

A Dashboard for System Trustworthiness Properties Evaluation

Diego Camargo, Felipe Nunes Gaia, Tania Basso, Regina Moraes

¹University of Campinas - UNICAMP - Campinas, Brazil

{kmargod,felipegaia.comp}@gmail.com, {tbasso@cotil, regina@ft}.unicamp.br

Abstract. *Understanding the trustworthiness of a cloud system is a difficult task, because it encompasses a large diversity of properties such as security, privacy, performance, among others. Evaluating and improving the system regarding trustworthiness require the analysis of huge data considering their historical status. The goal of this paper is to present a dashboard developed for the visualization of these properties, the relationship among them and their relevance in the composition of a trustworthiness score over usage time.*

Resumo. *Entender a confiança de um sistema na nuvem é uma tarefa difícil, pois abrange uma grande diversidade de propriedades, tais como, segurança, privacidade, desempenho, entre outras. Avaliar e melhorar o sistema em relação a confiança implica na análise de um número grande de dados históricos dessas propriedades. O objetivo deste trabalho é apresentar uma interface desenvolvida para a visualização dessas propriedades, a relação entre elas e a relevância de cada uma na composição de uma medida da confiança ao longo do tempo de uso do sistema.*

1. Introduction

Cloud computing enables access to services and global information availability. This paradigm allows the sharing of massive, heterogeneous, elastic resources among users. Increasingly number of services is published using cloud resources, allowing organizations to better serve their customers around the world [D. Miller 2017].

In more recent times, the popularization of the use of cloud environment is due to the availability of more robust resources that can be used under the pay-per-use policy [Mell and Grance 2011]. This technology improvement has allowed the massive use of cloud systems as well as the opportunity to handle large volumes of heterogeneous data (Big Data).

Besides dealing with Big Data, it is necessary to extract knowledge from these data to support organizational decisions, process or computational systems improvements. Big data analytics examines large amounts of data to uncover hidden patterns, correlations, market trends and other insights, helping organizations harness their data using it to identify new opportunities as well as refinement and improvements to be done on organizational processes and systems. Machine learning, data mining, natural language processing and statistics are some analytic techniques that can be used, in different ways, to extract prepared and blended data, providing analysis for the businesses [A. Verma 2018]. Although the importance of analytic techniques is indisputable, care must be taken with the results made available so as not to incur data leakage,

i.e., the unauthorized transmission of data, violating security and privacy information [Basso et al. 2016]. This could compromise the level of trust the user places on the system.

Trust, in cloud computing, can be defined as *“the accepted dependence of a component, on a set of properties (functional and/or non-functional) of another component, subsystem or system”*[Veríssimo et al. 2003]. To increase trust, cloud providers should explicitly communicate their capabilities to ensure important functional and non-functional requirements, and can be defined as the level in which a cloud service meets a set of those requirements, i.e., the worthiness of cloud services for being trusted. Trustworthiness can be understood as *“the measure in which a component, subsystem or system, meets a set of properties (functional and/or non-functional)”*[Veríssimo et al. 2003]. These properties can be security, privacy, fairness, fault tolerance, dependability, among others. All of them have other subattributes that can be combined in a multi-dimensional construct.

Thus, trustworthiness is an important property of the system to be considered for decision making. As important as a method for measuring system trustworthiness is a mechanism for presenting the results of these analyzes. It is important to show the trustworthiness attributes (security, privacy, dependability among others) and their subattributes scores. Moreover, it is necessary to allow analysts to configure the parameters that contribute to improve the scores in focus. A dashboard engine can provide an interface for presentation of the attributes scores on a computer display device and can further identify user interactions with the system view through the dashboard.

The goal of this work is to present a dashboard development and its preliminary evaluation. This dashboard was developed in the context of the ATMOSPHERE (Adaptive, Trustworthy, Manageable, Orchestrated, Secure, Privacy-assuring Hybrid, Ecosystem for Resilient Cloud Computing) project¹, a collaborative project between Europe and Brazil, whose main objective is to provide a solution for assessing the trustworthiness of cloud applications that handle large volumes of data.

The dashboard is based on a tree reference model, which can be further instantiated for each required trustworthiness attribute to calculate a global property score. This is done by walking through the model tree, from the leaves to its root, aggregating child scores. The instance of the model for a given attribute represents its quality model (as proposed by ISO/IEC 25000). In this work a privacy quality model is used to make a preliminary evaluation of the proposed dashboard. The assessment results show a good acceptance by the evaluators but important weaknesses related to the navigation were pointed out.

This paper is organized in six sections. After the first, the Introduction, Section 2 presents Background that is mandatory to understand the paper. Related Work is presented in Section 3 and Section 4 presents the proposed dashboard. The Section 5 presents some Discussions and Analysis. Conclusions and Future Works follow in Section 6.

¹<https://www.atmosphere-eubrazil.eu/>

2. Background

This section addresses, briefly, the issues that underpin this work. It discusses the need of dashboards, the use of quality models and presents the trustworthiness monitoring and assessment framework in which the dashboard is inserted.

2.1. Dashboard and Big Data

Large volumes of information are generated by several devices connected to the internet. The large amount of data that is generated at any moment can obscure the analysis processes and hinder decision making, compromising information maintenance, analysis time, specialized technical staff, among others. In this scenario, the concepts of Big Data and data mining arise. To present the large volume of information in an organized and concise manner, the use of dashboards for management purposes has been increasingly necessary, emerging as Business Intelligence (BI) tools. Using a dashboard, it is possible to display, in a practical, clear and objective way, the consolidated information from the crossing of the most relevant data from the systems. In the context of the ATMOSPHERE project, the dashboard is defined as the most viable tool to present the scores of trustworthiness properties as well as intermediate scores and attribute metrics composing these properties. To ease the necessary quick analysis, techniques to provide good usability must be used so that the dashboard contains only the necessary information, helping to retain a good understanding.

2.2. Reference and Quality Models

According to Bass et al. [Bass et al. 2003], a **reference model** is “*a division of functionality together with data flow between the pieces*”. **Quality model** is a reference model proposed in the ISO/IEC 25000 (SQuaRE) standard [ISO 2014] whose structure formalizes the interpretation of measures and the relationship among them. This way, it is possible to define how the cloud system properties measures should be aggregated and what procedures have to be used to homogenize their values, composing a trustworthiness score.

Figure 1 presents a generic Quality Model structure. The rectangles represent the properties that are important to characterize the system, i.e., the **attributes** that compose the score, starting in the root (*System*), going through composite attributes (*Attribute₁*, *Attribute₂*), reaching the leaf attributes (*Leaf Attribute_{1,1}*, *Leaf Attribute_{1,2}*, *Leaf Attribute_{2,1}*).

Leaf attributes represent metric definitions and their associated scores are based on the actual measures collected by monitoring probes in the cloud. They must be normalized to be evaluated against thresholds and ensure that operators aggregate values of same scale. This normalization process uses the maximum and minimum (*NormalMax* and *NormalMin*) values considered in the normalization function applied on leaf attributes. This means that the metric values received will be normalized in the range 0 up to 1, maintaining the proportion between measures. The **thresholds** (*Threshold Max* and *Threshold Min*) specify the maximum and minimum values used in the normalization function for specific attributes.

The values for each attribute are influenced by an adjustable element **weight** (W_1 , W_2 , $W_{1,1}$, $W_{1,2}$, $W_{2,1}$), which specifies a preference over one or more characteristics of

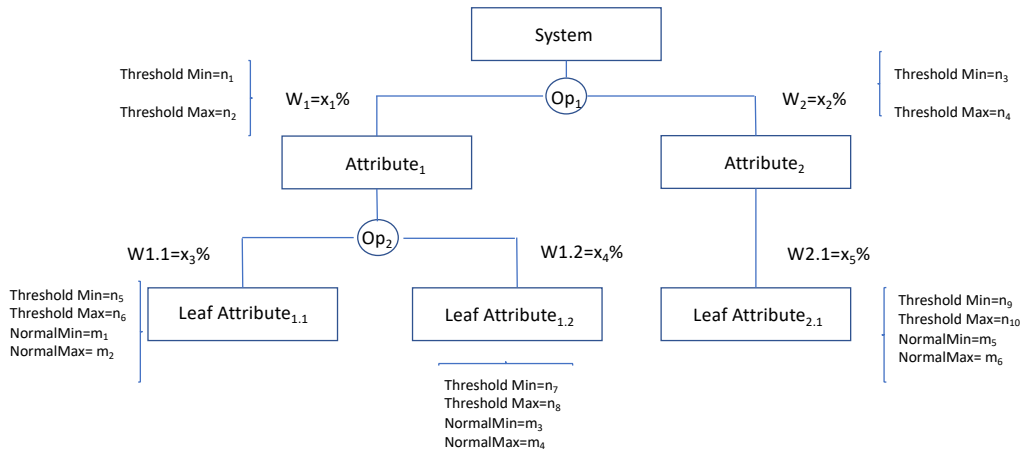


Figura 1. Example of Quality Model

the system, according to predefined requirements (e.g., in certain contexts memory usage might be more important than throughput).

The final score is computed using the aggregation of the values of the attributes, starting from the leaf-level to the root attributes, using the **operators** (Op_1 , Op_2), which describe the relation between them. Different types of operators may be used to define the conditions under which composite attributes are aggregated. For our context, we adopted the following ones: *neutrality* (represents the combination of simultaneous satisfaction requirements with replaceability capability); *simultaneity* (refers to that all requirements must be satisfied, i.e., a conjunction using the logical operator *and*); *replaceability* (used when one of the requirements has a higher priority replacing the remaining requirements, i.e., refers to a disjunction using a logical operator *or*).

Some works use quality models to characterize trustworthiness properties. For example, Basso et al. [Basso et al. 2019b] defined a quality model for representing and calculating data privacy scores. In the same direction, Casare et al. [Casare et al. 2020] defined a quality model to evaluate personal aspects that impact the sense of trust, which depends on the user experience when interacting with systems.

2.3. The ATMOSPHERE Monitoring and Assessment framework

The ATMOSPHERE solution proposed for monitoring and assessing trustworthiness in cloud platforms is based on the MAPE-K reference architecture [IBM Corporation 2006]. It was adopted due to its simplicity and extendable nature. Figure 2 presents this solution.

In Figure 2, the users or client applications submit queries to the database through the *Trustworthy Data Management Services (TDMS)* (Step 1). TDMS is a component of the ATMOSPHERE project similar to a database service in cloud systems dealing with mechanisms for data storage, access and management. When data stored in this component is retrieved they are filtering to comply with the configuration parameters and be available for the probe procedure consuming (Step 2).

Probes are responsible to collect information from the managed system (cloud resources and applications) under the MAPE-K architecture and send it to the *Trustworthiness Monitoring & Assessment (TMA) Framework*, specifically to the *Monitor*

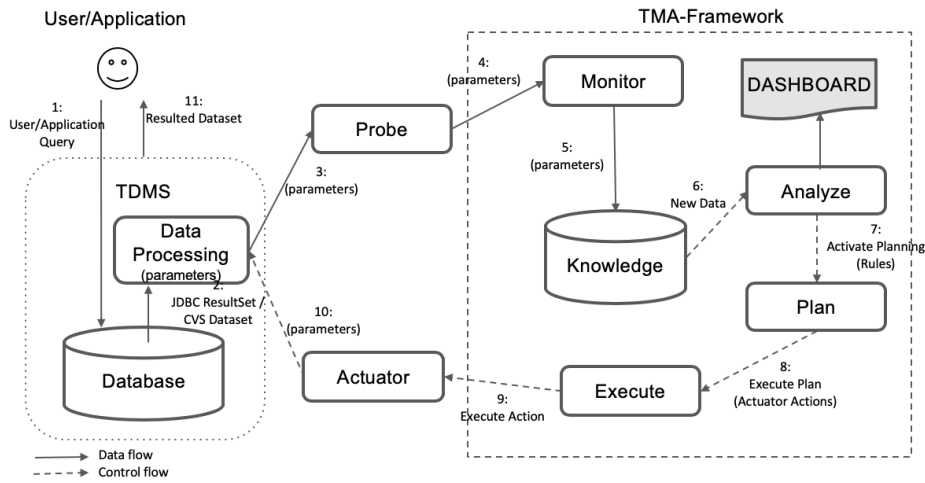


Figura 2. ATMOSPHERE trustworthiness monitoring and assessment platform (adapted from [Basso et al. 2019a])

component.

The *Monitor* is a component that receives the information sent by probes, and store them at the *Knowledge* repository (Step 5). The *Knowledge* repository stores all the trustworthiness-related information along with definitions of quality models, including their weights and thresholds. The *Analyze* component obtain the most recent measures stored at the *Knowledge* repository and re-calculate the scores (Step 6) based on the quality models. Since the ATMOSPHERE platform potentially generates large amounts of data to be analyzed, the use of dashboards for management purposes has been increasingly necessary. So, at this point, the **dashboard** we propose presents the scores of trustworthiness properties as well as intermediate scores from attribute metrics composing the properties. The goal is to help the analysis processes and decision making.

For every update on the scores, the *Plan* component is activated and is responsible for executing the verification of the predefined rules. For each rule that is not respected, one or more adaptations will be generated, which will be gathered in a set of adaptations (Step 7). These actions are sent to the *Execute* component (Step 8) that is responsible for runtime adaptation to achieve/maintain the desired levels of trustworthiness. The *Execute*, in turn, calls the *Actuator*, which invokes the proper operation to be performed in the corresponding resource (Step 9).

3. Related Work

Several works found in the literature address the importance of an efficient dashboard. For example, [Dyczkowski et al. 2014] discuss the challenge of applying Business Intelligence (BI) for projects dedicated to the management support of small and medium-sized companies. [Ferreira et al. 2016] presents a study on an intelligent dashboard to support decision making in Artificial Intelligence techniques in a logistics company.

Most of these works focus on the complexity of building efficient dashboards and assessing dashboard's usability. [Muriithi and Kotzé 2013] state that the adoption and

use of BI systems within the company remains low, despite the potential benefits that these systems bring. For greater adoption, the user needs to be able to interact with the application in such a way that decision making is not inhibited by an overly complex user interface. [Luzzardi 2003] state that when designing a dashboard, it is desirable that the greatest amount of information is presented in a single interface. The visualization technique must provide interactive resources that facilitate and assist the analysis of the data (for example, rotation, zoom or distortion techniques). [Smuts et al. 2015] argue that the use of interactive dashboards has become a popular technique for aiding users in BI analysis and data discovery. The BI tools usability, however, are not fully matured to a level where novice users can utilize their features efficiently and effectively, without assistance from IT experts.

Given the need for higher levels of usability on dashboards, some works address this issue. [Magdalena et al. 2019] defined a strategy for increasing the use of the BI dashboard in their company. The strategy involves the conduct of user testing and heuristic evaluation and had as result some recommendations that can be used to improve the BI dashboard. [Gesing et al. 2014] stated that there is a disappointment which domain scientists with scientific workflow systems and the way to convince them to start using workflows extensively is to hide technical aspects of workflows behind intuitive user interfaces. So, they proposed a dashboard with usability concerns to ease the use of workflows for scientists and other researchers. However, to the best of our knowledge, there is no dashboard for trustworthiness evaluation in cloud computing context, specially considering usability. This work aims to help filling this gap.

4. The Dashboard

As mentioned before, the dashboard is defined to present the scores of trustworthiness properties (and also intermediate scores from attributes composing the properties), as well as allow users to interact with the monitoring and assessment framework, i.e. to input the configuration values (e.g., weights, thresholds). Also, for this dashboard to be profitable, it must have good levels of usability. Thus, the designs of the dashboard need to accomplish the following main goals: (i) increasing efficiency while decreasing variability of the analysis process; (ii) meeting the needs of the ATMOSPHERE team allowing them to analyze information at runtime (including drilling into details); and, (iii) adjust the parameters and thresholds of the quality models to optimize the trustworthiness score along the time. Therefore, Section 4.1 lists the requirements of the dashboard while Section 4.2 describes the design and implementation decisions for achieving these objectives. A preliminary interface validation was performed and its detail is presented in Section 4.3.

4.1. Requirements to Design ATMOSPHERE Dashboard

This section presents the requirements that have been elicited with the help of the ATMOSPHERE project members. The developers had in mind from the beginning of the process the usability and information visualization heuristics to guide the discussions with the alignment to these precepts. There are three requirements in the point of view of the enterprise and sixteen technical requirements, which are presented hereafter.

Enterprise Point of View

[Req01] The dashboard shall be protected against security attacks, since its access will have exposure on the internet.

[Req02] Users shall view data analysis information related to historical data, that is, the scores evolution, so that they can analyze the quality of the environment over time.

[Req03] The dashboard user shall be professionals who work with technology, since the focus is to provide reliable measures for a system in the platform to follow improving its configuration.

Technical Point of View

[Req04] The dashboard shall be deployed in the cloud, since the current platforms use this technology.

[Req05] The dashboard shall use a fully virtualized environment, since virtualization allows users to better explore the hardware resources of the environment and facilitates the configuration by users.

[Req06] The dashboard shall use open-source technologies, since this facilitates the use of the environment where inserted by different users.

[Req07] The dashboard shall display collected measurements and calculated scores of the applications that use the platform, allowing an analysis of the quality of service delivered to the user.

[Req08] The dashboard shall display the collected measurements and calculated scores as charts, representing the quality model and its elements as a recursive tree.

[Req09] The dashboard shall be able to extract the information and to support large volumes of data efficiently, since the computational environment is geared towards the analysis of Big data.

[Req10] The dashboard shall follow usability concepts and visualization heuristics, since it is an interface between the user and the platform.

[Req11] The dashboard shall use a set of design tools in which, each tool can be deployed and executed independently of the others, so that the user has freedom of choice.

[Req12] The dashboard shall be modular and can be applied to the entire quality model or only its specific parts, so that users can choose the attributes they want to analyze.

[Req13] The dashboard as a tool to present information shall be supported by a monitoring infrastructure that collects reliable evidence from the environment.

[Req14] The dashboard shall receive and integrate data collected during the software execution, since it must present updated data whenever invoked.

[Req15] The dashboard shall provide adequate interfaces to allow the integration of tool sets, since this set can be configured by the user.

[Req16] The dashboard shall be able to operate with a variable number of collected points, database instances and set of tools available, since it is aimed at large volumes of data.

[Req17] The dashboard shall support the traceability of evolving requirements, so that the user can analyze the available configuration.

[Req18] The dashboard shall offer a sufficient level of flexibility to include different types of techniques that can be requested by the design methodology.

[Req19] The dashboard shall be applicable regardless of the programming language(s) used in the target software and hardware requirements.

These are the requirements used by the developers to design and implement the dashboard. The non-functional requirements are Req01, Req09, Req10, Req11, Req16, Req18 and Req19. The functional requirements are Req02, Req07, Req08, Req12 to Req15 and Req17. The requirements Req04 and Req05 refer to infrastructure specifications; the Req6 refers to technology specification and, finally, the Req03 refers to the target users of the dashboard. Next section presents the decisions at design and implementation time.

4.2. Design and Implementation Decisions and Tools

Taking into account the requirements presented in the previous section, some design and implementation decisions were made as well as some tool were chosen to support the implementation. The goal of this section is to present these design and implementation details. Some requirements compliance are indicated at some points in the text.

The ATMOSPHERE dashboard is based on Metabase tool², which aims at fostering the design and development of dashboards. After a search for available tools, we decided to use the Metabase tool, due to the reasons that follow. It is an open source business intelligence tool (complying with Req06 and Req11, ref. Section 4.1), written in Java and it can be run on a server allowing users to log in, share reports and dashboards. Also, Metabase provides an internal API for the interaction with a database to improve data access security (complying Req01). For instance, this tool can be used for deployment on any system that is running Docker (complying with Req04 and Req05) since a Metabase server is launch on port 3000 by default. These facilities have made us to choose Metabase to support the development of the project dashboard.

A limitation to be considered is that Metabase tool is not able to update information into the database. So, it is impossible to persist (or update) configuration data in the *Knowledge* module database (for example, to persist thresholds and weights), which is essential for the full use of the quality model. For this end, we developed a dynamic form using AngularJS³ that uses RESTful services implemented in Java to persist the configurations (complying with Req06).

To present the scores and metrics produced based on the defined quality model (referred to Section 2.2 for more details), the Metabase tool interacts with the *Knowledge* database (an open source MySQL database, complying with Req06) using an internal Metabase API implemented in Java. The access via API preserves security on data access (complying with Req01) and it is available when the user has an Admin role. Fig. 3 shows a schema of the implementation solution. The *Forms* obtains the quality model configuration data defined by the users and save (or update) them in the database through

²<https://metabase.com>

³<https://angularjs.org/>

the *Restful Webservices*. The *Graphs* presents the recovered data as required by the users (complying with Req08).

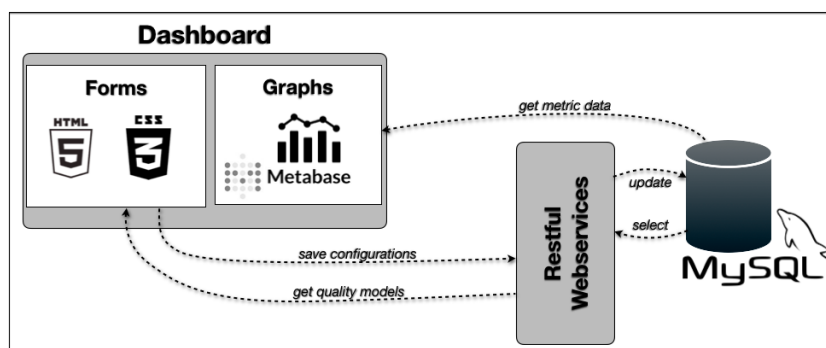


Figura 3. DASHBOARD Implementation

In order to navigate (drill down) through the scores and metrics in any level of the quality model we implemented a recursive algorithm (complying with Req08), in which the user can choose the property or attribute belonging to a quality model (complying with Req07), presenting the score calculation (for a property or composite attribute) or a metric value of a leaf attribute.

An example of the dashboard interface is presented in Fig. 4. We can see in the upper left corner of Fig. 4 the *Configuration Form* that allows the users to select the relevant attributes and configure weights, thresholds and periodicity of data collection. It should be used to prioritize attributes according to their relevance for each deployment context. Also, in the bottom left corner of Fig. 4, we can see the *Scores Tree View*, which allows the navigation through the scores and metrics (complying with Req07).

First, using the dashboard, the user chooses an attribute (or property) that belongs to the specific quality model. In the *Scores Tree View* the algorithm traverse all the quality model tree for that attribute (or property) and insert all the components in an array. With all the attributes inspected, for each attribute, the metrics and scores related to the period of interest were recovered from the Historical Data (complying with Req02). Then the operator defined in the quality model is applied and a view with this information is created. Metabase interacts via API with this view updating the dashboard according to a preset time and show the results (complying with Req14).

The calculation of the scores is made by the probes of each property and the dashboard presents the scores historical according to the time configuration. The user evaluates the scores through the configuration form and respective charts and can change the configuration (weights and thresholds) if necessary.

4.3. Interface Validation

A preliminary validation (a pilot test) was performed on the dashboard interface to understand its usability and to prepare a more complete validation with a larger number of users. Firstly, the validation methodology was design and submitted to the Research Ethics Committee (*Plataforma Brasil*) for the necessary authorization.

The validation methodology is composed by two questionnaires. The first

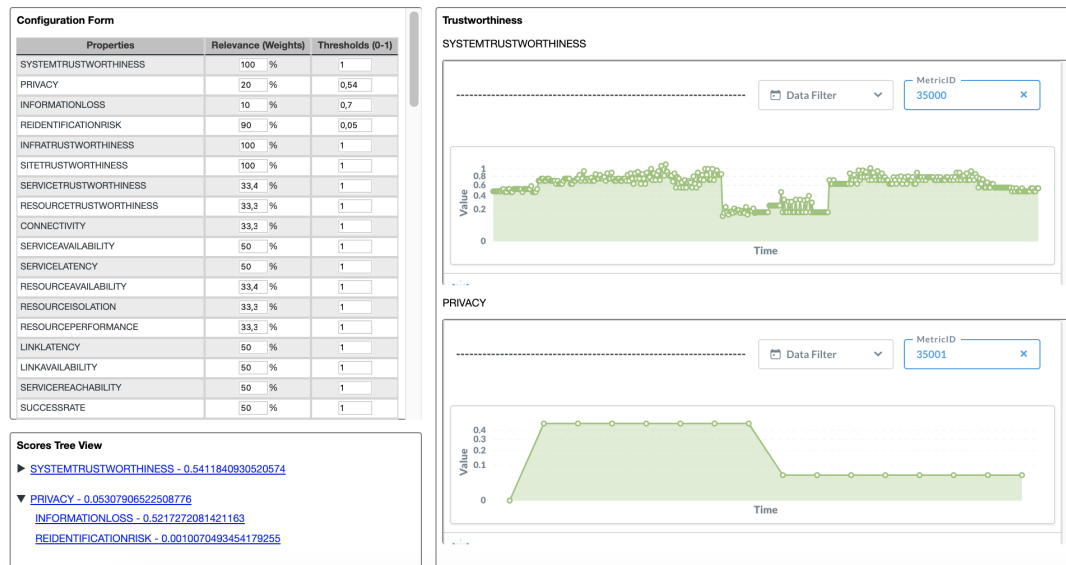


Figura 4. Dashed Interface

one is regarding the user profile, using six questions, mainly to understand their experience as a IT (Information Technology) professional as well as their experience in HCI (Human-Computer Interaction) domain. The second one is focused on the interface usability, composed by ten questions to evaluate the strengths and weaknesses of the dashboard interface. These questions were based on Nielsen’s usability [Nielsen and Mack 1994] and information visualization heuristics [Zuk et al. 2006]. Due to space restrictions we did not present the questionnaires here, but details can be found in our research group website [Faculty of Technology 2020]. Nevertheless, Table 1 shows a summary of the questions description from the second questionnaire.

At this point, only three users were elected to perform this preliminary evaluation. Even though they are few, Nielsen advocated that three professionals are enough for a heuristic evaluation [Nielsen 1994]. Moreover, the aim of this paper is to present the dashboard and refine the methodology to obtain better results in a wider evaluation in the future. The preliminary evaluation was performed during the middle of March 2020.

Besides the two questionnaires forms, it were provided two complementary documents to explain (i)the motivation of ATMOSPHERE project and the use of the dashboard and (ii) a background about the Quality Model as the dashboard uses its structure as the basis of the hierarchical representation of scores and metrics. More specifically, the evaluators used the privacy instance of the quality model to perform their assessments. The next section discusses the results and points out some suggestions for the evaluation process improvement.

5. Results Discussion and Analysis

This section presents the results of dashboard preliminary assessment (pilot test). To guarantee the required usability and information visualization level, an assessment with

the users is mandatory (complying with Req10). This assessment was based on data privacy quality model, presented in Fig. 5.

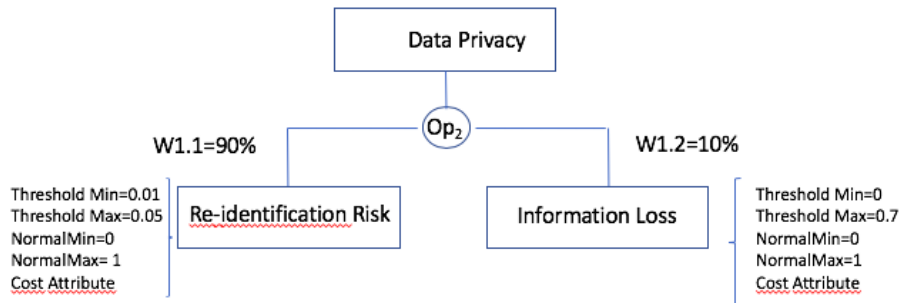


Figura 5. Privacy Quality Model Instance [Basso et al. 2019b]

Data privacy is one of the properties that makes up a trustworthiness system. In this work data privacy is obtained by anonymization techniques. Two main attributes are considered to compose data privacy: the re-identification risk (i.e., the probability of discovering an individual by matching anonymized data with publicly available information) and the information loss (i.e., the amount of information that can be obtained about the original values of variables in the input dataset). More details about the privacy quality model and its attributes (thresholds, weights, normalization values) can be found in the work of Basso et al. [Basso et al. 2019b].

The dashboard presents the configuration form and the attributes of each available tree (refer to the left side view in Fig. 4), being the privacy instance one of them.

As already mentioned in the previous section, two questionnaires were answered by each evaluator. The first one is about the evaluator's profiles regarding their professional skills. The results are: (i) their age, in average, is 40.66 years; (ii) 100% work in IT more than three years; (iii) 33% never worked before in the HCI domain and 67% never worked directly but frequently use material of this domain; (iv) 33% work sometimes with system's Front-End and system's requirements while 67% work with these matters frequently; (v) 67% work with system's testing sometimes while 33% use to work with systems testing frequently.

The second questionnaire is related to the dashboard interface and is composed by ten questions, whose results are presented in Table 1. The answers were based on the Likert scale, i.e., they offer a scale of points that include extremes - such as "totally agree" and "totally disagree". The categories of response are coded numerically, in percentage. The three last questions requires responses as "Yes" or "No".

Based on the results, the dashboard interface is nice, presents adequate volume, intuitive configuration form and results are ease to understand through the charts, with an average higher than 75% (it is also the minimum score among all evaluation in this group). On the other hand, analyzing the results, it is possible to realize that all evaluators have some problems to navigate in the Quality Model tree structure. One of them was really not able to realize how to navigate to open the structure and see the scores. So, navigation seems to be the most hard interface problem.

Tabela 1. Assessment Results

Questions'Description	Eval A	Eval B	Eval C	Average
Nice Interface	75%	75%	100%	83.33%
Adequate Volume	100%	75%	100%	91.67%
Intuitive Navigation	75%	50%	75%	66.67%
Intuitive Configuration Forms	75%	75%	75%	75%
Intuitive Tree Navigation	0%	75%	25%	33.33%
Intuitive Symbols	0%	50%	25%	25%
Results in Graphs Help?	75%	75%	100%	83.33%
Would you use the Dashboard?	N	Y	Y	N/A
Had you any problem in using the Dashboard	Y	N	Y	N/A
Do you want to suggest some improvements	Y	N	Y	N/A

Last two questions allow the evaluators to comment their difficulties using the dashboard and the comments corroborate with the analysis of the navigation problem. One of them commented that “The tree did not appear on her/his screen, making it difficult for him/her to find the tree”. A second comment is “The need to click the arrow to open the tree is not intuitive”. Both comments complain about the navigation through the tree, pointing the need for improvements in that specific part of the dashboard.

6. Conclusions and Future Work

This work presents a dashboard development and its preliminary evaluation. This dashboard was developed in the context of the ATMOSPHERE project, whose main objective is to provide a solution for assessing the trustworthiness of cloud applications that handle large volume of data.

ATMOSPHERE understood trustworthiness as a combination of several properties or characteristics and rely on a quality model, a tree-structured model, to calculate the main and intermediary scores. Then, the dashboard reproduces this structure in the interface, allowing the user to interact with the configuration parameters aiming to improve the system score.

The dashboard is very useful as part of the trustworthiness monitoring and assessment framework because it is updated in real time. Thus, system analysts are able to visualize data and act on improvements and decisions more quickly. Also, showing historical data allows the analysts to evaluate the improvements on the trustworthiness throughout the cloud application life cycle. This would help to identify past efficient strategies to be reused, i.e., that can be applied to other applications and contexts in order to increase the trustworthiness.

Considering the results of this preliminary dashboard assessment, the authors realized that the most important weakness is related to the navigation in the tree structure. So, they decided to rework and improve this interface detail before proceeding with the wider evaluation. It is important to highlight the importance of this preliminary user evaluation for the project success. This assessment allows improvement in the interface and questionnaires to proceed with a more consistent assessment involving now dozens of people.

As placed before, future work envisage a wider assessment including IT students,

professionals and researchers from inside and outside the ATMOSPHERE project. Based on these further results, and if necessary, a new release of the dashboard will be generated before deploying the dashboard into the platform.

Acknowledgment

This work has been partially supported by the project **ATMOSPHERE** (<https://www.atmosphere-eubrazil.eu/> - Horizon 2020 grant agreement No 777154 - MCTIC/RNP). Also, it has been partially supported by the **ADVANCE** project <http://advance-rise.eu/> - (Horizon 2020-MSCA-RISE grant agreement No 2018-823788).

Referências

- [A. Verma 2018] A. Verma (2018). Why is big data analytics so important? <https://www.whizlabs.com/blog/big-data-analytics-importance/>.
- [Bass et al. 2003] Bass, L., Clements, P., and Kazman, R. (2003). *Software architecture in practice*. Addison-Wesley Professional.
- [Basso et al. 2019a] Basso, T., de Oliveira Silva, H., Montecchi, L., de França, B. B. N., and de Oliveira Moraes, R. L. (2019a). Towards trustworthy cloud service selection: monitoring and assessing data privacy. In *Anais do XX Workshop de Testes e Tolerância a Falhas*, pages 6–19. SBC.
- [Basso et al. 2016] Basso, T., Matsunaga, R., Moraes, R., and Antunes, N. (2016). Challenges on anonymity, privacy, and big data. In *Dependable Computing (LADC), 2016 Seventh Latin-American Symposium on*, pages 164–171. IEEE.
- [Basso et al. 2019b] Basso, T., Silva, H., and Moraes, R. (2019b). On the use of quality models to characterize trustworthiness properties. In *International Workshop on Software Engineering for Resilient Systems*, pages 147–155. Springer.
- [Casare et al. 2020] Casare, A., Basso, T., and Moraes, R. (2020). Trust metrics to measure website user experience. In *The Thirteenth International Conference on Advances in Computer-Human Interactions*, pages 1–8. Iaria.
- [D. Miller 2017] D. Miller (2017). Think cloud compliance: an introduction to cloud computing for legal and compliance professionals. <https://download.microsoft.com/download/0/D/6/0D68AE95-6414-4074-B4B8-34039831E2BF/Introduction-to-Cloud-Computing-for-Legal-and-Compliance-Professionals.pdf>.
- [Dyczkowski et al. 2014] Dyczkowski, M., Korczak, J., and Dudycz, H. (2014). Multi-criteria evaluation of the intelligent dashboard for sme managers based on scorecard framework. In *the 2014 Federated Conference on Computer Science and Information Systems*, pages 1147–1155.
- [Faculty of Technology 2020] Faculty of Technology (2020). Software engineering department. URL: <https://www.ft.unicamp.br/pt-br/laboratorio/sistemas-de-informação-e-engenharia-de-software-seis> [Last access on May, 2020].
- [Ferreira et al. 2016] Ferreira, R., Martiniao, A., and Sassi, R. (2016). Dashboard inteligente para apoio à tomada de decisão em empresa de courier. *Revista Gestão & Tecnologia*, pages 39–72.

- [Gesing et al. 2014] Gesing, S., Atkinson, M., Filgueira, R., Taylor, I., Jones, A., Stankovski, V., Liew, C. S., Spinuso, A., Terstyanszky, G., and Kacsuk, P. (2014). Workflows in a dashboard: a new generation of usability. In *2014 9th Workshop on Workflows in Support of Large-Scale Science*, pages 82–93. IEEE.
- [IBM Corporation 2006] IBM Corporation (2006). An architectural blueprint for autonomic computing. *IBM White Paper*, 31:1–6.
- [ISO 2014] ISO (2014). International organization for standardization. systems and software engineering — systems and software quality requirements and evaluation (square) — guide to square.
- [Luzzardi 2003] Luzzardi, P. R. G. (2003). Critério de avaliação de técnicas de visualização de informações hierárquicas. In *Tese de Doutorado. Programa de Pós-Graduação em Computação. UFRGS, Brasil*.
- [Magdalena et al. 2019] Magdalena, R., Ruldeviyani, Y., Sensuse, D. I., and Bernando, C. (2019). Methods to enhance the utilization of business intelligence dashboard by integration of evaluation and user testing. In *2019 3rd International Conference on Informatics and Computational Sciences (ICICoS)*, pages 1–6. IEEE.
- [Mell and Grance 2011] Mell, P. and Grance, T. (2011). The nist definition of cloud computing recommendations of the national institute of standards and technology. *National Institute of Standards and Technology, Information Technology Laboratory*, 145:7.
- [Muriithi and Kotzé 2013] Muriithi, G. and Kotzé, J. (2013). A conceptual framework for delivering cost effective business intelligence solutions as a service. In *the South African Institute for Computer Scientists and Information Technologists Conference*, pages 96–100.
- [Nielsen 1994] Nielsen, J. (1994). *Heuristic Evaluation*. John Wiley Sons, Inc.
- [Nielsen and Mack 1994] Nielsen, J. and Mack, R. L. (1994). *Usability Inspection Methods*. John Wiley Sons, Inc.
- [Smuts et al. 2015] Smuts, M., Scholtz, B., and Calitz, A. (2015). Design guidelines for business intelligence tools for novice users. In *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists*, pages 1–15.
- [Veríssimo et al. 2003] Veríssimo, P. E., Neves, N. F., and Correia, M. P. (2003). Intrusion-tolerant architectures: Concepts and design. In *Architecting dependable systems*, pages 3–36. Springer.
- [Zuk et al. 2006] Zuk, T., Schlesier, L., Neumann, P., Hancock, M. S., and Carpendale, S. (2006). Heuristics for information visualization evaluation. In *the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 55–60.