Architecture and Business Logic Specification for Dynamic Cloud Federations

Ram Govinda Aryal[1], Jamie Marshall[2] and Jörn Altmann[1]

[1] Seoul National University, Seoul, Republic of Korea
aryal.rg@gmail.com, jorn.altmann@acm.org
[2] Amenesik SARL, France
ijm@amenesik.com

Abstract. Cloud federations have been seen as a possible solution for the volatility in the number of user requests and for the anti-competitive externalities of the economies of scale in the cloud service sector. In order for a federation to exist in the commercial market, an efficient mechanism for resource and revenue sharing is of paramount importance. In this paper, we design the architecture and specify the business logic for the dynamic operation of such federation platforms. The architecture and federation business logic specification include components, a federation SLA management framework, and revenue sharing mechanisms. It can also offer appropriate incentives to cloud providers for joining a federation. With such dynamism in the platform, cloud providers have the ability to automatically form and dissolve federations, to maintain resource compatibility, and to self-adapt to policies for managing contractual and economic relationships between federation members. This helps in streamlining the overall business process without being dependent on existing business relationships between service providers, between service providers of a federation, and between service providers and customers. This can encourage cloud providers to join in and be benefitted from the federation, thereby contributing to moving cloud computing to the next level.

Keywords: Dynamic Cloud Federation · Cloud Brokerage · Revenue Sharing · Cloud Interoperability · Cloud Federation Management · Shapley Value · Cloud Resource Sharing · Revenue Sharing · Federation Service Level Agreement.

1 Introduction

Although the effectiveness of the multitenancy model of cloud computing is proven [1], limitations exist with respect to inefficient resource utilization, restricted resource scaling, and discrimination by economies of scale. Cloud federation addresses many of these limitations by aggregating cloud resources [2][3][4][5]. Cloud federation can be considered as a voluntary arrangement among cloud providers, in which they agree to interconnect their infrastructure for sharing their resources among each other [2].

Besides marketplaces [6][7], cloud federation has been seen as a possible solution for the volatility in the number of user requests and for the anti-competitive externalities of the economies of scale in the cloud service sector [8][9]. Dynamic cloud federations allow small cloud providers to collaborate and gain economies of scale [10]. It also helps to ensure users' quality of service and to minimize costs [11]. By joining a federation, a cloud provider can also provide guaranteed availability of customer applications through reliable multi-site deployments [5].

Due to its promises, Cloud federation has been the area of research interest in recent years [1]. Despite these promises and ample research, it is important to state that there is no functional federation available in the commercial market. Extensive research has been done on optimizing the performance of federations and on dealing with challenges, such as resource sharing and interoperability [12][13][14]. Factors hindering providers to adopt cloud federation have also been investigated [2][15].

To form a federation, cloud providers need to perceive additional benefits and minimal risk in joining the federation. After a thorough review of the cloud federation literature [16][17][18][11], several factors were identified as important for incentivizing federations and coalitions. Revenue sharing issue has been acknowledged as one of the important factors. Revenue sharing governs how resources are shared to collectively generate revenue and how the collectively generated revenue is distributed.

Revenue sharing becomes more complicated with various innovative efforts, such as service composition for any application from a number of cloud providers and moving a virtual machine from one provider to another, in order to address the resource contention at a provider or to address dynamicity in an application footprint. As this phenomenon complicates the revenue sharing mechanism, it calls for tools that can dynamically manage contractual and economic relationships between members and provide a federation business logic for revenue allocation. Therefore, it can be stated that an effective and fair revenue sharing mechanism is required to encourage the formation of a cloud federation [39].

Previous research on architecture [3][4][19][20], resource allocation, and on revenue sharing [17][21][22][23][24] do not seem to analyze the problem from this perspective. This article deals with the architecture design and the business logic specification required for the formation and management of a dynamic cloud federation in the context of the BASMATI[1] cloud federation platform [25]. Dynamism in this context entails the ability to automatically form and dissolve federations, to maintain resource compatibility, to self-adapt to policies, and to achieve real-time situational data management. Our contribution includes (i) an architecture design of cloud federations that includes components and their interactions for SLA management and revenue sharing, and (ii) a specification of the federation business logic .

The paper is organized as follows. Related works are presented in section 2. Requirements for dynamic cloud federation are presented in section 3. In section 4, the general architecture for dynamic cloud federation management is given. Section 5 extends section 4 by detailing out the components for dynamic cloud federation management. Finally, the conclusion is presented in section 6.

[1] BASMATI – Cloud Brokerage Across Borders for Mobile Users and Applications

2 Related Works

Buyya et al. state three properties, which, they believe, are required at minimum to make the cloud federation effective [26]. It should (i) allow clouds in the federation to dynamically expand resources when needed; (ii) allow resource commercialization for providers with unused resources and providers in need to consume them; and (iii) deliver services with quality of service as specified in the SLA.

A number of studies deal with the architecture that supports the federation of cloud resources [3][4][19][20]. Ferrer et al. present challenges for reliable and scalable service platforms and architectures that support dynamic and flexible cloud service provisioning. They also developed a toolkit for cloud infrastructure and service providers that seek to optimize the cloud service life cycle [3]. Rochwerger et al. propose a cloud architecture that supports cloud federation and management of business services [4]. The proposed model facilitates a service-based economy, in which on-demand cloud services and resources are managed across clouds transparently. An architecture for a cloud broker, named CompatibleOne is proposed by Yangui et al. The architecture, which is based on open standard, aims at assisting end users of cloud services in choosing appropriate cloud providers for their applications by considering various factors and a large number of providers in the cloud service market [19]. The federation architecture of Carlini et al. supports horizontal and vertical integration of cloud platforms, regardless of technology. It aims to minimize the user burden on using cloud services that belong to different cloud providers and increase efficiency [20].

There is various research on resource allocation and revenue sharing in the context of cloud federations as well [22][17][23][21][24]. A participation-based method is proposed by Niyato et al. [22]. It uses a stochastic linear programming approach to a coalitional game for the formation of an optimal and stable coalition. The coalition is formed taking into account internal users demand and coalitional cost. Spot pricing, which is an auction-based method is proposed by Samaan et al. [17]. This method models cloud providers’ interactions as a repeated game played among a set of selfish providers, who aim at maximizing individual benefits. These providers interact with each other, to sell their unused resources in the spot market with individual profit maximization objectives. This method is applicable in non-cooperative settings, where smaller providers are discriminated due to economies of scale. Hassan et al. [23] proposes a varied form of the auction method, in which the auction is carried out with the aim of social welfare maximization rather than maximization of individual benefit. For the maximization of social welfare, a game model is proposed that looks for a set of cloud providers with low energy cost. As with other auction models, this has a negative effect on the fairness in revenue sharing. The method proposed by Hassan [21] includes a coalitional formation game that aims to maximize social benefits. It employs a hybrid method that combines participation-based methods and auction methods for revenue sharing. Provider resources are selected in such a way that the total cost is minimized. A broker fixes the revenue rate. It then receives a number of VM offers from cloud providers on that rate. Revenue rate is adjusted (increased or decreased) according to the actual participation of cloud providers and an optimal value is reached in a number of iterations. This method compromises individual freedom, and it is unfair as large providers can operate at low cost through economies of scale. How the offset revenue (profit) is distributed is not explained. A revenue sharing scheme in a cooperative setting is proposed by Mashayekhy et al. [24]. Their resource selection uses integer programming in a way that maximizes federation profit by minimizing cost. The revenue is allocated in proportion to their contribution, which is derived from the market share. Fairness may be an issue as market share is only considered for value estimation, discriminating new entrants, who may even contribute substantial resources but lack substantial market share.

3 Requirements for Dynamic Cloud Federation Management

3.1 Cloud Application Requirements

A customer’s cloud application and its requirements are introduced to the cloud management platform as TOSCA documents, describing technical characteristics, the topology of the required configuration, the service level objectives, and the constraints that are to be ensured and imposed.

3.2 Federation Business Logic Requirements

Federation business logic requirements, which are described in the federation business logic specification document, state service level agreements at the federation level and requirements for sharing revenue among federation members.

Federation Service Level Agreements Requirements. As a federation service level agreement (FSLA) is a derivation, or specialization, of the international standard known as WS-Agreement [27], it needs to describe a new service description element, which can be used to describe the technical and commercial details of a mono- or bi-directional relationship between two federation members [15]. The resulting agreement, when introduced into either, or both, of the partners, need to guide the actions of the component of the platform responsible. The actions should comprise the automated management of the technical and commercial aspects of the subsequent mutual interactions between the partners. These technical and commercial aspects include availability, price, placement, deployment, billing of resources, as well as a reference to the cost and revenue sharing mechanism.

Revenue Sharing Mechanisms Requirements. Revenue sharing is the distribution of profits and costs between stakeholders of a business or an organization. Although it is an existing concept, it has to be transformed and popularized in the context of platform-based content provisioning over the Internet [28]. Content can comprise, for example, applications, advertisements, music, and videos.

In the case of a commercial cloud federation, in which cloud service providers and application service providers work together in a cooperative manner for the collective provision of added value application service to their collective customers, cost and revenue sharing must be clearly defined.

Cost and revenue sharing mechanisms are important for cloud federations due to two factors. Firstly, cloud providers need an effective revenue sharing mechanism, which encourages them to participate in a federation. That means cloud providers will cooperate, if they receive a benefit [17][29][30].

Secondly, it determines how the allocation of revenue is performed. A fair system is needed, which ensures a proper compensation of all cloud providers for the number of resources that they invested in the federation [31]. For this study, fairness is defined as self-centered inequity aversion. This term relates to the behavior, at which "people resist inequitable outcomes; i.e., they are willing to give up some material payoff to move in the direction of more equitable outcomes" [18].

4 Architecture for Dynamic Cloud Federation Management

The cloud federation management proposed involves four components: the Cloud Management Platform, the Application Controller, the Federated Cloud Management, and the components handing the edge and cloud providers. The cloud management platform is the central component, interacting with the other components. The components and their primary relationships are shown in **Fig. 1**.

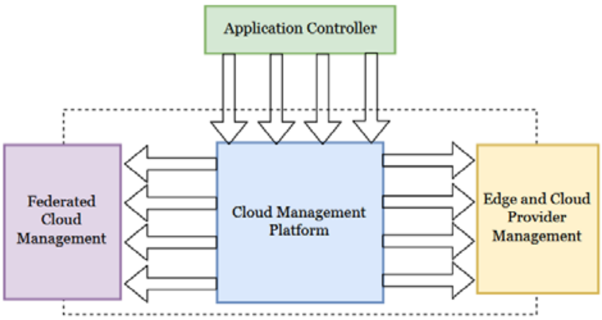

Fig. 1. Interaction of the cloud management platform with the federated cloud management component, using the federation business logic specification document, the application controller component and the edge and cloud provider management component).

Cloud Management Platform. The cloud management platform processes the cloud application requirements document (Section 3.1). It is also responsible for providing a deployment abstraction layer for the realization of resource deployment on existing public cloud providers and edge providers through edge and cloud provider management component, which is nowadays referred to as fog computing.

Application Controller. The application controller is responsible for the management and coordination of applications and their deployed and deployable application states. Following the requirements specified in the application description, the application controller uses a collection of application states, which allow resilient life cycle management of the application and its required resources.

Federated Cloud Management (using the Federation Business Logic Spefication document). This abstraction layer, when employed by multiple, individual commercial cloud service providers allows for automation of resource and revenue sharing between these providers. They are referred to as federation of cloud service providers, and each provider is referred to as a federation member. The core of the federated cloud management is the federation business logic specification document (Section 3.2 and Section 5).

Cloud and Edge Provider Management. It is responsible for the localization and exploitation of cloud and edge computing resources. This component encapsulates multiple private cloud interface technologies to be able to use specialized data centers for certain application-specific needs. This component allows interconnecting to the major commercial cloud platforms, namely Amazon Web Services (AWS EC2 and ECS), Microsoft Windows Azure, Google Compute (GCE and GKE), IBM Soft Layer, Cloud Sigma and other secondary cloud providers such as RackSpace, OVH, HP, DELL, to name but a few. These commercial vendors offer infrastructure as a service. Each provider publishes either a proprietary API or an adaptation of an Open API such as OpenStack, OpenNebula or Eucalyptus.

5 Specification of the Federated Cloud Management

The following figure (**Fig. 2**) shows the interactions of the federated cloud management component and its sub-components (i.e., the Cost and Revenue Sharing Mechanism component, the Cross-Cloud Interoperability component, the Federation SLA Manager component, and the Application Provider Accounting and Invoicing component) and the cloud management platform.

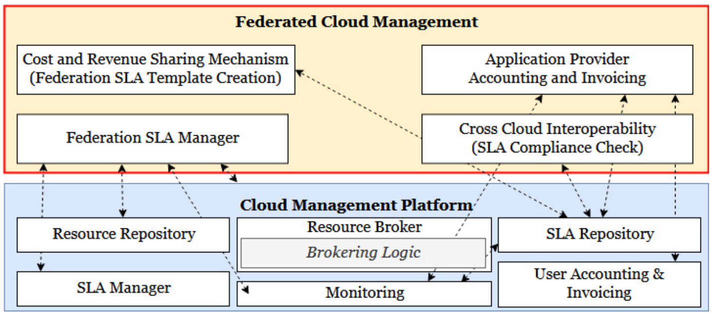

Fig. 2. Detailed specification of the federated cloud management, using the federation business logic specification document, and its interaction with the cloud management platform.

5.1 Federation SLA Manager

The federation-service level agreement (F-SLA), through which a cloud provider offers its resources within the federation, also describes the price of the corresponding offer of resources. Any service provider, which consumes resources made available through the federation, is required to make payments to the corresponding federation members providing these resources and presenting the relevant invoicing. The following figure (**Fig. 3**) depicts an example of how a F-SLA is used to control and manage relationships between the federation members, when they join a federation of cloud service providers.

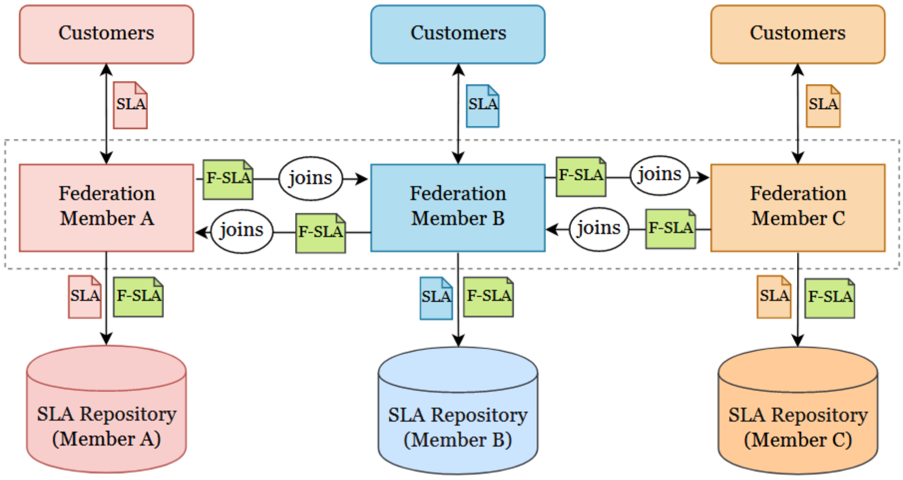

Fig. 3. Example of F-SLA based management of sharing between federation members.

Cloud providers (i.e., federation members) perform the service provisioning to customer applications based on a service level agreement (SLA) reached between the two parties. In a cloud federation scenario, theses SLAs may need to be served by federation members (e.g., Federation Member B or C of **Fig. 3**) other than the receiving one (e.g., Federation Member A of **Fig. 3**). In that case, the original SLA reached by the receiving cloud provider and the customer need still to be fulfilled. This requires for SLAs at a federation level (F-SLA) between the cooperating cloud providers, in addition to the SLA with the customers. Thus, as depicted in **Fig. 3**, the Federation SLA Manager of a federation member need to maintain two different types of SLAs in the SLA repository. The first group of SLAs is related to the applications of its own customers. The second group of SLAs is related to the customer applications of other federation members under the terms of the F-SLA.

The federation SLA manager has also to handle the construction and coordination of cloud service federation configurations, which are enabled through the FSLA. Sev-en of those configurations are:

(a) Simple Half-Duplex Configuration is the simplest configuration**,** where an application service provider is allowed access to the cloud capacity of a cloud service provider for the delivery of application services to its customers (**Fig. 4**).

Fig. 4. Simple half-duplex configuration.

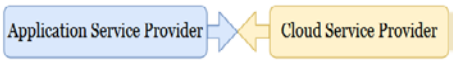

Fig. 5. Simple full-duplex configuration.

(b) Simple Full-Duplex Configuration is the logical extension to the simple half-duplex configuration. Federation members make use of their surplus service capacity available to each other through complementary federation SLAs (**Fig. 5**).

(c) Duplex Chain Configuration combines basic building blocks (i.e., simple half-duplex configuration and the simple full-duplex configuration) in a chain together, composing a linear federation configuration. In this configuration, each member of the federation is in relation with one or two other federation members (**Fig. 6**).

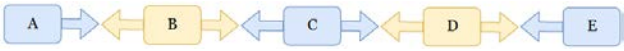

Fig. 6. Duplex chain configuration.

(d) Captive Configuration is a more complex but probably more realistic configuration. It can be envisaged if a federation member (called Federation Management Member) enters into individual federation agreements with application service providers and cloud providers (**Fig. 7**). In this case, the federation management member would be responsible for dispatching service requests to the individual federation members, A, B, C, D, and E.

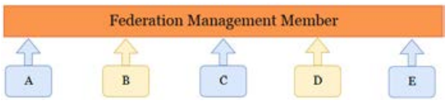

Fig. 7. Captive configuration.

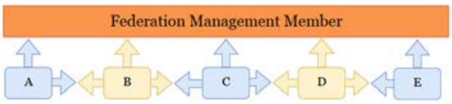

Fig. 8. Captive duplex chain configuration.

(e) Captive Duplex Chain Configuration is an extension to the preceding configurations. In this configuration, federation members are exposed to and managed through a central authority (i.e., the federation management member) for their introduction to the federation. The federation management member provides them with a federation resource catalog, which would allow them to establish point-to-point operations between members as and when required (**Fig. 8**).

(f) Multipoint Full-Duplex Configuration is a further enhancement to the duplex chain configuration. In this configuration, all federation members are effectively connected to all other federation members (**Fig. 9**). All members of the federation would enter into bi-lateral, full-duplex service level agreements with all other members. It should be noted that the total number of relationships and their accompanying agreements increase exponentially with the size of the federation.

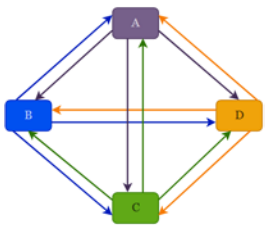

Fig. 9. Multipoint full-duplex.

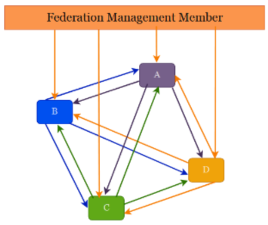

Fig. 10. Captive multipoint.

(g) Captive Multipoint is an extension to the *multipoint full-duplex* configuration and can be adapted to incorporate a central federation management member (**Fig. 10**). It allows for an efficient collection of specific data and an efficient management of the members.

5.2 Cost and Revenue Sharing Scheme

Depending on the cost and revenue sharing mechanism implemented, it has to be considered that federation members, who bring consumable resources to the federation, such as virtual machines, disk space, network bandwidth, IP addresses, application licenses, incur costs for the resources that they provide. Therefore, in any sharing mechanisms, it is normal to expect that the federation members, who provide these resources to the federation for use by other federation members, are reimbursed at least at cost value or, to some degree, with a financial gain.

Sharing Algorithm. There are several well-known mechanisms for cost and resource sharing in game theory models. However, each one of them provides different benefit, fairness, and stability values to the collaborations [29][30]. This may affect how the federations are created and, even, how they are dissolved. The sharing algorithm is described in the federation business logic specification document as part of the F-SLA. In the following, we introduce 3 different mechanisms of cost and revenue sharing: assigned resources mechanism, outsourcing mechanism, and Shapley value mechanism.

With the *assigned resources mechanism*, each cloud provider obtains a revenue share in proportion to the resources contributed (proportional revenue sharing mechanism) [31]. This mechanism is particularly strong in its fairness. This is a simple mechanism to implement, as it only considers the resource contributions of collaborating cloud providers for calculating the revenue share. Besides, it allows for combining resources that could not be sold separately [30].

The *oursourcing mechanism* has often been considered in connection with cloud federations, as they have been seen as a way for cloud providers to outsource some of their businesses to other cloud providers. Following this logic, collaborating cloud providers can implement a mechanism, by which the outsourcing provider will get a percentage of the revenue or a fixed fee. This revenue sharing allows a cloud provider to keep some of the revenue of the business it secured, even though it would not be able to fulfill it alone [30].

The *Shapley value mechanism* is named after Lloyd Shapley, who proposed a method to calculate the overall gain of all alternatives of a player that participates in a game with a large number of agents [18]. In cloud computing, the Shapley value is used to represent the marginal contributions of any cloud provider to the federation. In contrast with other mechanisms, this mechanism allows federations to allocate revenue according to the value created. In the simplest form, the value created by each provider can be calculated based on the resources that were made available for cloud service composition. Using this mechanism, other types of contributions (e.g., data center location, customer base) can also be considered as value additions to federations [30].

Calculation of Charges. For calculating the charges, a formula (i.e., a pricing scheme) is used that is expressed within the F-SLA, using WS-Agreement [27]. This formula is read by the invoicing and accounting component, to calculate the final charge and balance of payments. The invoicing and accounting component has to obtain the required input for the formula and calculate the corresponding charges based on the formula.

All actions performed by cloud providers and application service providers, for which an element of cost has been defined, result in a financial transaction being debited and credited to the accounts of the involved parties, for the amount described in the terms of the F-SLA or SLA. Invoice processing, often referred to as transaction collation, is performed on an account by account basis. It is performed by the accounting service of each platform operator. The resulting invoices are issued to the customers and the consumers of services, whether external or internal to a federation. All customers are liable for payment.

Due to the distributed and fully automated nature of the federation and the cloud abstraction technology provided, it can be envisaged that certain members of the federation could specialize in the management of accounting, invoicing, and cost and revenue sharing [32].

With respect to federation members, who provide application services to customers and their end users, they will invoice their customers for the services that they provide. This revenue stream is negotiated and decided between the customer and the application service provider and is clearly expressed in the terms of the SLA. For this, the charges can either be simply calculated based on the data collected for a specific customer or require the collection of accounting data from other members of a federation.

6 Validation

To validate the architecture and mechanisms proposed, we performed a simple simulation for observing the revenue distribution characteristics in federations.

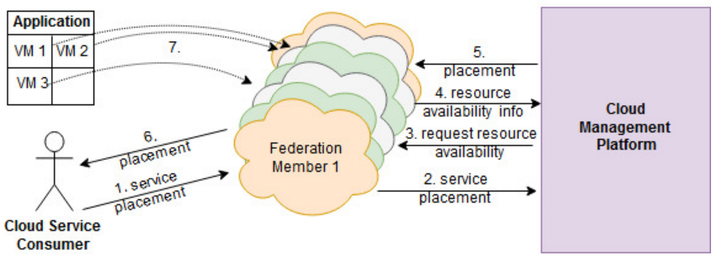

Fig. 11. Application deployment scenario for a captive configuration.

For this, we considered a scenario of application deployment in one of the aforementioned federation types (i.e., the captive configuration) and performed an analysis of revenue sharing as per one of the approaches mentioned, namely the Shapley value mechanism. **Fig. 11** shows the application deployment scenario for that configuration.

The application deployment scenario comprises 7 interactions: (1) The cloud service customer, who requires the deployment of its application, submits a service placement request to a cloud provider, who is a member of a cloud federation that (in our scenario) comprises of six providers of different capacities and characteristics. (2) The federation member forwards it to the cloud management platform. (3) The cloud management platform requests resource availability information from all federation members, (4) who respond within a certain time period, (5) and, based on these responses, calculates an optimal service placement plan by following an optimization technique, as the one described in [33]. The cloud management platform will take an account of the service provisioning, based on which the revenue shares are allocated to federation members using the Shapley Values mechanism (section 5.2). (6) The cloud service customer is informed about the placement plan. (7) The federation members, who are considered in the placement plan, are triggered to deploy the plan.

With this, we observe how the capacity utilization and earning per unit resource change for each federation members in comparison to the case when they worked individually. The results are shown in **Fig. 12**.

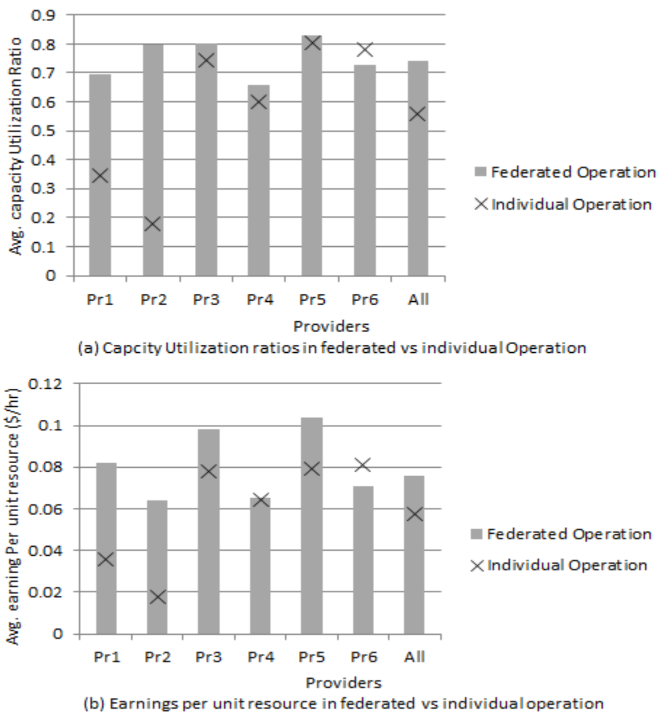

Fig. 12. Comparison of federated vs. individual operation.

The results of our simulation show that, for most of the providers (i.e., all providers except for provider Pr6), federation enables an increase of the providers' utilization ratios (**Fig. 12a**) and, hence, the earning per unit resource (**Fig. 12b**). Considering the sum of resource utilizations as well as the sum of earnings per unit resource, it is obvious that cloud federation improves the social welfare of the system of cloud providers in the market.

The results suggest that the federation operates properly as per the proposed architecture and the specified federation business logic. They also suggest that a proper federation business logic can increase the revenue stream for many federation members and can increase the social welfare of the system of cloud providers in a cloud computing market.

7 Conclusion

Federation platform operators require mechanisms for dynamic resource and revenue sharing, as they provide the motivation for cloud providers to participate in federations. Resource and revenue sharing mechanisms determine how cloud providers in a federation share their computational resources and, more importantly, the monetary benefits from the collaboration. Within this article, we presented the architecture and the federation business logic specification for such a mechanism. It allows for the formation of dynamic cloud federations.

The proposed architecture and the federation business logic specification follow the idea of an automated cloud federation management together with a cost and revenue sharing mechanism. In particular, the federation business logic describes the workings of the federation SLA management and the revenue sharing mechanism. A federation business logic enables offering incentives to cloud providers for joining federations and opens up opportunities for new sharing mechanisms.

With respect to the architecture proposed, cloud providers have the ability to automatically form and dissolve federations, to maintain resource compatibility, and to self-adapt to policies for managing contractual and economic relationships between members. Through this, a business process can dynamically be set up, independent of whether there are already business relationships between service providers, and between service providers and customers.

Overall, the proposed architecture and federation business logic specification enable cloud federations that can address specific, economic-related needs of cloud customers as well as their federation members.

Acknowledgements. The Institute of Engineering Research at Seoul National University provided research facilities for this work.

References

1. A. M. James Cuff, I. M. Llorente, and C. Hill, "Future challenges in federated cloud computing," 2017. [Online]. Available: https://rcc.harvard.edu/future-challenges-

federated-cloud-computing. [Accessed: 03-Apr-2018].
2. N. Haile and J. Altmann, "Risk-benefit-mediated impact of determinants on the adoption of cloud federation," in *PACIS* Proceedings, 17, 2015.
3. A. J. Ferrer, F. Hernández, J. Tordsson, E. Elmroth, A. Ali-Eldin, C. Zsigri, R. Sirvent, J. Guitart, R. M. Badia, K. Djemame, W. Ziegler, T. Dimitrakos, Srijith K. Nair, G. Kousiouris, K. Konstanteli, T. Varvarigou, B. Hudzia, A. Kipp, S. Wesner, M. Corrales, N. Forgó, T. Sharif, and C. Sheridan, "OPTIMIS: A holistic approach to cloud service provisioning," *Futur. Gener. Comput. Syst.*, vol. 28, no. 1, pp. 66–77, 2012.
4. B. Rochwerger, D. Breitgand, E. Levy, A. Galis, K. Nagin, I. M. Llorente, R. Montero, Y. Wolfsthal, E. Elmroth, J. Caceres, M. Ben-Yehuda, W. Emmerich, and F. Galan "The reservoir model and architecture for open federated cloud computing," *IBM J. Res. Dev.*, vol. 53, no. 4, pp. 1–4, 2009.
5. D. Petcu, "Consuming resources and services from multiple clouds," *J. Grid Comput.*, vol. 12, no. 2, pp. 321–345, 2014.
6. A. Bany Mohammed and J. Altmann, "A funding and governing model for achieving sustainable growth of computing e-infrastructures," *Ann. Telecommun. des télécommunications*, vol. 65, no. 11–12, pp. 739–756, 2010.
7. J. Altmann, C. Courcoubetis, and M. Risch, "A marketplace and its market mechanism for trading commoditized computing resources," *Ann. Telecommun. des télécommunications*, vol. 65, no. 11–12, pp. 653–667, 2010.
8. J. Altmann and M. M. Kashef, "Cost model based service placement in federated hybrid clouds," *Futur. Gener. Comput. Syst.*, vol. 41, pp. 79–90, 2014.
9. A. Bany Mohammed, J. Altmann, and J. Hwang, "Cloud computing value chains: understanding businesses and value creation in the cloud," in *Economic models and algorithms for distributed systems*, pp. 187–208, 2009.
10. K. Kim, S. Kang, and J. Altmann, "Cloud Goliath versus a federation of cloud Davids," in *Int. Conf. on Grid Eco. and Business Models*, 2014, pp. 55–66.
11. M. M. Hassan, M. S. Hossain, A. M. J. Sarkar, and E.-N. Huh, "Cooperative game-based distributed resource allocation in horizontal dynamic cloud federation platform," *Inf. Syst. Front.*, vol. 16, no. 4, pp. 523–542, 2014.
12. L. Heilig, R. Buyya, and S. Voß, "Location-aware brokering for consumers in multi-cloud computing environments," *J. Netw. Comput. Appl.*, vol. 95, pp. 79–93, 2017.
13. N. Haile and J. Altmann, "Evaluating investments in portability and interoperability between software service platforms," *Futur. Gener. Comput. Syst.*, vol. 78, pp. 224–241, 2018.
14. M. Risch and J. Altmann, "Capacity planning in economic grid markets," in *International Conference on Grid and Pervasive Computing*, 2009, pp. 1–12.
15. I. Breskovic, M. Maurer, V. C. Emeakaroha, I. Brandic, and J. Altmann, "Towards autonomic market management in cloud computing infrastructures," in *CLOSER*, pp. 24–34, 2011.
16. K. Jeferry, G. Kousiouris, D. Kyriazis, J. Altmann, A. Ciuffoletti, I. Maglogiannis, P. Nesi, B. Suzic, and Z. Zhao, "Challenges emerging from future cloud application scenarios," *Procedia Comput. Sci.*, vol. 68, pp. 227–237, 2015.
17. N. Samaan, "A novel economic sharing model in a federation of selfish cloud providers," *IEEE Trans. Parallel Dis. Sys.*, vol. 25, no. 1, pp. 12–21, 2014.

18. A. E. Roth, *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.
19. 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, 2014.
20. E. Carlini, M. Coppola, P. Dazzi, L. Ricci, and G. Righetti, “Cloud Federations in Contrail.,” in *Euro-Par Workshops (1)*, pp. 159–168, 2011.
21. M. M. Hassan, M. A. Al-Wadud, and G. Fortino, “A socially optimal resource and revenue sharing mechanism in cloud federations,” in *Computer Supported Cooperative Work in Design, IEEE 19th Intl. Conf.*, pp. 620–625, 2015.
22. D. Niyato, A. V Vasilakos, and Z. Kun, “Resource and revenue sharing with coalition formation of cloud providers: game theoretic approach,” in *Cluster, Cloud and Grid Computing, 11th IEEE/ACM Intl. Symp.*, pp. 215–224, 2011.
23. M. M. Hassan, M. Abdullah-Al-Wadud, A. Almogren, B. Song, and A. Alamri, “Energy-aware resource and revenue management in federated cloud: a game-theoretic approach,” *IEEE Syst. J.*, vol. 11, no. 2, pp. 951–961, 2017.
24. L. Mashayekhy, M. M. Nejad, and D. Grosu, “Cloud federations in the sky: formation game and mechanism,” *Trans Cloud Comput*, vol.3, no.1, pp.14–27, 2015.
25. J. Altmann, B. Al-Athwari, E. Carlini, M. Coppola, P. Dazzi, A. Juan Ferrer, N. Haile, Y.-W. Jung, J. Marshall, E. Pages, E. Psomakelis, G. Zulfa Santoso, K. Tserpes, and J. Violos, “BASMATI: An architecture for managing cloud and edge resources for mobile users,” in *Intl. Conf. on Economics of Grids, Clouds, Systems, and Services*, pp. 56–66, 2017.
26. R. Buyya, R. Ranjan, and R. Calheiros, “Intercloud: utility-oriented federation of cloud computing environments for scaling of application services,” *Algorithms Archit. Parallel Process.*, pp. 13–31, 2010.
27. M. Risch and J. Altmann, “Enabling open cloud markets through WS-agreement extensions,” in *Grids and Service-Oriented Architectures for Service Level Agreements*, Springer, pp. 105–117, 2010.
28. N. Haile and J. Altmann, “Structural analysis of value creation in software service platforms,” *Electron. Mark.*, vol. 26, no. 2, pp. 129–142, 2016.
29. R. G. Aryal and J. Altmann, “Fairness in revenue sharing for stable cloud federations,” in *Economics of Grids, Clouds, Systems, and Services, Intl. Conf. on*, pp. 219–232, 2017.
30. J. P. R. Coronado and J. Altmann, “Model for incentivizing cloud service federation,” in *Economics of Grids, Clouds, Systems, and Services, Intl. Conf. on*, pp. 233–246, 2017.
31. B. El Zant, I. Amigo, and M. Gagnaire, “Federation and revenue sharing in cloud computing environment,” in *Cloud Engineering, Intl. Conf. on*, pp. 446–451, 2014.
32. A. Caracas and J. Altmann, “A pricing information service for grid computing,” in *5th Workshop on Middleware for Grid Computing at the 8th Intl. Middleware Conf.*, 2007.
33. R. G. Aryal and J. Altmann, “Dynamic application deployment in federations of clouds and edge resources using a multiobjective optimization AI algorithm,” in *Fog and Mobile Edge Computing, Intl. Conf.*, pp. 147–154, 2018.