



HAL
open science

Are Cloud Platforms Ready for Multi-cloud?

Kyriakos Kritikos, Pawel Skrzypek, Feroz Zahid

► **To cite this version:**

Kyriakos Kritikos, Pawel Skrzypek, Feroz Zahid. Are Cloud Platforms Ready for Multi-cloud?. 8th European Conference on Service-Oriented and Cloud Computing (ESOCC), Sep 2020, Heraklion, Crete, Greece. pp.56-73, 10.1007/978-3-030-44769-4_5 . hal-03203298

HAL Id: hal-03203298

<https://inria.hal.science/hal-03203298v1>

Submitted on 20 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Are Cloud Platforms Ready for Multi-Cloud?

Kyriakos Kritikos¹ and Paweł Skrzypek² and Feroz Zahid³

¹ ICS-FORTH, Crete, Greece
kritikos@ics.forth.gr

² AI Investments, Skierniewice, Poland
pskrzypek@aiinvestments.pl

³ Simula Research Laboratory, Fornebu, Norway
feroz@simula.no

Abstract. Multi-cloud computing is getting a momentum as it offers various advantages, including vendor lock-in avoidance, better client proximity and application performance improvement. As such, various multi-cloud platforms have been developed, each with its own strengths and limitations. This paper aims at comparing all these platforms to unveil the best one as well as ease the selection of the right platform based on the user requirements and preferences. Further, it identifies the current gaps in the platforms to be covered so as to enable the full potential of multi-cloud computing. Finally, it draws directions for further research.

1 Introduction

Cloud computing promises the on-demand delivery of infrastructural and other kinds of services to assist in the applications development, deployment and adaptive provisioning. Further, it promises the reduction of costs, flexibility in resource management plus the ability to supply a potentially infinite amount of resources. As such, it has been widely adopted, leading to a multitude of applications being migrated to the Cloud. However, the initial Cloud computing platforms and providers offered services that encouraged vendor lock-in while these services performance was not always as expected or promised.

To this end, multi-cloud computing popped up [1], promising to address the above issues. This computing kind enables applications to be deployed in multiple clouds, one at a time. This allows not only to avoid vendor lock-in but also to achieve customer proximity via application spreading across different physical locations. Further, it enables application providers to select more reliable cloud providers in terms of the service level being delivered by their services. In addition, it allows to better satisfy application requirements and preferences, leading to applications with improved performance. A certain form of multi-cloud computing, cross-cloud computing, [1,2] has been also introduced, promising to deploy applications each time in not one but multiple cloud providers. This has the inevitable advantage of achieving true optimality, as application developers can select the best cloud services to realise their applications' functionality.

Based on the above analysis, multi-cloud computing is now getting a momentum such that a multitude of multi-cloud management platforms (MCMPs)

have been developed. Such MCMPs are either extensions of existing platforms (see Google Anthos⁴) or new ones that rushed to cover the respective market gap. These MCMPs have their own strengths and weaknesses. For instance, one MCMP can support a diversity of clouds while another deployment automation. Thus, users find it difficult to select the MCMP best suiting their needs.

As such, this paper attempts to review all these MCMPs by adopting an hierarchical criteria set, organised according to three main aspects: (a) how well the orchestration of multiple clouds is supported; (b) what is the support level for multi-cloud applications; (c) what is the MCMP intelligence extent for increasing the automation in application provisioning. All these aspects complement each other and together lead exploiting the full potential of multi-cloud computing.

The MCMP review based on the aforementioned set fulfils multiple purposes: (a) it enables to nominate one MCMP as the best; (b) it facilitate MCMP selection based on user requirements and preferences; (c) it unveils those functional gaps in MCMPs which need to be covered to fully support multi-cloud computing. Another contribution is the supply of interesting challenges, which, when addressed, can further boost the MCMP adoption.

The rest of the paper is structured as follows. The next section explains the MCMP selection process. Section 3 introduces the hierarchical set of criteria which can be regarded as multi-cloud computing goals. Section 4 presents and analyses the evaluation results. Finally, the last section 5 concludes the paper and supplies some research challenges for further boosting the MCMP adoption.

2 Platform Selection Process

We aimed at assessing only proprietary MCMPs, as these are complete products which can be actually exploited by users in the cloud market. To this end, our simplified MCMP selection process involved the following two main steps:

MCMP Search. To conduct the MCMP search, we have relied on: (i) using sophisticated web search engines (e.g., Google) by applying the following query string: “Multi-cloud AND Platform”; (ii) the snowball crawling method [3] where forwarding links are visited from the main web site currently inspected – useful in case the web site was suggesting or evaluating multiple from these platforms / tools; (iii) our own knowledge about some well-known MCMPs. While conducting (i) and (ii), we also came across two articles from Forrester [4] and Gartner [5] which enabled us to both identify MCMPs and validate those already discovered.

Filtering. Apart from the fact that the MCMPs should be either proprietary or offered in dual licencing mode, we have applied additional selection criteria which included: (i) the MCMP should be still in operation; (ii) it should have existing clients; (iii) it should support multi-cloud and not just hybrid application deployment; (iv) finally, it should also support application reconfiguration.

In result, 17 MCMPs were discovered (see Appendix 1 in <https://tinyurl.com/rby7m37>). In fact, the MCMP market is overruled by US companies and only

⁴ <https://cloud.google.com/anthos/>

2 MCMPs reside in Europe and just one in Canada. Further, with the exception of Google, no other major cloud provider is offering a MCMP. A subset of these MCMPs (see Table 1), the top-7, will be reviewed in Section 4 based on the criteria identified in the next section due to paper length restriction reasons.

Table 1. Overview of assessed MCMPs

Platform	Location	Description
Cisco CloudCenter Suite ⁵	US	Multi-cloud management platform with special focus on cost optimisation and CI/CD.
Rackware Hybrid Cloud Platform ⁶	US	A multi-function solution for workload migration to the cloud, disaster recovery, and multi-cloud resource management.
MorpheusData ⁷	US	A multi-cloud management platform for hybrid IT and DevOps automation with special focus on cost and performance optimisation.
CloudBolt ⁸	US	A multi-cloud and hypervisor management platform featuring continuous infrastructure testing and blueprints for repeatable and standardised application deployment.
Google Anthos	US	Hybrid application management platform for on-premise and public clouds with support for service mesh, containers, micro-services and functions as well as strong focus on security and workload migration.
Cloudify ⁹	US	End-to-end modular orchestration platform that abstracts applications and networks from underlying infrastructures. Provides also support for edge computing.
Melodic ¹⁰	Poland	Cross-cloud, data-intensive application management platform with strong focus on utility- and model-driven application reconfiguration.

3 Requirements

To evaluate the selected MCMPs, a sophisticated evaluation framework in form of an assessment criteria hierarchy was devised. The main focus is on three main dimensions: (a) *cloud orchestration support* – the degree of cloud service orchestration support and automation even across multiple abstraction levels; (b) *cloud application support* – the degree and level of management for cloud applications; (c) *platform intelligence* – what knowledge is derived from intelligent mechanisms to optimise the multi-cloud application provisioning. In the following, we present all dimensions and their hierarchy of criteria in separate sub-sections.

3.1 C1 – Cloud Orchestration Support

This criteria category evaluates the capability of an MCMP to orchestrate cloud services, even coming from different providers and/or different abstraction levels. It also covers aspects like the possibility to orchestrate private clouds in hybrid deployment scenarios and the support to cloud-related standards.

C1.1 – Cloud Support This sub-category assesses the orchestration support that a certain MCMP has on different clouds.

C1.1.1 – Cloud Diversity. This criterion assesses the diversity of public clouds that can be supported. As such, the evaluation of this criterion on a MCMP can take the following values: (a) *low*: only two public clouds are supported; (b) *medium*: three to six clouds are supported; (c) *good*: six to nine clouds are supported; (d) *high*: ten or more clouds are supported by the MCMP.

C1.1.2 – Private Cloud. This criterion assess the capability of an MCMP to support application deployment in private clouds to cover hybrid application

provisioning scenarios. This capability can be assessed as follows: (a) *no*: it is not exhibited by an MCMP; (b) *yes*: it is indeed featured.

C1.1.3 – Cross-Cloud. As indicated in Section 1, cross-cloud computing enables to achieve true optimisation of application deployment. As such, an MCMP can be evaluated on this as follows: (a) *no*: the MCMP does not support cross-cloud computing; (b) *yes*: otherwise.

C1.2 – Resource Diversity Due to various factors, including better workload and security isolation, other deployment alternatives have been presented apart from Virtual Machines (VMs) like containers and specialised resources kinds (e.g., GPUs) which enable boosting the performance of compute-intensive applications. As such, the more resource kinds are handled, the more kinds of workloads can be supported. Thus, an MCMP can be evaluated as follows here: (a) *low*: only the traditional resource of a VM is supported; (b) *medium*: also containers are supported; (c) *good*: three to four resource kinds are supported; (d) *high*: more than four resource kinds are supported.

C1.3 – BYON Users might have existing resources hosted in specialised infrastructures or private clouds. Thus, if an MCMP is able to exploit and manage such resources, it can increase the alternative deployment options for applications plus cater for hybrid application provisioning scenarios. As such, it can be evaluated on this criterion as follows: (a) *no*: this BYON capability is not featured by the MCMP; (b) *yes*: the MCMP can support deployment on user-specific resources.

C1.4 – Service Support This criteria sub-category assesses the kinds of services supported by an MCMP and the support versatility per each handled service kind with the rationale that the diversity of services in different abstraction levels leads to covering more advanced multi-cloud application provisioning scenarios.

C1.4.1 – Service Kinds Versatility. This criterion assesses the different abstraction levels (i.e., IaaS, PaaS, SaaS & BPaaS) covered by an MCMP in terms of respective cloud services. The rationale is that higher abstraction levels coverage enables to reduce administration burden and cost while facilitates completing the functionality of multi-cloud applications. As such, an MCMP can be evaluated as follows: (a) *low*: one service kind is only supported; (b) *medium*: two to three service kinds are supported; (c) *high*: all service kinds are supported.

C1.4.2 – Service Versatility. This criterion attempts to assess the versatility of support for the same service kind to enable moving or selecting multiple services of the same kind in the context of multi-cloud application provisioning. C1.1 covered this for infrastructural but not higher-level of services. As such, an MCMP can be evaluated as follows: (a) *low*: only one service is supported for a higher-level service kind; (b) *medium*: two to five services are supported; (c) *high*: more than five services are supported.

C1.5 – Automation This criterion assesses the level of automation in cloud service orchestration. Ideally, it is expected that based on a model of the user application, an MCMP could derive and execute a concrete orchestration plan

to relieve users from manually specifying it and the burden to delve into cloud-specific details. As such, an MCMP can be evaluated as follows: (a) *low*: the orchestration is manually conducted by the user; (b) *medium*: some orchestration parts can be automated by the MCMP; (c) *good*: the MCMP can orchestrate a plan given by the user; (d) *high*: it can automatically derive and execute a plan based on the user input (e.g., an application model).

C1.6 – Standards The conformance to standards comes with various benefits, also related to multi-cloud computing, like the ability to operate software plus exchange information and knowledge across different technologies, platforms and infrastructures. Already multiple cloud standards exist spanning, e.g., how cloud applications can be specified (e.g., TOSCA [6]) and the format of VM images (e.g., OVF), although their adoption level by the providers is not as expected. As such, we can evaluate a MCMP on this as follows: (a) *no*: no cloud standard is supported; (b) *low*: one standard is supported; (c) *medium*: two to three standards are supported; (d) *high*: more than three standards are supported.

3.2 C2 – Cloud Application Support

An outmost MCMP goal is to support the multi-cloud applications' management, spanning their whole lifecycle. This support needs to be built under solid bases, like model-driven engineering, which abstract from low-level technicalities and increase the automation level. Applications could also process and produce data, which can be large in volume in many cases. As such, these data should be also managed accordingly by the MCMPs. Finally, an MCMP must allow for high-levels of management flexibility plus cater for the continuous application evolution. All these aspects are covered by the following categories of criteria.

C2.1 – Modelling Support An MCMP should provide the right support level for multi-cloud application modelling, spanning the supply of a modelling framework, the capturing of multiple component configuration kinds, the rich resource specification plus the re-use of existing model parts for new application models.

C2.1.1 – Modelling Language. An MCMP should use a modelling language for the specification of multi-cloud applications which is rich enough to cover all relevant application management aspects (e.g., deployment, monitoring) while providing the right abstraction level. As such, an MCMP can be evaluated as follows: (a) *no*: no modelling language is offered; (b) *low*: it is offered but covers only one aspect; (c) *medium*: the MCMP's modelling language covers two to three management aspects; (d) *high*: it covers all necessary aspects;

C2.1.2 – Configuration Diversity. To support true multi-cloud computing, application components in one form need to be coupled with different cloud-specific configuration specifications. Further, the same functionality/component can take different forms (e.g., functions or big data processing tasks), each coming with its own configuration alternatives. Thus, an MCMP must support as many of these configuration kinds as possible to increase its applicability level. So, it can be evaluated on this criterion as follows: (a) *low*: the MCMP supports

only one configuration kind; (b) *medium*: the MCMP supports two or three configuration kinds; (c) *high*: more than three configuration kinds are supported.

C2.1.3 – Resource/Service Modelling . Cloud service offerings must be modelled to support matching service requirements and capabilities and the selection of the best possible services. This modelling is also imperative to realise BYON scenarios. However, it needs to cover all service kinds across multiple clouds. As such, an MCMP can be evaluated as follows: (a) *no*: the MCMP does not support service modelling; (b) *low*: it features a language via which only one service kind can be modelled; (c) *medium*: the language covers two service kinds; (d) *high*: the language covers the modelling of any service kind.

C2.1.4 – Policy Modelling & Enforcement. User organisations can have non-functional requirements and policies that should hold for any of their applications or for specific ones. For instance, due to privacy low conformance reasons, any data manipulated by any application must reside in its origin country. Further, an application’s response time could be restrained to be lower than an upper bound. Besides, different requirement kinds might need to be specified, like location, scaling, and quality requirements. Thus, an MCMP can be evaluated as follows: (a) *no*: it does not support requirement modelling; (b) *low*: only one requirement kind is supported; (c) *medium*: two to three requirement kinds can be modelled; (d) *good*: more than three kinds can be specified; (e) *high*: same as (d) but requirements are enforced at both the global and application level.

C2.1.5 – Editing Capabilities. Cloud modelling languages mostly adopt representation formats like YAML or JSON coming with their own generic editors. However, language-specific editors must be used instead so as: (a) to cover the modelling preferences and habits of the main target users (devops); (b) to have specialised language features (e.g., error highlighting, auto-completion) that enhance user experience and speedup the modelling process. Thus, an MCMP can be evaluated as follows: (a) *no*: no modelling language is supplied; (b) *low*: no editor is offered but generic editors exist for the language representation formats; (b) *medium*: a language editor exists with no special features; (c) *high*: a sophisticated editor is offered with the aforementioned added-value features.

C2.1.6 – Model Re-Use. Model re-use is essential to not only allow the rapid creation of language models but also to reduce the learning curve. It can also lead to a community-lead effort where both complete multi-cloud application models plus model fragments covering common elements like metrics become available. As such, we evaluate a MCMP as follows: (a) *no*: no model-reuse is allowed; (b) *low*: complete models can be re-used by just copying their parts; (c) *medium*: whole models and some complex model elements can be re-used/cross-referenced; (d) *high*: any kind of model element can be re-used/cross-referenced.

C2.2 – Lifecycle Management Suitable support to any application management activity (e.g., design and deployment) reduces the management burden and enables devops to focus more on the core application functionality and its improvement. As such, we can assess an MCMP as follows: (a) *low*: only one activity (usually deployment) is covered; (b) *medium*: two to three activities are covered; (c) *good*: four to five activities are covered; (d) *high*: all activities are supported.

C2.3 – Data Management (Big) Data are becoming a critical asset for organisations which strive to derive added-value knowledge from them to improve existing products and services. Due to their importance, they need to be properly managed to increase the automation level in data manipulation and reduce operational costs (e.g., by moving computation near data to save communication cost). With this criteria sub-group, we assess the pure data management activities offered by an MCMP.

C2.3.1 – Data Creation. Data sources can already exist outside the application management scope. However, in other cases, data sources and their encapsulated data must be created before a specific application can be deployed and executed. To this end, an MCMP can be evaluated as follows: (a) *no*: no data (source) creation facility is offered; (b) *yes*: otherwise.

C2.3.2 – Secure Data Migration. Data are not always bound to a certain place and might require to be moved for various reasons; e.g., the level of availability of the node currently storing them has been greatly reduced. However, data migration is not a simplified activity while it needs to be properly secured for privacy and protection reasons. As such, an MCMP can be assessed as follows: (a) *no*: data migration is not supported; (b) *medium*: only insecure data migration is supported; (c) *high*: secure data migration is fully supported.

C2.3.3 – Secure Data Destruction. While output data can be moved to a new place or intermediate data can be deleted, a new user of the same host or the cloud provider could still read them. As such, for security and privacy reasons, an MCMP must securely destruct data when they are no more needed and can be, thus, evaluated as follows: (a) *no*: data are simply destructed; (b) *yes*: data are securely destructed by utilising state-of-the-art mechanisms.

C2.3.4 – Data Source Versatility. Data can be encapsulated in different data source kinds, including relational or NoSQL DBs, and triples stores. Thus, data management should be conducted across all such kinds. As such, an MCMP can be evaluated as follows: (a) *no*: no data source is supported; (b) *low*: only one data source kind is supported; (c) *medium*: two to three data source kinds are supported; (d) *high*: the MCMP supports more than three data source kinds.

C2.4 – Workflow Support With this sub-group of activities, we assess the support level towards management and application workflows.

C2.4.1 – Management Flexibility. In many cases, MCMPs encode the management of application lifecycle activities in code form. This is inflexible as each time the handling of such activities must be modified, the code must be re-engineered. However, if lifecycle activities are encoded in a workflow form, the highest possible flexibility is achieved as workflows can be easily and effortlessly modified by using well-known workflow editors without changing any kind of code. Thus, we can evaluate an MCMP with: (a) *no*: management workflows are not directly supported; (b) *yes*: such workflows can be both edited and executed.

C2.4.2 – Workflow Automation. Applications do not take just the form of component agglomerations. They could also be BPaaS realised in form of workflows that must be properly provisioned and managed. Thus, we can assess

an MCMP as follows: (a) *no*: it does not support application workflows; (b) *yes*: it offers specialised workflow facilities and services.

C2.5 – Containerization By adopting the well-known and used micro-service paradigm, modern cloud applications are built and deployed as containers which can be easily managed in separation. With the following two criteria, we assess the support level towards building and managing a diversity of container forms.

C2.5.1 – Container Versatility. Docker is the most widely used container form. However, new container forms were recently developed (e.g., singularity [7]), which are more secure and lightweight, specializing for particular workload kinds (i.e., high performance ones). Thus, an MCMP must support handle all these container forms during application deployment to increase its applicability. So, it can be evaluated as follows: (a) *no*: containers are not supported; (b) *low*: only one container form is supported; (c) *medium*: two container forms are supported; (d) *high*: the MCMP handles more than two container forms.

C2.5.2 – Container Image Management. An MCMP must offer suitable facilities to manage container images; this is handy in the context of manipulating and building containerised application components. Thus, we can evaluate it as follows: (a) *no*: container images are not managed; (b) *yes*: the MCMP offers tools or facilities to manage container images.

C2.6 – CI/CD Support Applications can evolve over time due to market competition, requirements change and the advent of new technologies. Thus, an MCMP must support this evolution by adopting different continuous integration and deployment (CI/CD) paradigms. For instance, it could allow old and new application versions to be concurrently provisioned. As such, an MCMP can be assessed as follows: (a) *no*: CI/CD is not supported; (b) *low*: it is not directly supported but users can push new application versions for deployment; (c) *medium*: it is directly supported; (d) *high*: it can also be properly configured.

3.3 C3 – Platform Intelligence

An MCMP must derive extra knowledge to optimise the client’s application in terms of its initial deployment and its continuous reconfiguration. The ability to derive new knowledge plus exploit it relates to monitoring & runtime capabilities covering the user application’s functionality and data. As such, respective criteria categories were devised to assess MCMPs with respect to such capabilities.

C3.1 – Optimisation This criterion assesses how well a multi-cloud application is continuously optimised by an MCMP.

C3.1.1 – Utility Functions. Optimisation is usually accompanied by utility functions to derive a candidate solution’s utility. This enables checking the solution space to discover the solution with the highest utility, i.e., the optimal one. The formulation of such utility functions must be rich and precise via the use of complete mathematical specifications. As such, an MCMP can be evaluated as follows: *no*: it does not consider any optimisation objective; *low*: the objective

is rather fixed; *medium*: the objective is indirectly produced via the user specification of partial objectives (e.g., metrics) and their relative importance (in form of weights); *high*: a complete mathematical formula can be expressed.

C3.1.2 – Objective Versatility. Even if an optimisation objective is mathematically defined, we argue that apart from mathematical operators and functions, its content should include as variables: (a) *metrics*: these connect with the monitoring feedback to properly compute a solution’s utility; (b) *attributes*: they denote some quantities related to the solution space or the current application configuration. For instance, one attribute could express the maximum cost among all candidate offerings of an application component. Thus, an MCMP can be evaluated as follows: *no*: metrics and attributes cannot be used in objective specification; *low*: only fixed metrics can be used; *medium*: also fixed attributes can be used; *high*: any kind of metric or attribute can be involved.

C3.1.3 – Continuous Reasoning. An MCMP must continuously reason to derive new application deployment plans that more optimally address the current situation. This is an essential ability in the dynamic cloud environments under which applications operate (as various functional and non-functional faults can occur) while allows grabbing new opportunities for optimisation when they appear (e.g., better offerings for application components). As such, an MCMP can be evaluated with: *no*: reasoning is performed once for initial application deployment; *medium*: it is also performed at runtime for very limited occasions (e.g., SLO violations); *high*: deployment reasoning is continuously performed.

C3.2 – Monitoring This criterion assesses an MCMP’s monitoring features, including the monitoring and aggregation of any kind of metric plus the proper configuration of such measurement.

C3.2.1 – System Metrics. By allowing the automatic computation of some metrics, the client’s development effort in terms of application monitoring is reduced. This signifies the need to support system metrics across different abstraction levels. Thus, an MCMP is evaluated with: *no*: no system metric is offered; *low*: only infrastructural metrics are covered; *medium*: metrics at two abstraction levels are offered; *high*: system metrics cover all possible levels.

C3.2.2 – Custom Metrics. An MCMP cannot realise any metric kind, especially domain-specific ones, so clients must be able to inject their own metrics in the MCMP. Such an injection must be properly performed by defining sensor components within the application model and subsequently installing them in the application’s deployment infrastructure. Thus, an MCMP can be evaluated on this criterion with: *no*: no custom metrics can be exploited; *low*: only external sensors (to the application deployment infrastructure) can be utilised; *medium*: also internal sensor components can be indirectly incorporated but the devops must properly install them in the deployment infrastructure and integrate them with the MCMP’s monitoring sub-system; *high*: internal sensor components are properly installed and integrated with the MCMP’s monitoring sub-system.

C3.2.3 – Metric Aggregation. Application requirements rely on high-level measurements as aggregations of raw-level ones. Such an aggregation must be computed via mathematical formulas that apply basic or statistical functions

over metrics and attributes. This enables to compute any kind of composite metric, mapping to the outmost monitoring flexibility. Thus, an MCMP can be evaluated with: *no*: measurement aggregation is not supported; *low*: only fixed statistical functions can be applied on raw measurements; *medium*: mathematical expressions using fixed statistical functions can be expressed; *high*: any kind of mathematical expression including any kind of function can be utilised.

C3.2.4 – Metric Configuration. Metric measurement should be properly configured based on: the measurement schedule, i.e., how often to conduct it; the measurement window, i.e., how many measurement values are used to compute the aggregated measurement. While such information could be specified by users, this may lead to situations where mistakes can be performed or the monitoring sub-system becomes overloaded. Thus, an MCMP must optimally derive this information by finding the best trade-off between monitoring accuracy and performance. So, it can be evaluated with: *no*: it does not allow specifying such information and cannot derive it; *low*: it has a fixed mapping of metrics to schedules and windows based on their type (raw or composite); *medium*: it allows the user to specify this information; *high*: it automatically derives this information by respecting user preferences (e.g., bounds on both information pieces).

C3.3 – Runtime Adaptation An MCMP must globally adapt the user application according to the current situation and across multiple abstraction levels.

C3.3.1 – Scaling Support. Local application reconfiguration can be conducted in less critical situations through application (component) scaling. Different scaling kinds (horizontal and vertical) must be supported to cover different local reconfiguration scenarios. As such, an MCMP can be evaluated on this criterion as follows: *no*: scaling is not supported at all by the MCMP; *medium*: only horizontal scaling is supported; *high*: also vertical scaling is supported.

C3.3.2 – Global Reconfiguration. In more critical situations, single application component scaling does not suffice; a more radical application reconfiguration must be performed, even across multiple clouds, which can be guided through utility functions for discovering the best possible application deployment solution. As such, an MCMP can be evaluated as follows: *no*: it does not support global reconfiguration; *yes*: it does support it.

C3.3.3 – Higher-Level Adaptation. Problematic situations relate to issues which occur in one abstraction level and can be propagated up to higher levels. To this end, a workflow of level-specific adaptation actions must be executed in a coordinated manner to avoid conflicting action effects. As such, we can evaluate an MCMP on this as follows: *no*: it does not offer higher-level adaptation actions; *low*: adaptation actions at two levels are offered; *medium*: adaptation actions in all possible levels are supplied; *high*: workflows of level-specific adaptation actions can be specified to address problematic situations.

C3.4 – Event Management Problematic situations need to be detected before they can lead to triggering any kind of adaptation action. Thus, any kind of event needs to be captured, to cover all possible situation kinds, irrespectively of whether it is within the MCMP’s control sphere or not.

C3.4.1 – Event Versatility. As different kinds of events might occur in an application system spanning both functional and non-functional aspects, an MCMP must capture them all. As such, an MCMP can be assessed on this as follows: *no*: it cannot capture any event kind; *medium*: it is able to capture either functional or non-functional events; *high*: it can capture both event kinds.

C3.4.2 – Complex Events. In many cases, a problematic situation cannot be characterised by a single but multiple events, related to each other in different ways. This requires capturing hierarchical compositions of events on which time and logic-based operators can be applied. Thus, an MCMP can be evaluated with: *no*: composite events cannot be captured; *low*: only time or logical operators can be used in the event composition; *medium*: a limited set of both operator kinds can be used; *high*: a rich operator set of both kinds can be used.

C3.4.3 – External Event Integration. A diversity of applications might receive from as well as propagate events to their environment. In the context of cloud & edge computing, the environment could comprise sensors or external services like message queues. Thus, all such (external) entities must be properly integrated with an MCMP in a provider-independent way to support multi-cloud application deployment. So, an MCMP can be evaluated with: *no*: integration with external services is not possible; *medium*: there is a provider-specific integration; *high*: external event integration is realised across multiple providers.

C3.5 – Data Management Data may not be just consumed by an application but may have to be migrated for various reasons (e.g., they must move towards external sources). As such, data must be properly maintained and accounted for in application deployment optimisation.

C3.5.1 – MetaData Maintenance. To control and manage data, there is a need to support their continuous characterisation in terms of metadata. Such metadata could cover static information about the data like what is their origin plus dynamic information like their current size. As such, an MCMP can be evaluated as follows: *no*: metadata are not kept at all; *medium*: only static metadata are maintained; *high*: both static and dynamic metadata are preserved.

C3.5.2 – Data Monitoring. Dynamic metadata observation and data quality measurement require suitable data monitoring mechanisms, independent of the data source and format to increase an MCMP’s applicability in handling different data-intensive application kinds. Thus, an MCMP can be evaluated with: *no*: data monitoring is not supported at all; *medium*: data monitoring is supported but only for certain data sources and/or formats; *high*: the MCMP features a data-source and format-independent data monitoring mechanism.

C3.5.3 – Data-aware Optimisation. Data placement could have a tremendous effect on both application cost and performance as well as the component placement. Thus, there is a need to support data-aware optimisation of application deployment. This can be realisable by: (a) considering resource, platform and other kinds of constraints for the nodes on which data should be placed; (b) incorporating data-related metrics in the optimization objective(s). As such, an MCMP can be evaluated with: *no*: it does not support data-aware optimisation;

medium: it only accounts for constraints on data nodes; *high*: it also incorporates data-related metrics in the optimisation objectives.

C3.6 – Dynamic Resource Offering Discovery Due to the dynamicity in cloud environments plus the periodic upgrading of cloud offerings, cloud services might go down or cease to exist. Thus, if these services' offerings are not properly observed, an MCMP might propose deployment solutions that do not work or are not optimal. Further, cloud providers do not advertise all possible information for such services, especially their non-functional capabilities. As such, there is a need for mechanisms (e.g., benchmarking, monitoring) that enhance the service performance profiles with the missing knowledge. Thus, an MCMP can be evaluated with: *no*: it does not support dynamic resource offering discovery; *medium*: it does support this but cannot derive extra performance-oriented knowledge; *high*: both dynamic resource offering discovery and enhancement are supported.

4 Evaluation

Before we supply and analyse the evaluation results, we first explain how they were produced, especially in terms of aggregating the raw assessment values per each individual criterion. To this end, the following approach has been adopted:

- Per each individual criterion, the qualitative values are mapped to quantitative ones. As such, the following mappings were introduced:
 - *no* → 0.0
 - *low* → 1.0
 - *medium* → 3.0
 - *high* → 5.0
- we aggregate the evaluation results via a weighted sum in the two higher hierarchy levels. First, aggregation is conducted on the composite criteria level, where each individual criterion gets the same weight (same relative importance with the others). Then, aggregation is performed on the global dimensions level, where each component criterion gets again the same weight.
- Finally, for the global level the aggregated quantitative values are mapped to qualitative ones as follows:
 - [0.0, 1.0) → *very low*
 - [1.0, 2.0) → *low*
 - [2.0, 3.0) → *medium*
 - [3.0, 4.0) → *good*
 - [4.0, 5.0] → *high*

For presentation purposes, the above mapping is also applied to composite criteria as we regard that qualitative values better represent evaluation results in the eyes of the prospective reader. Please also note that due to paper length restriction reasons, the evaluation of all individual criteria is not shown (please follow the URL in the last paragraph of Section 2 to view the complete results).

Table 2. Overall Evaluation Results

MCMP	C1 – Cloud Orchestration Support	C2 – Cloud Application Support	C3 – Platform Intelligence
Cisco Cloud Center Suite	good	medium	medium
Rackware Hybrid Cloud Platform	good	very low	low
Morpheus Data	good	medium	low
CloudBolt	good	low	low
Google Anthos	good	medium	medium
Cloudify	good	medium	low
Melodic	good	medium	good

4.1 Overall Results

Table 2 depicts the evaluation results at the global level, indicating that Melodic is the best MCMP as it reaches a good score level for two out of three dimensions. Melodic is then followed by Google Anthos and Cisco Cloud Center Suite, which have attained medium score levels for two dimensions and a good level for one. Finally, Rackware Hybrid Cloud Platform seems to be the least performant, focusing mainly on supporting just the first from the three dimensions.

4.2 Cloud Orchestration Support

Table 3 depicts the overall results for this dimension which are quite encouraging as all MCMPs seem to attain a good score level. Further, there is some evaluation uniformity as most MCMPs seem to attain the same score for each criterion in the second hierarchy level (the intermediate one). Exceptions to this rule can be observed in the last three criteria of that level.

Table 3. Cloud Orchestration Support Evaluation Results

MCMP	C1.1	C1.2	C1.3	C1.4	C1.5	C1.6
Cisco Cloud Center Suite	good	good	yes	good	good	no
Rackware Hybrid Cloud Platform	high	good	yes	medium	good	no
Morpheus Data	high	good	yes	medium	good	no
CloudBolt	high	good	yes	medium	good	no
Google Anthos	high	good	yes	good	good	no
Cloudify	high	good	yes	medium	good	low
Melodic	high	good	yes	medium	high	no

In particular, for the service support, Google Anthos and Cisco Cloud Center Suite attained a good score due to their ability to support a multitude of services at the platform level. This makes them ideal for applications requiring multi-cloud support crossing multiple abstraction levels. Melodic is distinguished in the automation criterion as it can derive and execute automatically an application deployment plan. All other MCMPs can just execute a user-supplied plan.

Much improvement is needed for the standards support as only one MCMP just supports one standard, the TOSCA [6] cloud application modelling language.

Finally, we should remark the high support level for the first dimension criterion, which, in conjunction with the second one, highlights the main focus of MCMPs towards supporting the infrastructure level over a great diversity of cloud providers. In addition, cross-cloud support (not visible in Table 3) should be also highlighted. As such, we can conclude that the support level for this dimension is satisfactory but there is a need for further improvement in some criteria, especially the fourth and sixth ones (service support & standards).

4.3 Cloud Application Support

Table 4 depicts the overall results for this dimension which can be regarded as moderate such that much improvement is needed. This also unveils the current focus of MCMPs on supporting cloud orchestration and not providing any kind of additional support to cloud applications. Compared to the first dimension, there is also a high diversity of evaluation results per criterion.

Table 4. Cloud Application Support Evaluation Results

MCMP	C2.1	C2.2	C2.3	C2.4	C2.5	C2.6
Cisco Cloud Center Suite	good	low	medium	high	very low	medium
Rackware Hybrid Cloud Platform	low	low	medium	no	no	no
Morpheus Data	medium	low	low	medium	very low	high
CloudBolt	medium	low	low	medium	very low	low
Google Anthos	medium	low	good	no	good	high
Cloudify	good	low	low	medium	very low	high
Melodic	high	low	medium	medium	very low	low

Concerning the dimension criteria in the intermediate level, the worst performance is over the lifecycle management and the containerisation ones. For the first, this result seems logical as the MCMP focus is mainly on a few activities in the lifecycle (the deployment and provisioning ones). For the second, the result highlights that MCMPs expect that devops exploit already existing, external tools while Docker is the current container form & technology widely adopted. This indicates that other container forms, more suitable for high-performance computing (HPC) like Singularity [7], are not currently supported.

We now concentrate on the rest of the criteria where usually one MCMP prevails. For the modelling support, Melodic achieves the highest possible score as it adopts CAMEL [8], a rich multi-cloud application modelling language, covering multiple aspects related to the application lifecycle. Melodic obtains the highest score for all modelling support criteria apart from the policy modelling one as CAMEL concentrates on defining application-specific policies. More details about the way CAMEL covers multi-cloud computing can be found in [9].

Google Anthos obtains the highest score in data management as it features data creation and secure data migration while exhibiting a medium data source diversity. On the other hand, Cisco Cloud Center Suite achieves the highest score in workflow support as it enables both to edit management workflows and execute application workflows by using specific (PaaS-based) workflow engines.

Finally, it seems that CI/CD support has been well recognised as multiple MCMPs supply CI/CD capabilities to devops. This then enables to rapidly deploy new versions of multi-cloud applications to achieve fast time-to-market, an essential characteristic to survive in a very competitive business world.

4.4 Platform Intelligence

Melodic prevails in this dimension (see Table 5). This is due to its core ability to support the continuous optimisation of multi-cloud applications at the global level which presupposes appropriate support also to application monitoring and event management as the cornerstones for adaptive application provisioning. The rest of the MCMPs only support application component scaling. Melodic is superior also in data management where it supports maintaining metadata about the data manipulated by applications while it is the only MCMP accounting for (big) data features and optimisation objectives during application deployment reasoning. Lastly, Melodic is the only MCMP allowing the complete, mathematical specification of both metrics and optimisation objectives, thus covering essential measurability gaps and achieving true optimality by considering multiple optimisation objectives, spanning traditional cost & performance-oriented, data-specific and reconfiguration-specific objectives.

Table 5. Platform Intelligence Evaluation Results

MCMP	C3.1	C3.2	C3.3	C3.4	C3.5	C3.6
Cisco Cloud Center Suite	low	low	low	medium	good	medium
Rackware Hybrid Cloud Platform	no	low	low	low	good	no
Morpheus Data	low	medium	low	low	no	medium
CloudBolt	no	low	low	medium	no	medium
Google Anthos	no	medium	low	good	good	no
Cloudify	no	medium	low	good	medium	no
Melodic	high	medium	medium	good	high	medium

We now concentrate on the criteria requiring more improvement than the rest. In the monitoring (composite) criterion, the medium level reached by some MCMPs is related to the inability to support system metrics on all abstraction levels and the automatic derivation of metric configuration knowledge. From these MCMPs, Melodic is distinguished as it enables specifying any metric kind due to the use of a rich mathematical language while Google Anthos as it enables incorporating user sensors in a cloud application’s monitoring infrastructure.

For the runtime adaptation, no MCMP supports adaptation based on higher-abstraction-level actions. However, Melodic is distinguished as it supports global reconfiguration while Google Anthos prevails in terms of scaling support as it features both vertical and horizontal scaling of containerised components.

Finally, more than half of the MCMPs can dynamically discover cloud service offerings via cloud-specific APIs. However, they cannot also derive these services' non-functional profile. While the latter could have been an important knowledge to assist in selecting the right cloud services for a multi-cloud application.

5 Conclusions & Challenges

Due to the advantages that multi-cloud computing offers, various multi-cloud management platforms (MCMPs) have been developed, each with its own advantages and weaknesses. This paper attempted to review all these MCMPs based on a hierarchical criteria framework covering the dimensions of cloud orchestration support, cloud application support and platform intelligence. The review relied on a two-level aggregation approach over the assessment of individual criteria at the lowest hierarchy level. In this aggregation, equal weights were given to the criteria to signify their equal relative importance. This can be altered to better reflect the individual preferences of different users.

The evaluation results clearly showed that there is one MCMP that prevails, namely Melodic, providing the unique feature in terms of platform intelligence of continuous, global multi-objective application optimisation. They also showed which are those criteria requiring better support across all MCMPs, like those of monitoring and runtime adaptation as well as application lifecycle support.

We believe that the next challenges will further boost MCMPs' adoption.

BPaaS Management. MCMPs focus on traditional multi-tier applications. However, business processes (BPs) are currently moving to the cloud and require complete management support. Prototypes like CloudSocket [2,10] demonstrate that it is feasible to support such a management across multiple clouds and levels. However, there is still a need for further automation while the known business-to-IT gap must be better closed. Apart from this, technological developments like computational capability enhancement and serverless computing make more eminent the need to support a new breed of (multi-)cloud-based BPs (so called BPaaS) which are adaptive and flexible as well as incorporate a mixture of different tasks kinds (service-, function-, user- and analytics-based).

Polymorphic Applications. As indicated in Section 3, components might have cloud-specific configurations in one form but could also appear in different forms (e.g., micro-services, functions). Thus, the configuration diversity is quite large. Further, different component forms might exhibit different service levels when encountering the same workload. This requires not only to specify but also select and configure the right form of a component depending on user requirements. Such a selection could be also conducted either at design or even at runtime, leading to the era of polymorphic components and applications. In

our view, this represents the next breed of multi-cloud applications which will be adaptive to both their context and requirements, whenever these change.

High-Performance Multi-Cloud Computing. Traditional HPC infrastructures lead to long turnaround delays, especially in cases of high load. Recent work [11] identified that cloud computing resources can be used instead to attain a similar service level with a reduced cost and turnaround time. To further satisfy the ever increasing thirst for more computing power for heavy analytics tasks, the need to move HPC workloads in the cloud is recognized by some providers who rush to supply specialised and high-performant resources (e.g., GPUs and virtualised FPGAs) plus ultra-fast networks to provide suitable compute and networking support for HPC applications. This trend must be well covered by MCMPs to provide the right abstraction means enabling to properly select HPC-compatible resources that best fit application requirements. This also requires to properly specify this resources kind by the MCMPs' modelling languages [9] to support their matching with component requirements. Besides, MCMPs must support different container forms, like Singularity [7], stated as superior to Docker container forms, especially for HPC applications.

Acknowledgements. This has received funding from European Unions Horizon 2020 programme under grant agreement No 731664 (MELODIC)

References

1. Petcu, D.: Multi-cloud: Expectations and current approaches. In: MultiCloud, Prague, Czech Republic, ACM (2013) 1–6
2. Kritikos, K., Zeginis, C., Griesinger, F., Seybold, D., Domaschka, J.: A Cross-Layer BPaaS Adaptation Framework. In: FiCloud, Prague, Czech Republic, IEEE Computer Society (2017) 241–248
3. Goodman, L.A.: Snowball sampling. *Ann. Math. Statist.* **32**(1) (03 1961) 148–170
4. Nelson, L.E., O'Donnell, G., Caldwell, J., Reese, A.: The Forrester WaveTM: Hybrid Cloud Management, Q2 2018. Technical report, Forrester Research (2018)
5. Cheung, M., Fletcher, C., Byrne, P., Smith, D.: Magic Quadrant for Cloud Management Platforms. Technical Report G00369275, Gartner (2019)
6. Rutkowski, M., Boutier, L., Lauwers, C.: Topology and Orchestration Specification for Cloud Applications (TOSCA). Technical Report V1.2, Organization for the Advancement of Structured Information Standards (OASIS) (January 2019)
7. Kurtzer, G.M., Sochat, V., Bauer, M.W.: Singularity: Scientific containers for mobility of compute. *PLOS ONE* **12**(5) (May 2017) e0177459
8. Bergmayr, A., Breitenbücher, U., Ferry, N., Rossini, A., Solberg, A., Wimmer, M., Kappel, G., Leymann, F.: A systematic review of cloud modeling languages. *ACM Comput. Surv.* **51**(1) (2018) 22:1–22:38
9. Kritikos, K., Skrzypek, P.: Are Cloud Modelling Languages Ready for Multi-Cloud? In: 12th International Conference on Utility and Cloud Computing Companion (UCC'19 Companion), Auckland, New Zealand, IEEE/ACM (2019)
10. Woitsch, R., Utz, W.: Business Process as a Service Model Based Business and IT Cloud Alignment as a Cloud Offering. In: ES, IEEE (October 2015)
11. Netto, M.A.S., Calheiros, R.N., Rodrigues, E.R., Cunha, R.L.F., Buyya, R.: Hpc cloud for scientific and business applications: Taxonomy, vision, and research challenges. *ACM Comput. Surv.* **51**(1) (January 2018) 8:1–8:29