Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

User interface modeling technique in the cloud environment

Sang Young Lee *

Department of Health Administration, Namseoul University, Cheonan, South Korea

ARTICLE INFO

Article history:
Received 23 December 2021
Received in revised form
21 August 2022
Accepted 29 September 2022

Keywords:
Modeling process
User interface
Cloud computing

ABSTRACT

For designing a qualified user interface, there needs to be a graphic expert, requirement analyst, system designer, programmer, technical expertise, social activity scientist, and experts for each field. However, it is extremely difficult for these various experts to participate in such user interface design. This paper focused on design rules and modeling techniques of user interfaces that can support user availability. The visual cohesion of business events can be improved by modeling the prototype of the object-oriented user interface based on the object. The clustering method uses transaction objects and forms objects based on business event objects and task objects. We have studied it in detail so that a prototype of the user interface can be created. The significance and conclusions of this study are as follows. First, visual cohesion is improved by designing the object of functional, sequential, and communicative objects. Second, the object design rules of the user interface were created so that even an inexperienced designer could create a high-quality prototype. Third, it enhances the user's preference, ease, understanding, compliance rate, and quality of graphic layout by improving object-based visual cohesion. Fourth, functional, sequential, communicative, and procedural cohesion of business events is increased by the clustering of user interface objects. As a result, this paper is providing a modeling method of user interface technique in the cloud environment that could enhance the visual cohesion of user interface prototypes.

1. Introduction

Most of the services in the IT field are recently using cloud computing infrastructure (Pallis, 2010). In particular, services based on the fourth industrial revolution, such as artificial intelligence services, big data services, and edge computing services, have its root in the cloud (Zhang et al., 2010).

If anyone wants to be provided with service in a wide range of offered with high performance, availability, and stability, they have to be designed based on a cloud computing infrastructure. Along with the development of such a cloud environment, services through the web are expanding in various fields. As the use of the Internet becomes generalized and the cloud environment progresses, its application field has been expanding. Every kind of important document file related to personal information may be shared through the cloud or web-based environment system. Web applications generally provide services through interactions with operating systems and database systems. As the amount of information available on the Internet increases, the number of Internet users continues to increase. Accordingly, it is considered that the quality of web services has to be improved by analyzing information about users who use the web. The contents of the website are important, but the convenience of content users must be considered (Kim et al., 2020; Lim, 2017).

Analyzing and reconstructing a website that is already established is now becoming a major factor necessary for website management. Since a website is a user-centered medium, there is a need to understand and revise user needs constantly. SaaS (Software as a Service), which is the most widely used among cloud computing technologies, provides software services mainly to individuals and businesses online (Petcu, 2013; Ferry et al., 2013).

In other words, it is an "on-demand software service" that enables users to have application software that can be used in a cloud environment, which also can be accessed and used anywhere in the world with only a web browser. The second is PaaS (Platform as a Service), which provides an

* Corresponding Author.
Email Address: sylee@nsu.ac.kr
https://doi.org/10.21833/ijaas.2023.01.018
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-8291-9402

application program development environment. These days, BaaS (Blockchain as a Service) is providing blockchain as a service, and Azure of Microsoft is a representative example. IaaS (Infrastructure as a Service) offers various infrastructure services such as server resources, IP, Network, Storage, and power required for server operation. However, cloud providers provide different infrastructure services (Infrastructure-as-a-Service, hereinafter referred to as IaaS), and restrictions on the location, performance, and cost of resources, thus APIs are also different, making it difficult to be interlocked with various cloud infrastructures.

Global-scale service providers are paying a lot of attention to multi-cloud technology that can build and operate an optimal computing infrastructure environment for various public clouds to solve problems. Multi-cloud technology is a next-generation cloud-based technology for connecting, operating, utilizing, and managing two or more public or private clouds, which is a technology that integrates and operates multiple public cloud infrastructure (IaaS) services and enables flexible deployment, operation, and provision of cloud applications (PaaS, SaaS) on the multi-cloud infrastructure. With such multi-cloud technology, it is possible to flexibly utilize public clouds in various areas and overcome the regional limitations of a single cloud, thus it can provide services through the public cloud located closest to the user and improve the service delay problem.

From this point of view, the log file of the web server can be used as important data for understanding and analyzing the propensity of web users, and many studies have been conducted to analyze the log file of the web server.

However, it is difficult to involve such a wide range of experts in the design of the user interface. Thus, there is a need to conduct research on automatically designing functions for a user interface that can satisfy a wide range of expertise. The prototype for the user interface is to check the fundamental requirements by representing the basic direction of the user interface in the analysis stage to extract what is in the mind. Therefore, it is considered that an evaluation scale has to be developed to enhance the user's understanding and research on visual cohesion has to be conducted to design a high-quality user interface (Goonasekera et al., 2016; Han et al., 2020).

Visual cohesion for designing would help to improve the quality of the user interface as it provides a visible prototype to the designer or developer before the system implementation. In addition, visual cohesion can be a criterion for measuring the suitability between the user interface and semantic contents. The visual cohesion of events must be designed to improve the understanding of business work and enhance usability through clustering so that business events may have a semantically related relationship. Furthermore, professional developers prefer to visually aggregate user interfaces based on the scenarios created by various verifications so that they can use them easily (Kratzke and Quint, 2017).

This paper developed a scale of evaluation to enhance the user's understanding, and research a modeling technique to improve the visual cohesion for designing a high-quality user interface. The purpose of this paper is to study visual cohesion. User interfaces are created by clustering business events to be related to each other. The interface is to improve the understanding and usability of the business system. Therefore, in this study, a user interface prototype is studied. Here, we propose four types of objects that can improve visual cohesion. The object used applies the object-oriented design technique of clustering and the method of measuring visual consistency.

This paper's chapter 2 is about the design scale and the calculating method for visual cohesion, which is the standard for a user with a high-quality evaluation. And chapter 3 proposes an object type and an object clustering method for clustering business events in the user interface according to relevance. Final chapter 4 proposes the significance of this study and future tasks.

2. Analyzing the user interface

For analysis of the user interface, we can adapt to changes in the function and data of the service. It is required for coping with them and the type of parameter required for the cloud service and the method of access should vary according to the type of service. The following elements are necessary in order to enable open service access (Lassoued et al., 2020; Alhomoud et al., 2011).

- Model-based interface through service modeling: External software services have various types of access methods and data formats, and even though the targeted service to be accessed is changed, there is a need for an abstracted model suitable for using external services to connect to a consistent interface without system change. The model should have scalability and convenience according to the authentication method of external service and the type of service, and the system should be able to expand the service that can be connected only by replacing or adding the model (Lee, 2020).
- Standard specification of software service: The parameters required by each of the external services are different depending on the type. The standard service description method is needed to describe the parameters required for each service, such as the address of the service, the unit of parameters used in the service, the format of the parameter, the location where the parameter is placed, and the authentication method of the service.
- Parameter mapping for connecting between system and service: In order to use the data using an external service in the system to be integrated, it should be possible to perform data conversion

and mapping in a format that can be used by the integration target, not the data primarily acquired through the service.

Cloud services become the design criteria of the user interface. In other words, the technologies used include structural design. First, the interface structural design scale is considered in terms of configuration and layout. The user interface structure is the simplest to compute based on simplicity. This includes the number of basic visible business events on screens and dialog boxes, the amount and distribution of space in black and white, and the longest string that is deliverable. The structural design scale of the user interface lacks a clear theoretical principle related to availability and has a relatively low correlation with availability.

More important is that not all user interface designs are treated as very significant issues, and user interface designers have treated this as a really troublesome problem. Moreover, the content-related measurement includes the features of functions, manipulations, and meanings for users of interface business events. In addition, the task-related scale is based on the aspects of practical tasks and scenarios that can be executed through the user interface. In addition, the study is conducted to devise a scale suite applied for user interface design. The scale suite is a concept completely made of clear and transparent principles to devise design rules for good design. This scale consists of five scales:

- First, the efficiency of work is the basis for measuring the interface.
- Second, it is the measurement task transparency which measures whether it is designed to make it easy to access objects.
- Third, it is the layout uniformity that simplifies the complexity of the layout, measuring whether the graphical uniformity or regularity of the user interface layout is configured simply.
- Fourth, the user interface is a task of measuring whether the layout structure is appropriate for measuring the difficulty of use.
- Fifth, it is a visual cohesion that measures how much semantic relevance between business event concepts matches visual balance.

2.1. Criteria of visible cohesion

This scale refers to the visible cohesion of concepts that extended cohesion, which is already a well-established measure of software engineering complexity to the user interface. Visible cohesion measures a semantically or conceptually connected relationship between business events. The scale used utilizes the principle of combining semantically related elements into groups. Each group simplifies the structure by reducing interdependencies based on the understanding of meaning. Visible consistency leverages the consistency of events that appear in the user's programming unit. The overall structure is how units are arranged and grouped. Business events are grouped according to their user interface relevance. Here, the interface is utilized to make it easy to understand and use. User interfaces need the following definitions to apply cohesion to the overall design: Here, we propose an equation for calculating visual cohesion (VC).

$$VC = 100 \times \left[\frac{\delta_{\forall I} G_I}{\frac{\delta_{\forall I} N_I (N_I - 1)}{2}}\right] \left(G = \delta_{\forall, j, i} R_{i,j}\right)$$

The basic pattern compares the sum of consistency seen in forms and dialog boxes.

2.2. Measure of visible cohesion

To measure the visual cohesion of the user interface, we transformed the calculation formula proposed by Constantine into an object group as follows to calculate the visual cohesion.

$$VC = 100 \times \left[\frac{\delta_{\forall f} G_f + \delta_{\forall t} G_t + \delta_{\forall r} G_r}{\frac{\delta_{\forall f} N_f (N_f - 1)}{2} + \frac{\delta_{\forall t} N_t (N_t - 1)}{2} + \frac{\delta_{\forall r} N_r (N_r - 1)}{2}}\right]$$

This is because the cohesion per object unit must be calculated.

Subjective clustering is generally performed to define relevant groupings. However, this study takes an objective approach by clustering objects based on the availability of the user interface.

3. Development process and design model

The proposed design model is a modeling technique for improving visual cohesion by extracting object units and clustering them according to the relationship between user interfaces. Therefore, this study analyzed user interface details according to the similarity, relevance, and transition units of the user interface to cluster business events into an object suite. If object types can be clustered and visualized, it can increase the visibility of the user interface.

As shown in Fig. 1, the first step of business event object design is designed according to object modeling rules for the control type of the business event. The second step of the task object is designed by clustering business events into a task suite. In addition, the third step of the transaction object clusters business events into a transaction object suite. The fourth step of designing a form object is a modeling step that divides the form and designs how to divide the size of the form into task and transaction groups.

3.1. Business event objects

The user interface designs business event control types that make up the conversion data of the business event. Fig. 2 shows the field clustering graph.

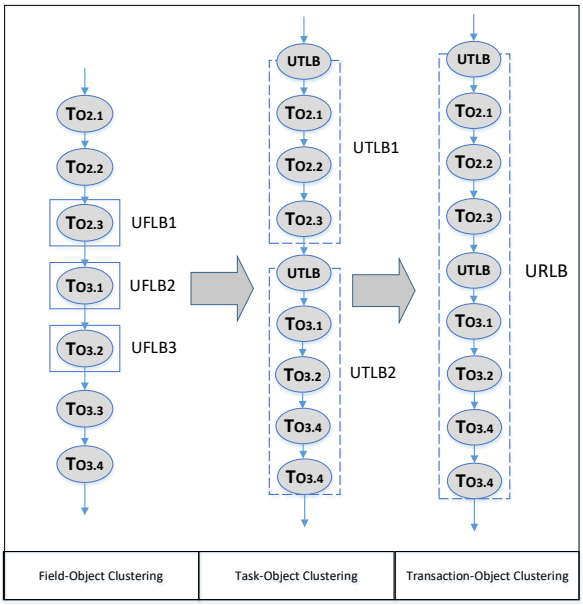
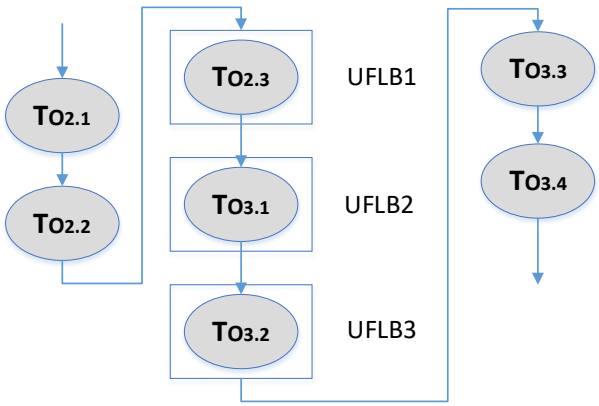


Fig. 1: Abstract clustering stage of business events for prototyping



Rule 1: UIFLB is a field that has a fixed instance.
Rule 2: The number of instances is not a UIFLB unless it is fixed so that the field can be entered.
Rule 3: A field that can contain up to 7 instances is a radio button-enabled item label (based on Miller's '7±2 chunks' theory).
Rule 4: A field that can contain at least eight instances is the item abstraction label that can use a drop-down field.

Fig. 2: Field clustering graph

Fig. 3 shows the object according to the design rule of the business event object. Auto-input is a design phase that analyzes input/output data of business event objects so that users can easily understand and use them. Fig. 3 helps you understand the business event object. The entity is the designed gender, product code, and product specification. In other words, gender is designed as a radio button. The product code and product specification are expressed as a business event object that can support users by designing as a combo box.

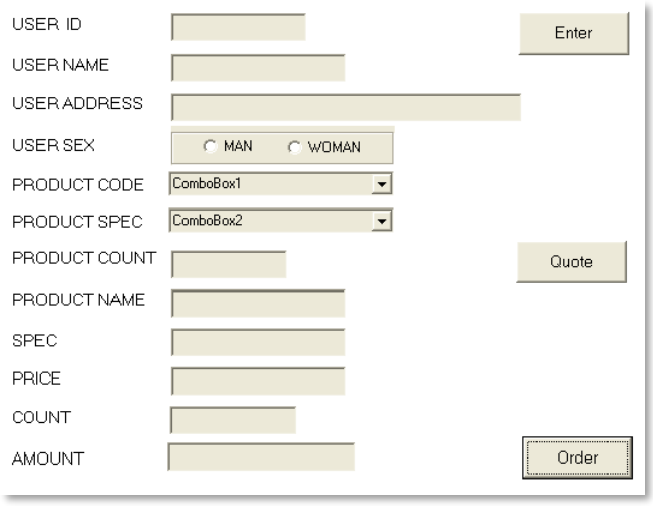


Fig. 3: Field abstract modeling

This effectively models the function of events in the user interface to increase the cohesion of events and increases the functional cohesion of the user interface and the reusability of instant data. In other words, it helps users understand input data and designs a control type that can reduce data input errors.

3.2. Task object

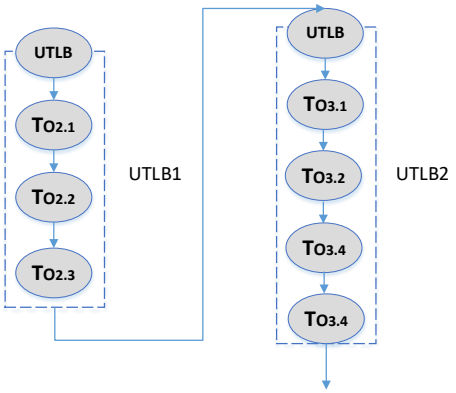
Task object of creating "User Interface Task Object." It represents a unit of task objects that clusters transition objects of business events that are simultaneously transferred to one task. This is a clustering step to enable users to classify task objects suites through block labeling of a task unit when there are two or more input or output business events that can be transferred when an event occurs.

This increases functional cohesion by allowing the user to understand the task units of business events. The rules for clustering the task object suite of business events that are transferred to one task are as follows. These task objects are clustering and classify the transition units of tasks to understand the flow of tasks. A business event is a family of task objects. Clustering specifies labels. User interfaces increase the communication and functional cohesion of business events. Fig. 4 shows a task clustering graph.

3.3. Transaction object

The design of a transaction object is the stage of creating a User Interface Transaction Object (hereinafter referred to as UIRO). In other words, UIRO organizes a collection of transaction objects into groups of input-control-output business events. A transaction object creates a user's request (input) and response (output) as a single family. A collection of transaction objects has rules for clustering.

Transaction object design is a design stage to provide a suite of transaction object transaction suite to users, which is a clustering method of transaction objects grouped into input task-control-output tasks. Fig. 5 shows transaction abstract modeling result.



Rule 1: Consecutive transition tasks (input field-button-output field) are UITLB.
Rule 2: Output UITLBs based on multiple records using StringGrid
Rule 3: The node that has a consecutive input link is the beginning of UITLB.
Rule 4: The node that has a consecutive output link is the end of UITLB.

Fig. 4: Task clustering graph

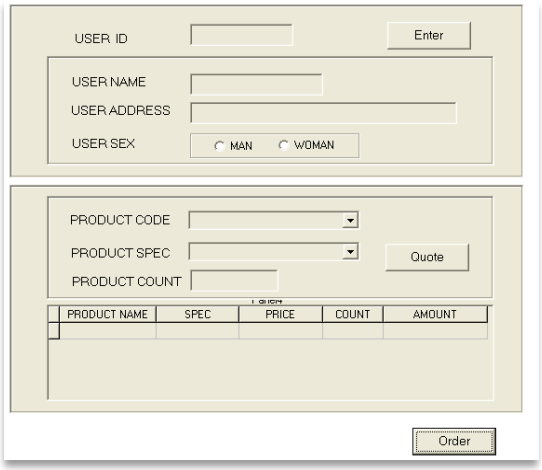
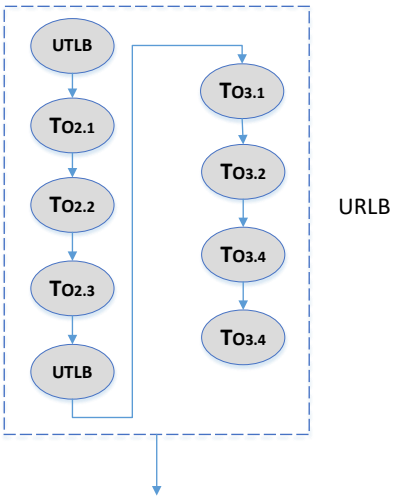


Fig. 5: Transaction abstract modeling result

This facilitates the user's understanding by visualizing the transaction object of business events in the user interface. Input task-control-output task is clustered and visualized in an object unit and the user can easily understand the transaction suite in the interface. Fig. 6 shows the transaction clustering graph.

The Fig. 7 shows the structure and design results of the transaction object in the user interface to help understand the transaction object.



Rule 1: The composition should be an input field, button, and output field.
Rule 2: One or more input and output fields are allowed by criteria.
Rule 3: The first node of consecutive input fields is the beginning of UIRLB.
Rule 4: The last link of consecutive output fields is the end of LIRLB.

Fig. 6: Transaction clustering graph

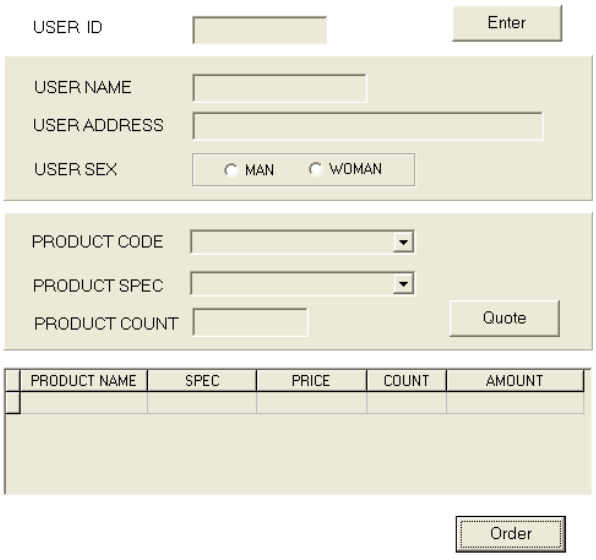


Fig. 7: Task abstract modeling result

3.4. Form objects

In the design stage of the Form Object, User Interface Form Object (hereinafter referred to as UIFO) is created. The user interface divides business events into forms and creates objects of the form. This should be configured as a separate form if the form size of the user interface is limited or the contents of the business task are different.

An object has dozens of input/output business events. The input-to-output form is selected or generated so that it can be clearly recognized by the user, such as an interrupt. The rules for clustering a suite of form objects are as follows. Namely, the efficient design of the form makes it easier to understand business tasks and reduces the complexity of software, thereby facilitating program development and maintenance work.

4. Conclusion

Cloud services require a general and flexible interface that can adapt to changes in service functions and data in order to cope with them as new services are constantly being progressed and improved, and cloud services vary on the required parameter types and access methods according to the type of service. Therefore, this study focused on the design rules and modeling techniques of user interfaces that can support users with availability.

The visual cohesion of business events can be improved by modeling the prototype of the object-oriented user interface based on the object. The clustering method has been studied in detail so that a prototype of the user interface can be created. The significance and conclusions of this study are as follows:

- First, visual cohesion is improved by designing the object of functional, sequential, and communicative objects.
- Second, the object design rules of the user interface were created so that even an inexperienced designer could create a high-quality prototype.
- Third, it enhances the user's preference, ease, understanding, compliance rate, and quality of graphic layout by improving object-based visual cohesion.
- Fourth, functional, sequential, communicative, and procedural cohesion of business events is increased by clustering user interface objects.
- Fifth, it is possible to reduce communication errors with users and reduce the number of repetitions of user interface prototyping by providing a user interface with improved visual cohesion through object modeling.
- Sixth, we proposed an object-oriented design method for the user interface that can improve the understanding of business work by visualizing object types and the availability of the user interface.

The aim of this paper is to present a modeling method for business event objects, task objects, and transaction objects. The visual cohesion of business events can be improved by modeling the prototype of the object-oriented user interface based on the object. A modeling method of user interface technique in the cloud environment that could enhance the visual cohesion of user interface prototypes.

Acknowledgment

Funding for this paper was provided by Namseoul University.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alhomoud A, Munir R, Disso JP, Awan I, and Al-Dhelaan A (2011). Performance evaluation study of intrusion detection systems. Procedia Computer Science, 5: 173-180. https://doi.org/10.1016/j.procs.2011.07.024

Ferry N, Rossini A, Chauvel F, Morin B, and Solberg A (2013). Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In the IEEE Sixth International Conference on Cloud Computing, IEEE, Santa Clara, USA: 887-894. https://doi.org/10.1109/CLOUD.2013.133

Goonasekera N, Lonie A, Taylor J, and Afgan E (2016). CloudBridge: A simple cross-cloud python library. In the XSEDE16 Conference on Diversity, Big Data, and Science at Scale, Association for Computing Machinery, Miami, USA: 1-8. https://doi.org/10.1145/2949550.2949648 **PMid:34423340 PMCid:PMC8375622**

Han J, Park S, and Kim J (2020). Dynamic overcloud: Realizing microservices-based IoT-cloud service composition over multiple clouds. Electronics, 9(6): 969. https://doi.org/10.3390/electronics9060969

Kim BS, Jung YW, Oh BT, Kim SY, Son S, Seo JH, and Kang DJ (2020). Multi-cloud technology introduction and research trends. Electronics and Telecommunications Trends, 35(3): 45-54. https://doi.org/10.22648/ETRI.2020.J.350305

Kratzke N and Quint PC (2017). Understanding cloud-native applications after 10 years of cloud computing-A systematic mapping study. Journal of Systems and Software, 126: 1-16. https://doi.org/10.1016/j.jss.2017.01.001

Lassoued Z, Alhendawi M, and Bashitialshaaer R (2020). An exploratory study of the obstacles for achieving quality in distance learning during the COVID-19 pandemic. Education Sciences, 10(9): 232. https://doi.org/10.3390/educsci10090232

Lee SY (2020). Cloud based blockchain technology for personal health. International Journal of Advanced Nursing Education and Research, 5(3): 14-20.

Lim YM (2017). The collaborative image editing tool based on the cloud computing. Journal of Korea Multimedia Society, 20(8): 1456-1463. https://doi.org/10.9717/kmms.2017.20.8.1456

Pallis G (2010). Cloud computing: The new frontier of internet computing. IEEE Internet Computing, 14(5): 70-73. https://doi.org/10.1109/MIC.2010.113

Petcu D (2013). Multi-cloud: Expectations and current approaches. In the international workshop on Multi-cloud applications and federated clouds, Association for Computing Machinery, Prague, Czech Republic: 1-6. https://doi.org/10.1145/2462326.2462328 **PMid:23316704 PMCid:PMC3606312**

Zhang Q, Cheng L, and Boutaba R (2010). Cloud computing: State-of-the-art and research challenges. Journal of Internet Services and Applications, 1(1): 7-18. https://doi.org/10.1007/s13174-010-0007-6