Electronic Service Architecture Model Assessment of Conformity to Cloud Computing Key Features

Peteris Stipravietis[1], Edzus Zeiris[2], Maris Ziema[3], [1-3]*Riga Technical University*

***Abstract* – The research examines electronic service execution possibilities in cloud computing environment and the key features of cloud computing. It also offers a method which allows quantitatively assess the conformity of existing e-service architecture model to cloud computing key features.The method allows evaluating the amount of necessary transformations and their efficiency.**

The offered solution is verified using the business process administered by Motor Insurance Bureau of Latvia – „The preparation of responsibility distribution" which originally was implemented using e-mail messages.

***Keywords* – e-services, cloud computing, architecture.**

I. Introduction

The World Wide Web has become one of the main channels of the exchange of information and services. Information and communication technologies are available to most members of society, and the demand for e-services grows steadily. Electronic services have become particularly demanded in public sector (government and municipal scope) where there is a wide range of services and many potential recipients. The electronization of public services contributes to the objective of e-government.

Until now, the providers of the services ran their processes using their own IT infrastructure, but now one must consider the trend that even more providers are running their processes or some part of them using the cloud capacity. The reason of this shift is that they are trying to minimize the costs of running their own data center or servicing staff. This trend also tells to consider how to migrate existing business processes to cloud efficiently, besides all other design time issues.

The first step in the solution of this task is to determine if migration is economically beneficial, i.e. will the future income generated by service outweigh the migration costs. One possible approach is to determine the degree of how much does the existing solution conform to cloud computing. This paper examines general cloud computing characteristics and key features of cloud solution which must be considered when transforming existing service process or designing new one. The objective of this paper is to define assessment method based on conformity of existing business process architecture to key features of cloud computing. The method can then be used to determine the efficiency of transformation.

II. Cloud Computing Characteristics

There are several approaches on how to define cloud computing environment characteristics. Looking from the infrastructure perspective, there are two types of cloud computing [1] – grid computing and transactional computing.

Grid computing is the simplest architectural model to migrate some process to. This architecture uses custom software which splits the data into small, isolated packets which can be processed autonomously. The best known example of this approach is the project SETI@Home [2], which deals in search for extraterrestrial civilizations. Every day the data from radio telescopes are divided into smaller units and sent to common workstations. The small capacity of each workstation sums up to enormous computing power. The same principle is used in other interstellar projects – MilkyWay@Home and Einstein@Home.

Transactional computing is more "traditional" computing approach, where data to be processed are dependent on other data within the system. The traditional solution commonly consists of relational data base and application server which serves the data to the service recipient using Web based interfaces. To achieve high availability clusters and load balancers are used.

The transactional computing model is the one to use for electronic services in the cloud environment. This model is already used in classical model. The grid computing model is not suitable because one cannot control the recipient of computable data, which can lead to leaking of a sensitive provider or recipient data to third parties.

Following the architectural types authors examine the types of provided services of cloud computing. Cloud computing environment consists of three layers – Software, Platform and Infrastructure – so called SPI model [3], [4]. This is the classical and most commonly used classification. Besides the SPI model the UCSB-IBM (University of California, Santa Barbara and IBM) ontology and Hoffmodel [3], [4] also are used. Table I shows SPI model and its partial correspondence to UCSB-IBM ontology.

UCSB-IBM approach defines five layers. In addition to first three layers which correspond to SPI layers, software kernel and middleware layers well firmware and hardware layer are added. Last layer (hardware) is also available to cloud user and is defined as Hardware as a Service – HaaS. This layer is commonly used by big enterprises – the HaaS approach relieves them of necessity to create and maintain their own data center.

The Hoff model is based on previous two models and is centered on the main three layers SaaS, PaaS and IaaS. These layers are divided further into sublayers [5]. The Hoff model differs from previous models with higher degree of detalization.

TABLE I

CLOUD COMPUTING CLASSIFICATION – SPI MODELAND UCSB-IBM ONTOLOGY

SPI view	Description	Example	UCSB-IBM
S – Software	Software used in Internet browser, alternative to desktop applications	Google Apps (Docs, SpreadSheet)	Software as a Service – SaaS
P – Platform	Platform (environment and API) to design, create and run applications	Google App Engine, Microsoft Azure	Platform as a Service – PaaS
I – Infrastructure	Infrastructure and hardware providing access to services of computing, storage, communications; possibility to run entire workstation or server	Privateclouds with virtualization, Amazon EC2, Eucalyptus	Cloudsoftware infrastructure: • Computing – IaaS • Data storage – DaaS • Communications – CaaS

SaaS layerpossesses the least extensibility (the software or service available to user is mostly fixed and non-customizable), the provider of the service has the greatest part of responsibility about the security. IaaS layer has it the other way round – the extensibility is limited only with hardware resources and the provider of the service must take care only of physical security of the data. PaaS layer is somewhere between those two regarding extensibility and security issues.

The authors focused on the first layer of SPI model and UCSB-IBM ontology– the software(SaaS) layer, because in most cases exactly this layer is used to host and supply e-services.

There exist four deployment models of the cloud [7]:

- Privatecloud. The infrastructure of private cloud is usually intended for use to one enterprise and its branches.
- Communitycloud. The infrastructure of community cloud is intended for use by a specific community of users or organizations that share some common goals or mission.
- Publiccloud. The infrastructure of public cloud is intended for general public use without specific restrictions.
- Hybridcloud. The hybrid cloud is a composition of two or more previously mentioned deployment models. Hybrid model is used, for example, to balance the load between distinct clouds.

There are no specific requirements regarding deployment when hosting and providing electronic services. It means that service provider may use either deployment model available.

After the examination of architectural types, classification and deployment models the general features of cloud computing solutions must be reviewed.The authors have identified five desirable key features of cloud computing solution [7], [8]:

- Self-service – in order to get the service and receive the result, the recipient of the service must be able to enter all the data needed without data operator assistance. This also means that the solution must provide sufficient computing power and data storage without the intervention of servicing personnel.
  In the context of electronic services this feature implies that the service user is responsible only of providing the necessary data. The recipient is not responsible of infrastructure and hardware issues.
- Pay-per-use – the accounting of the system resource usage must be short-term and precise; the recipient is able to release (and not pay for) resources as soon as they are not needed anymore.
  In the context of electronic services this feature is important where the billing between provider and recipient of the service is involved, although the business process of the specific service is not dependent of this feature.
- Elasticity – the cloud computing must create illusion that in every exact moment of time the user has access to unlimited computing power. The cloud computing solution must be able to provide extra resources when its load increases (scale-up) as well as to free them when the load decreases (scale-down).
  To ensure this feature the electronic service must be able to free resources as soon as they are not needed anymore, particularly when an exception has occurred, i.e., data base connections must be closed regardless of outcome.
- Customization – the resources of the cloud may be used by different users for different needs with different access levels.
  This feature applies to IaaS solutions, not so much to SaaS, which are commonly used to provide electronic services.
- Broad availability – the solution must be accessible to every user from every location and every device which supports standard data exchange protocols.
  In the context of electronic services this feature is interpreted as usability rather than availability. This feature is an ability of user to receive the service on different devices – smartphones, laptops, tablets, workstations and other devices, using both thin and fat client interfaces.

To continue the research the authors chose Self-service, Elasticity and Broad availability from these five features.

The customization was not chosen because it relates to the infrastructure of the cloud computing solution, not to the executable business processes behind electronic service. The pay-per-use feature in authors' view is not so important, at least when considering current data systems and services.

Considering that the elasticity is one of the most important cloud computing features, the security and performance issues must be addressed. The paper [9] shows multiple metric definitions which help to evaluate the performance of the cloud, while the research [10] introduces general cloud monitoring framework "GmonE". Regarding the data security in the cloud, multiple researches have been conducted, which deals in dynamic data protection [11] and general threat analysis [12].

III. The assessmentmethod

The next step towards the service migration to cloud computing environment is to determine the degree of conformity of the existing electronic service business process model to cloud computing key features.

Possible solution when transforming the electronic service is partial transformation when sensitive data and activities which need less computing power remains within enterprise data center. The activities which require more computing power and insensitive data are migrated to cloud. To ensure this approach the migration transformation is mapped which determines in which environment every activity and data packet will reside [13]. Though usable in practical solutions this method does not provide the answer if transformation is cost-effective and possible.

The method discussed in [14] is based on expert evaluation in different domains (Wide-Band Delphi Techniques) [15] and gives answer if it is worth to even begin migrate the processes of the entire enterpriseto cloud. This method is suited for defining and achieving strategic objectives, however, it is too general and complicated when assessment is needed for a single process only.

Other approaches, for example, 7-step migration [16] is based on iteration model and does not deal with preliminary assessment.

Seeing that aforementioned methods cannot be used when one must evaluate the efficiency of transformations, the authors propose new method that is intended for that very purpose. The assessment method is based on the conformity of the model of electronic service to key features of cloud computing.

To evaluate the conformity of an entire system or its part to a cloud computing feature, authors propose evaluation scale ranging from 0 to 4, as seen in Table II.

TABLE II

ASSESSMENT SCALE

Mark	Description
0	Total non-conformity, transformations are not possible. Such mark usually gets granted to parts of the system containing complicated business logic. To use that logic, user usually needs specific knowledge. Mainly this is unconformity to self-service feature.
1	At least 75% non-conformity, transformations are possible, but the costs may be too high. Migration at this level requires extensive transformations of business processes and user interfaces. Systems evaluated at this level are possible to improve and modify but one must compare these costs with the costs of design, development and deployment of a new solution. Such mark may be granted to parts of the system that are not available enough, for example, user interface supports only specific resolution or platform.
2	Largest part of the system does not conform, transformations are required. Medium to large transformations of the business processes are required, which may affect its logical flow. The control flow is affected almost always. The changes in user interfaces are required.
3	Largest part of the system conforms, transformations are required. Small transformations of business processes are required, which do not influence the logical flow of the process, but may affect the control flow. Changes in user interfaces may be necessary.
4	Total conformity, transformations are not required. Small changes in user interfaces may be necessary.

The total conformity of the model is calculated using following formula:

$$K(x) = \mathbf{0}\text{, if } SS(x) = \mathbf{0} \text{ or } EL(x) = \mathbf{0} \quad (1)$$

or

$$K(x) = \frac{kSS * SS(x) + kEL * EL(x) + kBA * BA(x)}{kSS + kEL + kBA},$$

$$kSS \in \mathbb{N}, kSS \geq 1,$$

$$kEL \in \mathbb{N}, kEL \geq 1,$$

$$kBA \in \mathbb{N}, kBA \geq \mathbf{1}, \quad (2)$$

$$\text{if } SS(x)! = \mathbf{0} \text{ and } EL(x)! = \mathbf{0} \text{ and } BA(x)! = \mathbf{0};$$

where

- $SS(x)$ = conformity of solution x to self-service feature;
- $EL(x)$ = conformity of solution x to elasticity feature;
- $BA(x)$ = conformity of solution x to broad availability feature;
- $K(x)$ = total conformity of solution x ;
- kSS, kEL, kBA = weights of features, initially equal to 1.

As seen from (1) total conformity will be evaluated as totally non-conforming even if system gets full marks in two features but is non-conforming to the third one.

The self-service for electronic service will be measured quite simply:

$$SS(x) = 4 * \left[\frac{F[simple]}{F[total]}\right] \quad (3)$$

where

- $F[simple]$ – the count of all the input fields during the life cycle of the service which can be filled by user without specific actions or knowledge;
- $F[total]$ – the total count of all the input fields during the life cycle of the service.
- If $F[total] = 0$, then $SS(x) = 4$.

The elasticity feature is the main one when the activity of the service must be measured.When assessing the conformity to elasticity, the design of the electronic service must be taken into account – total requests to data base, average resource requests, i.e., accessing file system, e-mail server, FTP server and others, during the life cycle of the service instance. The complexity of the service, measured in total lines of code and resource access, is also used.The ability of freeing up resources when they are not needed anymore could be taken into factor too.There exist other approaches [12], but they are based on financial indicators of the provider of the service when it is already hosted in cloud.The conformity to elasticity is measured as shown in (4).

$$EL(x) = 4 * \left[\frac{E[inst]}{(E[inst] + Err[inst])}\right] * \left[\frac{C}{1 + C}\right],$$

$$C = \frac{code}{resource} \tag{4}$$

$E[inst]$ – total count of electronic service instances during some period of time, always at least 1. $Err[inst]$ – the count of registered incidents and exceptions during the same period. $code$ – approximate count of lines of code, $resource$ – averagecount of resource requests in one instance, always at least 1.

As mentioned before, availability in the context of the electronic services is interpreted as usability rather as "real" availability (which must still be taken into account nonetheless). The formula to compute availability (5) is based on supported resolutions, platform-independent results and traditional availability and reliability measures. The "platform-independent results" are designation for binary data returned by service as the result (if ever present). For example, response as the Microsoft Word document would yield less than PDF document.

$$BA(x) = 4 * Res(x) * Doc(x) * R(x), \tag{5}$$

where

- $Res(x) = \frac{n}{m}$ – ratio between count of available resolutions n and count of required resolutions m;
- $Doc(x) = 0{,}75$, if response file format is not platform-independent;
- $Doc(x) = 1$, if response file format is platform-independent or response is not binary;
- $R(x) = \frac{E[successful]}{E[total]}$ is the measurement of reliability, taking into account count of successful instances of service and total count of instances of the service during some period of time.

All of data needed for calculations must be obtained from the environment hosting the electronic service, design documentations and other appropriate sources.

Weight values initially are set to "1". Raising value of any weight, the conformity to corresponding feature will be taken into total calculation with bigger impact. The choice of the weight values falls to the architect and provider of electronic service – which feature do they count as more important than others.

IV. The Verification of the Method

To check and verify the method the authors use the business process "The preparation of responsibility distribution", currently used in Motor Insurance Bureau of Latvia (MIB).

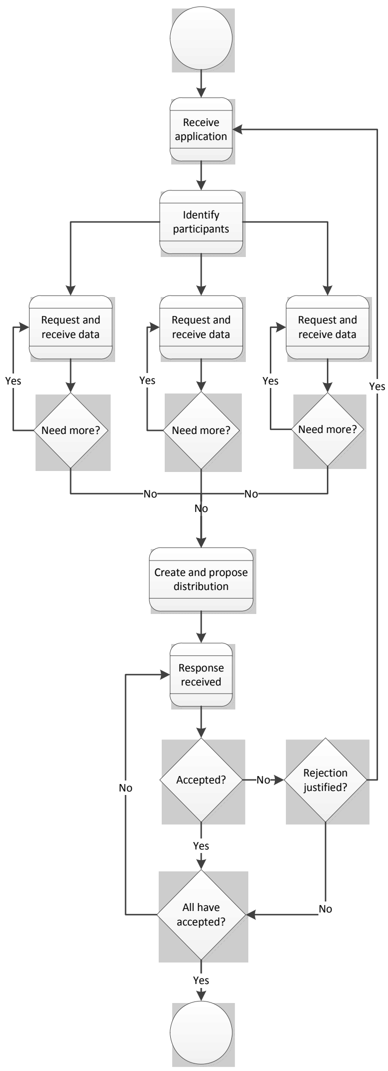

Fig. 1.Motor Insurance Bureau of Latvia, The preparation of responsibility distribution.

The process is quite simple, as shown in Fig. 1, however, many steps in the process may be repeated. The MIB receives the application from Insurer to evaluate relative amount of responsibility of causing the road accident for every participant in this accident.

The MIB clerk then gathers all data needed from all involved Insurers until the amount of data are sufficient to compose the distribution of responsibility. The count of blocks "Request and receive data" is equal to total count of distinct Insurers. The distribution is then proposed to all involved parties, which may accept or reject the distribution. If an Insurer rejects the distribution, MIB reviews the rejection grounds. If the grounds are sufficient, distribution is recompiled using new information. If the rejection grounds are not deemed sufficient, the initial distribution remains in force. The distribution is considered as finalized if all involved Insurers have accepted it (or failed to submit sufficient rejection reasons).

The process initially was presented to three experts who evaluated the conformity to cloud computing features based solely on their experience. Table III shows the results of their evaluation.

TABLE III

EXPERT EVALUATION

Evaluator	kSS	$SS(x)$	kEL	$EL(x)$	kBA	$BA(x)$	$K(x)$
Expert 1	1	1	1	2	2	4	2,75
Expert 2	1	3	2	3	2	4	3,20
Expert 3	1	3	1	2	2	0	0,00

Analyzing the results of expert evaluation it is seen that all experts consider the availability of the service as the most important feature, assigning the weight of "2". However, the respective assessments are quite different. Expert 3 bases his evaluation on assumption that new interfaces must be designed if data exchange is not realized with e-mail messages anymore. The e-mail was seen as weakest point also by Expert 1 in self-service assessment – the user may be forced to carry out actions not belonging to the process, for example, clean up the mailbox in order to receive e-mail message. The base value of the assessment is the average of Experts 1 and 2 evaluations, not taking into account the over-pessimistic opinion of Expert 3 – 2,98.

To determine total degree of conformity using the proposed method, the needed data was mined manually from the environment hosting the process.

To determine the degree of conformity to self-service, 1+4*[total involved vehicle count] fields were identified, shown in Table IV.

TABLE IV

SELF-SERVICE – FIELDS TO FILL

Field	For every vehicle?	Can be filled without specific actions or knowledge
Accident identifier	No	No
Registration number of vehicle	Yes	Yes
The number of insurance policy	Yes	Yes
Insurer	Yes	No
The relative responsibility, %	Yes	Yes

The distributions of responsibility mainly involve two vehicles – about 92% of all cases. The conformity to self-service will be calculated assuming for 2 vehicles:

$$SS(x) = 4 * \left[\frac{2*3}{1+2*4}\right] = 2\frac{2}{3} \sim 2{,}66$$

$$SS(x) = 2{,}77$$

In case of 3 vehicles..

To calculate conformation to elasticity the time period beginning on the 1st of January, 2011 and ending on the 30th of June, 2013 (total – 30 months) was chosen. During this period total of 4803 service instances were executedand132 incidents and/or exceptions registered concerning these instances.

Besides the e-mail server, the service is executed by middleware, containing approximately 1000 lines of code, and database layer, containing approximately 2500 lines of respective code. During the typical process (2 involved vehicles, 2 mutual adjustments, acceptance/rejection) approximately 20 e-mail messages are sent, 6 web services are invoked and 14 requests are sent to data base. Using these values, conformity to elasticity is evaluated as

$$EL(x) = 4 * \left[\frac{4803}{4803+132}\right] * \left[\frac{3500/40}{1+3500/40}\right] \sim 3.84$$

To determine the conformity to availability, one must take into account the future improvements. If the e-mail message exchange is scrapped and the new interfaces must be developed (say, for 4 different views), then $Res(x) = 0{,}25$, because one resolution is already processed. If the data exchange mechanism remains the same, then $Res(x) = 1$.

Prepared distribution of responsibility is accessible as PDF document, thus $Doc(x) = 1$, because PDF is platform-independent standard.

The reliability measurement is based on instances executed during the already mentioned period from 01-JAN-2011 to 30-JUN-2013. Total 4803 instances were registered, but only 4733 instances have completed successfully with status "Accepted" or "Rejected", making reliability $R(x) = 0{,}91$.

Putting together all intermediate results, the conformity to availability (if no new interfaces must be designed) is

$$BA(x) = 4 * 1 * 1 * 0{,}91 = 3{,}64$$

or

$$BA(x) = 4 * 0{,}25 * 1 * 0{,}91 = 0{,}91$$

if the new interfaces must be developed though.

Total values of conformity of the process $K(x)$ are shown in Table V. Separate results are for cases when new interfaces must or must not be developed. The availability assessment is carried out with different weight values – standard "1" and, influenced by experts, "2".

TABLE V
TOTAL CONFORMITY TO CLOUD COMPUTING FEATURES

kBA	New interfaces	
	NO	YES
1	3,38	2,47
2	3,45	2,08

Comparing the experts' evaluation with the method results, it is clear that in optimistic case (no new interfaces are needed) the method evaluation is higher, especially if the availability is considered more important than other features. On the other hand, if new interfaces are necessity, then the result drops significantly by almost 1 point comparing to average experts; value – 2,47 and 2,08 versus 2,98 – though not fully conforming the opinion of Expert 3, then at least tending to it.

Further analysis show that the method may not correctly compute specific cases of processes, for example, simple process with no data input and virtually no access to external resources will yield $SS = \mathbf{4}$, $EL = \mathbf{4}$, but, supporting only one resolution from 4, $BA = \mathbf{1}$. Total result will still be $K = \mathbf{3}$, although the costs to develop support for 3 new resolutions may be rather high. Still, no process transformations are needed.

V. SUMMARY

Currently there are no solutions in the market which allow efficient and cost-effective transformation of existing electronic service architecture to one that conforms to cloud computing. The objective is to create a method which would allow such transformation and take into account the quality criteria at the same time. The quality requirements may be contradictory between themselves so the optimum between them must be met. This paper aims to deal with the initial phase of the solution – the assessment of existing architecture model.

Five key features of the cloud computing architecture are shortly discussed in this paper as well as the assessment scale. The scale allows to quantitatively evaluating the conformity of existing model to every feature of cloud computing solution. Using these values it is possible to determine overall degree of conformity. The overall degree then is used to determine further actions – does one proceed with transformations of the existing model or develops entirely new model. New model usually is the best choice if the existing model does not conform to cloud computing at all or the amount of investment required for transformations is not cost-efficient.

The offered solution is verified using the business process administered by Motor Insurance Bureau of Latvia – „The preparation of responsibility distribution" which originally was implemented using e-mail messages. The results of the method use on practical example shows that it can be used in general cases and gives plausible results, although it may struggle in cases of specific processes behind some electronic services.

Considering the amount and type of data needed to execute the assessment method, it may be beneficial to use specific tools intended for data mining from execution environment – the instance count, incident count, reliability and availability data could be extracted using Process Mining Framework [13]. It is also worth to consider if data needed to compute the conformity to self-service could be extracted automatically from structured design documentation.

REFERENCES

[1] G. Reese, *Cloud Application Architectures*, O'Reilly, 2009, pp. 7-10.
[2] SETI@home [Online]. Available: http://setiathome.berkeley.edu/ [Accessed 26 October, 2013].
[3] R. L. Krutz, R. D. Vines, *Cloud Security: A Comprehensive Guide to Secure Cloud Computing*, John Wiley & Sons, 2010, pp. 33-61.
[4] S. A. Ahson, M. Ilyas, *Cloud Computing and Software Services: Theory and Techniques*, CRC Press, 2010, pp. 3-13.
[5] C. Hoff, *Christofer Hoff blog: Rational survivability*. [Online]. Available: http://rationalsecurity.typepad.com/blog/[Accessed 26 October, 2013].
[6] S. Fremdt, R. Beck, S. Weber, *Does Cloud Computing matter? An analysis of the Cloud model Software-as-a-Service and its impact on operational agility*, Proceedings of the Annual Hawaii International Conference on System Sciences, 2013, Article number 6479957, pp. 1025-1034.
[7] P. Mell, T. Grance, *The NIST Definition of Cloud Computing*, National Institute of Standards and Technology, Information Technology Laboratory, Technical Report Version 15, 2009.
[8] R. Buyya, J. Broberg, A. Goscinski, *Cloud Computing Principles and Paradigms, Wiley*, 2011, pp. 16-17.
[9] E. Folkerts, A. Alexandrov, K. Sachs, A. Iosup, V. Markl, C. Tosun, *Benchmarking in the cloud: What it should, can, and cannot be*, Lecture Notes un Computer Science, Volume 7755 LNCS, 2013, pp. 173-188.
[10] J. Montes, A. Sanchez, B. Memishi, M. S. Pérez, G. Antoniu, *GMonE: A complete approach to cloud monitoring*, Future Generation Computer Systems, 2013.
[11] H. Lu, X. Xia, X. Wang, *How to dynamically protect data in mobile cloud computing?* Lecture Notes in Computer Science, Volume 7719 LNCS, 2013, pp. 364-371.
[12] Z. Min, *Analysis of security issues in cloud computing*, Advances in Information Sciences and Service Sciences, Volume 5, Issue 5, March 2013, pp. 473-500.
[13] E. F. Duipmans, L. F. Pires, S. Da Silva, L.O.B., *Towards a BPM cloud architecture with data and activity distribution*, Proceedings of the 2012 IEEE 16th International Enterprise Distributed Object Computing Conference Workshops, EDOCW 2012, Article number 6406222, pp. 165-171.
[14] R. Dargha, *Cloud Computing Key Considerations for Adoption*, Infosys Technologies Whitepaper, 2009.
[15] A. Stellman and J. Greene, *Applied Software Project Management*, O'Reilly Media, pp. 39-48.
[16] R. Buyya, J. Broberg, A. Goscinski, *Cloud Computing Principles and Paradigms, Wiley*, 2011, pp. 51-54.
[17] S. Islam, K. Lee, A. Fekete, A. Liu, *How a consumer can measure elasticity for cloud platforms*, ICPE '12 Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering, 2012, pp. 85-96.
[18] W.M.P. van der Aalst, B.F. van Dongen, C.W. Gunther, R.S. Mans, A.K. Alves de Medeiros, A. Rozinat, V. Rubin, M. Song, H.M.W. Verbeek, and A.J.M.M. Weijters. *ProM 4.0: Comprehensive Support for Real Process Analysis*. In J. Kleijn and A. Yakovlev, editors, *Application and Theory of Petri Nets and Other Models of Concurrency* (ICATPN 2007), LNCS, vol. 4546, Springer, Berlin 2007, pp. 484-494.

Peteris Stipravietis earned his master's degree in computer sciences from University of Latvia in 2006. Currently he has finished his doctoral studies at Riga Technical University and is working on his dissertation.
He is working for SIA ZZ Dats, Riga, Latvia as a project manager and system analyst, beginning his career as a programmer.
His main research interest is design of electronic services, simulation and transformation of underlying business processes and workflows. Last publication in this field was on April, 2012 – "The Selection of the Electronic Service Considering the Process Flow", the 5th Int. Scientific Conference Applied information and communication technologies.
Contact information: peteris.stipravietis@zzdats.lv

Edzus Zeiris, Dr. Sc. ing., is working for SIA ZZ Dats, Riga, Latvia as a deputy director.
His main field of research is design of electronic services, security and evaluation of electronic services.
Contact information: edzus.zeiris@zzdats.lv

Maris Ziema, prof., Dr. Sc. ing., teaches in Riga Technical University since 1982. Since 2010 he is director of Institute of computer control, automation and computer engineering. Currently he is giving lectures on multimedia technologies and user interface design.
His main field of research is design of electronic services and computer systems; he has published over 80 publications in this field.
Contact information: maris.ziema@rtu.lv

Pēteris Stipravietis, Edžus Žeiris, Māris Ziema. E-pakalpojuma arhitektūras modeļa atbilstības mākoņskaitļošanai novērtēšana
Uz šo brīdi tirgū nav pieejami atbilstoši risinājumi, kas ļautu efektīvi atrisināt e-pakalpojumu sistēmu arhitektūras pielāgošanu mākoņskaitļošanai. Problēmas būtība ir veikt esošā e-pakalpojuma biznesa procesa atbalsta sistēmas arhitektūras modeļa transformāciju uz mākoņskaitļošanai piemērotu arhitektūras modeli, tajā pašā laikā meklējot optimumu starp sistēmas kvalitātes kritērijiem, kas bieži vien ir savstarpēji pretrunīgi. Šīs publikācijas ietvaros tiek aplūkota vispārīgā risinājuma sākotnējā daļa – esošās sistēmas arhitektūras novērtējums.
Publikācijā īsi aprakstītas piecas galvenās vispārīga mākoņskaitļošanas risinājuma pamatiezīmes, kā arī definēta vērtējumu skala, kas ļauj kvantitatīvi novērtēt e-pakalpojuma sistēmas arhitektūras modeļa atbilstību katrai no minētajām pamatiezīmēm. Izmantojot iegūtās vērtības, ir iespējams noteikt kopējo modeļa atbilstības pakāpi – balstoties uz šo lielumu, iespējams pieņemt lēmumu par tālāko darbību – un turpināt izstrādāt modeļa transformācijas vai arī izstrādāt jaunu modeli. Jauns modelis parasti tiek izstrādāts, ja pēc kopējā novērtējuma esošais modelis pilnībā neatbilst mākoņskaitļošanas pamatiezīmēm vai arī nepieciešami pietiekami lieli resursu ieguldījumi, lai transformācija nebūtu ekonomiski izdevīga.
Piedāvātais risinājums tiek pārbaudīts ar piemēru – Latvijas transportlīdzekļu apdrošinātāju biroja pārvaldībā esošu procesu „Atbildības sadalījuma sagatavošana”, kas sākotnēji tiek realizēts ar e-pasta ziņojumiem.

Петерис Стиправиетис, Еджус Жейрис, Марис Зиема. Оценка соответствия модели архитектуры электронных служб особенностям облачных вычислений.
В настоящее время на рынке нет решений, которые позволяют эффективно и экономично преобразовать архитектуру существующей электронной службы на такую, которая соответствует облачным вычислениям. Целью исследования является разработка метода, который реализует такую трансформацию и в то же время учитывает критерии качества. Требования к качеству могут противоречить друг другу, поэтому метод должен найти оптимальное решение между ними. В данной статье рассматривается начальная фаза решения - оценка существующей модели архитектуры.
В статье кратко обсуждается пять ключевых особенностей архитектуры облачных вычислений, а также шкала оценки соответствия. Шкала позволяет оценить соответствие существующей модели каждой особенности решения облачных вычислений. Используя эти значения можно определить общую степень соответственности. Общая степень затем используется для определения дальнейших действий - можно ли продолжить преобразования существующей модели или начать разработку совершенно новой модели. Разработка новой модели обычно является лучшим выбором, если существующая модель вообще не соответствует облачным вычислениям или сумма инвестиций, необходимых для преобразования, не является экономически эффективной.
Предлагаемое решение проверяется с помощью бизнес-процесса, который используется в Латвийском Бюро страховщиков транспортных средств - "Подготовка распределения ответственности", который первоначально был реализован с помощью сообщений электронной почты.