CLOUD
ACCOUNTABILITY
PROJECT

D:C-5.1 Metrics for Accountability

Deliverable Number: D35.1

Work Package: WP 35

Version: Final

Deliverable Lead Organisation: UMA

Dissemination Level: PU

Contractual Date of Delivery (release): 30th September, 2013

Date of Delivery: 18th October, 2013

Editor

David Nuñez , Carmen Fernandez-Gago (UMA)

Contributors

David Nuñez (UMA), Carmen Fernandez-Gago (UMA), Isaac Agudo (UMA), Alain Pannetrat (CSA), Jesús Luna (CSA), Stefan Berthold (KAU), Siani Pearson (HP), Massimo Felici (HP), Erdal Cayirci (UiS), Aryan TaheriMonfared (UiS), Antorweep Chakravorty (UiS), Tomasz Wiktor Wlodarczyk (UiS)

A4Cloud
www.a4cloud.eu
Accountability For Cloud and Other Future Internet Services

SEVENTH FRAMEWORK PROGRAMME

FP7-ICT-2011-8-317550-A4CLOUD

Executive Summary

Accountability in the cloud-computing paradigm is an important concept that it is influenced by other concepts or *attributes*. The conceptual framework that has been designed in WP:C-2 establishes that accountability is influenced by a collection of what we called *attributes.* These attributes are *Transparency, Responsibility, Remediability, Liability, Observability, Verifiability and Attributability.* Thus, it would be logical to think that if we are interested in assessing how accountable an organisation is we should be able to assess or provide techniques for measuring the attributes that influence on accountability. How much or to what extent they should be measured is a key issue. One of the goals of A4Cloud is, therefore, to develop a collection of metrics for performing meaningful measures on the *attributes* that influence accountability.

The main goal of this deliverable is to set up the foundations towards the elicitation of metrics for accountability attributes. We start by a review on the definitions of the basic concepts and terminology regarding metrics. The concepts related to Metrology range from what is to be measured (attributes) to what is a measure, scale or measurement method. These basic concepts on Metrology are provided by different standards, being our main sources the ISO/IEC 27004:2009(E) and ISO/IEC 15939:2007 standards. We have to frame these definitions in the context of the A4Cloud project. Thus, we have identified as relevant metrics concepts for our work those of attribute, metric or measurement result, measure, measurement method, indicate and evidence.

As our intention is to determine in a first step to what extent the accountability attributes are measurable we have performed a thorough analysis of the accountability attributes listed above from the metrics perspective. This analysis will allow us to assess their usefulness with respect to metrics as well as to identify some requirements regarding the definition of attributes in order to be measurable. The analysis focuses on aspects such as suitability of the definition from the metrics point of view, any ambiguity of the definition, inconsistencies, vagueness, etc. It is also important to analyse whether the attribute could be decomposed in other subattributes, interdependencies with other attributes, what type of evidence support metrics for an attribute, and what type of metrics could be defined for it.

Once the attributes to be measured are analysed from the metrics point of view we have to define methodologies for performing meaningful measurements for them. We describe a metamodel for metrics for accountability attributes, which constitutes the basis for the process of elicitation of metrics for accountability. This metamodel is intended to serve as a language for describing accountability attributes and sub-attributes and for identifying the elements involved in their evaluation. We believe that one of the key components of the metamodel is to identify the type of evidence the attribute uses. For identifying it, the analysis performed on the attributes is essential. Other components of the metamodel are the definition of the attribute and whether it can be decomposed in other sub-attributes, as well as the possible metrics or whether they can be decomposed. It is of paramount importance for the metamodel to identify other elements that are relevant for it such as the entity that is acting in the process and the action, which is the target of the measurement. As an example of how the metamodel can be used for accountability attributes we have used it for modelling Transparency. However, it can be applied to all the accountability attribues that we consider in the context of A4Cloud.

Additionally, we append to this deliverable the outcomes of the preliminary work, namely a collection of attributes that influence accountability and a study on measurement techniques that are defined for them.

Table of Contents

1 Introduction

Accountability is a complex concept, whose definition varies depending on the discipline where it has to be applied. Thus, for the A4Cloud context, the consortium has agreed on using the following definitions of Accountability [3]:

> *"**Conceptual Definition of Accountability:** Accountability consists of defining governance to comply in a responsible manner with internal and external criteria, ensuring implementation of appropriate actions, explaining and justifying those actions and remedying any failure to act properly.*
>
> ***A4Cloud Definition of Accountability**: Accountability for an organisation consists of accepting responsibility for the stewardship of personal and/or confidential data with which it is entrusted in a cloud environment, for processing, storing, sharing, sharing, deleting and otherwise using the data according to contractual and legal requirements from the time it is collected until when the data is destroyed (including onward transfer to and from third parties). It involves committing to legal and ethical obligations, policies, procedures and mechanisms, explaining and demonstrating ethical implementation to internal and external stakeholders and remedying any failure to act properly"*

One of the important aspects behind the accountability concept is the ability of an organization to demonstrate their conformity with required obligations [21]. The ultimate objective of the A4Cloud project is the demonstration of this through the measurement of the degree of such conformity and the provision of meaningful evidence. Thus, measurement becomes an important tool for assessing the accountability of an organization by external authorities (and organizations themselves, in the case of self-assessment).

In this deliverable we will introduce the methods and models that are going to be used within A4Cloud in order to define and use metrics for the accountability attributes that are identified in the project. These attributes are transparency, verifiability, observability, remediability, liability, responsibility and attributability. Metrics can be of different types (quantitative and qualitative), and they can be supported by different kinds of evidence. Thus, for the case of Accountability we need to determine which are the most suitable ones for each case. The methods and models that we will introduce in this document will depend on different aspects such as the context and the nature of the attribute to be measured. For this reason one of the key points of this deliverable is to carry out an analysis of the attributes identified in MS:C-2.2 [3] from the metrics point of view.

Once the analysis is done our focus will be on measuring accountability attributes. We introduce a metamodel that permits to model different non-functional properties of cloud services, and in particular, those that comprise accountability within the A4Cloud scope, and referred as attributes of accountability. Our purpose is to use this metamodel as part of a methodology for the elicitation of properties and defining metrics for them.

This document is structured as follows: Section 2 serves as an introduction to basic concepts and definitions on measurement. Section 3 constitutes an analysis of the working definitions of the accountability attributes, from the metrics' perspective. Finally, Section 4 describes the metamodel for modelling the accountability attributes and metrics. In addition, we gather in the appendices a collection of attributes that influence accountability.

2 Background Concepts and Definitions for Metrics

One of the objectives of this project is to develop measurement techniques for the non-functional properties that influence or are influenced by accountability. Such properties, referred as *attributes of accountability*, include transparency, verifiability, observability, liability, responsibility and attributability. Essentially, what is needed for ensuring accountability is to be able to demonstrate that the accounts provided by an organisation (to regulators, auditors, data subjects or other service providers) are adequate and appropriate for the context by meeting certain internal and external criteria, and to have in place mechanisms for dealing with the situation (including sanctions and other measures possibly leading to the remediation of failures) if this is not the case. From an organisational point of view the focus is on measuring whether the fundamental types of activities that an accountable organisation should undertake are in place and effective.

From a technical viewpoint, metrics are an instrument for verifying the compliance of non-functional requirements, such as those related to security, privacy, or accountability. Metrics are also a tool that facilitates the decision making process, since they can be seen as an input of the management review process of an organization [48]. For example, they are an important aspect of maturity models such as CMMI and PRINCE2 Maturity Model, since they are used to support management decisions, improve quality assessment, monitoring of performance, etc.

2.1 Review of Basic Concepts and Definitions on Metrology

This work package has a strong component of metrology, as its main objective is the development of a set of metrics for measuring accountability-related attributes. In consequence, part of our initial work has been the study of metrology within the context of information security and privacy, in order to undertake the development of new metrics in latter stages of the work package. This initial study allows us to define a metrics framework for A4Cloud.

Metrology is defined as the scientific study of measurement [64]. As such, there already exist a broad selection of reference material regarding metrology concepts, including standards, books, research papers and guidelines. In this section we will provide a brief review of the most important sources. In particular, we will use the following material as the main reference on metrology and information security measurement:

- ISO/IEC 27004:2009 (E) – Information Technology – Security techniques – Information Security Management – Measurement [48]: This standard belongs to the ISO/IEC 27000 family on information security. In particular, the 27004 standard provides guidance on the development and use of measures with respect to Information Security Management Systems (ISMS). Most of the definitions regarding measurement proposed for this project are extracted or adapted from this standard.
- NIST SP 800-55 (revision 1) – Performance Measurement Guide for Information Security [66].
- Complete Guide to Security and Privacy Metrics (Debra S. Herrman) [41]**:** As its title states, this book provides extensive guidelines for developing security and privacy metrics, as it is based on a wide selection of metrology and information security standards and guidelines.
- Software Metrics and Software Metrology (Alain Abran) [6] is another useful source of metrology concepts, in this case with a focus on the software area. It provides basic concepts for designing measurement methods.

2.1.1 Fundamental Metrology Definitions

In this section, we gather relevant definitions from the aforementioned sources. From this collection of definitions and concepts we will propose a set of basic definitions for this WP in section 2.1.2. The standard ISO/IEC 27004:2009(E) [48] is our main source of basic concepts and definitions regarding information security metrics. Note that most of these definitions are in turn based on ISO/IEC 15939:2007 [47]:

- **Attribute:** property or characteristic of an object that can be distinguished quantitatively or qualitatively by human or automated means [47].

- **Scale:** ordered set of values, continuous or discrete, or a set of categories to which the attribute is mapped [47].
- **Measurement method:** logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale [47].
- **Decision criteria:** thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result [47].
- **Information need:** insight necessary to manage objectives, goals, risks and problems [47].
- **Measure:** Variable to which a value is assigned as a result of measurement.
- **Measurement:** process for obtaining information about the effectiveness of ISMS (Information Security Management System) and controls using a measurement method, a measurement function, an analytical model, and decision criteria.
- **Base measure:** measure defined in terms of an attribute and the method for quantifying it [47].
- **Derived measure:** measure that is defined as a function of two or more values of base measures [47].
- **Analytical model:** algorithm or calculation combining one or more base and/or derived measures with associated decision criteria [47].
- **Measurement function:** algorithm or calculation performed to combine two or more base measures [47].
- **Measurement results:** one or more indicators and their associated interpretations that address an information need.

We also collect some useful concepts from [41], which in turn are adapted from other sources ([34][35][44][45][66]). Original sources are referenced where applicable:

- **Metric**: a proposed measure or unit of measure that is designed to facilitate decision making and improve performance and accountability through collection, analysis, and reporting of relevant data [66][35].
- **Measurement:** the process by which numbers or symbols are assigned to entities in the real world in such a way as to describe them according to clearly defined rules [34][35]. The comparison of a property of an object to a similar property of a standard reference [44][45].
- **Primitive:** data relating to the development or use of software that is used in developing measures of quantitative descriptions of software. Primitives are directly measurable or countable, or may be given a constant value or condition for a specific measure. Examples include error, fault, failure, time, time interval, date, and number of an item [44][45].

Another interesting definition from [6]:

- **Attribute:** the property of an entity that can be determined quantitatively, that is, for which a magnitude can be assigned. In the metrology vocabulary, this is called a measurable quantity, or quantity for short.

2.1.2 Proposed Definitions

For the specific work we are aiming at WP C-5, framed into the view of the A4Cloud project, we need to adapt the definitions presented in the previous section. We propose the following concepts for the scope of this work package:

- **Attribute**: property or characteristic of an object that can be distinguished quantitatively or qualitatively by human or automated means [47].
- **Metric or measurement result**: A set of indicators, together with an associated interpretation, that is designed to facilitate decision-making and improve performance and accountability through collection, analysis, and reporting of relevant data (adapted from [48][41]).
- **Measure**: variable whose value is assigned as a result of measurement [47].
- **Measurement method**: logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale [47].
- **Indicator:** measure that provides an estimate or evaluation of specified attributes derived from an analytical model with respect to defined information needs [48].

- **Evidence**: data collected to support a metric, including the data inputs necessary to calculate and validate the metric (adapted from [66]).

These definitions are included in the A4Cloud glossary [2], which is currently under development. Note that we have decided to use the term "attribute" rather than "property". This decision is based on the fact that the ISO/IEC 27004:2009 standard uses the "attribute" term for referring to the measurable concept. Taking into consideration the high relevance of this standard, we believe it would be wiser to coincide in this term with it. In addition, the term "property" is often used to refer to functional properties of a system. Therefore, in our case, attribute is used as a synonym of "non-functional property". The term "attribute" is also the one used in the Conceptual Framework for describing the main concepts that comprise accountability, and that will be subject to the definition of metrics by this work package.

2.1.3 Scales of Measurement

In the classical theory of measurement [79], the *scales of measurement* (or *levels of measurement*) are a set of categories for classifying measurement methods regarding their characteristics. Identifying the scale for each particular metric is essential for interpreting and analysing its results. Moreover, since each scale has a set of permitted operations, knowing its scale allows us to assess the validity of a metric, or at least, to discard senseless metrics.

- Nominal scales: This type of scale is applicable for mapping entities to names or categories. It is also known as categorical scale. Values in a nominal scale do not have any kind of relation to each other. For this reason, only the equality operation (=) is permitted for nominal values. From a statistical viewpoint, only modes can be computed.
- Ordinal scales: This scale permits to assign an order relation to its values, which is used to put measured entities in order. For this reason, ordinal scales are said to have magnitude. However, there is no information for measuring the differences between values. A simple example of this scale is the set of values "Low – Medium – High". There is an order relation that permits to state that High is greater than Medium, which in turn is greater than Low, but it makes no sense to measure the difference between Low and Medium. Ordinal scales are also nominal. Ordinal scales therefore permit to use equality (=) and inequality (≤) operations, as well as medians and percentiles. Certain non-parametric statistical tests that only require ordinal data, known as ranking tests [76], can also be performed.
- Interval scales: This type of scale permits to measure differences between values. Additionally, interval scales are also ordinal scales. Thus, their values can be compared and ordered. Interval scales permit additions and substractions of their values. Therefore, means and standard deviations can also be computed. However, multiplications and divisions, and hence any other operations that depend of those, such as ratios, cannot be performed.
- Ratio scales: This type of scale improves interval scales by adding a meaningful zero value. Ratio scales are also interval scales. All the operations that are valid for interval scales apply here. In addition, multiplication and division are also meaningful.

Nominal and ordinal metrics are often grouped as **qualitative** metrics, whereas interval and ratio metrics are **quantitative**. This differentiation is very important when facing the processing the results of metrics, which will happen when aggregating and compositing metrics or when producing interpretation of the results of a metrics. Qualitative metrics may need to be converted to quantitative, in order to make possible complex processing, such as aggregated metrics. Note that this process often consists on defining a transformation from a qualitative domain (which at most possess a partial ordering) to a numeric one, which implies making assumptions on the validity of such transformation. On the contrary, quantitative metrics may need to be converted to qualitative when facing the reporting of final assessments, in order to be easily interpreted by people; for example, a numeric metric could be transformed to a simple Green/Yellow/Red label.

3 Analysis of Accountability Attributes from the Metrics Perspective

This subsection constitutes an analysis of the Accountability Attributes defined in MS:C-2.2 [3], which are the following:

- Responsibility
- Attributability
- Liability
- Transparency
- Remediability
- Verifiability
- Observability

Note that our analysis will be based on the results from the Conceptual Framework at the time of MS:C-2.2. Further developments regarding the accountability attributes, will imply an update on this analysis. Other concepts that the Conceptual Framework identified previously in [2], but left pending of finding the relations with the accountability attributes, are: Access control, Assurance, Attribution, Audit, Contracts, Control, Data protection, Data stewardship, Demonstration, Evidence, Immutability, Non-repudiation, Penalty, Privacy, Privacy by design, Privacy impact assessment, Redress, Risk, Trust, Sanctions. It is an open issue to define properly these concepts and to analyse their influence on the accountability attributes from the metrics perspective.

In this section we will analyse the attributes of the Accountability attributes in order to assess their usefulness with respect to metrics. This analysis will also allow us to identify some requirements regarding the definition of attributes in order to be measurable. From the metrics perspective we will be focusing on the following aspects.

- *Do the definitions of the accountability attributes are valid from the point of view of metrics? Is there any ambiguity in the definition given by the Conceptual Framework? Is the attribute to be evaluated well identified from the definition?* Inconsistencies, vagueness, and significant overlappings of the definitions of the Accountability Attributes should be identified and (ideally) a correction should be proposed.
- *Can the attribute be decomposed in other sub-attributes?* For some cases the definitions from WP:C-2 are very abstract and high-level. However, we can identify particular cases for each attribute depending on its nature and context that may be more concrete and useful from a metrics viewpoint. For example, the transparency requirements between data processors and data controllers are not the same as those requested between data controllers and data subjects. The latter is what we called DataProtectionTransparency in the example provided in Section 4. In that example, Transparency is a high-level goal, which is decomposed into several related attributes (although we only specified DataProtectionTransparency). A similar discussion should be done with respect to other sub-attributes identified for each accountability attribute. Hence, this is a recursive process: if a sub-attribute is identified, the analysis should be also done on the sub-attribute.
- *Interdependencies with other attributes (whether they affect positively or negatively other attributes or sub-attributes)*: Currently, the conceptual framework identifies seven attributes of accountability: Responsibility, Attributability, Liability, Transparency, Remediability Verifiability, and Observability. However, there exist other sub-concepts that may influence the main accountability attributes.
- *What type of evidence support metrics for this attribute?* It is very important to identify what elements can be used as tangible evidence for supporting the evaluation of each attribute (or sub-attributes).
- *What type of metrics could be defined for this attribute? Are there any requirements for a metric for this attribute?* We will give an evaluation of the viability and potential of measuring each attribute, and identify the possible characteristics for a metric for such attribute (type of

evidence, type of scale, etc.) depending of its nature and context. If possible, we will identify potential metrics.

3.1 Transparency

The following is the definition of transparency from MS:C-2.2 [3]:

> **Transparency:** Transparency involves operating in such a way as to maximise the amount of and ease-of-access to information which may be obtained about the structure and behaviour of a system or process. It is an attribute of an object, process or system that its creation or behaviour can be observed. For example, a cloud provider offers transparency of its security processes if it provides a web page with current and historical availability. It provides further transparency if it offers explanations for outages. More specifically, *'ex ante transparency'* should enable the anticipation of consequences before data is actually disclosed (usually with the help of privacy policy statements), whereas '*ex post transparency'* informs about consequences if data already has been revealed (i.e. what data is processed by whom and whether the data processing is in conformance with negotiated or stated policies) [42]. Transparency encompasses the property of an accountable system that it is capable of "giving account" of, or providing visibility of how it conforms to its governing rules and commitments: "Information Accountability means that Information usage should be transparent so it is possible to determine whether a use is appropriate under a given set of rules" [83]. More broadly, an accountable organisation is transparent in the sense that it makes known to relevant stakeholders the policies defined about treatment of personal and confidential data, can demonstrate how these are implemented and provides appropriate notifications in case of policy violation, as well as responding adequately to data subject access requests. Note that transparency does not involve revealing the personal or confidential data itself, as that should be kept confidential, with the exception that data subjects have the right to access their own data (cf. data subject access). This is analogous to the privacy principle of transparency, as for example elucidated in the OECD privacy guidelines [67], which is about the need for transparency of privacy policies and not of the personal data.

We can see that this definition is, in fact, comprised of several definitions of transparency, and other concepts related.

1. The first definition is very general and abstract, speaking in terms of systems and objects. It is later illustrated with an example related to a cloud service provider: "*Transparency involves operating in such a way as to maximise the amount of and ease-of-access to information which may be obtained about the structure and behaviour of a system or process. It is an attribute of an object, process or system that its creation or behaviour can be observed. For example, a cloud provider offers transparency of its security processes if it provides a web page with current and historical availability. It provides further transparency if it offers explanations for outages*". Giving the scope of the project, this first definition is too vague for being helpful for the definition of metrics.
2. Next, two sub-concepts of transparency are identified: "*More specifically, 'ex ante transparency' should enable the anticipation of consequences before data is actually disclosed (usually with the help of privacy policy statements), whereas 'ex post transparency' informs about consequences if data already has been revealed (i.e. what data is processed by whom and whether the data processing is in conformance with negotiated or stated policies) [42]*". This identification is useful for decomposing the transparency concept.
3. Next, a more precise definition of transparency in the scope of the project is given: "*Transparency encompasses the property of an accountable system that it is capable of "giving account" of, or providing visibility of how it conforms to its governing rules and commitments: "Information Accountability means that Information usage should be transparent so it is possible to determine whether a use is appropriate under a given set of rules" [83]*". In this definition, the concepts of "giving account" and "providing visibility of conformance" are stressed.
4. Next, a richer definition is presented: "*More broadly, an accountable organisation is transparent in the sense that it makes known to relevant stakeholders the policies defined about treatment of personal and confidential data, can demonstrate how these are*

implemented and provides appropriate notifications in case of policy violation, as well as responding adequately to data subject access requests". In this definition, several sub-concepts of transparency are enumerated. We will use this definition as a basis for decomposing the transparency attribute into transparency practices.

5. Finally, a note clarifying that in our context, transparency is related to the organizational processes regarding treatment of personal data and not to personal information itself: "*Note that transparency does not involve revealing the personal or confidential data itself, as that should be kept confidential, with the exception that data subjects have the right to access their own data (cf. data subject access). This is analogous to the privacy principle of transparency, as for example elucidated in the OECD privacy guidelines [67], which is about the need for transparency of privacy policies and not of the personal data.*"

Taking into consideration all these sub-definitions, we provide a single definition of transparency that is more useful from the metrics perspective:

> ***Transparency*** *is a property of an organization or a system about how well it implements and demonstrates the implementation of the following three transparency practices:*
>
> - *Informing upstream*[1] *stakeholders about data protection policies and their implementation practices.*
> - *Notification in case of policy violation and other events that have been agreed upon in the policy, which includes explanation of the actions taken on such event.*
> - *Responding to data subject access requests about data handling, e.g., data storing and processing.*
>
> *A transparent organization will implement procedures for supporting these practices, and will provide means for demonstrating the existence and quality of such procedures.*

Note that the first transparency process can be identified with the "ex ante transparency" concept, whereas the second with the "ex post transparency" concept.

Hence, from a high-level viewpoint, a **transparency metric** would measure the susceptibility of an organization's policies and procedures regarding data protection to be inspected by relevant parties (such as data subjects), as well as the quality of the transparency processes held in place by the organization.

There are several dimensions for assessing transparency. We identify the following:

- Accessibility: This dimension is related to the level of ease for obtaining the necessary information by the relevant stakeholders. The more transparent an organization is, the easier for stakeholders will be to obtain the information they need.
- Effectiveness: Even if information is fully accessible, it may not be effective, as for transparency to exist it is necessary that the receptor is capable of processing, digesting and using the information [43]. That is, this dimension is related with the usefulness of provided information. For example, the provision of excessive amounts of information, although accesible, renders it useless. The same aspect applies to the format and method of the provision of information.
- Timing: This dimension is related to assessing when the transparency actions are taken with regard to the event that triggered (this dimension has more sense with aspects such as notification). For example, it is possible to measure quantitatively the elapsed time between the event of the violation of a privacy policy and the corresponding notification.
- Other dimensions can be framed as combinations of accessibility, effectiveness, and timing. For instance, the provided information may be incomplete at the beginning, an accessibility problem, but may be completed after further user requests, which is also a timing problem.

With regards to evidence, quality and effectiveness of transparency procedures can be assessed from information of two main sources:

[1] The word *upstream* refers to the stream of personal data from the end user *down* to the cloud service provider and their subcontractors. The upstream path leads through the intermediate cloud service providers eventually back to the end user.

- Certification from third parties: Organizations may be audited and/or certified by trusted third-parties, who can then assert the existence and quality of the procedures held in practice for supporting transparency, e.g., for acquiring an EuroPrise Privacy Seal.
- Self-assertion: Organizations may also inform of the details and characteristics of such procedures, using self-asserted means such as the CSA STAR (CSA Security, Trust & Assurance Registry).

Certification and assertions of effectiveness should be supported by the findings of end user tests that research whether the provided information is comprehensible, i.e., leads to reasonable conclusions, and whether the end users are aware of possible data processing and their rights, e.g., to request information.

3.2 Responsibility

The following is the definition of responsibility from MS:C-2.2 [3]:

> **Responsibility:** Responsibility may be defined as the state of being assigned to take action to ensure conformity to a particular set of policies or rules.
> Accordingly, a responsible entity is one that is assigned to take action to ensure conformity to a particular set of policies and rules. For example, if a policy states that incidents of level ‘severity 3’ and above should be reported to National Regulatory Authority, a person or system component has to be assigned to take this action once an incident occurs. This person is responsible for incident reporting.
> *Attribution* of responsibility is a key element of accountability, as is apparent from definitions given in dictionaries, which tend to centre on accountability as the quality or state of being held to account for one’s actions and an obligation or willingness to accept responsibility for one’s actions. For example: “Accountability is the obligation and / or willingness to demonstrate and take responsibility for performance in light of agreed upon expectations. Accountability goes beyond responsibility by obligating an organisation to be answerable for its actions” [20]. Specifically, an accountable organisation is responsible for the stewardship of personal and confidential data with which it is entrusted.

As in the case of Transparency (cf., Section 3.1), the working definition of Responsibility presented above, is still too general (thus forbidding the elicitation of the respective metrics). Some of the ambiguities that we can highlight from the current definition are:

- The working definition considers only the “conformance” with respect to a predefined set of policies/rules, but does not state anything about the “non-conformance” aspects. For example, with the current definition it is not clear if an entity can be responsible for not acting in accordance with some policy/rule.
- Similar to the previous point, the notion of “delegation” is missing in the definition. That is, we need to clarify if an entity is allowed to delegate its responsibility into another entity. Being responsible can mean either being accountable for a state of affairs without necessarily any implication of a direct causal connection, or being the primary cause of a result. We must name these two distinct types of responsibility e.g., as consequential and causal responsibility respectively.
- A more rigorous definition and models are needed to clearly state the involved entities and roles. As proposed by the IST MAFTIA project [58], a responsibility model is needed “to determine whether an individual is intruding/misbehaving or merely carrying out some responsibility of which the detector is unaware[2]”. Such responsibility model could be the starting point to elicit relevant metrics.
- The notions of authenticity and non-repudiation (close related to “attribution of responsibility”) are also needed to facilitate the elicitation/understanding of related metrics. However, it should be taken into account that “attribution of responsibility” overlaps with the working definition of “Attributability”.
- The last part of the working definition is also ambiguous (i.e., “[…] an accountable organisation is responsible for the stewardship of personal and confidential data with which it is entrusted”),

[2] Notice that this also implies that we must understand the responsibilities of the detector.

because seems to overlap with the Liability, Obligations and Sanctions attributes. The interdependences between these concepts should be clearly stated.

In order to start the metrics' elicitation process, our approach considers analysing in further detail the Responsibility attribute. In particular we start by identifying the elements that are mentioned by the definition:

1. Entity responsible for processing the personal data (could be the result of a responsibility delegation process).
2. Conformance: the process of attesting that the entity is actually fulfilling a defined set of policies/rules.
3. Policies/rules with respect to which, the entity's conformance is evaluated.
4. Attribution of responsibility, which is tightly coupled with the notion of non-repudiation and authenticity.

A preliminary re-definition of Responsibility for the purposes of eliciting applicable metrics will be based on MAFTIA's [58]:

> ***Responsibility*** *is a relationship between two entities regarding a specific Responsibility Target (policy/rules/states of affairs), such that the Responsibility Holder is responsible to the giver of the responsibility, the Responsibility Principal.*

According to this definition, the Responsibility attribute should take into account *any operation* performed by the responsibility holder, then the policy should be used to evaluate if performed action was conformant or not. As shown in Figure 1, the important point in the Responsibility attribute is that responsibilities cannot be looked at in an isolated way but must always be considered as a relationship between two agents. The Responsibility Target for which responsibilities are held may be at any level of granularity of the organization.

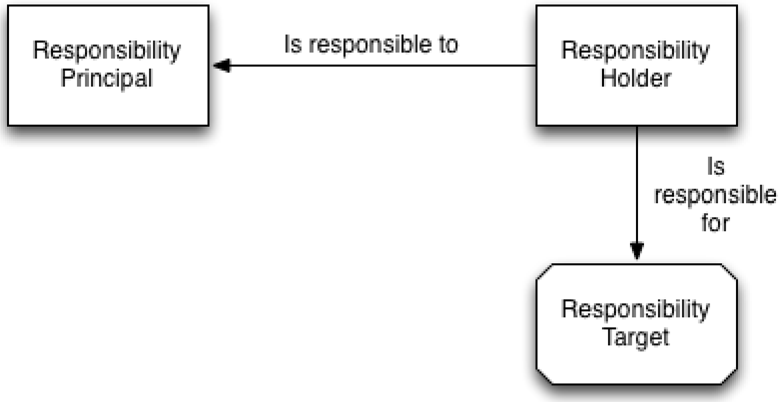

Figure 1: Responsibility Relationships

The following *responsibility practices* can be derived from the above definition[3]:

- Responsibility granting, the process where the responsibility principal grants the actual responsibility to the responsibility holder. It should be noted that, as discussed below in this section, the granting of responsibility can actually involve a chain of Responsibility Holders, as shown in Figure 2. For example, the primary Cloud Service Provider (data processor) might be responsible towards the Cloud Customer (data controller), if one of the sub-processors carrying out processing operations on his behalf do not implement the appropriate security measures for the protection of personal data.

[3] This list is by no means complete, and will be updated in future versions of this document.

- Responsibility assessment/attribution, the process where conformance of the Responsibility Holder's performed actions is evaluated with respect to the Responsibility Target. This practice can be subdivided into the following:
    - Non-repudiation, comprehending the unambiguous authenticity and integrity of the Responsibility Holder's identity.
    - Authentication, as required to assess the identity of the Responsibility Holder. As mentioned in the example above, in a chain of Responsibility it can be possible that the person responsible of a malicious action, is not also the legal responsible.
    - Integrity, needed to assess that the Responsibility Holder's identity and actions have not been tampered with.

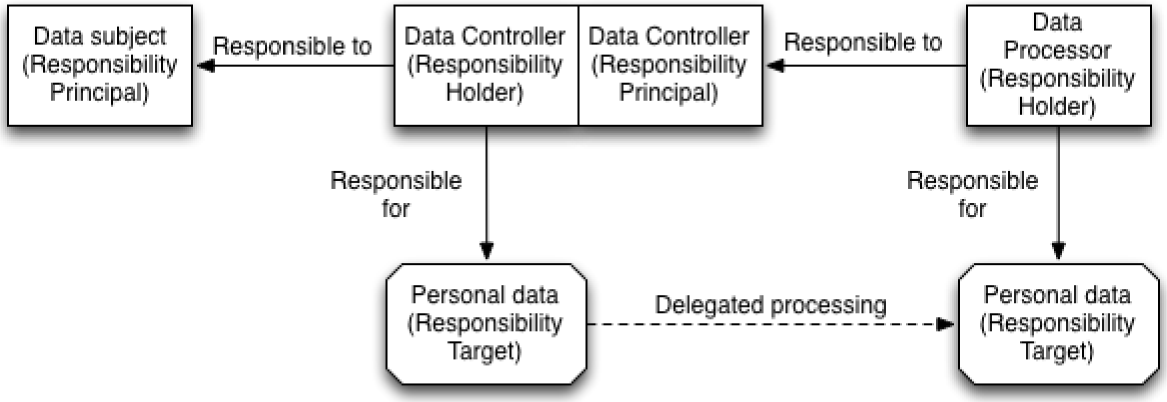
Data subject (Responsibility Principal)
Responsible to
Data Controller (Responsibility Holder)
Data Controller (Responsibility Principal)
Responsible to
Data Processor (Responsibility Holder)
Responsible for
Responsible for
Personal data (Responsibility Target)
Delegated processing
Personal data (Responsibility Target)

Figure 2: Responsibility Chain

If we want to measure (either qualitatively or quantitatively) an entity's responsibility (i.e., its *Responsibility Level*) with respect to (i) some specific action and (ii) a set of policy/rules, then some high-level metrics to take into account are:

- Level of Authentication (LoA): different organizations are likely to deploy different authentication mechanisms, therefore we cannot expect the same assurance in the responsible entity's (unambiguous) identification process.
- Delegation of Responsibility: this metric should assess the responsibility delegation process. It is clear that responsibility will attenuate in long delegation chains.
- Integrity: in analogy to the LoA metric(s), the inherent assurance of the adopted integrity mechanisms must be assessed to measure the organization's responsibility. For example, an organization using MD5 to protect the integrity of their log files cannot have the same Responsibility Level of other organization using SHA-512, due to the inferior integrity level offered by MD5 with respect to SHA-512.
- Duty/Role separation: the model used to split the responsibilities (e.g., n out of m), must be clearly stated in order to determine the responsible entity/entities. Notice that this metric is somehow related with the Delegation metric.

The overall Responsibility Level should be computed from the aggregation/composition of individual Responsibility metrics (quantitative or qualitative). For this purpose, state of the art security quantification techniques like [55] can be used. These techniques, along with a more comprehensive and rigorous definition of the Responsibility model depicted in Figure 1, are for discussion in a later version of this deliverable.

3.3 Remediability

The definition of remediation from the MS:C-2.2 is the following:

> **Remediation[4]:** Remediation is the act or process of correcting a fault or deficiency. In IT literature, remediation generally refers to being able to restore systems to earlier states in case of system failures, which may require going back many months for a known-good configuration. In relation to data and securities breaches, remediation is part of the "incident response, notification, and remediation". When harm occurs due to a failure of an organisation's privacy practices or to a lapse in its compliance with its internal policies, individuals should have access to a recourse mechanism [20], which can be triggered by an incident report. The organisation acts upon the incident report by notifying the relevant stakeholders (e.g., affected data subjects, regulators, services elsewhere in the service chain) and by repairing the damages. This may involve restoring data to the state prior to the incident, but also support forensic recording of incident data. In a broader context remediation also relates to legal remedies. When data is lost or misused, users may suffer financial damage. Remediation in this sense may refer to claiming compensatory damages or even punitive damages. In the context of accountability, the accountable organisation is required to take corrective action in case of failure to apply governing rules and honour commitments. This is one of the five elements of accountability mentioned by the Galway project [20]. Remediation is also explicitly specified in our definition of accountability given in Section 2.1.

According to these definitions, **remediability** is a property of an organization on the quality of its internal processes for taking corrective and compensatory actions in case of failing to comply with their commitments and policies. Remediation is supported by three main practices:

- Notification, which implies informing the relevant stakeholders (e.g., affected data subjects, regulators, services elsewhere in the service chain) about the failure, breach or disclosure.[5]
- Reparation, which is related with taking corrective actions and technical remedies for restoring the system to the state prior the damage, if possible. This implies restoring data and supporting forensic recording.
- Redress, which implies legal remedies due to the damage suffered. These remedies may imply that the affected part claims compensatory, or even, punitive damages.

The Remediability concept is built upon the existence of a relation of responsibility between two entities, the responsibility holder and the responsibility principal (as described in Section 3.2) and the occurrence of a failure to comply with the responsibility target. Remediability also adds a fourth entity called **remediation agent**, such as a court or a dispute resolution entity, which may be used as a third-party by the responsibility principal and the responsibility holder in order to arbiter the remediation actions. The intermediation of this entity is optional. These relationships are shown in Figure 3.

[4] Given the ongoing discussions in WP:C-2 and the recent replacement of "remediation" by "remediability", quotes from MS:C-2.2 still show the previous term.

[5] It can be noted that the concept of Notification in this case is identical to the same concept related to Transparency. Thus, notification is a practice that supports both remediation and transparency.

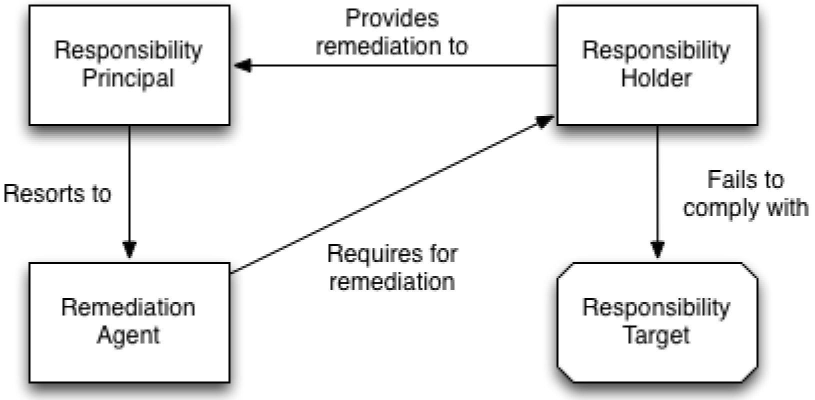

Figure 3: Remediability Relationships

Remediation is at the end of a process that starts with the occurrence of an incident and continues with its identification and analysis by Incident Management. Whereas Incident Management is more centered in preventing and mitigating the effects of incidents in order to minimize damage, Remediation is oriented to the corrective and compensatory actions after any damage has been already produced.

It is important also to note that the legal implications for redress and remediation vary depending on the nature of the implied data. In the A4Cloud context, we are tackling both personal information and business confidential data. The former is ruled by Data Protection regulations, such as the EU Data Protection Directive [30], whereas the former is constrained merely to the contracts and agreements between provider and customer. For example, the EU Data Protection Directive regulates the liability of Data Controllers in case of unlawful processing and defines the mechanisms for establishing remedies and sanctions; however, remedies and sanctions for breach of confidentiality obligations with respect to business sensitive information are provided by contractual agreements and may be held through courts [4].

A **metric for remediability** would measure the quality of the remediation practices held in place by an organization. There are several aspects that can be assessed with respect to the quality of remediation of an organization:

- As notification is part of the remediation process, one can evaluate the quality of the notification procedures. Some aspects that can be used:
  - Existence and quality of the notification processes: A naïve approach could be simply to assess the existence of internal policies within an organization for addressing the notification of the affected parties after any damage has occurred. However, a more rich approach could be taken in order to evaluate the quality of these procedures.
  - Timing of notification: As in the case of Transparency, the relevance of notification is affected by the elapsed time between the occurrence of the damage and the effective time of notification. Quantitative measures can be extracted here.
  - Effectiveness of the notification means: Even if notification is provided, it may not be useful for the relevant stakeholder. For example, indirect notification, such as publication of a notice in a web site is not as useful to a direct notification by email. Also, the information included in the notification should be useful enough for the affected party, such as a proper explanation of the incident and the taken actions, and a description of the possible options for seeking for remediation.
- In the case of reparation activities, metrics could be defined to evaluate the quality of the technical remedies and corrective actions:
  - Preparedness level: Evaluates the existence and quality of the preemptive actions intended to prepare the organization in advance to the event of a failure and the necessity of restoring to a prior state. Some of these practices are:
    - Data recovery: restoring data to state prior to the incident.
    - Support forensic recording of incident data
  - Repairability level: Assess the level of reparation of an organization to restore a failure, from the perspective of the affected party. For example, restoring damaged

data from a back-up can be enough, while the disclosure of personal data that has already taken place cannot be entirely corrected.

- As for a metric for redress, it could measure aspects that impact the quality of the redress actions planned and taken by the organization, such as:
    - Proper definition of compensations
    - Standard vs custom compensation
    - Number of incidents that end up with compensatory/punitive damages
    - Expenses due to compensatory damages (e.g. average/total redress per upheld complaint)
    - Number of complaints
    - Time to resolve a complaint
    - Type of compensatory damages (monetary, service credits, etc.)
    - Number and type of sanctions: The quality of the redress procedures are negatively affected by the sanctions the organization has received. Although the number is relevant, it is more important the type of sanctions. For example, the EU DPD defines different types of sanctions:
        - A notice addressed to the Data controller (e.g. for compulsory audit)
        - A fine
        - An injunction dictating the end of processing operations
        - A (temporary or permanent) revocation of the authorization allowing the processing of personal data
- Another dimension that can be of use to evaluate remediation is to measure its proactivity towards remediation. That is, an organization can take either a proactive or a reactive attitude with respect to remediation actions. Hence, remediation actions can be taken in a proactive manner by the organization, or in a reactive way, after complaints of the customer.

3.4 Liability

The following is the definition of liability from MS:C-2.2:

> **Liability:** Liability is the state of being liable (legally responsible). Correspondingly, a liable entity is an entity, which is legally responsible for the (legal) consequences of a certain action. Often damages will trigger liability. The entity that is held liable is then responsible for repairing damages (e.g. through financial redress). Other forms of liability include criminal liability and other statutory liability (e.g. on the basis of Data protection regulation). For example, if failure to report incidents results in a fine of 2% of total wealth and Bob is liable for reporting incidents, then if an incident is not reported, Bob is liable to a value of 2% of his total wealth for failure to report incidents. Liability is an element of almost every definition of accountability. For example, Koppell's five elements of accountability include "Liability: Did the organisation face consequences for its performance?" An accountable organisation takes liability in respect to the obligations (cf. policies) that they have defined. According to the A4Cloud definition, accountability extends liability in the sense that ethical elements are introduced when determining obligations.

The working definition of Liability (as presented above) is ambiguous in the sense that seems to overlap with other attributes defined in MS:C-2.2, such as Responsibility, Obligations and Sanctions, because of the "legally responsible" notion included in the working definition. As described in Section 3.2, Responsibility is also applicable to a chain of actors, therefore the adopted concept of liability should also take this into account. It is important to note that in case of data processing, in which data processors are participating, the data controller is always liable (legally responsible) towards data subjects, even if any damage is caused by a data processor. It can be seen that defining and differentiating liabililty and responsibility is pretty complex. On the one hand, responsibility is a requirement for liability to be established. On the other hand, although an entity might be responsible, it might not be considered at the end liable (for instance, due to an incident that happened, which the responsible entity could not predicted or prevented).

We consider important to amend observed ambiguities in order to proceed with the actual elicitation of relevant Liability metrics. In particular, it should be interesting to agree on the different classes of consequences that are implied from the Liability attribute, because based on those then it is actually possibly to determine how well an organization behaves with respect to their Liability attribute. For example, one given organization might specify only financial consequences, where another might also add legal consequences (thus both have a different *Level of Liability*).

As mentioned by Ryan and Heckman [72], the actual concept of Liability on the IT security field is not clear, however for the purposes of A4Cloud it is possible to say that *the attribute Liability is related with the consequences (e.g., legal and economic) that must be paid if an organization is found responsible for not fulfilling its obligations*.

The first step towards eliciting relevant Liability metrics is to decide which are the actual *consequences* to consider. For example, if a re-definition of Liability only considers economic consequences, then we can derive a set of economic-driven metrics (EDM). State of the art works on the EDM field (like Innerhofer [46]) have studied this topic in detail and, can be the starting point for A4Cloud's set of Liability metrics.

3.5 Observability

The following is the definition of observability from MS:C-2.2:

> **Observability:** is a property of an object, process or system that describes how well the internal actions of the system can be described by observing the external outputs of the system.
>
> The term observability originates from control theory and was introduced by Kalman [50]. While the formal matrix-based definitions of system observability might be difficult to directly apply to service accountability, they do offer a strong and useful basis for guiding metric definition and construction of framework of evidence. Particularly of interest is the related weaker term *detectability*. Detectability is the property that assumes that all unobservable elements are stable, that is, they do not change the outputs of the system [85].
>
> Observability may have additional effects. Experiments in the psychology of economics have shown that a considerable improvement in contribution towards a public good (which could also include responsible data stewardship) can be achieved by increasing the degree to which a human process is observable – see, for example, [37]. The strong link between accountability and deterrence is also brought out within [33].

We can see that this definition is directly based on earlier use of the term in the control theory. The question is how well such definition can be adapted to use for accountability in Cloud and future Internet services. It can be easily argued that a Cloud or other Services are in fact systems. Therefore, the definition based on control theory is applicable, even if quite general. However, with a deeper look at the implementation of concept of observability we notice that leads to situation where internal state of the system can be determined using system outputs. This might seem not achievable taking into account the complexity of Cloud; however, it ultimately could depend on level of abstraction one would apply.

Further, we notice that observability can be related to *detectability*, which allows for a certain amount of unobservable elements as long as they do not change the observed output. Application of this concept seems useful from Cloud perspective. Additional aspect of observability has to do with secondary effects observation has. It has been shown that the effect that processes are observed contributes towards improvements in how they are executed. This means that the sole fact of observation can prevent breaches.

The provided definition despite its generality provides relatively clear objectives for further developing the definition from the metrics perspective.

Observability is a property of an accountable system that describes how well it implements and demonstrates the following characteristics:

- Internal actions in the system can be determined using external inspections;
- Actions that cannot be determined do not significantly influence actions of the system;

An observable system will count with processes and procedures for supporting these characteristics, and will provide means for demonstrating the existence and quality of such procedures. In the first case, an observable system will provide “openings” for inspection; that is, means for independent inspection by third parties. In the second case, an observable organization must demonstrate and provide evidence of the low influence of unobservable actions in the state of the system. This aspect may be more difficult to fulfill. Hence, from a high-level viewpoint, an **observability metric** would measure the quality and effectiveness of such procedures.

It is important to note that the quality of accountability evidence, as currently understood by WP C-8 (Framework of Evidence), is dependent on observability. The higher the observability, the higher the quality of accountability evidence.

Quality and effectiveness of observability can be assessed mainly from information based in certification from third parties. Organizations may be audited and/or certified by trusted third parties, who can then assert to what extent external inspections relate to internal system functioning.

Basing on the analysis in C-2 we also know that observabiltiy together with transparency contribute to individuals’ capabilities to assess risks and benefits and choose the right options. From a social perspective, transparency implies both observability and verifiability (and vice versa, transparency is obtained by combining observability and verifiability).

3.6 Verifiability

The following is the definition of verifiability from MS:C-2.2:

> **Verifiability:** is a property of an object, process or system that its behaviour can be verified [confirmed] against a requirement or set of requirements.
>
> Quality or level of verifiability depends directly on the available evidence [14]. It is important to notice that some argue that verifiability can be purposefully limited in the contract specification, [11]. A closely related notion is *validation*, which relates to the property of accountability whereby it allows users, operators and third parties to verify a posteriori if the system has performed a data processing task as expected [16]. Similarly, *verification* is a process that evaluates whether a system complies with related governing regulations [70], and in the context of accountability is the ability to provide *ex post* evidence for compliance to governing rules (again mentioned by the Galway project [20]).

We can see that this definition is, in fact, circular and does not provide sources; therefore, it is of limited use. However, further analysis of comments to the definition provides useful conclusions. Process of verification evaluates system’s compliances with regulations. This seems a general but useful basis for further analysis. In other words, verifiability is a potential for documentable alignment with some set of rules.

We also note the related term validation which is an *a priori verification*, which in this context verification is seen a posterior, ex post or an ongoing process.

Finally, verifiability can be purposefully limited. The goal of this is to make verifiability and related processes manageable. Verifiability could lose its purpose in case its implementation is more costly than benefits it brings.

In a different part of MS:C-2.2 document another definition is provided. Verifiability is defined as an ability of external party to observe a given aspect of a contractual relationship through the collected evidence. While it is not incorrect, it seems it mixes the notion of verifiability with the notion of observability. However, it points to the important of contract (which is a set of rules) and evidence. We exploit these elements below.

Taking into consideration all these sub-definitions, we provide a single definition of verifiability that is more useful from the metrics perspective:

Verifiability is a property of a process or system describing how well it implements and demonstrates the implementation of the following practices:

- Compliance of process or system behaviour with rules is documentable
- Continuous documentation
- Scope of documentable compliance is a balance between benefits and costs

A verifiable process or system will implement procedures for supporting these practices, and will provide means for demonstrating the existence and quality of such procedures.

Accountability evidence relates to the documentation that should be collected in relation to compliance process. The scope of accountability evidence is based on the balance between benefits and costs.

3.7 Attributability

The definition of attributability from the MS:C-2.2 is the following:

> **Attributability:** Attributability describes a property of an observation that discloses or can be assigned to actions of a particular actor (or system element). It is the property of an act or object that it can be attributed to an entity. Accountability can be regarded as an extension of attributability when the action is governed by regulations [82]. This is related to liability since in order for liability to function; it must be attributable to a legal or natural person. In case of a deviation from the expected behaviour (fault), accountability should provide attribution in that it reveals which component is responsible [16]. Evidence is also important in the context of attributability (and hence liability), and thereby in proving non-compliance to governing rules, as well as compliance to governing rules. These governing rules could include obligations in the sense that we use them below, i.e. including legal requirements, contractual requirements and stakeholder requirements (including normative expectations about behaviour).

The given definition is adequate to start with the metrics elicitation process. It includes a first definition, given in terms of systems. Note that this definition of attributability is given from a more generic point of view than other accountability attributes, as it is defined in terms of a system, and not in terms of an organization. However, we can safely assume an organization to be considered as a system composed of people and processes, where complex interrelations between these entities exist. After the generic definition, the relationships of Attributability with Accountability and Liability are identified, and the role of evidence within attributability is stressed.

Attributability implies the existence of two attributability processes:

- An **evidence collection** process that provides data regarding the effects of the actions of an actor in the system. For example, a logging component within an information system.
- An **attribution** process that maps evidence to actors. Log analysis is an example of this kind of process.

The different elements of attributality can be seen in Figure 4.

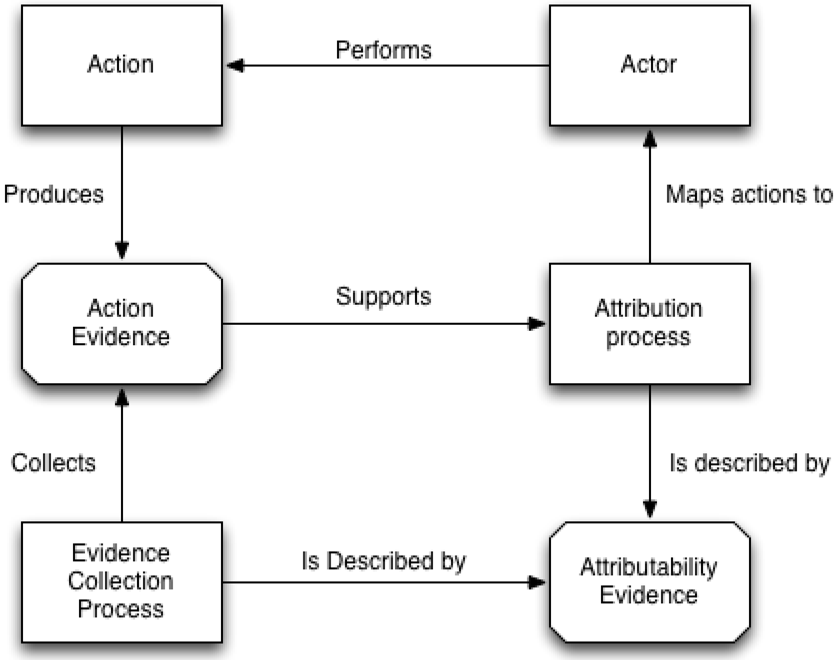

Figure 4: Attributability Relationships

According to the attributability definition, attributability is independent of regulations; that is, attributability processes should function whether regulations exist or not. Accountability is what extends attributability by taking regulations in consideration.

Attributability is a precondition of liability, as proper means for attributing actions to actors are essential for establishing legal responsibilities in case of failures to comply with commitments.

Other concepts related to Attributability are Traceability and Non-repudiation. **Traceability** is term commonly used in logistics and supply chain management to describe the ability to trace information related to goods during their production. It can be extended to information management to describe the ability to track the complete set of operations (access, modification or deletion of data) that were performed on a specific set of data. **Non-repudiation** describes the ability for an entity to produce data elements in a way their origin cannot be subsequently refuted.

A **metric for attributability** should measure the quality of the attributability processes of a system in order to ascribe actions to actors. Thus, when facing the assessment of attributability within an organization, the processes of attribution and evidence collection must be identified and described. These descriptions, which are considered the evidence of attributability, are what support a metric for attributability.

The attribution process is related to (un)linkability analysis in the privacy research domain. Linkability [69] is a metric that measures whether two items of interest are related or not. More general, linkability is the likelihood of the relation between two items of interest. The term “item of interest” refers basically to anything and would have to be restricted to observations and entities in order to match the definition of attributability. Also, the definition of attributability implies that the relation, which is to be measured is causal dependency based on evidence.

An often cited, yet far from undisputed [38], linkability metric is an approach using equivalence classes for modelling the relation between the items and information theory for transforming evidence to a linkability number [77]. Simpler metrics could be derived from set size-based approaches, e.g., anonymity sets [69]. The evidence could be used to assign all entities either to the set of entities that could have caused the observation or to the set of all other entities. Attributability would be the better the smaller the first set is.

The implementation of the attributability metric has to be chosen depending on the use-case and the available evidence. For instance, in legal scenarios, it could be required that the observation is unambiguously and provably attributed to a set on entities (usually one). For example, the observation of the factual circumstances of processing might lead to the attribution of the role of data controllers to two or more entities (called *joint data controllers*). In this case, the set notation makes sense and the evidence must be good enough to reduce the set size of the set of entities that could have caused the observation to the minimum. In other scenarios where strong indication for attribution is required, but not unambiguity, approaches based on information theory are more likely to yield the intended results.

Aspects such as Data stewardship, Data Lifecycle Management and Log Management also directly affect the quality of attributability of an organization. Thus, metrics for these subconcepts will be very useful for deriving metrics for attributability.

4 Measuring Accountability Attributes

In this section we will present a metamodel for describing metrics for accountability attributes, which will help during the process of elicitation of metrics for accountability. This metamodel is intended for the modelization of complex properties, as the accountability attributes, and metrics for measuring them. One of the main goals of this metamodel is permitting a top-down and recursive decomposition of properties. This aspect is detailed in Section 4.1.

The accountability attributes belong to the family of non-functional properties, which include all properties that are not directly related to functionality, but to a quality or behavioural attribute of a system [77]. Non-functional properties, such as the ones related to security and privacy, are of key importance with regard to the analysis and evaluation of the different aspects of a system, a service or an organization, such as quality and trustworthiness. However, their evaluation is traditionally complicated because of several reasons. Firstly, because of their subjective and ambiguous nature; secondly, non-functional properties usually present multi-dimensionality, possessing several facets; and finally, in some cases, the optimization of a non-functional property may be inconsistent with others.

The goal of defining meaningful measures for accountability attributes is subject to the problems associated with non-functional properties. We currently lack methodologies and tools for properly defining, evaluating and reasoning about such properties. As aforementioned, one of the main problems of this kind of properties is their lack of a clear definition, as they are usually described in abstract terms that are not useful from a measurement perspective. For this reason, sometimes it is very difficult to assess if such a property has been met since there is no clear-cut criteria for that. This problem is very similar to the one of non-functional requirements in the area of requirements engineering [62].

Among the problems we find when facing the elicitation of metrics for accountability attributes due to their non-functional nature are:

- Level of abstraction: Most of the time, non-functional properties are defined in a very abstract fashion, which makes them of little use from the metrics point of view. Another problem is the variety of levels of abstraction between properties.
- Ambiguity: Natural language permits vague definitions, prone to different interpretations. Definitions also tend to be similar among some properties, which facilitate their overlapping. We identify two problems:
    - Homonymy: The same name is used to designate different properties, as in the case of transparency.
    - Synonymy: A property is designated by different names. This could be a desired effect, as each name could identify a subtle variation of the property; however, in reality, most of the time, designations are arbitrarily interchanged.
- Subjectivity: Non-functional properties are often interpreted differently depending on the stakeholder and are very sensitive to the context of application (e.g.: law, computer science, social science, etc.), so in most cases there is no widely accepted definition for this kind of properties. As an illustration, we can take Transparency as an example of a non-functional property. The Cloud Industry Forum's Code of Practice [24] broadly speaking interprets transparency in the sense of transparency between the data processor and the data controller. However, within the data protection community, transparency instead is usually taken to refer to transparency of the data controller with respect to the data subject. This kind of inconsistency causes difficulties during the process of defining metrics.
- Overlapping of properties: In most cases, some of the identified properties partially or fully overlap with others. This is not negative by itself, as it is natural that two properties share some characteristics; however, from the metrics point of view, this phenomenon leads to confusion. Clearer and more disjunct definitions are needed.
- Interdependencies between properties: An exhaustive analysis of property interlinks would probably have as a result an intricate network of influences and dependencies between properties. This also makes the process of properly specifying properties and defining measurement techniques for them very difficult. As stated by the Conceptual Framework,

there exist emerging relationships (e.g. implication and inclusion) among attributes dependant on different viewpoints of analysis (which are related to different accountability perspectives, for instance, like societal, legal and ethical perspectives). For instance, from a legal perspective, responsibilities imply obligations, which consequently may lead to sanctions if these particular obligations are not met. From a social perspective, transparency implies both observability and verifiability (and vice versa, transparency is obtained by combining observability and verifiability).

4.1 Top-Down Decomposition Approach for Eliciting Accountability Metrics

It is clear then that the non-functional nature of accountability attributes is an important hindrance for defining meaningful metrics. As stated before, most of the problems we face are related to the level of abstraction of the attributes of accountability. Some of such attributes are defined in a very high-level of abstraction, which is prone to vagueness and ambiguity, and are then not useful from a metrics perspective. Furthermore, there is a disparity in the level of abstraction between different attributes. Thus, a tentative solution is to consider a stratified view of the attributes, where high-level attributes represent more vague and wide concepts and low-level attributes represent more tangible and empirical notions. This would allow also a fine-grained decomposition of attributes, if needed. Hence, we propose a top-down decomposition approach that works on two different levels:

- The **conceptual level**, where all the high-level concepts related to Accountability (e.g., the core attributes for accountability) are defined as well as the relations among them. These high-level concepts can be further refined into more concrete ones. This level will include the attributes being identified in WP:C-2 as the core attributes, and will also comprise sub-attributes that still are high-level enough for not being useful for metrics, but needed in order to define correctly the concepts related to accountability. Thus, the rationale for their definition is mainly conceptual.
- The **measurable level**, where we deal with "tangible" and empirical concepts. In certain cases, these attributes could be decomposed even more. Metrics will be initially defined for these peripheral concepts.

The idea we propose is to first go downwards in order to "break down accountability" into simpler and more low-level concepts, constructing a tree-like model (possibly, a directed graph) until we reach measurable things. This is a common approach in security metrics. Therefore, in this model, measurable concepts are in the peripheral nodes. Next, from this model, and using inference techniques over its relations, we could go upwards and construct metrics for high-level concepts. This aspect is currently under development and will be provided in next versions of the deliverable.

4.2 Metamodel for Metrics for Accountability Attributes

In this section, we propose a model-driven approach that includes the definition of a metamodel for describing metrics and accountability properties. The goal of this metamodel is to serve as a language for describing: (i) accountability properties in terms of entities, evidence and actions, and (ii) metrics for measuring them. Note that this metamodel could be extended for its application to non-functional properties in general, however, this is out of the scope of this work since we are currently focused on those related to the accountability concept.

One of the main aspects of this metamodel is that metrics are defined to take two main kinds of inputs: **Evidence** and **Criteria**. From our point of view, any assessment or evaluation (i.e, a metric) can only be made using as input some tangible and empirical evidence, such as an observation, a system log, a certification asserted by a trusted party, a textual description of a procedure, etc. That is, a metric does not directly measure a property of a process, behaviour, or a system, but uses the evidence associated with them in order to derive a meaningful measure. That is the idea that we are trying to capture in this metamodel: Evidence is the fundamental support of any evaluation method and is what gives an objective dimension to assessments. On the other hand, criteria are all the elements that convey contextual input that may constrain what should be measured, such as stakeholder's preferences, regulations and policies. It is clear then that each metric will have different nature depending on the criteria. Therefore, in this metamodel, both Evidence and Criteria are central to the definition of metrics.

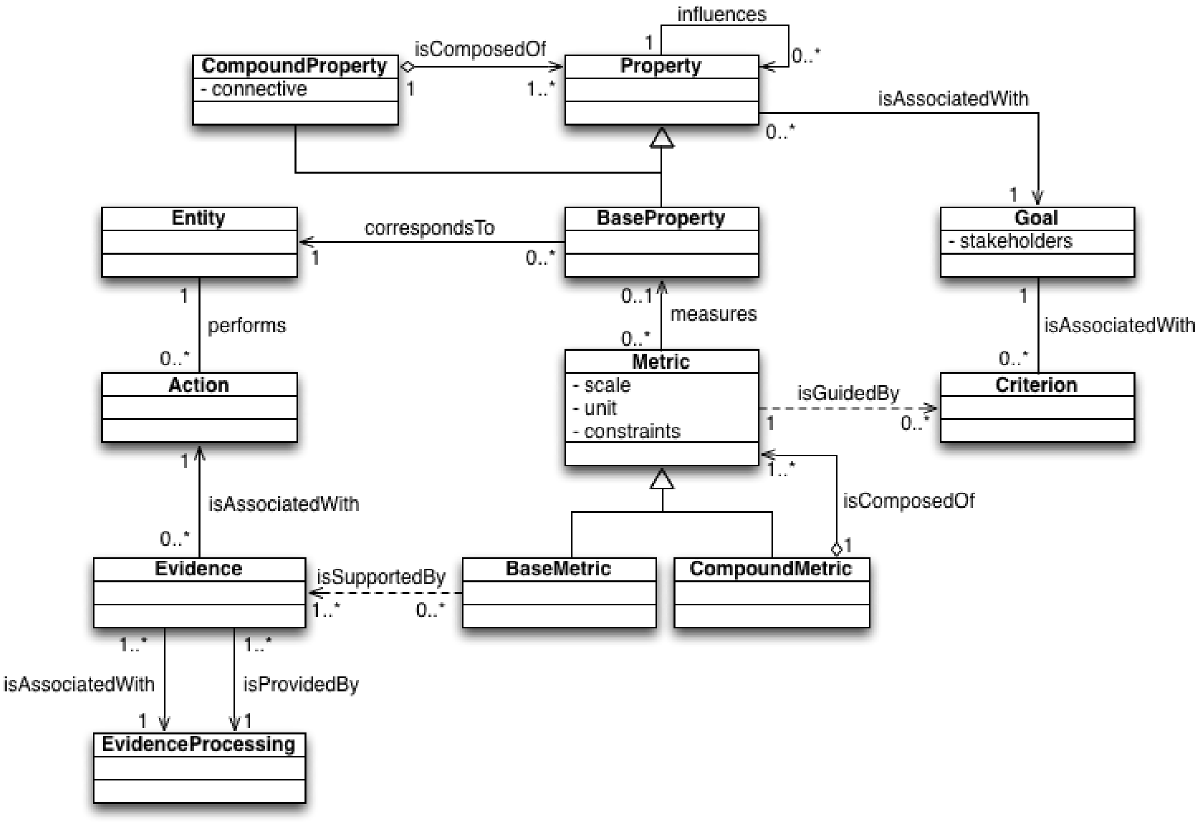

Figure 5: Metamodel for Metrics for Accountability

In this section, we will present our metamodel (see Figure 5), and provide a detailed description of each of its elements and the relations among them:

- **Goal:** High-level description of the property (or family of properties) that is modelled. These elements also contain a reference to the stakeholder (or stakeholders) for which the goal is oriented.
- **Property:** As mentioned earlier, non-functional properties are qualities or behavioural characteristics of an entity. Ideally, properties can be distinguished quantitatively or qualitatively by some evaluation method; however, properties may be defined as very high-level concepts. Thus, we consider that properties can be further decomposed into more basic ones in some cases. In these cases, **BaseProperty** elements can be defined in terms of entities and the actions between them, whereas **CompoundProperty** elements are defined in terms of other properties, making possible a top-down decomposition of properties, from a high-level and abstract way to a tangible and more accessible one. **CompoundProperty** elements then have a connective attribute, which is used for describing the logical connective used for combining properties. In addition, properties may also influence other properties, not necessarily taking part of a composition relation; the model then permits to express these influence relations between properties.
- **Entity:** This element is used to describe the entity that meets the modelled property. An entity is a physical or conceptual object that performs actions and that meets properties. For example, an organization, a process or a system can be considered as entities.
- **Action:** We define this as a process that occurs over a period of time and is performed by or has an effect on entities. Even though, actions have an effect in the environment, we cannot deal directly with these consequences, but with the evidence associated to them.

- **Evidence:** We define evidence as a collection of information with tangible representation about the effect of actions. Evidence is used to support a metric. That is, evidence is not an abstract concept about the consequence of activities, but actual data that can even be processed by a machine. Note, however, that evidence may come from sources with different levels of certainty and validity, depending on the method of collection or generation of such evidence.
- **EvidenceProcessing**: In our model, we assume that evidence, although it is associated to the effect of actions, does not directly stem from them. Instead, evidence is originated or collected by means of an **EvidenceProcessing** element. In this way, we model the fact that there may not exist a perfect correlation between the effects or consequences of actions and the evidence associated with them. The **EvidenceProcessing** element makes this difference explicit. With the inclusion of this element in our metamodel, we emphasise that the method of collection and processing of evidence is as important as the evidence itself. For this reason, there should also be evidence associated with each **EvidenceProcessing** element, describing how it works. Such evidence may be used by a metric during the evaluation process.
- **Metric:** We define this as an evaluation method for assessing the level of satisfaction of a non-functional property in a quantitative or qualitative way, on the basis of evidence and contextual criteria. Metrics can be of two types: **BaseMetric** for metrics that use evidence as inputs for their calculations, and **CompoundMetric** for aggregated metrics that are defined as a function of other metrics. Aggregated metrics may rely on auxiliary metrics that are not associated with any property and that are defined solely for facilitating the definition of the parent metric. In both cases, metrics may use **Criterion** elements for guiding the evaluation with respect to the context of the metric. This element has the following fields:
  - **Scale**: This field describes the type of measurement scale used in this metric. The scale can be either nominal, ordinal, interval or ratio.
  - **Unit**: This field represents the measurement unit adopted as standard for measuring the property. The definition of a measurement unit is only necessary in the case of quantitative metrics.
  - **Constraints**: This field conveys the contextual constraints that may affect the application and validity of the metric.
- **Criterion**: This element captures all the contextual input that may constrain what should be measured by the metric, such as regulation, best practices, organisational policies and contracts, and stakeholders' preferences. It could be the case that one could define different metrics for the same property. The assessment methodology for each metric will depend on the contextual input given for the metrics evaluation. The **Criterion** element will be the responsible of conveying such contextual information.

The intention behind this metamodel is to be used as part of the process of elicitation and evaluation of accountability properties in a cloud context. Hence, the stakeholder who is interested in assessing such properties would be the one that takes the role of owner of the model described using this metamodel. Each particular model defined using this language reflects the viewpoint of the model owner with regard to the context of application. Customization of models to specific situations is then done in different ways:

- Decomposition and interlinking of properties: the modeler can freely identify the goals and their associated properties, which can be further decomposed into other subproperties or interlinked through influence relations.
- Modelling of entities and their actions: Entities and actions can be modelled with the level of abstraction desired by the model owner, as the metamodel simply dictates that entities perform actions.
- Identification of meaningful evidence sources: the **EvidenceProcessing** element is used to model the sources of evidence that stem from the effect of actions.
- Definition of different metrics in terms of evidence and criteria: the possibility of defining different metrics for the same property is another characteristic that supports the customisation of models. Thus, the context and preferences of the model owner with regard to evaluation of properties can be reflected. Each metric would have different sources of evidence and criteria.

4.3 Modelling the Transparency Attribute

In order to illustrate our proposal, we show how one particular attribute for accountability, namely Transparency, could be (partially) modelled from its definitions given before (see section 3.1). From a high-level viewpoint, a transparency metric would measure the susceptibility of an organization's policies and procedures regarding data protection to be inspected by relevant parties (such as data subjects), as well as the quality of the transparency processes held in place by the organization. Figure 6 shows a model of the Transparency attribute using our metamodel.

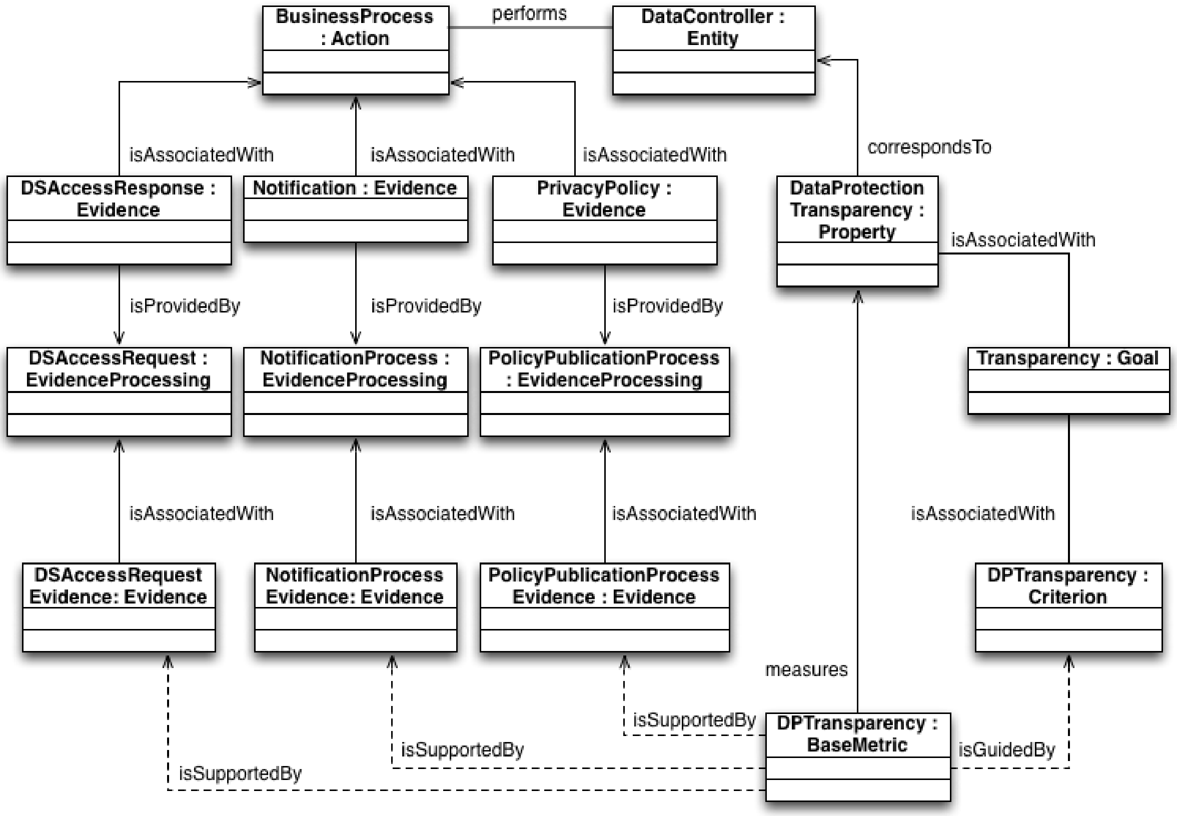
BusinessProcess : Action
performs
DataController : Entity
isAssociatedWith
isAssociatedWith
isAssociatedWith
correspondsTo
DSAccessResponse : Evidence
Notification : Evidence
PrivacyPolicy : Evidence
DataProtection Transparency : Property
isAssociatedWith
isProvidedBy
isProvidedBy
isProvidedBy
DSAccessRequest : EvidenceProcessing
NotificationProcess : EvidenceProcessing
PolicyPublicationProcess : EvidenceProcessing
Transparency : Goal
isAssociatedWith
isAssociatedWith
isAssociatedWith
isAssociatedWith
DSAccessRequest Evidence: Evidence
NotificationProcess Evidence: Evidence
PolicyPublicationProcess Evidence : Evidence
DPTransparency : Criterion
measures
isSupportedBy
DPTransparency : BaseMetric
isSupportedBy
isGuidedBy
isSupportedBy

Figure 6: Model of the Transparency attribute

In this example, the high-level goal is represented by the **Transparency** element. This goal could have associated several properties related to Transparency. In this case, we are referring to transparency with respect to data protection (represented by the element **DataProtectionTransparency**), as we are dealing with the treatment of personal data. This property is defined upon an organization that acts as data controller (since it determines the purposes and means of the processing of personal data). In other words, a metric for this property would evaluate how transparent this organization (i.e., the **DataController** element) is with respect to data protection. In this example, the actions of the **DataController** are subsumed into one **Action** element and called **BusinessProcess**. One might want to be more specific and could model particular business processes, but in this case, it is not necessary.

In order to achieve the property of transparency, the **DataController** must implement and demonstrate the application of certain practices that contribute to enhance its transparency, as defined in section 3.1. We identify three practices or transparency processes:

- Informing stakeholders about data protection policies and their implementation practices
- Notification in case of policy violation
- Responding to data subject access requests

These practices are directly mapped in our model example to the following **EvidenceProcessing** elements:

- **PolicyPublicacionProcess**: This element represents the internal procedures of the DataController towards the publication and communication of data protection policies to the relevant stakeholders. This element has associated two **Evidence** elements:
    - **PrivacyPolicy**: This **Evidence** is produced by the **PolicyPublicationProcess**; that is, the result of this process is an examinable description of the data protection policy that can be accessed by relevant stakeholders. This element by itself could not be relevant to the **DataProtectionTransparency** property. That is, individual privacy policies are not assessed by a Transparency metric as transparency is focused on making the policies known. In this case, only the existence of these elements could be assessed. However, the contents of the privacy policy could be interesting for measuring other properties such as Compliance of particular policies.
    - **PolicyPublicationProcessEvidence**: This instance of **Evidence** is associated with the transparency process that publishes privacy policies and describes its nature and characteristics. For example, it could answer questions such as "*Are all the policies published? Are the policies consistent with the real procedures in practice? Who asserts this consistency? Is it self-asserted or certified by a trusted party?*". The answers to these questions are the aspects that may influence the definition of a Transparency metric and its evaluation.
- **NotificationProcess**: This element represents the internal practices of the **DataController** with respect to notification to the relevant stakeholders about any violation of data protection policies. This element has associated two **Evidence** elements:
    - **Notification**: This element represents the **Evidence** generated by the **NotificationProcess** in case of a policy violation.
    - **NotificationProcessEvidence**. This element represents a description of the nature of the process of notification. That is, it answers questions such as "*Does a notification process exist? Are the means of notification appropriate? Are notifications consistent with privacy policies? Who asserts this consistency? Is it self-asserted or certified by a trusted party?*". These answers are the aspects that may influence a Transparency metric.
- **DataSubjectAccesRequestProcess**: This element represents the internal procedures of the **DataController** for permitting data subjects to request access to their data and for properly responding to such requests. This element has associated two **Evidence** elements:
    - **DataSubjectAccessResponse**: This element is the evidence representing the response generated by the **DataSubjectAccessRequestProcess** in case of an access request from a data subject.
    - **DataSubjectAccessRequestEvidence**: This element represents a description of the characteristics of the process for permitting data subject access requests. That is, it answers questions such as "*Does a process for data subject access requests exist? Is this process accessible to data subjects? Is it consistent with privacy policies? Who asserts this consistency? Is it self-asserted or certified by a trusted party?*" These answers are the aspects that may influence a Transparency metric.

Hence, it is the **Evidence** elements associated to the **EvidenceProcessing** elements, and not the evidence produced by them, the ones that are evaluated by the **DataProtectionTransparency** metric. On the other hand, the evidence generated by these processes could be evaluated by metrics for other attributes (for example, the **PrivacyPolicy** evidence could be evaluated by a Compliance metric, in order to assess whether the practices and rules established in the privacy policies are in line with the regulations of application).

As an illustration, based on the existence of the transparency processes that stem from the definition of Transparency (publication of policies, notification, or permitting data subject access requests), a naive qualitative metric for **DataProtectionTransparency** is shown in Table 1.

Level	Description of the level
None	No transparency processes are implemented by the Data Controller
Low	One transparency process is implemented by the Data Controller
Medium	Two transparency processes are implemented by the Data Controller
High	All the transparency processes are implemented by the Data Controller

Table 1: Naive example of a Metric for Transparency

The **DataProtectionCriterion** is the element that conveys the stakeholders' criteria with regard to that particular metric. In this case, the stakeholder is only interested in counting with the existence of transparency processes. Note that this metric does not evaluate the quality of these processes (and hence, this is why we consider this metric as naive).

A more complex metric could be as follows. This metric will only count the existence of a transparency process if it has been audited by a trusted third party, such as a government agency or an IT audit firm. Moreover, the metric's owner gives different weights to each transparency process: 0.5 to the publication of policies, 0.3 to the notification and 0.2 to the handling of data subject access requests. Note that one may be tempted to produce a formula such as:

$$0.5*TP_1 + 0.3*TP_2 + 0.2*TP_3$$

where TP_1, TP_2 and TP_3 can take values 0 or 1 if the transparency process is implemented (respectively, publication of policies, notification, or permitting data subject access requests). Such a formula could give the impression of having defined a metric with an interval or ratio scale from 0 to 1. However, in reality there is no real meaning for the differences between the possible values. Thus, a more valid approach could be to define an ordered scale such as the one in Table 2, which still conveys the same intention from the stakeholder:

Level	Description of the level
0	No transparency processes are implemented by the Data Controller
1	Only a process for data subject access requests is implemented by the Data Controller
2	Only a process for notification is implemented by the Data Controller
3	Either the process for publication of policies or the processes for notification and data subject access requests are implemented by the Data Controller
4	The processes for publication of policies and data subject access requests are implemented by the Data Controller
5	The processes for publication of policies and notification are implemented by the Data Controller
6	All transparency processes are implemented by the Data Controller

Table 2: Second example of a Metric for Transparency

The definition of the metric could be a bit more sophisticated, of course, not just covering which transparency processes are implemented in a binary way (i.e., yes or no), but assuming that there should be something implemented and including some judgement about the degree to which some aspects are implemented. For example, all the questions we defined before with respect to the characteristics of the transparency processes could be considered for defining criteria for the evaluation.

An open question is how to derive quantitative metrics from inherently qualitative attributes. From a strict point of view, one cannot simply assign a number to a quality value and perform operations. In that case, it is preferably to analyse the intended formula and produce a metric with an ordered scale, as in this example. This scale can be then more complex (i.e., with more levels) and still be valid. Other approaches, such as the use of fuzzy logic techniques could be applied.

Finally, it can be observed that a different definition of transparency could lead to a different model; that is the reason why we consider that a first requirement towards creating metrics is agreeing on a clear, concise and stable definition of the property to be measured, so that an appropriate model can be defined.

5 Conclusion and Future Work

This deliverable lays the foundations for development of metrics in the context of the A4Cloud project. It includes: (i) the definition of basic concepts for developing metrics, (ii) an analysis of the accountability attributes from the metrics perspective, and (iii) a metamodel for describing such attributes and their metrics.

The analysis of accountability attributes has helped us to refine the concepts involved in the definitions and to identify plausible sources of evidence in order to support the evaluation of such attributes. The results of the analysis will be fed to the Conceptual Framework in order to achieve a consensus regarding definitions of the accountability attributes. With respect to the metamodel for accountability metrics, it constitutes the first step in the metrics elicitation process. The metamodel serves as a language for describing the accountability attributes in terms of entities and activities among them. Moreover, it also allows us to describe the sources of evidence involved in those activities and to identify the evidence elements that can be used to support metrics. Thus, this metamodel is a valuable tool for guiding the process of defining metrics.

There are pending issues that need to be tackled. First of all, we are planning to use the metamodel for all the accountability attributes and not only for Tansparency, which we have used in this document as an example. Also, we must define specific mechanisms for conveying stakeholders' criteria for evaluating such attributes in the metamodel for metrics for accountability attributes. Currently, they are only described in an abstract manner, but tangible means should be defined. In the case of privacy preferences, languages such as the Primelife Policy Language (PPL) will be explored. Another aspect that we will cover in the next deliverable is the study of techniques for aggregation and composition of metrics, and specially, in the case of qualitative ones. Special attention will be given to techniques that also consider (i) the weighting of individual attributes and, (ii) the use of fuzzy numbers to operate with qualitative metrics.

6 References

[1] A4Cloud project. Description of work (DoW)

[2] A4Cloud project. MS:C-2.1 – Scoping report and initial glossary. December 2012.

[3] A4Cloud project. MS:C-2.2 – Initial framework description report. March 2013.

[4] A4Cloud project. MS:C-5.1 – Initial report on metrics for accountability. February 2013.

[5] A4Cloud project. QMUL/TiU. Remedies and sanctions (internal report). July 2013.

[6] A. Abran. Software Metrics and Software Metrology. 2010.

[7] I. Agudo, C. Fernandez-Gago, and J. Lopez, "A Model for Trust Metrics Analysis," in Trust, Privacy and Security in Digital Business, S. Furnell, S. K. Katsikas, and A. Lioy, Eds. Springer Berlin Heidelberg, 2008, pp. 28–37.

[8] I. Agudo, C. Fernandez-Gago, J. Lopez, A scale based trust model for multi-context environments, Computers & Mathematics with Applications, Volume 60, Issue 2, July 2010

[9] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro, "A Security Analysis of Amazon's Elastic Compute Cloud Service", in ACM SAC 2012.

[10] R. Bayardo and R. Agrawal. Data Privacy through Optimal k- Anonymization Proceedings of the 21st International Conference on Data Engineering, 2005; 217–28.

[11] B.D. Bernheim, and M. Whinston M. "Incomplete contracts and strategic ambiguity" Amer. Econ. Rev., 88, 902–932. 1999.

[12] S. Berthold and R. Böhme, "Valuating Privacy with Option Pricing Theory," in *Economics of Information Security and Privacy*, T. Moore, D. J. Pym and C. Ioannidis, Red., Springer, 2010, pp. 187-209.

[13] E. Bertino, D. Lin, and W. Jiang, "A Survey of Quantification of Privacy Preserving Data Mining Algorithms".

[14] J. Bull, and J. Watson. "Evidence disclosure and verifiability", Journal of Economic Theory, Volume 118, Issue 1, September 2004, Pages 1-31, ISSN 0022-0531.

[15] C. Castelfranchi and R. Falcone. Trust is much more than subjective probability: Mental components and sources of trust. Proc. of the 33rd Hawaii Int. Conf. on System Sciences (HICSS2000). Vol. 6

[16] C. Castelluccia, P. Druschel, S. Hübner et al. Privacy, Accountability and Trust - Challenges and Opportunities, ENISA. 2011.

[17] D. Chaum, "The Dining Cryptographers Problem: unconditional Sender and Recipient Untraceability," *Journal of Cryptography,* vol. 1, nr 1, pp. 65-75, 1988.

[18] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM,* vol. 24, nr 2, pp. 84-90, 1981.

[19] H. Chen, H. Wu, X. Zhou, and C. Gao, "Reputation-based Trust in Wireless Sensor Netwoks". In Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07), 2007.

[20] Centre for Information Policy Leadership (CIPL), "Demonstrating and Measuring Accountability - A document for discussion", Oct 2009, Hunton & Williams.

[21] Centre for Information Policy Leadership (CIPL), "Implementing Accountability in the Marketplace – A Discussion Document. Accountability Phase III – The Madrid Project". November 2011

[22] The Center for Internet Security, "The CIS Security Metrics v1.1.0", November 2010.

[23] S. Clauss, "A Framework for Quantification of Linkability Within a Privacy-Enhancing Identity Management System," i *Proceedings of the ETRICS*, 2006.

[24] Cloud Industry Forum, "Code of Practice for Cloud Service Providers," http://www.cloudindustryforum.org/code- of- practice/code- of- practice.

[25] C. Diaz, S. Seys, J. Claessens, and B. Preneel, "Towards measuring anonymity," in *Workshop on Privacy Enhancing Technologies*, 2002.

[26] C. Dwork, "Differential Privacy," in 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP), 2006.

[27] ECRYPT FP7 Project. "ECRYPT II Yearly Report on Algorithms and Keysizes (2010-2011)", http://www.ecrypt.eu.org/

[28] K. E. Emam, F. J. Dankar, "Protecting Privacy Using k-Anonymity". Journal of the American Medical Informatics Association. v. 15, no. 5, Sep/Oct 2008.

[29] ENISA, "Procure Secure - A guide to monitoring of security service levels in cloud contracts", 2012.

[30] EU Parliament and EU Council. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.

[31] EU Parliament and EU Council. DIRECTIVE 2002/58/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications) -- As amended by Directive 2009/136/EC.

[32] EU Parliament and EU Council. Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), Brussels, 25.1.2012, COM(2012) 11 final, 2012/0011 (COD)

[33] J. Feigenbaum, A.D. Jaggard, and R.N. Wright. "Towards a Formal Model of Accountability", In Sean Peisert, Richard Ford, Carrie Gates, and Cormac Herley, editors, NSPW , page 45-56. ACM, 2011

[34] N. E. Fenton and Pfleeger. Software metrics: A rigorous and practical approach. 2nd edition. 1998

[35] N. E. Fenton, R. W. Whitty and Y. Iizuka. Software quality assurance and measurement: a worldwide perspective. 1995.

[36] Federal Information Processing Standards, “FIPS 140-2: Security requirements for cryptographic modules” 2001.

[37] E. Filiz-Osbay and E.Y. Osbay. Effect of an Audience on Public Goods Provision. May 2012

[38] M. Franz, B. Meyer, and A. Pashalidis. “Attacking unlinkability: The importance of context”. In Privacy Enhancing Technologies (pp. 1-16). Springer Berlin Heidelberg. 2007.

[39] D. Giry, and BlueKrypt - Cryptographic Key Length Recommendation - v 26.6 - October 7, 2011. http://www.keylength.com/

[40] I. Goldberg and D. Wagner, “Randomness and the Netscape Browser, “ January 1996 Dr. Dobb's Journal.

[41] D. S. Herrman. Complete Guide to Security and Privacy Metrics. 2007.

[42] M. Hildebrandt. Biometric Behavioural Profiling and Transparency Enhancing Tools. FIDIS Project Deliverable 7.12. 2009.

[43] C. Hood, and D. Heald, Transparency: The key to better governance? (No. 135). Oxford University Press. 2006.

[44] IEEE Std. 982.1-1988 – IEEE Standard Dictionary of Measures to Produce Reliable Software. 1998.

[45] IEEE Std. 982.2-1988 – IEEE Guide for the Use of IEEE Standard Dictionary of Measures to Produce Reliable Software. 1998

[46] F. Innerhofer et al., “An empirically derived loss taxonomy based on publicly known security incidents,” in Proc. of Intl. Conf. on Availability, Reliability and Security (ARES), 2009

[47] ISO/IEC 15939:2007 – Systems and software engineering – Measurement process. 2007.

[48] ISO/IEC 27004:2009 (E) – Information Technology – Security techniques – Information Security Management – Measurement. 2009

[49] A. Jøsang and R. Ismail. The Beta Reputation System. In Proceedings of the 15th Bled Electronic Commerce Conference, Bled, Slovenia, June 2002

[50] R.E. Kalman. "On the General Theory of Control Systems", Proc. 1st Int. Cong. of IFAC, Moscow 1960 1 481, Butterworth, London 1961

[51] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, The EigenTrust Algorithm for Reputation Management in P2P Networks, In Proceedings of the Twelfth International World Wide Web Conference, 2003.

[52] M. Kantarcioglu, J. Jin and C. Clifton. When do data mining results violate privacy? In: Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 599–604. Seattle, WA (2004). URL http://doi.acm.org/10.1145/1014052.1014126

[53] R. Levien and A. Aiken, “Attack-resistant trust metrics for public key certification,” in Proceedings of the 7th conference on USENIX Security Symposium - Volume 7, Berkeley, CA, USA, 1998, pp. 18–18.

[54] N. Li, T. Li, and S. Venkatasubramanian, ”t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,” i *Proceedings of the IEEE 23rd International Conference on Data Engineering*, 2007.

[55] J. Luna, R. Langenbergand and N. Suri, “Benchmarking Cloud Security Level Agreements Using Quantitative Policy Trees” In Proc. of the ACM Cloud Computing Security Workshop. 2012.

[56] D. Ma and G. Tsudik. A New Approach to Secure Logging. ACM Transactions on Storage (TOS), Volume 5 Issue 1, March 2009

[57] A. Machanavajjhala, D. Kifer, J. Gehrke och M. Venkitasubramaniam, ”l-diversity: Privacy beyond k-anonymity,” *ACM Transactions on Knowledge Discovery from Data,* vol. 1, nr 1, March 2007.

[58] Malicious-and Accidental-Fault Tolerance for Internet Applications. IST Research Project IST-1999-11583. 1 January 2000 - 28 February 2003

[59] P. K. Manadhata and J. M. Wing. An Attack Surface Metric. IEEE Transactions on Software Engineering, Volume 37, Issue 3, 2010.

[60] D.H. McKnight and N. L. Chervany. The Meanings of Trust. Technical Report 96-04, MISRC Working Paper Series, University of Minesota, Management Information Systems Research Center, 1996.

[61] P. Mell, K. Scarfone and S. Romanosky. A Complete Guide to the Common Vulnerability Scoring System Version 2.0. June 2007.

[62] J. Mylopoulos, L. Chung, S. Liao, H. Wang, and E. Yu. Exploring alternatives during requirements analysis. Software, IEEE, 18(1), 92-96. 2001

[63] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *Proceedings of the IEEE 29th Symposium on Security and Privacy*, Oakland, CA, USA, 2008.

[64] New Oxford American Dictionary

[65] NIST, National Vulnerability Database Version 2.2, http://nvd.nist.gov/

[66] National Institute of Standards and Technology. NIST SP 800-55 – Performance Measurement Guide for Information Security. 2008.

[67] OECD, “Guidelines for the protection of personal data and transborder data flows”. 1980.

[68] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[69] A. Pfitzmann and M. Hansen. Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management - A Consolidated Proposal for Terminology. 2010.

[70] PMBOK. IEEE Guide – Adoption of the Project Management Institute (PMI(R)) Standard A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide) – Fourth Edition. 2011.

[71] M. K. Reiter och A. D. Rubin, ”Crowds: anonymity for Web transactions,” *ACM Transactions on Information and System Security (TISSEC),* vol. 1, nr 1, pp. 66-92, 1998.

[72] D.J. Ryan and C. Heckman. "Two views on security software liability. Let the legal system decide," Security & Privacy, IEEE , vol.99, no.1, pp.70,72, Jan.-Feb. 2003.

[73] B. Schwarz, H. Chen, D. Wagner, J. Lin, W. Tu, G. Morrison and J. West. Model Checking An Entire Linux Distribution for Security Violations. ACSAC '05 Proceedings of the 21st Annual Computer Security Applications Conference.

[74] A. Serjantov and G. Danezis, Towards an information theoretic metric for anonymity,” i *Workshop on Privacy Enhancing Technologies*, 2002.

[75] C. E. Shannon. A Mathematical Theory of Communications. *Bell System Technical Journal,* vol. 27, pp. 379-423, 623-656, 1948.

[76] S. Siegel,“Non parametric statistics,” The American Statistician, vol.11, no. 3, pp. 13–19, 1957.

[77] N. Siegmund. Measuring and Predicting Non-Functional Properties of Customizable Programs. PhD Thesis. 2012

[78] S. Steinbrecher and S. Köpsell. Modelling Unlinkability. In *Workshop on Privacy Enhancing Technologies*, 2003.

[79] S. Stevens, “On the theory of scales of measurement,” Science, vol. 103, no. 2684, pp. 677–680, 1946.

[80] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, nr 5, pp. 571-588, 2002.

[81] G. Tóth, Z. Hornák and F. Vajda. Measuring Anonymity Revisited. In *Proceedings of the 9th Nordic Workshop on Secure IT Systems*, 2004.

[82] G. Watson. “Two Faces of Responsibility.” Philosophical Topics 24: 227–248. 1996

[83] D.J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, G.J. Sussman. Information accountability. Communications of ACM 51(6), p. 87, June 2008.

[84] L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer e-commerce communities. In IEEE International Conference on E-Commerce, 2003. CEC 2003, 2003, pp. 275 – 284.

[85] J. Zabczyk. "Mathematical Control Theory: An Introduction", Birkauser Boston, 1992.

[86] Y. Zhang and A. Juels. Cross-VM Side Channels and Their Use to Extract Private Keys. In Proceedings of the 2012 ACM conference on Computer and communications security, October 2012.

7 Appendices

In MS:C-5.1 [4], an initial set of attributes that influence accountability and metrics for measuring them was provided. However, a bottom-up approach was followed due to the lack of a clear understanding of accountability and its related concepts at the moment of preparation of such document. That is, attributes and associated metrics were proposed, based on their assumed influence in accountability, but without a clear direction on where to focus the analysis. Further developments in the Conceptual Framework, such as stable definitions of accountability and the identification of a core set of attributes of accountability, implied a change on the approach for metrics elicitation. At that moment, we opted for a top-down approach for the elicitation of accountability metrics, starting on the core accountability attributes, as explained in Section 4. The following lists of attributes and associated metrics are extracted from MS:C-5.1.

7.1 Appendix 1: Preliminary Collection of Attributes Relevant to Accountability

This section provides a preliminary list of attributes (or non-functional properties) that influence accountability to a certain extent. These attributes are especially relevant in the A4Cloud framework. At the end of this section, we provide a summary of the attributes that we have identified in Table 3.
We categorize these attributes in three main areas:

- Privacy attributes
- Security attributes
- Cloud-specific attributes

Note that risk, trust and reputation are not considered attributes per se, but concepts that can be applied to attributes in order to aid the decision-making process. For instance, “Risk of not satisfying attribute X” or “Reputation of the cloud provider of fulfilling attribute X” are attributes that can be measured with metrics similar to the ones presented in section 5.

7.1.1 Privacy Attributes

The relation between privacy and accountability is complex and materializes in several ways. For example, digital transactions are easily recorded by service providers and third parties, leading to increasing tracking and profiling of individuals. This fact clashes with several aspects of both the current ([30][31]) and the proposed ([32]) EU data protection law, such as the right of individuals to recieve specific information regarding the processing of their personal data (i.e the purpose or purposes of processing) or the right to be forgotten). Privacy and accountability are in this case related concepts, as providers of IT systems (i.e. data-collecting parties) should be accountable for the protection and treatment of the personal information they gather.

This aspect is just a particular case of a more general relation between privacy and accountability. Organizations should be accountable for the degree of conformity with their privacy-related obligations. These obligations could be either of regulatory, contractual or ethical nature, among others. Measuring the degree of conformity with these obligations is very important to assess the level of accountability of an organization.

A minor instance of the relation of privacy and accountability (within the A4Cloud context) can be seen also in the trade-off between anonymity and accountability of users. If we want to hold users accountable of their use of IT resources, some compromises must be made regarding their privacy. At one end of the spectrum, a fully anonymous system difficulties accountability, as it is not possible to trace the identity of users who misbehave; in this case the accountability of this system will presumably be low. At the other end, a fully accountable system difficulties anonymity. Therefore, the level of users’ privacy in a system influences its accountability and viceversa, but it is not clear to what extent. However, this aspect is further from A4Cloud scope.

7.1.2 Security Attributes

7.1.2.1 Availability

Given that the current lifestyle is largely based upon automated processing, the inability to access or use personal data can have consequences that range from a minor inconvenience to life threatening consequences. For example, when a bank ATM network becomes unavailable, it will simply result in discontent of card owners; by contrast, if a hospital system is unable to access patient data it may have very detrimental effects. It should be noted that data protection is not limited to protecting privacy, but is also concerned about broader goals such as assuring that data is notably processed "fairly and lawfully" [30] and that this processed in a secure manner [30]. In this regard, availability can be just as important as confidentiality and integrity of data. Additionally, data subjects are entitled, for example, to access, update or erase their data under certain circumstances. Therefore, the system supporting the personal data must be available. For all these reasons, the ability to measure availability of a system is relevant to accountability.

7.1.2.2 Incident Response

Incident response practices are closely linked to accountability since they describe how a provider responds to incidents that may affect the confidentiality, integrity and availability of data. This notably includes:

- How quickly the provider informs a client of an incident.
- How quickly the provider implements measures to remedy the incident.

These elements by their very nature contribute to the chain of accountability. Incident response can more generally be linked to the current trend to force data controller to notify personal data breaches both in the US and Europe (see article 4 in [31] and article 32 in [32]).

7.1.2.3 Data Lifecycle Management

Data lifecycle management describes the provider's data handling practices [28]. The following attributes from this topic are identified:

- Storage quality: how reliable storage is or how good the backup mechanisms are. This can be viewed as a special case of availability.
- Portability: how well data can be exported back to the client in case of necessity. Portability is considered as a potential legal obligation in upcoming data protection regulation [32].
- Destruction quality: how well/quickly all copies of an object are removed from the system. Data protection rules usually mandate that data should be kept (in identifiable form) "*for no longer than is necessary for the purposes for which the data were collected or for which they are further processed*" [30]. This attribute is often overlooked but is an integral part of accountability.

7.1.2.4 Vulnerability Management

Vulnerability management reflects how an organisation manages, quantifies and responds to information security threats.
The main aspects that we want to measure with regard to vulnerabilities are:

- The quality of proactive measures in place: frequency and strength of vulnerability scans.
- The severity of vulnerabilities, that is, an indication of the potential damage to the assets that could be exposed by the vulnerability. This aspect can be measured through different perspectives (availability, integrity, risk, etc.).
- How well an organisation reacts to vulnerabilities by implement counter measures.

7.1.2.5 Data Confidentiality

In certain cases, cloud providers act as stewards of users' personal data. A cloud provider may state that it will ensure the confidentiality of the data under its custody, but it is necessary to assess to which extent this statement is true, since it is important to hold the cloud provider accountable of any data disclosure of data under its custody. Therefore, a cloud provider should be accountable with respect to the confidentiality of the data that it stewards. For these reasons, we see the **level of data confidentiality** as an attribute of the cloud system that influences accountability.

7.1.2.6 Cryptographic Key Management

Cryptography provides tools most commonly used in the cloud to protect the confidentiality and integrity of data in transit or at rest. Yet, cryptographic mechanisms may not provide the expected security guaranties notably if:

- The confidentiality of cryptographic keys is not protected (in particular for protecting data at rest).
- The use of cryptographic keys is not well controlled (key management rules are not enforced).
- The cryptographic mechanisms are out-dated or use inadequate key-lengths.
- The cryptographic mechanisms are not well implemented.

Some of these problems reflect aspects of the Key Management System, and can be seen as measurable attributes, such as:

- **Key strength**: Indication of how strong a key is from the cryptographic point of view.
- **Device security**: Indication of the security level of a device for cryptographic purposes.
- **Key exposure**: Indication of the level of protection of cryptographic keys

7.1.2.7 Log Management

Providing logs and traces of events is an important part of security. The analysis of logs themselves can provide some security metrics that are context specific and which are not in the scope of this work. However if logs are poorly implemented, unreliable or potentially unprotected, it will have a detrimental effect on accountability. Therefore, some attributes related to log management, such as **log unalterability** and **log accuracy** should be measured; nonetheless, both concepts would need further formalization to be used in practice.

7.1.2.8 Non-repudiability, imputability, traceability

Traceability is term commonly used in logistics and supply chain management to describe the ability to trace information related to goods during their production. It can be extended to information management to describe the ability to track the complete set of operations that were performed on a specific set of data.

Imputability describes the ability to ascribe actions (access, modification or deletion of data) to a certain person or entity. As such it can be viewed as a special case of traceability.

Non-repudiation describes the ability for an entity to produce data elements in a way their origin cannot be subsequently refuted.

Depending on the case, measuring some of these attributes can provide a measure of how accountable a system is, since it can help to demonstrate good stewardship of data. One of the important points is also to establish how trustworthy or verifiable these attributes are.

7.1.3 Cloud-specific attributes

7.1.3.1 Elasticity

Elasticity describes how well a cloud provider adapts to its clients demand in resources (e.g. adding compute instances, storage, etc.) according to stablished agreements (SLAs). As detailed in [28], the CSP may provide specific guaranties a SLA regarding elasticity, through the offering of guaranteed reserved capacity or burst tolerance over a period of time. This attribute can be seen as a specialized case of availability.

7.1.3.2 Data Isolation

An accountable cloud system must ensure that data is isolated among cloud tenants, in order to ensure its integrity and authenticity. Therefore, we propose to measure the following attributes:

- **Present data isolation**: Cloud tenants should not be able to read or write data from other clients that is currently transmited, stored or processed. If this property is not assured, in particular in the case of writing, accountability could be compromised.
- **Deleted data isolation:** Cloud tenants should not be able to read previously deleted data from other clients.

7.1.3.3 Data location

The location of data is an important aspect from the legal and regulatory viewpoint, even more so since data in cloud environments is not static and may vary during its life cycle. Therefore, data location is an attribute that influences accountability.

7.1.4 Summary

This table summarizes the attributes identified in this section:

Conceptual area	Attribute	Rationale
Privacy	Data stewardship	Describes how well an IT provider fulfills privacy requirements with respect to data stewardship.
	Influence of privacy on accountability with respect to users	If we want to hold users accountable of their use of resources in an IT system, some compromises must be made regarding their privacy. The idea is that a fully anonymous user presumably cannot be held accountable.
Security	Availability	One of the 3 pillars of information security along with confidentiality and integrity, it is a natural part of data protection.
	Incident response	Describe the ability of a provider to respond to unexpected events, and to report back to clients.
	Data-life cycle management - Storage quality	Quality of storage backup mechanisms. Proper retrievability of data is important for guaranteeing an accountable system.
	Data-life cycle management - Portability	Has been proposed in future EU data protection legislation, and reflects a growing concern of cloud customers regarding data stewardship.
	Data-life cycle management – Destruction quality	Data that is supposed to be deleted must be truly removed from the system. Required by data protection legislation.
	Vulnerability management - Severity	Evaluate the threats that may affect a system, and prioritize remediation actions.
	Vulnerability management – Detection	Cloud providers should be able to describe how well they proactively detect vulnerabilities in their systems.
	Vulnerability management - Remediation	Cloud providers should be able to describe how quickly they remediate potential threats to their systems.
	Data confidentiality	A cloud provider should be accountable with respect to the confidentiality of the data that it stewards
	Key Management – Key strength	The cryptographic strength of a key is one of the factors that influence the confidence that the user can have in a cryptographic procedure.
	Key Management – Device security	Indication of the trust in the security of a cryptographic device. Accountability sometimes relies on this property (e.g. a TPM)
	Key Management – Key exposure	Indication of the security of the method for storing keys. This attribute directly influences confidentiality, and therefore, accountability.
	Log unalterability	Logs must be unaltered in order to guarantee the accountability of the monitored system
	Log accuracy	Logs need to be accurate enough for enable a proper and meaningful interpretation
	Non-repudiability / imputability / traceability	These attributes allow quantifying how well the provider is able to give account of what happens to the data he receives from his clients.
Cloud-specific concepts	Elasticity	Elasticity is highly related to the fulfillment of SLA
	Present data isolation	Accountability could be compromised if data isolation is not guaranteed across cloud tenants
	Deleted data isolation	Cloud tenants should not be able to read previously deleted data from other clients.
	Data Location	The location of data is an important legal aspect, even more so since data in cloud environments is not static and may vary during its life cycle

Table 3: Summary of attributes related to accountability

7.2 Apendix 2: Review of Measurement Techniques for Non-Functional Attributes

7.2.1 Privacy metrics

Privacy-related concepts, such as anonymity or unlinkability, are defined in the project's glossary [2] [6]. Privacy metrics have been applied in anonymity networks, anonymity in databases, and unlinkability for individuals. The first documented privacy metrics in modern science have been applied to databases. Apart from the area of application, other dimensions for structuring privacy metrics are the *type of metrics*, e.g., qualitative vs. quantitative, relative vs. absolute, or the *type of evaluation*, e.g., by experts or by information theory. Most metrics presented in this section are quantitative and absolute, and directly or indirectly relate to Shannon's information theory [75].

One of the simplest privacy metrics is the *anonymity set* [69]. It is a counting measure, i.e., the metric is determined by the number of set members. The set members are those individuals that could be the individual the adversary is looking for. The adversary will use all he knows about the target individual in order to exclude as many of the other individuals from the anonymity set as possible. Anonymity is preserved as long as there are several set members in the anonymity set. The target individual is identified if the adversary is able to reduce the set size such that just a single individual is left in the anonymity set. In this interpretation, anonymity becomes a binary property, i.e., either it is present or absent. The anonymity set size can also be seen as a discrete quantification of anonymity, i.e., anonymity is preserved better the greater the set size is.

Privacy Preserving Data Mining (PPDM) algorithms are procedures to extract relevant knowledge from large amounts of data while protecting at the same time sensitive information. In other words, searching in the data sets is possible without revealing the identities of the individuals who contributed to the data set with their personal data. In this case, we say anonymity in databases is preserved.

E. Bertino et al present in [13] a survey of privacy preserving data mining algorithms. Following their analogy, the key criteria for evaluating PPDM algorithms is:

- **Privacy Level:** Indicates how closely sensitive information that has been hidden can be estimated using different techniques
- **Data Quality:** Quality of raw data as wells as results from data mining after use of appropriate privacy preserving algorithm.

The same ideas used in the definition of the anonymity set can be used for data. Data privacy can be quantified based on the degree of uncertainty achieved by a PPDM algorithm. Another area of evaluation is whether the results of data mining or analysis violate privacy requirements. Most of the PPDM algorithms, transform the data to hide sensitive information. Such perturbation could lead to decrease in data quality. Therefore data quality metrics are important to evaluation of PPDM techniques. Further, the quality of data mining algorithms can also be evaluated through data quality metrics measuring key parameters of their results.

- Quality of Data resulting from PPDM techniques: Among the various parameters the most relevant for measuring data quality are *accuracy*, *completeness*, *consistency*. Accuracy is the measure of closeness of the original data set to the transformed set after application of PPDM techniques. Completeness is the degree of missed data in the transformed data set and consistency is the trueness of relationships among data elements after they are transformed.
- Quality of Data Mining Results: Quality of results derived from mining algorithms is another area of evaluation. This kind of metrics are to evaluate the data are used for its intended purposes. Information loss, fulfillment to intended use, are few measures against which a particular type of knowledge model would be evaluated.

7.2.2 Availability

There are many ways to define and measure availability [28]. We will examine the following methods:

- Target percentage of total operational time
- Target percentage of total requests
- Mean recovery time (MRT)
- Mean time between failures (MTBF)

[6] A4Cloud Glossary is currently under development

For any measurement of availability to be meaningful it is necessary to define precisely what the service is and what constitutes an "unavailability event". In turn defining an unavailability event requires specifying the following parameters of the measurement method as described in [28]:

- *The definition of failure to serve a single request*: what are the criteria that define that the service has failed (any specific "error codes" that express a failure or any other criteria defining that the requested function was non-operational)? Is there a specific threshold defining the maximum time for serving a response to a request, beyond which the request is considered as failed (regardless of its final delivery)?
- *The scope of the service*: Does it apply to the customer only or is it an average over all customers? Is it specific to a geographical region? Does it cover end-to-end fulfillments of the request between the client and the CSP, or does it stop at the nearest network connection point to the CSP?
- *A sample size:* this is the time or number of requests over which a percentage of failures should persists in order to define an "unavailability event" (for example, 50% of request failures during 2 minutes).

Additionally, as with most other security metrics described in the document, it is important to specify the **period over which the mean is calculated**. Measuring average unavailability over a very short period of time (minutes) does not have the same significance as measuring availability over a year where extreme events are smoothed out. Cloud service providers usually define a commitment period in their SLA, which specifies over which duration availability should be measured.

- Target percentage of total requests.

Based on the definition of a single request failure it is possible to define the *target percentage of total requests* measurement method as the ratio (T-F)/T where T is the total number of requests and F the number of request failures (over the measurement period). In this measurement method, the above definition of a "sample size" is not required. However a threshold defining the minimum number of requests needed to establish the measurement is required (otherwise a single failure can produce 0% of availability if F=T=1).

- Target percentage of total operational time.

A measurement of the *target percentage of total operational time* is the ratio (P-F)/P, where P is the duration of the measurement period and F is the sum of the duration of all unavailability events (i.e. taking into account the sample size).

- Mean recovery time (MRT)

The mean recovery time is the average time necessary to recover from an "unavailability event". The Cloud Service Provider may provide a Recovery Time Objective (RTO) defining the maximum acceptable delay for recovery from an availability incident. Cloud clients will then compare the MRT against the RTO. Consequently, a secondary indicator may be defined as the **percentage of availability incidents resolved within RTO**.

- Mean Time Between Failures (MTBF)

The mean time between failures is measured as the average time between two consecutive "unavailability events".

7.2.3 Incident Response

As described in [28], an incident is any event which falls outside of the normal operation of the service and which causes, or may cause, interruption or reduction in the quality of the service. As such, an incident is usually defined in terms of others metrics, such as "*if the availability falls below 90% for a user over a month*", or "*if a high level vulnerability is detected*". What can be measured is mainly how the CSP responds to and recovers from incidents. As a starting point it is usually necessary to characterize incidents through:

- *A severity level*: A classification of the incident according to a well-defined severity scale.
- *Time to respond*: The time between the notification of the incident and the implementation of a remedial response.

With these characteristics, it is possible to define the following metrics:

- Percentage of incidents (of severity S) resolved within a defined period.

The value (100 * R/T) where R is the total number of incidents (of severity S) resolved within a period of time P and T the total number of reported incidents (of severity S) over the same period.

- Percentage of incidents (of severity S) responded to within a defined period.

The value (100 * R/T) where R is the total number of incidents (of severity S) to which a response was provided within a period of time P and T the total number of reported incidents (of severity S) over the same period. In many scenarios, this metric is more significant than the previous one regarding the resolution of an incident.

- Mean time to report (incidents of severity S).

The mean time between the discovery of an incident (of severity S) and its reporting to the customer (if such a reporting is defined in the contract between the client and the provider).

- Mean time between incidents (of severity S).

This is the average time that elapses between two consecutive incidents (of severity S). This is similar to the MTBF defined for availability, but the scope of an "incident" is broader.
For these metrics to be comparable between cloud providers, there needs to be a common and precise definition of severity levels, which is rarely the case.

7.2.4 Data Lifecycle Management

As mentioned in section 4, data lifecycle management comprises the following attributes: storage quality, portability and destruction quality. Measuring how well data is stored by the CSP relates to availability (of storage resources). Portability can be tested by exporting to the client a representative subset of the data held by the CSP (taking into account the possible limitation imposed by data size, network, etc.). Measuring how data is destroyed is often overlooked but is important with regard to data protection rules at EU level, which require personal data to be kept for no longer than necessary for the explicit purpose (or purposes) for which it was collected.

The following measurement methods are can therefore be used to describe storage management:

- Restoration success ratio

The ratio S/T, where S is total number of restoration operations that were successful during a period P, and T is the total number of restoration operation that where conducted during the period P. A restoration operation is considered successful if it is *delivered, complete, passing an integrity check and conforming to a pre-defined format* [28]. Note that this metric is driven by cloud client demands: if few restoration operations are requested by clients the results will have low statistical value.

- Backup test frequency and results

A pair (T,S) formed by the total number T of restoration tests that were performed during a period P and S the total number of successful restorations (given a specific integrity test) verified during the period P. A restoration test should be understood as a restoration performed on all backed-up data (or on a randomly chosen representative sample of the data) in a controlled test environment different from production. Such a test can be performed automatically and independently of the cloud client's own "real" backup restoration demands.

- Average restoration speed

The average time needed to recover data from a backup. In some cases it might be appropriate to additionally report the "**average restoration speed of the first octet**", which might be less dependent on the size of the backed-up data.

- Age of the most recent recovery point

The time elapsed since the last point in time where the system is guarantied to be restorable from backups. This is usually contrasted with a recovery time objective (RTO), which is the maximum tolerated period during which data might be lost.

- Durability

Durability is the percentage of data (or objects) that are guaranteed not to be lost (or damaged) during a certain period of time. A durability of 99% of objects over a year means that on average 1 object out of 100 will be lost over a year. This measurement is often used by cloud providers, which do not necessary use "traditional" backup mechanisms, but rely on fault tolerant real-time replication of data.
To our best knowledge there is no measurement method defined to evaluate "**portability**", though we believe that this might be an interesting metric to develop. One indicator of portability is the speed of export.
To address the issue of deletion, it is important to emphasize that deletion does not only cover cases where an object is not accessible anymore by the client but that it should be understood as the effective and certain removal of an object from the system, including from redundant storage and backups. In a cloud environment the time necessary to fully remove all possible copies of an object

may be influenced by the probabilistic behaviour of data replication mechanisms. We propose to define the following metrics:

- Scheduled deletion failure rate:

What is the percentage of deletion requests executed during a period P that have been effectively complete under a delay D (e.g. 90% of deletions requests are in effective is less than 1 hour).

- Mean effective full deletion time

This is the average time that elapses between a client's request to delete data (or an object) and its effective removal from the system.

- Average time for X% of deletion requests to be implemented

Due to the nature of complex systems such as clouds, we might be interested to provide the delay D under which X percent of deletion requests are effective, choosing X as close as possible to 100% (e.g. "It takes on average 38.52 hours to reach a deletion success of 99,99%).

- Quality of destruction

It should be possible to rate how well data is deleted from a system. At the lowest level, data is simply dereferenced with actually being erased, while in the most sensitive environment data is removed with a process that guaranties that it cannot be recovered (e.g. degaussing or multi-pass rewrites). This could alternatively be expressed as the effort needed to recover data. Just like for metrics related to the location of data, it is usually impossible to strictly "prove" that data has been deleted and we must rely on the trustworthiness of the cloud providers to provide that information. Some testing is still possible, for example if a cloud client randomly requests access to data that should have been deleted to verify if it is still recoverable. The quality of destruction relates closely to the effective implementation of the right to be forgotten, which is currently being discussed under the proposed Data Protection Regulation.

7.2.5 Vulnerability Management

Evaluating and comparing vulnerability exposure is a non-trivial task. For example, if two distinct CSPs run vulnerability scans, then the fact that the first one reports 1 vulnerability and the second one reports 5 vulnerabilities does not say much about the relative strength of each platform alone. Many elements can influence the quality of a vulnerability scan:

- The number of vulnerabilities tested: running 10 tests has a lower risk of exposing vulnerabilities than running 200 (including the same 10)
- The quality of the tests: running tests that target vulnerabilities that are 10 years old or that mainly target another platform (OS) is less likely to expose vulnerabilities.

Additionally the risk posed by a vulnerability is a function of the affected assets, which may vary from customer to customer.

One of the key challenges is therefore to find a common baseline to conduct and compare vulnerability scan results. To our best knowledge this is still an open problem, and we propose the following ideas for further investigation:

- Start with a standardized database of vulnerabilities: The US National Vulnerability Database (NVD) [65], for example, provides a set of open vulnerabilities each with a specific description, a score according to a standard scheme (CVSS) [61] and a publication date. A common reference database is needed as starting point of reference for measurements.
- Define a reference period covered in the database: The scan should encompass only vulnerabilities published during the defined period (e.g. the last 3 years).
- Define the following values:
    - M: The total number of vulnerabilities that are relevant to the platform during the selected period.
    - T: The total number of vulnerabilities that were actually tested on the system.
    - V: The total number of vulnerabilities found by the test.

If P defines the total number of published vulnerabilities during the reference period (all platforms included), we have P > M >= T >= V. Over the reference period, we can define an indicator formed by two values: first the ratio T/M, which describes the proportion of relevant vulnerabilities the test covers, and second V the total number of vulnerabilities found (or alternatively V/T).

It should be noted this proposal provides a degree of comparison between CSP vulnerability exposures, but still has some limitations:

- Because of the use of a common standard database, it typically addresses “open” vulnerabilities from commercial off the shelf and open source software. CSP use specialized software that is likely to be partially out of scope of the database.
- Vulnerabilities scanners may report “false positives” either because the scanner does not have the ability to verify fully that the vulnerability is present [9] or because the cloud service provider has implemented additional controls that compensate for the vulnerability.

7.2.5.1 Scoring Vulnerabilities

When a vulnerability is found it is usually important to evaluate the risk associated with it. This notably requires taking into account the potential damage to the assets that could be exposed by the vulnerability. However, as noted in [28], this characteristic is usually unknown to the CSP (in IaaS and PaaS at least).

Instead, we may rely on a “base severity score” as a less precise indicator, such as the “Base Score” provided in the Common Vulnerability Scoring System Version 2.0 [61], which proposes a vulnerability score based on the following indicators:

- Access Vector: Can the vulnerability be exploited through local, adjacent or remote access?
- Access complexity: Does the attack require specialized access or condition (social engineering, DNS spoofing, etc.), or can the attack be exploited without any specific circumstances?
- Authentication: Does the attack require to be authenticated once, twice or more?
- Confidentiality impact: Is the confidentiality not affected, partially affected or fully affected.
- Integrity impact: Is the integrity not affected, partially affected or fully affected.
- Availability impact: Is the availability not affected, partially affected or fully affected.

A formula described in [61] provides a score between 0 and 10, often described through 3 levels:

- *“Low”* if the base score is less than or equal to 3.9
- *“Medium”* if the base score is in between 4.0 and 6.9
- *“High”* if the base score is greater than or equal to 7.0

Taking into account the limitations that were highlighted previously regarding vulnerability scans, we can provide some metrics that were notably detailed by [22]:

- Percent of systems with known high (res. medium or low) severity vulnerabilities

The total number of systems with at least one “high severity” (resp. medium or low) vulnerability divided by the total number of systems tested. This metric works in cases where a cloud services can be broken down into a set of well-defined systems (e.g. hosts or virtual machines in IaaS).

- Mean time to mitigate vulnerabilities (of severity level L)

The average time between the discovery of a vulnerability (by the security community) and the implementation of mitigating measures (by the CSP).

Some researchers [59] have proposed an alternative metric of vulnerability expressed as the “attack surface”, which roughly speaking is a measurement of the number of “openings” the system offers to an adversary (e.g. API.Calls, network sockets, data items). This has the advantage of not relying on a vulnerability-testing tool, but it is not clear yet how well this approach reflects the actual vulnerability of a system in practice. Other researchers have used automated model checking tools to discover potential vulnerabilities form source code, including full operating systems [73], which could also be used as a vulnerability metric.

7.2.6 Data Confidentiality

7.2.6.1 Encryption level

We propose the following measure to indicate the level of encryption of data in a cloud-based system:

Level	Description
0	Data is not cryptographically protected by the cloud provider.
1	Data is cryptographically protected in transit.
2	Data is cryptographically protected at rest and in transit.
3	Data is cryptographically protected even at execution time.

Table 4: Data confidentiality level

A system with Level 0 of data confidentiality does not use any cryptographic protection; it may, however, use other types of security measures, such as access control policies.

A system that achieves Level 1 protects data that is transmitted from and to the cloud provider. This kind of system achieves security against an eavesdropper, but data is in clear inside the cloud provider, and therefore, susceptible to insider attacks or security breaches.

Level 2 implies that data is protected also at rest. Proper mechanisms for key management need to be used. However, data should be decrypted before processing and could be accessed by malicious software or insiders.

In a system with Level 3, the cloud provider does not decrypt data prior processing because the cryptographic scheme enables the processing of encrypted data. This level could be achieved with the aid of Fully Homomorphic Encryption schemes, but current proposals are not viable in practice. However, for certain applications, such as secure auctions and e-voting, there are simpler homomorphic schemes that are efficient and usable, and could reach this level.

7.2.6.2 Confidentiality objective

We propose a measure to indicate the level of confidentiality achieved by a system regarding client data independently of the means used to achieve this objective (see Table 5).

Level	Description
0	Data confidentiality does not satisfy any of the above levels.
1	Data may be accessible by the cloud provider personnel for regular operational purposes, under the control of an authentication, authorization and accounting (AAA) mechanism.
2	Technical and organizational measures are in place so that data may only be accessible to privileged CSP personnel (administrators) for debugging or maintenance purposes, under the control of an AAA mechanism.
3	Technical and organizational measures are in place so that data is only accessible to privileged CSP personnel to respond to law enforcement or extraordinary requests made by the client, under the control of an AAA mechanism.
4	Data is encrypted by the client with cryptographic keys that cannot be ascertained by the provider.

Table 5: Level of Confidentiality

The last level represents the best possible protection for a cloud client, however it will limit the ability of cloud providers to process the data except for storage purposes.

7.2.7 Cryptographic Key Management

Cryptography provides tools most commonly used in the cloud to protect the confidentiality and integrity of data in transit or at rest. Yet, cryptographic mechanisms may not provide the expected security guarantees notably if:

- The confidentiality of cryptographic keys is not protected (in particular for protecting data at rest).
- The use of cryptographic keys is not well controlled (key management rules are not enforced).
- The cryptographic mechanisms are outdated or use inadequate key-lengths.
- The cryptographic mechanisms are not well implemented.

As a consequence there is no simple metric that can determine how “well” a system is protected by cryptography. Despite these difficulties, we can propose some indicators:

- Key length

There are many recommendations on the adequate cryptographic key length for cryptographic algorithms (see [39] for a summary). We can highlight the one proposed by the European Network of Excellence in Cryptology II [27], which distinguishes the following security levels:

Security Level	Symmetric Bit length	Protection
1	32	Attacks in "real-time" by individuals. Only acceptable for authentication tag size
2	64	Very short-term protection against small organizations Should not be used for confidentiality in new systems
3	72	Short-term protection against medium organizations, medium-term protection against small organizations
4	80	Very short-term protection against agencies, long-term protection against small organizations (until 2014)
5	96	Legacy standard level (2020)
6	112	Medium term protection (2030)
7	128	Long term protection (2040)
8	256	Foreseeable future

Table 6: Key Length

It is important to note that Table 6 provides only key length information for standardized symmetric encryption algorithms as an example, but ECRYPT also provides equivalences for asymmetric encryption, discrete logarithms, elliptic curves and hashes.

- Device security level

The FIPS 140-2 standard provides a well-known definition of the security level of a device used for processing cryptographic data [36]. Level 1 describes hardware with minimal security requirements, while on the other end of the scale, Level 4 describes hardware that has been hardened to resist advanced logical and physical penetration attempts. By adding a “level 0” describing software solutions to this scale we can construct a simple indicator of the level of security of cryptographic processing, which can be further distinguished by its intended functionality (encryption, message-authentication, signature, etc.). This metric is mostly relevant for core security components of a cloud system.

- Key exposure level

When encryption is used to protect the confidentiality of data at rest in the cloud, there are many approaches to the distribution and the use of cryptographic secrets that ultimately protect the data in the cloud. The cloud client can encrypt the data before it even reaches the cloud, with a key that is only known to the client. More often however the cloud provider will encrypt the data with a key that is under its control. We propose a novel indicator of **key exposure** to reflect the level of confidentiality afforded to cryptographic secrets, from a cloud client point of view:

Level	Description
1	Access to decrypted data or cryptographic secrets by the CSP is necessary to provide some functionalities of the service.
2	Access to decrypted data or cryptographic secrets is available to specific personnel of the CSP, for administrative or debugging purposes only.
3	Access to decrypted data or cryptographic secrets is available to specific personnel of the CSP, for administrative or debugging purposes only. It is governed by the principle of dual control and split knowledge.
4	Access to decrypted data or cryptographic secrets is available to specific personnel of the CSP in exceptional circumstances only. It is governed by the principle of dual control and split knowledge, under the supervision of a hardware security module.
5	Cryptographic secrets needed to decrypt the data are known to the cloud client only.

Table 7: Key Exposure Level

- Authentication level

The NIST standard 800-63 provides a well-defined set of levels to measure the quality of authentication mechanism, to which we can add a level 0 describing no authentication at all. At level 1, simple challenge response mechanisms are allowed and no identity proofing is required. At level 4, strong cryptographic mechanisms are required along physical tokens with a FIPS 140-2 level greater than 2.

- Entropy

Randomness can play an important role in some security mechanisms, most notably in cryptographic algorithms where secret keys should not be guessable by anyone. If random generators are predictable, they may create security holes [40]. Some standardized tests can be used to give indications regarding the quality of the output of random generators.

7.2.8 Log Management and Forensics

We focus our interest on qualitative assessments of logging facilities, with the following indicators:

- Average client log access time

Average time needed to provide logs, in response to a cloud client request. This usually refers to a subset of logs defined by contractual agreement.

- Log integrity level.

To what extent are the logs protected against alteration? Is the log facility provided a strongly redundant platform? Is it a WORM device (Write Once Read Many)? Does a trusted third party hold the log? Is the log secured by a cryptographic mechanism allowing the detection of alterations [56]?

- Log accuracy

What is the probability that an event is not logged by the facility (with write access) or that an event, which did not occur is logged?

7.2.9 Cloud-specific Metrics

7.2.9.1 Elasticity

Following a similar approach to [28], we propose to define the **elasticity ration**, a quantitative measure of elasticity as the ratio:

$$\frac{T-F}{T}$$

where F is the total number of failures of resource provisioning requests over a period P (the commitment period), and T is the total number of provisioning requests over period P.

As with availability, it is important to define what constitutes a provisioning failure. As a consequence, elasticity metrics may require the definition of additional parameters. When the system explicitly returns an error code indicating that a provisioning request could not be satisfied, such an event is clearly counted as a provisioning failure. However, it might also be necessary to consider that if a provisioning request is not answered within a specific maximum delay, this also constitutes a provisioning failure, just as we do for availability.

7.2.9.2 Data Isolation

Data isolation is about ensuring the confidentiality, integrity and availability of data between different cloud clients [28]. In traditional IT systems, isolation was partially enforced through the physical separation of resources dedicated to each client (disks, databases, operating systems, networks, etc.). In cloud systems, isolation relies on a more complex set of mechanisms, primarily the use of virtualization technologies and advanced authentication services.

When it comes to data isolation, we can ask the following questions.

- Can a cloud client read/modify a memory block, storage data or network packets produced by another client?
- Can a cloud client still read a memory block or storage data once another client has deleted it?

Recent research shows the additional risk of side channel attacks, whereby a cloud client can discover information about another client, including in particular the value of secret cryptographic keys, by observing the temporal behaviour of the system [85].

- Data isolation testing level.

To our best knowledge there are no metrics associated with data isolation in the cloud. As a first step, we propose to define the following indicator called data isolation testing level, which describes the level of testing that has been done by the cloud provider to assess how well data isolation is implemented:

Level	Description
0	No data isolation testing has been performed.
1	Read/write isolation has been tested.
2	Secure deletion has been tested, in addition to read/write isolation.
3	Absence of known side channel attacks has been tested, in addition to read/write and secure deletion.

Table 8: Data isolation testing level

It is important to note that in order to use such a metric, the resources in the scope of the measurement need to be well defined (storage, CPU, network, memory, database, etc.). Additionally, a standard set of tools or procedures need to be defined to establish the tests that should be conducted to assess each level.

7.2.9.3 Location

In a cloud environment, providing the exact location of the data center that holds the client's data is neither strictly useful nor necessarily desirable (for physical security reasons). On the other hand, there is a strong case for providing a country or regional indicator of the location of the data, since it has strong regulatory implications (in combination with other location information, such as the location of the company's headquarters or the location of the equipment being used if it other than the location of the compnany's headquarters). In some circumstances, the data may however reside in two or more datacenters during its life cycle.

In practice, it is usually impossible to strictly "prove" that data is only in a particular location (and not elsewhere) and we must rely on the trustworthiness of the cloud providers to provide that information.

We propose to define a location indicator as follows:

- Location.

A list of pairs (location, certainty) as an indicator for location in the cloud, where:

  - **Location** refers to the ISO 3166-1 alpha-2 country or region code where the data resides.
  - **Certainty** refers to the probability that the data will be located in this location at least once during its lifecycle (according to the CSP).

Note that identical copies of data may be simultaneously in two different locations. Additionally, the proposed "**certainty parameter**" could also be expressed as the average percentage of time that the data spends in a location during its lifecycle, but this appears to be more difficult to measure.

7.2.10 Trust and Reputation

The term trust has been used with a variety of meanings [60]. The Oxford Reference Dictionary defines trust as "the firm belief in the reliability or truth or strength of an entity." Reputation is a close concept usually defined as the opinion that people in general have about about a natural or legal person (ie. company). In the electronic world reputation is considered as a more objective measure than trust. Most trust systems are based on reputation, being entities more trusted when the reputation is higher. Disparate kinds of trust and reputation metrics had been proposed for specific contexts, such as ad-hoc networks, wireless sensor networks or P2P communities.

Trust can be measured in many different ways but apart from the measure, the semantic of each particular trust metric is also important. In any case the measure is related to some expectance on the future behavior of entities but each metric gives a specific meaning.

7.2.11 Risk Metrics

We here provide some generic methods for developing a risk metric. A4Cloud risk metrics will be used to assess risk in performing according to the agreed accountability measures. Therefore, we need the accountability metrics first to clearly define the way to assess risk in achieving them. A way to assess the risk can be based on one or all of the following statistical measures for dispersion and tendency of the historic data, which is updated continuously. Measurements for selected accountability metrics are continuously made, and the average of the values that are below the level of agreed value (the value agreed with customer and in SLA) is considered as the expected risk. Semi-variance is computed as standard variance but the observations above the agreed value are not taken into account because they are positive and cannot be perceived as risk.

If the observations on the selected accountability metric are distributed according to the normal distribution, the expected value with a lower bound according to a α value (e.g., 2.5%) can be agreed as the promised service level, and the lower tail of the distribution according to the α value can be accepted as the risk.

Another way in developing the risk metrics can be based on modelling and historic tendency and dispersion data together. For this, queuing theory can be used with the parameters derived from historic data for each of the service providers. By using this approach, a grade of experience (GoE) metric can be developed for every accountability metrics. This can be more useful especially for nested accountability concept because it allows modelling the risks associated with various combinations of service providers.

8 List of Figures

9 List of Tables