Establishing Trust in Cloud Services via Integration of Cloud Trust Protocol with a Trust Label System

Vincent C. Emeakaroha[1], Eoin O'Meara[2], Brian Lee[2], Theo Lynn[3] and John P. Morrison[1]
[1]*Irish Centre for Cloud Computing and Commerce*
University College Cork, Cork, Ireland
[2]*Irish Centre for Cloud Computing and Commerce*
Athlone Institute of Technology, Athlone, Ireland
[3]*Irish Centre for Cloud Computing and Commerce*
Dublin City University, Dublin, Ireland
. {*vc.emeakaroha, j.morrison*}*@cs.ucc.ie, eomeara@research.ait.ie, blee@ait.ie, theo.lynn@dcu.ie*

Keywords: Cloud Trust Label, Cloud Trust Protocol, Cloud Monitoring, Quality of Service, Service Level Agreement, Cloud Computing

Abstract: Cloud computing has transformed the computing landscape by enabling flexible compute-resource provisioning. The rapid growth of cloud computing and its large-scale nature provide many advantages to business enterprises. Lack of trust in cloud services however remains a major barrier to the adoption of cloud computing. Trust issues typically relate to concerns about the location, protection and privacy of data. The concept of trust includes trust in both novel technologies and cloud service providers and is composed of both persistent and dynamic trust elements. We have previously developed a trust label system that is capable of facilitating both persistent and dynamic trust building in both cloud services and cloud service providers; however, it lacks reliable information delivery mechanisms. The Cloud Trust Protocol (CTP) aims to provide a reliable information communication system for cloud service consumers and thus offers a natural complement to the trust label system by providing a reliable and interoperable information exchange mechanism. In this paper, we propose a novel integration of the trust label system with CTP to provide end-to-end visibility of cloud service operational information to consumers.

1 INTRODUCTION

The adoption of cloud computing is growing at such rates that some analysts forecast public IT cloud services will account for more than half of worldwide software, server, and storage spending growth by 2018 (Gens, 2014). Highly scalable, shared and distributed infrastructure that can be rapidly deployed at arms-length can accrue many business benefits, not least quicker response times to customer requirements, work efficiencies, and reduced IT costs (Buyya et al., 2009; Hogan et al., 2011; Low et al., 2011). Unsurprisingly, enterprises are attracted by the combination of IT efficiencies and business agility that the cloud computing model can offer (Kim, 2009).

Despite the widely reported benefits of cloud computing, lack of trust in cloud services remains a major barrier to its adoption. Such trust issues typically relate to concerns about the location, protection and privacy of data in storage, in transit and while being processed in the cloud. However, these issues consist of a wider class of security concerns including increased vulnerabilities relating to virtualisation, authentication and identification, web application and network security (Leimbach et al., 2014; Bradshaw et al., 2012; Subashini and Kavitha, 2011; Pearson and Benameur, 2010). The concept of trust in computing is complex. It involves trust in both a novel technology and external cloud service providers and is composed of both persistent and dynamic trust elements (Lynn et al., 2016; Pearson and Benameur, 2010). The global and distributed nature of the cloud exacerbates trust issues by introducing a lack of clarity on the identity and role of various actors within the chain of service provision, their associated responsibilities, and potential liabilities in the event of service failure.

Given these challenges, various mechanisms have been proposed for introducing accountability and assurance (Pearson and Benameur, 2010; Lynn et al.,

2013; Zissis and Lekkas, 2012; Pearson, 2013) in an effort to address trust issues but they suffer limitations, such as lack of dynamism. Previously, the authors have presented a validated trust label to collate and communicate trust in both cloud services and cloud service providers (Lynn et al., 2016; Emeakaroha et al., 2016; Masevic et al., 2016). The trust label system facilitates both persistent and dynamic trust building in cloud services and cloud service providers; however, it assumes reliable information delivery mechanisms are in place to communicate the required data to populate the interface.

The Cloud Trust Protocol (CTP) (Cloud Security Alliance, 2015a) is a recent initiative by the Cloud Security Alliance (CSA). It aims to provide a reliable information communication system for cloud service consumers to ask for and receive information about essential security configuration and operational characteristics as applied to cloud service providers. As such the CTP provides a natural complement to the trust label system by providing a reliable and interoperable information exchange mechanism. In this paper, we propose a novel integration of the trust label system with CTP to provide end-to-end visibility of cloud service operational information to consumers. This is facilitated by the implementation of a robust bridge library. Based on a use case scenario, we demonstrate the efficiency of the integrated system.

The rest of the paper is organised as follows: Section 2 discusses the challenges and presents some background information on the trust label system and CTP. In Section 3, we detail the system integration and describe the supporting components. Section 4 describes the implementation choices and the realisation of the system while Section 5 presents its evaluation. In Section 6, we discuss related work and conclude the paper in Section 7.

2 BACKGROUND

This section discusses trust-related challenges in cloud computing and how trust labels and the Cloud Trust Protocol purport to address these challenges.

2.1 Challenges

There is a well-established literature base on the importance of trust as a determining factor in the acceptance and use of online technologies (Gefen, 2003; Pavlou and Gefen, 2004; Mcknight et al., 2011; Bente et al., 2012). Similarly, in the cloud computing context, lack of trust has been consistently identified as a major barrier in the adoption of cloud computing in both academic and professional literature (Hwang and Li, 2010; Bradshaw et al., 2012; Leimbach et al., 2014). Trust is commonly defined as a "....psychological state comprising the intention to accept vulnerability based on positive expectations of the intentions or behavior of another" (Rousseau and Sitkin, 1998). In the context of information technology, the object of our trust is not necessarily a human person but a technology artefact and/or the provider of that artefact (McKnight and Chervany, 2001). Similarly, stakeholder expectations in relation to trusting technology are different, focussing on functionality, reliability and helpfulness in an IT context (Mcknight et al., 2011). The nature of cloud computing exacerbates issues of trust in many different ways, primarily by increasing perceived risk. Cloud computing involves an element of outsourcing of system and/or infrastructure to a third party on a shared distributed basis. As such, in adopting cloud computing, stakeholders face relational, performance, and regulatory and compliance risks. Relational risk is concerned with risk associated with poor cooperation and opportunistic behaviour while performance risk relates to operational or performance factors that may undermine the success of an outsourcing project (Das and Teng, 1996). Compliance and regulatory risk is concerned with the end-consumer failure to adhere to regulatory standards because of provider errors (Anderson et al., 2014). These high-level risks surface in commonly cited practical concerns relating to data jurisdiction and location, security and data protection, and portability and technology transparency (Bradshaw et al., 2012; Pearson and Benameur, 2010).

In cloud computing, trust issues are typically addressed through contracting (Pearson and Benameur, 2010). As cloud computing involves both persistent trust (trust in the long term underlying properties or infrastructure) and dynamic trust (trust specific to certain states, contexts, or short-term or variable information), parties to cloud computing contracts seek to minimise their risk by relying on both a general contract and a service level agreement.

While contracting may reduce risk and distrust, it does not necessarily build trust in the same way that trust mechanisms based on experience between parties or with a service might (Lewicki and Bunker, 1996; Deitz and DenHartog, 2006).Lynn et al. (Lynn et al., 2013) propose a dynamic trust label to enable and maintain consumer trust in cloud services. Similar to a food nutritional label, they argue such a trust label for cloud computing would work by succinctly communicating up-to-date values for a number of high-level dependability measures generated

from the underlying cloud service. Using an online adaptation of the Delphi conference method, Lynn et al. (Lynn et al., 2016) identified the information required for the label from a panel of 28 industry experts. Figure 1 presents a graphical view of the trust label interface. It comprises 81 information components, covering the cloud service provider (e.g. physical location, legal jurisdiction), the cloud service itself (e.g. data location, security, backup, certification), and a historical service-level summary (e.g. uptime data, support response times).

2.2 Trust Label System

Emeakaroha et al. (Emeakaroha et al., 2016) present the underlying trust label system as shown in Figure 1. The historic service-level summary, in particular, is based on a monitoring framework that is capable of monitoring the different cloud layers and presenting service-level data to the trust label in real time. Masevic et al. (Masevic et al., 2016) report that the proposed trust label performed as hypothesised in validation experiments with a sample of 227 potential cloud consumers.

2.3 Cloud Trust Protocol

As mentioned earlier, the Cloud Trust Protocol (CTP) (Cloud Security Alliance, 2015a) is a protocol, designed by the Cloud Security Alliance (CSA), to generate evidence-based confidence that everything that is claimed to be happening in the cloud is indeed happening as described. The CTP is a means by which cloud service consumers ask for and receive information about the elements of transparency, i.e. important pieces of information concerning the compliance, security, privacy, integrity, and operational security history of service elements being executed in the cloud. It aims to enable user confidence in the security and performance capabilities of cloud services both when choosing a service or continually monitoring a service to validate its agreed SLAs.

As such CTP provides evidence to build dynamic trust in cloud services in the vei if Pearson and Benameur (Pearson and Benameur, 2010) Figure 2 shows an overview of CTP deployed in a multi-cloud service provider environment. CTP presents a standardised unified API that is designed to be a RESTful protocol. The APIs are used by consumers to query cloud service providers for information relating to a particular service.CTP provides the information to be queried using a standardised data model that can be adapted to various scenarios that implement or intend to use CTP.

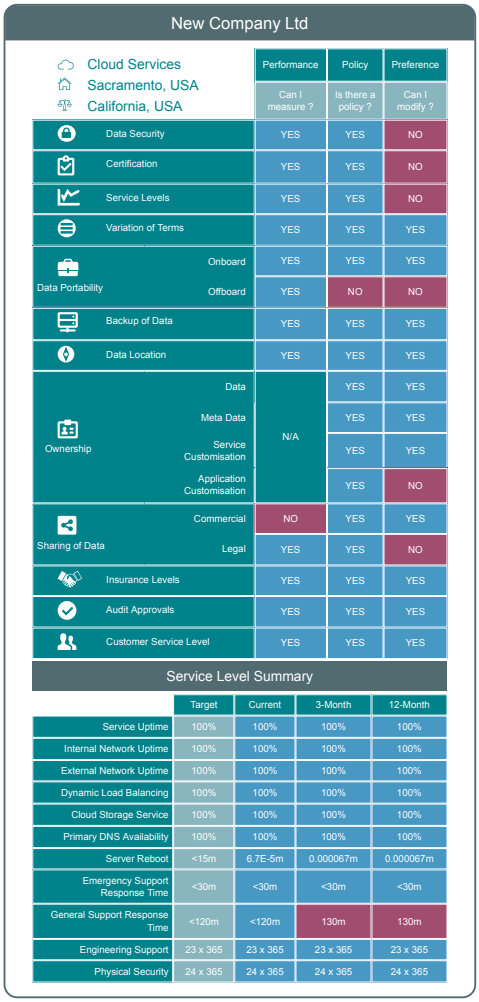

Figure 1: Trust Label Interface Overview

3 SYSTEM ARCHITECTURE INTEGRATION

This section describes the integration of the trust label system (Emeakaroha et al., 2016) with the Cloud Trust Protocol.

The aim of this integration is to add compliance and transparency to the communication of trustworthiness information in cloud service provisioning. Such information is gathered by a monitoring framework that is usually part of a provider's management layer. In our case, it is communicated through the trust label interface to the cloud consumers. Figure 3 presents an overview of the system integration architecture.

This system integration demonstrates the feasibil-

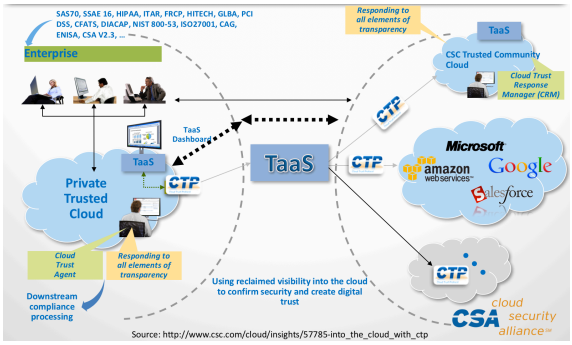

Figure 2: CTP Deployment Overview

ity of communicating information about cloud service operations in a standards compliant and transparent manner. As shown in Figure 3, the Monitoring Framework gathers the information for eventual transmission to the Trust Label system, which displays the data on a customisable web interface for cloud consumers. To realise a standard and transparent means of communicating this information, we adopted the CTP Server prototype provided by the Cloud Security Alliance.

The CTP Server provides a standard data model for describing cloud services and management data such as monitoring information. This is key for providing compliance and it should be clearly understood at design and implementation time to enable interoperability and compatibility.

In the integrated system as shown in Figure 3, the monitoring framework is extended with a bridge library that provides access to the CTP server. The bridge library enables communication between the monitoring framework and the CTP Server as well as between the CTP server and the trust label interface. This is realised using REST APIs. The communication directions could be both ways. We use the dotted arrows to indicate that the CTP server supports a query-response pattern of communication.

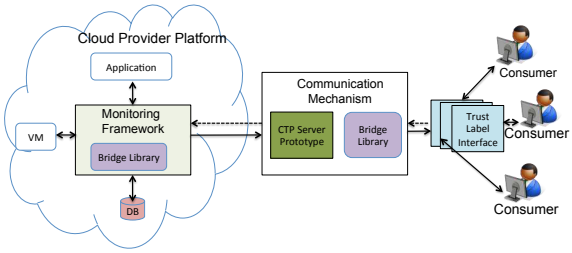

Figure 3: System Integration Architecture

The trust label interface is customisable and can be setup independently for different consumers based on their Service Level Agreements (SLA) with the cloud provider.

4 IMPLEMENTATION DETAILS

This section describes the implementation of the integrated system. We focus on the bridge library and provide a brief overview of the components with references for further details.

4.1 Bridge Library

The CTP bridge is a Spring Boot application written in Java. This component implements interfaces to the CTP server, Monitoring framework and Trust Label interface. It provides APIs for writing monitoring data after formatting them using the standard data model into the CTP server and also for reading from the CTP server to update the Trust label interface. The APIs for the different communications are as follows:

- CTP Admin REST API - for updating the CTP server prototype with the monitoring data and other information.
- CTP REST API - for getting information/data from the CTP server prototype.
- Trust Label REST API - for updating the Trust Label interface with the information gathered from the CTP server.

The above three APIs are supported by their own Java classes, each a Spring component, which inherit common code from another class for the basics of REST API invocation, including passing tokens in headers for security. The common code includes standard Spring mechanisms to invoke REST APIs.

The CTP Bridge uses Spring Scheduling to invoke the code to copy from the CTP Server to Trust Label every 10 seconds. A simple annotation on the method to invoke it ensures it is triggered automatically by the Spring Framework. Gradle is used as the dependency manager and build system for the project.

4.2 Other Components

Monitoring Framework: This component is implemented using the Java Programming language. It uses monitoring agents for gathering low-level resource metrics. At the application level, it employs HTTP pings among other methods to supervise application execution status. For storing monitoring data, it uses the MySQL database. Hibernate is used to realise easy interaction between the Java classes and the database. Further details about this component can be found in (Emeakaroha et al., 2016).

CTP Server Prototype: This is implemented as a Unix-style daemon written in the Go language. It

uses the MongoDB NoSQL database as a backend to store data. CTP has been tested on Linux and Mac OS X environments. The current implementation includes APIs for reading and writing to the server. Further information can be read from (Cloud Security Alliance, 2015b).

Trust Label Interface: This component is realised using Ruby on Rails. It uses a backend database to maintain the displayed content. Due to the generic nature of Ruby, it is capable of using different database engines such as SQLite, Postgre SQL, MySQL, etc. It implements RESTful APIs and uses the JSON data interchange format for structuring data. Additional information is available from (Emeakaroha et al., 2016).

5 EVALUATION

This section presents a demonstration of the integrated system as a proof of concept. We show how quality of service information is being communicated based on a use case scenario and evaluate the performance of the bridge library.

5.1 Use Case Scenario

We present a use case scenario of service provisioning between a cloud provider and a consumer. The monitoring of this service and the communication of the service status facilitate trust establishment between the provider and the consumer. The service is a transactional video-serving web application that responds to requests and makes queries to storage databases. It runs on top of a load balancer that distributes the incoming request to compute resources to sustain heavy load. The expected quality for the service is specified in a Service Level Agreement (SLA) document that is signed between the provider and the consumer to guide the provisioning of the service. Table 1 presents the SLA metric values for this particular service.

Table 1: Service SLA Document.

Metric Name	QoS Target
Service Uptime	100%
Internal Network Uptime	100%
External Network Uptime	100%
Dynamic Load Balancing	100%
Cloud Storage Service	100%
Primary DNS Availability	100%
Server Reboot Time	$<$ 5mins

The cloud provider makes every effort to guarantee the QoS targets specified in the SLA. Any violation of the SLA incurs a penalty for the provider. In order to assure consumers of their service quality, the provider needs to provide them with real time evidence of their service operations.

5.2 Quality of Service Performance

As discussed in the use case scenario, providers should furnish consumers with continuous evidence of their service performance in order to gain their trust. This section details how our proposed integrated system facilitates the communication of such information.

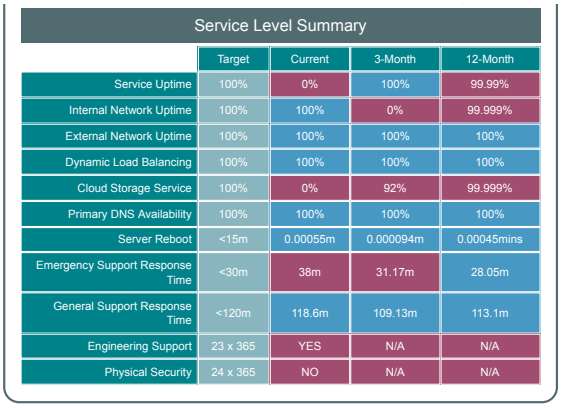

Figure 4: QoS Performance Information

Figure 4 depicts a screenshot of an extract of the Trust Label interface showing in real time the communication of the quality of service metrics. The information contains the current monitored values for the SLA metrics specified in Table 1 and other information. For simplicity, we modelled the display of the SLA values on the Trust Label interface in terms of 0% or 100% whereby 0% indicates a violation and 100% shows satisfied.

As shown in Figure 4, the Trust Label interface conveys to the consumer the current performance status and three and twelve month performance averages. These data give the consumer a clear overview of the service operation and a basis to trust the cloud provider. Our proposed integrated system facilitates this trust establishment by enabling this end-to-end communication of trustworthiness information.

5.3 Bridge Library Benchmarking

In this section, we present our benchmarking of the bridge library communication performance. The aim is to demonstrate the non-intrusiveness of this component in the end-to-end integration.

Figures 5 and 6 present a plot of an extract from the use case scenario execution. The application in the

use case scenario was executed for about three hours and we selected the middle hour execution period to avoid startup and wind-down effects. Thus, the figures show a full hour execution.

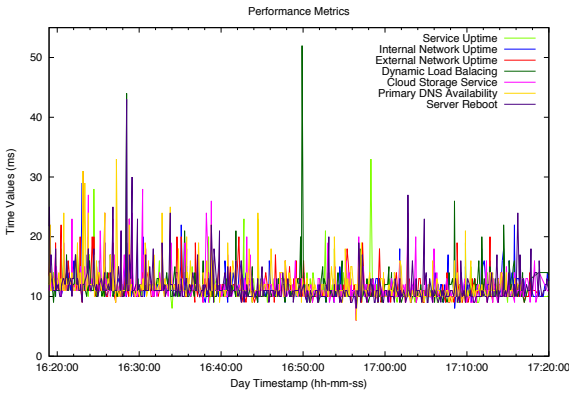

Figure 5: Monitor-to-CTP Performance

Figure 5 shows the time it takes to communicate each of the monitored SLA metrics from the cloud platform to the CTP server in parallel. As can be observed from Figure 5, the bridge library takes an average of 12 milliseconds to communicate data from the monitoring framework to the CTP server.

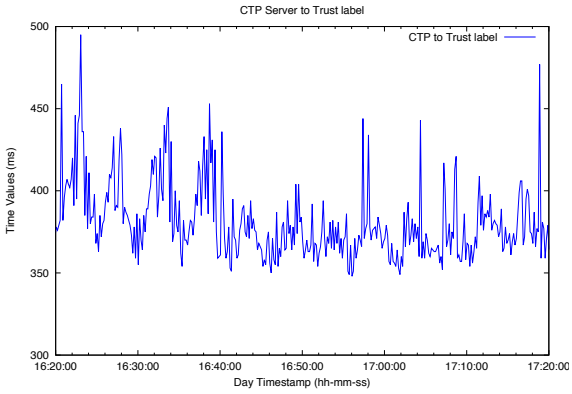

Figure 6: CTP-to-Trust Label Performance

Figure 6 presents the time for communicating the whole information gathered from the CTP server to the Trust Label interface. This data is bundled together and sent in one communication. It can be observed that it takes an average of 380 milliseconds in this case. This is also an impressive performance for the bridge library.

Combining the results of these two benchmarks indicates quick communication and therefore reflects the efficiency of the bridge library. We could not effectively evaluate the overall end-to-end communication time since the CTP server is being used as a black box. Besides, the communication between the CTP and Trust Label components is a pull operation at periodic intervals. This is why we opted to focus on the implemented bridge library.

5.4 Discussion/Recommendation

This end-to-end integration demonstration for communicating cloud data through CTP and Trust label system provides a practical reference. This can be very beneficial to cloud and service providers including brokers who are looking for a means of establishing and maintaining evidence-based trust with their consumers.

What we presented in this paper is simply a means of adopting and integrating CTP. There are other possible ways of using these software components. In fact, in a production environment, it is likely that the cloud monitoring framework would be more closely integrated with the CTP server; indeed they could be so closely coupled that there would be no REST API between them. Similarly the functionality of the CTP Bridge and Trust Label could be merged. An ideal architecture therefore would have Trust Label use the CTP REST API to communicate directly with the cloud provider. In this scenario, the CTP functionality and monitoring need not be separate services.

6 RELATED WORK

The Cloud Trust Protocol is at an early stage of adoption and validation by both industry and academia. Some recently completed EU funded research projects have adopted this protocol to address different challenges. The FP7 SPECS (Secure Provisioning of cloud Services based on Service Level Agreement (SLA) Management) project (SPECS project, 2016) is adopting CTP to achieve transparent communication between cloud providers and cloud users. The goal of this project is to provide comprehensible and enforceable security assurance by cloud providers to their users. The project takes an SLA-based approach for negotiating the cloud security parameters in the SLA and includes monitoring of a limited number of metrics to enable their enforcement. This project is similar to ours; however, we consider a broader range of performance metrics that can facilitate consumer trust in cloud services.

A4cloud (Cloud Accountability project, 2016) focuses on accountability for clouds. One of the areas this project investigates is means to enable cloud providers to give their users transparent control over how their data is being used or handled in clouds. The

goal is to boost user confidence and to ensure compliance. The project is using CTP in addressing these challenges. This project differs from ours because it focuses on data location management. Our trust label system includes data location management, alongside other trustworthiness information, to enable evidence based trust of cloud services.

The CUMULUS (Certification Infrastructure for Multi-Layer Cloud Services) project (CUMULUS project, 2016) aims to provide an automated process to certify security properties of cloud services at the different layers of cloud stack. The goal is to overcome the manual, costly and lengthy inspections and auditing schemes for certification purposes. Such schemes do not support dynamic change in the structure, deployment or configuration of resources underlying cloud provisioning. This project is applying CTP towards their goal of achieving compliance in certifying cloud services. The objective of this project is to enable user trust through certification. This approach is similar to ours, however, we aim to communicate trustworthiness information directly to consumers to enable evidence-based trust instead of establishing certification seals.

The Security Content Automation Protocol (SCAP) (NIST, 2016) is also considered a cloud trust interoperability protocol (Ron Knode, 2010). SCAP (pronounced ess-cap) consists of a suite of specifications for standardising the format and nomenclature by which software flaws and security configuration information are communicated, both to machines and humans. The CTP developers have however incorporated several parts of the SCAP specification into the development of CTP and so the two protocols can be seen as complementary rather than as competitors. In relation to our trust label system that provides broad metrics to facilitate consumer trust in cloud services, these two protocols are considered as extension possibilities as we have demonstrated in this paper.

To the best of our knowledge, none of the existing literature has reported on practical integration of CTP to facilitate consumer trust in cloud services and thereby increase the adoption of cloud computing.

7 CONCLUSION

In this paper, we presented a practical integration of Cloud Trust Protocol (CTP) with a trust label system designed to facilitate both persistent and dynamic trust building in both cloud services and cloud service providers. The trust label system includes a novel monitoring framework to supervise service provisioning and to gather performance metrics data.

We described the trust label system and CTP deployments while highlighting the challenges to be addressed. Our focus was mainly on the integration architecture that details the components and the realisation of the end-to-end communication of trustworthiness information between cloud service providers and consumers. The design and implementation of this architecture including a Bridge library were presented.

For the evaluation, a practical use case scenario was introduced. The use case describes an end-to-end consumer service provisioning. It uses a sample SLA document to explain how the quality of service elements of the consumer service were monitored and communicated through the integrated system. We demonstrated the correct working of the system and presented some efficiency results.

In the future, we aim to carry out real validation of the system by end-users to gather empirical evidence about the effect of the system on their trust of cloud services. This will inform on the novelty of our approach. We intend to publish the system code under open source licence to open up its usage by others and thereby buttress our vision of increasing cloud service adoption.

ACKNOWLEDGEMENTS

The research work described in this paper was supported by the Irish Centre for Cloud Computing and Commerce, an Irish national Technology Centre funded by Enterprise Ireland and the Irish Industrial Development Authority.

REFERENCES

Anderson, S. W., Christ, M. H., Dekker, H. C., and Sedatole, K. L. (2014). The use of management controls to mitigate risk in strategic alliances: Field and survey evidence. *Journal of Management Accounting Research*, 26(1):1–32.

Bente, G., Baptist, O., and Leuschner, H. (2012). To buy or not to buy: Influence of seller photos and reputation on buyer trust and purchase behavior. *Int. J. Hum.-Comput. Stud.*, 70(1):1–13.

Bradshaw, D., Folco, G., Cattaneo, G., and Kolding, M. (2012). Quantitative estimates of the demand for cloud computing in europe and the likely barriers to up-take. In *IDC Analize de Future*.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computing Systems*, 25(6):599–616.

Cloud Accountability project (2016). A4Cloud. http://www.a4cloud.eu/objectives Accessed on 22/08/2016.

Cloud Security Alliance (2015a). Cloud trust protocol. https://cloudsecurityalliance.org/group/cloudtrust-protocol/ Accessed on 22/08/2016.

Cloud Security Alliance (2015b). Cloud trust protocol daemon prototype. https://github.com/cloudsecurityalliance/ctpd Accessed on 07/09/2016.

CUMULUS project (2016). Certification infrastructure for multi-layer cloud services. http://www.cumulus-project.eu/ Accessed on 22/08/2016.

Das, T. and Teng, B. S. (1996). Risk types and inter-firm alliance structures. *Journal of management studies*, 33(6):827–843.

Deitz, G. and DenHartog, D. (2006). Measuring trust inside organizations. *JPersonnel Review*, 35(5):557–588.

Emeakaroha, V., Fatema, K., Vanderwerff, L., Healy, P., Lynn, T., and Morrison, J. (2016). A trust label system for communicating trust in cloud services. *IEEE Transactions on Services Computing*, PP(99):1–1.

Gefen, D. (2003). TAM or just plain habit: A look at experienced online shoppers. *Journal of End User Computing*, 15(3):1–13.

Gens, F. (2014). Worldwide and regional public cloud it services 2014-2018 forecast. https://www.idc.com/research/viewtoc.jsp?containerId=251730 Accessed on 20/10/2016. IDC Market Analysis.

Hogan, M., Liu, F., Sokol, A., and Tong, J. (2011). Nist cloud computing standards roadmap. *NIST Special Publication*, 35.

Hwang, K. and Li, D. (2010). Trusted cloud computing with secure resources and data coloring. *IEEE Internet Computing*, 14(5):14 –22.

Kim, W. (2009). Cloud computing: Today and tomorrow. *Journal of object technology*, 8:65 – 72.

Leimbach, T., Hallinan, D., Bachlechner, D., Weber, A., Jaglo, M., Hennen, L., Nielsen, R., Nentwich, M., Strau, S., Lynn, T., and Hunt, G. (2014). Potential and impacts of cloud computing services and social network websites. Science and Technology Options Assessment (STOA) http://www.europarl.europa.eu/RegData/etudes/etudes/join/2014/513546/IPOL-JOIN_ET(2014)513546_EN.pdf Accessed on 15/08/2016.

Lewicki, R. and Bunker, B. (1996). Developing and maintaining trust in work relationships. Technical report, Trust in organizations: Frontiers of theory and research. edited by Roderick M. Kramer, Tom R. Tyler.

Low, C., Chen, Y., and Wu, M. (2011). Understanding the determinants of cloud computing adoption. *Industrial management and data systems*, 111:1006 – 1023.

Lynn, T., Healy, P., McClatchey, R., Morrison, J., Pahl, C., and Lee, B. (2013). The case for cloud service trustmarks and assurance-as-a-service. In *Intl. Conference on Cloud Computing and Services Science Closer*, pages 8–10.

Lynn, T., van der Werff, L., Hunt, G., and Healy, P. (2016). Development of a cloud trust label: A delphi approach. *Journal of Computer Information System*, 56(3):185–193.

Masevic, I., van der Werff, L., Emeakaroha, V., Morrison, J., and Lynn, T. (2016). Validating a cloud trust label: Influencing consumer trust. In *Cloud British Academy of Management Conference*.

Mcknight, D. H., Carter, M., Thatcher, J. B., and Clay, P. F. (2011). Trust in a specific technology: An investigation of its components and measures. *ACM Trans. Manage. Inf. Syst.*, 2(2):12:1–12:25.

McKnight, D. H. and Chervany, N. L. (2001). What trust means in e-commerce customer relationships: An interdisciplinary conceptual typology. *Int. J. Electron. Commerce*, 6(2):35–59.

NIST (2016). Security content automation protocol. https://scap.nist.gov/revision/1.2/index.html Accessed on 22/08/2016.

Pavlou, P. A. and Gefen, D. (2004). Building effective online marketplaces with institution-based trust. *Info. Sys. Research*, 15(1):37–59.

Pearson, S. (2013). Privacy, security and trust in cloud computing. In *Computer Communications and Networks*, pages 3 – 42.

Pearson, S. and Benameur, A. (2010). Privacy, security and trust issues arising from cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 693–702.

Ron Knode (2010). Cloud trust 2.0. https://scap.nist.gov/events/2010/itsac/presentations/day2/Security_Automation_for_Cloud_Computing-CloudTrust_2.0.pdf Accessed on 22/08/2016.

Rousseau, D. M. and Sitkin, S. B. (1998). Not so different after all: A cross-discipline view of trust. *Academy of Management Review*, 23(3):393–404.

SPECS project (2016). Secure provisioning of cloud service based on SLA management. http://www.specs-project.eu/project/description/ Accessed on 22/08/2016.

Subashini, S. and Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34:1 – 11.

Zissis, D. and Lekkas, D. (2012). Addressing cloud computing security issues. *Future Generation computer systems*, 28(3):583 – 592.