration performance, the VMs can be selected from a preprovisioned pool. 2.	Ivon Miranda Santos
Sailer2018-Healthcare_application_migration_in_compliant_hybrid	CLOUD MANAGEMENT > DevOps	Modern healthcare applications present particular challenges in cloud and hybrid cloud environments. Keeping these applications up-to-date in live cloud environments can be costly and time consuming. HIPAA compliance, in particular, introduces technical and security challenges that are an overload to the developers and operators of cloud native solutions. Our experimentation has shown that through the use of open source tech-nologies, best of breed automation tools and the PaaS Platform interface we are able to implement a DevOps methodology that addresses and meets the HIPAA requirements.	Ivon Miranda Santos

Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Performance	Applications being built today are 'cloud native', or demonstrate qualities that are indigenous to cloud. These applications provide significant performance, maintainability, and scalability. This is a significant driver for migrating legacy applications to the cloud, to exercise these advantages.	Ivon Miranda Santos
Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Cost	There have also been numerous studies on the cost-benefit tradeoff for migration of different applications [2].	Ivon Miranda Santos
Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Performance	A study on the different popular cloud vendors was conducted to analyze the tools available for automating rehosting and replatforming applications, as well as the support available for re-architecting. Amazon Web Services, Microsoft Azure, Google Cloud Platform and Oracle Cloud Infrastructure were studied to analyze the offered tools, performance of migrated applications, cost of hosting, and finally the documentation and resources available to aid both companies and individuals to use their tools effectively. The results from these case studies are documented in Table I.	Ivon Miranda Santos
Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Performance	A. Django CMS This application had a Django framework and a web-compatible UI and thus could be easily rehosted, when compared to the other applications. Its functionalities were maintained post rehosting, and its accessibility and performance was satisfactory. It improved in terms of database scalability after replatforming.	Ivon Miranda Santos
Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Performance	The application was not re-architected. Table III illustrates a comparison of the three migration strategies on Django CMS where metrics such as performance, scalability, throughput etc were assessed. Performance is an aggregate metric that combines first contentful paint(10%), speed index(10%), time to interactive(10%), total blocking time(30%), largest contentful paint(25%) and cumulative layout shift(15%). TABLE III.	Ivon Miranda Santos
Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Performance	The primary motivation for migrating such applications is to take advantage of higher network speeds, reducing technical debt and cost of maintenance. Small applications that require a temporary boost in performance would be good candidates for this approach. Large applications which are resource-intensive may not gain significant benefits and may need other approaches to take advantage of the cloud.	Ivon Miranda Santos
Shastry2022-Approaches_for_migrating_non_cloud-native_applicati	CLOUD MANAGEMENT > Cost	Some of the existing work dealt with the assessment of application suitability to cloud and the cost-benefit analysis of migration.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	Recently, Amazon and Microsoft advertised the same SLA with 99.95% avail-ability for each; hence, it will be converted into 99.9999% availability in the case of using a combination of both clouds.32 In MCE, the atomic web services are composited to deliver a user value-added coarse-grain service.33-35 In the other words, instead of single sourcing, a user deploys multisourcing as “best-in-breed” sourcing from an immense cloud market to find composited services with low cost and better QoS, which comply with his/her SLA.36 Since the cloud market is ever increasing in terms of providers and miscellaneous services and there is no one service directory unit, the broker mecha-nism is applied to mitigate the complexity of user service delivery.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	3 However, in the related works, there are some drawbacks as well as benefits. For instance, some of the related works focus on just sheer economic or limited factors, which have less influence on security issues.	Ivon Miranda Santos

Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	The bioptimization problem should be solved with regard to minimization of both cloud service cost and cloud cybersecurity risk to constitute a Pareto set from the ever increasing large search space of multicloud providers along with their bunch of services in the market, which is why we apply a genetic algorithm in a combinatorial algorithm to reach a Pareto frontier.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	In the main phase, our biobjective optimization algorithm explores the search space, enlisted in the primary phase or enlisted off-line, to select appropriate providers for web service composition based on cost and security risks derived from the cloud broker's log file, thus integrating reliable composition.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	In our evaluation, each cloud with low service cost is not appropriate unless its security risk is also at a low level.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	(1) On-demand self-service: individuals and organizations can use cloud on-demand instances if they cannot present their accurate resource usage pattern; otherwise, they can take benefit of cloud reserved/bid instances to reduce their costs.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	Furthermore, it reduces OPEX for economies of scale and increases resource utilization for the sake of applying virtualization techniques and multitenancy.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > multi-tenant	Furthermore, it reduces OPEX for economies of scale and increases resource utilization for the sake of applying virtualization techniques and multitenancy.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	For instance, SaaS is economic for customers using software with a high license fee such as CRM, ERP software, etc. Software developers can apply cloud platforms to reduce development costs. They can also supply their cloud-based services over platforms such as Microsoft Azure, Google App Engine, etc. Moreover, IaaS such as usage of bandwidth, disks, servers, etc, is a seductive option for the sake of delegating maintenance costs, human efforts, and electricity bills to a third party	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	To address the problem of technology delivery complexity to users, clouds take benefit of brokers; this way, customers can select cost-effective services with better QoS complying with SLA.32,44,48 In this paper, we specifically concentrate on MCE security for mission-critical applications.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	Finally, a cloud com-poser selects reliable composition with both low cost and security risk inclination to cover BUs' requirements.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	For instance, the transaction cost theory determines the costs incurred by making market exchange or cost of participating in the market.55 It comprises 3 cost categories: search and information cost, bargaining cost, and policing and enforcement cost.56,57 The transaction cost theory in the IT context takes all of potential costs into account; hence, it includes other cost theories such as coordination, maintenance, negotiation, adoption, allocation, and IT service costs.58,59 As all of cloud adoption processes are systematic and can be easily customized rather than other industries and businesses, we apply the transaction cost theory with IT service costs along with maintenance costs theory placing into TCO variables in our cost model.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	3.2 Cost model This section indicates that the development of a cost model needs a deep understanding of the cost implications. Here, cost variables are formalized by using the transaction theory in the TCO variable.	Ivon Miranda Santos

Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	3.2.1 Cost specification In this subsection, first, all cost types must be well understood and categorized to be placed into the model. For example, Tak et al10 classified costs in several unnecessarily disjoint classes, including direct, indirect, quantifiable, nonquantifiable, predictable, and nonpredictable costs. The quantifiable costs are amenable to precise quantification such as the server price or human salaries. Cost prediction as the hardest part is related to indirect, nonquantifiable, and nonpredictable costs such as software porting costs in a hybrid scenario. Here, we develop a cost model with direct variables regardless of any deployment option. It is time to extend the participated terms FC and VC in transaction cost estimation.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Portability	Multicloud also brings several benefits such as vendor lock-in avoidance and system fault tolerance. As such, open APIs provide portability and a high degree of flexibility. Service deployment is typically available, which depends on the business and security requirement.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > multi-tenant	According to previous reports,3,38,70 the topmost threats in a cloud environment revolve around shared technology vulnerabilities such as multitenancy risks, data leakage, malicious insiders, denial of service, and service hijacking. If the aforesaid threats materialize, a portion or all of the security requirements are not met	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	3.4 Cloud decision model By using multicloud as a multisourcing option to cover business functional and nonfunctional requirements, several bene-fits are brought to users (whether individuals or organizations), such as a fault-tolerant system, lock-in avoidance, low risk, etc. As such, web service composition is selected to meet the needs, which is made possible by multicloud SOA reusabil-ity. Figure 5 shows the web service composition in cloud application. There are several cloud applications with different composition patterns, ie, sequence, parallel, and branch and loop,77-79 which are supported by Business Process Execution Language,80 but for the sake of simplicity, we consider the sequence pattern, as can be seen in Figure 7. Consequently, the cloud decision model intends to find the optimal composition to minimize both service costs and security risks.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	• The cloud combiner selects a suitable cloud set, which has the most appropriate services to accomplish the user's security requirement along with cost, and produces a cloud combination list based on the set.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Performance	Note that the notation WSij indicates that web service WSi is provisioned by Cloudj. Moreover, web service WSij has the same functionality as WSik to cover task Ti, but differs in cost and security performance. Hence, the main goal of the decision model is to solve biobjective optimization problems as shown in the following equation:	Ivon Miranda Santos

Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Portability	Parameters p and P denote the minimum and maximum deployed providers. The former is to create a fault-tolerant system especially in infrastructure failure, avoid lock-in, etc, whereas the latter is to avoid increasing deployment and portability costs. There is a need to outsource a business process at least over more than one provider to be released from one supplier dependence.27,33,34,51 On the other hand, to reduce portability costs, there is a need to limit the number of suppliers to reduce portability costs.27,33,34 Moreover, parameters p and P can be obtained empirically by experimental results.HOSSEINI SHIRVANI ET AL. 19 4 EVALUATION	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	Then, we compare it as an optimal mul-ticloud solution with on-premises cost to make a final decision.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	4.3.2 Genetic algorithm to find the Pareto front of COST and RISK in multicloud Since the GA is typically designed to optimize 1 criterion such as to minimize the cost function, we change the GA toward our bioptimization problem. We particularly designed our roulette wheel so that solutions are fairly dis-tributed along both criteria and from an accurate Pareto curve. Algorithm 1 and Figure 10 elaborate on the enhanced GA procedure.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	More-over, in this section, Op2 is representative of the multicloud adoption option, and to harness data transfer cost, we use at most 3 clouds per 10 requested web services in MCE.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	As Figure 12 shows, a single-objective model beats a random biobjective model because a single-objective model, at least, strives to minimize the cost function despite neglecting security risks of used clouds.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	Consequently, it finds clouds with low cost and good security SLA coverage.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	With this basis, we extend pure cost calculations for the 5-year investment along with placing the amount of cyberse-curity risk, based on our approach, cost into the total cost of cash flow to reach a sustainable decision about multicloud migration; then, it is compared with on-premises option as Op1.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	As previously stated, annually, the resource requirement makes increasing costs of cloud options in comparison with	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	It shows that the fourth option, ie, Op4, is the best option provided that cloud providers continuously decrease 15% of total service costs for the sake of Moore's law and racing in the competitive market; accordingly, the third option, ie, Op3, is the worst option provided that cloud providers continuously increase 15% of total service costs for the sake of energy cost.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	For instance, in strict mission-critical applications, the impact of matrix information depends on business criticality, which may have a drastic effect on the AMFC variable and, consequently, on the total amortized cost, creating hesitation for multicloud adoption from policymakers, ie, the Op2 bar would be taller than that of Op1.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	However, the flexibility and choice features of multicloud provide an opportunity for customers to find better cost and better QoS/SLA in the market.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	application importance; hence, the large total cost indirectly shows that cloud security disability creates hesitation from the policymakers to adopt MCE.	Ivon Miranda Santos
Shirvani2018-An_iterative_mathematical_decision_model_for_cloud	CLOUD MANAGEMENT > Cost	Although reliance on MCE is a promising alternative, further number of used clouds may potentially lead to more communication costs.	Ivon Miranda Santos

Shyamasundar2017-Information_flow_control_for_building_security	CLOUD MANAGEMENT > Cost	Rapid deployment of hybrid clouds for utility, cost, effectiveness and flexibility has made it necessary to assure the security and privacy of hybrid clouds as it transcends different domains.	Ivon Miranda Santos
Shyamasundar2017-Information_flow_control_for_building_security	CLOUD MANAGEMENT > Monitoring	V. A SECURE ARCHITECTURE FOR HYBRID CLOUD In this section, we provide a general approach for securing a hybrid cloud by integrating an RWFM monitor into the cloud service manager. Further, we also illustrate the working of the approach with a concrete example (no loss of generality), and compare the benefits of our approach with related works.	Ivon Miranda Santos
SOrheller2018-Implementing_cloud_erp_solutions_A_review_of_soci	CLOUD MANAGEMENT > Cost	The benefits of cloud-based ERPs relate to cost effectiveness, time savings, scalability and ease of updates [1, 3, 4].	Ivon Miranda Santos
SOrheller2018-Implementing_cloud_erp_solutions_A_review_of_soci	CLOUD MANAGEMENT > Cost	The benefits of cloud-based ERPs relate to cost effectiveness, time savings, scalability and ease of updates [1, 3, 4].	Ivon Miranda Santos
SOrheller2018-Implementing_cloud_erp_solutions_A_review_of_soci	CLOUD MANAGEMENT > Cost	Due to low implementation costs and simplicity, vendors mostly target SMEs. Furthermore, several of the articles reviewed are not stating explicitly the size of organization under study. It is important for research to be properly contextualized in order to be useful for further development and we urge researchers to report as much contextual information as possible (e.g. organization size, industry, years in operation).	Ivon Miranda Santos
Sousa2016-Automated_Setup_of Multi-Cloud_Environments_for_Micro	CLOUD MANAGEMENT > Performance	Email: firstname.lastname@inria.fr Abstract—Multi-cloud computing has been proposed as a way to reduce vendor dependence, comply with location regulations, and optimize reliability, performance and costs. Meanwhile, microservice architectures are becoming increasingly popular in cloud computing as they promote decomposing applications into small services that can be independently deployed and scaled, thus optimizing resources usage.	Ivon Miranda Santos
Sousa2016-Automated_Setup_of Multi-Cloud_Environments_for_Micro	CLOUD MANAGEMENT > Performance	This includes protecting applications from cloud outages or failures, but also avoiding vendor lock-in. • Optimizing performance and costs. Different cloud providers offer specific advantages.	Ivon Miranda Santos
Sousa2016-Automated_Setup_of Multi-Cloud_Environments_for_Micro	CLOUD MANAGEMENT > Cost	Thus, when setting up a multi-cloud environment to deploy this application, we will be look-ing to optimize costs or improve the quality of service while complying with application requirements and constraints.	Ivon Miranda Santos
Sousa2016-Automated_Setup_of Multi-Cloud_Environments_for_Micro	CLOUD MANAGEMENT > Performance	If a concept is not in the ontology, the developer can use the application specific ontology to include new concepts. For example, one could add a new ontology class called HighPerformanceWebContainer and express that it is equivalent to a WebContainer that has more than 4 virtual CPUs. Once a new concept is added it can be used as part of a service requirements description.	Ivon Miranda Santos

Sousa2016-Automated_Setup_of Multi-Cloud_Environments_for_Micro	CLOUD MANAGEMENT > Performance	By decomposing this process into three steps we avoid doing more expensive calculations when not needed and we reuse intermediate results. C. Experimentation To validate the effectiveness of our approach and evaluate the performance of the developed tools we elaborated an example application based on the requirements described in Section II and measured the time consumed at each stage ofprocessing, as well as the total time to get from requirements to a concrete multi-cloud configuration. The application requirements were described using the domain-specific language for multi-cloud requirements and include 13 services, with 5 cloud variables and 15 instance groups. In addition, we elaborated feature models and on-tology mappings for four popular cloud providers:	Ivon Miranda Santos
Sousa2016-Automated_Setup_of Multi-Cloud_Environments_for_Micro	CLOUD MANAGEMENT > Performance	AVERAGE TIME FOR OPERATIONS IN THE GENERATION OF A MULTI-CLOUD CONFIGURATION. To evaluate the performance in the presence of more providers we randomly generated extra 46 providers in a way that no generated provider would completely support the features required by the example application. With this new set of 50 providers, 52s were need to find the same 32 solutions.	Ivon Miranda Santos
Stavru2013-Challenges_for_migrating_to_the_service_cloud_paradi	CLOUD MANAGEMENT > Performance	In terms of technical challenges, the maintenance and troubleshooting difficulties (T1), together with the lack of cloud support (T3), might require more involvement from upper management (in order to assure the commitment of the cloud provider). Also, the architectural and technical constraints (T2) and the need to consider quality aspects (T2, T4, such as interoperability, security, performance, etc.) might require considerable efforts to be made for architecture and design, except for coding and testing.	Ivon Miranda Santos
Stavru2013-Challenges_for_migrating_to_the_service_cloud_paradi	CLOUD MANAGEMENT > Performance	), managing increased complexity (e.g. through highly collaborative and knowledgeable workers), increased team responsiveness and support (e.g. team support for issues troubleshooting and resolution, improvement of used tools and procedures, identifications of impediments and external dependencies, etc.), secured quality in terms of security, interoperability, performance, etc. Other agile techniques, highly rated were Whole Team and Continuous Integration (from XP) and Product Backlog, Spring Backlog and Daily Scrum (from Scrum).	Ivon Miranda Santos

Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	Today, cloud-based computing, communication, and collaboration have created new frontiers and emerging paradigms towards re-engineering of work cultures in the organizations. In order to enhance the performance with extended features, next-generation computing, communication, and collaboration the intranet needs redesign and migration strategy over alternative technology platforms. This research paper tries to answer the research questions that how an alternative technology strategy or pathway can be explored for enhancing the performance and extending the features of the exiting designs of educational Intranets. Further; how an on-premise intranet can be migrated over cloud platforms with enhanced performance and extended/add-on features. After analysis of collected facts, understanding the issues, challenges and limitations of the existing state of art intranets, a strong need for performance enhancement and add on features was observed for Intranets. The study deeply investigated and analyzed the issues, challenges and limitations i.e. features and performances of the current state of the art of the intranets in general and on-premise Intranet of AMU in specific. Finally, an Intranet Migration Strategy Model over Hybrid Cloud was designed and developed using SaaS (i.e.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	This design is proposed to be Migration Strategy Model for existing intranet designs. The major aim to improve the performance and add on advanced features [1]. B. Cloud Computing	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	C. Intranet over Cloud Intranet over cloud is proposed to be the next generation technology strategy for boundary-less access at any time for the user's community via internet. The Cloud based Intranet Migration Model especially designed for the academicians, researchers and students can enhance the systems efficiency and performance in compare to the traditional/on-premise Intranets. The next generation intranets needs on-demand Scalability, Open boundary Accessibility, Service reliability, High Availability (Uptime), Cost effectiveness, High responsive, fault tolerant and high Performance with promised Security, Anytime, Anywhere over Any device Availability[4,5,7,8]. As on date such issues, challenges and limitations (on-demand Scalability, Open boundary Accessibility, Service reliability, High Availability (Uptime), High responsive, Fault tolerant with promised Security, Intelligent Search Content Filtration, Cost-effectiveness, and Anytime, Anywhere over Any device Accessibility) in the existing designs and practices over intranets are not addressed by specific design or models.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	This study by nature is the mixed version of exploratory and constructive research. Initially this research starts with an effort to explore the possibility of Migrating intranet over cloud for performance and feature enhancements. It explores the answer of the several questions for migrating intranet over cloud with new knowledge in terms of better performance and add-on features. Further, the research constructs a migration strategy model with functional prototype designed over Interact-Intranet platform.	Ivon Miranda Santos

Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	This study focuses on intranets in general and university intranets in specific. As a case analysis, this study selected the Intranet of Arba Minch University (AMU) for assessing its standardized performance and features. This intranet provides many ICT enabled services efficiently and effectively. However, in current digital revolutions; if a system completes its five years of age, it needs to be reviewed or reorganized so as update and upgrade in the light of modernized systems and technologies towards better sustainability and performance. If it continued in its old ways, it can go beyond the control, governance and management and the same is true about the existing state of art Intranets.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	They provided in-depth inputs and the facts about issues, limitations and challenges of the existing state of art intranet in the university system. From their responses, it was revealed that 97.37% respondents are aware about cloud and 89.47 % are using cloud services with promised trust and benefits and interested to have an alternative model/solution for Intranet for modernizing and upgrading the intranet features or services with enhanced performance of the existing Intranet. The fig.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Cost	During the functional demo and survey it was clearly revealed that the current state of art intranets in the educational institutions of Ethiopia are not scalable, cost effective, rapidly upgradable, reliable, all time available with zero down time, and intelligent, and boundary-less accessible.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	v. Designing and Developing the Conceptual Intranet Migration Strategy Model over Cloud: In this phase, enhanced performance measures, add-on features which are mostly selected and popularly accepted by user's communities are required to be considered for conceptual design of the Ideal Intranet over Cloud Platforms. vi.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	vii. Measuring the performance and effectiveness of the prototype of the Intranet over Cloud: Finally, performance of the Intranet migration Strategy model over Cloud can be measured for its performance, features and efficiency.	Ivon Miranda Santos
Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	This research study is an effort to explore the new knowledge contribution in constructing a migration strategy model for Intranets over Cloud. The study was focused on how different organization's traditional computing platforms and intranets are challenged in terms of poor performance and lack of advanced features. The research considered case based analysis of the AMU's Intranet. The research study tried to investigate and analyze the issues, challenges and limitations in existing state of art intranets and their performance, features by analyzing the satisfaction levels of the participants. Finally, the research designed a Migration Strategy Model named as "DPS-AA Model" for the Intranet over Cloud Platforms towards enhancing the performance and extending the features like on-demand Scalability, Open boundary Accessibility, Service reliability, High Availability (Uptime), High responsive, Fault tolerant with promised Security, Intelligent Search Content Filtration, Cost-effectiveness, and Anytime, Anywhere over Any device Accessibility. As a contextual analysis of salient features of the intranet networks; this research conducted feature based analysis of three different types of computer networks namely Internet (the world-wide network of computers accessible to anyone), Intranet (a network that is not available to the world outside of the Intranet), and Extranet (an Intranet that is partially accessible to authorized outsiders).	Ivon Miranda Santos

Tona2020-DPS-AA_Intranet_migration_strategy_model_for_clouds	CLOUD MANAGEMENT > Performance	Modern Education and Computer Science, 2020, 5, 55-63 demonstrated for the performance evaluation. The designed migration strategy model and its functional prototype with limited features have shown the promised results towards performance enhancements and Add-On features in the existing state of art intranets. These result oriented conclusions showcased new pathways towards strong needs for modernization of intranets over clouds.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	Such software is deployed on the self-managed on-premises servers. Monolithic architecture systems introduced many difficulties when transitioning to cloud platforms and new technologies due to scalability, flexibility, performance issues, and lower business value. As a result, people are bound to consider the new software paradigm with the separation of concern concept.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	The results showcase that most researchers and enterprise-grade companies use microservice architecture to develop cloud-native applications. On the contrary, they are struggling with certain performance issues in the overall application. The acquired results can facilitate the researchers and architects in the software engineering domain who aspire to be concerned with new technology trends about service-oriented architecture and cloud-native development.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	A German university conducted research related to the features of the microservice architecture [11]. According to their research, security, performance resilience, reliability, latency, and fault tolerance are the most important features of the microservice architecture. Most people use the microservice architecture to get proper scalability, extensibility, and agility.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	With these intentions, engineers focus on the security of the microservices because it is deployed in the distributed environment. When services are deployed in remote locations, developers need to scrutinize the entire application performance and the response time. Most of the researchers conduct the review using standards research approaches such as the quantitative research approach and the qualitative research approach.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	It can categorize people’s intentions to choose the architecture as product, cost, and process. When considering the product, most of the engineers focus on the product scalability over the cloud, product maintainability, performance, and as well as product security [33]. Most people are moving towards digital services, and service consumers tend to use digital services as well.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	But in the microservice architecture, developers can implement several checkpoints to validate security. Software performance can be measured using several aspects such as response time, throughput, and software capacity. Most of the real-time systems look for less response time.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	asynchronous programming, cloud enablement, etc. By choosing the framework, developers need to speculate several factors like framework maturity, in-build features, development support, performance, and software license. In consideration to all these factors, choose the most used and popular microservice development framework to perform the systematic reviews such as Java base Spring Boot framework [59], Go-based Go Micro framework [60], Node.js based Molecular framework [61], and Java bases, Vert. x framework [62].	Ivon Miranda Santos

Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	4 Architectural patterns in a microservice architecture Based on the above table 1, Spring Boot is the most popular and the number one trending microservice framework of the world, that comprises the most critical quality attributes. Go-micro-framework is also turning out to be famous in the industry because of its performance. Vert.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	But, Vert. X is implemented using the multi reactor pattern and therefore possesses good performance. The molecular framework is not widely popular in the microservice development industry. However, according to its architecture, it is suitable for high-performance microservices. World-famous multimillionaire companies have transited their systems to the microservice-based architecture to get more advantages in terms of the product, cost, and the process.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	Service decomposition patterns illustrate how the system can be independent of several services. Monolithic system holders can use this pattern to segregate their services to maximize the application performance. To use this pattern, architects need to have proper knowledge of the business domain, as well as the technology associated with the domain.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	Some frameworks support validating the security tokens such as JSON Web Token. But, validating the JWT token on each service will lead to an overall performance issue [78]. Another main concern behind the microservice is the application performance in terms of latency [79]. There are a lot of protocols, for instance, HTTP/HTTPS, gRPC, JMS, AMQP, and web sockets to service-to-service communication.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	Nowadays chief technology offices around the world's KPI is to bring the business software into the cloud-native platform. When moving to the cloud-native architecture, cost and application performance are the biggest Sidath Weerasinghe& Indika Perera / IJETT, 70(3), 222-233, 2022 230 limitations. To get better performance with low cost in the cloud-native environment, the application should possess the capability to run on low computational specifications and need to perform with better results. Most of the microservice frameworks are now built according to the cloud-native architecture concepts.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	As an example, spring-cloud supports the GCP / Azure / AWS / Alibaba cloud Integration for microservices. Go micro kind of microservice frameworks enable high performance with the lightweight applications to achieve low cost. Other frameworks are actively working on developing cloud-native support libraries.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Cost	Or else, it is mandatory to use the cloud services (which are inclusive of certain costs).	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	The industry is rapidly moving to microservice architecture to associate with considerable challenges as well. The main challenge is the performance in terms of latency because of the inter-service communication between microservice in the distributed environment. Skill for development of the microservices is another channel in the software industry.	Ivon Miranda Santos

Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	Research questions are chosen to capture all aspects of the microservice research area. Most people are moving their monolithic system to microservice architecture to achieve the quality attributes such as scalability, performance, security, maintainability, etc. In the current context, with the emergence of world pandemic situations, most of the services are served via online platforms and a lot of users are moving towards the online platforms to get the services.	Ivon Miranda Santos
Weerasinghe2022-Taxonomical_classification_and_systematic_revie	CLOUD MANAGEMENT > Performance	Another problem is the troubleshooting issue on the microservice, which is challenging for the support engineers. Continuous research of this to find a solution for the performance issue in the microservice architecture. With the selected research questions, can comprehensively go through the microservice concepts and their related research areas.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	With the availability of large-scale resources from the increasing number of utility providers, mechanisms are required to enable resources to be added to and removed from clouds in an automated and application-centric way. We use a resource commodity market approach to managing resources within our cloud that is used to define which resources should be used and for how long they should be used and that monitors the performance of resources in an attempt to optimise cloud costs. This model is application centric because how the cost of a resource is calculated is specified by the application.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Cost	We use a resource commodity market approach to managing resources within our cloud that is used to define which resources should be used and for how long they should be used and that monitors the performance of resources in an attempt to optimise cloud costs.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	An organisation can create a pool of internal resources using a cloud structure which should lead to improved resource utilisation, reduced management costs, provide a better return on investment and may enable greener operation through consolidation. By combining clouds we can form a marketplace of resources[2] which can be consumed by applications, select-ing the best (application-specific: e.g. highest performance, most bang for buck) available resource to meet their require-ments at any given time. In this marketplace researchers and businesses can match the cost of their computing and storage to their demand.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	By constantly evaluating the reprovisioning cost as well as the runtime cost in another cloud it is possible to automate a system to transition cloud infrastructures as long as the runtime savings will pay for the reprovisioning cost. This system can be further enriched by adding application-specific resource performance metrics (demonstrated in Section IV) which can evaluate the performance of other resources to determine how cost-effective that resource would be. We called the estimated cost of a usage scenario a scenario cost and typically (in our examples) an application will have multiple scenarios.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Cost	By constantly evaluating the reprovisioning cost as well as the runtime cost in another cloud it is possible to automate a system to transition cloud infrastructures as long as the runtime savings will pay for the reprovisioning cost.	Ivon Miranda Santos

Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	D. Dynamic scaling and a resource commodity market The infrastructure is usually deployed with the minimum size initially. Once in place, the infrastructure will scale according to demand (scaling to ensure performance stays within acceptable levels as defined within the deployment metadata). For example, when the streaming pool reaches 70% load, additional capability is deployed.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	This opportunistic behaviour takes advantage of the unpredictable times involved with transcoding and the way in which cloud providers bill for resources (generally rounding up usage by treating part-hours of usage as full hours). At a broad level, the system benchmarks cloud resources using a test video, storing performance data which is used to estimate the future performance of that resource. The provisioning layer searches for resources with no constraints, using a cost scenario which estimates the cost of the upload, transcode and download operations.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	When the resources are returned the cost function then produces an estimate of the efficiency for each resource, selecting the highest-performing resource. An unfortunate reality of cloud systems is performance variation in seemingly identical resources: often this is due to a heterogeneous hardware at the cloud provider (e.g. a mix of AMD and Intel processors of varying generations) but performance can also be impacted by ”noisy neighbours”. Our system identifies under performing resources whose performance is below the expected range for that resource 595	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	Frames per dollar (efficiency) over 5 day period class (or whose performance level can now be bettered by another resource) and swaps it for a faster resource. By taking a cost-based approach we enabled our system to automatically migrate transcode components around the world to the cheapest compute resources at any given time.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Cost	It is clear that over time the cloud and cloud bursting solutions cost less than the fixed resources.	Ivon Miranda Santos
Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Cost	In terms of total cost the all cloud solution was lowest at $122, followed by the cloud bursting solution at $172 and the fixed infrastructure cost $316.	Ivon Miranda Santos

Wright2011-A_commodityfocused_multi-cloud_marketplace_exemplar_	CLOUD MANAGEMENT > Performance	Indeed we migrated our own private cloud from OpenNebula to Eucalyptus without changing the application at all–a significant saving in time and effort. Many of our applications require high compute and stor-age performance and thus the correct selection of resources is important. The capability to select resources based on service requirements and (more recently) on historical data on what is actually delivered by a provider has proven invaluable. In addition, the capability to define scenarios and define cost models to compare performance based on these models has delivered tangible performance benefits to our application and enabled an application focused choice. Thus, for example, we can balance the CPU speed required by our transcoding services to the bandwidth at a provider and the storage infrastructure performance to enable a practical choice for a media streaming service location. In our experience what is actually delivered by provider resource can differ significantly from that advertised by a provider and having access to historical data has proven invaluable in choosing the most appropriate resources.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	Cost Reduction in Hybrid Clouds for Enterprise Computing	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Performance	This benefit mainly stems from the cloud's economies of scale and the buy-on-demand model of cloud computing. However, migrating the entire enterprise application to public cloud may introduce issues in security, performance and reliability [1]. In response to these concerns, several solutions based on hybrid cloud infrastructures, which involve both on-premise cloud and public cloud, have been proposed.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	Since we consider a static scenario in this paper, the operation cost benefit of each server vi using each public cloud	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	Thus the total operation cost reduction with migration policy x can be represented as	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	In addition to the operation cost, the Internet communication cost is another financial cost relevant to server migration.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	denote the Internet communication cost increment of migration server vi and server vj to two clouds due to the second reason.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	Thus the Internet communication cost increase with the migration x can be formulated as	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	The total cost reduction with the migration x is the total operation cost reduction minus the total Internet communi-cation cost increment.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	The application and four migration plans with the corresponding cost reduction and completion time.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	The unit Internet communication costs of the three clouds are the same.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	We adopt the Amazon EC2 cloud pricing [12] to calculate the cost of running a server in each cloud.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	The unit communication cost and the delay between each cloud are 0.1 and 1, respectively.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	Cost reduction is defined as the ratio of the total cost reduction after migration and the total cost before migration.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Performance	3(b) show that by varying the value of time constraint, one can obtain a large cost reduction with large time overheads. Choosing a proper value for an application depends on the performance requirement of the application manager. The effect of user location.	Ivon Miranda Santos

Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	shows that migrating applications that have larger percentage of external user to cloud will bring more cost reduction than migrating the ones that have smaller percentage of external user.	Ivon Miranda Santos
Zhou2017-Cost_reduction_in_hybrid_clouds_for_enterprise_computi	CLOUD MANAGEMENT > Cost	In this paper, we study the problem of migrating enterprise applications to hybrid cloud for cost benefits maximization.	Ivon Miranda Santos
Wolfswinkel2013-Using_grounded_theory_as_a_method_for_rigorousl	CLOUD MANAGEMENT > Performance	// absolventen.at. Her research appeared in various international journals such as the Journal of Service Management, International Journal of Technology and Human Interaction, Human Systems Management, European Journal of Information Systems, European Journal of Manage-ment, Team Performance Management; she currently serves on the Editorial Board of the British Journal of Manage-ment. Furtmueller, regularly presents her research at the European Conference on Information Systems, Interna-tional Conference on Enterprise Information Systems, Academy of Management Meetings and Conference on Business Information Systems.	Ivon Miranda Santos