

Cloud Brokerage: A Systematic Survey

ABDESSALAM ELHABBASH, FAIZA SAMREEN, JAMES HADLEY, and YEHIA ELKHATIB, Lancaster University, UK

Background— The proliferation of cloud services has opened a space for cloud brokerage services. Brokers intermediate between cloud customers and providers to assist the customer in selecting the most suitable service, helping to manage the dimensionality, heterogeneity, and uncertainty associated with cloud services. *Objective*— Unlike other surveys, this survey focuses on the customer perspective. The survey systematically analyses the literature to identify and classify approaches to realise cloud brokerage, presenting an understanding of the state of the art and a novel taxonomy to characterise cloud brokers. *Method*— A systematic literature survey was conducted to compile studies related to cloud brokerage and explore how cloud brokers are engineered. These studies are then analysed from multiple perspectives, such as motivation, functionality, engineering approach, and evaluation methodology. *Results*— The survey resulted in a knowledge base of current proposals for realising cloud brokers. The survey identified differences between the studies' implementations, with engineering efforts directed at combinations of market-based solutions, middlewares, toolkits, algorithms, semantic frameworks, and conceptual frameworks. *Conclusion*— Our comprehensive meta-analysis shows that cloud brokerage is still a formative field. Although significant progress has been achieved in this field, considerable challenges remain to be addressed, which are also identified in this survey.

CCS Concepts: • **Computer systems organization** → **Cloud computing**; *Reliability*; *Maintainability and maintenance*; • **Software and its engineering** → Software design tradeoffs;

Additional Key Words and Phrases: Cloud computing; Cloud brokerage; Systematic literature review; Survey.

ACM Reference Format:

Abdessalam Elhabbash, Faiza Samreen, James Hadley, and Yehia Elkhatib. 2018. Cloud Brokerage: A Systematic Survey. *ACM Comput. Surv.* 0, 0, Article 0 (April 2018), 28 pages. <https://doi.org/0000001.0000001>

1 INTRODUCTION

Cloud computing is the delivery of computing and storage resources to cloud customers as services in a pay-as-you-go manner [Armbrust et al. 2010]. Despite the promised benefits of cloud computing (such as low cost, high availability, and flexible application deployment), the adoption of cloud brokers in practice remains sparse [Eisa et al. 2016; Elkhatib 2016; Satzger et al. 2013]. The market for cloud services is overwhelmed with a high number of heterogeneous cloud offerings, making the selection of a cloud service a challenging task for the Cloud Services Customer (CSC) [AppDirect 2017; (CSC) 2016; Elkhatib et al. 2018; Ghrada et al. 2018]. Furthermore, considering that each Cloud Service Provider (CSP) exposes their unique Application Program Interface (API), designing and developing an application so that it can be deployed on a specific CSP does little to mitigate the development efforts to move the application, e.g., if it performs badly, on the selected CSP.

Authors' address: Abdessalam Elhabbash; Faiza Samreen; James Hadley; Yehia Elkhatib, i.lastname@lancaster.ac.uk, Lancaster University, MetaLab, School of Computing and Communications, Lancaster, LA1 4YW, UK.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

0360-0300/2018/4-ART0 \$15.00
<https://doi.org/0000001.0000001>

To bridge the above gaps, the cloud community has long proposed that cloud brokers intermediate between CSCs and CSPs, thereby mitigating the risk of selecting any given CSP. A cloud broker is an intermediary between CSCs and CSPs to support CSCs decision making and enhance service delivery [Barker et al. 2015]. Brokerage can benefit the CSCs by abstracting away differences between CSPs and by helping CSCs to find the most suitable cloud services. These benefits aim to enable CSCs to lower costs and seamlessly switch between CSPs to ensure that their application requirements are always met. These benefits can also encourage forgoer businesses to trust and thus adopt cloud computing [Anastasi et al. 2017].

To realise brokerage, applications must be able to cross the boundaries of any given CSP. This gives rise to the field of *cross-cloud computing* [Elkhatib 2016] that aims to support application developers with the challenges associated with interoperability, resource scheduling strategies, dynamic deployment, and migration. Within this growing field, the term ‘brokerage’ has been used to refer to different intermediation models. One such model is that of cloud federation, mandating that competing CSPs agree and implement common technologies (such as virtualisation technologies and APIs) to enable the broker to select a CSP service, deploy use CSC’s application and adapt deployments. In contrast, the multi-cloud model does not assume common technologies. Instead, to facilitate switching between CSPs, the broker’s role must also encompass abstracting away the differences between CSPs. A third model is the decision support system, in which brokerage is limited to recommending a cloud service to CSCs based on their requirements. In this model, the practical aspects of deploying and relinquishing services are left to the CSC.

The breadth of challenges surrounding cloud brokerage has brought about so much research there that is a need to systematically analyse the proposed solutions. Few works focused on surveying research efforts related to cloud brokerage such as cloud services selection approaches [Eisa et al. 2016], interoperability and portability in the cloud [Loutas et al. 2011], and resource allocating mechanisms [Sheikh. and Navimipour 2018]. They touched on brokerage from the CSP prospective; focusing on interoperability between the CSP cloud services. Moreover, to the best of our knowledge, there has not been a Systematic Literature Review (SLR) on cloud brokerage, which is required to assess the progress in this field and identify dimensions for future research. A SLR is a methodological survey that aims to systematically capture efforts relating to a specific topic to develop a comprehensive and unbiased knowledge base around it [Kitchenham 2004].

This paper provides the results of a SLR that was carried out on the topic of cloud brokerage. The SLR results are analysed and presented as part of a taxonomy that should aid researchers in this field to appreciate both the big picture and some of the finer details of work in this area. In this sense, this paper aims to review and analyse existing solutions in terms of:

- exploring the reasons that motivate the need for cloud brokers;
- developing a fundamental understanding of the tasks carried out by cloud brokers;
- investigating approaches to engineer and realise cloud brokers;
- examining approaches to evaluate proposed solutions; and
- identifying the main limitations of current solutions and highlighting areas for future research where improvements can be made.

The remainder of this paper is organised as follows. The next section briefly presents other reviews related to cloud brokerage. Section 3 describes the methodology followed to conduct this survey including the objectives and research questions. Section 4 gives an overview of the selected papers. Section 5 answers the research questions. Section 6 presents a characterisation framework of the selected papers to synthesise and discuss the results and identify the limitations of current solutions. Section 7 reflects on the results and discusses threats of validity of this SLR. Finally section 8 concludes the article and identifies future areas of research.

2 RELATED WORK

To the best of our knowledge, there is no other comprehensive or systematic survey of works on cloud brokerage. Cloud brokerage is a large but fragmented field; there is considerable discord between contributions and the wording used to describe them in research papers. To the best of our knowledge, this is first survey to address this discord and, in doing so, to contribute a thorough analysis of the state of the art and detailed and well-supported suggestions for future work.

In spite of the considerable differences between this survey and others, this survey briefly presents other reviews that relate at varying degrees to cloud brokerage. It presents a comparison summary of this survey to others in Table 1.

[Eisa et al. \[2016\]](#) surveys some cloud services selection approaches. The authors focus on analysing three commercial CSP search tools that help CSCs to search for cloud services. The survey also presents a number of academic works related to cloud selection. In their conclusions, the authors report the need for cloud brokers that can extend and use cloud services selection tools to assist CSCs. Other similar surveys also focus on cloud service selection, e.g., [Sun et al. \[2014\]](#). [Sheikh. and Navimipour \[2018\]](#) also survey service selection but is the only other survey to our knowledge that thoroughly differentiates between CSC and CSP-centric selection.

[Aldawsari et al. \[2015\]](#) is a position paper that compares the architecture of a small selection of cloud brokers, focusing primarily on Service Level Agreement (SLA) negotiation, service selection and, in particular, energy efficiency. The authors remark that the current state of the art does not address the demand for intermediaries that can reduce energy consumption and propose ideas for an energy-aware cloud broker.

[Barker et al. \[2015\]](#) reviews prominent commercial solutions in cloud brokerage from an academic perspective, classifying them into one of four categories (performance, migration, theoretical models and data), and outlines a research roadmap in light of these efforts. [Grozev and Buyya \[2014\]](#) analyse a number of the early cross-cloud application brokering mechanisms, while [Assis and Bittencourt \[2016\]](#) focus on cloud federations.

[Al-Dhuraibi et al. \[2018\]](#) surveys work on several elasticity challenges in the cloud including inherent challenges to horizontal and vertical scaling and application configuration challenges. The authors also touch on challenges specific to scaling beyond a single CSP. The focus of this survey is to determine contributions to a range of individual challenges in cloud scaling.

There is a limited number of surveys on interoperability and portability issues in the cloud. These studies tend to focus on the interoperability challenge from a particular angle: [Loutas et al. \[2011\]](#) is focused predominantly on semantic divergence in the cloud ecosystem as a root cause for the interoperability challenge. [Zhang et al. \[2013\]](#) presents a high-level taxonomy of issues relating to interoperability at the IaaS level, encompassing broad issues ranging from APIs and GUIs to virtualisation technologies, encryption mechanisms and SLA verification. [Kaur et al. \[2017\]](#) surveys and analyses approaches implementing interoperability and portability in different inter-cloud models. Although their work shares some similarities with ours, our analysis is more from the CSC's perspective (*i.e.* how the brokers benefit the CSCs) whereas their analysis is more from the CSP's perspective (*i.e.* the interoperability and collaboration between the CSPs).

There have been surveys on other aspects of cloud computing, such as design issues (*cf.* [Zhang et al. 2010](#)), resource management (*cf.* [Manvi and Shyam 2014](#); [Zhan et al. 2015](#)), monitoring (*cf.* [Aceto et al. 2013](#)), migration (*cf.* [Jamshidi et al. 2013](#)), service composition (*cf.* [Jula et al. 2014](#)) and security (*cf.* [Iankoulova and Daneva 2012](#); [Patel et al. 2013](#)), among other subjects.

Table 1. Comparison to other reviews.

Focus	Interop-erability	Cloud Security	Interop-erability	Cloud Security	Service Selection	Position Paper	Service Selection	Cloud Federations	Interop-erability	Cloud Elasticity	Resource Allocation	Cloud Brokerage	This Paper
Differentiates CSP and CSC-centric					✓						✓		✓
Selection methods	SLR	Adhoc	Adhoc	Adhoc	Adhoc	Adhoc	Adhoc	Adhoc	Adhoc	Adhoc	Adhoc	SLR	43
Brokers considered	19	N/A	N/A	N/A	45	13	8	45	30	5	80	681	
Brokers analysed in detail	8	N/A	N/A	N/A	16	7	8	16	5	0	17	33	
Discerns public/hybrid/private cloud							✓	✓	✓				✓
Framework to analyse brokers													✓
Compares motivations					✓		✓	✓	✓				✓(RQ1)
Compares functionalities					✓		✓	✓			✓		✓(RQ2)
Compares approaches													✓(RQ3)
Year	2011	2012	2013	2013	2014	2015	2016	2016	2017	2018	2018	2018	2018

3 SURVEY METHODOLOGY

The purpose of this study is to comprehensively survey the literature to identify the state of the art when it comes to brokerage in cloud computing. This field is a vast and growing one: thousands of papers have been published on many different aspects of cloud computing and by researchers from different backgrounds (*e.g.*, distributed computing, high performance computing, grid computing, optimisation theory, financial derivatives, etc.). As such, there is considerable variance in the terminology used in the literature which is supported both by previous work (*cf.* [Elkhatib 2016]) and by our preliminary survey. Adopting a systematic surveying approach is thus the most rigorous way of identifying all relevant work produced by researchers across disciplines. Figure 1 depicts the methodology followed to conduct this study.

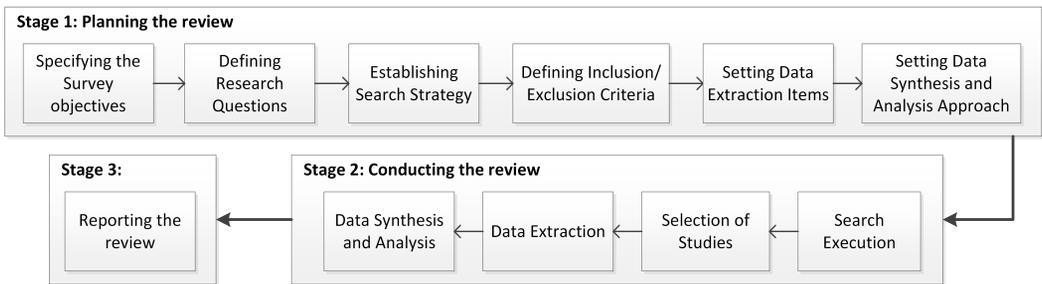


Fig. 1. Research methodology.

3.1 Research Objectives

The objectives of the research herein presented are the following:

- 01 Build a library of work that is related to the general topic of cloud brokerage, and provide this to other researchers as an open dataset.
- 02 Identify a more focused set of works that have shaped research in cloud brokerage.
- 03 Characterise existing solutions in the field of cloud brokerage, and clarify the similarities and differences between them using a characterisation framework.
- 04 Produce a taxonomy of the state of the art to further highlight the adopted approaches and methods.
- 05 Distinguish gaps in the state of the art in terms of research challenges and approaches.

3.2 Research Questions

We conduct this study by addressing the following research questions.

RQ1 *What is the motivation for designing cloud brokers?*

This question addresses the need to identify the requirements that have stimulated the research and development of cloud brokerage systems. The goal of this research question is to identify the challenges in the cloud computing paradigm that motivated the need for cloud brokerage.

RQ2 *What are the functionalities of a cloud broker?*

This question identifies the range of operations that are performed by cloud brokers in order to achieve their goals. This research question is helps to investigate the gaps between the motivations and capabilities of the current solution and consequently identify areas of future research.

RQ3 *What are the approaches for engineering cloud brokers?*

This question identifies engineering approaches used to implement cloud brokers and investigate links between approaches and outcomes. The goal of this research question is to investigate how cloud brokers have been realised.

3.3 Search and Selection Strategy

Cloud computing is an extremely active field. It is for this reason that our survey method had to empirically draw related work from all relevant sources.

For work from the research community, we built a knowledge base from seven major publishers and online databases in computer science: ACM DL ¹, IEEEExplore ², ScienceDirect ³, SpringerLink ⁴, Wiley Online Library ⁵, PLOS ⁶, and ArXiv ⁷. The works from these sources have been captured using a rigorous **five-phase procedure** that is described below. Aggregate indexing services such as Scopus and Web of Science were also consulted but eventually disregarded as they had only duplicated works harvested from the original publisher databases.

3.3.1 Search Phase. In the first phase, we query the main publication portals using a set of different search terms. The retrieved results along with their metadata (including title, abstract, and publication outlet) are stored in a local knowledge base which will be filtered in the subsequent phases.

The search string combinations used are outlined in Figure 2. The primary keywords were chosen as key identifiers of work in the area of cloud brokerage. However, cloud brokerage is a relatively young and developing field [Elkhatib 2016], and as such we needed to augment these keywords with others that are analogous. The second level keywords were therefore added to capture any publications that tackle the heterogeneity challenge albeit not under the label of brokerage. Alternative spellings (e.g., “inter-cloud”, “intercloud”, and “inter cloud”, American and British spelling variations, etc.) were included to ensure comprehensive coverage. Primary keywords (K_P) were used with any of the secondary (K_S) or additional (K_A) keywords; *i.e.* $\forall K_P \wedge \forall K_S \wedge \forall K_A$.

Finally, we augmented this with keywords extracted from well known works on cloud brokers that we were familiar with before carrying out the survey (such as [Ferrer et al. 2012; Tordsson et al. 2012]). These are shown on the far right in the diagram.

3.3.2 Screening Phases and Selection Criteria. The subsequent three phases filter the knowledge base based on the contents of the publication title, abstract, introduction and conclusion sections, respectively. Authors and publishing outlet (journal, conference, etc.) were intentionally left out to avoid bias towards publications based on their provenance and to ensure that relevance was judged purely on the content conveyed by the paper’s metadata and in key sections. We considered that not all titles and abstracts are truly reflective of the contents of a paper. However, we concluded that our approach was necessary to avoid provenance-based preconceptions.

A large number of works reduce brokerage to a mere resource allocation problem, e.g., [Mehrotra et al. 2016], and as such assume cooperation between CSPs to gloss over interoperability and comparability challenges. Other studies (e.g., [Javed et al. 2016]) have a strong expectation of active

¹<https://dl.acm.org/>

²<https://ieeexplore.ieee.org>

³<https://www.sciencedirect.com/>

⁴<https://link.springer.com/>

⁵<https://onlinelibrary.wiley.com/>

⁶<https://www.plos.org/>

⁷<https://arxiv.org>

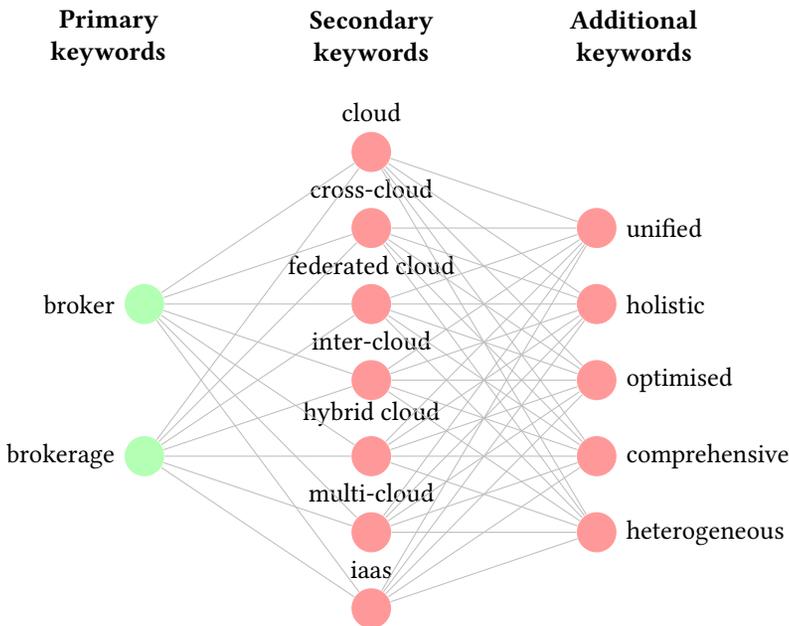


Fig. 2. Search queries used to identify works to include in our knowledge base

involvement of cloud providers in issuing tenders or responding to calls for bids ⁸ These have been disregarded as history has shown us that CSPs are not interested in this form of marketplace [Johnston 2014].

Table 2 summarises the inclusion and exclusion criteria that were adopted to perform the selection. Every effort was made to retrieve manuscripts behind paywalls through our institutional subscription. Of the papers that reached the third phase, only 3 were not accessible (0.03% of the total, 1.68% at phase 3).

When the same work was published in more than one outlet, as is common with conference papers that then get expanded into a journal article, only the original work is included unless there is an enhancement in the follow up publication that is significant in brokerage-related aspects.

During each screening phase, every publication is screened by at least two of this paper’s authors. Each screener is asked to judge the relevance of the paper and, accordingly, choose to either *Accept*, *Reject*, or *Defer*. If the two screeners do not agree in their choices or if one of them chooses to defer his/her decision, then the publication in question is marked for discussion by the screeners until a decision is reached. In the final screening phase, the screeners also extracted keywords from the remaining set. These are used to identify the most suitable reviewer to further scrutinise the paper. Only papers with an original technical contribution are selected; editorials are ignored at this phase, but the most pertinent of these will be revisited in the Related Work section.

At the cost of additional deferred inspection effort, we decide to err on the side of caution and accept papers that are deemed to still be borderline even after deliberation between the screeners.

3.3.3 Inspection Phase. The final phase is to thoroughly read the accepted publications to ascertain their relevance and contribution in the field of cloud brokerage. The papers are categorised

⁸A process where both providers and users place their bids, and the broker matches the available buying and selling prices.

Table 2. Studies Selection Criteria

Inclusion Criteria	
I1.	Research papers presenting new and emerging ideas.
I2.	Literature published as books, book chapters, collections, and technical reports.
I3.	Papers developing or extending brokerage systems in the cloud paradigm.
I4.	Papers discussing aspects of brokerage in the cloud paradigm.
Exclusion Criteria	
E1.	Papers not in the form of a full research paper, <i>i.e.</i> in the form of abstract, tutorials, presentation, or essay.
E2.	Papers without an available abstract, or behind a paywall.
E3.	Papers not written in the English language.
E4.	Papers reducing brokerage to a mere resource allocation problem or restricting to a single application type such as MapReduce.
E5.	Papers focusing on brokerage in other paradigms; <i>i.e.</i> not the cloud paradigm.

according to their contributions to achieve a systematic knowledge base which is the core goal of this study.

4 REPORTING THE REVIEW

This section provides an overview of the selected studies.

4.1 Overview of the intermediate selection process outcome

This subsection presents the high-level results of the SLR. Table 3 presents a concise quantitative summary of the results of our search, screening and inspection phases across the 7 publication libraries. The survey captured a very large number of papers. Screening phases helped reduce this number but at a much lower rate than expected. This is mainly because many papers promise a brokerage solution when in fact they offer a supporting technology or model. A representative example of this is CloudCmp [Li et al. 2010], an early system to compare multiple cloud providers in terms of performance and cost. However, in essence, CloudCmp is a standard benchmarking suite for Java virtual machines. Other work followed using similar approaches, *e.g.*, [Khajeh-Hosseini et al. 2012, 2011]. Such work could certainly be used to support the construction of a broker but does not in itself implement the full range of services typically associated with a broker. Other examples include scheduling optimisations, negotiating algorithms, and ontologies.

4.2 Overview of the selected studies

The survey eventually identified a set of 33 principal papers on cloud brokerage solutions. As shown in Figure 3, a significant number of these papers were published in journals (sixteen), followed by a smaller number of publications (eleven) in conferences, four in workshops, and two in other venues. When these results are plotted against publication year (Figure 4), we discern the typical pattern of early work being published in specialist conferences such as CCGrid, followed by later works in journals. The spread of original publishers (Figure 5) is dominated by Elsevier and Springer; the former due to many papers being published in their FGCS journal whose call for papers was one

Table 3. Selection phases and results.

Phase	Process	Selection criteria	ACM DL	arXiv.org	IEEEExplore	PLOS	ScienceDirect	SpringerLink	Wiley Online Library	Total
1	Search	Keywords (Figure 2)	3517	606	502	21	1574	4413	214	10847
2	Screening	Title	101	74	156	2	142	190	16	681
3	Screening	Abstract	34	18	42	0	44	36	5	179
4	Screening	Introduction and Conclusion	17	11	20	0	27	19	4	98
5	Inspection	Full paper	9	2	4	0	10	6	2	33

of the earliest seekers of advancements in cloud brokerage, and the latter due to their pattern of publishing specialist conference proceedings.

With respect to the reusability of the developed systems, the source code of 7 out of the 33 studies are made available to the public; 6 of them are published online and 1 is mentioned to be available upon request.

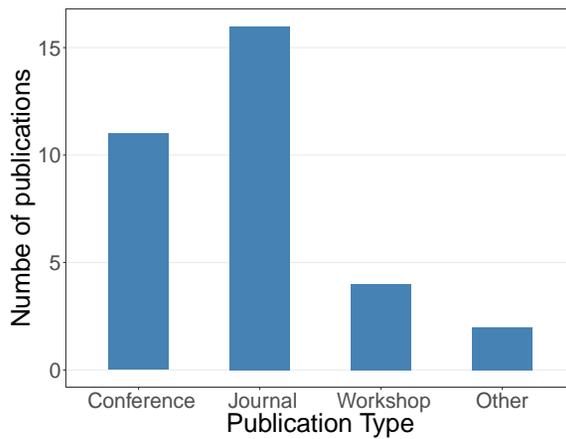


Fig. 3. Total distribution of publication types of selected studies.

5 REVIEW RESULTS

This section presents answers to the posed research questions.

5.1 Motivation for designing cloud brokers (RQ1)

This section looks at the challenges that motivated the cloud community to design cloud brokers and shows how selected papers addressed these challenges. The majority of papers clearly state

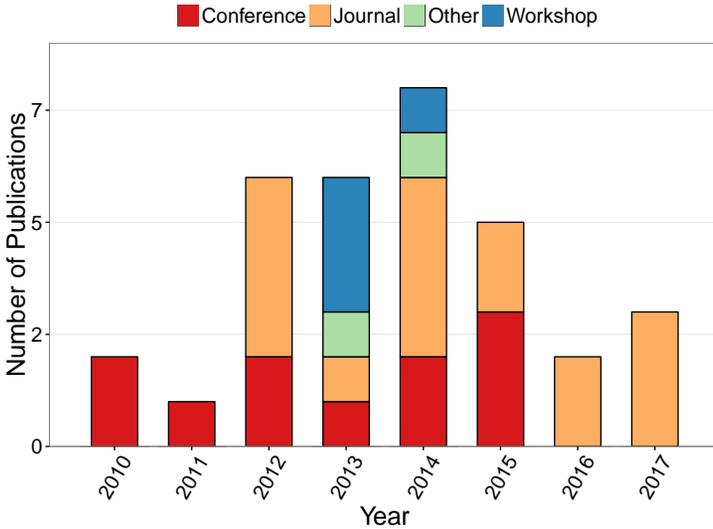


Fig. 4. Distribution of selected studies by type over publication year.

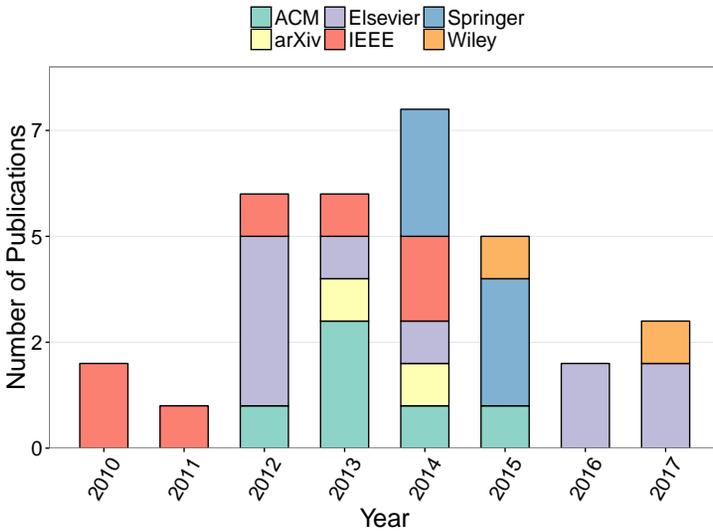


Fig. 5. Distribution of selected studies by publisher over publication year.

their motivation. It has been found from this that cloud brokers are motivated by the need to address one or more of four key challenges: (i) Dimensionality, (ii) Vendor lock-in, (iii) Meeting requirements, and (iv) Pathological.

Figure 6 depicts where the surveyed papers lie across these different motivations. Most papers declare one or two motivations, with the need to tackle dimensionality and meet operational requirements being the most common. More than a third of the papers declare two or three motivations. None declare all four.

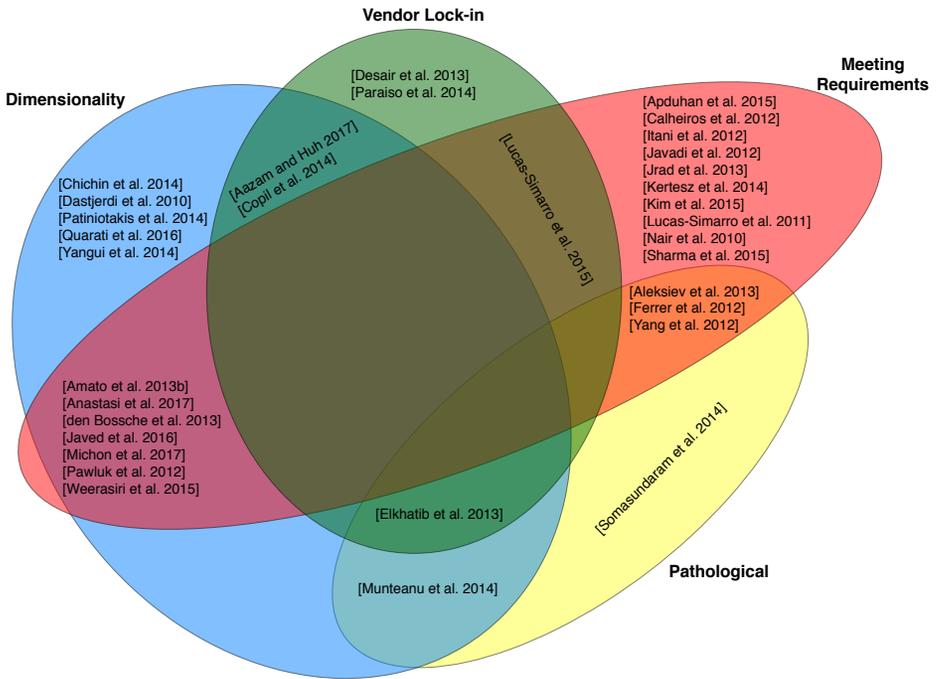


Fig. 6. The four main motivation categories and the papers in each.

In the following, each category of motivation is described in detail.

Dimensionality. The vast number of cloud providers and their respective offerings makes the task of selecting a provider and service challenging. Figure 7 shows the wide range of instance types offered in the IaaS market, and it illustrates how this wide range increased further between 2015 and 2017. In view of this, the current process of manually selecting the optimal service can overwhelm a human decision maker. This challenge motivates the development of systems to support decision making [Aazam and Huh 2017; Amato et al. 2013; Anastasi et al. 2017; Chichin et al. 2014; Dastjerdi et al. 2010; den Bossche et al. 2013; Javed et al. 2016; Michon et al. 2017; Patiniotakis et al. 2014; Pawluk et al. 2012; Quarati et al. 2016; Weerasiri et al. 2015; Yangui et al. 2014].

Vendor Lock-in. Vendor lock-in refers to the difficulty and expense that customers must sometimes incur to switch their service provider [Hsu et al. 2014; Leavitt 2009]. These costs and difficulties arise due to the technical differences between service providers, such as incompatibilities between CSP services and the financial cost of migrating large files. Desair et al. [2013] state that vendor lock-in limits the general exploitation of cloud computing due to the partial loss of control over data and applications. Paraiso et al. [2014] are motivated by the need to develop CSP-independent applications, which is currently unfeasible due to vendor lock-in.

Meeting Requirements. Many of the surveyed works cite the inability of a single CSP to satisfy user requirements as a key motivator. For instance, Calheiros et al. [2012] motivate the need for brokerage to enable applications to scale over multiple cloud data centres when the resources of one data centre are oversubscribed. Jrad et al. [2013] are motivated by the need to run workflow

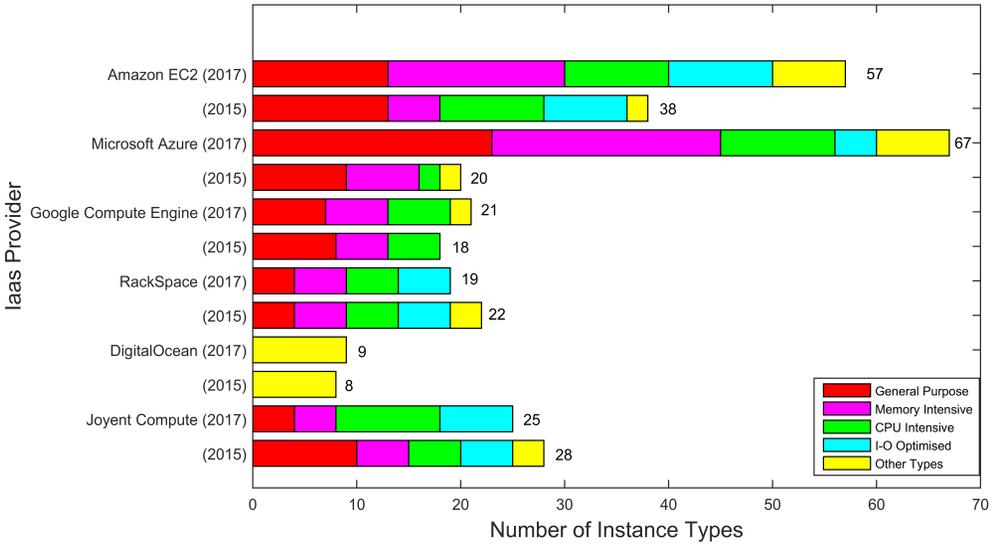


Fig. 7. The number of on-demand Linux cloud instance types from major CSPs in August 2015 and July 2017.

applications⁹ on multiple clouds. Kim et al. [2015] offers a similar motivation in the context of to mobile clouds. Additionally, observing the real performance of cloud services and assuring SLAs motivates the need for brokers in [Anastasi et al. 2017; Ferrer et al. 2012; Itani et al. 2012; Kertész et al. 2014; Michon et al. 2017] to help customers select between different resources provisioning strategies, and decide on optimal VM placement [Lucas-Simarro et al. 2011, 2015]. The provision of availability and reliability guarantees at low cost also motivate many works [Apuhan et al. 2015; Javadi et al. 2012; Sharma et al. 2015]. A recurring example is that of cloudbursting, *i.e.* exploiting public cloud resources only when a situation arises where private cloud resources are insufficient or inadequate to satisfy application needs. This requires brokers to facilitate the addition of resources to scale beyond the limits of owned infrastructure by seamlessly adding and removing rented resources [Aleksiev et al. 2013; Nair et al. 2010]. [Copil et al. 2014] argues that the existence of different APIs of the various providers motivates the need for a controller that provides complete view of the multi-cloud services so that users can have elastic control of their applications.

Pathological Motivations. These motivations vary between achieving specific interoperability guarantees (*e.g.*, between different grid and cloud resources [Aleksiev et al. 2013; Munteanu et al. 2014; Somasundaram et al. 2014]), enabling collaboration between users from different backgrounds [Elkhatib et al. 2013], hosting compute-intensive applications [Yang et al. 2012], and addressing complex legislative issues [Ferrer et al. 2012].

5.2 Cloud broker functionality (RQ2)

This section addresses RQ2 which looks at the functionalities that are performed by the cloud brokers. Eight main functionalities have been extracted from the selected papers, which are now discussed.

⁹Workflow applications are “parallel applications that consist of a series of computational tasks logically connected by data- and control-flow dependencies” [Jrad et al. 2013].

Decision Support. Approaches implementing this functionality support the CSCs in making decisions concerning selection of resources. These approaches require that the CSC specifies functional and non-functional requirements to enable the system to recommend cloud providers and instances that best fit their needs. Users ultimately make the decision and allocate resources. Examples of systems implementing this functionality include [Chichin et al. 2014; Javed et al. 2016; Jrad et al. 2013; Patiniotakis et al. 2014; Quarati et al. 2016].

Resource Monitoring. This concerns the ongoing collection of data from cloud services that could ultimately assist the broker in making effective decisions. Collected data generally includes metrics around performance and availability. This functionality can be implemented at the VM level as in [Pawluk et al. 2012], the hardware level as in [Yang et al. 2012], or the application level as in [Quarati et al. 2016].

Policy Enforcement. User-defined policies provide constraints that guide selection and allocation decisions. Such constraints relate to the location of deployment, security, storage encryption and cost, among others. Brokers that implement policy enforcement implement and apply mechanisms to ensure that selected cloud services meet the constraints specified by the CSC. The broker may also use the policies to make adaptation decisions, *e.g.*, VM migration.

SLA Negotiation. SLA negotiation involves the broker capturing the user's requirements and enabling CSPs to tender cloud services that meet them. The result of the negotiation process is an SLA that defines commitments between the provider and the customer. SLA negotiation requires cooperation between CSPs and the broker to manage tenders and mechanisms to deal with SLA breaches. Examples of brokers that implement SLA negotiation are, however, not plentiful [Azam and Huh 2017; Ferrer et al. 2012; Nair et al. 2010].

Application Deployment. Application deployment is related to procuring the environment where that application will be made available. In the context of cloud brokerage, some brokers provide the functionality of deploying the application on the selected CSP service using particular settings. Others do not support the actual deployment and just recommend one or more deployment environments.

Migration. This addresses the challenges involved in modifying the deployment, *i.e.* changing the allocation of resources after resources are deployed. The need to migrate may be driven by changes in the CSC's requirements or by changes in the CSP's conditions including performance and availability. Six out of the selected studies support this functionality [Ferrer et al. 2012; Munteanu et al. 2014; Nair et al. 2010; Sharma et al. 2015; Yang et al. 2012; Yangui et al. 2014], all at the VM level.

API Abstraction. Abstracting away technical differences between CSPs is fundamentally necessary to achieve seamless brokerage. This can be implemented in one of two ways. First, the broker may identify overlap between CSPs' APIs and present CSCs with the least common denominator (LCD). Alternatively, the broker may expose its own meta-API, calls to which are translated into native API calls. In both cases, the user must use the API exposed by the broker, that is likely to be substantially different to the CSPs' own APIs. The difference between these approaches lies in translation effort: the former prioritises ease of translation (ideally eliminating it completely) and thus results in a restrictive LCD API; the latter prioritises coverage over conformity.

VM Interoperability. This functionality means that the broker allows the conversion of the execution unit (*e.g.*, VM) between different provider formats. In [Nair et al. 2010] interoperability is achieved through the standard API abstraction (OCCI) which is assumed to be adopted by all of the cloud providers. The work of [Dastjerdi et al. 2010] (which applies ontology-based brokerage) introduces mediators that are defined as elements that handle interoperability between different ontologies or services.

5.3 Engineering approaches (RQ3)

This section addresses RQ3 which looks at the approaches proposed to design and construct the cloud brokers to highlight how they are realised.

5.3.1 Market-based. Approaches in this category model proposed brokers as markets in which providers publish their services and consumers bid for them. In this model, economic models form the matchmaking process between providers and consumers. The *Smart Cloud Marketplace* (SCM) [Chichin et al. 2014] architecture consists of agents (*e.g.*, market-agent, buyer-agent and seller-agent) that perform the role of a cloud service *exchange*, making intelligent decisions on behalf of customers and providers. SCMs implement a variety of trading protocols such as fixed-price markets, negotiation, tendering, and auctions. Another example is the *SpotCheck* broker [Sharma et al. 2015] that re-sells resources from IaaS providers in a unified market with varying guarantees and pricing models. This approach nests VMs within spot instances and migrates them during price spikes. In catering to spot instances, a service currently only offered by Amazon EC2, this broker uniquely relies on a single CSP. Javed et al. [2016] present a Cloud Market Maker (CMM) to increase providers' return on investment using a dynamic pricing strategy and to support CSCs in selecting CSPs. The CMM adapts the price of offered resources at operating-time using a supply-demand model that presents providers with the market equilibrium price, thereby encouraging providers to respond to demand.

5.3.2 Frameworks and architectures. The majority of work falls under this category where designed architectures that contain components, methods, and relationships are proposed. In some cases the authors provide implementations (either real or simulation-based using *e.g.*, CloudSim [Calheiros et al. 2011]) for the framework while in others the frameworks remain at the conceptual levels. The proposed frameworks can be categorised into the following:

Frameworks for executing applications on multi-clouds. These frameworks focus on selecting multiple clouds to execute an application towards a better performance (*i.e.* the application concurrently runs on multiple clouds). The application is divided into tasks so that each can be executed in a certain cloud. Jrad et al. [2013] implemented a framework for deploying workflow applications on inter-cloud environment, assuming that a standard API is adopted by the providers. The framework consists of a match-maker component that selects the data centres that satisfy the entire user predefined functional and non-functional requirements. It also consists of a scheduler component to distribute the workflow using the round robin scheduling policy. The brokerage system of Quarati et al. [2016] also addresses the execution of workflow applications on grids and clouds. The system includes two main components. The first is the CB-Portlet, which contains a number of portlets¹⁰ to enable users to specify their workflow requirements. The second is the DCI-BRIDGE that enables the interoperability between the different computing resources (cloud or grid-based) by creating the descriptions of the jobs that will be executed on the selected VMs; assuming the Open Cloud Computing Interface (OCCI)¹¹ standard is adopted. The framework of Copil et al. [2014] is a simple one controller component for an application deployed across multiple clouds. The controller builds a dependency graph model to represent vendor-specific elasticity capabilities. It enables the user to specify requirements (regarding monitoring, elasticity constraints, strategy directives) at different levels (*e.g.*, service topology, code region). These requirements are then analysed by a runtime and an action plan is generated accordingly to fulfil the user's requirements. Yang et al. [2012] proposed a framework to enable public clouds to cooperate to satisfy the infrastructure demands of CPU-intensive applications. The framework includes a coordinator that

¹⁰A portlet is a pluggable Java web component with a user interface definition.

¹¹<http://occi-wg.org/>

guarantees an SLA with hosts and provides the user with a token. This token enables the user to deploy to the host directly. The SLA is monitored and enforced by migrating some of the workload to another CSP. The *soCloud* framework, presented by Paraiso et al. [2014], extends the OASIS SCA¹² standard to build multi-cloud PaaS applications with added policy to ensure load balancing and high availability. The brokerage part is simple; it implements logic to ensure availability and minimum QoS. The application has to be written in a component-based fashion using the FraSCAti implementation of SCA. Negotiation among CSPs and pricing model variances are not considered.

Frameworks for elastic application execution. These frameworks focus on scaling up the cloud resources when the application's demands change. The VM-MAD framework Aleksiev et al. [2013] is specifically proposed for cloudbursting linux-based High-Performance Computing clusters. When the queue of a HPC cluster is overloaded, then cloud VMs are spun up to carry some of the load. Otherwise, cloud VMs are shut down when load recedes. Calheiros et al. [2012] also proposes a cloud coordinator that manages the purchase of resources when changes in the application demands occur. The coordinator is assumed to be part of each CSP architecture and provides modules to enable CSPs to negotiate and exchange resources among themselves to satisfy elastic applications.

Frameworks for scheduling and resource allocation. These frameworks focus on selecting the cloud resources based on application requirements and in some cases to increase the broker's revenue. Azam and Huh [2017] propose a framework to predict, reserve and allocate cloud resources. The framework is intended to be managed by a third-party for-profit body (*i.e.* not to be user controlled). The authors propose algorithms to deploy the optimal services using a prediction of future demand (to maximise the profit of the third party). The framework of Kertész et al. [2014] combines negotiation, brokering and deployment. The role of the broker is limited to managing the virtual resources. The broker functionality includes meta-brokerage and brokerage. The role of the meta-broker is to decide which broker is capable of satisfying the user requirements. On the other hand, the broker interacts with virtual and physical resources. Lucas-Simarro et al. [2011, 2015] divide the cloud broker into three components: the VM manager, the scheduler and the cloud manager. The role of the scheduler is to decide about the VMs placement among the available clouds. The cloud manager addresses the monitoring and management of the VMs life cycle. The authors focus on the scheduler component which is realised by algorithms that optimise the cost of required resources, based on the historical prices. In Javadi et al. [2012] the framework has three main components, namely, "InterGrid Gateways (IGGs)", the "Virtual Infrastructure Engine (VIE)" and "Distributed Virtual Environment (DVE)" manager. The IGG selects a suitable provider that for an incoming request. The VIE manages the private cloud resources (starts, pauses, resumes, and stops VMs) while the DVE allocates and manages resources on behalf of applications. The STRATOS framework [Pawluk et al. 2012] provides automated decision making for resource acquisition between different vendors based on two steps: determining the number of resources required, and determining where to place the resources.

5.3.3 Toolkits. In the context of cloud brokerage, toolkits can be defined as development tools that are used to design and develop broker functionality. The OPTIMIS toolkit [Ferrer et al. 2012] consists of a set of components that allow (in theory) a variety of architectures of multiple clouds (a broker is one of the architectures). The toolkit allows the broker to act as a CSC of the CSPs and as a CSP for the CSCs. The toolkit enables aggregating resources from CSPs and provisioning them to the CSCs. The work of Nair et al. [2010] focuses on addressing security concerns in OPTIMIS. The toolkit in Elkhatab et al. [2013] enables a shared space to facilitate data access and exchange between multiple communities on a common environmental issue. The tool enables collaborators to access shared data that are stored in federated clouds. It includes a Resource Broker (RB) that

¹²<http://www.oasis-open.org/sca>

selects a cloud service that is appropriate for the type of computation required. The contribution of Anastasi et al. [2017] is a tool that assists users in selecting the most cost-efficient resources to execute scientific applications. The paper addresses the lack of knowledge about the actual cost of running an application is uncertain before runtime. This means that measuring the impact of alternative execution strategies (*i.e.* using alternative resources) can be expensive. The proposed tool, called Schlouder, is a simulator that predicts the cost of executions under various strategies; supporting the user to make decisions. The authors demonstrate the effectiveness of the simulator by showing that cost predictions are accurate. However, they admit that accuracy depends on users' ability to accurately estimate the duration of each task, which is difficult in practice.

5.3.4 Middlewares. A middleware is a system that provides bridging and interoperability between various systems. In the cross-cloud context, a middleware provides interoperability between different cloud vendors. Few approaches introduced their cloud brokers as middlewares. The CompatibleOne [Yangui et al. 2014] middleware allows developers to combine different cloud services provided by different providers. It provides an abstraction called CORDS (CompatibleOne Resource Description System) to model and manage cloud services on CompatibleOne. CORDS abstracts, adds to and maps to OCCI. The goal is to enable developers to specify a service in enough detail to enable it to be created at any CSP. Munteanu et al. [2014] also presents a middleware for interfacing with multiple IaaS and PaaS providers. The paper proposes a basic technique of SLA negotiation between more than one candidate provider. This is done using descriptors that are semantically matched to CSP capabilities. The presented descriptor-driven framework is VM-specific, relying on a brief SLA comparison phase followed by execution of predefined scripts. This is limited by the expressiveness of the SLAs provided by different vendors and the semantic differences therein. It also seems to exclude brokering other types of cloud provisions such as storage, appliances. PaaS Hopper [Desair et al. 2013] is a policy-driven middleware that attempts to mitigate the problem of losing control over application and data when adopting cloud resources. It includes two layers, namely, the abstraction layer and the policy-driven distributed execution layer. The former deals with heterogeneity and interoperability with a uniform API for interaction to middleware. The latter allows users to specify constraints to select the location of application execution (in the private or the public cloud) in addition to other criteria related to data security *e.g.*, connection security and storage encryption.

5.3.5 Semantics. These approaches deal with the heterogeneity of cloud resources' configuration and the cloud services' APIs. The general case aims at unifying the CSPs resource configuration annotations and common programming APIs. They propose ontologies to achieve this goal and assume that CSPs will adopt such common technologies. Weerasiri et al. [2015] propose a common resource configuration language, called "Resource Configuration Service" (RCS) which enables abstracting the heterogeneous CSPs services interfaces through a unified interface. The authors also propose a "Cloud Resource Configuration Description" (CRCD), which enables provider-independent resource requirement descriptions. Such abstractions enable CSCs to model and execute cloud applications by focusing on specifying the applications' requirements. Dastjerdi et al. [2010] propose an ontology to model the virtual resources based on the Web Service Modeling Ontology (WSMO)¹³. The ontology provides the domain concepts and vocabulary for representing resources advertisements and requestor requirements. Apduhan et al. [2015] propose a broker-server that resides on the private cloud resources within a hybrid cloud. The proposed broker-server is based on adopting a cloud ontology in order to search, rank and select cloud services. Somasundaram et al. [2014] propose a domain specific ontology that provides a mechanism to represent grid and

¹³<http://www.wsmo.org>

cloud resources information. The ontology is used in a framework for resource discovery, SLA negotiation, and resource scheduling.

5.3.6 Algorithms. Some works focus on proposing algorithms to realise (in some cases partially) the brokerage system as a main contribution. [Amato et al. \[2013\]](#) propose a distributed algorithm where the broker is decomposed into a set of distributed brokers (each resides at a CSP) that cooperate to satisfy the users requests. The global broker sends out a call for proposals to satisfy the user's requirements, that include quantitative specifications as defined in OCCI and qualitative factors such as location. Each cloud runs an agent that offers at least one proposal to the broker to satisfy the user's requirements. The global broker then collects the proposals and selects the most appropriate one according to a brokering policy. [Itani et al. \[2012\]](#) propose an algorithm for brokerage in federated clouds inspired by the BGP routing protocol. Public clouds are advertised as if they were an autonomous systems in BGP. The broker routes procurement requests to one of the clouds according to its policy. The CSP either satisfies the request entirely or satisfies it transitively. For example, if the user requests a database server, the first CSP may request a storage server from another CSP as a prerequisite. [Kim et al. \[2015\]](#) propose a two-phases algorithm for workflow execution in mobile cloud, namely, the "Greedy Based Workflow Co-scheduling (GBWC)" phase and the "Resource Profiling Based Placement (RPBP)" phase. This two-phase algorithm is then used by the "Mobile Cloud Broker (MCB)" in the proposed system for cost adaptive VM management. The GBWC reduces the cost of utilising resources by simultaneous scheduling of tasks on the same VMs. The RPBP algorithm selects cloud physical resources to host a VM using simple profiling algorithms. The profiling is based on either the CPU capacity or the historical network delay. The profiles are used to rank the resources then the highest rank resources are selected subject to the budget constraints. [den Bossche et al. \[2013\]](#) propose an algorithm to schedule deadline-constrained workloads on a hybrid cloud. A hybrid cloud scheduler decides (based on the budget and deadline) whether a task can be scheduled on the private cloud or if it should be transferred to the public cloud. The tasks are stored in a queue based on their deadlines. A queue scanning algorithm is used to detect tasks that cannot be satisfied within the deadline (based on the elapsed and remaining time of the tasks in the queue). These tasks will then be executed on the public cloud. In [Patiniotakis et al. \[2014\]](#), a cloud service recommender (called PuLSaR) is proposed as a multi-criteria decision making approach to compare and recommend cloud services to customers. The selection is based on the customer's preferences towards the services metrics. The recommender system uses the notion of imprecise metrics along with precise metrics to capture the fuzzy or linguistic values provided as requirements by the customer. Then a fuzzy analytic hierarchical process is used for comparison and ranking. In [Michon et al. \[2017\]](#), the QBrokerage algorithm is proposed based on a genetic algorithm to select cloud providers. First, the cloud application is modelled as a directed graph in which the vertices represent virtual machines and edges represent the communication paths between virtual machines. Second, the graph is passed to a mapping component which searches for services that satisfy the requirements of the VM.

5.4 Summary

The three research questions posed earlier (Section 3.2) have thus been answered in systematically reviewing the literature. First, Section 5.1 distinguished four main concerns as the key motivators for the design of cloud brokers: support for making decisions in an increasingly high-dimensional space, the notion of vendor lock-in which calls for less tethering with any particular CSP, managing resource utilisation to meet increasing and dynamic user requirements, as well as other problem-specific motivations such as interoperability and legal issue. Second, Section 5.2 mapped out the functionalities observed in the selected research works. These functionalities seek to provide more

control and feedback to CSCs in order to consume cloud services in a more optimised and tractable manner. The nature of these functionalities is, perhaps, unexpected as the cloud market has been fiercely driven by the major CSPs [Barker et al. 2014]. However, the surveyed literature shows a number of strategies to support CSC decisions, and means of enabling more mobility between CSPs. This is centered around increasing awareness (monitoring, decision support), actuation (policy enforcement, deployment, migration), and increasing freedom through intermediation (abstraction, interoperability, SLA negotiation). Third, and finally in Section 5.3, the surveyed literature exhibits a wide range of employed approaches in how cloud brokers are engineered. This includes explicit approaches such as operating marketplaces for offering and bidding on resources, and implicit ones where programming toolkits are used to amass suitable resources from across the CSP landscape. Other discussed approaches also include programming frameworks and middleware solutions for the commissioning and management of cloud resources, ontological frameworks for defining and identifying heterogeneous cloud resources, and a myriad of resource allocation algorithms.

6 CHARACTERISATION AND DISCUSSION OF THE STATE OF THE ART

This section reflects on the selected studies as a representation of the state of the art in cloud brokerage. It first compiles and summarises the above results through a characterisation framework then it presents general observations and discuss a number of limitations we identified in the literature.

6.1 Characterisation Framework

We now give an overall characterisation of the selected studies in Table 4. To achieve this, we identified a framework of 15 main characteristics that differentiate the surveyed works. These are as follows:

- (1) *Motivation* – the motivation for proposing a cloud broker: dimensionality (D), vendor lock-in (V), meeting requirements (M), or pathological motivations (P).
- (2) *Provisioning Level* – the type of resources that the system brokers: HaaS (H), IaaS (I), PaaS (P), SaaS (S), or unspecified (U).
- (3) *Comparison Framework* – if a standard index or ontology is used to differentiate cloud vendors (e.g., OCCI), if the authors developed their own ontology (Own), or none is adopted (*blank*).
- (4) *API Abstraction* – whether the broker exposes a unified API with which developers can control resources at CSPs, (e.g., libcloud), their own (Own), or not (*blank*).
- (5) *Bidding Style* – refers to the strategy used by the broker to select resources, and it could be:
 - Resource-oriented (C) where the user indicates a minimum specification of computational resources;
 - Deadline-driven (D) where the user provides a target for executing a set of jobs within, possibly in addition to minimum computational resource requirements;
 - QoS-oriented (Q) where the user is more interested in maintaining certain levels of QoS rather than total execution time;
 - Budget-based (B) where monetary or energy budgets are declared or identified, which the scheduler needs to remain within (*i.e.* hard budget) or tries to remain within (*i.e.* soft budget), possibly in addition to minimum computational resource requirements;
 - Resale (R) where the broker buys spare resources from vendors and resells to the users;
 - Marketplace (M) where both providers and users place their bids and the broker matches the available buying and selling prices.

- (6) *Monitoring Level* – the monitoring granularity: hardware level (H), VM level (V), appliance or service level (A), or none (*blank*).
- (7) *Policy Enforcement* – if the broker ensures the execution of user-defined resource procurement and management policies (✓) or not (*blank*).
- (8) *SLA Negotiation* – if the broker offers any mechanisms to negotiate SLA specifics with the vendor on behalf of the user (✓) or not (*blank*).
- (9) *Network Costed* – whether the cost of deploying to a new cloud provider include the network latency and bandwidth utilisation costs incurred (✓) or not (*blank*).
- (10) *Application Deployment* – whether the broker handles deployment to different cloud providers (✓), or if it merely acts a consultation service that offers recommendations (*blank*).
- (11) *Migration* – whether the broker provides support for migrating between different cloud providers (✓), or not (*blank*).
- (12) *VM Interoperability* – if the broker offers tools to convert VMs or other execution units between different vendors (✓), or not (*blank*).
- (13) *Lifecycle Management* – if the broker directly manages the creation, maintenance and graceful destruction of VMs or other execution units (✓), or not (*blank*).
- (14) *Evaluation* – how the proposed solution is tested: on a number of real providers (R), in a testbed / private cloud (T), using simulation / replay of metrics (S), or a prototype / proof of concept (P).
- (15) *Source Available* – if the source code is released under an open source license (✓with a hyperlink), available upon request (R), or non-specified (*blank*).

6.2 General Observations

In characterising the selected studies, the following have been observed.

The **motivational factor** for most research works ($\approx 67\%$) involves designing a solution to meet user-specific requirements such as auto-scaling, resource provisioning, VM placements, cloud bursting and support for multiple clouds. Moreover, 8 of the works also provide support for decision making for the provider as well as service selection. These trends are also visualised in Figure 6 indicating 14 papers ($\approx 42\%$) that take account of the dimensionality aspect. Further to that, 8 of the 14 papers also take into consideration fulfilment of customer-specific requirements, and only 1 paper addresses vendor lock-in.

Interoperability is a concern only recently addressed. Early brokerage attempts focused mainly on the relatively simple challenge of cloudbursting (*i.e.* saturate private cloud resources before adding public cloud resources if/when necessary) *e.g.*, [Aleksiev et al. 2013; den Bossche et al. 2013; Javadi et al. 2012]. Other early attempts employed rudimentary selection mechanisms, using round robin and similarly naïve techniques, *e.g.*, [Itani et al. 2012]. Moreover, many were scheduling oriented and deadline driven and did not implement policy enforcement or SLA negotiation.

Basic bidding prevails. To further elaborate on the previous point, we look into the distribution of bidding styles. Most works (12 $\approx 36\%$) follow a QoS-oriented model. This is closely followed by naïve resource-oriented methods (11 $\approx 33\%$) that offer a minimum resource provisioning level. More sophisticated models such as Marketplace are only observed in more recent works. The Resale bidding style is surprisingly missing from the selected works. We have come across this approach in a number of works that did not make it past the screening phase mainly because they are designed for homogeneous cloud federations.

Table 4. Summary of accepted papers.

Reference	Motivation	Provisioning Level	Comparison Framework	API Abstraction	Bidding Style	Monitoring Level	Policy Enforcement	SLA Negotiation	Network Costed Application Deployment Migration	VM Interoperability	Lifecycle Management	Evaluation	Source Available
1 [Dastjerdi et al. 2010]	D	I	WSMO		Q	A		✓	✓			P	
2 [Nair et al. 2010]	M	I		Own	Q	V	✓	✓	✓	✓	✓	P	✓
3 [Lucas-Simarro et al. 2011]	M	I		deltacloud	C	V			✓			S	
4 [Ferrer et al. 2012]	MP	I		Own	Q	V	✓	✓	✓	✓	✓	P	✓
5 [Itani et al. 2012]	M	I			C		✓					S	
6 [Pawluk et al. 2012]	DM	P	SMI	deltacloud	C	V	✓		✓		✓	R	✓
7 [Javadi et al. 2012]	M	I			D	V			✓			S	
8 [Yang et al. 2012]	MP	H			Q	HA	✓	✓	✓	✓	✓	P	
9 [Calheiros et al. 2012]	M	I			C	V		✓	✓			RT	
10 [Elkhatib et al. 2013]	VMP	I	Own	jclouds	Q	V	✓		✓		✓	P	
11 [Jrad et al. 2013]	M	I	OCCI		Q	V						S	
12 [Desair et al. 2013]	V	P		Own	C	A	✓		✓			P	
13 [Amato et al. 2013]	DM	SPI	OCCI		MBCD			✓	✓			S	
14 [Aleksiev et al. 2013]	MP	I			Q	V	✓				✓	T	✓
15 [den Bossche et al. 2013]	DM	I			D	V			✓			T	R
16 [Kertész et al. 2014]	M	I	Own		Q	A		✓	✓			S	
17 [Paraiso et al. 2014]	V	SP	OASIS SCA	FraSCAti	Q	A	✓		✓		✓	RT	
18 [Munteanu et al. 2014]	DP	PI	Own	mOSAIC	M		✓	✓	✓	✓	✓	P	✓
19 [Copil et al. 2014]	DM	PI	Own		C	A			✓			P	
20 [Somasundaram et al. 2014]	P	I	Own		D	V		✓	✓			TS	
21 [Patiniotakis et al. 2014]	D	U	SMI,USDL		C		✓					P	
22 [Chichin et al. 2014]	D	U			M		✓	✓				S	
23 [Yangui et al. 2014]	D	PI	OCCI	Own	C	V	✓	✓	✓	✓	✓	P	
24 [Lucas-Simarro et al. 2015]	VM	I		deltacloud	C	V			✓			S	
25 [Kim et al. 2015]	M	I			BQ	V		✓	✓			T	
26 [Weerasiri et al. 2015]	DM	SP		Own	C				✓			P	✓
27 [Apuhan et al. 2015]	M	I	Own		B	V		✓				S	
28 [Sharma et al. 2015]	M	I		Own	M	V	✓	✓	✓	✓	✓	PS	
29 [Javed et al. 2016]	DM	I			M			✓				S	
30 [Quarati et al. 2016]	D	I	OCCI	DCI Bridge	B	A	✓		✓		✓	S	
31 [Aazam and Huh 2017]	V	U		Own	B			✓	✓		✓	S	
32 [Michon et al. 2017]	DM	I			Q	V	✓		✓			RT	✓
33 [Anastasi et al. 2017]	DM	I			Q				✓			S	

IaaS is the most commonly addressed provisioning level. Most work focuses on brokerage at the IaaS level, with a few works also including PaaS and/or SaaS, and only one solution to brokerage for HaaS. This is unsurprising for a few reasons. First, IaaS offers the most flexibility in terms of software and hardware resources. Typically, the user selects from a range of operating system templates and hardware configurations and is responsible for configuring the system themselves. Because operating system templates and hardware specifications are relatively homogeneous among CSPs, offers can be simply and accurately compared at the IaaS level. For example, it is trivial to identify offers with, say, Ubuntu Server 16.04, 2 cores and 4GB RAM as opposed to any environment provisioned at the PaaS level. Second, the level of control afforded to IaaS users empowers them to migrate between service providers more easily. Third, brokering at the IaaS level offers the most room for economic benefit. Resources provided at higher levels come at a premium and as such the margins for reducing cost across providers becomes thinner at PaaS and SaaS levels. For the same reasons, monitoring is also mostly done at the IaaS level. Consequently, the most common level of granularity is the VM.

Standards and common libraries are not commonly used. Most papers do not use an ontology to compare cloud service offerings. Those that do tend not to follow standards. The most prevalent and mature ontology, OCCI, is only employed by four studies, while six create a bespoke comparison framework. The lack of enthusiasm to implement standards is not restricted to service comparison frameworks; API abstraction frameworks including common open source libraries, *e.g.*, libcloud and jclouds, are not at all widely adopted. Despite their pitfalls (*cf.* [Elkhatib 2016]), these libraries are fairly mature by the time of the majority of these works are published. As such, this reinventing of the wheel is surprising as some authors choose to develop very similar API abstraction libraries that have no discernible benefits over open source community efforts.

Different interpretations of delegation exist. The characterisation framework provides a comprehensive overview of the challenges deemed most pressing in cloud brokerage. It is clear that deployment is a central functionality for most works: 25 papers ($\approx 76\%$) provide a solution to deploy VMs or appliances. This seems to indicate that delegation is a key priority. However, many of these works fail to provide delegation beyond deployment. For instance, only 11 works ($\approx 33\%$) provide post-deployment life cycle management, and only 6 ($\approx 18\%$) tackle migration.

The legacy of the grid remains. There is a significant number of works that either explicitly or implicitly emerged from previous projects and research groups focussing on grids and HPC computing infrastructures (*e.g.*, [Javadi et al. 2012; Somasundaram et al. 2014]). It is, however, unfortunate that the majority of these works are mere reapplication of old technology with little consideration of the different nature of the cloud resources, such as access policies and management strategies.

Network blindness is very common. Network-related aspects are often ignored in the literature: only 5 papers ($\approx 15\%$) consider network costs. It is worth noting that only 2 of the works that propose migration support incorporate network costs into their solutions.

Real-world tests are limited. The majority of work rely on simulations or prototypes for verifying their solutions. For instance, 12 works ($\approx 36\%$) provide proof of concepts through prototype implementation, and 14 works ($\approx 42\%$) involve simulation-based evaluation. Only one piece of research uses both. The notable point is that only 5 works ($\approx 15\%$) evaluate solutions offered on the real cloud. This could be attributed to the costs associated with evaluating on public cloud resources and the ease of use associated with widely used simulators such as CloudSim.

6.3 Critical Reflections on the State of the Art

By reflecting on the state of the art in both its details (Sections 5.1–5.3) and using the characterisation framework (Table 4 and the rest of Section 6.1), a number of limitations have been identified that are now discussed.

Interoperability support. Little has been done to tackle interoperability challenges. This is evidenced by the sparse adoption of mapping and translation libraries to abstract away differences between service providers' APIs. In addition, many frameworks assume that CSPs will adopt a common API so that they can collaborate to satisfy users' requirements. In reality, the cloud market is has demonstrated its resistance to allowing customers to freely move between competitors.

Comparability approaches. The vast majority of literature assumes that data published by CSPs is comparable. However, it is obvious that the services offered are fundamentally heterogeneous; each CSP has its own description of the offerings. Similar to the above, some approaches assume that the CSPs will adopt standard offering descriptions (e.g., OCCI) which is also unrealistic.

Adaptive deployment. Among frameworks that handle deployment, very few handle migration. This is, perhaps, unsurprising as such operation is non-trivial. However, this reduces the majority of the contributions to consultancy services rather than fully developed brokers. Furthermore, the approaches that support migration do it at the 'heavy' VM level. No brokerage solution currently takes advantage of the migration capabilities offered by container technologies.

Intelligence. The selection of CSPs and instances is based only on matching services to requirements. In other words, the selection is based on what the CSPs promise to provide not on the actual provisioning of resources. Recent analysis of some cloud instances show surprising results of inconsistent performance of the promised offerings. For example, as has been reported in [Samreen et al. 2016], the performance of Amazon c4.xlarge instance is quite the same as the performance of c4.large although the former is twice both in specification and cost of the latter. This realistic performance and cost issues cannot be observed by basing the selection only on the offerings' specifications, which can result in inability to meet the users' requirements or/and 'unfair' costs.

Customer assistance. The vast majority of the research assumes that CSCs are acutely aware of their applications' technical requirements *before* deployment. In fact, it is very difficult for users to ascertain the necessary hardware specifications in advance [Lango 2013]. This is particularly true when demand for the application is expected to be volatile, as is often the case when IaaS is chosen above bare metal.

Realistic evaluation. Evaluation is carried out mostly using simulations or using testbeds made up of private cloud resources. Several papers suggested that this is due to the costs of using public cloud services. Indeed it does cost money to carry out benchmarks, develop use cases, and evaluate prototypes. However, we know from experience of our own experiments [Samreen et al. 2016] that such costs in total are well below the price of purchasing comparable hardware that could be used as private cloud resources. Moreover, several CSPs (such as Amazon and Microsoft) offer programmes to fund researchers with credits to use their cloud services for research purposes. This observation, coupled with the fact that API interoperability and open source releases are ignored by many works, leads us to suspect that most works are merely proof of concept ideas that are of little practical use to other researchers and practitioners. In other words, the literature still lacks pragmatic brokers that are evaluated on real cloud infrastructures, and that are usable by others.

7 LIMITATIONS AND THREATS TO VALIDITY

We followed the systemic approach of [Kitchenham 2004] to avoid selection bias rather than relying only on our own knowledge and background. As researchers from different communities and backgrounds tend to use different terminologies for the same topics, we performed the search using

as many terms as we know that are related to cloud brokerage, and results from all publication databases have been inspected accordingly. Our survey's search was concluded in December 2017, so papers appearing after that date would not have been captured. Reflecting on this methodology, the study is potentially subject to the following limitations. We describe our mitigation strategy associated with each.

Sampling bias. The initial search phase was based on publication meta-data (*i.e.* abstract, title, and keywords). It is possible that studies that have addressed brokerage in cloud computing have been overlooked by this method if none of their meta-data mention our search terms. However, this is unlikely as our search terms were rather broad, as evidenced by the huge search return in Phase 1 (Table 3). Furthermore, as meta-data are provided by the authors of the selected works, we believe it is reasonable to rely on the quality of the classification and indexing of the papers by the publishing databases.

Reporting bias and construct validity. Our reporting of the review was based on a set of predefined research questions and guided by a deterministic list of inclusion and exclusion criteria. The selected studies were reported in light of the research questions, using thorough understanding of the different works to answer the research questions and in turn untangle the overlap between the studies. Works were also discussed between the researchers carrying out the survey, reducing chances of misrepresentation and, subsequently, further reducing threats to construct validity.

Data extraction and internal validity. The labour-intensive nature of this research could potentially contribute to inaccuracy in data extraction. We designed our methodology (described in section 3.3.2) to minimise the possibility of human error in the data extraction process. Our core methodology was complemented by extensive and profound discussions among this paper's authors to ensure coherent and correct extraction of data.

Reproducibility. The study could be easily replicated using the search and selection strategies we outlined. However, full reproducibility of our work hinges on automated means of data collection. We found automated data collection to be more difficult to obtain than we had expected. First, most online libraries do not expose public web APIs and some even forbid web scraping. For those that do offer APIs, the search capabilities are rather limited and divergent¹⁴. Some APIs are restricted to paying customers and partners only; *e.g.*, IEEE and Wiley. ACM's API was extremely rudimentary and limiting, requiring significant post-processing of the retrieved results. Consequently, investment into automated development is a significant effort.

External validity. Our survey was concerned with peer-reviewed academic works only, and is only generalisable for such type of brokerage works only. We did not attempt to capture industrial efforts that are not published through academic research channels, although we are aware of a number of such works. This decision was made in order to avoid systematic errors and maintain the quality at which the survey is conducted.

8 CONCLUSIONS AND FUTURE DIRECTIONS

This study set out to build an understanding of different solutions in the burgeoning field of cloud brokerage. For this, we followed a systematic literature survey in order to ensure thorough coverage of such solutions. We aimed to accurately represent the state-of-the-art in cloud brokerage and identify key accomplishments and challenges. Seven main publication databases were considered to perform the search. We considered 10,847 papers from which we found 33 papers exhibiting acceptable relevance to the survey topic.

¹⁴We found arXiv's API to be the most advanced in terms of functionality and reliability of results as search terms change, with those of PLOS, ScienceDirect and Springer close behind.

The main findings show that brokerage in the cloud emerged as a cross-cloud model that is motivated by the heterogeneity and dimensionality of the current cloud services in addition to the limitation of the single-cloud paradigm to satisfy the customers requirements. The proposed cloud brokers are intended to perform a number of tasks related to supporting customers' decisions making, application deployment, SLA negotiations, and resources monitoring, among others. These brokers have been engineered using diverse approaches including middlewares, toolkits, frameworks, and others. Section 5.3 has provided some details about the methods and techniques used to clarify the ways in which brokers are realised. The limitations of the current cloud brokers have also been identified. In general, we observed that most proposals implement a subset of the functionality typically associated with a broker. We see this normal as the field is still in its formative stage. The field needs further work to assist CSCs in specifying their applications' requirements, adaptation and intelligent decision making related to the selection of the cloud providers and services.

Based on our investigation and reflection, we identify a number of future avenues in the field of cloud brokerage:

Customer Assistance. As already mentioned in Section 6.3, little has been done to assist customers in specifying the necessary low level technical requirements that best suits their high level requirements. In this context, further work is needed to translate high level objectives into cloud service specifications. This can be achieved by, for example, developing Domain Specific Languages (DSLs) that capture customers Service Level Objectives (SLOs) and translate them into measurable operational goals. Another approach is to learn application requirements at runtime based on the actual performance of applications relative to the cloud services in which they reside. Learnt models can then be used to inform the low level requirements of similar applications. This also requires developing methodologies and measures to define and quantify the similarity of the cloud applications. Furthermore, a trace-and-replay approach is another option to assist in specifying the applications requirements [Li et al. 2011]. However, reducing the costs of adopting such approach needs to be taken into consideration.

Adaptive and Fluid Deployment. Containers are lightweight execution units that provide an efficient alternative to 'classical' VMs in terms of resource utilisation. The layer of abstraction offered by containers makes CSC-centric migration a promising approach to adapt applications deployment. However, this is challenged by the need for methodologies to benchmark the containers so that the heterogeneous resources can be compared and ranked [Varghese et al. 2016]. Furthermore, the fact that containers do not provide same level of isolation as VMs (containers share the operating system) means that the degree of performance interference is higher compared to the VMs [Sharma et al. 2016], which makes satisfying the SLA more challenging. Other challenges relate to the management of the container lifecycle in the cross-cloud context and quantifying the cost of migration, among others.

Intelligent Decision Making. Currently, decision making systems rely heavily on simplistic views of application requirements and service provider performance. Machine-learning techniques can help to quantify the extent to which providers fulfil their offerings and the extent to which the applications are satisfied in practice. Also, machine learning models can be deployed to adapt deployments proactively, *i.e.* before the adaptation becomes necessary. A major challenge here is to accurately predict when requirements will change and the feasibility adapting the deployment at that time. Furthermore, feedback on the performance of the adopted learning approaches is necessary as learning in different ways can lead to different results.

ACKNOWLEDGMENTS

This work is partly supported by the Adaptive Brokerage for the Cloud (ABC) project, UK EPSRC grant EP/R010889/1.

REFERENCES

- Mohammad Aazam and Eui-Nam Huh. 2017. Cloud broker service-oriented resource management model. *Transactions on Emerging Telecommunications Technologies* 28, 2 (2017). <https://doi.org/10.1002/ett.2937>
- Giuseppe Aceto, Alessio Botta, Walter de Donato, and Antonio Pescapé. 2013. Cloud monitoring: A survey. *Computer Networks* 57, 9 (2013), 2093 – 2115. <https://doi.org/10.1016/j.comnet.2013.04.001>
- Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah, and Philippe Merle. 2018. Elasticity in Cloud Computing: State of the Art and Research Challenges. *IEEE Transactions on Services Computing* 11, 2 (March 2018), 430–447. <https://doi.org/10.1109/TSC.2017.2711009>
- Bandar Aldawsari, Thar Baker, and David England. 2015. Towards a Holistic Multi-cloud Brokerage System: Taxonomy, Survey, and Future Directions. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. 1467–1472. <https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.219>
- Tyanko Aleksiev, Simon Barkow-Oesterreicher, Peter Z. Kunszt, Sergio Maffioletti, Riccardo Murri, and Christian Panse. 2013. VM-MAD: a cloud/cluster software for service-oriented academic environments. *CoRR abs/1302.2529* (2013). <http://arxiv.org/abs/1302.2529>
- Alba Amato, Beniamino Di Martino, and Salvatore Venticinque. 2013. Cloud Brokering as a Service. In *8th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*. 9–16. <https://doi.org/10.1109/3PGCIC.2013.9>
- Gaetano F. Anastasi, Emanuele Carlini, Massimo Coppola, and Patrizio Dazzi. 2017. QoS-aware Genetic Cloud Brokering. *Future Generation Computer Systems* 75 (2017), 1 – 13. <https://doi.org/10.1016/j.future.2017.04.026>
- Bernady Apduhan, Muhammad Younas, and Toshihiro Uchibayashi. 2015. *Improving Reliability and Availability of IaaS Services in Hybrid Clouds*. Springer International Publishing, 557–568. https://doi.org/10.1007/978-3-319-21404-7_41
- AppDirect. 2017. AppDirect SMB Cloud Service Adoption Report. <https://dcnxfkgt2gjxz.cloudfront.net/Files/AppDirect-SMB-Survey-Report-2017.pdf>
- Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. 2010. A View of Cloud Computing. *Commun. ACM* 53, 4 (April 2010), 50–58. <https://doi.org/10.1145/1721654.1721672>
- Marcio Miranda Assis and Luiz Fernando Bittencourt. 2016. A survey on cloud federation architectures: Identifying functional and non-functional properties. *Journal of Network and Computer Applications* 72 (2016), 51 – 71. <https://doi.org/10.1016/j.jnca.2016.06.014>
- Adam Barker, Blesson Varghese, and Long Thai. 2015. Cloud Services Brokerage: A Survey and Research Roadmap. *CoRR abs/1506.00485* (2015). <http://arxiv.org/abs/1506.00485>
- Adam Barker, Blesson Varghese, Jonathan Stuart Ward, and Ian Sommerville. 2014. Academic Cloud Computing Research: Five Pitfalls and Five Opportunities. In *6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14)*. USENIX Association. <https://www.usenix.org/conference/hotcloud14/workshop-program/presentation/barker>
- Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. 2011. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Softw. Pract. Exper.* 41, 1 (Jan. 2011), 23–50. <https://doi.org/10.1002/spe.995>
- Rodrigo N. Calheiros, Ayed Nadjaran Toosi, Christian Vecchiola, and Rajkumar Buyya. 2012. A coordinator for scaling elastic applications across multiple clouds. *Future Generation Computer Systems* 28, 8 (2012), 1350 – 1362. <https://doi.org/10.1016/j.future.2012.03.010>
- Sergei Chichin, Mohan Baruwat Chhetri, Quoc Bao Vo, Ryszard Kowalczyk, and Marcin Stepniak. 2014. Smart Cloud Marketplace – Agent-Based Platform for Trading Cloud Services. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 3. 388–395. <https://doi.org/10.1109/WI-IAT.2014.193>
- Georgiana Copil, Daniel Moldovan, Hong-Linh Truong, and Schahram Dustdar. 2014. On Controlling Cloud Services Elasticity in Heterogeneous Clouds. In *Conference on Utility and Cloud Computing (UCC)*. 573–578. <https://doi.org/10.1109/UCC.2014.88>
- Cloud Standards Coordination (CSC). 2016. *CSC Phase 2: Cloud Computing Users Needs - Analysis, conclusions and recommendations from a public survey*. Special Report 003 381 V2.1.1. The European Telecommunications Standards Institute (ETSI). 12–19 pages. <http://csc.etsi.org/phase2/UserNeeds.html>
- Amir Vahid Dastjerdi, Sayed Gholam Hassan Tabatabaei, and Rajkumar Buyya. 2010. An Effective Architecture for Automated Appliance Management System Applying Ontology-Based Cloud Discovery. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. 104–112. <https://doi.org/10.1109/CCGRID.2010.87>

- Ruben Van den Bossche, Kurt Vanmechelen, and Jan Broeckhove. 2013. Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Generation Computer Systems* 29, 4 (2013), 973 – 985. <https://doi.org/10.1016/j.future.2012.12.012> Special Section: Utility and Cloud Computing.
- Tom Desair, Wouter Joosen, Bert Lagaisse, Ansar Rafique, and Stefan Walraven. 2013. Policy-driven Middleware for Heterogeneous, Hybrid Cloud Platforms. In *12th International Workshop on Adaptive and Reflective Middleware (ARM)*. ACM, New York, NY, USA, Article 2, 6 pages. <https://doi.org/10.1145/2541583.2541585>
- Mona Eisa, Muhammad Younas, Kashinath Basu, and Hong Zhu. 2016. Trends and Directions in Cloud Service Selection. In *IEEE Symposium on Service-Oriented System Engineering (SOSE)*. 423–432. <https://doi.org/10.1109/SOSE.2016.59>
- Yehia Elkhatib. 2016. Mapping Cross-Cloud Systems: Challenges and Opportunities. In *8th Conference on Hot Topics in Cloud Computing*. USENIX Association, 77–83.
- Yehia Elkhatib, Gordon S Blair, and Bholanathsingh Surajbali. 2013. Experiences of Using a Hybrid Cloud to Construct an Environmental Virtual Observatory. In *3rd Workshop on Cloud Data and Platforms*. 13–18. <https://doi.org/10.1145/2460756.2460759>
- Yehia Elkhatib, Faiza Samreen, and Gordon S. Blair. 2018. Same Same, but Different: A Descriptive Differentiation of Intra-cloud IaaS Services. *CoRR* abs/1802.03641 (2018). arXiv:1802.03641 <http://arxiv.org/abs/1802.03641>
- Ana Juan Ferrer, Francisco Hernández, Johan Tordsson, Erik Elmroth, Ahmed Ali-Eldin, Csilla Zsigri, Raúl Sirvent, Jordi Guitart, Rosa M. Badia, Karim Djemame, Wolfgang Ziegler, Theo Dimitrakos, Sriji K. Nair, George Kousiouris, Kleopatra Konstanteli, Theodora Varvarigou, Benoit Hudzia, Alexander Kipp, Stefan Wesner, Marcelo Corrales, Nikolaus Forgó, Tabassum Sharif, and Craig Sheridan. 2012. OPTIMIS: A holistic approach to cloud service provisioning. *Future Generation Computer Systems* 28, 1 (Jan. 2012), 66 – 77. <https://doi.org/10.1016/j.future.2011.05.022>
- Nadir Ghrada, Mohamed Faten Zhani, and Yehia Elkhatib. 2018. Price and Performance of Cloud-hosted Virtual Network Functions: Analysis and Future Challenges. In *IEEE Workshop on Approaches, Analyses, and Performance Issues in Virtualized Environments and Software Defined Networking (PVE-SDN)*.
- Nikolay Grozev and Rajkumar Buyya. 2014. Inter-Cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience* 44, 3 (2014), 369–390. <https://doi.org/10.1002/spe.2168> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/spe.2168>
- Pei-Fang Hsu, Soumya Ray, and Yu-Yu Li-Hsieh. 2014. Examining cloud computing adoption intention, pricing mechanism, and deployment model. *International Journal of Information Management* 34, 4 (2014), 474–488. <https://doi.org/10.1016/j.ijinfomgt.2014.04.006>
- Iliana Iankoulova and Maia Daneva. 2012. Cloud computing security requirements: A systematic review. In *6th International Conference on Research Challenges in Information Science (RCIS)*. 1–7. <https://doi.org/10.1109/RCIS.2012.6240421>
- Wassim Itani, Cesar Ghali, Ramzi Bassil, Ayman Kayssi, and Ali Chehab. 2012. BGP-inspired Autonomic Service Routing for the Cloud. In *27th Annual ACM Symposium on Applied Computing (SAC)*. ACM, New York, NY, USA, 406–411. <https://doi.org/10.1145/2245276.2245356>
- Pooyan Jamshidi, Aakash Ahmad, and Claus Pahl. 2013. Cloud Migration Research: A Systematic Review. *IEEE Transactions on Cloud Computing* 1, 2 (July 2013), 142–157. <https://doi.org/10.1109/TCC.2013.10>
- Bahman Javadi, Jemal Abawajy, and Rajkumar Buyya. 2012. Failure-aware resource provisioning for hybrid Cloud infrastructure. *J. Parallel and Distrib. Comput.* 72, 10 (Oct. 2012), 1318 – 1331. <https://doi.org/10.1016/j.jpdc.2012.06.012>
- Barkha Javed, Peter Bloodsworth, Raihan Ur Rasool, Kamran Munir, and Omer Rana. 2016. Cloud Market Maker: An automated dynamic pricing marketplace for cloud users. *Future Generation Computer Systems* 54 (2016), 52 – 67. <https://doi.org/10.1016/j.future.2015.06.004>
- Sam Johnston. 2014. Simple Workload & Application Portability (SWAP). In *1st International Workshop on CrossCloud Systems*. IEEE. <https://doi.org/10.1109/INFCOMW.2014.6849165>
- Foued Jrad, Jie Tao, and Achim Streit. 2013. A Broker-based Framework for Multi-cloud Workflows. In *International Workshop on Multi-cloud Applications and Federated Clouds (MultiCloud)*. ACM, New York, NY, USA, 61–68. <https://doi.org/10.1145/2462326.2462339>
- Amin Jula, Elankovan Sundararajan, and Zalinda Othman. 2014. Cloud computing service composition: A systematic literature review. *Expert Systems with Applications* 41, 8 (2014), 3809 – 3824. <https://doi.org/10.1016/j.eswa.2013.12.017>
- Kiranbir Kaur, Sandeep Sharma, and Karanjeet Singh Kahlon. 2017. Interoperability and Portability Approaches in Inter-Connected Clouds: A Review. *Comput. Surveys* 50, 4, Article 49 (Oct. 2017), 40 pages. <https://doi.org/10.1145/3092698>
- Attila Kertész, Gabor Kecskemeti, and Ivona Brandic. 2014. An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous Cloud environments. *Future Generation Computer Systems* 32 (2014), 54 – 68. <https://doi.org/10.1016/j.future.2012.05.016> Special Section: The Management of Cloud Systems.
- Ali Khajeh-Hosseini, David Greenwood, James W. Smith, and Ian Sommerville. 2012. The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. *Software: Practice and Experience* 42, 4 (2012), 447–465. <https://doi.org/10.1002/spe.1072>

- Ali Khajeh-Hosseini, Ian Sommerville, Jurgen Bogaerts, and Pradeep Teregowda. 2011. Decision Support Tools for Cloud Migration in the Enterprise. In *IEEE 4th International Conference on Cloud Computing*. 541–548. <https://doi.org/10.1109/CLOUD.2011.59>
- Woo-Joong Kim, Dong-Ki Kang, Seong-Hwan Kim, and Chan-Hyun Youn. 2015. Cost Adaptive VM Management for Scientific Workflow Application in Mobile Cloud. *Mobile Networks and Applications* 20, 3 (2015), 328–336. <https://doi.org/10.1007/s11036-015-0593-4>
- Barbara Kitchenham. 2004. *Procedures for performing systematic reviews*. Technical Report TR/SE-0401. Keele University.
- Jason Lango. 2013. Toward Software-defined SLAs. *Queue* 11, 11, Article 20 (Nov. 2013), 20:20–20:31 pages. <https://doi.org/10.1145/2557963.2560948>
- Neal Leavitt. 2009. Is Cloud Computing Really Ready for Prime Time? *Computer* 42, 1 (Jan. 2009), 15–20. <https://doi.org/10.1109/MC.2009.20>
- Ang Li, Xiaowei Yang, Srikanth Kandula, and Ming Zhang. 2010. CloudCmp: Comparing Public Cloud Providers. In *10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. ACM, New York, NY, USA, 1–14. <https://doi.org/10.1145/1879141.1879143>
- Ang Li, Xuanran Zong, Srikanth Kandula, Xiaowei Yang, and Ming Zhang. 2011. CloudProphet: Towards Application Performance Prediction in Cloud. In *ACM SIGCOMM Conference (SIGCOMM '11)*. ACM, New York, NY, USA, 426–427. <https://doi.org/10.1145/2018436.2018502>
- Nikos Loutas, Eleni Kamateri, Filippo Bosi, and Konstantinos A Tarabanis. 2011. Cloud Computing Interoperability: The State of Play. In *IEEE 3rd International Conference on Cloud Computing Technology and Science (CloudCom)*. 752–757. <https://doi.org/10.1109/CloudCom.2011.116>
- Jose Luis Lucas-Simarro, Rafael Moreno-Vozmediano, Ruben S. Montero, and Ignacio M. Llorente. 2011. Dynamic placement of virtual machines for cost optimization in multi-cloud environments. In *International Conference on High Performance Computing and Simulation (HPCS)*. 1–7. <https://doi.org/10.1109/HPCSim.2011.5999800>
- Jose Luis Lucas-Simarro, Rafael Moreno-Vozmediano, Ruben S. Montero, and Ignacio M. Llorente. 2015. Cost optimization of virtual infrastructures in dynamic multi-cloud scenarios. *Concurrency and Computation: Practice and Experience* 27, 9 (2015), 2260–2277. <https://doi.org/10.1002/cpe.2972>
- Sunilkumar S. Manvi and Gopal Krishna Shyam. 2014. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. *Journal of Network and Computer Applications* 41 (2014), 424 – 440. <https://doi.org/10.1016/j.jnca.2013.10.004>
- Rajat Mehrotra, Srishti Srivastava, Ioana Banicescu, and Sherif Abdelwahed. 2016. Towards an autonomic performance management approach for a cloud broker environment using a decomposition-coordination based methodology. *Future Generation Computer Systems* 54 (2016), 195 – 205. <https://doi.org/10.1016/j.future.2015.03.020>
- Étienne Michon, Julien Gossa, Stéphane Genaud, Léo Unbekandt, and Vincent Kherbache. 2017. Schlouder: A broker for IaaS clouds. *Future Generation Computer Systems* 69 (2017), 11 – 23. <https://doi.org/10.1016/j.future.2016.09.010>
- Victor Ion Munteanu, Călin Șandru, and Dana Petcu. 2014. Multi-cloud resource management: cloud service interfacing. *Journal of Cloud Computing* 3, 1 (2014), 1–23. <https://doi.org/10.1186/2192-113X-3-3>
- Srijith K Nair, Sakshi Porwal, Theo Dimitrakos, Ana Juan Ferrer, Johan Tordsson, Tabassum Sharif, Craig Sheridan, Muttukrishnan Rajarajan, and Afnan Ullah Khan. 2010. Towards Secure Cloud Bursting, Brokerage and Aggregation. In *8th IEEE European Conference on Web Services*. 189–196. <https://doi.org/10.1109/ECOWS.2010.33>
- Fawaz Paraiso, Philippe Merle, and Lionel Seinturier. 2014. soCloud: A service-oriented component-based PaaS for managing portability, provisioning, elasticity, and high availability across multiple clouds. *CoRR* abs/1407.1963 (2014). <http://arxiv.org/abs/1407.1963>
- Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari, and Joaquim Celestino Júnior. 2013. An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of Network and Computer Applications* 36, 1 (2013), 25 – 41. <https://doi.org/10.1016/j.jnca.2012.08.007>
- Ioannis Patiniotakis, Yiannis Verginadis, and Gregoris Mentzas. 2014. Preference-based Cloud Service Recommendation As a Brokerage Service. In *2nd International Workshop on CrossCloud Systems (CCB)*. ACM, New York, NY, USA, Article 5, 6 pages. <https://doi.org/10.1145/2676662.2676677>
- Przemyslaw Pawluk, Bradley Simmons, Michael Smit, Marin Litoiu, and Serge Mankovski. 2012. Introducing STRATOS: A Cloud Broker Service. In *5th International Conference on Cloud Computing*. IEEE Computer Society, Washington, DC, USA, 891–898. <https://doi.org/10.1109/CLOUD.2012.24>
- Alfonso Quarati, Andrea Clematis, and Daniele D’Agostino. 2016. Delivering cloud services with QoS requirements: Business opportunities, architectural solutions and energy-saving aspects. *Future Generation Computer Systems* 55 (2016), 403 – 427. <https://doi.org/10.1016/j.future.2015.02.009>
- Faiza Samreen, Yehia Elkhatib, Matthew Rowe, and Gordon S. Blair. 2016. Daleel: Simplifying cloud instance selection using machine learning. In *IEEE/IFIP Network Operations and Management Symposium*. 557–563. <https://doi.org/10.1109/NOMS.2016.7502858>

- Benjamin Satzger, Waldemar Hummer, Christian Inzinger, Philipp Leitner, and Schahram Dustdar. 2013. Winds of Change: From Vendor Lock-In to the Meta Cloud. *IEEE Internet Computing* 17, 1 (Jan 2013), 69–73. <https://doi.org/10.1109/MIC.2013.19>
- Prateek Sharma, Lucas Chaufourrier, Prashant Shenoy, and Y.C. Tay. 2016. Containers and Virtual Machines at Scale: A Comparative Study. In *17th International Middleware Conference (Middleware '16)*. ACM, New York, NY, USA, Article 1, 13 pages. <https://doi.org/10.1145/2988336.2988337>
- Prateek Sharma, Stephen Lee, Tian Guo, David Irwin, and Prashant Shenoy. 2015. SpotCheck: Designing a Derivative IaaS Cloud on the Spot Market. In *10th European Conference on Computer Systems (EuroSys'15)*. ACM, Article 16, 15 pages. <https://doi.org/10.1145/2741948.2741953>
- Fereshteh Sheikh. and Nima Jafari Navimipour. 2018. Auction-based resource allocation mechanisms in the cloud environments: A review of the literature and reflection on future challenges. *Concurrency and Computation: Practice and Experience* 30, 16 (2018). <https://doi.org/10.1002/cpe.4456>
- Thamarai Selvi Somasundaram, Kannan Govindarajan, Usha Kiruthika, and Rajkumar Buyya. 2014. Semantic-enabled CARE Resource Broker (SeCRB) for managing grid and cloud environment. *The Journal of Supercomputing* 68, 2 (2014), 509–556. <https://doi.org/10.1007/s11227-013-1047-z>
- Le Sun, Hai Dong, Farookh Khadeer Hussain, Omar Khadeer Hussain, and Elizabeth Chang. 2014. Cloud service selection: State-of-the-art and future research directions. *Journal of Network and Computer Applications* 45 (2014), 134 – 150. <https://doi.org/10.1016/j.jnca.2014.07.019>
- Johan Tordsson, Rubén S. Montero, Rafael Moreno-Vozmediano, and Ignacio M. Llorente. 2012. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. *Future Generation Computer Systems* 28, 2 (2012), 358 – 367. <https://doi.org/10.1016/j.future.2011.07.003>
- Blesson Varghese, Lawan Thamsuhang Subba, Long Thai, and Adam Barker. 2016. Container-Based Cloud Virtual Machine Benchmarking. In *IEEE International Conference on Cloud Engineering (IC2E)*. 192–201. <https://doi.org/10.1109/IC2E.2016.28>
- Denis Weerasiri, Boualem Benatallah, and Moshe Chai Barukh. 2015. 20th International Conference on Database Systems for Advanced Applications (DASFAA). Springer International Publishing, 334–350. https://doi.org/10.1007/978-3-319-18120-2_20
- Xiaoyu Yang, Bassem Nasser, Mike Surridge, and Stuart Middleton. 2012. A business-oriented Cloud federation model for real-time applications. *Future Generation Computer Systems* 28, 8 (Oct. 2012), 1158 – 1167. <https://doi.org/10.1016/j.future.2012.02.005> Including Special sections SS: Trusting Software Behavior and SS: Economics of Computing Services.
- Sami Yangui, Iain-James Marshall, Jean-Pierre Laisne, and Samir Tata. 2014. CompatibleOne: The Open Source Cloud Broker. *Journal of Grid Computing* 12, 1 (2014), 93–109. <https://doi.org/10.1007/s10723-013-9285-0>
- Zhi-Hui Zhan, Xiao-Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu-Hung Chung, and Yun Li. 2015. Cloud Computing Resource Scheduling and a Survey of Its Evolutionary Approaches. *Comput. Surveys* 47, 4, Article 63 (July 2015), 33 pages. <https://doi.org/10.1145/2788397>
- Qi Zhang, Lu Cheng, and Raouf Boutaba. 2010. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications* 1, 1 (2010), 7–18. <https://doi.org/10.1007/s13174-010-0007-6>
- Zhizhong Zhang, Chuan Wu, and David W.L. Cheung. 2013. A Survey on Cloud Interoperability: Taxonomies, Standards, and Practice. *SIGMETRICS Performance Evaluation Review* 40, 4 (April 2013), 13–22. <https://doi.org/10.1145/2479942.2479945>

Received April 2018; revised August 2018; accepted August 2018