IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 12, 2020, accepted October 25, 2020, date of publication November 4, 2020, date of current version November 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035816

Optimal Selection Techniques for Cloud Service Providers

GIUSEPPE TRICOMI[1], (Member, IEEE), GIOVANNI MERLINO[1,2], (Member, IEEE), ALFONSO PANARELLO[2], AND ANTONIO PULIAFITO[1,2,3], (Member, IEEE)

[1]Dipartimento di Ingegneria, Università degli Studi di Messina, 98122 Messina, Italy
[2]SmartMe.IO S.r.l., 98121 Messina, Italy
[3]National Interuniversity Consortium for Informatics (CINI), 37139 Rome, Italy

Corresponding author: Giuseppe Tricomi (gtricomi@unime.it)

This work was supported in part by the Italian Project TOOLSMART financed by Open Community PA 2020 – Pon Governance 2014–2020.

ABSTRACT Nowadays Cloud computing permeates almost every domain in Information and Communications Technology (ICT) and, increasingly, most of the action is shifting from large, dominant players toward independent, heterogeneous, private/hybrid deployments, in line with an ever wider range of business models and stakeholders. The rapid growth in the numbers and diversity of small and medium Cloud providers is bringing new challenges in the as-a-Services space. Indeed, significant hurdles for smaller Cloud service providers in being competitive with the incumbent market leaders induce some innovative players to ''federate'' deployments in order to pool a larger, virtually limitless, set of resources across the *federation*, and stand to gain in terms of economies of scale and resource usage efficiency. Several are the challenges that need to be addressed in building and managing a federated environment, that may go under the ''Security'', ''Interoperability'', ''Versatility'', ''Automatic Selection'' and ''Scalability'' labels. The aim of this paper is to present a survey about the approaches and challenges belonging to the ''Automatic Selection'' category. This work provides a literature review of different approaches adopted in the ''Automatic and Optimal Cloud Service Provider Selection'', also covering ''Federated and Multi-Cloud'' environments.

INDEX TERMS Algorithms, cloud federation, matchmaking, multi-cloud, optimal selection, survey.

I. INTRODUCTION

Cloud computing [1] is a powerful paradigm for the delivery of on-demand services in a transparent way to end users, based mostly on a pay-per-use business model. The ''Cloud Providers'' leverage owned computing, storage and networking resources using virtualization technologies, that enable the abstraction of computer hardware resources, thus backing, through multiplexing, a (typically much higher) number of virtual resources. These virtualization capabilities enable any company to *elastically* manage the use of internal Information Technology (IT) assets, avoiding the sizing for all of the company's hardware infrastructure according to the *maximum* number of users possibly engaging the resources concurrently at any time, but rather to a lower, average value of usage. This approach allows to reap several benefits, especially in the medium/long term: reducing expenses invested in the procurement of resources (CAPital EXpenditure, CAPEX), reducing the maintenance of owned assets (OPerating EXpense, OPEX), improving energy saving, and so on. Respecting Service-Level Agreements (SLAs), or satisfying requests for change of SLA by the customers asking for, e.g., additional capabilities, is a requirements for Cloud Service Providers, which are bound to the terms set in the SLAs they advertise for their own services.

Cloud Service Level Agreement Standardisation Guidelines [2] are provided by the European Union (EU) as part of the *European Cloud Computing Strategy* to foster the adoption of the Cloud paradigm by EU businesses. Relevant items include: (i) availability and reliability of the Cloud service; (ii) quality of support services end users are expected to receive from their Cloud provider; (iii) security levels; (iv) management and lifecycle for data kept in the Cloud.

As small and medium enterprises often do not have enough resources to meet peak demand for services (respectively Software, Platform and Infrastructure as a Service SaaS,

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Arafatur Rahman.

PaaS, IaaS, according to the layer), an approach could be to ask for additional resources to the underlying service layer (*delegation*) [3]. However not every provider implements all the service layers, e.g., when not enough economic resources are available to expand their Cloud infrastructure, providers typically have no choice but to *vertically* resort to services available from so-called Mega Providers (Google, Amazon, etc). This is the only way for the small and medium Cloud Providers to ensure Quality of Service (QoS) levels do not sustain degradation. However, there is an alternative, emerging approach, to avoid the aforementioned scenario: the Cloud Federation. Leveraging this configuration of virtual computing infrastructure it is possible to reach higher levels of efficiency in providing service to customers, either private (single users/enterprises) or public organizations.

We define a Cloud Federation as an agreement for the cooperation among medium-sized Cloud providers, enabling them to share computing, storage and networking resources. In other words, by the concept of Cloud Federation, an agreement is implied among several independent providers, to rent virtual resources, such as virtual machines (IaaS) [4], or services (PaaS) [5] too, and so on. In this sense an environment is envisioned where different Clouds, belonging to separate administrative domains, may interact each other, at the same time playing the role of customers, and service providers as well.

The interaction and cooperation, among federated entities in this scenario, are far from being easily implemented, and there is an ongoing effort from the research community to thoroughly investigate these topics. Moreover, guidelines for the design and implementation of the functionalities of a federated environment have not conclusively agreed upon yet.

Anyway, there is a vast repertoire of scientific literature that tries to outline the requirements and challenges to implement a Federation. In Panarello *et al.* [4] and Toosi *et al.* [6] in-depth analyses of the state of the art on the topic of Cloud Federation are presented. The authors in particular highlight the need to automatically detect and choose services and resources available from Cloud providers worldwide, and the main results to be achieved yet. In order to accomplish such a goal there are two possible approaches, namely the **Brokered** or the **Decentralized** ones. A deep explanation about these two subjects with a special focus on their pros and cons is presented in Calcavecchia *et al.* [7].

Moreover, based on the discussed approaches, there is the need to make a basic distinction between the concept of Cloud Federation and Multi-Cloud. What changes mostly is the viewpoint: the first can be considered as a Cloud-centric solution. Several Clouds share their own resources with other federated Clouds. This sharing is totally transparent to the end user.

Conversely the Multi-Cloud represents a user-centric solution where a user is aware about the presence of different Clouds, and either the user or another third party is able to make choices about the selection of the Cloud where services or resources will be instantiated. In Petcu *et al.* [8] a distinction between the concept of Federation and Multi-Cloud is provided. Moreover a partitioning of the Multi-Cloud requirements in three different macro areas namely ‘‘Development, Deployment and Execution’’ has been made.

Focusing on the deployment area, many requirement points have been underlined, some of them concern the ability in the selection of the Clouds' services and resources, use of automated deployment procedures, etc. Thus, methodologies to compare Cloud services in a Federated or Multi-Cloud scenario are needed.

This work extends the existings literature by reviewing challenges and approaches to optimally detect and select the Cloud service providers in a Federated or Multi-Cloud environment, with a special focus on how these aspects have been handled in recent years.

II. EU PROJECTS OVERVIEW

One of the markers under analysis to verify the relevance of the topic related to Cloud Service Provider (CSP) selection in Federation and Multi-Cloud is the number and funds provided by European Commission of projects, in this domain. The following section highlights some of the main European Union (EU) projects, either ongoing or over already, including publications ensuing from such projects, about Federation and Multi-Cloud topics funded in the last 12 years (2007 to 2019).

The projects discussed below are ordered by time and in our opinion, they show how the EU has pursued an approach to contrast the dominance of huge CSPs (Amazon, Google, Azure, etc.). Indeed, as a reader can understand from the evolution of the topics pursued by the projects described below, there was a strong interest in the design of systems that would get value from cooperation among CSPs. Having a look at the topics covered by the projects, we can trace a path that begins with virtualization mechanisms without constraints imposed by the infrastructure, shared by several entities (grid computing systems as ancestors of CSPs), cooperating as if they were on the same administrative domain; then going through a project in which a CSP Federation is able to instantiate reconfigurable private Networks, realized on different environment virtualized on different CSPs, until reaching an integration level that enables the management of resources coming from several CSPs through a federated system.

A. MAIN EU PROJECTS ANALYZED

RESERVOIR: *Resources and Services Virtualization without Barriers* [9] is a European Commission Integrated Project funded by *"Seventh Framework Programme"* (FP7/2006-2013). It started in November 2007 and ran until January 2011. RESERVOIR initially aimed at extending, combining and integrating three different technologies: virtualization, grid computing and business service management (BSM). This research strategy has been very successful within the domain of High Performance Computing, especially for scientific computing. RESERVOIR added ''virtualization-awareness'' to grid computing, by moving the focus from job scheduling, typical in grid computing environments, towards the creation and placement of virtual computing resources and generalized workloads. It primarily focuses its attention to the construction of an infrastructure in which virtual machines are dynamically instantiated at any node, regardless of location, network configurations and administrative domains. RESERVOIR developed the ability to place virtual machines in an effective and cost-efficient fashion, finding the best placement of virtual machines, or their mapping to physical machines, within a Federation.

OPTIMIS project [10] has been aimed at enabling organizations in outsourcing services and applications automatically to *trustworthy* Cloud providers, according to a hybrid-based model. It started in June 2010 and ran until May 2013. ***OPTIMIS*** [11] supported and facilitated an ecosystem of providers and consumers that benefit from the optimal operation of services and infrastructures, thanks to the optimization of the whole service lifecycle, including services for building, deployment, and operations.

StratusLab [12] was a project aimed to provide system administrators and resource providers with mechanisms to enable the efficient exploitation of computing resources. It was a 24-months project started in June 2010, co-funded by the European Community Seventh Framework Programme. StratusLab was also able to leverage external resources, following a hybrid architecture model, thus morphing the local Cloud into a hybrid infrastructure.

4CaaSt [13] was an European FP7 ICT project which started in June 2010 and ran until August 2013. It was aimed at creating advanced *PaaS* Cloud platforms, supporting the optimized and elastic hosting of Internet-scale and multi-tier applications. Building the PaaS Cloud of the future meant incorporating all the necessary features to ease development of rich applications and enable the creation of a real business ecosystem, where applications from different vendors can be customized and combined together.

BonFIRE: *Building service testbeds for Future Internet Research and Experimentation* [14] was an FP7 project aimed at designing, implementing and managing a multi-Cloud environment supporting applications, services and systems research in the field of the Future Internet (FI). It ran from June 2010 to December 2013. ***BonFIRE*** [15] targeted the Internet of Services community and provided a test infrastructure that is ideal to perform experiments related to distributed applications and services. ***BonFIRE*** adopted a federated multi-platform approach, providing interconnection and interoperation between novel service and networking testbeds. BonFIRE infrastructure spanned over 7 geographically distributed testbeds across Europe, providing heterogeneous Cloud resources, such as compute, storage and networking ones. The ***BonFIRE*** project envisioned a broker-based Cloud federation model where a broker component interacts between the user requests, the experimenters, and the different infrastructure instances, and provided a common OCCI-based interface [16] to expose all the Clouds' and networks' features as resources to the user. The ***BonFIRE*** and ***mOSAIC*** projects had some features in common: for example, the service discovery capabilities.

CONTRAIL [17] was an European FP7 ICT project focused on *Open Computing Infrastructures for Elastic Services*. It started in October 2010 and ran until January 2014. Specifically it introduced a new Platform-as-a-Service layer enabling an easy management and deployment of applications, also focusing on data storage [18]. The CONTRAIL project presents many relationships with RESERVOIR, as both dealt with Cloud Federation. In particular, while CONTRAIL mainly dealt with identity management, RESERVOIR focused on the technological issues related to migration among federated Clouds.

VISION Cloud: *Virtualized Storage Services Foundation for the Future Internet* [19] was an European Commission Integrated Project funded by the Seventh Framework Programme (FP7/2006-2013). It started in October 2010 and ran until December 2013. The goal of VISION Cloud was to design a novel, scalable and flexible storage Cloud architecture, able to serve a large number of concurrent users, thanks to a high level of scalability and flexibility *by design*, also enabling the delivery of different types of storage services. The idea of VISION Cloud was to build a Cloud storage environment on top of an infrastructure made up of multiple, potentially worldwide, distributed data center. Each datacenter has multiple storage clusters, hosting physical resources that have computational, storage and network capabilities. In literature several works can be found about the VISION Cloud project. For example, Bruneo *et al.*, in [20], provided an analytic model for the availability of a storage cluster, in the context of the storage Cloud environment proposed by the VISION Cloud project, whereas Kolodner *et al.*, in [21], presented two real application scenarios from the healthcare and media domains, to demonstrate the validity of their proposed architecture, which try to address the challenges of providing data-intensive storage Cloud services.

MOSAIC: *Multi-Modal Situation Assessment and Analytics Platform* [22] was a project devoted to the Cloud-based application developers, maintainers and users. It enabled them to specify service requirements in terms of Cloud ontologies, also useful to find Cloud services fitting the best to their actual needs, and efficiently outsource computations. The MOSAIC project began in April 2011 and was successfully completed in July 2014. ***MOSAIC*** presented a framework based on multi-agent brokering mechanisms, able to search for services matching the application requirements, and possibly set up a service *composition* if no direct hit is found.

The French ***CompatibleOne*** project [23] consists of an open source Cloud services broker, able to provision, deploy and manage any type of Cloud services provided by heterogeneous service providers. CompatibleOne can be considered as an element supporting the federation and interoperability of Cloud systems. It has got the capabilities for federating heterogeneous resources and integrating Cloud services by different Cloud providers. CompatibleOne provided an own object-based description model of Cloud resources called *CORDS (CompatibleOne Resource Description System)*. The CORDS is an OCCI-based model. This brokering platform exposes services to provide Cloud resources from different IaaS and PaaS providers, selecting them according to Service Level Agreement (*SLA*). The Broker then processes the provided *SLA* and any user requirements to create an instance of service. The Broker thus performs a selection process of the most appropriate providers for the provisioning of the described resources.

ECO2Clouds: *Experimental Awareness of CO2 in Federated Cloud Sourcing* [24] was a project supported by the FP7 program of the European Commission. The project started in October 2012 and completed in September 2014. The overall goal was the introduction of ecological concerns (e.g., energy efficiency and CO2 footprint) while developing Cloud infrastructure and Cloud-based applications. The project outcomes (i.e., models, architectures, software tools, design methodologies and guidelines) have been validated through challenging case studies, and by engaging the **BONFIRE** project partnership specifically.

CloudWave: Agile Service Engineering for the Future Internet [25] was an European Commission Seventh Framework Programme (FP7/2006-2013) funded Integrated Project started at the beginning of November 2013 and ran until October 2016. The overall aim of the *CloudWave* project was to provide a powerful foundation for the development, deployment, and management of a new generation of Cloud-aware services. Cloud services are hosted and deployed by Cloud providers in Cloud operation centers, and therefore can be controlled also at that point of delivery. In order to improve service quality and optimized resource utilisation, the *CloudWave* approach dynamically adapts Cloud services to their environment. Nowadays there are situations where it is not possible to improve application performance by simply factoring extra hardware in, so there is the need to change the program and its logic to adapt to new situations. *CloudWave* [26] is able to automatically select the best adaptation method such as asking the browser or mobile phone to take on additional work, providing additional resources to the Cloud application, migrating application components.

Cloud for Europe [27] is a project co-funded by the European Commission under the Framework Programme for Research and Innovation (FP7). It started in June 2013 and ran until November 2016. Cloud for Europe used *pre-commercial procurement (pcp)* as an instrument to address the objectives of the *European Cloud Partnership*. The project helped partners to adopt a well-defined *European Cloud Computing Strategy* for the public sector. The main objectives of Cloud for Europe were: (i) identifying obstacles for Cloud use in the public sector; (ii) defining services set to overcome these obstacles; (iii) procuring research from industry to find innovative solutions for Cloud services.

The EU-FP7-ICT-610531 ***SeaClouds*** project [28] aimed to provide an adaptive multi-Cloud management of complex service-based applications by developing Cloud Service Orchestrators and a set of tools to manage them. It started in October 2013 and ran until March 2016. *SeaClouds* [29] was set to provide and support skills such as service orchestration, adaptation and verification in a multi-Cloud context, providing a unified Cloud-independent procedure for managing services distributed across several Cloud Providers. To achieve these challenges *SeaClouds* is aligned with the most important interoperability standards such as OASIS CAMP and TOSCA.

BEACON [30], [31], was a project funded by the European Unions's Horizon 2020(H2020) Research and Innovation Programme. It started in February 2015 and ran until October 2017. This project was set to reach two main goals: (i) discover and build solution to federate Cloud network resources and (ii) to find out a way to integrate a management Cloud layer enabling an efficient and secure deployment of federated Cloud applications. Thus BEACON can be seen as a brokered architecture able to coordinate application deployment, with special emphasis on inter-Cloud networking, to support the automated deployment of applications and services across different Clouds and datacenters.

SUNFISH [32] was a project aiming to provide a specific and new solution to face the lack of infrastructure and technology to enable them to integrate their computing clouds. It started in January 2015 and ran until December 2017. SUNFISH enables the secure federation of private clouds based on the requirements of the Public Sector to federate private clouds belonging to different Entities, in order to share data and services transparently, while maintaining required security levels. The SUNFISH project developed and integrated software enabling secure cloud federation, as required by European Public Sector bodies. From another point of view, this project improved security in federated ''cross-border'' clouds, boosting the development of a cloud computing market in sectors where privacy and control of information propagation are essential (e.g., e-government, e-health, etc.) while encouraging a better resource utilization of cloud infrastructure for the Public Administration.

NEPHELE [33] is a project that was funded by the H2020 programme and it ran from February 2015 until January 2018. This project developed an end-to-end solution extending from the datacenter architecture and optical subsystem design to the overlaying control plane and application interfaces. Its purpose was to develop a solution able to create an hybrid system working on electronic-optical network architecture scaling linearly in relation to the number of datacenter hosts. This system is also able to consolidate compute and storage networks over a single Ethernet optical TDMA network. We have added this project to our selection here because it highlights the willingness of the European commission to create a cooperation among datacenters or, extending the scope, among clouds.

ENTICE [34], is the acronym of *dEcentralized repositories for traNsparent and efficienT vIrtual maChine opErations*. It was funded by the H2020 programme and it ran from February 2015 until January 2018. This project proposed an alternative approach for the management of VM images, to be stored in repositories designed to: i) simplify the creation of lightweight and optimized VM images; ii) decompose and distribute automatically VM images based on multi-objective optimization (performance, economic costs, storage size, and QoS requirements) and iii) Enable auto-scaling of Cloud resources, that supports interoperability of VMs across Cloud infrastructures without provider lock-in.

SUPERCLOUD [35], is a project that was funded by the H2020 programme and it ran from February 2015 until January 2018. The acronym means *User-Centric Management of Security and Dependability in Clouds of Clouds*. The target of this project was the proposal of a security approach, for infrastructure management paradigms, that is both user-centric or self-managed, for a Multi-Cloud system (authors refer to it as ‘‘clouds of clouds’’), managing security among the CSPs involved, according to the policy defined by the SUPERCLOUD service that ran upon the SuperCloud Distribution Layer coordinating the appliances provided by CPSs.

FIESTA [36], is a project called *Federated Interoperable Semantic IoT/cloud Testbeds and Applications*, which was funded by the H2020 programme and it ran from February 2015 until June 2018. This project created a system able to interconnect and enable the interoperability of different IoT testbeds, through a federated architecture to collect data from several CSPs, and analyze semantically, before making them available through the FIESTA-IoT System.

III. FEDERATION VERSUS MULTI-CLOUD

No universally accepted terminology has been defined to identify a Cloud computing scenario where each Cloud service provider collaborates ‘‘horizontally’’ or ‘‘vertically’’ with other Cloud service providers.

However, in our opinion, there are two terms that primarily identify scenarios where multiple Cloud providers interact each other with the aim of improving the service levels provided to users: Cloud Federation and Multi-Cloud. These terms refer to two such scenarios differ both in terms of the interaction between existing Cloud providers and in terms of operating modes.

In a Cloud Federation context, basically a Cloud service provider shares its (currently unused) own resources with other Cloud service providers participating in the same Federation. In this way a Cloud service provider is able to transparently and dynamically enlarge and optimize its own resource capabilities by instantiation of new virtual environments to keep up with incoming user requests. Thus, such a Cloud service provider does not plan to ever deny service or reject requests from their clients, thus keeping a high level of QoS. This interaction is completely transparent to the end-user, completely unaware that her Cloud provider is requesting additional resources from other Cloud providers. Moreover the user is not aware whether her service is hosted by her reference Cloud provider or across multiple federated Cloud providers.

For this reason, we can state that it is reasonable to affirm that the concept of Cloud Federation is *Cloud-oriented* and not *enduser-oriented*. In other words, from a Federation perspective, the users of the system are the federated Clouds operating within the Federation and not the end user that asked for service (IaaS, Paas or SaaS).

Cloud Federation is meant to give additional benefits and new business opportunities to Cloud Service Providers. Conversely, the end-user is actually the consumer of any service in a Multi-Cloud scenario.

Generally, there is a Service Provider (a Broker) which is responsible for the provisioning of services for its users. The Service Provider picks out the services from different Cloud Providers taking into consideration the users’ requests. In this scenario there is no collaboration or interaction among the Cloud providers engaged by the user. The Broker performs management, negotiation, deployment, monitoring and migration operations only, in order to fulfill the users’ requirements.

IV. BROKERED VERSUS DECENTRALIZED APPROACH

When we take into consideration environments where multiple Cloud domains are involved, solutions to orchestrate all communications and actions are necessary in order to setup and maintain such an environment. The main approches are **brokered** and **decentralized**, respectively. These two schemes are applicable both to a Federation context and a Multi-Cloud one.

In a centralized solution there is a single intermediate entity, the Broker, that is in charge of managing the environment where multiple Clouds are participating. The broker has several tasks to perform: (i) acts on behalf of users identifying suitable Cloud services; (ii) aggregates resources from multiple Clouds; (iii) negotiates with the Clouds for allocation of resources that meet the QoS requested by users; (iv) schedules and controls resources; (v) performs matching algorithm to sort the Service Providers offers that best fit users’ requests, etc. In other words, it acts as an interface towards several heterogeneous Clouds.

The Broker entity can use two approaches in order to interact with the participating Clouds, either via administrative’s API or via user’s API. The differences between the two interaction paths are based on the depth of the agreements over interaction, signed by the centralized intermediary and the involved Clouds. In fact in the first case a strict agreement needs to be signed, because the target Cloud has to provide administrative rights to the Broker. The second case requires only a normal agreement between Clouds and the Broker, that will be able to use a restricted sets of API, corresponding to the normal user API’s provided by the Cloud to its end-users. A common point for both is the communication interface used by the Broker, based on the availability of specific drivers,

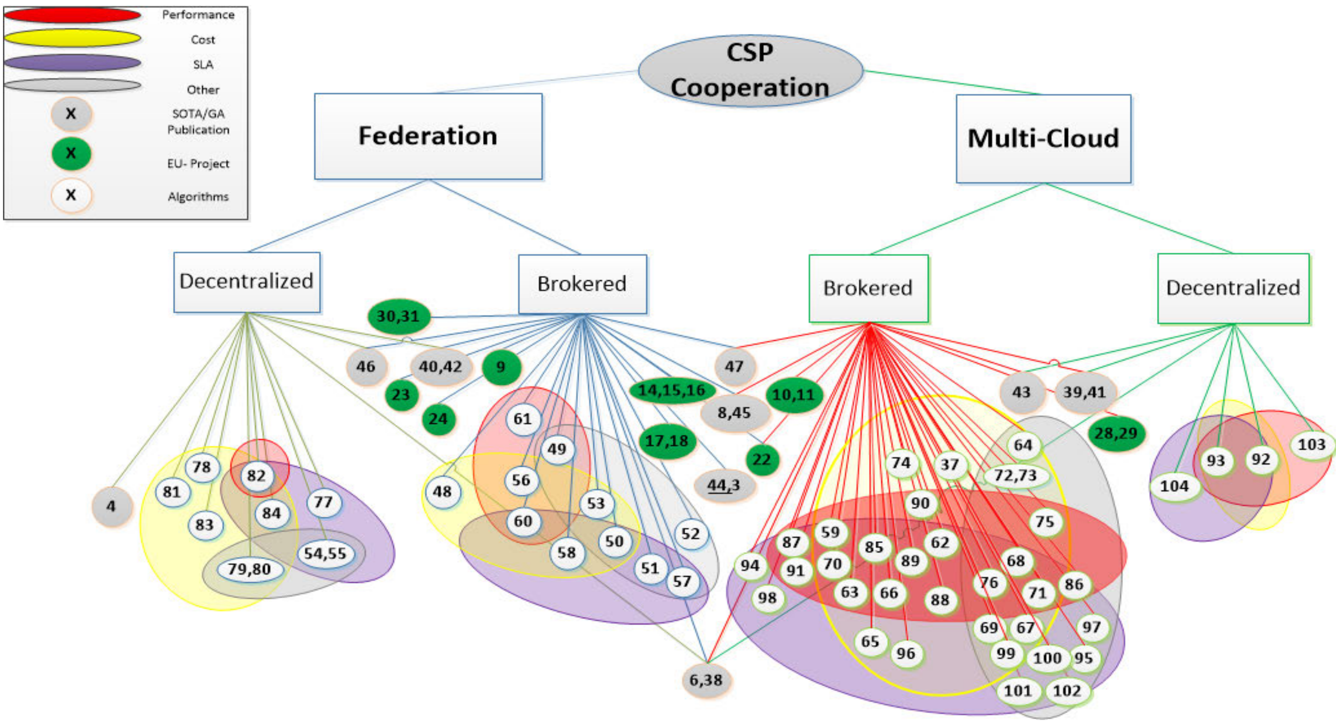

FIGURE 1. Taxonomy Tree.

adopted as interfaces for the communication with a specific set of Clouds, (Openstacks, Amazon, Azure, etc.).

In a decentralized environment, instead, this coordination feature needs to be present within each involved Cloud, thus this configuration does not need the presence of a third party entity. Each involved entity performs all the tasks listed above, by itself. Thus the decentralized scheme presents itself as more flexible if compared to the centralized one, but at the same time it is harder to be implemented due to the heterogeneity of the technologies adopted by each single Cloud. The approaches to interaction available for a decentralized environment are mainly two: **Peer Cloud** and **Hybrid Cloud**. The Peer Cloud approach to interaction enables the Clouds involved to interact each other directly with horizontal (logical) communication between similar entities (e.g., local Network Manager with remote Network Manager, and so on).

The Hybrid Cloud approach, by comparison, has its roots in the private cloud space, but leveraging public clouds when needed for whatever reason, e.g., to cloudburst, or just to mitigate any private cloud infrastructure failure or downtime. This way, by playing to the strengths of both cloud deployments (and business models), and trying to possibly meet providers', owners' and users' requirements, limits in using services from a single cloud may be easily (and granularly) overcome, trying to possibly meet providers', owners' and users' requirements at the same time. Summarizing, it is possible to distinguish between brokered and decentralized approaches analyzing how the cooperation happens. First of all, as described above, the management and interactions for the allocation of resources are drastically different. In one case, the end-user request received by the Cloud will be forwarded, when the Cloud itself needs it, to a central entity that will take care of everything (e.g., CSP selection, Network management, forward the access credential to the requesting Cloud, and so on). In the other case, there will be one or more mechanisms for dissemination of the request (e.g., it could be a framework, a peer-to-peer network protocol, and so on) to allow the requesting and cooperating Clouds to interact autonomously to fulfill the end-user requests.

V. LITERATURE ANALYSIS

Figure 1 shows an overview of the analyzed works. They are organized in a taxonomy tree structure. The nodes of the taxonomy tree are the key words that we selected and that characterize the selected papers.

The tree's root represents, as expected, the research topic "*Cooperation strategy among Cloud Service Providers*" having special care on "*optimal selection techniques of Cloud service providers*". The considered tree has four levels. The leaves of the tree are the reviewed papers: the white leaves are publications referring to "**Algorithms**", instead, the grey leaves are publication referring to "**SOTA/GA** (State Of The Art / Generic Analysis)"; in green, are highlighted the publication related to the EU-Project.

At the level two of the main tree we have two nodes namely "*Federation*" and "*Multi-Cloud*". This is an important distinction because even if Federation and the Multi-Cloud topics may appear to be very close, they are driven by different needs [6]. These two nodes, in turn, can be seen as the roots of two specular sub-trees.

As is possible to see from Figure 1, our attention falls in particular on publications related to "Algorithms" (white leaves of the tree). At the level three there are the nodes "Brokered" and "Decentralized". They grouped the works proposing "Brokered" or "Decentralized" solutions in best provider selection. Finally we grouped the algorithms in 4 sets (Cost, Performance, SLA, Other) depending on the types of parameters that the algorithms take into account during their execution, these last groups are shown in the tree with elliptic shapes that identify sets based on color. These parameters are listed in Table 1. Table 3 and Table 4 show exactly what are the parameters, listed in Table 1, that each paper considers during the selection process.

TABLE 1. Key-Parameters of the considered grouping sets.

SLA	COST	PERFORMANCES	OTHER
QoS, Availability, Reliability, Usability, Resiliency/fault tolerance, Interoperability, Recoverability, Security/Confidentiality, Agility, Durability, SLO (service level objective)	Financial charges, Communication, VM & Service, Storage Cost, Down/Upload, Electric Power, Computing Power	Execution Time, Response Time, Throughput, Network, Service Response Time	Functional (CPUs, Memory), Number Of Available Service, Service Type, Collision/Collusion, Number of Available VMs, Computational Capacity, Trust/Reputation, Lock-in, Co2 Reduction

Specifically, the leaves are grouped in 4 set that are: "**Cost**" containing the algorithms considering non-functional parameters (i.e., cost or price); "**Performance**" algorithm aiming to a performance optimization as execution time or communication time; "**SLA**" such as Availability, Reliability, Usability, etc.; and "**Other**" parameters not included in the previous non-functional parameters (e.g., placement constraints or reallocation constraints or number of available services, Collision or Collusion [37], or functional parameters like virtual CPU number, amount of Memory and so on. See Table 1 for the details). The last level is composed by the papers included in the survey.

In o ur analysis we are focusing mainly on "optimal selection" topic, collecting in this way 87 reviewed papers organized following the structure presented above. The discovered papers, presenting various kinds of analysis (State of the Art-SOTA or Requirements) of such a topic, represent approximately 16, 1% (the grey circles) of all the papers; the other works analyzed are related to European Projects and represent the 17, 2% (the green circles). The remaining 66, 7% (the white circles) of the papers provides instead different algorithmic solutions for the problem at hand. We put greater attention on this algorithm papers. An overall vision of the papers' distribution is given in Figure 2.

Specifically in order to better understand the percentage results of our research, we extracted from the "Taxonomy Tree" four main sets and we placed the papers inside them according to Figure 1, thus showing the percentage value of papers belonging to a certain set.

However, these results are expected in a *Multi-Cloud* environment, in which a certain "Cloud Provider" may not be aware of other "Cloud Providers". As a consequence, there is the need of a central aware entity able to work with different administrative domains. The analysis of publications is organized in four sections, discussed after the following preliminary analysis in section V-A

A. SoTA AND ANALYTIC WORKS IN GENERAL

Even if the publications belonging to this set are not a category in itself, we chose to describe it separately because these papers discuss or analyze multiple categories as in the case of [38], where a preliminary categorization for clouds federation or cooperation according to a Multi-Cloud approach (authors call it InterCloud), or in the case of [6] in which are analyzed all the scenario related to Cloud cooperation. As the reader can appreciate from the taxonomy tree and the detailed description provided below the **Brokered** cases (both for Federation and Multi-Cloud) are investigate more thoroughly.

Our analysis of the State of the art begun from following review papers; these are focused on several aspects that involve Cloud cooperation. Some of the most recent surveys that we found are [39], [40] and [41], talking about services composition on clouds, respectively the first referring to approaches and analysing their issues, the second reviewing the mechanisms used for service composition on Cloud environments and the third reviews the mechanisms for service composition specifically on Multi-Cloud environments.

Another survey analyzing functional and not-functional requirements of cloud federation, is [42]. The work of Sun *et al.* [43] did a deep state of the art analysis regarding the Cloud service selection topic. The authors considered seven points of view in building their survey, namely: context, goal, data representation models, selection techniques, selection parameters, methods for quantifying qualitative parameters, and criteria weighting methods. Each of the examined selection methods have been put in a specific group according to the approach (decision-making-based, optimization-based, and logic-based approaches) followed to accomplish that goal. In contrast to the surveys already available in literature, this work aims to provide a fully comprehensive analysis of Cloud Service provider-level scenarios, that will help the reader identify the most pressing concerns under investigation by the community, and the related challenges still to be addressed. In this survey, the authors are taking into account:

1) Cooperation paradigms (Multi-Cloud and Federation),
2) Management approaches (Brokered or Decentralized),
3) Key techniques (algorithms, mathematical model, and so on),
4) Key metrics (SLA, Cost, Performance, Other).

In [3] the authors presented a way to create a federation among several Cloud providers by means of a Broker.

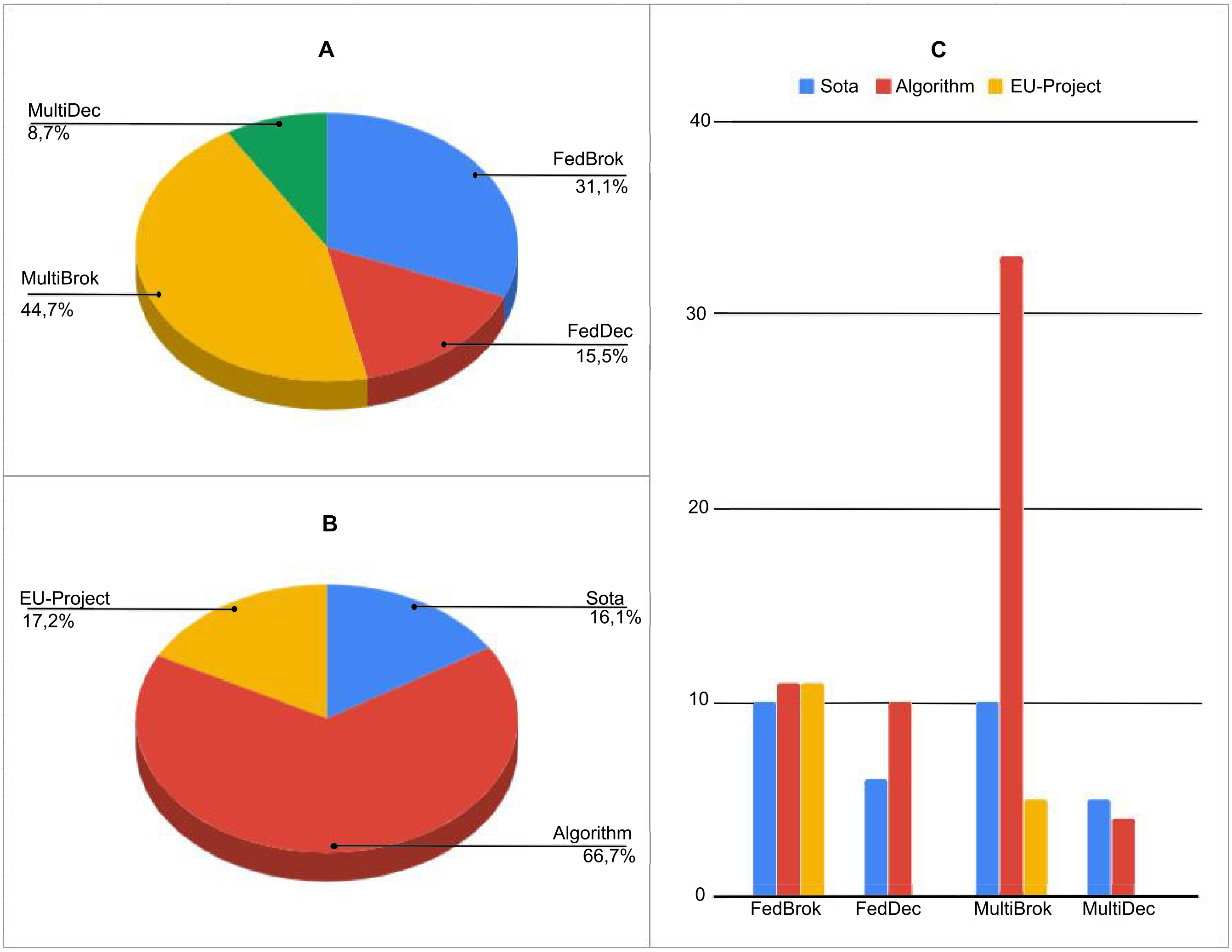

FIGURE 2. Pie Chart Concerning the Paper Organization: a)Distribution of all works analyzed in the four branches. b)Distribution of all works respect the publication types. c) Numbers of works grouped by type and approach.

In order to explain their federation concept, they used a **WRF** *(Weather Research and Forecasting service*) use case application. They illustrated a layered Cloud service model (SaaS, PaaS, IaaS) making a comparison between the delegation and federation concepts. In the first case, when a layer (SaaS) cannot fulfil the user's request it asks for additional resources to the underlying layer (PaaS) that in turn may ask the IaaS layer for additional virtual resources *(Vertical elasticity*). In the second case, when a layer (SaaS, PaaS or IaaS) cannot fulfil the user's request, the broker asks to its foreign peer layer for external resources (*Horizontal elasticity*). Moreover the authors argue the main goals of the Broker for each layer (SaaS, PaaS or IaaS). Anyway the considered work does not focus on the provider selection topic but on the importance in the decoupling of the Cloud layers (*aaS) in order to distribute the application execution over different providers.

Buyya *et al.* in [44] present "*InterCloud*" which is a federated environment to support the scaling of applications across multiple Cloud providers. They identified three main kinds of actors for the presented *InterCloud*: *Cloud Coordinators*, *Brokers*, and an *Exchange*. The first two manage a specific Cloud enterprise and its membership to the federation. The *Broker* acts on behalf of users and it looks up available Cloud services by means of the "*Exchange*" and negotiates with the Coordinators for the resources allocation. The "*Exchange*" plays the role of information registry storing all the required federated Clouds' information, and providing a match-making service to map the users' requirements to suitable service providers.

In the [8] and [45] the authors made a deep analysis of the current state of the art regarding the use cases for Multi-Clouds environments. In [45] they focused on the Interoperability and Portability challenges and ensuing requirements, classifying them in several solution groups. In [8] the author divided the Multi-Cloud requirements in three different groups, namely: Development, Deployment and Execution. One of these requirements is the ability in the selection of Clouds' services and resources. So methodologies to compare Cloud services in a Multi-Cloud scenario are needed. But these works are only surveys and they do not provide actual ways to accomplish goals such as an optimal selection of Cloud service providers in a federated or Multi-Clouds scenario.

In [4], an analysis of the requirements for the instantiation of an IaaS Cloud Federation is proposed. The outlined process is based on different requirements, covering all the challenging fields of a federated environment. In particular such an environment should feature automation, scalability, versatility, and security capabilities. More specifically the paper emphasizes that when a demand of additional external resources is received, a cloud provider needs to be able to discover and then to select one or more of the available federated cloud providers to be asked for the unused resources.

Subramanian *et al.* in [46] proposed a thorough study concerning the resource provisioning in a federated context by analysing several deployment architectures from literature. The authors proposed their own perspective on different "Federated architectures", that have been categorized in "*Cloud Bursting Architecture*", "*Cloud Broker Architecture*", "*Aggregated Cloud Architecture*" and "*Multi-site Cloud Architecture*" analysing their specific features. *Cloud Bursting* is a vertical exploitation of Public Cloud resources when a Cloud runs out of its own internal resources. This type of architecture is loosely coupled. It means that there are basic requirements in terms of control, monitoring and security. As for a Cloud Provider to manage integration and selection of external services from several Cloud providers is a very hard task, it delegates such a task to an entity called Broker. This is the case of *Cloud Broker*. Here the aim is different from the previously analyzed architecture. In the *Cloud Bursting*, in fact the goal is to be able to manage the peak demands. In *Cloud Broker architecture* the target is to optimize parameters as cost, performance and reliability. However also this scenario is loosely coupled as the first. In both the architecture there is no cross-site management. In the *Aggregated Cloud Architecture*, providers inter-operate and aggregate their resources based on contracts providing to the users a virtual and infinite set of resources. The main goal is to meet peak demands. This scenario adds complexity from the management point of view. It is a scenario partial coupled. In fact, there is the need to properly control the virtual environment, to monitor the virtual resources and the agreements among the involved parts. Moreover, the Cloud providers has to be able to manage the cross-site virtual-networks and storage. *Multi-site Cloud Architecture* is a Cloud provider having a geographically distributed infrastructure. The goal of this scenario are: scalability, isolation or multiple-site support. It is tightly coupled in terms of control, monitoring and security. This last scenario has to meet the cross-site live migration and high availability.

An interesting approach based on Request Description Language is presented in [47]; there the authors described a framework called "A2SC", a system able analyze and assign user requests through analysis made by two layers: Automated Request Processing Layer (**ARPL**) and Automated Service Composition Layer (**ASCL**). The system presented may be considered able to work both in Federation or in Multi-Cloud environment and it is broker-based system that is able to compose a series of services in order to answer to a user request.

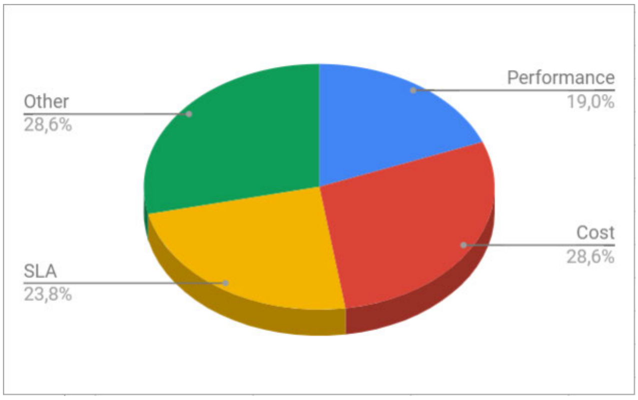

FIGURE 3. Pie Chart Concerning the distribution of the paper in each sub-category.

Insights: Architectural approaches have been proposed to manage the cooperation between Clouds (and from a general point of view CSPs). However, most of the works are focused on management driven by a third party. Furthermore, the interaction based on decentralized management and cooperation seems not to be able to attract the interest of researchers.

VI. BROKERED SOLUTIONS

In this section, we analyze the works based on a broker component. The broker component, both for Federation and Multi-Cloud scenarios, is a central component that is able to manage the entire system of systems according to the evaluations made on parameters, collected by the broker itself, related to the clouds involved in the environment. Collection of data may be done periodically or starting from event-based broker's requests made to CSPs, or may be driven by the CSPs themselves. The evaluation process is based on policy decisions driven by the analysis made on the parameters previously collected. In our analysis, we grouped the publications with respect to the topics where the policies fall.

A. BROKERED SOLUTIONS IN A FEDERATION

The papers grouped under this section are those providing algorithmic optimization solutions to the Cloud selection task in a Federated environment. This means that the broker knows exactly which CSPs are involved in the federation at the time of selection.

1) COST

The first three papers in the following all share the same perspective to provide solutions aiming to optimize the service provider selection from the *profit/cost* point of view.

The first work focuses on the selection in the instantiation of the federation. Whereas the other two papers in the same subset propose solutions for the selection in the context of a Federation being already setup. Specifically Mashayekhy *et al.* [48] proposed an algorithm to improve the automatic scaling capabilities of a Cloud provider in fulfilling the users' demands. They proposed a "*Cloud Federation Formation Game*" that considers the profit achieved from a Cloud taking part to the federation. The algorithm is based on the idea that a Cloud will want to participate to the federation, if the profit that it will obtain is higher than that it would have staying alone. The process is run by the *Broker*, having all the information about the Cloud Providers, such as their available resource, costs, etc. The proposed algorithm focuses on the selection of Cloud Providers to create a Federation able to meet the users' requests whilst also maximizing the federation profit. Thus this work, together with Gahlawat *et al.* [49] and Abdo *et al.* [50], is grouped in a subset of works that aim to optimize costs and profits.

In fact Gahlawatet al. [49] proposed a framework able to improve the federated VMs selection process by means of a Divided KD tree data structure in a market-oriented Cloud computing. The selection process is based on a matching mechanism between the VMs under offer and the requests for VMs. The search looks like that for a binary search tree. When an offer arrives the VM is added to the tree. When a request arrives, a matching mechanism starts to look for a similar VM in the tree. If any of the virtual machine matches the criteria, that VM is removed from the Tree.

The authors do not provide any implementation details about the selection algorithm. The proposed approach can be considered a selection at the IaaS layer. Besides, this work takes into consideration functional constrains like number of CPUs, amount of memory and so on.

For this reason it is counted also in another set (**Other**) together with Badidi *et al.* [51], Abdo *et al.* [50], and Caballer *et al.* [52].

In this paper Aryal *et al.* [53] follow a similar idea of the previous one. They focused on a very important aspect of a federated environment: a fair shared of the revenue. The considered federation is a brokered model where a Cloud broker collects and publishes the price and the number of the available resources provided by the CSPs belonging to the federation. Each CSP provides a specific amount of resources in terms of VMs. The VMs have a fixed price depending on the processing power, memory, and storage. A CSP communicates with the Broker to maintain the list of VMs updated. The main goal of the paper is to present a mathematical model by means of it is possible to chose the best combination of VMs, to be provided to the end user, in order to maximize the revenue and the profit for each CSP in the Federation. The authors define Revenue as $\sum_{i=1}^{n}\sum_{j=1}^{m} R_j * S_j$ where R is the set of requested VMs and S is the price of each VM. The Profit is defined as $\sum_{i=1}^{n}\sum_{j=1}^{m} R_j * (S_j - C_{ij})$ where in this case C is the cost charged to the CSP. The interesting aspect of the paper is that the Federation is the means to increase the profit. In fact applying the Shapley fair model to a federated Cloud environment it is possible to fairly spread the revenue between the CSPs. The carreid out simulations proves the validity of the model in fact CSPs can obtain big profits by participating to the federation than acting as stand-alone Cloud.

Abdo *et al.* [50] proposed a *rearrangement* of the ***Cross Cloud Federation Manager (CCFM)*** process presented by Celesti *et al.*

In the authors' opinion the fully distributed approach followed by Celesti *et al.* ([54], [55]) in performing the discovery and matchmaking processes is valid but it presents several shortcomings in terms of communication delay, overhead and reputation mechanisms. The authors' solution rearranges the ***CCFM*** in a *Broker-based CCFM*. They argue that *Brokering* in ***CCFM*** can improve a given *Cross Cloud Federation Manager*. In this novel scenario, as the *Broker* periodically updates a centralized file containing the Clouds' status, a home Cloud, only with a message pair, can obtain all the information about Clouds' offers. Moreover the algorithms for the matchmaking (selection of best Cloud Providers) should be run by the *Broker*. In other words the authors move the policy from the "Cloud" to the "*Broker*", thus reducing messages traffic between the "Home Cloud" and the "Foreign Cloud". The authors present the high level algorithms concerning the discovery and matchmaking processes. The algorithm takes into consideration Functional (amount of resources) and Non-Functional parameters (Cost, SLA and QoS, Trust Relationship and etc.) so in Figure 1 it is represented under three nodes: **Cost**, **SLA** and **Other**.

The last paper of this section, Aryal *et al.* [56], presents a decision-making algorithm for VM placement, which a Cloud federation broker needs to meet multiple optimization objectives, derived from application requirements. The proposal lies in the context of the BASMATI project, supporting its Cloud federation architecture. In particular, a genetic algorithm (GA) approach has been taken for the design of the resource allocation algorithm, and generated solutions have been proven Pareto optimal.

Insights: The analyzed works aim to optimize the *profit/cost* deriving from Cloud selection made by a federated broker, where the cost of the resources are dependant on the physical resources used by the federated Cloud environment to host the instances due to requests coming from the Federation. The analyzed publications present algorithms for placement optimization and CSP selection, constrained by cost parameters but even starting from these works it is possible to appreciate the advantages coming from the presence of a broker inside a federated environment. The analyzed solutions go from a situation in which the Federation is not set up from the beginning to the case in which the federation is already created.

2) SLA

In a federated environment, Cloud providers need to establish guarantees about the sharing of their own resources by the stipulation of a priori agreements between interested parties. The papers of this set aim to perform a Cloud provider selection considering the SLA parameters throughout the execution of the algorithm. Nine *et al.* [57] share the "**SLA**" subset with the previous cited work Abdo *et al.* [50]. It proposes a *Broker*-based approach to optimize the outsourcing of user's requests.

In particular, the *Broker*, taking into consideration the available resources the Cloud Providers, has to decide whether to forward the request to another Cloud provider of the federation or assign that request to the provider that first received that service request. The proposed approach considers both functional and non-functional requirements and tries to find the best way to fulfill users' requests without violating the *SLA*s constraints.

The approach proposed by Barreto *et al.* [58] treats the problem starting from the user's perspective. The user formalizes its requests in form of a contract that is proposed to an entity named Federation Support (FS). This entity extracts the SLA from the received contract and rigs up a broker entity able to interact with the Cloud provider to negotiate and to control the SLA agreed by FS. The broker, publishes the users' request on the "Resource Panel", waiting for a Cloud Provider (CP) response message; the message will describe the CP that will provide totally or partially the required resources. Whenever an offer by a CP is received then the broker fulfills the request, also, as a composition of several CPs' offers. The final broker decision is taken using optimizing criteria (e.g. costs) that could generate better benefit for the Cloud users.

In Villari *et al.* [59] The authors present a solution for an OpenStack Cloud federation scenario managed via central broker. This solution, developed inside the BEACON project, aims to easily deployment of virtual resources. The broker analyses a manifest provided by the "Borrower" (this is Tenant[1] of the Cloud federation), splits it in sub-manifests and instantiate each one on a set of Clouds selected by all the Clouds available in a particular geographical area according to manifest description. The broker is also able to create virtual networks between Clouds involved, avoiding the usage of public IP if it is not required inside the manifest. The authors provide a description about architectural design and on enhancement done at the standard used to write manifest for OpenStack Orchestrator (HEAT). The broker mainly aims to make a placement policy based on a strategy able to maximize federation user's profit and to provide a Location aware elasticity (Scalability). The selection algorithm is used to identify the Cloud where atomic elements have to be instantiated.

Tricomi *et al.* [60] follows the work presented in "Orchestration for the Deployment of Distributed Applications with Geographical Constraints in Cloud Federation"; the authors present a brokered solution useful both for federated and Multi-Cloud environments. It consists in a broker able to manage Multi-Cloud application deployment according to application descriptor provided. The application descriptor (called BEACON Service Manifest) contains each Cloud Service ARchive (CSAR, defined in TOSCA standard) and for everyone contains also directive about the geographical area in which deploy it. The presented Broker aims to improve scalability, minimize cost, maximize performance. The solution is presented with an example of application deployment using TOSCA on a generic scenario composed by a group of federated Clouds, respect the starting paper, the broker is enhanced with a module able to translate TOSCA standard in Heat Orchestrator Template (HOT) standard. **Insights:** The set of publications above described present algorithms that aim to uphold the SLA defined between users and federation. In this scenario, the broker evaluates a set of parameters according to user's requests and CP constraints and requests assigning resources and jobs on the federated CSPs. Some of the work analyzed start the evaluation on the user's request, other instead taking in account the CSPs constraints and status.

[1]The tenant is a person or a society, that uses resources and services provided by a Cloud in order to provide to its clients.

3) PERFORMANCE

Liu *et al.* [61] is the only publication that takes into consideration exclusively performance requirements. It illustrates a mathematical model to allocate virtual resources in a federated Cloud environment in an efficient way. The *Federation Portal* through a monitoring and profiling system collects information about both the status of the federated resources and the application behaviour. The algorithm aims to allocate the resources in the federation reducing the *communication time* and the system *throughput* due the VMs repacking to obtain better performances. The authors' approach is based on: *(i)* the estimated communication time ***ECommT***, *(ii)* estimated computation time ***EcompT***, *(iii)* the $CCR = \frac{ECommT}{ECompT}$, *(iv)* the Estimated execution time $EET = ECommT + ECompT$ of each Cloud. When a job arrives the *Federation Portal* calculates the average value of CCR of the first J jobs in waiting queue of each Cloud and considers this value as $CCRthreshold$. After, for each job the system checks its CCR value and depending on whether it exceeds or not the threshold, decides if to allocate resources in one of the federated Cloud minimizing the EET. The algorithm can aggregate resources from some participated Clouds by selecting a sub-set of Clouds whose resources sum is equal to the job requested ones. **Insights:** In this subsection the discussion is focused on one work only that is not already included in the two previous subsections. This work describes an algorithm that reduces communication times, and the system overhead due to VM repacking actions when a VM has to be moved among CSPs.

4) OTHER

Badidi [51] and Caballer *et al.* [52] presented two different brokered frameworks. The first is a brokered solution driven by a mathematical optimization function. The second is a resource description language based on matching solution between the users' requirements and the resource constraints.

In detail [51] describes a framework for "Quality of Context (QoC)"-driven selection of context services, specifically proposing a "Federation" of Cloud-based *context Brokers*. In this proposal *Context Consumers* (CCs) consume the context information, *Context Brokers* (CBs) allow Clouds to cooperate each other and *Context Providers* (CPs) deploy their context services offering several types of context information. Each *Context Provider* monitors, collects and elaborates sensors' data to determine the **QoC** value. The authors describe the mathematical selection algorithm based on a probability of correctness function $U_P(p)$ and the freshness function $U_F(f)$ which are the only two considered quality attributes for the tested algorithm. These two functions put in relation the offer of the *Context Service* (*p* and *f*) with the required value of the *Context Consumer* ($\alpha_P and \beta_f$). Each of the *Broker*s execute the selection algorithm to find out the best *Context Service* which maximizes the global utility function $U = w_1 U_1 + w_2 U_2 + \ldots + w_n U_n$.

In Caballer *et al.* [52], a framework for the easy and automatic deployment, selection, configuration and monitoring of Virtual Machine Images (VMIs) is presented. The architecture of the framework counts different components: (i) "Resource and Application Description Language" (RADL) which is a language describing the requirements of the virtual infrastructure; (ii) a "**VMRC** - Virtual Machine image Repository and Catalog" used to find the available VMIs taking in consideration the users' requirements; (iii) the IM - "Infrastructure Manager" that is the central part of the system with in turn three main components the "*Cloud Selector*", the "*Cloud Connector*" and the "*Configuration Manager*". "*Cloud Selector*", in particular selects the best combination of VMIs and Cloud Providers by querying the **VMRC** to pick out the VMIs that better meet the users' requirements. After that it retrieves the list of the available Cloud Providers and selects those that are compatible with the selected VMIs. The two last previous papers talk about selection approaches that consider "functional and non-functional parameters. They are placed under the "Other" node. But while Badidi *et al.* [51] present a mathematical algorithm Caballer *et al.* [52] focus on the *RADL* language for the requirement descriptions without providing details of the selection process.

Insights: In this sub-section are described two of the six publications contained in the *Other* set. The two approaches shown place the resources according to the choices of the broker that analyzes the federated CSPs, in one case the algorithm tries to evaluate both requests and resource available, in the second case the broker bases its choices on the matching done through the description language here defined.

B. BROKERED SOLUTIONS IN A MULTI-CLOUD

By analyzing Figure 1 we can see that the totality of the reviewed papers take into consideration "Cost Optimization" or "**Other**" various optimization goals.

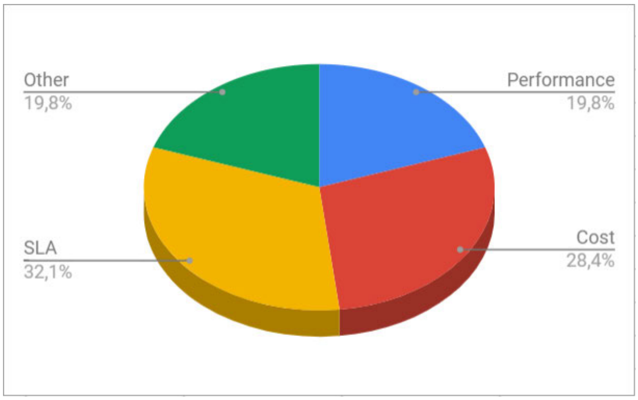

FIGURE 4. Pie Chart Concerning the distribution of the paper in each sub-category.

These two big sub-sets include another smaller one namely the "**Performance**" one.

Only the 60% of the papers have as goal a performance optimization. Finally there is the 80% of the solutions that are "SLA-based". Some papers presented below, besides the "Cost-optimization", have other similar characteristics so that these papers will be present in other sub-sets.Table 2 gives a summarizing vision of as just said.

TABLE 2. Percentage paper population values of each considered set of parameters.

	Federation		***Multi-Cloud***	
	brokered	**Decentralized**	**brokered**	**Decentralized**
Cost	28,6%	23,1%	28,4%	25%
Performance	19%	7,7%	19,8%	37,5%
SLA	23,8%	38,5%	32,1%	37,5%
Other	28,6%	30,8%	19,8%	0%

1) COST

This sub-set contains the 28, 4% of the papers under the node "**Multi-Cloud**". All of them proposed a Broker-based framework adopting different ways and technologies to perform the selection process but in order to make easier for the reader the analysis of the approaches we have grouped in subset.

Selection Strategies

This first subset contains papers that will reduce costs through application of CPS selection strategies. The first three works present matching-based selection strategies.

In particular in Sundareswaran *et al.* [37] is presented an interesting architecture where a *Cloud Broker* manages an automatic service selection in a Multi-Cloud environment. Their approach is based on "*Unique Indexing Technique*" (CSP-index). The *Cloud Broker* stores the Cloud Providers information in a *B-tree* data structure. It is a structure similar to the binary tree but in the *B-tree* only the leaf nodes contain information and other nodes work like a directories, having more then two children. Then the Broker constructs his B-tree encoding the *CP*s information ($Key_s p$, SID, $p_1, p_2, \ldots, p_n$). When a user sends his service request in form of $Q = (D_1, D_2 \ldots D_n)$, where D_i are the expected properties of the Cloud Providers, the *Broker* encodes that request in the same way of the *CP*s' properties information obtaining a Key_q that is the index key of the request. After, the *Broker* performs a searching process browsing the *CSP*-index to find the k candidate service providers whose encoded properties are the k nearest neighbours for that request. The search starts from the root and follows the path with the smallest **hamming distance** to Key_q.

In Jrad *et al.* [62] the presented *Broker*-based framework aims to deploy application workflows in a Multi-Cloud environment. The framework is able to perform two main tasks: an automatic selection of the *Cloud Service Providers* and a workflow data management respecting the user's **SLA**s requirements. The brain of the system is a *Cloud Service Broker* that performs several tasks like: Identity management, *Match Making*, SLA, Discovery management and Scheduling. It's the mediator between the client and the *Cloud Service Providers*. The authors show the deployment flow of the application workflow. Regarding the *Match Making*, the authors used a simple matching policy called "*Sieving*". It selects only the Cloud that meets all the user's requirement and the algorithm will select randomly only one if more Clouds match the users' requirements.

Jrad *et al.* [63] gave the details of (i) simple static matching scheme, called the "*Sieving*" algorithm and (ii) the utility-based matching algorithm which takes in consideration both the functional and non-functional parameters (response time, throughput, price etc.). The *Sieving* algorithm performs a one by one comparison between the user's requirement parameters and the "*SLA*" metrics ($availability = 100\%$) of the *Cloud Provides* which are the two inputs of the algorithm. The output is a set of selected Clouds matching all of the requirement attributes ($price < 0.02.\$/h$).

Ngan *et al.* [64] instead proposed a selection strategy based on a Semantic approach and Web Ontology Language (**OWL-S**), through the proposal of an OWL-S Based Semantic Broker system which provides service discovery and selection capabilities. The system allows to semantically discover and pick out those Cloud provider services that meet the Cloud consumers' requirements. The *Cloud Providers* offer their services to the brokerage system. All of these services are stored in a Semantic Service Repository (SSR). The user contacts the Broker asking for a service and the *Broker* executes a matching algorithm to pick out the best service composition. The authors provided a wide and deep analysis of the problem and also a verbal explanation of the algorithm.

The following four works focus on the Storage Service Provider Selection and aim to find a trade-off between the storage service cost and the QoS. The first and the fourth ones are strongly mathematical-based solutions. The second is middle way between a mathematical and match-making approach and finally the third paper instead is closer to a matching solution. In particular, from an overall point of view, the ideas presented in the first, third and fourth papers are similar each other.

Entering into the detail, the first Mansouri *et al.* [65] developed three different complex mathematical algorithms for the best placement of object's replica chunks among different independent data center(DC). Each object is split in more chunks and they have a number of replicas r usually $r = 2$ or $r = 3$. These algorithms are run by the *Broker* to help users in their research for a suitable placement of objects as long as the required *QoS* is fulfilled. The three algorithm are: (i) The *Minimizing cost with given expected availability* finds a subset of *DC*s that minimize the cost while fulfilling the required *QoS*; (ii) *Maximum expected availability with given Budget* to maximize the availability of the replica chunks without overstep a usable budget; (iii) *Optimal Chunks Placement* that operates taking in consideration the number of data center n and the number of the replicas r to find the optimal chunks placement. The authors reported in this work the pseudo-code of the above algorithms; these are mathematical-based algorithms that could be applicable in several different scenarios too.

The second work, Esposito *et al.* [66] conceived a selection strategy which makes use of the subjective preferences of the customers. The authors provided a fuzzy-inference-based mathematical matching algorithm to accomplish the storage service provider selection. The algorithm has to match the *QOS*-based users' requests to the crisp *QoS* provided by the Cloud storage providers (*SSP*). The algorithm can be run in a centralized or non-centralized manner. In the first case will be a *Broker* to collect the *QoS* levels of the Cloud Providers and to run the genetic algorithm; each *SSP* sends periodically this information to the *Broker*. In the non-centralized version, that is Dempster-Shafer theory-based, the customer' requests can be served by passing them directly to the *SSP*s involved in the selection. It is a general algorithm which could be applied on a single Cloud, Federation or Multi-Cloud scenario.

The third work, written by Papaioannou *et al.* [67], even if related to Storage Service Provider Selection, will be described in section VI-B3 because it is more focused on SLA management.

Finally Yao *et al.* [68] showed a Multi-Cloud architecture that by means a mathematical algorithm is able, starting from a set "*A*" of Cloud Providers, to optimally select a subset "*B*" of the available ones for data placement. The idea is to split the data in more chunks and store each of them in a different Cloud provider using ***IDA*** *(Information Dispersal Algorithm)*. The mathematical algorithm considers different factors for the best selection: (i) *storage cost* (upload and storing cost along the time), (ii) the service availability of a specific service provider; (iii) Network performance; (iv) Vendor lock-in; (v) Algorithm time execution cost. They define an objective function that takes into consideration all of the above factors. The best selection means to minimize the objective function by minimizing (i)(iv)(v) and maximizing (ii)(iii). The authors present the main components of the architecture and the pseudo-code of the mathematical algorithm.

Broker-based Architectures and Frameworks

Calheiros *et al.* [69] focus on the architecture of the *Cloud Coordinator* component of the Broker-based architecture presented in Buyya *et al.* [44]. In the latter, the *Broker* acts on behalf of users in looking for available Cloud services by means an interaction between *Broker* and the *Cloud Coordinator* (*CC*) that must be present in all the Cloud service providers.

Instead in [69], the user does not need support from the *Broker*, but he can directly interact with the ''*CC*'' that will discover and negotiate for the resources. The authors thought the ''*CC*'' as a data center without available resources. The main functionalities of the *CC* are: (i) to communicate with the other Cloud coordinators; (ii) to mediate the interaction between *CC*s; (iii) to intermediate the access to the local infrastructure and (iv) to determine the resources price. All the needed messages are SOAP-based message. Anyway this papers lacks of details regarding the algorithm used by the *CC* to select the best Cloud Service Provider (*CSP*) offering its *aaS.

Another interesting framework is the ''*SMICloud*'' proposed in Garg *et al.* [70]; this ***Broker***-based framework lets the comparison between several Cloud providers' offerings, helping the users in selection of services according with his/her requirements. The framework performs first a measurement of the services' **SMI** (Service Measurement Index) attributes, that dynamically vary over the time, and after performs a ranking of these services according to the measured **SMI** attributes. In this way the users can exploit the *SMICLoud*' features during the selection of Cloud services.

Another framework working at the IaaS layer, that uses **SMI** to select the right CPS, is presented in Subramanian *et al.* [71]. The authors proposed a Cloud *Broker*-based architecture that aims to provide an optimal virtual resource placement in a Multi-Clouds environment. The architecture counts 3 main actors: *Consumer*, *Broker* and *Cloud Provider*. All the Cloud Providers publish the offered resources in a central *Service Catalogue*. The placement process is divided in three steps. In the first one the *Broker* receives the user's service request description and the weights of the **SMI**. The *SMI* value is provided by a component that calculates the *SMI* values of the published offerings in the service catalogue. The user's request description contains all the needed information: type of application, VMs configuration, location, minimum **SMI** score, etc. In the second step the *Broker* calculates a possible set of Cloud resources and services that can satisfy the request. In the last step an *SMI*-based evaluation of the Cloud Providers is performed and a cost optimized placement is developed. The authors proposed for each considered parameter its mathematical model. Cloud services are ranked according to the Analytic Hierarchy Process technique(AHP). Anyway the paper focuses more on the ''ranking'' process than on the ''selection'' process of the ranked available services and furthermore it does not present any algorithm regarding such a purpose.

Another brokerage system called *BASMATI* [72] is presented in Santoso *et al.* [73]; it is a platform providing smart decision and optimization algorithm for the selection of Cloud resources. The Brokerage system aims to select the best CSP taking into account both the user requirements and the CSP's performance. The decision making algorithm reduces considerably both the vendor lock-in problem and the cost for the end user as well. This is possible thanks to the actions of the Cloud Broker that can help the user to select the most suitable CSP or to migrate from a CSP to a new more convenient one in terms of cost. The authors present the basic components of the *BASMATI* system, explaining the roles of each of them. These components are:(i) Application that sends to the system details about user requirements; ii) Monitoring that oversees the load in the system; iii) Knowledge Extractor that must me able to understand what the infrastructure needs to run the user application; iv) Decision Maker, this is the component that retrieves information from Knowledge Extractor and the Broker and makes decisions about the migration of a specific application; v) Resource Broker and vi) Application Controller play control the best available resources and the life cycle for the application; vii) CSP Manager talks with multiple Cloud providers through an interface CSP Manager and finally viii) *BASMATI*'s Platform component is the hypervisor level of the CSPs. However the authors do not presented the algorithm employed in the Decision Maker component.

Mathematical Algorithms and Models

The remaining works of this sub-set tackle the selection topic following mathematical approaches. The working area of their presented frameworks is the IaaS layer of the Cloud service stack.

Chaisiri *et al.* [74] a *Cloud Broker* and a mathematical algorithm to optimize the VMs placement across multiple Clouds are presented. But here the algorithm optimizes the placement taking into consideration three different provisioning plans: (i) *reservation*, (ii) *utilization* and (iii) *on-demand*. It considers the cost due to resources' in each of the considered plans. The algorithm aims to minimize the placement cost by optimizing a mathematical objective function. The system counts four main components: *Cloud Broker*, *VM repository*, *Users* and *Independent Cloud Providers*. In order to solve the placement problem the *Cloud Broker* executes the algorithm. That considers two use cases: (i) the amount of requested resources is a priori known; (ii) the requested resources are not known precisely. In the first case the problem can be solved by means a deterministic integer program; in the second case a stochastic integer program is needed. Anyway this paper focuses only on the mathematical algorithm without giving details about the technologies and standards used by the *Broker*.

In Lucas-Simarro *et al.* [75] the authors depict a *Broker*-base architecture able to work with several automatic scheduling strategies for a Multi-Cloud service deployment. The idea is to allow the users to distribute their services among multiple available Clouds in a transparent manner taking into account different factors (among these cost optimization and performance optimization). A service is composed by two or more components and each component can be deployed at a different Cloud Service Provider. The *Broker*-based middleware counts three main components: *Cloud Manager*, *Scheduler*, *VM Manger* (VMm) beside a central *DB*. The *Administrator* sets the configuration of the *Broker*, the *User* instead receives information from the *Broker* and sends a service deployment request describing the service by means a service description file. The *CM* periodically collects in the central *DB* all the information regarding the available resources and their price. The *VMm* finally deploys the virtual resources among multiple Cloud Providers managing and monitoring them. Moreover the authors give a wide explanation of the mathematical algorithms for scheduling processes.

Kurdi *et al.* [76] presents a combinatorial optimization framework (COM2) to develop a service composition across multiple-Clouds. The Framework has 4 main components: (i) A Multi-Cloud environment composed by several Clouds, Each one having a set of services file "*F*" and each file containing several services "*S*"; (ii) a user interface for the user's requests and to show the service composition sequence; (iii) the *Cloud combiner* which selects the suitable set of Clouds and composes a Cloud combination list based on that set; (iv) the *service composer* taking as input the combination list from the *Cloud combiner*. It determines the services that best fulfill the user's request, producing a service composition sequence. The selection algorithm aims to select a combination of services minimizing the overhead, and therefore maximizing the service performances beside a cost reduction. The user gives as input a set of service files. Starting from the Cloud with the higher number of service files, the Cloud combiner checks if there is an intersection between the set of the required files and the set of the provided *F*. If yes, the Cloud is added to the Cloud combination list and the *F* files in the composer list. The algorithm finishes when the *composer list* is equal to the set of the user's requested files.

Insights: As can be inferred from Taxonomy tree, among this set of papers, there are a lot of works that provide a wide range analysis of the selection approach. We had catalogue in three subsets: i) Selection Strategies, ii) Broker-based Architectures and Frameworks and iii) Mathematical Algorithm and Models. Most of the papers using similar approaches that works on different parameters or collection of parameters; they uses indexing techniques (based on Hamming distances or Service Measurement Index or through Analytic Hierarchy Process technique or etc) to support their selection strategies and algorithms.

TABLE 3. Paper Classification by Key-Parameters in a Federation[1].

		Papers																				
		[48]	[49]	[50]	[51]	[52]	[54]	[55]	[57]	[61]	[77]	[78]	[79]	[80]	[81]	[82]	[58]	[53]	[83]	[60]	[84]	[56]
6*SLA	SLO																					
	Du, Int																✓					
	Sca																			✓	✓	
	Sec & Pr						✓	✓								✓						
	Av, Re, Us										✓					✓				✓	✓	
	QoS			✓	✓				✓		✓						✓					
6*Cost	FC	✓										✓			✓	✓				✓	✓	✓
	Com													✓	✓							
	VM, Ser			✓													✓	✓	✓			
	St											✓					✓	✓				
	EP											✓	✓						✓		✓	
	CP																					
6*Perform.	ET																					
	RT		✓																	✓		
	Th									✓												
	Net									✓												
	SRT															✓						
	Lat															✓						✓
5*Other	FR		✓	✓	✓	✓	✓	✓	✓													
	Available Resources			✓					✓					✓				✓				
	VLi																	✓				
	Tr/Rep			✓			✓	✓						✓								
	CoR												✓									

2) PERFORMANCE

This sub-set includes about the 19, 8% of the papers of the "**Brokered**" sub-tree. Eight of them (Jrad *et al.* [62], Yao *et al.* [68], Jrad *et al.* [63], Subramanian *et al.* [71], Lucas-Simarro *et al.* [75], Garg *et al.* [70], Esposito *et al.* [66], Ngan *et al.* [64]) are shared with the sub-set "**Cost**" VI-B1.

Another work that could be linked to cost set is the one of Tordsson *et al.* [85] a Broker-based approach to optimize the VMs placement across multiple Clouds is introduced. The *Broker* has two main tasks: the optimal deployment and the interfacing management task. To accomplish the first task a scheduling mechanisms based on pseudo mathematical algorithm is provided. When the user asks for virtual resources it uses a *service description template* to specify the requested resources and optimization criteria. Cloud Providers are able to offer several VM configurations. The *Broker* has two main components: the *Cloud scheduler* and the *VIM* (Virtual Infrastructure manager). The first one performs the optimization algorithm, the latter one provides an abstraction layer on top of the heterogeneous Cloud Providers by means a uniform and generic interface to communicate with the other different Cloud Providers. The VIM is an open-nebula-based system. The algorithm maximizes a mathematical optimization function named "*TIC*" (Total Infrastructure Capacity). To this aim the authors select a scheduler based on the so called *AMPL* modelling language to solve the optimization function.

A different approach was followed by Rehman *et al.* [86] and Duan *et al.* [87]; they propose selection algorithms without considering cost factors in their execution but they consider instead a multi-factor performance optimization. In the first one a multi-criteria Cloud service selection methodology is presented. The authors did a mathematical formalization of the problem. The service selection process is based on the comparison between the users' requirements and the descriptor vectors of the service. The algorithm selects the service which has the descriptor vector that best matches with the user requirement vector. Due to the generality of this algorithm it is possible to consider any kind of functional and non-functional requirements. In the second one [87], instead, the authors present a multi-layer framework able to perform a federated selection of network and Cloud services in a SDN environment. Specifically they propose a management platform to make simple the selection and composition of network-Cloud services. The platform is called **SDCE** and integrates the software-defined networking and software-defined computing. The architecture presents 5 main layers. In particular the layer called "Service Layer" provides the features about the service selection and the orchestration. A specific registry is used to check the availability of network/Cloud service in order to perform a selection among them and to compose an end-to-end network-Cloud service. The result is a Network as a Service (NaaS). The adopted selection algorithm composes the service taking into account the users' requirements meeting the specified QoS. In particular, it focuses on the optimization of the network performance based on the minimum bandwidth and the maximum delay for data transmission. However in the paper the term "federated" is referred to the resulting composite service and not to a federation among several Cloud providers. Therefore it is logic consider this work about a Multi-Cloud environment where each Cloud provider does not have any awareness of the other Cloud peers.

In Carvalho *et al.* [88] the authors assert the necessity in using a dynamic selecting approach for multi-Cloud providers. Starting from the *PacificClouds* architecture proposed in [89], they focus on hosting (i.e., deploying and managing) applications based on *microservices*. The authors define microservices as a set of autonomous, independent, self-contained services, in which each service has a single goal, is loosely coupled, and interact to build a distributed application. Their selection process takes into account both the user and the microservice requirements (i.e., constraints) to select the Cloud providers, that means to select each Cloud service for each microservice. The Clouds selection uses the *Simple Additive Weighting (SAW)* method to rank the competitive Cloud providers (i.e., a set of candidate services). The parameters that have been chosen for the service model formalization are the *application response time (execution time plus delay)*, *Cloud availability* and the *application execution cost*. All parameters are based on user-defined thresholds. The authors also developed a tool and set up scenarios to evaluate their approach. In order to map the multi-Cloud selection process, they develop the *multi-choice knapsack problem* by using the *dynamic programming technique*. Although the authors do not explicitly refer to the use of a broker, we can deduce their algorithm can be implemented at a centralized broker. More specifically, the work selects a provider to a microservice based on *Multi-Criteria Decision Making (MCDM)* methods.

Another interesting approach is presented in Kurmai *et al.* [90]. This work specifically presents an optimization problem in the context of Multi-Cloud brokered systems for IoT use cases, where multiple targets, such as broker profit and request latency, are pursued. In particular, a multi-objective particle swarm optimization (MOPSO) approach has been proposed by the authors, and shown outperforming GA and random search approaches. In [91] the authors present a broker based adaptive learning algorithm to analyze user requests, decompose them, and assign *micro-requests* extracted this way to a series of services exposed or provided by Multi-Cloud environment. **Insights:** The works considered in this section are mostly related to optimization of the execution (for single or multi parameters).

[1] *Tables Legend*: SLO = **SLO**; Durability = **Du**; Interoperabilty = **Int**; Scalability = **Sca**; Security = **Sec**; Privacy = **Pr** Availability = **Av**; Reliability = **Re**; Usabability = **Us**; Financial Charges = **FC**; Communication = **Com**; QoS = **Qos**; VM = **VM**; Service = **Ser**; Storage = **St**; Electric Power = **EP**; Computing Power = **CP**; Execution Time = **ET**; Response Time = **RT**; Throughput = **Th**; Network = **Net**; Service Response Time = **SRT**; Latency = **Lat**; Functional Req = **FR**; Geographic Constrains = **GC**. Available Resources = **AR**; Vendor Lock-in = **VLi**; Trust= **Tr**; Reputation = **Rep**; CO2 Reduction = **CoR**;

TABLE 4. Paper Classification by Key-Parameters in a Multi-Cloud Context. PART-1[1] = Parameters Not Specified, ✓ = Parameters Specified.

		Papers																					
		[92]	[93]	[62]	[63]	[64]	[69]	[70]	[65]	[66]	[67]	[68]	[85]	[71]	[74]	[75]	[76]	[86]	[87]	[94]	[95]	[96]	[97]
6*SLA	SLO																			✓			
	Du, Int.							✓			✓		✓	✓							✓		
	Sca						✓																✓
	Sec & Pr		✓					✓		✓		✓		✓								✓	✓
	Av, Re, Us	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓		✓				✓				✓	✓
	Qos							✓	✓	✓							✓		✓				✓
6*Cost	FC	✓		✓	✓	✓	✓		✓	✓	✓			✓	✓	✓	✓						
	Com					✓		✓			✓	✓					✓						
	VM, Ser					✓	✓			✓			✓		✓	✓						✓	
	St	✓	✓			✓		✓	✓	✓	✓	✓										✓	
	EP																						
	CP							✓			✓												
6*Perform.	ET													✓									
	RT			✓	✓			✓															
	Th			✓	✓								✓										
	Net	✓	✓							✓		✓					✓		✓				
	SRT									✓				✓									
	Lat																						
5*Other	FR						✓														✓		✓
	AR																✓						
	GC					✓					✓												
	Tr/Rep																	✓					✓
	VLi											✓		✓									
	CoR																						

TABLE 5. Paper Classification by Key-Parameters in a Multi-Cloud Context. PART-2[1].

		Papers			
		[59]	[73], [72]	[88], [89]	[90]
7*SLA	Int				
	SLO				
	Du	✓			
	Sec & Pr				
	Av, Re, Us	✓		✓	
	QoS				
7*Cost	FC	✓		✓	✓
	Com				
	VM		✓		
	St				
	EP		✓		✓
	CP				
6*Perform.	ET			✓	
	RT	✓		✓	✓
	Th				
	Net				
	SRT				
	Lat			✓	
5*Other	FR				
	AR				
	GC				
	VLi		✓		
	Tr /Rep				
	CoR				

3) SLA

This sub-set contains about the 32, 1% of the papers of the "**Brokered**" sub-tree. The papers Jrad *et al.* [63], Garg *et al.* [70], Subramanian *et al.* [71], Mansouri *et al.* [65] and Sundares *et al.* [37] are just analysed in the VI-B1 while the approach presented in Redl *et al.* [94] falls out of that sub-set although it has a *SLA*-based approach like them. In this paper the authors present an automatic method to find a semantic equality among different *SLA* elements and a method which allows an automatic selection of optimal services. The system is based on a Cloud market platform that performs a semantic matching between different *SLA* template documents. The algorithm is based on a probability of equivalence of the *SLA* templates. This value is used to calculate the *SLA*s' elements equivalence. In order to provide an automatic selection of the Cloud Providers, a semantic equivalence between the user's private *SLA* template and more public *SLA* templates is assumed. At the end of the process the public *SLA* template with the highest equivalence probability is chosen as the optimal offering. The algorithm anyway is able to select only one of the available Clouds.

Papaioannou *et al.* [67] introduced a broker-based architecture named *Scalia*, able to continuously adapt the placement of the stored data among several Cloud Providers, taking in consideration the users' *SLA* requirements and the access pattern to the data. The data is split in n chunks and they are stored among m Cloud Providers with $n = m$; the solution is able to reconstruct a complete copy of the data from a m-subset of Clouds. *Scalia* has a three layer architecture (i) Engine Layer, (ii) Caching Layer (iii) Database Layer. Our attention falls on the first layer which is the one responsible of the best data placement choice. It is composed by multiple engine components able to manage independently the selection algorithm. It considers the object access history, which is a list of statistics of that data object (used storage, incoming bandwidth, outgoing bandwidth as well as the number of operations). The algorithm uses the user's *SLA* requirements (e.g. durability, availability, etc..) to pick out, from all the possible combinations of available Cloud Providers, the one with a lower price than desired by the user. In other words the authors want to minimize price, while satisfying the minimum availability, durability, and lock-in constraints.

In D'Andria *et al.* [95] a semantic match-making solutions to find the best overlapping between the user's requirements and the Provider's offering is exploited. More in detail D'Andria *et al.* present the Cloud4SOA project which introduces a Brokered architecture aiming to face the semantic interoperability between PaaS service providers. In order to reduce the vendor lock-in problems in the PaaS Cloud layer, a uniform and global language in PaaS offering definition is needed. The main goals of Cloud4SOA are: (i) to help the application developers in deploying their application at the PaaS provider that better meets their necessity; (ii) to allow the applications to be deployed and seamlessly migrated between PaaS Cloud Providers using the same technologies but a different language to define and model them. CLOUD4SOA uses a semantic-based matchmaking process to equalize the user application requirements' terms to the PaaS offerings ones. The powerful of the Cloud4SOA semantic matchmaking process is to harmonize the differences between different terms standing for the same concept. through a set of relations between those terms used by PaaS providers. In order to deploy an application on a specific PaaS Provider, one taking into consideration the application's requirements, the "deployment" module of Cloud4SOA creates a specific application descriptor that is compatible with the format used by the PaaS Provider chosen by the user to place that application. The "migration" module has to semantically translate the application requirements in a new application descriptor compatible with the new PaaS Provider.

Finally Massonet *et al.* [96] faces the application deployment optimization in a Multi-Cloud environment taking into account not only the cost factors but the security constrains too. The case of study of the work is an application deployed over different Clouds (public and private) belonging to different administration domains. The idea is to place the web servers and application servers close the customers (UK and Germany) and the database server in a private Cloud located in Spain. The authors assert that the during the selection process of the best CSP where to deploy an application component security considerations have to be considered. To this end the application components have to be modelled by making a description of the artifacts, artifacts' security, scalability requirements etc. This model construction represents the "configuration" phase of the deployment workflow. The second phase is the "deployment" in which a software component considering the constraints present in the model build a deployment plan. Specifically the "Reasoner" component will perform a matching process between the application's security and the CSP features. The third and last phase is the "execution" where a other software component will execute the deployment plan. The selection process uses security service level objectives *SLO* in the *SLA* to create a common language in describing the security assurances. The "reasoner" through the analysis of the deployment model creates a"utility function" that optimizes the deployment goal. The paper shows the pseudo-mathematical equations aiming to optimize the deployment cost and to select a CSP per component. The selection process for each application's component will select from those CSPs that meet the security constraints that one having the lower price. We decided to put this work in this section because the first phase of the selection performs a matching between the security constraints of the component and the CSP features.

An interesting work is presented by Karimi *et al.* in [98]. The authors describe a genetic algorithm for the composition of services spread over the involved clouds, that are clustered according to the services offered. Another interesting work is presented by Lin *et al.* [99] (SLA, Cost, Other), where the authors describe another approach useful to select the best CSPs offering services that satisfy the user's requests. The system catalogues and, when needed selects clouds through an indexing tree that takes into account several parameters of the cloud services (e.g. cost, type, QoS, instance size and so on).

Insights: The works described in this section have in common the approach used to solve the CSPs selection, through matchmaking algorithms. Even if the works under analysis are related to Multi-Cloud scenarios there isn't an approach specialized for these characteristics.

4) OTHER

The papers contained in this set represent the 19, 8% of the Brokered Multi-Cloud papers analyzing algorithms. The 62, 5% of the papers in this set is in common with the previously analyzed "*Cost*" set in VI-B1. The 31, 25% is shared with the "*Perfomance*" one discussed in VI-B2 and the 75% is shared with the "*SLA*". Fan *et al.* [97] consider a mathematical approach based on the user's recommendations according with customer satisfaction in service usage.

Specifically Fan *et al.* propose a mechanism that performs a service selection taking in consideration multi-dimensional users' trust feedback ratings to create a reputation for a specific Cloud service provider. The mechanism is based on two different kinds of databases: (i) "trust value db" and (ii) reputation value database. The user of a target Cloud service assigns, according to his satisfaction, the local trust value to the used service, storing on the service that value. The reputation base, instead, stores the reputation value of each service which is obtained by aggregating all the users' feedback.

In [84] the authors present an energy-aware Brokering Algorithm (eBA) allowing to push down carbon dioxide emissions through a Multi-Cloud ecosystem by running instances at the most convenient Cloud sites. The work addresses medium and small size Cloud providers towards solutions allowing them to compete with large Cloud providers in a more sustainable service marketplace. The authors watch to a dynamic scenario where Cloud providers share their IT resources among a *community* of Cloud sites (i.e., datacenters) in order to reduce costs and energy-efficiency gap if compared with the top Cloud computing service providers (e.g., Amazon, Google, Rackspae, etc.). An automated negotiation process facilitates the bilateral negotiation between the centralized broker and multiple providers to achieve several objectives for the community members. The eBA algorithm has been designed to make the best choice in resources allocation based on a set of sustainability parameters at each Cloud site, power consumptions, service running time, number of offered instances in each offer, number of instances in each request, availability and service price.

In [100] is focused on power consumption optimization done by an algorithm to select the minimum set of Clouds that expose IoT services able to satisfy the users' requests in an energy-efficient manner. A nice preliminary work is presented in [101], where the authors define a SLO-ML (Service Level Objective Modelling Language) that, through a broker-based architecture, is able to construct cloud applications requested by the user through a pair of models defined by SLO-ML and by Infrastructure as Code (IaC) (e.g.,Terraform HCL, TOSCA,etc.). The last work analyzed in this category is [102], it presents a model for brokering cloud service plans based on Fuzzy logic and inferences. The approach presented helps in assessing cloud service plans according to the user's requests. The presented broker supports user requests expressed in a natural language, based on high-level concepts.

Insights: The works presented in this section are focused on selection algorithm of the best provider, working on one or more parameters.

VII. DECENTRALIZED SOLUTIONS

The decentralized solutions make up branches of the taxonomy tree that aren't strongly populated. Most of the works are in the field of Federation, and most part of the latter analyzing the problem from the perspective of cost optimization. The works related to Multi-Cloud scenarios, instead, analyze the cloud selection problem from a range of viewpoints, spanning all categories.

A. DECENTRALIZED APPROACHES IN A FEDERATION

All the following works tackle the optimal provider selection in a decentralized way. These works represent the 47, 6% of the papers focusing on Algorithms to be leveraged in the Federation context. All the papers in this sub-section consider multiple parameters like ''Cost'' and ''SLA'', or ''Cost'' and

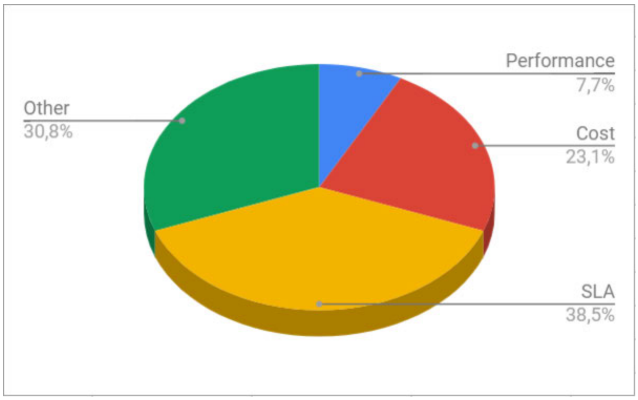

FIGURE 5. Pie Chart Concerning the distribution of the paper in each sub-category.

''Other'' constraints, with the exception of Kertesz *et al.* [77] considering only the ''SLA'' ones. As a consequence, we can simplify the reading by separating the papers tackling the Cost optimization from those ones aiming to optimize the SLA parameters.

1) COST

Carlini *et al.* [78] present a novel system to select Cloud providers in a decentralized federated environment. The system is able to host services enhancing the overall performance, reducing the hosting Cost and improving the profit. The presented solution has been modelled in a Markov-chain. The entities of the system are Clouds and services. Each Cloud can host more services, with each service having a specific cost; the budget of a Cloud is computed as a sum of its resources. The idea is to avoid that, during the assignment process, a Cloud exceeds its budget and thus the final profit is maximized. The authors provided mathematical model for the assignment problem.

Giacobbe *et al.* [79] contribution surely falls in Cost category, but it should be considered a work belonging to the other category. The authors in fact present a selection algorithm that makes able the federated Cloud Providers to determine and so choose the best destination, from the reduction carbon dioxide point of view, where to migrate their VMs. The algorithm counts two steps, the first one creates a ''destination/granularity matrix'' for ''energy cost-evaluation'' in terms of carbon dioxide emissions-per-kWh, the second one instead finds the ''optimum migration path for carbon dioxide emission reduction''. During the first step of the algorithm a monitoring of each site for the forecast period is performed. It evaluates energy consumption at the i-th site for a specific workload that has to be migrated. If the monitored i-th host site has sufficient capabilities to manage the considered workload then it is added to the destination matrix. The second algorithm's step has in input the destination matrix and extracts from it that destination with the minimum CO2 emission for the considered forecast period. Moreover it gives as output a matrix with the best migration path.

Even Yeh *et al.* [80] work is placed in the same set of papers and its authors present a dynamic way to allocate resources across multiple Clouds in a federated environment. The basic considerations made in this work are similar to those discussed in Celesti *et al.* [55] (placed in the successive set of papers). Anyway the Federation here is created following the concept of *Cross-IdP*. We can see this concept as a ***chain of trust***. It works like the syllogism concept: *if* $A \xrightarrow{trusts} B$ & $B \xrightarrow{Trusts} C$ *Then* $A \xrightarrow{Trusts} C$. So when a Cloud provider needs of external resources, it asks for them to the *IdP* where it has a trusted relationship. Firstly a local Cloud asks for resources to those external Clouds having a trusted relationship with the same *IdP* (first relationship layer). Therefore, if the provided resources in this first layer are not sufficient to fulfill the user's request, it virtually considers the external Clouds in the first relationship layer as local Clouds and then forwards the request to the Clouds having a direct trusted relationship with them and so on. The algorithm finishes when the external rent resources meet the requested ones. The algorithm takes into account three main parameters. (i)*The IdP's reliability value* that is incremented of 0,05 whenever that IdPs is used; (ii) *Resource information* of each Cloud provider (idle resources and requested resources); (iii) *The network transmission cost* between two Clouds. The first step is to choose the *IdPs* with the major reliability value. Therefore, in order to select the external Cloud, the ration $\frac{IdleResource}{NetworkCost}$ with highest value is chosen. If the request is fulfilled the algorithm finishes; otherwise the research continues first in the same layer and, if the layer is not able to fulfill the request, in the other layers. Rebai *et al.* [81] proposed a mathematical algorithm that helps the Cloud Providers in a Federation to automate and optimize the resources allocation among multiple Clouds. The users' requests are modelled as *Graph*. The nodes are the requested VMs and the edges are the traffic flow between neighbouring nodes. The algorithm is executed in each Cloud provider and aims to optimally distribute the requests across the Federation in order to maximizing revenues and to minimize the Costs at each provider.

In [83], the authors set a "light" Federation where providers announce their service prices to each other through a *data exchange system* identifying a *publish subscribe system* or a *marketplace*. The proposed algorithm is used by each Cloud provider involved in the Federation individually and locally, thus maximizing the revenue respecting constraints without put their resources in common. In order to serve workloads and service demands coming from the end users and the Federation members, each federated Cloud has a limited amount of CPU, memory, storage and communication resources to use and share. Out-sourcing and in-sourcing decisions depend on the actual Costs of provisioned services from each provider. Moreover, providers can also put in "sleeping mode" or shut down unused machines or resources to reduce operational Costs and to minimize the energy consumption. This process is enabled by a dynamic pricing model and it is an example of Cloud Federation optimization problem based on the Costs-benefits analysis.

Insights: The works here analyzed are mainly related to mathematical approaches useful for optimization of the CSP selection process. The works presented here are based on a federation broker that works in a distributed way upon the federated CSP.

2) SLA

Paper [54] falls in two sets: "**SLA**" and "**Other**". It proposes an approach for the Federation establishment considering generic Cloud architectures according to a *three-phase model*. It represents also an architectural solution for Federation by means of a Cross-Cloud Federation Manager (CCFM): a software component in charge of executing the three main functionalities required for a Federation. In particular, the component explicitly manages: (i) the discovery phase in which information about other Clouds are received and sent, (ii) the match-making phase performing the best choice of the provider according to some utility measure and (iii) the authentication phase creating a trusted context for the federated Clouds. In particular the authors give a deep mathematical explanation of the matchmaking algorithm to select the Cloud provider that better meets the Cloud consumer's requirements.

A high level description of the *XACML*-based algorithm is provided in Celesti *et al.* [55]. The matchmaking selection is based on two different evaluation tasks. The first one takes as input the set of the discovered Cloud Providers $S_D = A, B, C, D \ldots.$ and gives as output a new subset of Clouds $S_R = A, B, D$ that better satisfy the home Cloud requests based on the resources availability (CPU,RAM). The second evaluation step takes as input the set of the discovered Cloud Providers $S_D = A, B, C, D \ldots.$ and gives as output a new subset of Clouds $S_{IdP} = A, D$ having a trusted relationship with the same IdP where the Cloud consumer has a valid identity. Therefore an intersection of the two output subsets results in a matching subset $S_M = S_R \cap S_{IdP}$. The algorithm considers as metrics requested resources R_{req} and offered resources $R_{off}(F_i)$ by the Cloud Providers F_i. The matchmaking agent sorts the S_M subset in a offered resources descending order $F_{ord(S_M)}$ and considers the first k Clouds of the $F_{ord(S_M)}$ that satisfy the condition $R_{req} \leq \Sigma_{i=1}^{k} R_{off}(F_i)$, $1 \leq k \leq n$.

Finally Kertesz *et al.* [77] proposed a solution to manage a federated environment by using a way looking like to a distributed brokered system. Each federated Cloud has its own broker that manages the internal resources. The proposed architecture is an entry point for the Cloud Federation. The most important component is the "meta-brokering" (interface) able to interconnect different Cloud *Brokers* in the system. It is able to decide among *Brokers* taking in to account the metrics gathered from a service monitoring subsystem (**SALMon**). An important key actor of the system is the *Generic Service Registry* where the services' information are stored (WSDLs, VMIs VAs). The meta-brokering layer receives the users' service calls, checks if the service exists in the **GSR**, and selects a suitable *Cloud Broker*. It runs a matchmaking algorithm that combines the **GSR** information, the status information on Broker and **SALMon**. The GMBS (Generic Meta-Broker Service) builds a *Cloud Federation* interconnecting different Clouds by means *Broker* interconnections. The incoming calls are queued in a specific IaaS system and after they are scheduled to available VMs in a queue managed by the *Broker* that IaaS system; the calls are associated to the Virtual Machines contained in that queue. More simply the *Broker*'s task is to manage the arriving service calls and the queue of the VMs' queue, associating a call to a specific VM able to meet the call's requirements.

Insights: The works here presented are related to matchmaking solutions that work on a distributed broker architecture or based upon cooperative broker infrastructure.

3) PERFORMANCE

Hadji *et al.* [82] contribution follows in this section. The authors proposed Gomory-Hu based on a scalable and decentralized solution in a Cloud federated environment aiming to minimize the execution time simultaneously and the maximization of the revenue. The proposed algorithm performs selection and placement decisions for the best allocation of critical resources taking in account several different cost parameters such as hosting costs, network costs etc. To place user's requests, the algorithm evaluates the requests by analysing the graphical representation of request, in which the computing nodes (VMs) are the vertices and the connection between two VMs are the links. Both vertices and links have a weight representing the importance of the nodes. The idea at the basis of the algorithm is to place the critical (higher weight) nodes inside the local provider and the secondary nodes and links into other Cloud providers taking part to the same Federation. The weight of a node is assigned by analysing it on the basis of stronger protection and security higher availability.

B. DECENTRALIZED APPROACHES IN A MULTI-CLOUD

How already mentioned at the beginning of the section V the "*Multi-Cloud*" sub-tree is strongly unbalanced. The 89, 19% of the *Multi-Cloud* tree leaves, discussing about Algorithms, falls under the "Brokered" branch. It is due to the lack of a holistic awareness, on the Cloud service providers' part, of the surrounding Cloud panorama. Usually the Cloud Providers in a Multi-Cloud environment make available tools, APIs, libraries to make accessible their own resources to other Cloud entities. As a consequence, a brokering entity is necessary in order to manage these multitudes of APIs, and tools provided by heterogeneous Cloud Providers. With reference to Figure 1 the first two papers falling under the "Decentralized" node are Negru *et al.* [92] and Kajiura *et al.* [93]. Both of them share the **Cost** and **Performance** subsets, while Kajiura *et al.* [93] presented a SLA-based approach too.

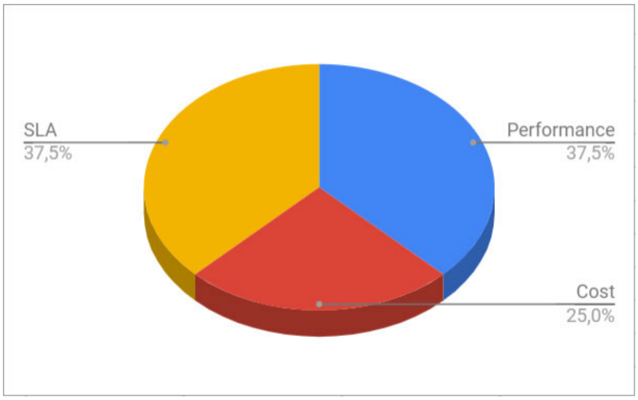

FIGURE 6. Pie Chart Concerning the distribution of the paper in each sub-category.

More specifically Negru *et al.* [92] present a mathematical algorithm to optimally select a sub-set of Cloud Storage Service Providers (**CSSP**). They store large amount of data coming from different sources geographically distributed. In their presented scenario, they consider to have included several distributed sources acquiring environmental data. Data from each source are stored in a different **CSSP**. Finally the whole amount of data of all **CSSP** is sent to a central Cloud Provider to be processed. The algorithm is matrix-based and performs an overall best provider selection considering cost and latency constrains. Anyway it is a general algorithm and it is not referred to a specific Cloud scenario.

Kajiaura *et al.* [93], instead, propose a mechanism for the dynamical choosing of the optimal Cloud service provider combination in a heterogeneous Multi-Cloud environment. The authors assume environment is composed by several Cloud Storage Service Providers **CSSP**. Each service they implement has the related XML-based SLA file containing the mandatory conditions. The authors consider 4 indicators for the users' requirements: (i) costs; (ii) confidentiality; (iii) availability; (iv) transfer time, and 4 *SLA* items for the services: (*a*) costs; (*b*) leakage probability; (*c*) operating rate; (*d*) communication speed. Authors mathematically present the correspondence between the users' requirements and the Clouds' services SLA items. Even though their selection technique is presented as a user-centric solution we consider the presented algorithm can be generally useful and applicable in several selection contexts.

The work presented by Mezni et. al [103] is an example of performance and SLA-based CSPs selection, useful for the composition of services spread over a Multi-Cloud environment. The solution proposed is based on a service composition enabled by a derived lattice theory approach that is the Formal Concept Analysis (**FCA**), a clustering technique for knowledge representation, data analysis, and information management used to select the best combination of CSPs that are able to host user's request, whilst optimizing usage of resources in particular networking.

Another interesting work was proposed by Farokhi *et al.* [104] for a hierarchical management of SLA-based service selection (**HS4MC**). The approach discussed realizes the service selection in two steps: 1) new SLAs, named ''InterCloud-SLAs'', are constructed, and contain the provider's requirements; regarding QoS, those are after split between functional and not-functional requirements. 2) using a selection algorithm, the system selects the appropriate services that satisfy the request.

VIII. CROSS-CATEGORY ANALYSIS

In order to have a complete overview of the approaches used in the analyzed publications, we made another kind of categorization that analyzes and catalogs the papers without taking care about cooperation pattern (multi-cloud or decentralized) and management schema (brokered or decentralized), but we take care only about selection parameters used in the selection (SLA, Cost, Performance, and Other).

In our analysis we have identified three main categories of work: i) Optimization techniques, ii) Algorithms, and iii) Frameworks and Architectures; in particular the second category, Algorithms, albeit further categorized according to a few subcategories, the latter are quite varied, even if we can still identify most popular approaches: i) Selection, ii) Match-Making, iii) Multi-Criteria Evaluation, and iv) Best Placement. Figure 7 provides an overview of the results obtained from the cross-category analysis.

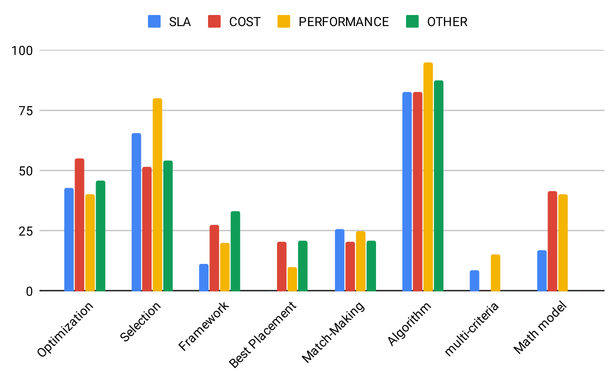

FIGURE 7. Distribution of the papers along the categories coming from cross-category analysis.

A. SLA

The Service Level Agreement (SLA) represents a policy of cooperation based on a priori agreements between interested parties; the agreements so negotiated may be related to one of the parameters characterizing the resources provided in the cooperation among CSPs. In the taxonomic tree shown in Figure 8 the papers referring to SLAs are organized in terms of the parameters involved that, as the reader can see, span the space, from *Durability*, to *QOS*, including *Scalability, Reliability, Fees, Privacy*, and so on.

Furthermore, Figure 8 provides a visual categorization of papers in terms of both the categories introduced in Section VIII and also in relation to SLA as function of perspective on the analysis (i.e., the authors take care of: i) provider's SLA, ii) user's SLA, and iii) not specified explicitly). Each category highlights the kind of algorithmic approaches pursued by the authors (e.g., selection algos, optimization algos, matchmaking, and so on). As it is possible to see from Figure 8, the most popular approaches belong to three categories of algorithms: selection, optimization and matchmaking.

The taxonomy tree highlights also the percentage of occurrence: 65.71% for Selection, 42.86% Optimization, 25.71% Matchmaking, 11% Framework & Architecture, 17.14% Mathematical Models, 8.57% Multi-criteria evaluation; in general Algorithm branch contains 82.85% of publications.

B. COST

The solutions using the cost as criteria of provider selection aim to minimize the cost associated to the request. Cloud service Provider selection is hardly influenced by the costs evaluation, the attribution of a cost to an allocation of resources, (or to an operation) may depend on several factors, such as: financial cost, communication cost, VM's instantiating cost, Service's execution cost, Storage cost, Electrical power consumption or about Computing power.

Cost is central factor in the decision when the scenario contains the word ''multi-Cloud'' or ''Federation'' (that means different actors with different pricing models), and as it is possible to see in the following taxonomic tree (Figure 9) the works identified in literature are splitted in three categories: i) Algorithms, ii) Architecture & Frameworks, and iii) Cost Optimization; as introduced in Section VIII. Most of the works analyzed (82.75%) are related to algorithm solutions; respectively the other macro-category contain 27.58%, and 55.17%. The sub-categories of *Algorithm* are taken in account and the respective percentages are shown (51.72% Selection, 20.69% Matchmaking, 41.38% Mathematical Model, and 20.69% Best Placement).

In Figure 9 are also shown the some peculiarities of the works under analysis that have few relevance, and in order to avoid color collisions and confusion on the works presenting multiple characteristics we have added a branch to ''Algorithms'' related to the ''Mathematical Models''.

C. PERFORMANCE

The publications related to the Performance are, even in this case contained in the three categories: i) Algorithms, ii) Architecture & Frameworks, and iii) Performance Optimization. The first category contains 4 subcategories: i) Multi-criteria Evaluation, ii) Selection, iii) Best Placement, and iv) Match-Making. Similarly to other subsections, even in this case we have observed that the majority of paper are works describing algorithms. Figure 10 depicts how the publications are grouped in the taxonomic tree. Also in this case the most populated area is related to ''Selection Algorithms'',

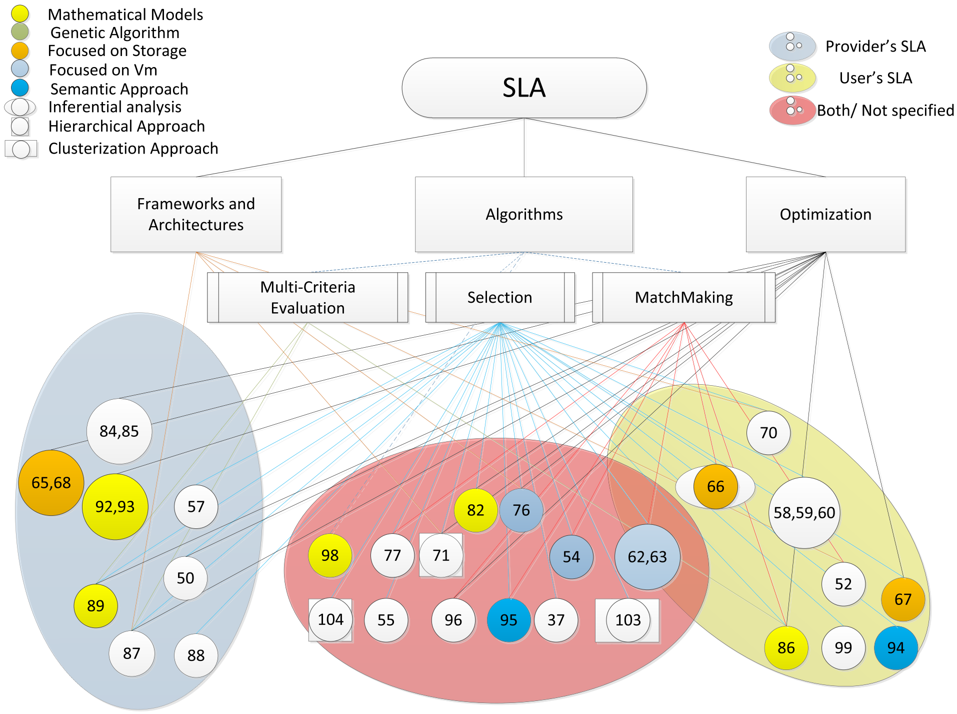

FIGURE 8. Taxonomy Tree related to SLA-based Approaches.

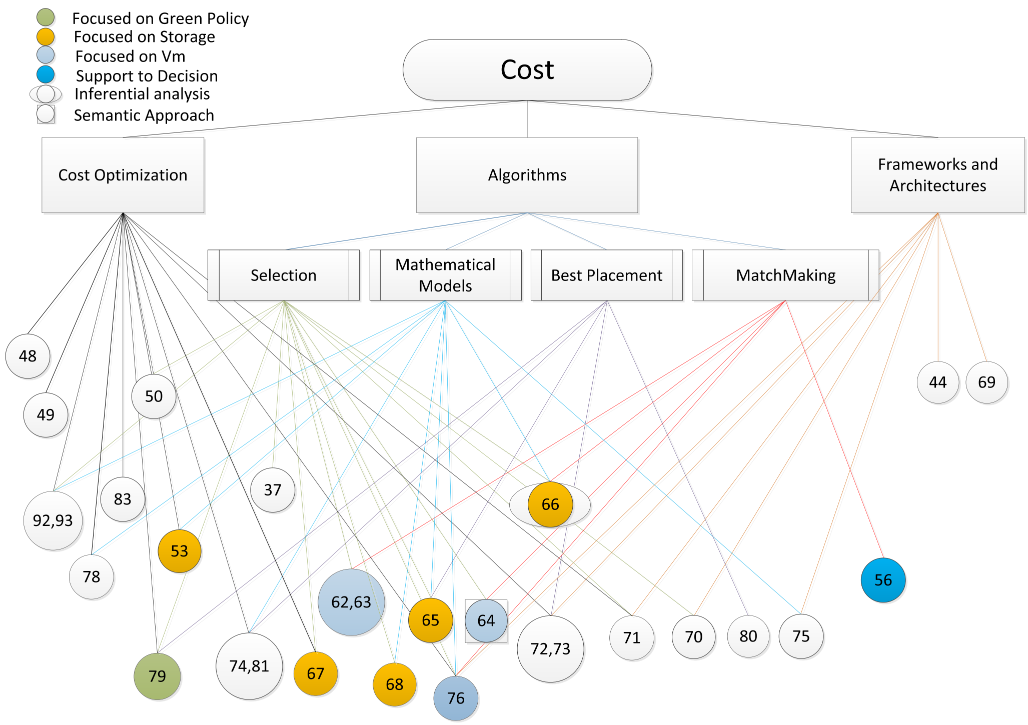

FIGURE 9. Taxonomy Tree related to Cost-based approaches.

"Mathematical Models" and "Optimization", respectively the 80%, and the other two 40%. The category "Matchmaking" follows with 25%, "Framework and Architecture" with 20%, "Multi-criteria" 15%, and "Best Placement" with 10%. As general consideration the 95% of works fall (directly or not) in the category "Algorithms".

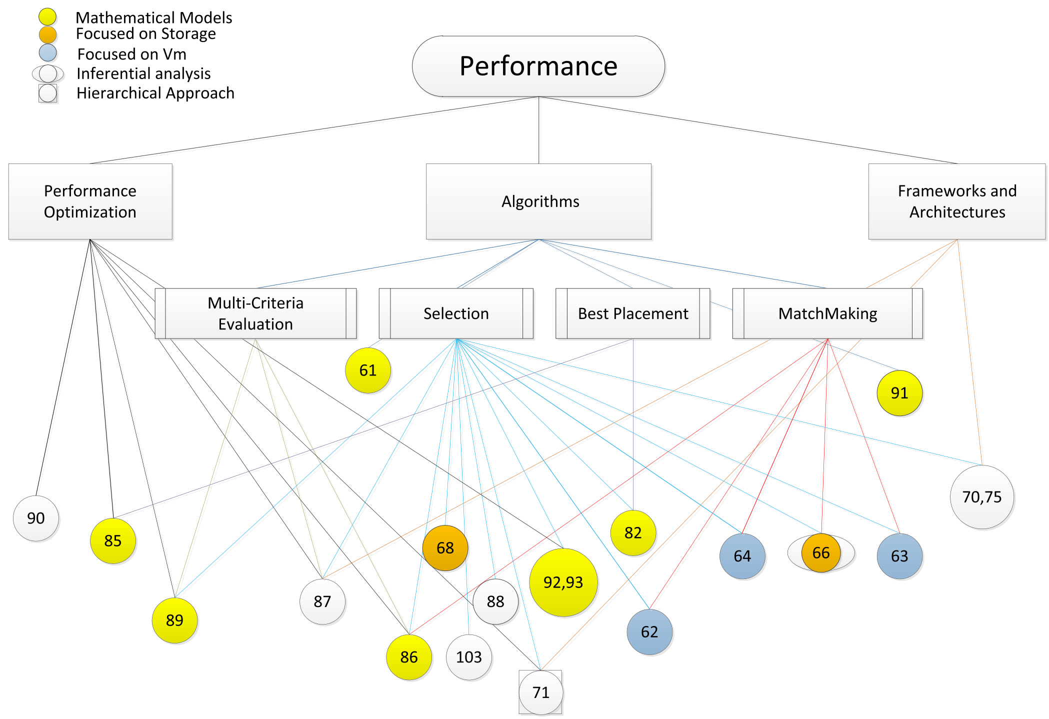

FIGURE 10. Taxonomy Tree related to approaches for Performance.

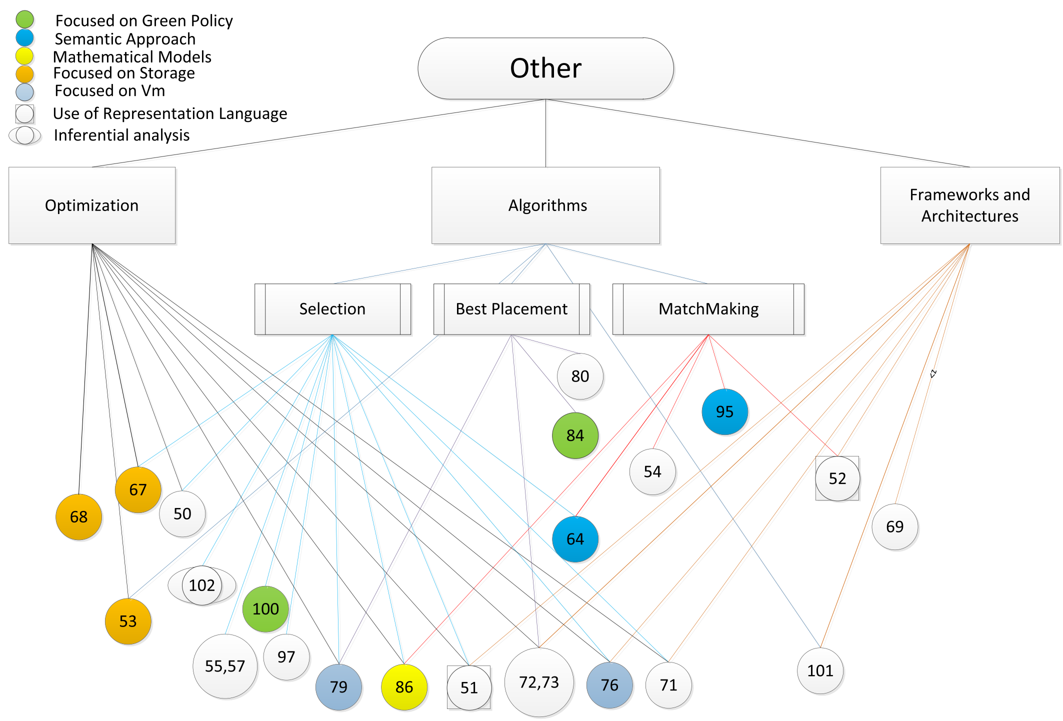

FIGURE 11. Taxonomy Tree related to Other Approaches.

D. OTHER

This category contains the systems using as approach (or characteristic) something useful to select the CSP in which deploy or move workflows or resources, that is not considered in the previous three categories. Similarly to the other cases the most populated area is related to "Optimization" and

"Selection Algorithms" with respectively the 45.8%, and the 54.16%. The category "Framework and Architecture" follows with 33.33%, instead both the other two contain the 20.83%. As general consideration the 87.5% of works fall (directly or not) in the category "Algorithms".

IX. SUMMARY, CONCLUSION AND FUTURE DIRECTION

A. CONTRIBUTION

In this survey we investigated an important topic in the Cloud computing landscape, by reviewing, firstly, the European efforts in solving *Cloud Federation* or *Multi-Cloud* issues and, secondly, analysing works available in literature about the automatic Cloud service provider selection. We conducted our analysis organizing the 87 related papers we found in a tree structure, and placing each of them under a specific subtree, according to the type of solution being presented. We deeply examined the covered solutions, differentiating the works presenting mathematical algorithms from those adopting other types of approaches, e.g., heuristics, also highlighting the papers that present a framework of their own. Our analysis highlighted that the majority of the papers propose *Multi-Cloud* solutions, and in particular centralized brokering approaches, while only a few papers provide solutions for federated environments. In general, while there is a (limited) number of other existing surveys on the topic of approaches to the selection of Cloud provider services, to the best of our knowledge none of the works in literature has included the requirements due to federations specifically.

B. FUTURE WORK

Our work highlighted the current shortage of investigation concerning the topic of selection of Cloud providers for services in a federated environment, particularly with regard to decentralized solutions. Only about 17, 24% of the total amount of inspected works tried to improve the process of selection, considering available Cloud providers as being part of a federated scenario. In our opinion this situation represents a shortcoming, because "Centralized Brokers" are losing their appeal. Moreover it can be possible to extrapolate many of the "Broker" features (discovering, ***selection/matchmaking***, or adaptation, between several different standards), implementing them in a decentralized way. Besides to the discussed advantages provided by the "Federations", as presented in section I, we think that a decentralized solution results to be more fruitful from the point of view of research challenges and benefits. In fact "Federation" is a strongly dynamic environment where Cloud providers may join and leave whenever they choose to. As a consequence a decentralized "brokered" approach (where each Cloud provides by itself to arrange relationships with others Clouds) could be more responsive and efficient than a centralized one. Moreover, small and medium size Clouds could be able to share their own resources dynamically in a peer-to-peer manner without geo-location limits. Based on these considerations, which are only a subset of the benefits, we believe there is a significant opportunity to contribute to research by developing decentralized approaches in "Optimal Service Provider Selection" in "Cloud Federation".

REFERENCES

[1] P. M. Mell and T. Grance, "SP 800-145. The NIST definition of cloud computing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-145, 2011.

[2] E. C. Service. (2014). *Eu Cloud Service Level Agreement Standardisation Guidelines*. [Online]. Available: https://ec.europa.eu/digital-single-market/en/news/cloud-service-level-agreement-standardisation-guidelines/

[3] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, L. Fong, S. Masoud Sadjadi, and M. Parashar, "Cloud federation in a layered service model," *J. Comput. Syst. Sci.*, vol. 78, no. 5, pp. 1330–1344, Sep. 2012.

[4] A. Panarello, A. Celesti, M. Fazio, M. Villari, and A. Puliafito, "A requirements analysis for IaaS cloud federation," in *Proc. 4th Int. Conf. Cloud Comput. Services Sci.* Setúbal, Portugal: Scitepress, 2014, pp. 1–6.

[5] A. Panarello, A. Celesti, M. Fazio, A. Puliafito, and M. Villari, "Costs of a federated and hybrid cloud environment aimed at MapReduce video transcoding," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 258–263.

[6] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected cloud computing environments," *ACM Comput. Surv.*, vol. 47, no. 1, pp. 1–47, Jul. 2014.

[7] N. M. Calcavecchia, A. Celesti, and E. D. Nitto, "Understanding decentralized and dynamic brokerage in federated cloud environments," in *Achieving Federated Self-Manageable Cloud Infrastructures*. Hershey, PA, USA: IGI Global, 2012, pp. 36–56.

[8] D. Petcu, "Multi-cloud: Expectations and current approaches," in *Proc. Int. Workshop Multi-cloud Appl. Federated Clouds (MultiCloud)*, 2013, pp. 1–6.

[9] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Mu, and G. Tofetti, "Reservoir–When one cloud is not enough," *Computer*, vol. 44, no. 3, pp. 44–51, Mar. 2011.

[10] Optimis. (2013). *Optimis–Optimized Infrastructure Service*. [Online]. Available: http://www.optimis-project.eu/

[11] A. J. Ferrer, F. HernáNdez, J. Tordsson, and E. Elmroth, "OPTIMIS: A holistic approach to cloud service provisioning," *Future Gener. Comput. Syst.*, vol. 28, no. 1, pp. 66–77, Jan. 2012.

[12] Stratuslab. (2012). *Stratuslab, Darn Simple Cloud*. [Online]. Available: http://www.stratuslab.eu/

[13] 4CaaSt. (2013). *The 4Caast Project*. [Online]. Available: http://www.4caast.eu/

[14] Bonfire. (2013). *Bonfire Project*. [Online]. Available: http://www.bonfire-project.eu/

[15] A. C. Hume, Y. Al-Hazmi, B. Belter, K. Campowsky, L. M. Carril, G. Carrozzo, V. Engen, D. García-Pérez, J. J. Ponsatí, R. Kűbert, Y. Liang, C. Rohr, and G. V. Seghbroeck, "BonFIRE: A multi-cloud test facility for Internet of services experimentation," in *Testbeds and Research Infrastructure. Development of Networks and Communities* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering). Springer, 2012, pp. 81–96.

[16] D. García-Pérez, J. A. L. del Castillo, Y. Al-Hazmi, J. Martrat, K. Kavoussanakis, A. C. Hume, C. V. López, G. Landi, T. Wauters, M. Gienger, and D. Margery, "Cloud and network facilities federation in bonFIRE," in *Proc. Eur. Conf. Parallel Process. (Euro-Par)* Springer, 2014, pp. 126–135.

[17] CONTRAIL. (2014). *Contrail–Cloud Federation Computing Project*. [Online]. Available: http://contrail-project.eu/

[18] R. G. Cascella, L. Blasi, Y. Jegou, M. Coppola, and C. Morin, "Contrail: Distributed application deployment under SLA in federated heterogeneous clouds," in *The Future Internet*. Springer, 2013, pp. 91–103.

[19] VISION. (2013). *Vision Cloud Project, Funded by the European Commission Seventh Framework Programme (fp7/2006-2013) Under Grant Agreement n. 257019*. [Online]. Available: http://www.visioncloud.eu/

[20] D. Bruneo, F. Longo, D. Hadas, and E. K. Kolodner, "Analytical investigation of availability in a vision cloud storage cluster," *Scalable Comput., Pract. Exper.*, vol. 14, no. 4, pp. 279–290, Jan. 2014.

[21] E. K. Kolodner, S. Tal, D. Kyriazis, and D. Naor, "A cloud environment for data-intensive storage services," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2011, pp. 357–366.

[22] Mosaic. (2014). *Mosaic–Multi-Modal Situation Assessment and Analytics Platform*. [Online]. Available: http://www.mosaic-fp7.eu/

[23] S. Yangui, I.-J. Marshall, J.-P. Laisne, and S. Tata, "CompatibleOne: The open source cloud broker," *J. Grid Comput.*, vol. 12, no. 1, pp. 93–109, Nov. 2013.

[24] ECO2Clouds. (2014). *Eco2clouds–Experimental Awareness of CO_2 in Federated Cloud Sourcing Project*. [Online]. Available: http://www.eco2clouds.eu/

[25] C. Project. (2013). *Cloudwave Project, Funded by the European Commission Seventh Framework Programme (fp7/2006-2013) Under Grant Agreement n. 610802*. [Online]. Available: http://cloudwave-fp7.eu/

[26] A. Nus and D. Raz, "Migration plans with minimum overall migration time," in *Proc. IEEE Netw. Oper. Manage. Symp. (NOMS)*, May 2014, pp. 1–9.

[27] T. C. for Europe project. (2016). *The Cloud for Europe Project*. [Online]. Available: http://www.cloudforeurope.eu/

[28] SeaClouds. (Feb. 2016). *Seaclouds Projects*. [Online]. Available: http://www.seaclouds-project.eu/

[29] A. Brogi, M. Fazzolari, A. Ibrahim, J. Soldani, J. Carrasco, J. Cubo, F. Durán, E. Pimentel, E. Di Nitto, and F. D. Andria, "Adaptive management of applications across multiple clouds: The SeaClouds Approach," *CLEI Electron. J.*, vol. 18, no. 1, p. 2, 2015. [Online]. Available: http://www.scielo.edu.uy/pdf/cleiej/v18n1/v18n1a02.pdf

[30] BEACON. (2015). *The Beacon–Enabling Federated Cloud Networking Project*. [Online]. Available: https://cordis.europa.eu/project/rcn/194143/factsheet/en

[31] R. Moreno-Vozmediano, E. Huedo, I. M. Llorente, R. S. Montero, P. Massonet, M. Villari, G. Merlino, A. Celesti, A. Levin, L. Schour, C. Vázquez, J. Melis, S. Spahr, and D. Whigham, "BEACON: A cloud network federation framework," in *Communications in Computer and Information Science*, vol. 567. Springer, 2016, pp. 325–337, doi: 10.1007/978-3-319-33313-7_25.

[32] SUNFISH. (2017). *Sunfish–Secure Information Sharing in Federated Heterogeneous Private Clouds Project*. [Online]. Available: https://cordis.europa.eu/project/rcn/194230/factsheet/en

[33] NEPHELE. (2018). *Nephele–End to End Scalable and Dynamically Reconfigurable Optical Architecture for Application-Aware SDN Cloud Datacenters Project*. [Online]. Available: https://cordis.europa.eu/project/rcn/194293/factsheet/en

[34] ENTICE. (2018). *Entice–Decentralized Repositories for Transparent and Efficient Virtual Machine Operations Project*. [Online]. Available: https://cordis.europa.eu/project/rcn/194162/factsheet/en

[35] SUPERCLOUD. (2018). *Supercloud–User-Centric Management of Security and Dependability in Clouds of Clouds Project*. [Online]. Available: https://cordis.europa.eu/project/rcn/194123/factsheet/en

[36] FIESTA. (2018). *Fiesta–Federated Interoperable Semantic IoT/Cloud Testbeds and Applications Project*. [Online]. Available: https://cordis.europa.eu/project/rcn/194117/factsheet/en

[37] S. Sundareswaran, A. Squicciarini, and D. Lin, "A brokerage-based approach for cloud service selection," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Jun. 2012, pp. 558–565.

[38] B. K. Rani, B. P. Rani, and A. V. Babu, "Cloud computing and interclouds–Types, topologies and research issues," *Procedia Comput. Sci.*, vol. 50, pp. 24–29, 2015, doi: 10.1016/j.procs.2015.04.006.

[39] F. Lahmar and H. Mezni, "Multicloud service composition: A survey of current approaches and issues," *J. Softw., Evol. Process*, vol. 30, no. 10, p. e1947, Oct. 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.1947

[40] A. Vakili and N. J. Navimipour, "Comprehensive and systematic review of the service composition mechanisms in the cloud environments," *J. Netw. Comput. Appl.*, vol. 81, pp. 24–36, Mar. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S108480451730005X

[41] S. Asghari and N. J. Navimipour, "Service composition mechanisms in the multi-cloud environments: A survey," *Int. J. New Comput. Architectures Their Appl.*, vol. 6, no. 2, pp. 40–48, 2016, doi: 10.17781/P002033.

[42] M. R. M. Assis and L. F. Bittencourt, "A survey on cloud federation architectures: Identifying functional and non-functional properties," *J. Netw. Comput. Appl.*, vol. 72, pp. 51–71, Sep. 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804516301436

[43] L. Sun, H. Dong, F. K. Hussain, O. K. Hussain, and E. Chang, "Cloud service selection: State-of-the-art and future research directions," *J. Netw. Comput. Appl.*, vol. 45, pp. 134–150, Oct. 2014.

[44] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Algorithms and Architectures for Parallel Processing*. Springer, 2010, pp. 13–31.

[45] D. Petcu, "Portability and interoperability between clouds: Challenges and case study," in *Towards a Service-Based Internet*. Springer, 2011, pp. 62–74.

[46] T. Subramanian and N. Savarimuthu, "A study on optimized resource provisioning in federated cloud," in *Proc. Int. Conf. Comput., Cybern. Intell. Inf. Syst. (CCIIS)*, Mar. 2015.

[47] A. Singh, D. Juneja, and M. Malhotra, "A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 29, no. 1, pp. 19–28, Jan. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1319157815000841

[48] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 14–27, Jan. 2015.

[49] M. Gahlawat and P. Sharma, "VM selection framework for market based federated cloud environment," in *Proc. Int. Conf. Comput., Commun. Autom.*, May 2015, pp. 695–698.

[50] J. B. Abdo, J. Demerjian, H. Chaouchi, K. Barbar, and G. Pujolle, "Broker-based cross-cloud federation manager," in *Proc. IEEE 3rd Int. Conf. Inf. Sci. Technol. (ICIST)*, Dec. 2013, pp. 244–251.

[51] E. Badidi, "A context broker federation for QoC-driven selection of cloud-based context services," in *Proc. 9th Int. Conf. Internet Technol. Secured Trans. (ICITST-)*, Dec. 2014, pp. 185–190.

[52] M. Caballer, I. Blanquer, G. Moltó, and C. de Alfonso, "Dynamic management of virtual infrastructures," *J. Grid Comput.*, vol. 13, no. 1, pp. 53–70, Apr. 2014.

[53] R. G. Aryal and J. Altmann, "Fairness in revenue sharing for stable cloud federations," in *Economics of Grids, Clouds, Systems, and Services*, C. Pham, J. Altmann, and J. A. Bañares, Eds. Cham, Switzerland: Springer International Publishing, 2017, pp. 219–232.

[54] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *Proc. IEEE 3rd Int. Conf. Cloud Comput.*, Jul. 2010, pp. 337–345.

[55] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "Three-phase cross-cloud federation model: The cloud SSO authentication," in *Proc. 2nd Int. Conf. Adv. Future Internet*, Jul. 2010, pp. 94–101.

[56] R. G. Aryal and J. Altmann, "Dynamic application deployment in federations of clouds and edge resources using a multiobjective optimization AI algorithm," in *Proc. 3rd Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Apr. 2018, pp. 147–154.

[57] M. S. Q. Z. Nine, M. A. K. Azad, S. Abdullah, and N. Ahmed, "Dynamic load sharing to maximize resource utilization within cloud federation," in *Cloud Computing and Big Data*. Springer, 2015, pp. 125–137.

[58] L. Barreto, J. Fraga, and F. Siqueira, "Conceptual model of brokering and authentication in cloud federations," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 303–308.

[59] M. Villari, G. Tricomi, A. Celesti, and M. Fazio, "Orchestration for the deployment of distributed applications with geographical constraints in cloud federation," in *Cloud Infrastructures, Services, and IoT Systems for Smart Cities*, A. Longo, M. Zappatore, M. Villari, O. Rana, D. Bruneo, R. Ranjan, M. Fazio, and P. Massonet, Eds. Cham, Switzerland: Springer, 2018, pp. 177–187.

[60] G. Tricomi, A. Panarello, G. Merlino, F. Longo, D. Bruneo, and A. Puliafito, "Orchestrated multi-cloud application deployment in OpenStack with TOSCA," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2017, pp. 1–6.

[61] C.-Y. Liu, K.-C. Huang, Y.-H. Lee, and K.-C. Lai, "Efficient resource allocation mechanism for federated clouds," *Int. J. Grid High Perform. Comput.*, vol. 7, no. 4, pp. 74–87, Oct. 2015.

[62] F. Jrad, J. Tao, and A. Streit, "A broker-based framework for multi-cloud workflows," in *Proc. Int. Workshop Multi-Cloud Appl. Federated Clouds (MultiCloud)*, 2013, pp. 61–68.

[63] F. Jrad, J. Tao, A. Streit, R. Knapper, and C. Flath, "A utility-based approach for customised cloud service selection," *Int. J. Comput. Sci. Eng.*, vol. 10, nos. 1–2, p. 32, 2015.

[64] L. D. Ngan and R. Kanagasabai, "OWL-S based semantic cloud service broker," in *Proc. IEEE 19th Int. Conf. Web Services*, Jun. 2012, pp. 560–567.

[65] Y. Mansouri, A. N. Toosi, and R. Buyya, "Brokering algorithms for optimizing the availability and cost of cloud storage services," in *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2013, pp. 581–589.

[66] C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione, "Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2348–2362, Aug. 2016.

[67] T. G. Papaioannou, N. Bonvin, and K. Aberer, "Scalia: An adaptive scheme for efficient multi-cloud storage," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2012, pp. 1–10.

[68] W. Yao and L. Lu, "A selection algorithm of service providers for optimized data placement in multi-cloud storage environment," in *Communications in Computer and Information Science*. Springer, 2015, pp. 81–92.

[69] R. N. Calheiros, A. N. Toosi, C. Vecchiola, and R. Buyya, "A coordinator for scaling elastic applications across multiple clouds," *Future Gener. Comput. Syst.*, vol. 28, no. 8, pp. 1350–1362, Oct. 2012.

[70] S. K. Garg, S. Versteeg, and R. Buyya, "SMICloud: A framework for comparing and ranking cloud services," in *Proc. 4th IEEE Int. Conf. Utility Cloud Comput.*, Dec. 2011, pp. 210–218.

[71] T. Subramanian and N. Savarimuthu, "Application based brokering algorithm for optimal resource provisioning in multiple heterogeneous clouds," *Vietnam J. Comput. Sci.*, vol. 3, no. 1, pp. 57–70, Dec. 2015.

[72] J. Altmann, B. Al-Athwari, E. Carlini, M. Coppola, P. Dazzi, A. J. Ferrer, N. Haile, Y.-W. Jung, J. Marshall, E. Pages, E. Psomakelis, G. Z. Santoso, K. Tserpes, and J. Violos, "BASMATI: An architecture for managing cloud and edge resources for mobile users," in *Economics of Grids, Clouds, Systems, and Services*, C. Pham, J. Altmann, and J. A. Bañares, Eds. Cham, Switzerland: Springer, 2017, pp. 56–66.

[73] G. Z. Santoso, Y.-W. Jung, S.-W. Seok, E. Carlini, P. Dazzi, J. Altmann, J. Violos, and J. Marshall, "Dynamic resource selection in cloud service broker," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, Jul. 2017, pp. 233–235.

[74] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimal virtual machine placement across multiple cloud providers," in *Proc. IEEE Asia–Pacific Services Comput. Conf. (APSCC)*, Dec. 2009, pp. 103–110.

[75] J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Scheduling strategies for optimal service deployment across multiple clouds," *Future Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1431–1441, Aug. 2013.

[76] H. Kurdi, A. Al-Anazi, C. Campbell, and A. Al Faries, "A combinatorial optimization algorithm for multiple cloud service composition," *Comput. Electr. Eng.*, vol. 42, pp. 107–113, Feb. 2015.

[77] A. Kertesz, G. Kecskemeti, M. Oriol, P. Kotcauer, S. Acs, M. Rodríguez, O. Mercè, A. C. Marosi, J. Marco, and X. Franch, "Enhancing federated cloud management with an integrated service monitoring approach," *J. Grid Comput.*, vol. 11, no. 4, pp. 699–720, Jun. 2013.

[78] E. Carlini, M. Coppola, P. Dazzi, M. Mordacchini, and A. Passarella, "Self-optimising decentralised service placement in heterogeneous cloud federation," in *Proc. IEEE 10th Int. Conf. Self-Adapt. Self-Organizing Syst. (SASO)*, Sep. 2016, pp. 110–119.

[79] M. Giacobbe, A. Celesti, M. Fazio, M. Villari, and A. Puliafito, "An approach to reduce carbon dioxide emissions through virtual machine migrations in a sustainable cloud federation," in *Proc. Sustain. Internet ICT Sustainability (SustainIT)*, Apr. 2015, pp. 1–4.

[80] K.-H. Yeh, "An efficient resource allocation framework for cloud federations," *Inf. Technol. Control*, vol. 44, no. 1, Mar. 2015.

[81] S. Rebai, M. Hadji, and D. Zeghlache, "Improving profit through cloud federation," in *Proc. 12th Annu. IEEE Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2015, pp. 732–739.

[82] M. Hadji, B. Aupetit, and D. Zeghlache, "Cost-efficient algorithms for critical resource allocation in cloud federations," in *Proc. 5th IEEE Int. Conf. Cloud Netw. (Cloudnet)*, Oct. 2016, pp. 1–6.

[83] M. Hadji and D. Zeghlache, "Mathematical programming approach for revenue maximization in cloud federations," *IEEE Trans. Cloud Comput.*, vol. 5, no. 1, pp. 99–111, Jan. 2017.

[84] M. Giacobbe, M. Scarpa, R. Di Pietro, and A. Puliafito, "An energy-aware brokering algorithm to improve sustainability in community cloud," in *Proc. 6th Int. Conf. Smart Cities Green ICT Syst.*, 2017, pp. 1–8.

[85] J. Tordsson, R. S. Montero, R. Moreno-Vozmediano, and I. M. Llorente, "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers," *Future Gener. Comput. Syst.*, vol. 28, no. 2, pp. 358–367, Feb. 2012.

[86] Z. U. Rehman, F. K. Hussain, and O. K. Hussain, "Towards multi-criteria cloud service selection," in *Proc. 5th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jun. 2011, pp. 44–48.

[87] Q. Duan and A. V. Vasilakos, "Federated selection of network and cloud services for high-performance software-defined cloud computing," *Int. J. High Perform. Comput. Netw.*, vol. 9, no. 4, p. 316, 2016.

[88] J. Carvalho, D. Vieira, and F. Trinta, "Dynamic selecting approach for multi-cloud providers," in *Cloud Computing–CLOUD*, M. Luo and L.-J. Zhang, Eds. Cham, Switzerland: Springer, 2018, pp. 37–51.

[89] J. O. de Carvalho, F. Trinta, and D. Vieira, "PacificClouds: A flexible microservices based architecture for interoperability in multi-cloud environments," in *Proc. 8th Int. Conf. Cloud Comput. Services Sci.* Setúbal, Portugal: Scitepress, 2018, pp. 1–8.

[90] T. Kumrai, K. Ota, M. Dong, J. Kishigami, and D. K. Sung, "Multiobjective optimization in cloud brokering systems for connected Internet of Things," *IEEE Internet Things J.*, vol. 4, no. 2, pp. 404–413, Apr. 2017.

[91] X. Wang, J. Cao, and Y. Xiang, "Dynamic cloud service selection using an adaptive learning mechanism in multi-cloud computing," *J. Syst. Softw.*, vol. 100, pp. 195–210, Feb. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121214002398

[92] C. Negru, F. Pop, O. C. Marcu, M. Mocanu, and V. Cristea, "Budget constrained selection of cloud storage services for advanced processing in datacenters," in *Proc. 14th RoEduNet Int. Conf., Netw. Edu. Res. (RoEduNet NER)*, Sep. 2015, pp. 158–162.

[93] Y. Kajiura, S. Ueno, A. Kanai, S. Tanimoto, and H. Sato, "An approach to selecting cloud services for data storage in heterogneous-multicloud environment with high availability and confidentiality," in *Proc. IEEE 12th Int. Symp. Auto. Decentralized Syst.*, Mar. 2015, pp. 205–210.

[94] C. Redl, I. Breskovic, I. Brandic, and S. Dustdar, "Automatic SLA matching and provider selection in grid and cloud computing markets," in *Proc. ACM/IEEE 13th Int. Conf. Grid Comput.*, Sep. 2012, pp. 85–94.

[95] F. DAndria, S. Bocconi, J. G. Cruz, J. Ahtes, and D. Zeginis, "Cloud4SOA: multi-cloud application management across PaaS offerings," in *Proc. 14th Int. Symp. Symbolic Numeric Algorithms Sci. Comput.*, Sep. 2012. pp. 407–414.

[96] P. Massonet, J. Luna, A. Pannetrat, and R. Trapero, "Idea: Optimising multi-cloud deployments with security controls as constraints," in *Engineering Secure Software and Systems* (Lecture Notes in Computer Science). Springer, 2015, pp. 102–110.

[97] W.-J. Fan, S.-L. Yang, H. Perros, and J. Pei, "A multi-dimensional trust-aware cloud service selection mechanism based on evidential reasoning approach," *Int. J. Autom. Comput.*, vol. 12, no. 2, pp. 208–219, Dec. 2014.

[98] M. B. Karimi, A. Isazadeh, and A. M. Rahmani, "QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm," *J. Supercomput.*, vol. 73, no. 4, pp. 1387–1415, Apr. 2017, doi: 10.1007/s11227-016-1814-8.

[99] D. Lin, A. C. Squicciarini, V. N. Dondapati, and S. Sundareswaran, "A cloud brokerage architecture for efficient cloud service selection," *IEEE Trans. Services Comput.*, vol. 12, no. 1, pp. 144–157, Jan. 2019.

[100] T. Baker, M. Asim, H. Tawfik, B. Aldawsari, and R. Buyya, "An energy-aware service composition algorithm for multiple cloud-based IoT applications," *J. Netw. Comput. Appl.*, vol. 89, pp. 96–108, Jul. 2017, doi: 10.1016/j.jnca.2017.03.008.

[101] A. Elhabbash, Y. Elkhatib, G. Blair, Y. Lin, and A. Barker, "A framework for SLO-driven cloud specification and brokerage," in *Proc. 19th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2019, pp. 666–667.

[102] S. De Capitani di Vimercati, S. Foresti, G. Livraga, V. Piuri, and P. Samarati, "A fuzzy-based brokering service for cloud plan selection," *IEEE Syst. J.*, vol. 13, no. 4, pp. 4101–4109, Dec. 2019.

[103] H. Mezni and M. Sellami, "Multi-cloud service composition using formal concept analysis," *J. Syst. Softw.*, vol. 134, pp. 138–152, Dec. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0164121217301760

[104] S. Farokhi, S. Farokhi, F. Jrad, I. Brandic, and A. Streit, "Hs4mc-hierarchical SLA-based service selection for multi-cloud environments," in *Proc. 4th Int. Conf. Cloud Comput. Services Sci.*, vol. 1. Setúbal, Portugal: SciTePress, 2014, pp. 722–734.

GIUSEPPE TRICOMI (Member, IEEE) was born in Messina, Italy, in 1984. He received the bachelor's and master's degrees from the Università degli Studi di Messina, Messina, where he is currently pursuing the international Ph.D. degree in cyber-physical systems. His current research interests include cyber-physical systems, smart environment (smart city, smart building, and so on), cloud continuum computing (this include cloud and fog and edge computing), and cooperative pattern to be applied to these technologies. He worked as a Software Developer and an Architect in National and European projects that were related (from the former to latter) on Cloud Computing (SIMONE), Risk Assessment Through Cloud Computing Federation (SIGMA), and on Cloud Federation (BEACON). He has been employed to the Italian Ministry of Defence since 2017.

GIOVANNI MERLINO (Member, IEEE) received the international Ph.D. degree in computer and telecommunications engineering from the University of Catania. His research interests include cloud and edge computing, the IoT, network virtualization, smart sensing environments, and crowdsensing. In particular, most activities currently deal with monitoring and management of IoT, by adapting the Infrastructure-as-a-Service paradigm to infrastructure at the edge, with applications in the Smart City domain. Recently, he also focused on applying blockchains and smart contracts to access control, delegation and auditing for IoT/edge infrastructure and services, as well as deep learning in edge-powered surveillance, safety, and e-Health applications. He has coauthored over 60 papers in international journals and conferences, been involved in several FP7/H2020 EU projects, and participated in technical program committees of international conferences. He is co-leading the design of Stack4Things IoTronic, an OpenStack-based I/Ocloud framework. He is actively involved in a crowdfunded research initiative (#SmartME) to deploy and operate an experimental Smart City testbed in Messina and has co-founded an academic spin-off, SmartMe.IO S.r.l.

ALFONSO PANARELLO received the master's degree in computer engineering from the University of Catania, Italy, in 2014, and the Ph.D. degree, in June 2017. Since 2014, he has been one of the members of the Mobile and Distributed Systems Laboratory (MDSLAB), Messina. From January 2016 to July 2016, he was a Visiting Ph.D. Student with the University of Stuttgart, Germany, working on cloud federation orchestration and composition of distributed services. He is currently working as a Software Engineer with SmarteMe.IO S.r.l. His research interests include distributed systems, cloud computing, cloud federation, big data, high performance parallel processing, social media, and multimedia processing.

ANTONIO PULIAFITO (Member, IEEE) is currently a Full Professor of computer engineering with the University of Messina, Italy. He is the author and coauthor of more than 400 scientific articles. He has been acting as an Expert in ICT for the European Commission since 1998. Till September 2018, he acted as the President of the Center on Information Technologies, University of Messina. He has participated in several European projects, such as Reservoir, Vision, CloudWave, and Beacon. He has contributed in the development of several tools, such as WebSPN, ArgoPerformance, GS3, and Stack4Things. He is also a member of the management board of the National Center of Informatics in Italy (CINI) and the Director of the CINI Italian Laboratory on ''Smart Cities & Communities''. He is in charge of the #SmartME crowdfunding initiative, to develop a smart city infrastructure in the city of Messina. His research interests include distributed systems, networking, the IoT, and cloud computing.