Backup Monitoring in Large Heterogeneous Business Cloud Computing Environments
A Framework for the Development of a Reliable Automated Process

Krassie Petrova¹ and Anne Abbott²
¹Department of Information Technology and Software Engineering
²Knowledge Engineering and Discovery Research Institute
Auckland University of Technology, Auckland, New Zealand

Abstract - With the growing popularity of cloud computing technologies, the expectations towards cloud service providers (CSPs) increase not only in terms of functionality, cost, and service delivery speed, but also for the security of customer data stored remotely. Backup and restore processes are an important part of data security management, as they ensure data integrity and availability. CSPs can be certified for their compliance with established industry standards for data security. However, there is a lack of guidance on how to design internal processes in order to achieve compliance and improve data security management. The study addresses the gap by building a proof of concept software artefact and proposing a conceptual framework that may be used to guide the development of a reliable backup monitoring process and related software, particularly in large heterogeneous system environments.

Keywords: Cloud computing, cloud services, data availability, backup monitoring, information security, design science research, heterogeneous systems, CSP

1 Introduction and background

Cloud computing is a way of dynamically delivering computing power, storage and applications based on customers’ current needs. Three different service models define the scope of the services provided, namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The service can be deployed within one of the four different deployment models known as public, private, community and hybrid cloud. Combining the service models with the deployment models creates a range of service scenarios and risk profiles [1].

Cloud service providers (CSPs) need to guarantee the security of the data they host, and comply with customers’ data backup requirements [2, 3]. As a means of increasing their trustworthiness and reliability, CSPs can be certified for compliance with established data security industry standards [4]. Therefore, it becomes important to develop tools and techniques that assist CSPs in conducting security self-evaluation, as an essential step towards gaining customer trust by meeting cloud security audit obligations [3, 5]. Recently developed standards for secure cloud computing services such as ISO/IEC 27002 and ISO/IEC 27017 also highlight the need for tools that allow backup monitoring and gathering data for the purposes of backup audits [6].

While it has been suggested earlier that such security management solutions should be “inherent in the cloud architecture” [7] there is still a lack of guidance on how to design CSP’s internal processes in order to ensure protection of the entrusted data and compliance with audit requirements. The research presented in this paper addresses this gap by proposing a conceptual framework for the development of backup monitoring software for large heterogeneous cloud computing environments. The main research question of the study can be formulated as: “How to monitor effectively and efficiently backups and their properties for a large number of systems in a heterogeneous business cloud computing environment?”

The study was conducted in the context of ORGX, a very large CSP whose portfolio has diversified through a series of acquisitions. It includes a number of cloud applications used mostly by small and medium enterprises, mobile cloud solutions connected to customers’ backend systems that serve as a lightweight mobile device interface for system access “on the go”, and an in-memory database that facilitates big data analytics. The CSP had identified a need for a comprehensive software system that would allow them to cope with the increasing heterogeneity of their systems and monitor efficiently customers’ data backups. The latter had become a challenge while attempting to unify procedures and consolidate cloud application management.

The rest of the paper is organized as follows: The next two sections introduce relevant key concepts and definitions and present the study’s methodology. The outcomes of the research are described and discussed in the section following. The conclusion summarizes the research and its contributions, highlights its implications and addresses some of the pertinent limitations.
2 Key concepts and definitions

2.1 Backup technologies

“Backup” refers to a series of copies of a particular set of data made at a particular point in time that is created at regular intervals and stored independently from the original data [8, 9]. As early as 1992, Rangachari defined a set of general criteria that a good backup process should meet in order to be heterogeneous and scalable [10]. These criteria (easy addition of new systems, no limitation of storage space, centralized administration, and reasonable backup and restore timeframes) are still relevant despite the shift from magnetic tape backup to the now standard snapshot technology [8]. During the snapshot process copied data are first stored locally in order to minimize backup time, and then transferred to the actual backup media. The snapshot approach does not affect negatively system services availability and also supports easy recovery [8, 9, 11].

As companies entrust their vital data with a CSP, it is essential that they consider data security. This includes data encryption, access control, and compliance with relevant policies such as Statement on Auditing Standards (SAS70) and Sarbanes Oxley Act (SOS) that have to be met in accounting processes. In terms of functionality, the CSP should offer cross-platform support, easy scalability, and guaranteed availability of their systems [12]. Furthermore, it is important that CSPs are trustworthy in terms of quality and economic stability, in order to avoid data being lost due to mishandling or insolvency [13].

2.2 Heterogeneity

Heterogeneity, or diversity of systems, describes the combination of dissimilar artefacts in a specified setting to achieve a shared goal. In particular, data heterogeneity refers to data that may originate from two or more sources and may be presented in a number of formats [14]. Federated database management systems (FDBMS) are well equipped to deal with heterogeneous data. The main advantage of FDBMS (compared to conventional distributed database management systems) is that the individual databases remain largely autonomous. This is achieved by adding a logical layer to each of the databases that create the federation [15]. The approach is applicable to a number of scenarios such as the heterogeneous data center proposed in [16].

2.3 Secure cloud services

Virtualization is one of the most important concepts for the technical architecture of cloud computing [17]. In a virtualized environment, the physical (i.e., hardware), and computational (i.e., software) layers are decoupled and the resources are partitioned and shared amongst users [18]. The encapsulation of a client’s applications and data in a virtual machine (VM) enables the client to configure “their” VM without affecting other VMs. In particular, encapsulation fosters the swift creation of system copies, which can then be used to set up a clone of the VM for another client, or as a snapshot backup copy. Furthermore, establishing a multitenancy environment in which several tenants’ data are held physically together but virtually apart from each other is a common practice [19]. Performance monitoring is a critical factor for monitoring the complex cloud infrastructure, however, there seems to be a lack of software available to support administrators [20].

Due to its special architecture, cloud computing implicates many risks, such as unauthorized (physical and virtual) access to data, unknown storage location, data ownership issues, confidentiality breaches, network outage, or hacking attacks motivated by the large number of possible victims [18, 21-23]. Therefore, significant attention has been paid to cloud customers’ need to ensure that data they put in the cloud are safe [18, 23, 24]. To this end, several cloud service-specific standards and frameworks exist that can be used to assess a CSP’s security measures, e.g., ISO 17788, ISO 17789, ISO 19086-1 [25-27]. For the purposes of certification (as a means of demonstrating a CSP’s compliance with recommended standards), providers can use the Service Organization Controls (SOC) framework developed by the American Institute of Certified Public Accountants (AICPA). More specifically, SOC 2 and SOC 3 assess compliance with the three main security objectives confidentiality, integrity, and availability [28]. More current developments include ISO 27017 and ISO 27018 which are both cloud-specific and certifiable [6, 29].

In order to mitigate the risks and create a secure system, CSPs may implement frameworks that adhere to the standards mentioned above [21, 30]. As part of the process, an external certification agency should be invited to analyze and evaluate the CSP’s risk management process and the efficacy of the proposed countermeasures [22]. Such audits should be held on a regular basis to reflect the frequently occurring changes of infrastructure and instances, and they should take cloud-specific characteristics into account [21-23, 31].

3 Research approach

Addressing the research question in the context of ORGX required to identify the software design requirements, to build a software artefact that simplified audit data collection, and to propose a method for the development of a robust backup monitoring process and associated software. Overall, the study followed a design science research approach. It produced both a software solution, and a method for developing such solutions in similar contexts [32].

3.1 Research process

The research process was modelled on the design science research methodology (DSRM) developed by Peffers,
Tuunanen, Rothenberger and Chatterjee [33]. The original DSRM sequence of process stages was adapted to include conducting a pilot pre-study as a means of testing the feasibility of the concept, and to incorporate a full software development lifecycle in the design and development stage. As the research attempted to solve an already identified problem, the research process commenced at the second stage of the original sequence (Figure 1).

The easy-to-use tool that was created at the pre-study stage supported collecting and presenting backup monitoring data for one of the ORGX’s cloud services. Its purpose was to pilot providing reliable audit evidence and improving the efficiency of the process as a means of testing the feasibility of the concept. The pilot confirmed that: (i) the time needed to produce accurate, reliable and comprehensive reports covering the scheduled backups of private and shared productive systems was significantly reduced, and (ii) the inbuilt historicization (i.e., storing the states of each system and its backup each day) provided audit evidence that was reliable and reproducible.

A systematic identification of requirements was carried out at the design and development stage in order to provide a well-informed foundation for the design of the proof of concept software artefact. The latter was implemented and integrated into the existing cloud management system using available hardware libraries and interfaces, and then tested against the requirements. Once the development was finished, the application was made available to the research participants for demonstration and evaluation purposes. After observing how it addressed backup monitoring, research participants provided evaluative feedback that was used to support conclusions about how well the artefact met the research objectives and answered the research question.

The final stage involved a methodological generalization of the software development process to a method applicable outside the context of ORGX, in the form of a conceptual framework for the design and development of backup monitoring software. The framework refers to the software artefact as an instantiation of the method [32].

3.2 Research participants

The software design requirements were derived from the views and opinions of ten experts whose positions in ORGX and areas of responsibility are summarized in Table 1. The research participants were interviewed (face-to-face or by videoconferencing) following a semi-structured approach. Most interviews were individual, except for interviews P5* and P8* which refer to group interviews (two participants in each group).

Table 1. Research participants

<table>
<thead>
<tr>
<th>ID</th>
<th>Position and Relevant Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Cloud Security Officer: Ensuring that operational processes abide by audit-relevant standards</td>
</tr>
<tr>
<td>P2</td>
<td>Senior Support Engineer: Setting up cloud operations platform services and monitoring</td>
</tr>
<tr>
<td>P3</td>
<td>Technician: Ensuring the backup process runs according to audit standards</td>
</tr>
<tr>
<td>P4</td>
<td>Software Developer: Developing new functionality in the area of backup</td>
</tr>
<tr>
<td>P5*</td>
<td>Senior Support Engineer: Ensuring audit compliance of PaaS systems</td>
</tr>
<tr>
<td>P6</td>
<td>General Manager: Creating reports using different tools</td>
</tr>
<tr>
<td>P7</td>
<td>Principal Consultant (IT Business Services): Overall cloud compliance coordination</td>
</tr>
<tr>
<td>P8*</td>
<td>Senior Consultant (Global IT Backup Management): Ensuring that the backup and restore processes are running properly</td>
</tr>
</tbody>
</table>

4 Results and discussion

4.1 Requirements identification

The interview analysis resulted in 36 requirements to be implemented in the software artefact. On these, an interdependency analysis was performed that resulted in multiple visual representations of requirement interconnections. The requirements were first categorized as belonging to one of the four categories that emerged (i.e., backend data collection, inner workings and data processing, technical aspects, and visualization) and then put in relation to each other within their category. A significant finding was the functionality grouping of requirements that became evident. Figure 2 shows a simplified version of the functionality diagram, with the category “Inner workings and data processing” expanded to include the underlying requirements.
and relationships between them; the bracketed numbers indicating the number of requirements in each category.

![Diagram](image)

Figure 2. Requirement interconnections

The categorization was informed by anticipated user interaction. For example, the requirement specifying that, if there were several suitable backups defined for one system, it should not matter which one of them was completed successfully as long as at least one captured all of the system’s data, was categorized as a backend requirement, because the user would usually not need this information. The requirement to include a functionality allowing user-defined, customizable reports was identified as a core requirement in the category "Inner workings and data processing", and a prerequisite to 13 other requirements in the same category. The functional groupings informed the design of the software artefact and identified closely related functionalities that needed to be implemented in the same software component. For example, the requirement to ensure backup monitoring data reliability by disallowing manual data entry and alteration (i.e., protecting data integrity), and the requirement to store data permanently and disallow deletion (i.e., providing an audit trail by ensuring historicization) were addressed in the database layout, and by restricting access to data manipulation to the automated data collection script.

Due to the considerable architectural heterogeneity of the cloud application management systems, it was not feasible to cover the backup processes of the whole range of cloud services offered. Rather, the software artefact was implemented across four different departments and used to inform the development of a conceptual framework that could guide the implementation in the remaining technical infrastructures, or in a consolidated environment.

### 4.2 Software design and development

Due to the characteristics of the specific implementation environment, five low priority and three medium priority requirements were considered outside of the design and implementation scope, for example the medium priority requirement to provide details explaining reason(s) for backup failure. It was determined that the error messages generated during backup were not detailed enough to allow a thorough analysis and address the root cause of the problem. One high priority requirement (to provide a list of customers affected by a failed backup) could not be satisfied due to insufficient functional data. Another high priority requirement (monitoring backup file transfer form primary to secondary storage) was reclassified as of medium priority and also dropped due to technical limitations of the underlying infrastructure. Finally, the medium priority requirement to automatically produce and send out reports via email was included in the design but not implemented due to time constraints. Overall, the software developed at this stage met functionally 25 of the 36 requirements identified, including eight high priority requirements.

A three-tier architecture (data, logic, and presentation layers) underpinned the design. Information about systems and their backups (including the number of systems scheduled to get a backup, and failed backups per day) is stored in an economical yet accessible way providing a common data schema. The logic layer provides the means to collect, analyze and pre-process backup data. An automated script runs daily in the background and collects the data needed to calculate backup success rates. Data are analyzed and stored in appropriately structured tables.

The display functions of the user interface (presentation layer) allow the user to interact with the backup data by requesting information from the backup monitoring application. The stored pre-processed backup data are displayed according to the selected timeframe and application. The significant system heterogeneity and size caused by the presence of a very large number of cloud applications with different characteristics presented a major challenge. The issues were addressed in several ways. First, to enable creating backup reports, each system was represented by a unique entity to which a backup object that stores all information about the system's backup was connected. Second, configurable queries enabled users to specify a set of monitored systems customized to their needs. Third, the backup process data storage requirements were minimized by saving the go-live and go-offline dates (rather than creating an entry for every day) for systems scheduled for back up. Data duplication was avoided by using the query ID as a foreign key in the rest of the tables.

The two audit criteria used by ORGX to monitor the backup process were the percentage of successful backups and the amount of systems that did not have a successful backup for three or more days. Other companies may use different monitoring criteria and schedules, depending on their policies and customer contracts. As cloud computing becomes more prevalent, the number of cloud applications will increase rapidly, calling for management solutions that are easily adaptable to changing requirements and circumstances.

### 4.3 A framework for the development of backup monitoring software

The proposed conceptual framework for developing backup monitoring software consists of three main stages...
which should be executed in order, with frequent and effective communication across the stages creating a feedback cycle throughout the whole implementation project (Figure 3). First, a comprehensive and detailed analysis should be performed to get an initial overview of the processes involved in creating and monitoring cloud customer data, and the respective compliance and reporting requirements. Only if these requirements are fully understood, will the outcome of deploying the framework be beneficial. It is also important, especially in heterogeneous environments, to identify the responsible employees and obtain, preserve and update on a regular basis their details for the purposes of communication and issue resolution.

4.4 Backup monitoring software architecture

It is recommended to follow a top-down approach and build a three-tier architecture containing a presentation, a logical, and a data layer. Presentation layer decisions are made with the help of the business experts who are the future users of the software. The reporting criteria are defined so that design of the solution can follow them. The framework is flexible and allows to create a company-specific backup monitoring software application.

The logical layer supports backup process data retrieval and pre-processing, such as calculating backup performance measures and providing further detail upon user request. In order to ensure that no backup cycle is missed (and no backup failure is overlooked), the data collection should be performed from a system-perspective, i.e., first retrieving a list of systems that need a backup on a particular day as specified by the process experts, and later checking if all of required backups were successful.

Finally, the logical layer should also accommodate flexible input of user-defined groups of systems. This is especially important for addressing the issue of heterogeneity. For example, if the underlying system repository has a search feature, it can be employed to gather data about all systems belonging to a particular group, by defining a search query. Groups of systems should be defined by sets of shared characteristics. However, it must be ensured that this combination of characteristics is accurate enough to allow the correct retrieval of the systems needed for a specific report.

An optimized data storage model has to be created which forms the bottom tier of the architecture. One of the challenges that could be faced by a CSP implementing the framework is the large number of systems and backups that have to be managed. Therefore, special attention needs to be paid to the layout of the database tables in terms of storage optimization and fast retrieval of information. The writing of data does not necessarily have to be optimized for time efficiency as it runs automatically in the background. Therefore, the process of filling the database tables can and should involve a reorganization of the input information. The aim is to store it in a format which supports the efficient provision of the data identified as “most important” by the business experts (at the analysis stage). The final task of the development stage involves testing the functionality of the software against the business experts’ requirements, and verifying the performance indicators and other calculated data for correctness and accuracy.

5 Conclusion

As indicated earlier, data backup security is one of the architectural security challenges facing cloud computing services [23]. Cloud service users need an assurance that CSPs’ data backup processes meet baseline expectations and
audit requirements [34]. Normally, the quality and the effectiveness of a CSP’s internal data security controls are attested to by an audit process [22]. Current research has highlighted the importance of auditing CSPs’ ability to protect data availability and integrity and comply with security standards’ requirements [35], and more specifically, the importance of CSP performance monitoring as a means of an ongoing validation of the CSPs trustworthiness [36].

Proposed cloud security auditing frameworks provide a formalized approach towards gathering evidence that CSPs follow secure processes including data storage and retrieval [37, 38]. While these studies consider the issue of cloud data security predominantly from the perspective of a cloud service customer who needs to be assured that the services received comply with security standards and negotiated service parameters, the study presented in this paper addresses the data security issue from a service providers’ perspective. More specifically, the research focuses on creating evidence demonstrating a CSP’s compliance with data backup audit requirements.

The study contributes to the body of knowledge in the area of cloud security auditing by proposing a conceptual framework for the design and development of software backup monitoring applications for large heterogeneous system landscapes. The framework aims to support CSPs in assuring customer data availability and integrity which is recognized as a critical issue in business cloud computing [23, 35, 36]. It is independent of technical constraints and can be applied in a range of contexts and environments, with adjustments specific to the needs of the respective CSP.

Study limitations such as the relatively low number of research participants and the technical constraints of the specific implementation environment may have affected the development of the software artefact, but their impact on the proposed framework and practitioner guidelines is relatively low. A CSP following the stages of the framework should be able to set up a functioning backup monitoring and reporting system that is robust, scalable, and flexible. Directions for further research include deploying and validating the framework in different organizational contexts, and identifying a set of software design requirements for the development of backup monitoring and reporting systems that may be applicable in multiple CSP contexts.

6 References


[15] A. P. Sheth and J. A. Larson, "Federated database systems for managing distributed, heterogeneous, and


