UDC 004.032

M. YANOVSKY[1], O. YANOVSKAYA[1], V. BUTENKO[1], M. DEVETZOGLOU[2,3]

[1] National Aerospace University named after N. Y. Zhukovsky "KhAI", Ukraine
[2] International Creativity Engineering Group, Greece
[3] MechanoChemica S.A.

DISTRIBUTED CLOUD SYSTEM: SECURITY AND KEYWORD INDEXING ISSUES WITH IMPLEMENTATION STRATEGY AND BUSINESS LIMITATIONS

The article focuses on several technical and business issues that are essential for the development of distributed cloud system related to security, keyword indexing and business implementation. The presented solution on security makes it possible to ensure resource integrity and validate node identity. The keyword indexing enables access to relevant resources located in a distributed cloud that is required to use any resource type. Finally, the concept of business development and the implementation of the project with limited expenses enables a project to start from zero, using the distributed cloud architecture as an extension or as a component of an open source software that could potentially be part of a private cloud within an enterprise's IT infrastructure.

***Keywords**: cloud architecture, peer-to-peer networks, cryptography, content indexing.*

Introduction

The distributed cloud technology is a new technology based on a peer-to-peer (P2P) network approach that aspires to change the way companies and organizations work with regards to creating, publishing and sharing resources [1]. While the P2P network contains a very large number of peers and provides virtually unlimited storage capacities, it brings challenges in the design and implementation of an efficient security mechanism and indexing-retrieval strategy, especially when scalability is required with respect to bandwidth consumption, storage requirements and load balancing.

The best practices for network security implementation in large-scale systems are based on Open PGP standard (RFC 4880) [2]. Several approaches [3, 4] related to the usage of PGP protocol were proposed for decentralized mobile and crowd-sourced P2P networks. The current task is to develop a basic PGP-based security concept that would be adapted to the distributed cloud architecture.

The desired features of the searching algorithms implemented in P2P systems include high-quality results, minimal routing state maintained per node, high routing efficiency, load balance, resilience to node failures and support of complex queries. In most cases, the first feature is measured by the relevance of results to the user query. The routing state and efficiency refer to the amount of neighbours each node maintains and the number of overlay hops per query, respectively. Different searching techniques over structured and unstructured P2P systems provide different trade-offs between these features. Searching in highly structured systems is based on defined neighbours' links, which guarantee finding an existing data. However, the strict structure may impose a high overhead for handling frequent node join-leave. On the other hand, the unstructured P2P systems are resilient to such nodes' behaviour, but searching in unstructured networks may not always provide the flooding scheme or its variation. A detailed overview of main searching techniques in unstructured P2P, such as iterative deepening, k-walker random walk, direct BFS, intelligent search, local and routing indices based search, adaptive probabilistic and dominating set based search is presented in [5]. The in-depth survey on main information retrieval techniques for highly structured P2P systems are presented in [6 – 8].

Thus, the main purpose of the article is to develop the basic strategies of security, keyword search and implementation within distributed cloud environment.

1. Basic security concept

The PKI infrastructure was created for security purposes of the system. The core idea of the security concept is based on the principles used for the Maidsafe project [3] that requires no servers or centralised control. The main types of identifiers are the node identifier (ID_node), the global resource identifier (domain name) and the unique resource identifier (ID_res). For each new node in the system, regardless of the role, two pairs of public and private keys are generated (Kpub1, Kpriv1, Kpub2, Kpriv2) as well as the unique node identifier ID_node. The first key pair is used for standard communication processes while the second one is used for node validation process in order to protect against identity theft. The node identifier is generated as follows:

$$ID\_node = Hash\,(Kpub1 + Sig_{Kpriv2}(Kpub1)), \quad (1)$$

where ID_node – the unique node identifier;

Kpub1 – public key 1;

Kpriv2 – private key 2;

Hash (Data) – hash function value of “Data”;

$\mathrm{Sig}_{\mathrm{Kpriv2}}$(Kpub1) – digital signature of Kpub1 using key Kpriv2.

The unique resource identifier is a hash function value of its content that is stored in an additional DNS (TXT) record:

$$\mathrm{ID\_res} = \mathrm{Hash}\ (\mathrm{Content}\ (\mathrm{res})), \qquad (2)$$

where ID_res – the resource identifier;

Hash (Data) – hash function value of “Data”;

Content (res) – content of resource “res”.

The verification of resource integrity and replica’s identity is performed by calculating a hash function of the obtained resource and then comparing it with the resource identifier (ID_res). In addition, the node identity validation process is carried out at the intermediate stage, before content downloading starts. This process includes several stages as shown in the scheme below (fig. 1). The node that provides the resource (replication node) sends the message to the client node containing its identifier (ID_node), public keys (Kpub1, Kpub2), digital signatures ($\mathrm{Sig}_{\mathrm{Kpriv2}}$(Kpub1), $\mathrm{Sig}_{\mathrm{Kpriv2}}$(Kpub2)) and data in order to prove an identity.

The first step of the identity validation procedure involves checking the public keys obtained from the replication node using a standard digital signature verification mechanism. If the public keys are correct, then it is necessary to check the equality of expression:

$$\mathrm{ID\_node} = \mathrm{Hash}\ (\mathrm{Kpub1} + \mathrm{Sig}_{\mathrm{Kpriv2}}(\mathrm{Kpub1})) \qquad (3)$$

where ID_node – the unique node identifier contained in the message;

Kpub1 –public key 1;

Hash (Data) – hash function value of “Data”;

$\mathrm{Sig}_{\mathrm{Kpriv2}}$(Kpub1) – digital signature of Kpub1 using key Kpriv2 as transmitted in the message.

If the expression is true, the validation is successful and the identity is considered to be confirmed. Otherwise, the validation fails because of transmission errors or identity theft. For data privacy protection purposes, replication nodes can store an encrypted resource content. In this case, the encryption-decryption scheme shown in fig. 2 is applicable. The scheme includes several stages. In the first stage, the resource owner encrypts the “data” content that is confidential information using a symmetric encryption algorithm (e.g., AES) with the private key (Kpriv_data). The unique resource identifier is defined as a hash function value of the encrypted content. Then, if according to the security policy the client is a trusted node, the request message is sent to the resource owner specifying its own unique node identifier (ID_node), public keys (Kpub1, Kpub2), digital signatures ($\mathrm{Sig}_{\mathrm{Kpriv2}}$(Kpub1), $\mathrm{Sig}_{\mathrm{Kpriv2}}$(Kpub2)) and requests for Kpriv_data private key.

Further to that, it is necessary to validate the node’s identity, send the encrypted data and provide the client with the Kpriv_data private key. The identity validation procedure is performed according to the scheme described above and shown in fig. 1.

If the validation is successful, then the resource owner encrypts the Kpriv_data private key using an asymmetric encryption algorithm (e.g., RSA) with the node’s Kpub1 public key and after that, the message is sent back in the format presented in stage 6 on fig. 2.

The client then validates the owner’s node identity. Once the owner’s identity is confirmed, the client node decrypts the obtained encrypted Kpriv_data key using the Kpriv2 private key.

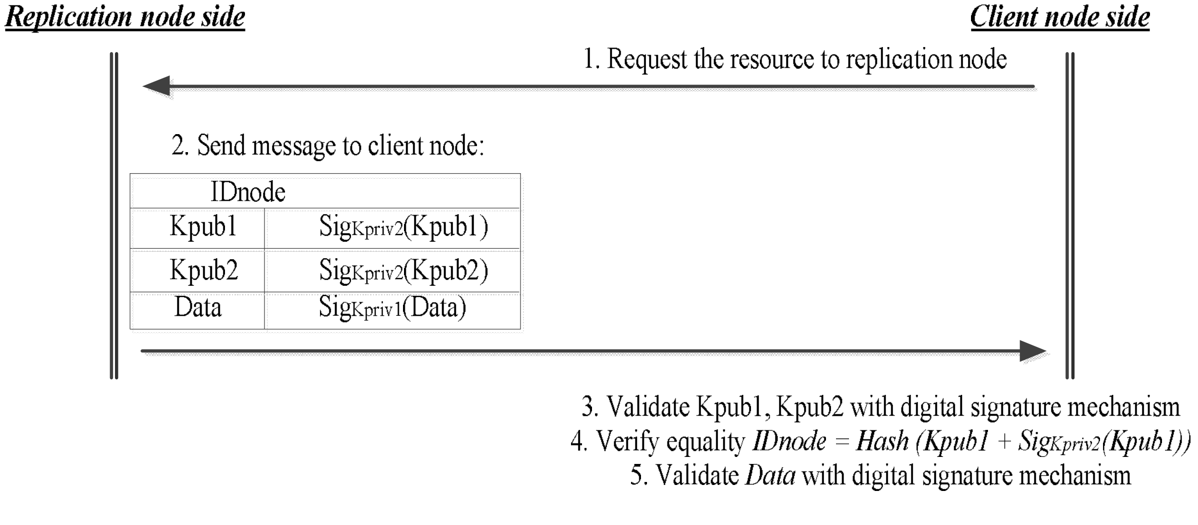
Replication node side
Client node side
1. Request the resource to replication node
2. Send message to client node:
IDnode
Kpub1
SigKpriv2(Kpub1)
Kpub2
SigKpriv2(Kpub2)
Data
SigKpriv1(Data)
3. Validate Kpub1, Kpub2 with digital signature mechanism
4. Verify equality IDnode = Hash (Kpub1 + SigKpriv2(Kpub1))
5. Validate Data with digital signature mechanism

Fig. 1. Validation scheme

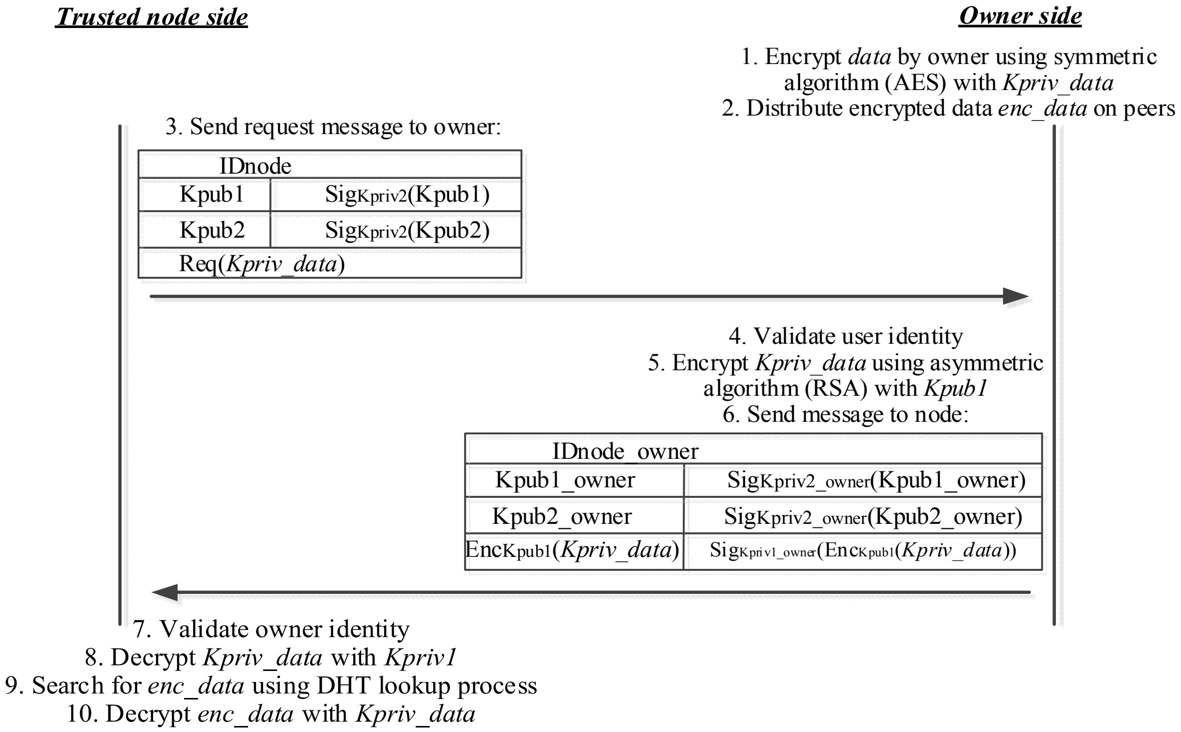
Trusted node side
Owner side
1. Encrypt data by owner using symmetric algorithm (AES) with Kpriv_data
2. Distribute encrypted data enc_data on peers
3. Send request message to owner:
IDnode
Kpub1 | SigKpriv2(Kpub1)
Kpub2 | SigKpriv2(Kpub2)
Req(Kpriv_data)
4. Validate user identity
5. Encrypt Kpriv_data using asymmetric algorithm (RSA) with Kpub1
6. Send message to node:
IDnode_owner
Kpub1_owner | SigKpriv2_owner(Kpub1_owner)
Kpub2_owner | SigKpriv2_owner(Kpub2_owner)
EncKpub1(Kpriv_data) | SigKpriv1_owner(EncKpub1(Kpriv_data))
7. Validate owner identity
8. Decrypt Kpriv_data with Kpriv1
9. Search for enc_data using DHT lookup process
10. Decrypt enc_data with Kpriv_data

Fig. 2. Encryption/decryption scheme

Then, the client software launches a standard DHT lookup process using the XOR metric to measure the distance between replication nodes, find the closest one and then downloads the encrypted data. The last stage includes the decryption process of the obtained data using the known Kpriv_data private key.

2. Keyword indexing-retrieval strategy

The general indexing-retrieval strategy is illustrated on Fig.3. As it was mentioned previously, the developed distributed architecture is based on the DHT (distributed hash table) idea.

Generally, a DHT is a hash table whose table entries are distributed among different peers, located in arbitrary locations. Each data item is hashed to a unique numeric key. Each node is also hashed to a unique ID in the same key space. Each node is responsible for a certain number of keys. This means that the responsible node stores the key and the data item with that key or a pointer to the data item with that key. Keys are mapped to their responsible nodes. The searching algorithms support two basic operations: *lookup(key)* and *put(key)*:

1. lookup(k) is used to find the location of the node that is responsible for the key k;

2. put(k) is used to store a data item (or a pointer to the data item) with the key k in the node responsible for k.

In a distributed storage application using a DHT, a node must publish the files that are originally stored on it before these files can be retrieved by other nodes. A file is published using put(k) [5]. The developed distributed cloud architecture treats web-content files as the main peer documents. Let us consider a structured P2P network with N peers P_i, $1 \leq i \leq N$, and a possibly very large document collection D, consisting of |D| documents d_j, $1 \leq i \leq |D|$. When a new peer connects to the network, it should pass the preparation stage and complete the local documents indexation. The full-text indexation is performed on the local documents that can be potentially visible through the network. During this stage the basic text mining techniques are applied to build the accurate set of terms that will correspond to the unique local document. Furthermore, the keys ($k_i^{(n)}$, where i – countable number of a key, n – peer identification number, $1 \leq i \leq m$, $1 \leq n \leq N$) are extracted by using hash-functions over the resulting document terms. As a result, the peer builds its personal Posting List (PL) which contains a set of keys, reference to the documents which contains them, frequency of the key in this document, frequency of keys through all documents and amount of queries that contained this key (key-query frequency). The created unique posting lists are further published in DHT, thus forming a part of the global index. However, initial indexing of large document collection can be too complicated for a unique peer, thus a mechanism of parallel index processing is required. Considering the developed architecture, each *peer generates the query* as follows. Using the distributed cloud client application, the word-based query is firstly transformed into a key using the hash-function and is then further transmitted through the connector to the DHT, which initiates the traditional lookup process. When the

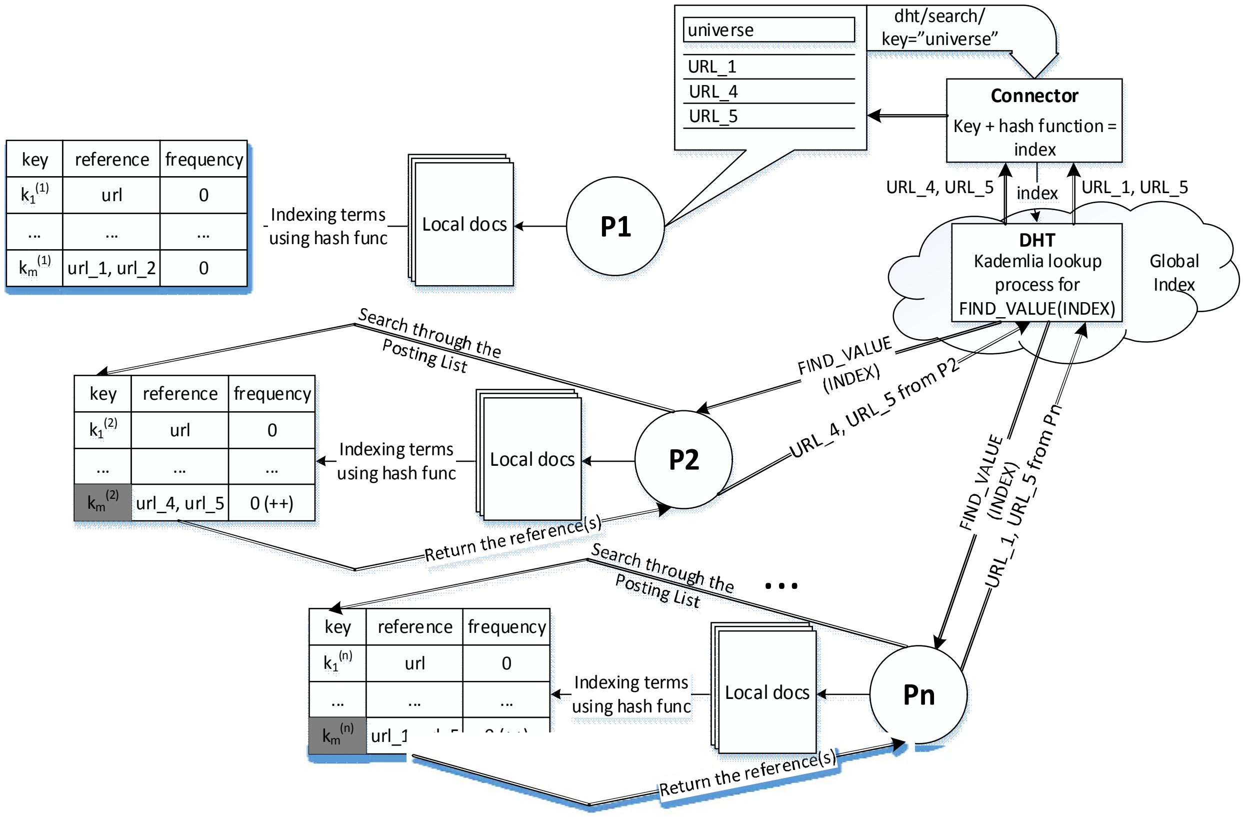

Fig. 3. Indexing-retrieval stategy

peer receive key-based query, it searches through the PL and, using relevance techniques, it returns the set of references to the DHT, which then initiates the P2P lookup process.

3. Implementation strategy and business limitations

In the case of the presented technology, the team approaches a Second-Generation Open Source model (OSSg2) [9], where OpenStack [10] is used as the base platform. The model is expected to be tailor-made in order to accommodate the team's specific needs and to better exploit the participatory value creation model.

The way for the presented open source project to grow is to use open source software as a platform on which to build software as a service (SaaS) offers. Building proprietary code and components on top of the original project and then selling the service leads to monetization for the SaaS product, rather than the open source product. Distributed cloud can be added upon OpenStack platform. It is then communicated to customers and users to test its efficiency and competitiveness. Based on the feedback received, the software components are modified to better suit needs. Furthermore, extra components may be built within distributed cloud, leading to the development of an enhanced product – SaaS. As a result, it is the combination of open source software (as a platform) with the development and addition of commercial components (software code) that creates value to the market as well as the developers. This approach leads to a hybrid model of open source and proprietary software, combing the best aspects of traditional and innovative business models.

Conclusion

Distributed Cloud Computing opens the way to new innovative models and strategies of security, content search and implementation. Within the presented strategies, each peer is responsible for proper maintenance of its local documents collection, i.e. correctly manage the PL with updates for new documents and changes for the already indexed. As the PL for large documents collection may contain too many terms, the key-query frequency can be used to truncate to the bounded number of their top-ranked elements. Further research is required to provide a more detailed analysis of this topic.

References (GOST 7.1:2006)

1. Service and Business Models with Implementation Analysis of Distributed Cloud Solution. [Text] / O. Yanovskaya, M. Devetzoglou, V. Kharchenko, M. Yanovsky // ICT in Education, Harmonization and

Knowledge Transfer ICTERI 2015. – 2015. – P. 446-461.

2. Callas, J. IETF RFC 4880: OpenPGP Message Format [Electronic resource] / J. Callas, L. Donnerhacke, H. Finney, et al. – Access mode: http://www.ietf.org/rfc/rfc4880.txt. – 10.03.2016.

3. Irvine, D. "Peer to Peer" Public Key Infrastructure [Electronic resource] / D. Irvine – Access mode: http://maidsafe.net/Whitepapers/pdf/PeerToPeerPublic KeyInfrastructure.pdf. – 10.03.2016.

4. Dewan, P. P2P Reputation Management Using Distributed Identities and Decentralized Recommendation Chains [Text] / P. Dewan, P. Dasgupta // IEEE Transactions on Knowledge and Data Engineering. – 2010. – V. 22, No. 7. – P. 1000 – 1013.

5. Li, X. Searching techniques in peer-to-peer networks [Text] / X. Li, J. Wu // Handbook of Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks. – 2006. – P. 875.

6. Lv, Q. Search and replication in unstructured peerto-peer networks [Text] / Q. Lv, P. Cao, E. Cohen, K. Li, S. Shenker // Proc. of the 16th ACM International Conference on Supercomputing (ACM ICS'02). – 2002. – P. 84 – 95.

7. Crespo, A. Routing indices for peer-to-peer systems [Text] / A. Crespo, H. Garcia-Molina // Proc. of the 22nd International Conference on Distributed Computing (IEEE ICDCS'02). – 2002. – P. 23 – 32.

8. Rhea, S. C. Probabilistic location and routing [Text] / S. C. Rhea, J. Kubiatowicz // Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02). – 2002. – V. 3. – P. 1248 – 1257.

9. The business of open source [Text] / R. T. Watson, M. C. Boudreau, P. T. York, et al. // Communications of the ACM. – 2008. – V. 51, No. 4. – P. 41 – 46.

10. OpenStack Official Website [Electronic resource] – Access mode: http://www.openstack.org/. – 10.03.2016.

References (BSI)

1. Yanovskaya, O., Devetzoglou, M., Kharchenko, V., Yanovsky, M. Service and Business Models with Implementation Analysis of Distributed Cloud Solution. *ICT in Education, Harmonization and Knowledge Transfer ICTERI 2015,* 2015, pp. 446- 461.

2. Callas, J. Donnerhacke L., Finney H., et al. *IETF RFC 4880: OpenPGP Message Format*. Available at: http://www.ietf.org/rfc/rfc4880.txt (accessed 10.03.2016).

3. Irvine, D. *"Peer to Peer" Public Key Infrastructure*. Available at: http://maidsafe.net/Whitepapers/ pdf/PeerToPeerPublicKeyInfrastructure.pdf (accessed 10.03.2016).

4. Dewan, P., Dasgupta, P. P2P Reputation Management Using Distributed Identities and Decentralized Recommendation Chains. *IEEE Transactions on Knowledge and Data Engineering,* 2010, vol. 22, no. 7, pp. 1000-1013.

5. Li, X., Wu, J. Searching techniques in peer-to-peer networks. *Handbook of Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks,* 2006, pp. 875.

6. Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S. Search and replication in unstructured peerto-peer networks. *Proc. of the 16th ACM International Conference on Supercomputing (ACM ICS'02)*, 2002, pp. 84 – 95.

7. Crespo, A., Garcia-Molina, H. Routing indices for peer-to-peer systems. *Proc. of the 22nd International Conference on Distributed Computing (IEEE ICDCS'02)*, 2002, pp. 23 – 32.

8. Rhea, S. C., Kubiatowicz, J. Probabilistic location and routing. *Proc. of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02),* 2002, vol. 3, pp. 1248 – 1257.

9. Watson, R. T., Boudreau, M. C., York, P. T., et al. The business of open source. *Communications of the ACM,* 2008, vol. 51, no. 4, pp. 41 – 46.

10. *OpenStack Official Website*. Available at: http://www.openstack.org/. (accessed 10.03.2016).

Поступила в редакцию 11.03.2016, рассмотрена на редколлегии 14.04.2016

РОЗПОДІЛЕНА ХМАРНА СИСТЕМА: КОНЦЕПЦІЯ БЕЗПЕКИ, ІНДЕКСАЦІЯ КОНТЕНТУ І СТРАТЕГІЯ ВПРОВАДЖЕННЯ В УМОВАХ ОБМЕЖЕННЯ ВИТРАТ

М. Е. Яновський, О. В. Яновська, В. О. Бутенко, М. А. Деветзоглу

У статті розглядається низка технічних і економічних питань, які відіграють важливу роль для розвитку розподілених хмарних систем і пов'язані з безпекою, індексацією контенту і реалізацією бізнес стратегії. Розроблено концепцію безпеки, що дозволяє забезпечити цілісність ресурсів й ідентифікацію вузлів. Запропоновано механізм індексації контенту для пошуку за ключовими словами ресурсів різного типу. На закінчення представлена стратегія впровадження проекту від початкової стадії в умовах обмеження витрат з використанням програмної реалізації розподіленої хмарної архітектури як доповнення або компонента платформи з відкритим вихідним кодом, що потенційно може стати частиною приватної хмари в рамках корпоративної IT-інфраструктури.

Ключові слова: хмарна архітектура, однорангові мережі, криптографія, індексація контенту.

РАСПРЕДЕЛЕННАЯ ОБЛАЧНАЯ СИСТЕМА: КОНЦЕПЦИЯ БЕЗОПАСНОСТИ, ИНДЕКСАЦИЯ КОНТЕНТА И СТРАТЕГИЯ ВНЕДРЕНИЯ В УСЛОВИЯХ ОГРАНИЧЕНИЯ ЗАТРАТ

М. Э. Яновский, О. В. Яновская, В. О. Бутенко, М. А. Деветзоглу

В статье рассматривается ряд технических и экономических вопросов, которые играют важную роль для развития распределенных облачных систем и связаны с безопасностью, индексацией контента и реализацией бизнес стратегии. Разработана концепция безопасности, позволяющая обеспечить целостность ресурсов и идентификацию узлов. Предложен механизм индексации контента для поиска по ключевым словам ресурсов различного типа. В заключение представлена стратегия внедрения проекта от начальной стадии в условиях ограничения затрат с использованием программной реализации распределенной облачной архитектуры в качестве дополнения или компонента платформы с открытым исходным кодом, что потенциально может стать частью приватного облака в рамках корпоративной ИТ-инфраструктуры.

Ключевые слова: облачная архитектура, одноранговые сети, криптография, индексация контента.

Яновский Максим Эдуардович – канд. техн. наук, доцент кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: M.Yanovsky@csn.khai.edu.

Яновская Ольга Владимировна – аспирант кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: O.Yanovskaya@csn.khai.edu.

Бутенко Валентина Олеговна – канд. техн. наук, старший преподаватель кафедры компьютерных систем и сетей Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ», Харьков, Украина, e-mail: V.Butenko@csn.khai.edu.

Деветзоглу Мария Анна – магистр по направлению материалы для энергетики и окружающей среды, магистр делового администрирования, International Creativity Engineering Group, MechanoChemica S.A., Афины, Греция, e-mail: M.Devetzoglou@interceg.com.

Maxim Yanovsky – PhD, Associate Professor at the Computer Systems and Networks Department, National Aerospace University named after N. Y. Zhukovsky "KhAI", Kharkiv, Ukraine, e-mail: M.Yanovsky@csn.khai.edu.

Olga Yanovskaya – MSc, PhD student in Information technologies, Computer Systems and Networks Department, National Aerospace University named after N. Y. Zhukovsky "KhAI", Kharkiv, Ukraine, e-mail: O.Yanovskaya@csn.khai.edu.

Valentina Butenko – PhD, Senior lecturer at the Computer Systems and Networks Department, National Aerospace University named after N. Y. Zhukovsky "KhAI", Kharkiv, Ukraine, e-mail: V.Butenko@csn.khai.edu.

Maria Anna Devetzoglou – MSc, MBA, International Creativity Engineering Group, MechanoChemica S.A., Athens, Greece, e-mail: M.Devetzoglou@interceg.com.