A SOFTWARE ARCHITECTURE TO DETERMINE CONTEXTUAL COMPONENTS IN CLOUD COMPUTING

C. Sushama[1], Dr. A. Rama Mohan Reddy [2]

[1] Dept of CSSE, Sree Vidyanikethan Engineering College, Tirupati, A. P., India
[2] Dept. of CSE, SVU College of Engineering, Tirupati, A.P., India

Corresponding author: C. Sushama, sushmasvuphd@gmail.com

ABSTRACT: This paper claims that the contextual qualities of Cloud-enabled software architecture should be identified and understood in a different way by Cloud customers. The existing architectures are not designed to manipulate the contextual qualities of Cloud computing. The appearance of Cloud computing virtually forces Cloud customers to re-evaluate their application architectures in light of the Cloud computing viewpoint which requires relinquishing control over most architectural components to Cloud suppliers. In a cloud-enabled architecture, the move to possession and control over architectural components from customers to Cloud suppliers has the powerful impact on the ways Cloud customers design their Cloud architecture. In this perspective, we go beyond the traditional definition of Cloud architecture and present the concepts of architectural scope, control and design components of the architecture.

KEYWORDS: software architecture, conceptual components, cloud-enabled architecture, scheduling.

1. INTRODUCTION

Cloud computing is an appealing model for the provisioning of IT solutions. The main idea behind cloud computing is to provide *x-as-a-service* to organizations, meaning *x* could substitute by any dynamically scalable service such as software, database, security, hardware, platform, storage, etc. Instead of installing application software on their machines, buying powerful computers and managing servers, organizations can use application software as well as computers owned and managed by someone else. Consumers need not have much expertise in or control over the computing resources that support the services they consume.

Regardless of the clear benefits offered by Cloud computing, how companies will determine their software architecture in the cloud-enabled environment is a task for the pushing Cloud computing adoption on a broader scale. The software architecture of a company has an effect on how it controls its processing needs and resources in Cloud computing [Boo96]. In current methods, software architecture includes several architectural elements [Kae10] that are mostly owned and handled by the company that uses it, and limit within the border of that company. Companies currently have full control over and possession of their architectural components. The Cloud computing is currently in the process of changing this practice. This change has an effect on how computing resources is distributed, used, operated, and managed by companies.

The migration from nearby computing environment to exterior services provides difficulties to software architects [KG09]. The difficulties include interpreting contextual qualities of architectural components, finding their effect, understanding the managing factors of architectural components, and making a balance between consumer-controlled and Cloud computing architecture [DAV14]. These issues power software architects to reconsider the current methods in light of the new frontier of architectural difficulties previously unaccounted. Although the contextual qualities have no directly visible expression in the software architecture, these do point out something important for all Cloud computing stakeholders.

Cloud computing has disturbed the architectural scenery [RKR14] that companies have so far experienced in developing and handling to meet their computing requirements. The restricted knowing of the actual contextual qualities of software architecture stops many companies from determining the real effect of cloud-enabled services. Current methods of software architecture are either too IT product-specific or too superficial [Koz11]. A further knowing in cloud-based architectures is required to allow companies for a broader adopting of this model. The interaction between the software architecture of a company and the new features of cloud computing should be well healthy and recognized by all stakeholders. In this paper, we mainly focus on software architecture to recognize its contextual qualities [PW92] in cloud computing.

With many providers looking for to move services to the Cloud to save on implementation expenses, take care of fast growth or usually reduce themselves from the liability of allocating the resources like software or hardware, bandwidth and power. There is a crucial need to ensure that the same service quality will maintain when depending upon Clouds while usually providing on the guarantee of decreasing expenses by reducing over-provisioning through efficient downscaling and upscaling of services [BK14].

This paper mainly concentrates on contextual qualities of software architecture that are essential for all stakeholders to know in a cloud-based environment. The recognized qualities are required to promote the understanding of application structure on Cloud-enabled architecture. In the remainder of the paper, Section II provides the architectural design patterns with an example. In Section III, contextual qualities are mentioned in regards to Cloud enabled architecture. Cloud computing and the associated Scheduling architectural issues are presented along with the major cloud service model in section IV. Section V presents the experimental design. Section VI concludes the paper.

2. ARCHITECTURAL DESIGN PATTERNS

Software architecture is a conceptual design that describes architectural components from which a system is designed using rules of connectors and interactions [ZZ09]. More accurately, software architecture is designed from the following elements:

Architectural Components: A set of architectural components that execute some predetermined features, or represent specific details. These are either the computational models with well-defined connections such as procedures, modules, filters, processes; or pieces of information such as databases, object, data, etc.

Connections: These are the compositional systems for sticking the components together in a topology. Illustrations of connections include procedure calls, messages, distributed factors, communication protocols, events, pipe streams, etc.

Architectural styles: It is a set of guidelines that mediate interaction, synchronization, and connections among architectural components [SG96]. The choice of communications is usually advised by the guidelines of an architectural design, and the connections types the geometrical framework [B+11] and control flow of the software system. In other words, architectural design describes a terminology of architectural components [PAB11], their communications and the control flow among elements. Illustrations of designs are pipe-and-filter, object-oriented, process model [FT02], etc. Different designs have different framework and topology.

To set the right perspective and illustrate the above elements, consider an example of a system that is developed in the object oriented architectural style. The cloud computing environment of Software as a Service (SaaS) architecture [DV13] in which it has three use case roles called as IT Administrator, SaaS Provider, and End user. The IT administrator is responsible for maintaining the data, managing the services and security. SaaS provider provides the services externally from the Internet to the company. End users are responsible for requesting the services and utilizing the services that are provided by the company.

Figure 1 shows a snapshot of the software architecture that uses object-oriented architectural model. The use cases (oval shape) are the architectural components, and the arrows and lines are the connections. The name of the use case specifies it role and responsibilities. The arrows and lines serve as conduits for the stream, transmitting of an output of one use case to an input of another.

3. CLOUD ENABLED ARCHITECTURE

A cloud-enabled architecture may consist of the consumer's on-premise as well as cloud resources such as services, components, application and the communications of elements. The management over architectural components is the most important contextual property [KV13] in Cloud computing. In Cloud architecture, the cloud service providers have their independent management circulation of the system. In a non-cloud environment, the software techniques and components are nearby in the company that uses it. Companies have specific management circulation in the architecture updated to personalize their business needs.

An architectural component can be on the border of either the customer or cloud provider in either On-premise or off-premise levels respectively. Architectural opportunity represents the actual physical submission of the architecture and its components. The topology of architecture may distribute across several allocated places beyond the customer company at one of two different levels (on premise and off-premise). That is, some of the architectural components may be situated outside of the consumer's business border [DV13]. For example, a process of software architecture may run on a server that connected to another company. In the same way, a server in the components architecture of a company may be set up in another organization [NMV13]; the disks may situate in a third company.

In Figure 2 it represents the SaaS Cloud reference architecture in which it dealt with two different architectures. One is for local cloud architecture, and another is for external cloud architecture. The local cloud contains the on-premise architecture in which the server is the local server is maintained to store and retrieve the files, and the data transfer will be of off-premise in the external server is maintain with the cloud provider. Components in movement, such as a part of the information, can have two levels in two different periods. A part of information can travel from the consumer's company to the provider's border. In that situation, it can be in two different levels once on-premise and then off-premise while it is prepared or saved on the cloud. However, a procedure or a components system can have only one level in its lifetime because these components are fixed.

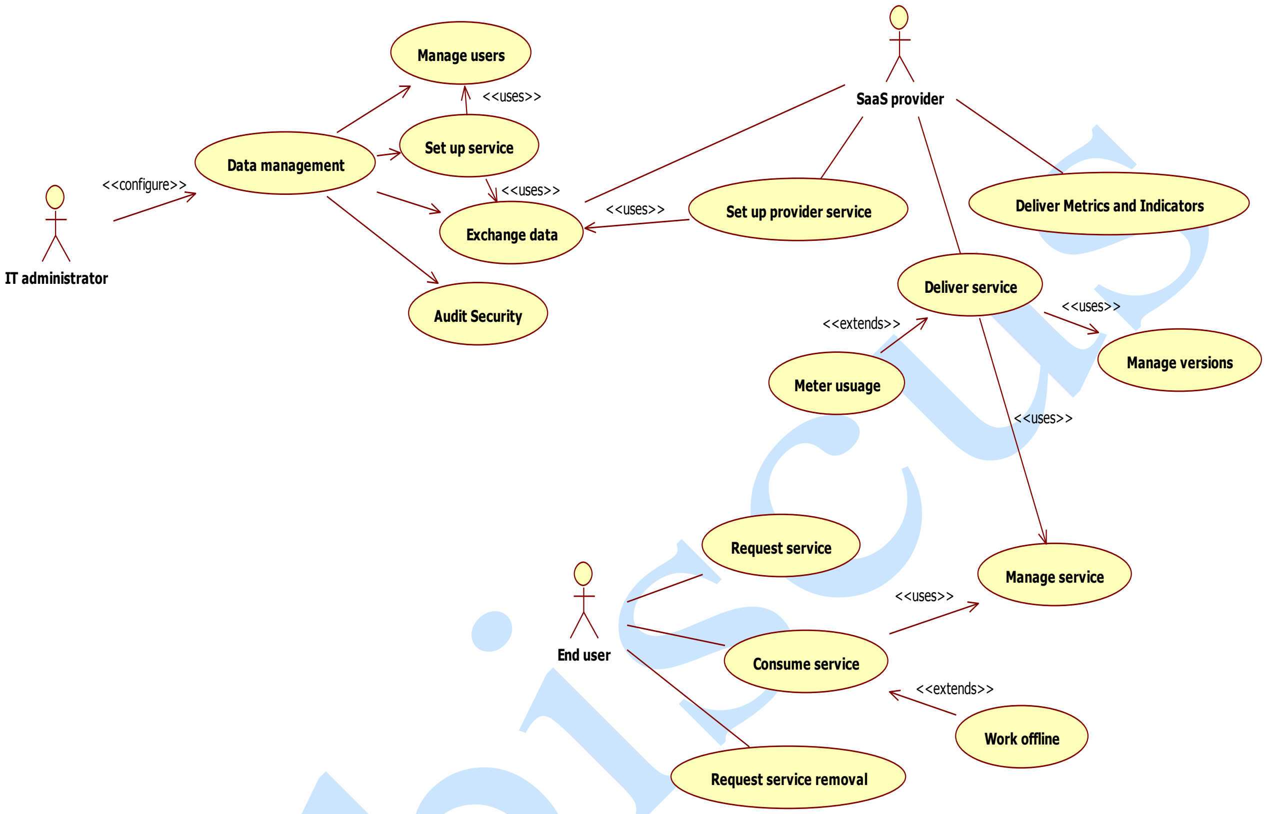

Figure 1: Use case Model for Software as a Service in Cloud

In Figure 3 it shows that with the contextual qualities, SaaS providers could clearly recognize which architectural components they own and which they don't. Although possession is associated with management, this is not always the situation as we have seen in the example. It is obvious that in cloud-enabled architecture, customers quit much of their management of their information and procedures due to modifying contextual qualities in a cloud-based architecture [Kha11]. Once the consumer's information keep the business border, management is organized by the cloud support agency in most situations.

The Cloud reference architecture contains the component of workflow engine which is responsible for scheduling process. This scheduling process is essential in cloud computing environment. It plays a vital role in managing the resources through the cloud management, and how these scheduling contextual qualities rely on the environment will be shown in the next section.

4. SCHEDULING ARCHITECTURE IN CLOUD COMPUTING

Scheduling is a decision process, and its content is implementing resources to applications of different customers at a suitable time, or during a specific period. Cloud computing scheduling model will be constructed by Architectural components like Client, Cloud Broker, Resources, Resources promoter and information Service [KV10]. Figure 3 shows the scheduling architecture of Cloud. The tasks applied can be separated into the parallel program, sequence program, and parameter checks out the program, collaboration program and so on.

Different customers use resources at different prices. The component Cloud Broker [RVK13] is a Middle interface between customers and resources. It is used to find resources, choose resources, accept tasks, return scheduling results, and exchange information. Cloud Broker facilitates different scheduling policies, which can spend resources and routine tasks in compliance with the requirements

of customers [RVK11]. The broker is constituted by Schedule Advisor, Job Control Agent, Explorer, Traveler, Deployment Agent, and Trade Manager.
Job Control Agent: It is accountable for tracking tasks in the software system, such as schedule generation, tasks development, the position of tasks and interacting with clients and schedule advisor.

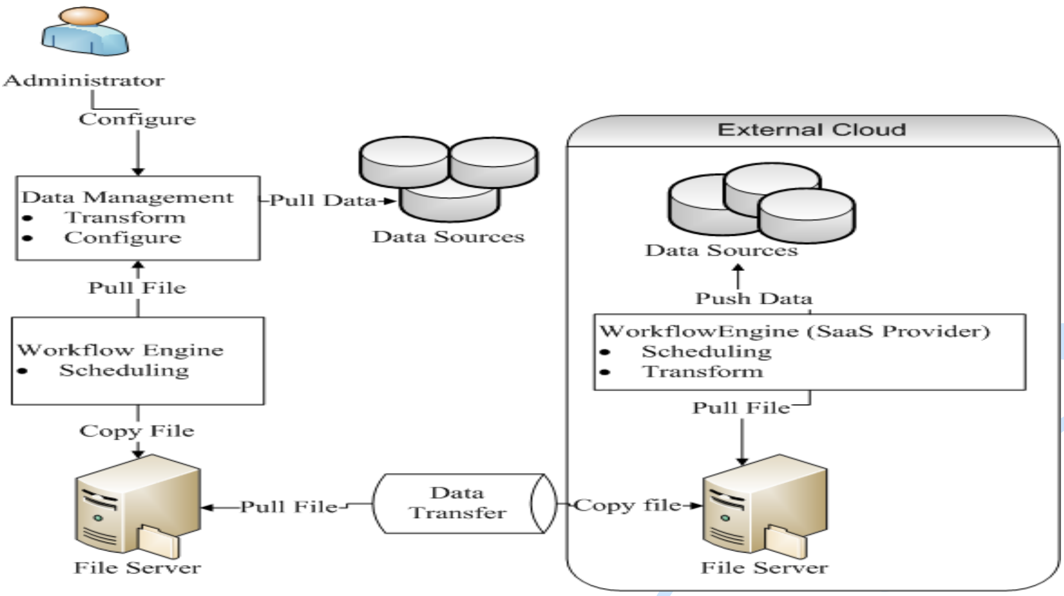

Figure 2: SaaS Cloud Reference Architecture

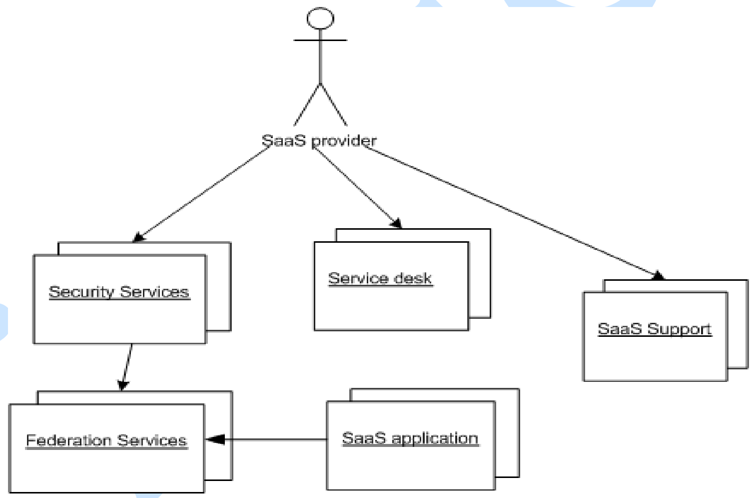

Figure 3: SaaS provider Components

Schedule Advisor: It is used to find out resources, assign available resources that fulfill the requirements of customers such as due date and cost, and to assign tasks.
Cloud Explorer: It corresponds to cloud information service to identify resources, recognize the list of approved machines and resource record status information.
Trade Manager: It identifies resources access cost and tries to converse with resources at a low price under the supervision of schedule advisor.
Deployment Agent: It employs schedule training to start the execution of tasks and revise the status of tasks execution to Job Control Agent at usual intervals.

This scheduling architecture can be implemented by using the scheduling algorithms. There are many scheduling algorithms, but here the scheduling architecture is tested on First Come First Serve (FCFS) algorithm. This FCFS is a very efficient policy to allocate the resource components to the specified agents.

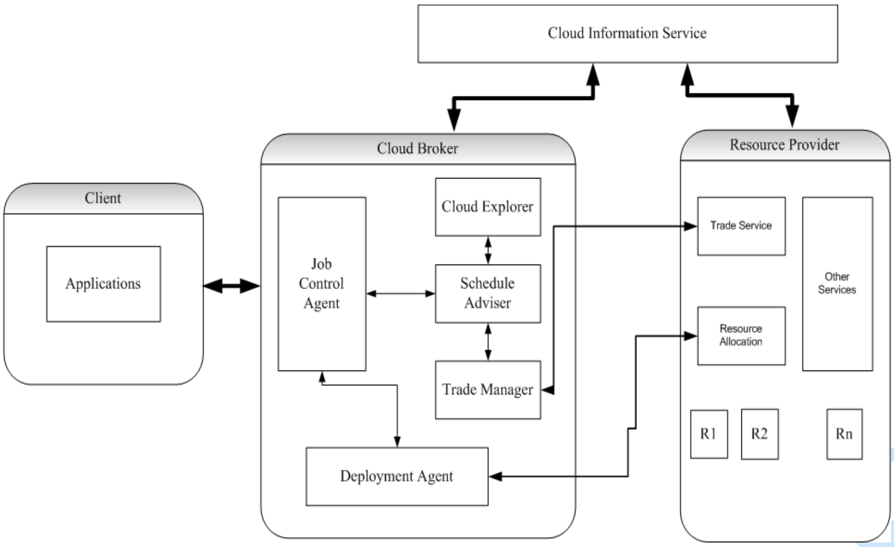

Figure 4: Scheduling Architecture in Cloud

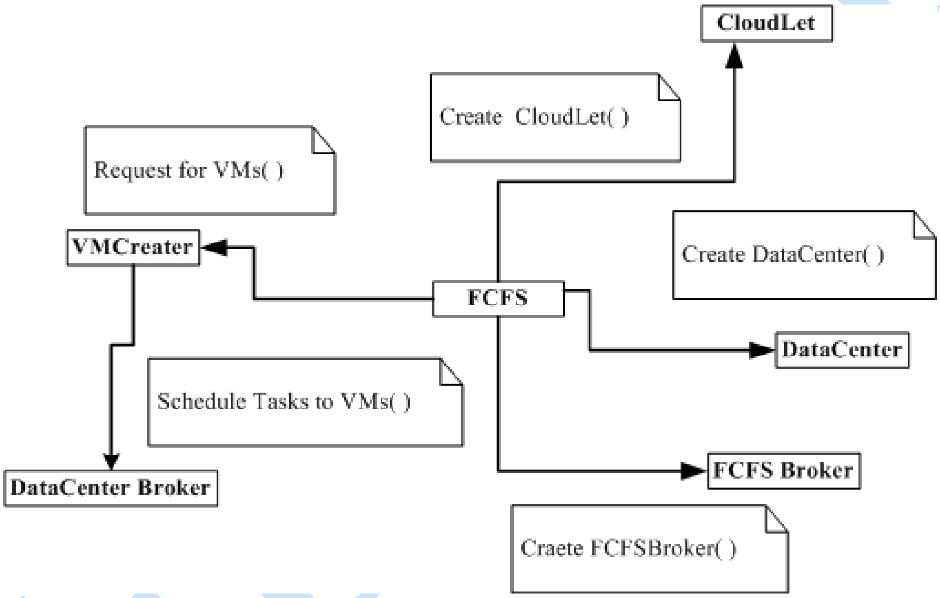

Figure 5: FCFS Testbed Architecture

5. EXPERIMENTAL DESIGN

The simulation setup for FCFS in Cloud enabled architecture has carried in a simulator named CloudSim. CloudSim is a Java component based work environment in which it is used for cloud application development. The scheduling of resources in the cloud environment can be understood easily based on this simulator. The real components that are involved in scheduling can be designed in a simple manner by observing their behaviors in cloud computing.

5.1 Experimental setup

For evaluating the performance of FCFS scheduling algorithm, CloudSim 3.0.3 is used to simulate the execution of workflow in cloud-enabled architecture. The configurations of Host, VMs, RAM, and Bandwidth are shown in Table1.

Table 1: Simulation design in CloudSim

Host	10
VMs	10-50
RAM	512
Bandwidth	1000
Scheduling	FCFS

The average execution time and response time is calculated by using the FCFS scheduling in the cloud; the test bed environment is created for the 10,20,30,40 cloudlets. At each level of allocation of VMs to the cloudlets records the response time and execution time.
Here we are using the constant VMs count that is taken as 15, 25 and 45. The experimental results for FCFS scheduling is shown in Table 2.

Table 2: Simulation results for FCFS Scheduling

Cloudlets	VMs	Execution Time (sec)	Response Time (ms)
	15	52.4	41.3
10	25	46.2	37.2
	45	41.9	32.4
	15	64.3	52.9
20	25	53.7	48.5
	45	47.2	44.6
	15	73.5	61.4
30	25	63.1	57.9
	45	57.3	53.8
	15	81.8	72.7
40	25	73.1	67.3
	45	67.2	62.8

5.2 Observations of FCFS Scheduling Architecture

In figure 6 it shows the comparison graph of execution time and no. of cloudlets that participate in the workflow scheduling. It defines that execution time is increased automatically when the no of cloud lets increases. The average response time is shown in figure 7. It is variable when we are using random VMs.

With the applying of above mentioned architectural components on Cloud based architecture along with the contextual qualities, the cloud customer could find more about their level of control over the architectural components, the possession along with the responsibilities and the architectural scope. These would allow them to plan their resources, architectural addressing, and software design with their business needs and observe the effectiveness of the Cloud-centered architecture. As Grady Booch [1] appropriately declares in a different perspective, architecture is an announcement of the distributed truth that symbolizes a typical perspective among a set of stakeholders and showed a set of interlock designs. It is quite real for a reasoning centered structure where both the customer and the Cloud provider discuss their knowing of the system in interlock relics. Software architecture associated with the contextual qualities could improve the interlock, and enhance the knowing of the relics.

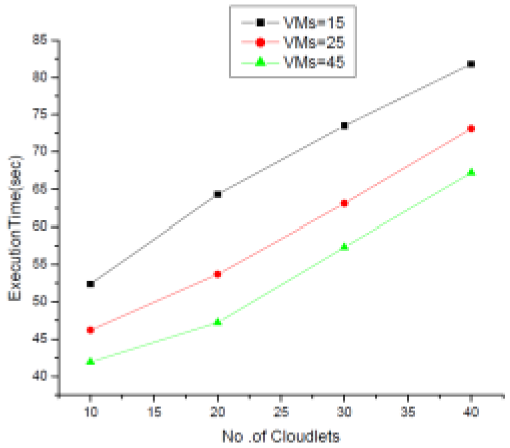

Figure 6: FCFS test results for Execution time

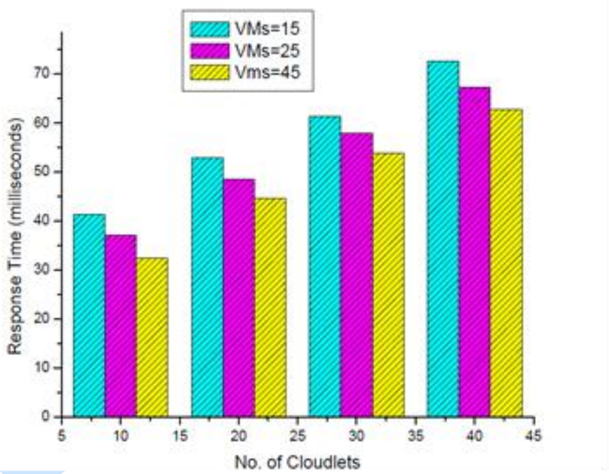

Figure 7: FCFS test results for Response Time

CONCLUSION

The new way of service intake in cloud computing needs the contextual qualities of software architecture regarding its key architectural components, their communications, the management circulation, possession of components and topological submission of components. The contextual qualities of application structure with clear management circulation provide a business positioning to everyone in the company to understand the effect of freelancing processing needs to Cloud. Architecture in a cloud-enabled environment is predicted to show how and where Cloud computing solutions fit into IT strategy, and how it has an effect on the way Cloud computing provides solutions to the business objectives of the company.

Applying a traditionally-structured architecture onto a Cloud environment without right contextual qualities would mean trying to fit a rectangle item into a circular gap. The architecture must create in such a way that it is quickly easy to understand and controllable across the companies. A healthy environment in creating architectures that can be used to offer enough technological details as well as contextual qualities to all stakeholders. Software architecture might be under the surface, but it is based on Cloud customers because they could see how fulfilling the solutions are, offered by the actual architecture and its elements.

This study demands further research to formalize the contextual qualities, and discovery on how to evaluate the effect of these qualities on software architecture. It contains developing techniques that could reason about the existence or lack of these qualities in software architecture and their effect on business procedures. We recognize that this subject needs further research to flesh out a complete taxonomy of cloud-enabled software architecture and the related contextual information.

REFERENCES

[Boo96] **G. Booch** - *Architecture as a Shared Hallucination*, IEEE Software, Jan-Feb. 2010, Vol. 27 (1), pp. 95 – 96.

[BK14] **L.D. Babu, P. V. Krishna** - *An execution environment oriented approach for scheduling dependent tasks of cloud computing workflows*, International Journal of Cloud Computing, 3(2), 209-224, 2014.

[B+11] **P. Banarjee et al.** - *Everything as a Service: Powering the New Information Economy*, IEEE Computer, Vol. 44(3), March 2011, pp. 36 – 43.

[DV13] **L. D. Dhinesh Babu, P. Venkata Krishna** - *Versatile Time-Cost Algorithm (VTCA) for Scheduling Non-Preemptive Tasks of Time Critical Workflows in Cloud Computing Systems*, Int. J. Communication Networks and Distributed Systems, Vol. 11, No. 4, pp. 390-411, Inderscience Publishers, 2013.

[DV13] **L. D. Dhinesh Babu, P. Venkata Krishna** - *Applying Operations Management Models for Facility Location Problem in Cloud Computing Environments*, International Journal of Services and Operations Management, Vol.15, No.1, pp.1 – 27, Inderscience Publishers, 2013.

[DAV14] **L. D. Dhinesh Babu, Angappa Gunasekaran, P. Venkata Krishna** - *A Decision Based Pre-emptive Fair Scheduling Strategy to Process Cloud Computing Work-flows for Sustainable Enterprise Management*, International Journal of Business Information Systems, Vol. 16, No.4, pp. 409-430, Inderscience Publishers, 2014.

[FT02] **R. Fielding, R. Taylor** - *Principled design of themodern Web architecture*, ACM Transactions on Internet Technology, Vol. 2(2), 2002, pp. 115 – 150.

[Kae10] **G. Kaefer** - *Cloud Computing Architecture*, Siemens AG 2010 Corporate Technology, Munich, May 2010.

[Kha11] **Y. Khalidi** - *Building a Cloud Computing Platform for New Possibilities*, IEEE Computer, Vol. 44(3), March 2011, pp. 29 – 34.

[Koz11] **H. Koziolek** - *The SPOSAD Architectural Style for Multitenant Software Applications*, Proceedings of the Ninth Working IEEE/IFIP Conference on Software Architecture, 2011, pp. 320 – 327.

[KG09] **K. Khan, N. Gangavarapu** - *Addressing Enterprise Architecture in Cloud Computing: Issues and Challenges*, IT Cutter Journal, November 2009, pp. 27 – 33.

[KV10] **K. Kalaiselvan, P. Venkata Krishna** - *Grid To Cloud (G2C) – A Infrastructure based transition*, CSI Communications, Vol. No. 33, Issue No. 11, pp. 22-25, February 2010. (ISSN 0970-647X).

[KV13] **K. Kalaiselvan, P. Venkata Krishna** - *Performance Based QoS for Cloud's Infrastructure as a Service*, International Journal on Cloud Computing, Vol. 2, No. 4, pp. 325-339 Inderscience Publishers, 2013.

[NMV13] **Neha Tekriwal, Madhumita, P. Venkata Krishna** - *Integration of Safety and Smartness Using Cloud Services- An insight to future*, Innovations and Advances in Computer, Information, Systems Sciences, and Engineering, Lecture Notes in Electrical Engineering Volume 152, pp 293-303, Springer, 2013.

[PW92] **D. Perry, A. Wolf** - *Foundations for the Study of Software Architecture*, ACM

SIGSOFT Software Engineering Notes, Vol. 17(4), Oct 1992, pp. 40 – 52.

[PAB11] **M. Papazoglou, V. Andrikopoulos, S. Benbernou** - *Managing Evolving Services*, IEEE Software, May/June, 2011, pp. 49 – 55.

[RKR14] **T. S. K. Reddy, P. V. Krishna, P. C. Reddy** - *Power Aware Framework for Scheduling Tasks in Grid based Workflows*, Int. J. Communication Networks and Distributed Systems, Inderscience Publishers, 2014.

[RVK11] **M. Rajasekhara Babu, P. Venkata Krishna, M. Khalid** - *Optimization Techniques and Performance Evaluation of a Multithreaded Multi-core Architecture Using OpenMP*, LNCS series of Computer Communication and Information Science (CCIS) 190, pp. 182–191, 2011, (Springer-Verlag Berlin Heidelberg 2011).

[RVK13] **M. Rajasekhara Babu, P. Venkata Krishna, M. Khalid** - *A Framework for Power Estimation and Reduction in Multi-Core Architectures Using Basic Block Approach*, Int. J. Communication Networks and Distributed Systems, Volume 10, Number 1/2013, pp. 40-51, Inderscience Publishers, Netherlands, 2013.

[SG96] **M. Shaw, D. Garlan** - *Software Architecture: Perspectives on Emerging discipline*, Prentice-Hall, 1996.

[ZZ09] **L. Zhang, Q. Zhou** - *CCOA: Cloud Computing Open Architecture*, Proceedings of the IEEE International Conference on Web Services", 2009, pp. 607 – 616.