DuDE-Cloud: A Resilient High Performance Cloud

Peter Danielis, Jan Skodzik, Vlado Altmann, Frank Golatowski, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics and Computer Engineering
18051 Rostock, Germany, Tel./Fax: +49 381 498-7277 / -1187251
Email: peter.danielis@uni-rostock.de

***Abstract*—In recent years, cloud computing, which provides users with network resources such as memory and computing power depending on the users' needs, has gained enormously in importance. Providers such as Amazon allow the users to access their cloud storage and computing resources by different interfaces. In this regard, guaranteeing compatibility is a severe issue as many different proprietary interfaces exist for accessing data in different clouds. The solution to this problem is the RESTful Cloud Data Management Interface (CDMI) standard, which has passed the ISO audit as first standard and is therefore evolving into the most common standard for accessing clouds. This paper investigates the combination of CDMI with a P2P-based storage and computing back-end in order to realize a self-organizing cloud for distributed data storage and processing called DuDE-Cloud. The basis is the self-organizing hash table (DHT)-based P2P storage and computing back-end called DuDE, which utilizes the DHT protocol Kad. DuDE is able to bundle existing storage and computing resources of different devices dynamically and has already proven its advantageous performance over centralized solutions. In a test scenario, resources of DuDE-Cloud were successfully accessed through CDMI using a GUI front-end thereby proving the proper functionality. As a result, a resilient high-performance distributed cloud solution is available that is compatible with other clouds.**

I. Introduction

Today, an ever increasing number of devices, sensors, and all kinds of electronic components are networked in a dynamic Internet of Things, in which these nodes communicate with each other directly and also process and store data. They offer services to each other so that the Internet of Thing has become an Internet of Things and Services. As a prerequisite, it is assumed that there are tasks that the devices can complete cooperatively such as collecting and aggregating data. After having processed this data, the nodes can distribute the results to other nodes interested in these results. This procedure describes a collaborative, participatory, and cooperative sensing and is often referred to as mobile cloud computing nowadays.

This Internet of Things and Services is even predicted to find its way into the factory. For this development, e.g., in Germany the term Industry 4.0 has been coined [1]. In the most general sense, a networked and real-time capable industrial production is aspired globally. The US-American company General Electric (GE) recently initiated a comprehensive research initiative called Industrial Internet [2]. Thereby, not only the industrial production but also the whole industrial infrastructure shall be intelligently networked. GE forecasts for the future that there will be more intelligent devices, which have to be connected to interact with each other dynamically; both Cisco and Ericsson talk about 50 billion devices by 2020. As part of these efforts, companies aim prospectively at connecting their devices, storage systems, and sensors as well as actors to form cyber-physical systems. In that way, intelligent devices, storage systems, and sensors as well as actors shall be created in the industrial production, which exchange data in a self-organizing way, trigger actions, and control each other in real-time.

Existing centralized storage and computation systems will soon no longer be able to process and store these ever increasing data volumes, which results from the rising number of devices [2]. Consequently, we have developed a reliable scalable peer-to-peer (P2P)-based back-end called DuDE—a Distributed Computing System using a Decentralized P2P Environment [3], [4]. The distributed hash table (DHT)-based P2P network Kad has beeen utilized to exemplarily connect the devices of an Internet service provider's access networks—called access nodes (ANs)—in a self-organizing way. However, DuDE is not limited to ANs but can be applied to connect devices in general. In the use case investigated in [3], [4], ANs store log data, from which statistics are calculated. DuDE allows for complex statistics calculation for a single AN or even for all ANs. Common statistics like the number of dropped packets, which ANs usually compile, are supported. Moreover, DuDE's modular design simplifies the incorporation of new statistics formats. DuDE has been both simulated and successfully implemented as software and has already proven its advantageous performance over centralized solutions [3], [4].

In this paper, we want to make the DuDE back-end accessible by standard Web services to use it as a cloud. In this regard, guaranteeing compatibility is a severe issue as many different proprietary interfaces exist for accessing data in different clouds. The solution to this problem is the Representational State Transfer (REST, RESTful) Cloud Data Management Interface (CDMI) standard, which has passed the ISO audit as first standard and is therefore evolving into the most common standard for accessing clouds [5]. This paper investigates the combination of CDMI with the P2P-based storage and computing back-end DuDE in order to realize a self-organizing cloud for distributed data storage and processing called DuDE-Cloud. Before we provide the proof of concept in Section V, we give a brief description of the DuDE system based on [3], [4]. Below, the following main contributions are briefly described:

- Description of the DuDE-Cloud architecture.
- Proof of concept by means of a test scenario.

The remainder of this paper is organized as follows: Section II contains a comparison with related work. Section III gives a brief description of CDMI and DuDE. The DuDE-Cloud architecture is explained in Section IV. Section V provides the proof of concept before the paper concludes in Section VI.

II. Related Work

This section is mainly intended to show the enormous diversity of standardization bodies and services in the field of cloud computing and underlines the necessity for a common standard.

A. Cloud Standardization Bodies

This subsection discusses the key institutions that deal with projects aiming at developing a uniform standard interfaces for clouds. Such an interface should serve as the basis for an improved communication between different cloud services. Below, organizations are listed, which work on specific solutions [5].

NIST: The USA was the pioneer for security efforts in the cloud computing sector. As first council, the National Institute of Standards and Technology (NIST) of the American Department of Commerce developed a standardization plan and a reference architecture. The council also follows the developments of Cloud Security Alliance (CSA) and the Open Grid Forum (OGF). The European Commission currently orients its cloud standardization efforts towards the work of the NIST [5].

IEEE: The IEEE usually works on networking standards. However, currently they are developing a design overview and a standard for interoperable cloud services called the IEEE P2301 project [6]. Content of this project is to identify and describe different profiles. These should allow hardware and software from different manufacturers to communicate with each other and to ensure the interoperability of platforms as well. In parallel, the IEEE working group P2302 is working on standards to be determined. these should allow for the cooperation of several cloud services. This should become the basis for a so-called Inter-Cloud [7].

SIENA: The Standards and Interoperability for E-Infrastructure Implementation Initiative (SIENA) is currently working on a project, which deals with a consistent European cloud infrastructure. They are closely connected with the standardization bodies such as IEEE, NIST, and the Open Grid Forum (OGF) [8].

OGF: OGF consists of developers, users, and merchants, which have already engaged in the field of the standardization of grid computing and are still working on further developments [9]. In 2006, OGF was founded as an organization consisting of the alliance of the Enterprise Grid Alliance and the Global Grid Forum. They deal with the identification of standards in the field of grid and cloud computing. Especially in the field of cloud computing, OGF is working on an interface, which enables the interaction with a cloud infrastructure. The main task is to ensure that the user can decide about the acquisition, monitoring, and determination of cloud services in the form of a remote management. Specifically, there is a working group, which is composed of community groups and leading organizations. Subsequently, the Open Cloud Consortium Interface (OCCI) was developed [10]. This interface should be used for the entire process of managing virtual machines. OCCI uses the REST standard. Thereby, the individual data sets can be addressed by Uniform Resource Locators (URLs). Interoperable programs can therefore be used for single tasks such as for autonomous scaling and monitoring of applications.

SNIA: The Storage Networking Industry Association (SNIA) is currently working on standards for virtualized storage resources and storage systems. The used resources and systems are to be adapted to the cloud environment. The search for a standard is limited to storage resources in the network. The SNIA has developed an interface called Cloud Data Management Interface (CDMI). CDMI provides the connection between the storage devices and storage systems. Thereby, users of cloud services can centrally access their data on all storage systems. The interface is available for all architectures of cloud computing, i.e., can be used in the public, private, and hybrid cloud [Sch11]. The CDMI standard is the first standard, which has passed the ISO audit and is therefore evolving into the most common standard for accessing clouds [11], [12], [5]. Consequently, it has been selected as interface in this paper.

TM Forum: The TM Forum is an emerging industry association, which develops solutions for the IT services sector [13]. Among other things, various standards have been developed by initiatives like the Cloud Service Initiative (CSI).

OASIS: The Organization for the Advancement of Structured Information Standards (OASIS) has developed standards like Extensible Markup Language (XML) and Web services. In the area of cloud computing, the organization is mainly concerned with security. The organization has launched a so-called Identity in the Cloud Technical Committee, which basically deals with safety standards. This resulted in the ID-Cloud, which includes requirements for identity management of cloud computing [14].

DMTF: The Distributed Management Task Force (DMTF) is a standardization organization comprising many IT companies [13]. The organization works on interoperability standards, which aim at avoiding compatibility issues and enabling standardized interfaces between cloud environments. The DMTF has developed the important Open Virtualization Format (OVF).

EuroCloud: The EuroCloud is an association of cloud computing service providers [15]. It has brought a quality seal for cloud services to the market. Standards such as ISO 2000 and SAS-70 are included in the evaluation of cloud computing services.

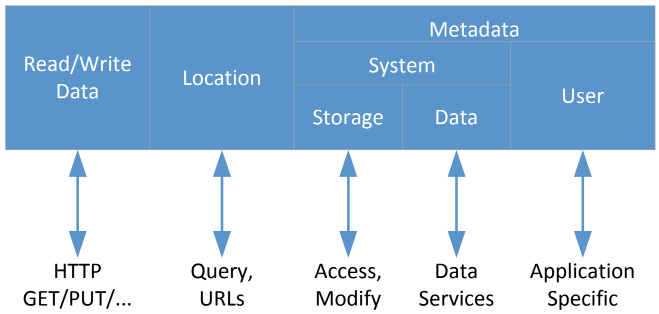

Fig. 1. General data storage interface for clouds [11]

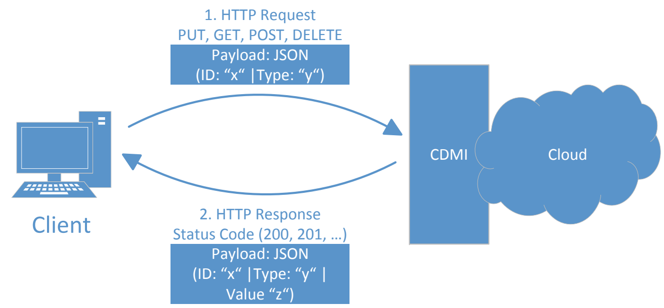

Fig. 2. Basic flow of CDMI [11].

B. Selected Cloud Environments

Amazon Web Services: Amazon Web Services (AWS) is the collective term for all Amazon cloud service and includes, among other things, Amazon Simple Storage Service (S3) [16]. By definition, the access to Amazon S3 is provided by Web Services using a Simple Object Access Protocol (SOAP) or REST interface. The latter is preferable, especially when dealing with large objects as REST can handle large objects better than SOAP.

Google App Engine: The Google App Engine is a comprehensive cloud solution including a programming environment, tool support, and an execution environment [17]. For the permanent storage of data, Google App Engine applies Datastore, which is a schema-less object-oriented database with a query engine offering guarantees regarding the atomicity of operations. The datastore implements both a proprietary interface and standard interfaces such as Java Data Objects (JDO) and Java Persistence API (JPA).

Eucalyptus: Eucalyptus allows the construction and operation of a private cloud infrastructure [18]. The interface of Eucalyptus is compatible with Amazon S3. The software development is done under the BSD license making it open source.

Apache Hadoop: Hadoop is an open source software platform that allows to process and analyze large amounts of data in a computer network in a simple manner [19]. The distributed file system of Hadoop works together with the Amazon S3 service.

OpenCirrus Project: The OpenCirrus activities include both developments on the infrastructure level, the platform level, and the application level. Unlike other cloud environments like Google App Engine or AWS, OpenCirrus allows scientists and developers full access to all system resources via an arbitrary interface [20] .

OpenStack: OpenStack is an open source software platform, which allows the construction of a cloud [21]. OpenStack is based on the utilization of open standards and interfaces, in particular RESTful interfaces based on HTTP. In addition, the project is compatible with the interfaces of AWS.

C. P2P-based Distributed Storage and Computing Back-Ends

DuDE, which is used as P2P-based distributed storage and computing back-end in this paper, has already been simulated and successfully implemented as software and has proven its advantageous performance over centralized solutions in terms of resilience, performance, and scalability [3], [4]. DuDE has been extensively compared to state-of-the-art works like [22], [23], [24], [25], [26]. For a more detailed description, the interested reader is referred to [3], [4].

III. Basics

In this section, basics are provided covering CDMI for accessing clouds and the DuDE functionality in general.

A. CDMI for Cloud Access

CDMI defines the functionality and security settings of a cloud service. In Figure 1, the general data memory interface for clouds is shown, which CDMI applies. By using this interface, the user has the ability to control and manage his data and storage container.

The interface is not only intended for the users but can also be used by administrative and management applications. It allows for managing objects, access rights, and information about customer and billing data. CDMI is particularly suitable when the storage access is done by protocols such as File Transfer Protocol (FTP), Web-based Distributed Authoring and Versioning (WebDAV) or the well known HyperText Transfer Protocol (HTTP) [11]. The RESTful HTTP protocol forms the core of the interface. A unique identification (ID) number is assigned to each object. With this number, the relevant data object can be accessed. This ID is a string containing requirements for generating and obtaining unique objects. Any vendor that uses the interface is able to identify objects without conflicts with other providers. The generated part of the customer data and metadata is secured by a consistent authentication using the Transport Layer Security (TLS). Using the metadata, functions are stored along with other data via CDMI and are assigned to the data containers [11]. The individual data services are designed as individual data elements that are determined by the metadata. This metadata specifies the requirements on the basis of individual data elements or a set of data elements, the so-called containers. The metadata is generated by the cloud storage system, specified by the cloud

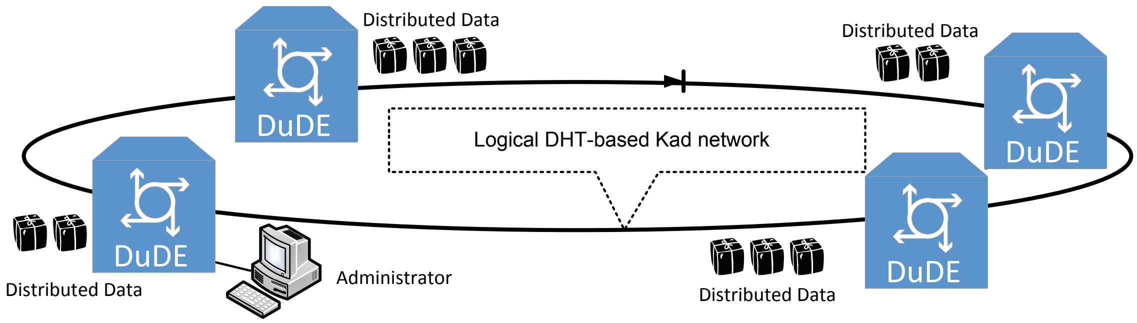

Fig. 3. Kad-based DHT ring connecting DuDE nodes for distributed statistics computing and distributed data storage.

user, and is mapped to a mounted file system directory. The data objects represent a fundamental memory component in CDMI and comparable to files in a file system.

The interface allows the client to locate resources available in the memory and to manage containers with their data. CDMI defines two ways to manage data. The data retrieval is referred to as data path. That part for managing data is called control path. To transmit the result of a RESTful operation, the HTTP status code is used. As data representation, the JavaScript Object Notation (JSON) format is used.

Figure 2 shows the communication via CDMI. The procedure is carried out by means of a request/response process whereby the CDMI client issues a request to the cloud. The communication is done via the HTTP protocol on the basis of a Transmission Control Protocol (TCP) connection. The request contains the task to be processed and uses HTTP methods such as GET, POST, etc. Furthermore, specific data can be added to the request in the JSON format. After the request was received and processed by the interface, the server responds with a response. It includes an HTTP status code, is generated by the interface and sent to the client. The status code indicates if a transfer has been completed successfully. This is possible due to incorrect or even invalid addresses. Specific data in the JSON format is included in the response as well.

B. DuDE Functionality

P2P technology in access networks: In the use case investigated in [3], [4], DuDE was directly implemented on the ANs of an Internet service provider's access network but it can generally be implented on any device offering sufficient memory and computing resources. The DuDE devices, i.e., nodes must cooperate to combine their computing power. Thereby, we avoid the usage of the client-server model as the server represents a bottleneck and single point of failure. Consequently, a decentralized system realized by exploiting the high scalability of P2P technology has been chosen.

DHT-based systems as structured decentralized P2P systems offer the best trade-off between lookup and storage complexity [27]. DHT systems associate nodes and functions or data with the help of a hash function. Moreover, DHTs are self-organizing and do not need a central control instance. Additionally, failing peers can be compensated and detected automatically within the system by means of maintenance mechanisms. In addition to the good complexity of DHTs, they show a deterministic lookup therby avoiding false negatives. Typical representatives of DHT-based P2P protocols are Chord [28], Tapestry [29], Pastry [30], and Kademlia [31]. The Kademlia protocol has been selected due to its best trade-off in terms of lookup and storage, flexible routing table, and its simple worst case analysis [27]. Every node has its own routing table containing some other nodes in the DHT network. This allows to perform lookups with the complexity of $log_2(N)$, whereby N denotes the number of nodes in the network.

Kad protocol modifications: Kad as an implemented realization of the Kademlia protocol has been used to realize DuDE [32]. All DuDE nodes are organized into a Kad-based DHT, whereby each DuDE node in the P2P system takes over the responsibility for a data part. DuDE stores data in the network redundantly, which increases data availability. Additional search objects have been implemented to extend the Kad protocol and to support distributed data collection. Moreover, new packets have been implemented to achieve an efficient communication between DuDE nodes. In Figure 3, the network is depicted as logical ring on top of the real network topology.

Performing the distributed computing: Selected distributed computing aspects are used to avoid overloaded components, which exist in centralized computation systems. Eligible DuDE nodes participating in the computation may become task watcher and/or job scheduler. Their suitability depends on their available resources. DuDE nodes can themselves decide about their participation, which leads to a high grade of flexibility and load balance. A job scheduler is a high-performance node, which is responsible for distributing computation parts (tasks) to DuDE nodes with sufficient available resources, which are called task watchers. Each task watcher computes a part and sends it back to the job scheduler after complete computation. An administrator is able to request computed statistics from any DuDE node and does not need to be connected to the job scheduler.

In summary, the novel system DuDE combines P2P technology, an extended Kad protocol enabling complex distributed data storage, and distributed computing. For more details on the DuDE functionality, the interested reader is referred to [3],

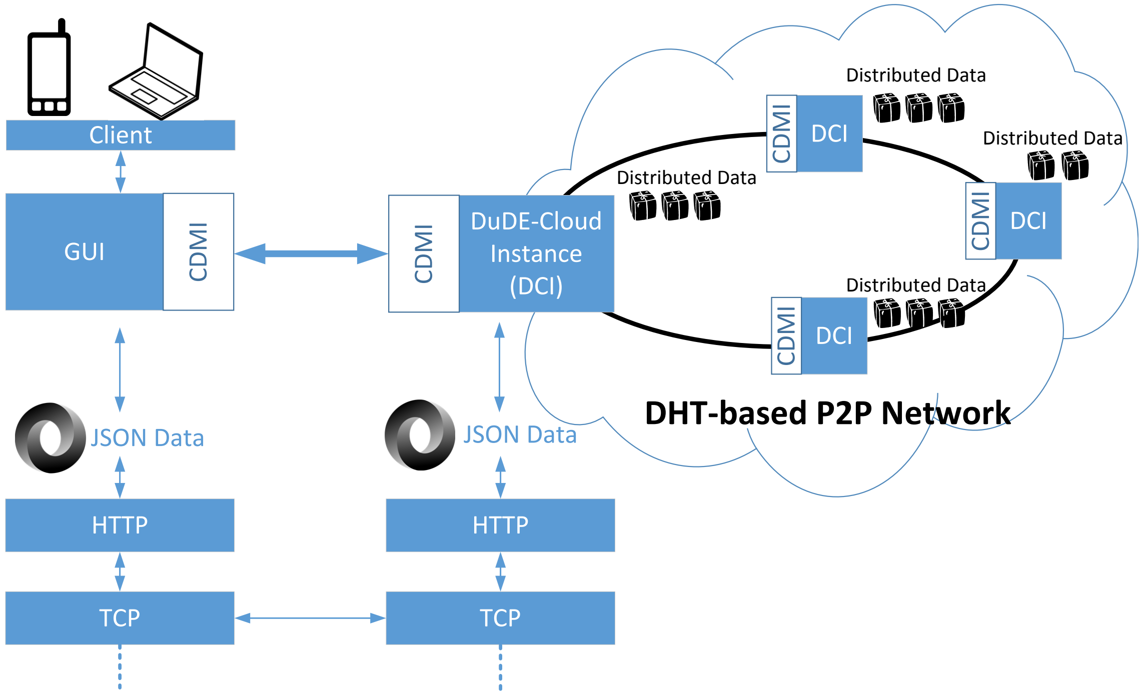

Fig. 4. DuDE-Cloud concept combining the P2P network with CDMI.

[4]. To make the DuDE back-end accessible by RESTful Web services in a standard compliant way and thereby achieving interoperability with other clouds, it has to be combined with CDMI.

IV. The DuDE-Cloud Architecture

The concept described in this section is the basis for the implementation of a standardized cloud interface. All necessary steps to integrate CDMI are described here. Furthermore, the presented concept combines CDMI with the DuDE GUI front-end to allow a comfortable and flexible access to the DuDE back-end. An overall view of the concept is depicted in Figure 4.

Any task defined by the user in the DuDE GUI is transmitted via the client CDMI to the connected DuDE-Cloud Instance (DCI). Each DCI has its own CDMI, via which clients are physically connected to it. The DuDE back-end is organized and structured via Kad and represents the cloud from the external point of view.

The DCI, which receives tasks from the clients, must interpret the message properly and gives a response to the corresponding client. The (en)coding of the task must be implemented on both sides (client and corresponding DCI). This is also realized in a standardized way by utilizing the JSON standard for the payload data describing the tasks.

A. GUI Front-End

To get access to the DuDE back-end, a GUI has been developed. The GUI already offers any client the possibility to discover all available features, which are directly supported by the DuDE back-end (and therefore the DuDE-Cloud). By this separations and platform independent implementation different clients like smartphones and common computers can be addressed. To ensure a reliable data transmission between the clients and DCIs, TCP connections are applied. This is realized by using the HTTP protocol to perform operations using CDMI.

The transmitted data is encoded as JSON, which is created by the DuDE GUI automatically. The DuDE GUI only requires the desired task as input from the client.

B. DuDE Back-End

The DuDE back-end has to store data and may also be used to solve a task such as, e.g., to determine different statistics like the CPU utilization of all participants in the network. These participants can be the connected clients or even the DCIs itself. The DuDE back-end receives the client message via CDMI and has to parse the JSON string to extract the necessary parameters of the given task.

In summary, CDMI requires two main aspects, which are the TCP-based HTTP communication and the JSON encoded data. Therefore, the functionality, which have to be implemented to realize the concept, consists of:

- Implementing the TCP-based HTTP communication between the clients and DCIs
- Implementation and integration of a JSON encoder within the GUI of the client
- Implementation and integration of a JSON parser on the DCI side
- Extension of the DCI side to interpret the content extracted from the JSON string to interact with the under-

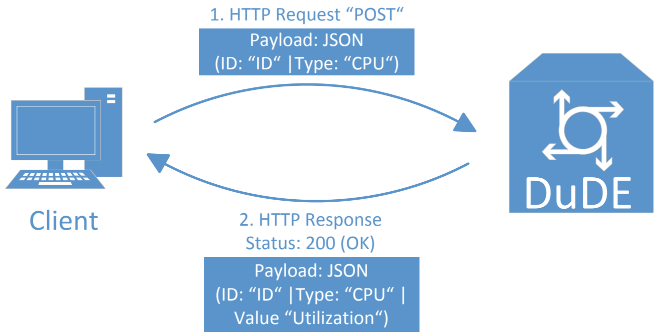

Fig. 5. Test scenario for proof of concept.

lying P2P network

V. Proof of Concept

This chapter provides the proof of functionality with respect to CDMI by means of a test scenario. Using the test scenario, CDMI is shown to be working according to its specification. For the practical implementation, the operating system Windows 7 Professional has been used. The DuDE back-end was implemented using the development environment Visual Studio 2011 C++. For implementing the GUI front-end, the Qt development framework 4.7 has been used.

A. *Test Scenario*

The test scenario is depicted in Figure 5. The GUI front-end and the DuDE back-end communicate via a HTTP connection. The GUI sends the generated JSON string to the DuDE back-end via CDMI exemplarily containing a request for CPU utilization of the nodes in the DuDE network. The JSON string is transmitted by an HTTP request using the POST method. From the DuDE back-end, the HTTP status code 200 is sent back to signal a successful processing. Further, the JSON string contains the requested CPU utilization. In the test scenario, it could be proven that our CDMI implementation works in accordance to its specification. The task of computing the CPU utilization could be completed and results were transmitted to the GUI via CDMI.

VI. Conclusion

The main result of this publication is a fully functional interface for accessing clouds. This interface follows the CDMI standard based on RESTful Web services. As cloud, the P2P-based decentralized back-end called DuDE was used. This back-end is based on the principle of distributed computing, in which each node is assigned a specific task. A GUI front-end has been developed for controlling the network. The combination of the GUI front-end, CDMI, and the DuDE back-end form the so-called DuDE-Cloud. By means of the GUI, a request with a task to be solved has been sent to the DuDE back-end. Exemplarily, the CPU utilization of DuDE nodes were calculated. This task has been formatted as JSON string and is transmitted to the P2P-based cloud sent via a HTTP request. In a test scenario, the functionality of the DuDE-Cloud could be successfully validated.

Prospectively, further investigation regarding real-time data processing in the DuDE-Cloud will be carried out.

Acknowledgement

Europäische Fonds EFRE, ESF und ELER in Mecklenburg-Vorpommern

This work is partly granted by the Research Fund Mecklenburg-West Pomerania, Germany, as well as the European Social Fund.

References

[1] Communication Promoters Group of the Industry-Science Research Alliance, acatech - National Academy of Science and Engineering, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0," Industrie 4.0 Working Group, Tech. Rep., April 2013. [Online]. Available: http://www.plattform-i40.de/finalreport2013

[2] P. C. Evans and M. Annunziata, "Industrial Internet: Pushing the Boundaries of Minds and Machines," General Electric, Tech. Rep., November 2012.

[3] J. Skodzik, P. Danielis, V. Altmann, J. Rohrbeck, D. Timmermann, T. Bahls, and D. Duchow, "Dude: A distributed computing system using a decentralized p2p environment," in *Local Computer Networks (LCN), 2011 IEEE 36th Conference on*, October 2011, pp. 1048–1055.

[4] P. Danielis, J. Skodzik, V. Altmann, B. Kappel, and D. Timmermann, "Extensive analysis of the kad-based distributed computing system dude," in *IEEE Symposium on Computers and Communications (ISCC 2015)*, July 2015.

[5] C. Saravanakumar and C. Arun, "Survey on interoperability, security, trust, privacy standardization of cloud computing," in *Contemporary Computing and Informatics (IC3I), 2014 International Conference on*, November 2014, pp. 977–982.

[6] Cloud Computing Portability and Interoperability: Cloud Portability and Interoperability, The Open Group. [Online]. Available: http://www.opengroup.org/cloud/cloud/cloud_iop/cloud_port.htm

[7] M. Kretzschmar and M. Golling, "Security management spectrum in future multi-provider inter-cloud environments - method to highlight necessary further development," in *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on*, October 2011, pp. 1–8.

[8] Standards and Interoperability for eInfrastructure Implementation Initiative (SIENA). [Online]. Available: http://www.sienainitiative.eu/

[9] Open Grid Forum (OGF). [Online]. Available: http://www.snia.org/about/alliances/ogf

[10] Open Cloud Computing Interface (OCCI). [Online]. Available: http://occi-wg.org/

[11] X. Luo and H. Li, "Experiment design for cloud storage application based on cdmi," in *Open-Source Software for Scientific Computation (OSSC), 2011 International Workshop on*, October 2011, pp. 148–152.

[12] Y. Weiwei, G. Chunhua, W. Feng, S. Hanyu, and Z. Jiping, "A cloud storage synchronization strategy based on cdmi," in *Innovative Computing Technology (INTECH), 2014 Fourth International Conference on*, August 2014, pp. 120–129.

[13] DMTF and TM Forum, "Cloud Management for Communications Service Providers," DMTF and TM Forum, Tech. Rep., 2012. [Online]. Available: http://www.dmtf.org/sites/default/files/standards/documents/DSP2029%20_1.0.0a.pdf

[14] OASIS, "OASIS Identity in the Cloud TC," 2015. [Online]. Available: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=id-cloud

[15] EuroCloud Organization, "EuroCloud Europe," 2015. [Online]. Available: http://www.eurocloud.org/

[16] Amazon Web Services Homepage. [Online]. Available: http://aws.amazon.com/de/

[17] Google Apps Engine. [Online]. Available: https://www.google.com/work/apps/business/

[18] Eucalyptus: Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems. [Online]. Available: open.eucalyptus.com

[19] T. White, *Hadoop: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2009.

[20] R. Campbell, I. Gupta, M. Heath, S. Y. Ko, M. Kozuch, M. Kunze, T. Kwan, K. Lai, H. Y. Lee, M. Lyons, D. Milojicic, D. O'Hallaron, and Y. C. Soh, "Open cirrustmcloud computing testbed: Federated data centers for open source systems and services research," in *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing*, ser. HotCloud'09. Berkeley, CA, USA: USENIX Association, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855533.1855534

[21] OpenStack Open Source Cloud Computing Software. [Online]. Available: https://www.openstack.org/

[22] D. Niu and B. Li, "Circumventing server bottlenecks: Indirect large-scale P2P data collection." ICDCS, 2008, pp. 61–68.

[23] D. Castella, I. Barri, J. Rius, F. Gine, F. Solsona, and F. Guirado, "CoDiP2P: A Peer-to-Peer Architecture for Sharing Computing Resources," in *Advances in Soft Computing*, vol. 50, 2008, pp. 293–303.

[24] K. Graffi, A. Kovacevic, S. Xiao, and R. Steinmetz, "SkyEye.KOM: An Information Management Over-Overlay for Getting the Oracle View on Structured P2P Systems." ICPADS, 2008, pp. 279–286.

[25] H.-M. Xu, Y.-J. Shi, Y.-L. Liu, F.-B. Gao, and T. Wan, "Integration of cloud computing and p2p: A future storage infrastructure," in *Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), 2012 International Conference on*, June 2012, pp. 1489–1492.

[26] J. Paiva, J. Leitao, and L. Rodrigues, "Rollerchain: A dht for efficient replication," in *Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, August 2013, pp. 17–24.

[27] R. Steinmetz and K. Wehrle, *Peer-to-Peer Systems and Applications*, ser. Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 2005.

[28] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications," in *ACM SIGCOMM*, 2001, pp. 149–160.

[29] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz., "Tapestry: A Resilient Global-scale Overlay for Service Deployment," in *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, 2004.

[30] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *IFIP/ACM International Conference on Distributed Systems Platforms*, 2001, pp. 329–350.

[31] P. Maymounkov and D. Mazieres, "Kademlia: A peer-to-peer information system based on the xor metric." IPTPS, 2002.

[32] R. Brunner, "A performance evaluation of the kad-protocol," Master's thesis, University of Mannheim, November 2006.