CLOUD ACCOUNTABILITY PROJECT

D:D-2.3: Initial reference architecture

Deliverable Number	D42.3
Work Package	WP 42
Version	Final
Deliverable Lead Organisation	HP
Dissemination Level	PU
Contractual Date of Delivery (release)	31/03/2015
Date of Delivery	01/04/2015

Editors

Frederic Gittler (HP), Theo Koulouris (HP), Siani Pearson (HP), Vasilis Tountopoulos (ATC) , Mehdi Haddad (EMN), Melek Önen (EURECOM)

Contributors

Richard Mark Brown (ATC), Jesus Luna (CSA), Alain Pannetrat (CSA), Jean-Claude Royer (EMN), Mohamed Sellami (EMN), Monir Azraoui (EURECOM), Kaoutar Elkhiyaoui (EURECOM), Niamh Gleeson (QMUL), Asma Vranaki (QMUL), Anderson Santana De Oliveira (SAP), Karin Bernsmed (SINTEF), Carmen Gago (UMA), David Núñez (UMA)

Reviewers

Rehab Alnemr (HP), Christoph Reich (HFU)

Executive Summary

The goal of the A4Cloud Reference Architecture is to provide an abstract but powerful model for designing accountability in modern cloud and future Internet ecosystems. It is as an essential step towards addressing the requirements of the target stakeholders by defining the architecture vision and capabilities and delivering a roadmap to implement such requirements in specific cases, aligned with selected business goals.

This document is a time-driven release, structured to provide a high-level view of the architecture for accountability the A4Cloud project is developing. As such it comprises an intermediate step towards the delivery of the full A4Cloud Reference Architecture, which will be released in its final form by March 2016.

The high-level architecture described in this document captures in a succinct manner the main technical areas to be developed in relation to one-another and creates a base for assessing the coherence of the tools. The document presents:

- A methodology for applying, from the organization's perspective, an accountability-driven governance approach, including the Accountability Maturity Model AMM), to quantitatively assess their accountability practices;
- An in-depth discussion of the account, which provides one of the principal means of demonstrating accountability;
- A set of accountability-support services proposed to tackle the challenges of extending accountability across complex cloud service provisioning chains;
- A description of the tools which are developed by the project to address specific accountability functions and an early view on their integration.

Table of Contents

1 Introduction

In A4Cloud, we adopt the following definition for the Reference Architecture: "*A Reference Architecture is, in essence, a predefined architectural pattern, or set of patterns, possibly partially or completely instantiated, designed and proven for use in particular business and technical contexts, together with supporting artifacts to enable their use*" [1]. As such, the goal of the A4Cloud Reference Architecture is to provide an abstract but actionable model for designing accountability in modern cloud and future Internet ecosystems. It is as an essential step towards addressing the requirements of the target stakeholders by defining the architecture vision and capabilities and delivering a roadmap to implement such requirements in specific cases, aligned with selected business goals.

The ultimate goal of the A4Cloud Reference Architecture is to offer all possible stakeholders in the cloud service provisioning chain, a guidance on how accountability can be implemented across different contexts in the cloud and Future Internet applications and, share technology-driven best practices and tools for delivering accountability-based solutions to end users.

The high-level architecture described in this document captures in a succinct manner the main technical areas to be developed in relation to one-another and creates a base for assessing the coherence of the tools developed by the project. The document presents:

- A methodology for applying, from the organization's perspective, an accountability-driven governance approach, including the Accountability Maturity Model (AMM), to quantitatively assess their accountability practices;
- An in-depth discussion of the account, which provides one of the principal means of demonstrating accountability;
- A set of accountability-support services proposed to tackle the challenges of extending accountability across complex cloud service provisioning chains;
- A description of the tools which are developed by the project to address specific accountability functions and an early view on their integration.

The project is providing further guidance to ease the technical integration of the various components in the form of a "guidelines and principles" document available separately [2].

This document is a time-driven release, structured to provide a high-level view of the architecture for accountability the A4Cloud project is developing. As such it comprises an intermediate step towards the delivery of the full A4Cloud Reference Architecture, which will be released in its final form by March 2016. The designs reported in this document are preliminary in nature and are subject to evolve.

2 Fundamental Concepts

Accountability in the context of handling personal and business confidential information is an important but complex notion that encompasses the obligation to act as a responsible steward of the personal information of others, to take responsibility for the protection and appropriate use of that information beyond mere legal requirements, to be transparent (give account) about how this has been done and to provide remediation and redress. The perceived lack of transparency and control over data governance, inherent in complex cloud service provision chains makes accountability a key market enabler which can help overcome barriers to cloud service adoption. Still, providing accountability both legally and technically in the cloud has proved to be very challenging.

In the A4Cloud project we propose a co-designed approach for accountability that combines a range of technological enhancements with legal, regulatory and governance mechanisms to provide the necessary basis for initiating and sustaining trustworthy data processing and a trusted relationship between data subjects, regulators and information and communications technology (ICT) providers.

Although the goal behind the A4Cloud Reference Architecture (the "RA" hereafter) is to propose a blueprint for end-to-end accountability across the entire cloud service provisioning chain, the starting point for a lot of the concepts and mechanisms discussed focus on making a single organization accountable. The rationale is that the problem of creating accountable cloud supply chains becomes much more tractable if the actors chained together are accountable. Therefore, we begin by defining what it means for an organization to be accountable. In this context, the A4Cloud project has developed a definition of "Accountability for Data Stewardship in the Cloud" and a corresponding model of how various elements of accountability, can be combined to create a roadmap for accountability [3].

As illustrated in Figure 1, the A4Cloud accountability model consists of:

- **Accountability attributes:** central elements of accountability (i.e. the conceptual basis, and related taxonomic analysis of accountability for data stewardship in the cloud). Namely, these are transparency, responsiveness, remediability, responsibility, verifiability, effectiveness and appropriateness.
- **Accountability practices:** emergent behaviour characterizing accountable organizations (that is, how organizations can incorporate accountability into their business practices). Specifically, an accountable organization:
    - Defines governance to responsibly comply with internal and external criteria, particularly relating to treatment of personal data and/or confidential data.
    - Ensures implementation of appropriate actions.
    - Explains and justifies those actions, namely, demonstrates regulatory compliance that stakeholders' expectations have been met and that organizational policies have been followed.
    - Remedies any failure to act properly, for example, notifies the affected data subjects or organizations, and/or provides redress to affected data subjects or organizations, even in global situations where multiple cloud service providers are involved.
- **Accountability mechanisms:** operational processes, non-technical mechanisms and technical tools that support accountability practices. Operational processes operate at the organizational business process level, by extending existing processes like auditing and risk assessment to support accountability practices. Non-technical mechanisms consist of accountability-reinforcing mechanisms that are predominantly non-technical, such as contracts, policies, codes of conduct, and various legal safeguards and deterrents. Finally, technical tools comprise the various software systems and components an organization may use to carry our various accountability-related operations.

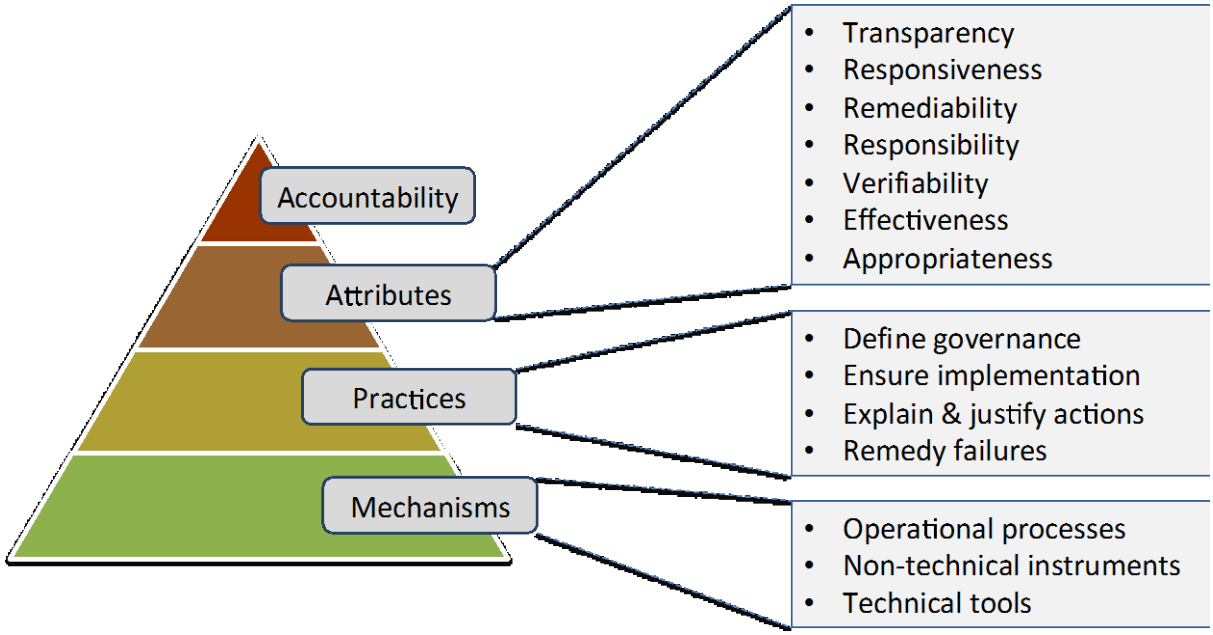

Figure 1: The A4Cloud accountability model.

We may further classify the accountability mechanisms into three categories:

1. Innovative mechanisms designed and built for purpose by A4Cloud (i.e. *"things we build"*).
2. External mechanisms which are imported/utilized by A4Cloud mechanisms (i.e. *"things we import"*).
3. External mechanisms with which A4Cloud mechanisms will co-exist (i.e. *"things we interface with"*).

According to [3], *accountability reflects an institutional relation arrangement in which an actor can be held to account by a forum (for example, a consumer organisation, business association or even the public at large). Accountability then focuses on the specific social relation or the mechanism that involves an obligation to explain and justify conduct.*

Accountability describes a relationship between two agents, namely, the accountor and the accountee [3]. The accountor is the agent that has the obligation to answer, explain and justify his or her conduct. The accountee is the agent that questions, assesses and criticizes the conduct of the accountor.

A core element of the concept of accountability is the "account". Within an accountable system, the "account" can be seen as an explanation or demonstration of the system's behaviour, norms or compliance. We identify three types of "account": proactive account, account of legitimate event(s) and account of incident(s) (see Section 3.5 for details). The description of an "account"-related event provides answers to the six "reporters' questions":

- Who: identifies actors involved in the described event.
- What: describes what the account is about.
- Where: describes where the event related to the account occurs (not only physical location).
- When: depicts when the described event occurs.
- Why: presents why the event happened (to respect policies/obligations for instance).
- How: illustrates the used means (logs, encryption, etc.) for the described event.

An "account" also comes with evidence, when possible, associated to these different answers and means for remediation if adequate (the case of an account on an incident for instance).

In the reminder of this document the core concepts of the accountability model (Figure 1) will be further depicted. In particular, the accountability attributes (transparency, responsiveness, remediability, responsibility, verifiability, effectiveness and appropriateness) will be related to the accountability metrics in section 3.4.1. Indeed, metrics constitute an instrument to verify the compliance of non-functional requirements. Therefore, metrics offer means to illustrate the support of accountability by privacy and security governance that are in use.

Accountability practices are discussed in Section 3.5 where different notions of account and their properties are presented. Two main types of account are discussed: evidence on compliance and data breach.

Accountability mechanisms are detailed in Section 4. In this section, it is explained how high-level goals need to be first decomposed into accountability artifacts and then recomposed to provide the assurance and the account. High-level goals express the privacy and security requirements as well as the laws and regulations. Accountability artifacts represent various accountability related information (e.g., obligations, evidence records, notification reports).

2.1 Actors and Roles

A key challenge when reasoning about accountability in a cloud context is the adoption of a common vocabulary for expressing fully and consistently elements coming from both the world of technology and from the domain of data protection. The need for a common vocabulary is particularly relevant for the definition of actors and roles in the A4Cloud Reference Architecture (RA).

The NIST Cloud Computing Reference Architecture [4] defines five major actors:

- Cloud Consumer: “A person or organization that maintains a business relationship with, and uses service from, Cloud Providers.”
- Cloud Provider: “A person, organization, or entity responsible for making a service available to interested parties.”
- Cloud Auditor: “A party that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation.”
- Cloud Carrier: “An intermediary that provides connectivity and transport of cloud services from Cloud Providers to Cloud Consumers.”
- Cloud Broker: “An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Consumers.”

Although these five roles are sufficient for representing the vast majority of interactions involved in a cloud service provision and procurement context, they do not effectively capture all the elements necessary to reason about accountability. Specifically, as it is, the NIST model cannot capture the following two roles:

- Data “owners”: Individuals, in particular data subjects or organizations who have some personal or confidential data processed in the cloud, and who may not necessarily be qualified as ‘cloud customers’ (or consumers) in the NIST taxonomy. Though more rarely, this also applies to businesses, which may have business confidential data processed by the cloud despite not being a cloud customer (rather customers of a cloud customer). They are essentially “invisible” in the NIST model, but represent the ultimate role in an accountability chain.
- Supervisory authorities: Data protection authorities or telecom regulators may be seen as auditors, but they also have the distinct characteristic of holding enforcement powers, which auditors lack.

In the interest of maintaining maximum compatibility and alignment with the NIST model which appears to be well understood amongst cloud stakeholders, we chose to extend it to cover these roles and support accountability, as follows[1]:

1) Cloud Subject: An entity whose data is processed by a cloud provider, either directly or indirectly. When necessary, we may further distinguish between:
   a) Individual Cloud Subject, when the entity refers to a person.
   b) Organization Cloud Subject, when the entity refers to an organization.
2) Cloud Customer: An entity that (1) maintains a business relationship with, and (2) uses services from a Cloud Provider. When necessary we may further distinguish between:
   a) Individual Cloud Customer, when the entity refers to a person.
   b) Organization Cloud Customer, when the entity refers to an organization.
3) Cloud Provider: An entity responsible for making a (cloud) service available to Cloud Customers

[1] For an extended discussion please refer to the A4Cloud Conceptual Framework document [3].

4) Cloud Carrier: The intermediary entity that provides connectivity and transport of cloud services between Cloud Providers and Cloud Customers
5) Cloud Broker: An entity that manages the use, performance and delivery of cloud services, and negotiates relationships between Cloud Providers and Cloud Customers
6) Cloud Auditor: "An entity that can conduct independent assessment of cloud services, information system operations, performance and security of the cloud implementation, with regards to a set of requirements, which may include security, data protection, information system management, regulations and ethics.
7) Cloud Supervisory Authority: An entity that oversees and enforces the application of a set of rules.

These roles both share similarities and articulate differences with the cloud computing roles defined in the NIST model in the following way:

- We introduce a new role (Cloud Subject) to designate an entity that owns data, which is either directly transferred to a cloud provider for processing, or indirectly through a cloud customer. We further distinguish Cloud Subjects as individuals or organizations.
- The role of Cloud Customer is aligned with the NIST definition (as a synonym of Cloud Consumer) but we further introduce a distinction between individual Cloud Customers and organizational Cloud Customers.
- The roles of Cloud Provider and Cloud Broker are adopted without modification from the definition provided by NIST.
- The role of Cloud Auditor is based on the definition provided by NIST but was altered to better reflect the goals of accountability, by additionally referencing data protection as well as regulatory and ethical requirements.

We note that the role of Cloud Carrier defined by NIST is unlikely to be considered in the context of accountability, since a Cloud Carrier does not normally take any responsibility for data stewardship but merely acts as a neutral transporter (much like an Internet Service Provider). In the case where a Cloud Carrier takes a stronger role in terms of data stewardship, or if the routing of data traffic matters, we may consider it as a Cloud Provider instead without loss of generality.

Even in its extended form, however, the cloud role classification alone cannot provide all the information necessary to fully characterize an actor. For example, a cloud provider can be either a data controller or a data processor, with fundamentally different responsibilities in each case. For that reason, the proposed model for fully specifying an actor's role in the A4Cloud Reference Architecture is to provide both the (extended) cloud and the data protection (95/46/EC and 2002/58/EC) role classifications. Table 1 below presents all the possible combinations of cloud computing and data protection role classifications identified in RA. In conclusion, to fully characterize an actor in the RA and documents produced by the A4Cloud project in general, the proposed nomenclature combining cloud and data protection roles, as presented in Table 1, should be used.

Extended NIST cloud roles	Data protection roles
Cloud subject	Data subject
Cloud customer	Data controller or Data processor
Cloud provider	Data processor or Data controller
Cloud carrier	Data processor or Data controller (unlikely) or Not applicable.
Cloud broker	Data processor or Data controller
Cloud auditor	(Not Applicable)

Cloud supervisory authority	Supervisory authority (DPA or NRA)
(Not Applicable)	Third party
(Not Applicable)	Recipient

Table 1: A4Cloud Reference Architecture roles.

2.2 Conceptual Model of the Reference Architecture (RA)

With the essential actors and roles necessary to describe accountability relationships identified, the building blocks of the RA model can now be examined. Again, emphasizing the importance of building upon established and standardized concepts instead of "re-inventing the wheel" we base the RA conceptual model on the corresponding NIST cloud reference architecture conceptual model [4]. Figure 2 below illustrates how the RA adapts and extends the NIST cloud reference architecture to support accountability, by adding new actors as well as "accountability" (yellow bars) as a cross-cutting concern alongside security and privacy, since in order to be effective, all three need to be implemented across all layers and functions.

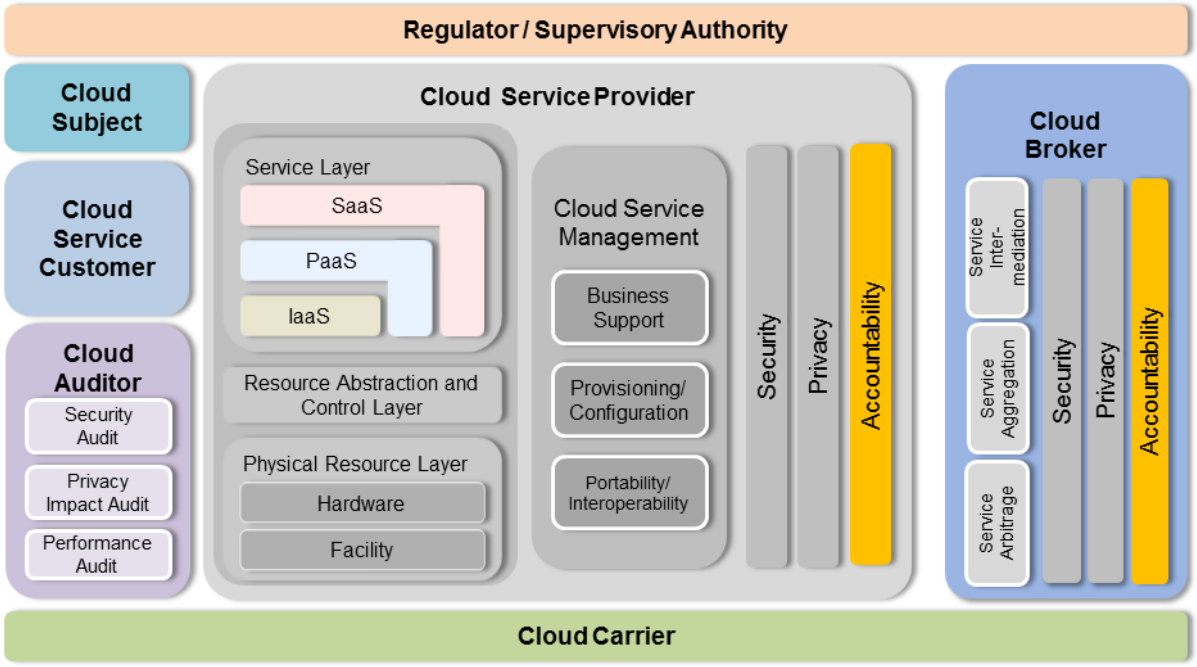

Figure 2: A4Cloud Reference Architecture conceptual model.

3 Accountability Process

Operating in an accountable manner is not simply a matter of deploying the tools to implement technical controls and to report on their behaviour. It actually starts at the very top of the organization, with the Board of Directors, is embedded in the foundation values of the organization ("organization DNA"), and is transmitted through the whole organization through governance. In the following sections, we will explore the practices required to operate in an accountable manner.

3.1 Accountable Organisations

3.1.1 Introduction

The Accountability for Cloud Conceptual Framework [3] defines accountable organisations as being *one that takes an accountability-based approach, implying the adoption of the entire set of the accountability practices*. The Conceptual Framework then expands on accountability at an organisational level, focusing on the ways they could be implemented in practice. For the benefit of the reader, these conclusions are listed below:

The Galway project [5] *has defined the central elements that an accountable organisation needs to address as being:*

1. *Organisation commitment to accountability and adoption of internal policies consistent with external criteria.*
2. *Mechanisms to put privacy policies into effect, including tools, training and education.*
3. *Systems for internal, ongoing oversight and assurance reviews and external verification.*
4. *Transparency and mechanisms for individual participation.*
5. *Means for remediation and external enforcement.*

Influenced by this approach, the Canadian privacy commissioners have specified the measures that an accountability management program (for the data protection domain) would ideally include [6]*:*

1. *establishing reporting mechanisms and reflecting these within the organisation's privacy management program controls*
2. *putting in place privacy management program controls, namely:*
   - *a Personal Information Inventory to allow the organisation to identify the personal information in its custody, its sensitivity and the organisation's authority for its collection, usage and disclosure*
   - *policies relating to: collection, use and disclosure of personal information (including requirements for consent and notification); access to and correction of personal information; retention and disposal of personal information; privacy requirements for third parties that handle personal information; security controls and role-based access; handling complaints by individuals about the organisation's personal information handling practices*
   - *risk assessment mechanisms*
   - *training and education*
   - *breach and incident management*
   - *procedures for informing individuals about their privacy rights and the organisation's program controls*
3. *developing an oversight and review plan that describes how the organisation's program controls will be monitored and assessed*
4. *carrying out ongoing assessment and revision of the program controls above*

Furthermore, the proposed EU General Data Protection Regulation (GDPR) [7] *includes many accountability elements including, in Article 22, a list of a Data Controller's accountability instruments:*

- *Policies*
- *Documenting processing operations*
- *Implementing security requirements*
- *Data Protection Impact Assessment*
- *Prior authorisation/consultation by Data Protection Authorities (DPAs)*
- *Data Protection Officer*

- *If proportional, independent internal or external audits*

While we have adopted a "*focus on the data protection domain and on accountability of organizations rather than individuals*" [3], one of our main concern is accountability in the context of IT supply chains based on the use of Cloud Services. End-to-end accountability, which is further analysed in section 4, requires all actors of the supply chain to be accountable organisations to a certain degree. However, the domain for which these organisations need to be accountable is not necessarily the data protection domain. For example, when an organization implements a service which is handling sensitive data (in regards to the data protection regulations) through the use of an IaaS cloud service provider, the latter is typically accountable for providing adequate security, and not for implementing an accountability-based data protection program.

It should not be noticed that, even if the focus as stated above is on organizations, the role of individuals involved is also essential and that accountability must be ensured down to the employee level. Accountable organizations must provide individuals with the necessary tools and procedures to be individually accountable.

In the remainder of this section, we have attempted to define the measures that should be implemented by an accountable organisation in a manner which remains agnostic to the domain. The recommendations have been identified based on work done for both the data protection domain, such as CNIL [8], ICO [9], and Nymity [10]. These have been augmented by more general organizational standards, such as COBIT [11] and ISO 27001 [12]. The author has also leveraged the HP Security Handbook [13] as well as his own professional experience.

3.1.2 Introduction to the Organisational Lifecycle for Accountability

The Conceptual Framework introduces the Organisational Lifecycle and introduces the Functional Elements of Accountability, which provides the reference model for this discussion.

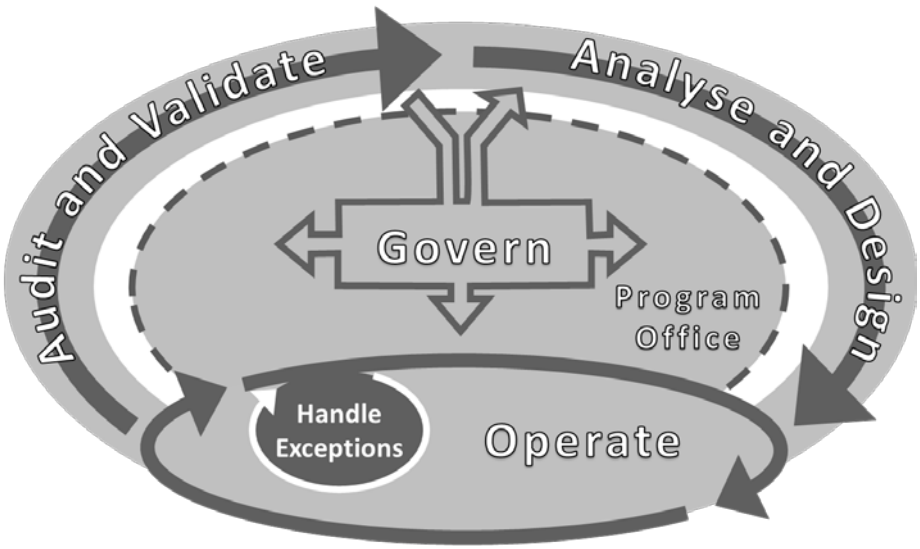

Figure 3: Organisational Lifecycle for Accountability

This lifecycle is organized around five phases which provide a structure to the solution development, operation, and maintenance. Specific to the Corporate Accountability scope, we introduce a sixth element, the Program Office, which provides the operational support to the governance body. Note that the first two elements (Govern and Program Office) are strongly associated with organisational accountability, while the three first lifecycle elements (Analyse and Design, Operate, Handle Exceptions) describe the lifecycle for building and operating an "accountable solution". The last element (Audit and Validate) is applicable to both domains, as the assessments can be either focused on the organisation as a whole or on a particular solution or business service.

1. Govern – This corresponds to the executive roles in the organisation establishing and maintaining a framework and supporting management structure and processes, as well as accepting and providing assignment of responsibility, to meet the obligations of the organisation in an accountable manner

2. Program Office – This is the operational body which supports the governance body in meeting its responsibilities in e.g. drafting guidelines, policies and procedures, defining the operational programs and infrastructure, and providing oversight and support for the implementation of the decisions of the governance body. This program office is typically in charge of both organisational accountability as well of the domain for which the organisation is accountable (e.g. the privacy program office or the security program office); it can either be an organisational or a logical structure.

3. Analyse and Design – This corresponds to the analysis and design phases related to the engineering of a solution. The work performed in this phase clearly separates identification of risks (based on business impact, not just technology), identification of controls, design of control implementation, and implementation of controls through technology and processes.

4. Operate – This corresponds to the operational (production) phase of the solution, and includes all the associated management processes

5. Handle Exceptions – This set of activities, which could be considered as an integral part of operations, has been singled-out due to its specific nature and high relevance to accountability. It includes all processes for the handling of complaints and breaches related to accountability obligations

6. Audit and Validate – This corresponds to the assessment of the effectiveness of the controls which have been deployed, the necessary reporting, and paves the way to the tuning (adaptation) of the measures deployed to ensure the obligations are being met.

The following sections describe in more detail the content of each of these phases. More than a general discussion and framing of the scope, we want to provide practical guidelines for implementing accountability. We are providing a series of recommendations which can be used as a checklist. We do not claim this describes a specific methodology but provides a general guideline on integrating accountability within an organization.

These lists are not comprehensive, and each of the points must be evaluated in regards to the size and structure of the organisation. The full list of recommendations may, in general, not be applicable to smaller groups such as SMEs. We acknowledge there are many common points between the recommendations identified below and the actions identified as required for topics like data protection, business continuity, disaster recovery, information security management, and trustworthy accounting. However our recommendations below are not intended to be a substitute for those lists of actions – an organisation must address all of them in order to have a comprehensive coverage and meet its obligations. The analysis focuses on the processes to be deployed by the accountor rather than those of the accountee.

3.1.3 Roles and Responsibility of Governance Bodies

In the scope of our analysis, we consider obligations which have to be met by the organisation as a whole and where the responsibility for fulfilling these obligations rests with the board members and executive managers, with some part of it going down to the employees. In this context, being responsible often goes beyond the civil responsibility as laws often assign penal sanctions for not meeting obligations and not performing due diligence.
The governance bodies are the owners of the strategic dimension of accountability. In order to fulfil this mission, the board and executive management must:

- Understand relevant obligations in breadth and in depth
- Understand the consequences of not fulfilling the obligations

- Accept responsibility for fulfilling these obligations in an accountable and responsible manner – this is applicable not only to the governance body as a whole, but must be an integral part of the mission of each member of the governance body, which must embrace the obligations, ensure support from each (relevant) functional area, and act as champions in the organisation
- Define the "internal criteria" for the organisation.
- Understand the risks associated with the operation of the business in regards to the obligations. Define a "risk appetite" used as guidance for operational decisions, taking into account the nature of the obligations (ethical, social, or industry norm, contractual, regulatory, legal, etc…). The acceptable level of risk acceptance delegated to the various levels of the organization will typically increase with the management levels in the organization.
- Appoint an executive-level owner who will oversee and be accountable for the fulfilment of the obligations. For example, this is typically the Chief Privacy Officer or the Chief Security Officer for (respectively) the data protection domain or the security domain.
- Ensure the proper integration of all responsibilities and actions across the whole organisation. Avoid operating the program as a "silo" or an afterthought.
- Ensure a proactive attitude towards the accountability domain (e.g. data protection) across the organization.
- Drive the adoption of an accountability-driven mindset. Ensure that this becomes part of the culture, and is integrated with the core values of the organisation (e.g. code of conduct, ethical guidelines, list of values).
- Ensure accountability is properly integrated in all relevant processes (e.g. business management, risk management, compliance management, reporting)
- Ensure that employees are properly trained to understand the concept of Accountability and their own obligations.
- Ensure that Employees are provided with appropriate tools and processes to fulfil their own part of the Accountability obligations.
- Ensure the organisation is ready to respond to discontinuities in compliance to obligations (incident response)
- Regularly review the status of the organization in regards to the compliance to the obligations

In order to fulfil this mission, depending on the scope, the organisation's high level management will typically create a Program Office to provide the operational support for the enactment of the governance decisions and more generally support the governance body in meeting its responsibilities. The Program Office will typically report to the executive-level owner. In relationship with this Program Office, the governance body will:

- Define the mission and charter of the Program Office
- Define the level of authority of the Program Office
- Ensure the Program Office is provided with the necessary means, in terms of resources, personnel, funding and authority so it can fulfil its mission
- Support and champion the various policies, programs, processes and other actions identified by the Program Office as necessary to meet the obligations
- Regularly review the work performed by the Program Office. Use external audits to get an external view on the performance of the Program Office

3.1.4 The Program Office

The program office may be focused on accountability across the organisation, but will more typically be in charge of both the domain (or one of the principal domains) for which the organisation is accountable (e.g. the privacy program office or the security program office) and of accountability. The program office can either be an organisational or a logical structure.

In its role to support the governance bodies in fulfilling their responsibilities, the mission of the Program Office can be divided in eight main areas:

- Inventory Obligations, Risk Assessment, and Risk Treatment
    - Maintain a registry of all obligations and associated operational standards

- Perform risk assessments and identify risks and exposure. This is focusing on the business and related (accounting, …) practices
- Identify how to treat the risk. The analysis must include cost, timing, alternatives, and comply with the "risk appetite" identified by the Board.
- Create a rollout plan
- Create a set of metrics to report on the state of the accountability program
- Investigate best practices and compliance frameworks, consider adoption, attestation or certification based on benefits, costs, and risk.

This sets the stage to perform due-diligence, which often defines the boundary where the responsibility of the officers of the organisation is engaged. Performing due-diligence is however not enough – it can only be used as defence if it can be demonstrated. The Program Office must be sure that this can be done.

- Company Culture, Practices and Standards
  - Review relevant company codes, operating guidelines, and standards in regards to obligations. This must take a holistic view and deal with all appropriate business functions: sales, marketing, business operations, IT, facility management, workplace solutions, finances, accounting…
  - Draft appropriate changes to these codes
  - Rollout these changes and ensure effective change of the documentation throughout the organization
  - Notify staff of changes through adequate communication programs (awareness)
  - When none exist, foster organisation-level codes and standards creation by or in collaboration with businesses & functions when required for alignment of the internal business processes. Ensure compliance through checklists and metrics used for reviews at different levels
  - Build templates for analysis (e.g. impact assessment, risk assessment) in regards to accountability and obligations, for use in the Analyse and Design phase of the lifecycle.
  - Define a sign-off process with responsible parties to validate the major milestones in the Organisational Lifecycle for Accountability

  It is important to adopt the principle that all actions performed be traceable to the persons performing and authorizing it (attributability). This is to be used primarily for root-cause analysis, continuous improvement and individual accountability.

- Incident Recovery and Response

  This is a critical success factor. It is the responsibility of the Program Office to ensure that an adequate structure and set of processes is effectively implemented in the organisation. Considering the scope, this is often best implemented as a pan-organisation structure, rather than smaller teams dedicated to individual product offerings or, at a minimum, that such a structure exists to support dedicated product teams in case of an exceptional event. Details of this program are provided in section 3.1.7.

- 3rd Party Engagement
  - Ensure, with active involvement of procurement and contract negotiator, that all service and other provisioning contracts are compliant with relevant obligations
  - If appropriate, define appropriate standards and practices in regards to engagement with third-parties
  - Ensure that procurement or other relevant organizations maintain and update a registry of third-party engagements and their relationship with obligations
  - Enforce strict compliance with the standards and practices, as well as reporting
  - Monitor that procurement or other relevant organizations ensure that contract renewal and changes in terms are properly tracked
  - Ensure 3rd party providers are regularly reviewed by procurement or other relevant organizations
  - Ensure that processes are in place by procurement or other relevant organizations to deal with non-compliance of 3rd parties

  The collection and archiving of contracts is required but not sufficient in most instances. It must be possible to get a quick understanding of the relationship of external engagements

with obligations across all engagements (hence the need to maintain a registry). Also refer to the discussion on 3rd parties in section 3.1.5.

- Employee Skills and Awareness
  - Ensure the organisation maintains a registry of job (function) profiles in relationship with the obligation and identifies "sensitive positions"
  - Ensure the organisation defines recruiting criteria for sensitive positions re. obligations
  - Ensure the organisation has specific training programs for sensitive positions re. obligations
  - Ensure compliance with legally-required training and certification
  - Inject topics in the on-boarding and recurrent employee code of ethics and business training programs
  - Ensure the organisation includes questions measuring accountability awareness and attitude in employee surveys
  - Ensure the organisation organizes and rolls-out specific accountability awareness campaigns, such as posters displayed on billboards and internal bulletins
  - Ensure that management "state of business" (coffee talks) presentations regularly address accountability
  - In more general terms, ensure the organisation addresses accountability in appropriate employee information vehicles with adequate messages for rollout through the organization
  - Ensure the organisation keeps skills of the specialized staff current – support staff to join industry associations, professional networks, specialized conferences, and get training as required

  Accountability for one's own actions, regardless of the domain to which it is applied, must become part of the culture of the organisation, must be embraced by all employees and contractors. Circumventing this must be treated as a serious performance issue. It must be noted that human interaction is one of the weakest points in the security of a system.

- Compliance Readiness
  - Appoint liaison to external domestic and regional compliance and regulatory agencies (as appropriate)
  - Appoint liaison to relevant foreign compliance and regulatory (as appropriate)
  - Ensure the organisation tracks "external criteria" – use a mix of specialized information services, industry associations, professional networks, specialized conferences, and consultants.
  - Ensure the organisation understands reporting requirements and ensures compliance
  - Maintain the legally-required documentation (or ensure it is maintained by the relevant departments)

- Deploy Individual Accountability Tools
  - Investigate and (if adequate) ensure deployment of techniques and tools supporting authorization based on duty segregation
  - Investigate and ensure deployment of tools guaranteeing that all actions are logged and allow the identification of the agent and of the authorizer (as appropriate). This must be done in compliance with legal constraints on the handling of individually identifiable information. Ensure that these tools bear a proper timestamp and are secured against tampering or destruction.

  There are some commercially-available tools which act as portals and allow the deployment of these types of controls even if the native applications do not support the functionality. In addition, using a uniform mechanism across the organisation allows for a streamlined management of the authorisation structure and of the audit logs. One must note that provisioning the trusted log with attribution is more important, less expensive, and less problematic than deploying the authorization framework due to the complexity of modelling and allocating the correct structure for the roles.

- Ensure continuity of the accountability program
  - Ensure that all the above documentation stays current.
  - Periodically review the various analyses performed

- Define and maintain a dashboard providing a synthetic view of the accountability program
- Define and track a set of metrics to measure effectiveness and progress. A significant part of these metrics must correspond to objective (vs. subjective) criteria.
- Have the accountability program and the Program Office audited regularly (one the order of once a year) by external auditors.

3.1.5 Provisioning for Accountability – Analyse and Design

By its very nature, accountability identifies and addresses potential risks, harms and expectations – there would be no need for accountability if one could provide a total and absolute guarantee that all these obligations are met. Designing for accountability is therefore naturally associated with the lifecycle dealing with security, data protection, and risk. There are many variants for this lifecycle – we will use the pragmatic model described in [13].

- Understand the product or service and related assets
  - Have a description of the functionality and associated non-functional requirements
  - Inventory the (related) obligations for which the organisation will be accountable
  - Inventory the assets related to the functionality and obligations
  - Perform an impact assessment for these assets in order to qualify the risk
  - Keep and update a record of assets and impacts

- Perform a risk analysis
  - Identify the position of the board and executive management
  - Perform a risk analysis – in regards to accountability, this should focus on obligations and the contributing factors, as well as on the accountability support system. The risk analysis must take in consideration all aspects impacting the organisation, including secondary impacts such as a loss of reputation, product or service implementation, slow down, regulators push back,...
  - Keep and update the record of threats and vulnerabilities, qualified with probability and impact. Associate this record with the record of assets and impacts.

  Risk analysis is a complex process for which many methodologies and software support exist for traditional security related domains. Risk analysis related to accountability is only a part of the overall risk analysis process with a lot less existing tools and practices. At this stage, we make no recommendation on a methodology or set of tools to use but more about this will be discussed in future stages of the project.

- Define the risk treatment
  - Define how risk will be handled as part of the investigation, design and engineering phases for the product or service
  - For each risk, identify if it will be reduced, mitigated, assigned, transferred, or accepted. The residual risk must be understood and treated in a recursive manner until the constraints associated with the obligations are met (typically, the residual risk in the last iteration will be transferred or accepted).
  - Define the controls which will be deployed to reduce or mitigate each risk, and define how the risk will be assigned or transferred, as appropriate. These controls can be based on a mix of technology and processes. In this latter case, a link must be established with the inventory and operational programs defined at the global level for the organisation (see in particular sections 3.1.4 and 3.1.7).
  - Define the metrics and dashboard for continuous monitoring of the state and effectiveness of the risk treatment plan.
  - Augment the above record of threats and vulnerabilities with the record of risk treatment and associated measures. Include all steps in the recursive treatment of residual risk.
  - Validate that the risk analysis takes in consideration the selected implementation decisions. Update it as necessary.
  - Ensure that the accountability mechanisms are tested as part of the solution testing (in both regression and system tests). Validate the effectiveness of the measures.

This step is tightly integrated with the solution design and engineering phase, and is recursive by nature as each implementation alternative has an impact on risk. The information obtained in this process provides the foundation for signoff and the creation of the first account.

The result of this phase will be the definition of a technical solution to implement the solution or offering. The technical or procedural controls implemented correspond to a due-diligence and best effort coverage of the requirements. It is in general impossible to guarantee that the mechanisms and procedures effectively deployed guarantee that the obligations will be met. The accountability system must be able to handle the unexpected.

- Selection of 3rd party providers
  Special attention must be given the selection of 3rd party providers. We will focus on the selection of cloud services. This section leverages the recommendations made in [14] and [8], but have been considerable modified to address accountability in general as opposed to DPR.
  When selecting a cloud provider, the accountable organization must:
  - Identify assets which will be processed or stored in the cloud environment.
  - Identify the related obligations and associated accountability requirements
  - Based on the initial risk analysis, identify the risk profile of the provider, the set of security (and other relevant attributes) that must be supported
  - Identify the links between the third-party accountability provisions and the accountability system of the organisation – list the associated requirements
  - Based on compliance obligations, identify the certifications and other levels of guarantees that must be offered by the provider (most often including constraints on the local for the data centre and the IT staff)
  - Check internal policies and procedures in terms of selection, registration, and tracking of third party service providers. Comply once the provider is selected.
  - Review and select provider based on a review of the offerings matching the above requirements. Validate the costs are in line with funding expectations. Review the risk appetite and risk treatment plans if there is a gap. The practice of increasing the levels of risk acceptance to meet cost expectation should be banned.
  - Ensure the proper contractual clauses are in place, especially as in regards to compliance to the requirements and the associated accountability measures. The contracts must be handled per organisational policies (see section 3.1.4)
  - Ensure the provider exploits the data only as intended and defined by the Data Controller
  - Exploitation for the benefit of the provider should be avoided, but if envisaged it must be carefully defined by contract and its consequences (liability and Data Controller role and relevant obligations) must be clearly understood by the organisation and be compliant with the allowable use of the data and legal requirements.
  - Identify the metrics that will be used during the lifetime of the relationship to continuously assess the compliance of the 3rd party.
  - Track changes in the service provider operations.
  - Ensure there is an adequate link between the provider exception handling processes and those of the organisation (see section 3.1.7). The associated procedures must be tested regularly and readiness assessments must be performed.

- Documentation and Signoff
  - The solution must be fully documented and placed under change management. Any evolution must be assessed against the requirement, obligations, and risks, and necessary adjustments must be performed.
  - The contracts associated with the solution must reflect what is actually implemented
  - An account reflecting the analysis linking obligations with actual controls must be produced and made available to stakeholders. This account is to demonstrate, through a static analysis, the effectiveness of the controls as due-diligence to meet the obligations.
  - The solution must go through a signoff process that will validate that all internal requirements and obligations have been met.

3.1.6 Operating in an Accountable Manner

This phase covers the support, management, and day-to-day operations. For the purposes of accountability, it focuses on two aspects:

- Operate the system as intended:
  - Gather and report on accountability and risk treatment metrics, keep the dashboards updated
  - Communicate with stakeholders as intended
  - Ensure that the collection of evidence is performed as intended
  - Ensure that all logs are effectively backed-up and are protected against tampering
  - More generally, ensure that all solution-specific processes and associated organisation-level processes are used and are operating with the intended effectiveness. This is also applicable to all processes related to 3rd parties.

- Look for signs of unexpected issues:
  - Continuously monitor the system, the operating environment, and the ecosystem for signs of incident, breach or significant change. Activate the exception handling processes as required (see section 3.1.7)

3.1.7 Handling Exceptions

It is critical to have a set of policies and processes to handle exceptions, along with the proper organisation to handle these exceptions quickly and completely. These exceptions can be significant, leading to a workload that cannot promptly be handled by an in-house structure of the organisation. Careful planning and coordination with external parties is therefore an important part of this process. In all cases, the organisation must ensure that the organisation and processes are defined with scalability and prompt reaction in mind as this is required for due-diligence The Program Office has a central role in organising this set of processes.

The core of the process is articulated on two series of coordinated suites of processes: the handling of complaints, which are externally generated, and the handling of incidents, which are detected by operational monitoring or are the result of complaints after investigation and qualification.

- Handling complaints: the organisation must be ready to:
  - Receive, handle, and respond to complaints and information requests in a timely manner as required by internal policies or legal requirements
  - Use a case management system to support the handling of requests, track progress, and provide global statistics for use by both the operational management, the Program Office, and the Board.
  - Have a FAQ to anticipate the most frequent information requests
  - Create a classification for requests and complaints. Define standard procedures for the handling of requests in each of the classes
  - Have defined escalation procedures

- Handling incidents and breaches: the organisation must be ready to (in sequential order):
  - Log and track the incidents in a secure, time stamped and reliable way
  - Provide a prompt operational response to the incident (“stop the bleeding”). This can mobilize staff from all business and technical departments of the organization as well as external experts
  - Notify the stakeholders (affected parties) of the incident, providing remediation actions, as soon as those have been identified
  - Report the incident to authorities when required by law or by the criticality of the incident – including both regulators and law enforcement
  - Perform a root cause analysis
  - Repair the affected applications and restore the business processes in full
  - Build and distribute an account for the incident, which in particular attributes the failure corresponding to the incident

- Preparedness: in order to be ready to perform the above mission, the organisation must:
  - Define the relevant processes and procedures
  - Allocate the responsibility and deploy the necessary staff
  - Deploy the required tools (e.g. case management and incident tracking)
  - Have a contingency plan to deal with events of an exceptional magnitude
  - Place required external resources on retainer, to handle incidents of an exceptional magnitude or the provide forensics expertise
  - Get insurance against risks (based on a risk / cost analysis)
  - Define metrics for the various processes, track performance, and report to the responsible organisations
  - Test the system based on simulated incidents
  - Regularly update these various elements to ensure they are current.

3.1.8 Audit and Validate

It must be noted that audits focus, in all or in part, on continuous improvement, providing a proactive view rather than solely looking to place blame. Audit requirements are typically mandated by the domains for which the organisation is accountable. There is however a pattern that can be identified:

- Internal audits
  - Regularly perform internal audits of the system.
  - Focus on both compliance to internal and external criteria.
  - Investigate the effectiveness of the risk treatment plan as implemented. Ensure that the objectives are being met.
  - Trigger a review and adjustment process when critical deficiencies (blind spots) are discovered
  - Prepare for external audits. Ensure that recommendations from previous external audits have been properly considered and dealt with.
  - Collect the material and perform the analysis which will allow to prepare updated accounts, to include actual indicators of effectiveness rather than solely relying on the theoretical analysis used in the Analyse and Design phase (see section 3.1.5)

  The internal audits are performed by internal auditors, who work within the organisation and report to the audit committee.

- External audits
  - External audits are performed as dictated by the regulations of each domain of accountability.
  - External audits are also required in most compliance or attestation frameworks.
  - Customers may also place contractual provisions for realizing audits. The industry trend, in particular as it regards Cloud Computing, is to define certification and attestation to best-practice frameworks which can be effective substitutes for client-directed audits.

  The external audits are performed by external auditors, which either perform on the basis of an auditing contract or are hired by an external party (e.g. stakeholder or certification authority).

3.2 The Role of Standards

There are many ways to classify standards. For example, in the A-5 work-package of this project, we notably distinguish *real* standards, from *technical specifications* and *best practices*. In the C-3 work-package of this project, we make another type distinction regarding the scope of standards, based on the four categories of standards defined CEN-CENELEC[2]. We will reuse this classification to structure the discussion of this section. If we restrict ourselves to the IT domain, these 4 categories can be expressed as follows:

1. **Fundamental standards** - *which concern terminology, conventions, signs and symbols, etc.*;

[2] http://www.cencenelec.eu/research/innovation/standardstypes/Pages/default.aspx

2. **Organization standards** - *which describe the functions and relationships of a company, as well as elements such as quality management and assurance, maintenance, value analysis, project or system management*, *etc.*
3. **Specification standards** - *which define characteristics of a product or a service, such as interfaces (APIs), data formats, communication protocols and other interoperability features, etc.*;
4. **Test methods and analysis standards** - *which measure characteristics of a system, describing processes and reference data for analysis*;

In the following paragraphs we examine the role and value of each category of standard as a driver for accountability for organizations.

3.2.1 Fundamental standards

Fundamental standards are like the foundation of a building; they are needed to create solid constructions. They play a role in setting the terminology and concepts that are used by organizations implementing IT systems and they influence other types of standards. From a strategic point of view, it is therefore important to include accountability as a crosscutting concept in fundamental cloud standards where relevant and possible. So far, most of the standardization initiatives in work-package A-5 has precisely been directed at putting accountability in core standards (see [15] for details).

3.2.2 Organizational standards

Organizational standards and Specifications standards form a complementary pair. In a simplified view, we can argue that organizational standards are useful to structure the internal processes of an organization to take accountability practices into account. By contrast, we can also argue that specification standards, by promoting interoperability, enable accountability across the supply chain with external entities. We will first discuss the role of organizational standards.

Organizational standards are not strictly necessary to enable accountability practices within an organization. In theory, this goal can be achieved by applying best practices that have been developed internally. Such practices could notably be inspired by the A4Cloud conceptual framework [3]. This approach has some important drawbacks however. First, it makes it complex for external entities to evaluate the quality of the accountability practices implemented by the organization. Second, it makes comparison between organizations largely impossible, since each organization will be using its own logic and criteria. Standardized approaches solve these two problems by structuring practices in a way that is recognized not only within an organization but also across the whole industry. In addition, such standards can be used as a foundation to build certification schemes, with independent third party auditors, with the benefit of recognition and enhanced trust. This could create a market for “accountability” certification, much like existing ISMS certification today.

There essentially are two competing approaches to embed accountability into organizational standards:

1. Take existing standards in security, governance and compliance, identify their gaps regarding accountability and extend them if needed to cover these gaps.
2. Build a new “accountability management standard”, mirroring ISO 27001 for security for example.

Both approaches have advantages and drawbacks.

Taking an existing organizational standard and extending it to cover accountability practices allows organizations to re-use a framework they already know. This normally minimizes the cost of “adding accountability” to current practices, which in turn facilitates adoption of accountability practices. The Cloud Control Matrix[3] (CCM) is an example of an organizational cloud control framework that uses this attractive approach: all CCM control reference back to existing equivalent controls in other frameworks when they exist (in ISO/IEC 27001, PCI-DSS, ISACA COBIT, NIST, etc.). Using this approach for accountability means however that accountability is “added” to current practices and is not the backbone of the organizational practices. Building a real accountability organizational standard from

[3] https://cloudsecurityalliance.org/research/ccm/

scratch would allow describing governance, risk and compliance processes that would be structured around accountability. Building such a standard with industry consensus is however a huge task in itself.

3.2.3 Specifications standards

Specification standards allow accountability to be expressed and transferred along the supply chain, by promoting common metrics, common semantics, common data formats. This ultimately leads to automation of accountability interactions, in turn bringing cost reduction, which makes the value proposition of accountability more attractive. These points are extensively discussed in task C-3 [16].

3.2.4 Test methods and analysis standards

This last category can be exemplified through software testing standards such as [17], but is less relevant to our work on accountability, so we will not discuss it further.

3.3 The Accountability Maturity Model

A4Cloud has identified the need to aid organisations (in particular, SMEs) to quantitatively assess their accountability practices as a first step to improve them. The proposal was to develop an Accountability Maturity Model (AMM) that could be used to quantitatively assess the maturity of the mechanisms deployed to support accountability. This practice is not new on ICT, where maturity models have existed for several years (for example [18]). However, the novelty of the AMM is its focus on capturing both the maturity of individual organisations in terms of accountability practices, as well as a measurement of the appropriateness of the measures used across whole cloud supply chains.
The rest of this section further develops the notion of AMM introduced in D:C-2.1 by:

- Refining the initially defined set of accountability controls (D:C-2.1) and associated metrics (D:C-5.2), in order to allow the implementation of semi-automated accountability assessment/certification processes (cf. Section 3.3.1).
- Associating the evaluation of controls and metrics from the AMM to the architecture proposed in this deliverable, in particular based on the accountability life cycle (Section Introduction to the Organisational Lifecycle for Accountability) and the more abstract cloud reference architecture. This contribution will be presented in Section 3.3.2.

In order to align the AMM (and its architectural perspective), to A4Cloud's standardization efforts (cf. Deliverable D:A-5.1), the context for the discussion presented in this section will be both the Cloud Security Alliance's Cloud Control Matrix[4] and then Open Certification Framework[5].

3.3.1 Accountability Controls and Metrics

This section further elaborates on the combined usage of both the accountability maturity model (AMM, cf. Deliverable D:C-2.1), and the accountability metrics (cf. Deliverable D:C-5.2) for the purposes of quantitatively assessing an organization's the level of accountability.

3.3.1.1 Accountability Maturity Model – controls and scoring

In analogy to widely used maturity models, the AMM is composed of two elements:

- ***Control Framework:*** a set of controls that an organisation will apply to address requirements such as security, privacy or accountability.

- ***Scoring Methodology:*** a technique used to assign a quantitative or qualitative value that rates the level of implementation of the control framework. The assigned value is known as a "maturity level": the score typically increases with the level of sophistication of control implementation.

[4] Please refer to https://cloudsecurityalliance.org/research/ccm/
[5] Please refer to https://cloudsecurityalliance.org/star/

The development of the AMM in D:C-2.1 resulted on a set of 74 accountability controls, which were selected from well-known control frameworks (CSA CCM, NIST 800-53 rev. 4[6], and AICPA/CICA Privacy Maturity Model[7]). The selection criteria (fully documented in D:C-2.1), started with a gap analysis based on a set of mapping rules that relate individual controls to each one of the accountability attributes defined by A4Cloud. At a glance, the applied mapping rules considered that the ultimate goal of accountability is to provide information to external stakeholders, therefore only if information to internal stakeholders is provided then a gap was identified. The full set of resulting accountability controls is shown in D:C-2.1, Appendix F. For the sake of alignment with A4Cloud standardization activities, D-2 will focus only on the subset of AMM controls[8] associated to CSA CCM (cf. Table 2).

No.	Control name	Control code	O	V	A	T	R	L	Rem
1	Business Continuity Management & Operational Resilience Business Continuity Testing	BCR-02							
2	Business Continuity Management & Operational Resilience Impact Analysis	BCR-09							
3	Business Continuity Management & Operational Resilience Management Program	BCR-10							
4	Change Control & Configuration Management Unauthorized Software Installations	CCC-04							
5	Change Control & Configuration Management Production Changes	CCC-05							
6	Data Security & Information Lifecycle Management Classification	DSI-01							
7	Data Security & Information Lifecycle Management Data Inventory / Flows	DSI-02							
8	Datacentre Security Asset Management	DCS-01							
9	Encryption & Key Management Entitlement	EKM-01							
10	Encryption & Key Management Key Generation	EKM-02							
11	Governance and Risk Management Management Support/Involvement	GRM-05							
12	Identity & Access Management Credential Lifecycle / Provision Management	IAM-02							
13	Identity & Access Management Trusted Sources	IAM-08							
14	Identity & Access Management *User Access Authorization*	IAM-09							
15	Identity & Access Management *User Access Reviews*	IAM-10							
16	Identity & Access Management *User Access Revocation*	IAM-11							
17	Identity & Access Management *User ID Credentials*	IAM-12							
18	Identity & Access Management *Utility Programs Access*	IAM-13							
19	Infrastructure & Virtualization Security Audit Logging / Intrusion Detection	IVS-01							
20	Security Incident Management, E-Discovery & Cloud Forensics *Incident Management*	SEF-02							
21	Security Incident Management, E-Discovery & Cloud	SEF-03							

[6] Please refer to http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf
[7] Please refer to http://www.cica.ca/resources-and-member-benefits/privacy-resources-for-firms-and-organisations/item47888.aspx
[8] Although these might be extended during the duration oft he work package.

No.	Control name	Control code	O	V	A	T	R	L	Rem
	Forensics *Incident Reporting*								
22	Security Incident Management, E-Discovery & Cloud Forensics *Incident Response Legal Preparation*	SEF-04							
23	Supply Chain Management, Transparency and Accountability *Data Quality and Integrity*	STA-01							
24	Supply Chain Management, Transparency and Accountability *Incident Reporting*	STA-02							
25	Supply Chain Management, Transparency and Accountability *Provider Internal Assessments*	STA-04							
26	Supply Chain Management, Transparency and Accountability *Supply Chain Agreements*	STA-05							
27	Supply Chain Management, Transparency and Accountability *Supply Chain Governance Reviews*	STA-06							
28	Supply Chain Management, Transparency and Accountability *Supply Chain Metrics*	STA-07							
29	Supply Chain Management, Transparency and Accountability *Third Party Assessment*	STA-08							
30	Supply Chain Management, Transparency and Accountability *Third Party Audits*	STA-09							
31	Threat and Vulnerability Management Anti-Virus / Malicious Software	TVM-01							

Table 2. Accountability controls from CSA CCM [19]. Legend: (O)bservability, (V)erifiability, (A)ttributability, (T)ransparency, (R)esponsibility, (L)iability, (Rem)ediability

In the proposed AMM, the set of accountability controls is complemented with a scoring methodology (quantitatively) representing how well all of these controls have been implemented by the organization under assessment. As mentioned in D:C-2.1 , it is worth to highlight that there is not standard scoring methodology adopted by state of the art maturity models, in many cases even the semantic associated with the numeric output (and also the actual number of maturity levels) is different. Once again, and for the particular purpose of D-2 will be adopted the methodology used by CSA CCM just as briefly presented next. A suitable procedure for assigning maturity levels based on a CCM assessment has been developed in the context of the Open Certification Framework (OCF). When an organisation is audited, a *Management Capability Score* (i.e., maturity level) will be assigned to each of the control areas on the CCM. For the sake of usability, the management capability of the *domains (not the individual controls)* is scored on a scale of 1-15. These scores have been divided into five different categories that describe the type of approach characteristic of each group of scores:

a) 1-3: No formal approach.
b) 4-6: Reactive approach.
c) 7-9: Proactive approach.
d) 10-12: Improvement-based approach.
e) 13-15: Optimising approach.

When assigning a score to a control domain, the following five factors are considered (all or any applicable combination of them):

1. Communication and Stakeholder Engagement.
2. Policies, Plans and Procedures, and a Systematic Approach.
3. Skills and Expertise.
4. Ownership, Leadership, and Management.
5. Monitoring and Measuring.

The lowest score against any one of those five factors will be the score awarded for the control domain. The organisation under evaluation will be awarded the lowest score it achieved for any of the factors assessed against the CCM domains. Once the assessor has assessed all of the control domains, there will be 16 scores (one per-domain of the CCM). The average score will be used to assign the overall Management Capability Score (or *award*) for the organisation, according to the following rules:

- If the organisation has an average score of less than 3, it will receive a certificate with *no award*.
- If the organisation has an average score between 3 and 6, it will receive a *bronze award*.
- If the organisation has an average score between 6 and 9, it will receive a *silver award*.
- If the organisation has an average score greater than 9, it will receive a *gold award*.

A typical (state of the art) maturity model would only implement the two elements discussed above (controls and scoring methodology), however two limitations appear in relationship to (i) the subjectivity associated with the underlying assessment process, and (ii) the level of automation that could be achieved. In order to overcome these limitations, EU projects like CIRRUS[9], CloudWatch[10], SPECS[11] and Cumulus[12] have been looking at potential mechanisms to implement the continuous assessment of security/privacy in a semi-automated manner for Cloud systems. A promising solution, is based on the use of metrics, just as proposed also by A4Cloud's Deliverable D:C-5.2. The rest of this subsection briefly summarizes the main findings from D:C-5.1/D:C-5.2, and elaborates about the relationship between the AMM and the developed accountability metrics.

3.3.1.2 Accountability Maturity Model – associated metrics

Metrics are defined by NIST [20] as *"a standard of measurement that defines the conditions and the rules for performing the measurement and for understanding the results of a measurement."*, and keep a close relationship to the concept of accountability being developed by A4Cloud. From a technical viewpoint, metrics are widely used as an instrument for verifying/monitoring 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 [21]. In this context, *accountability metrics* become an important aspect of the proposed AMM, since they can be considered as a means for showing that proper mechanisms for privacy, security and information governance are in place and indeed support accountability. However, in order to fulfil with this vision, it is necessary to relate the accountability controls from the AMM to the accountability metrics developed in the context of D:C-5.2. This is possible thanks to the approach proposed by A4Cloud to develop meaningful accountability metrics. The steps of the proposed (bottom-up) approach are as follow:

1. To analyse relevant control frameworks in the light of Accountability Attributes. The goal of this step is to select those controls that influence Accountability to some extent.
2. To study the nature of the control, in order to identify whether there is any quantifiable element in the description of the control that is susceptible to being measured. Qualitative elements may be identified too, if they have at least an ordinal nature.
3. To define a metric that measures the identified elements, using the qualitative or quantitative elements identified in the previous step.
4. To check that the metric supports the concept of Accountability and, in particular, the Accountability Attributes for which it is related to.

Applying the previously described approach to three selected control frameworks (CSA CCM, NIST 800-53 rev. 4, and AICPA/CICA Privacy Maturity Model), in D:C-5.2 where elicited a total of 39 accountability metrics (cf. Table 3).

[9] Please refer to http://www.cirrus-project.eu/
[10] Please refer to http://www.cloudwatchhub.eu/
[11] Please refer to http://specs-project.eu/
[12] Please refer to http://www.cumulus-project.eu/

Metric	Name	T	V	A	O	Rem	R	L
	Verifiability and Compliance							
1	Authorized collection of PII		X					
2	Privacy Program Budget		X					
3	Privacy Program Updates		X				X	
4	Periodicity of Privacy Impact Assessments for Information Systems		X					
5	Number of privacy audits received	X	X					
6	Successful audits received	X	X		X			
7	Record of Data Collection, Creation, and Update		X					
8	Data classification		X					
9	Coverage of Privacy and Security Training		X					
10	Account of Privacy and Security Training		X					
11	Level of confidentiality		X					
12	Key Exposure Level		X					
13	Data Isolation Testing Level		X					
	Transparency, Responsibility and Attributability							
14	Type of Consent	X						
15	Type of notice	X						
16	Procedures for Data Subject Access Requests	X						
17	Number of Data Subject Access Requests	X						
18	Responded data subject access requests	X			X			
19	Mean time for responding Data Subject Access Requests	X						
20	Readibility (Flesch Reading Ease Test)	X						
21	Rank of Responsibility for Privacy						X	X
22	Certification of acceptance of responsibility						X	X
23	Frequency of certifications		X				X	X
24	Log Unalterability		X	X				
25	Identity Assurance		X	X				
26	Mean time to revoke users			X			X	
	Remediability and Incident Response							
27	Mean time to respond to complaints	X				X		
28	Number of complaints	X				X		
29	Reviewed complaints	X				X		
30	Number of privacy incidents	X			X			
31	Coverage of incident notifications	X			X	X		
32	Type of incident notification	X				X		
33	Privacy incidents caused by third parties	X			X	X		
34	Number of Business Continuity Resilience (BCR) plans tested		X			X		
35	Maximum tolerable period for disruption (MTPD)					X		
36	Sanctions	X				X		X
37	Incidents with damages	X				X		X
38	Total expenses due to compensatory damages	X				X		X
39	Average expenses due to compensatory damages	X				X		X

Table 3. Catalog of Accountabiliy Metrics (D:C-5.2). Legend: (O)bservability, (V)erifiability, (A)ttributability, (T)ransparency, (R)esponsibility, (L)iability, (Rem)ediability

Based on this full catalogue of metrics, in Table 4 is shown the specific set associated to the AMM controls (as previously presented in Table 3).

Control group	Control name	Control code	Accountability Metric
Business Continuity Management & Operational Resilience	Business Continuity Testing	BCR-02	Metric 34. Number of Business Continuity Resilience (BCR) plans tested
	Impact Analysis	BCR-09	Metric 35. Maximum tolerable period for disruption (MTPD)
	Management Program	BCR-10	n/a
Change Control & Configuration Management	Unauthorized Software Installations	CCC-04	n/a
	Production Changes	CCC-05	n/a
Data Security & Information Lifecycle Management	Classification	DSI-01	Metric 8. Data classification
	Data Inventory / Flows	DSI-02	n/a
Datacentre Security	Asset Management	DCS-01	n/a
Encryption & Key Management	Entitlement	EKM-01	Metric 11. Level of confidentiality Metric 12. Key Exposure Level
	Key Generation	EKM-02	n/a
Governance and Risk Management	Management Support/Involvement	GRM-05	n/a
Identity & Access Management	Credential Lifecycle / Provision Management	IAM-02	Metric 25. Identity Assurance Metric 26. Mean time to revoke users
	Trusted Sources	IAM-08	n/a
	User Access Authorization	IAM-09	n/a
	User Access Reviews	IAM-10	n/a
	User Access Revocation	IAM-11	Metric 26. Mean time to revoke users
	User ID Credentials	IAM-12	n/a
	Utility Programs Access	IAM-13	n/a
Infrastructure & Virtualization Security	Audit Logging / Intrusion Detection	IVS-01	n/a
Security Incident Management, E-Discovery & Cloud Forensics	*Incident Management*	SEF-02	n/a
	Incident Reporting	SEF-03	Metric 22. Certification of acceptance of responsibility Metric 23. Frequency of certifications
	Incident Response Legal Preparation	SEF-04	Metric 31. Coverage of incident notifications Metric 32. Type of incident notification Metric 33. Privacy incidents caused by third parties Metric 39. Average expenses due to compensatory

Control group	Control name	Control code	Accountability Metric
			damages
Supply Chain Management, Transparency and Accountability	*Data Quality and Integrity*	STA-01	n/a
	Incident Reporting	STA-02	Metric 36. Sanctions Metric 37. Incidents with damages Metric 38. Total expenses due to compensatory damages
	Provider Internal Assessments	STA-04	n/a
	Supply Chain Agreements	STA-05	Metric 31. Coverage of incident notifications Metric 32. Type of incident notification Metric 33. Privacy incidents caused by third parties
	Supply Chain Governance Reviews	STA-06	n/a
	Supply Chain Metrics	STA-07	n/a
	Third Party Assessment	STA-08	n/a
	Third Party Audits	STA-09	n/a
Threat and Vulnerability Management	Anti-Virus / Malicious Software	TVM-01	n/a

Table 4. AMM - controls and metrics

As shown in Table 4, the accountability metrics elicited in D:C-5.2 cover approximately 32% (10) of the controls from the proposed AMM. The other 21 controls do not have any metric associated to them. Table 4 also shows that out of 39 accountability metrics, only 14 different metrics (approx. 35%) were mapped to the AMM derived from CSA CCM. The resulting 10 "measurable AMM controls", are associated with metrics that can be assessed either automatically (e.g., Metric 26 Mean time to revoke users) or through human intervention (e.g., Metric 38 Total expenses due to compensatory damages). Section 3.4.2 will further elaborate about the usage of these metrics, from the perspective of the A4Cloud's accountability life cycle and the more abstract cloud reference architecture.
The fact of having "non-measurable AMM controls" does not mean that these cannot be assessed at all, by the contrary in these cases the "traditional" audit practice will prevail and the control(s) will be evaluated through provided evidence while applying self-assessments or third-party assessments. However, it should be noted that in these cases the confidence associated to the results from the evaluation would be low. Measurable controls, allow higher levels of confidence thanks to the usage of associated metrics[13]. The relationship among assessment methods/consistency of the measurement, and resulting level of confidence can be observed in Figure 4.

Higher-levels of confidence can be achieved through the use of metrics, however the associated trade-offs have to be carefully analysed by organizations. In particular we refer to the associated economic (e.g., upgrading the ICT infrastructure to support continuous monitoring), and performance costs (e.g., automated assessment will probably introduce an overhead to the system under evaluation).

[13] That is the goal of SMART (Specific, Measurable, Achievable, Relevant, Timely) metrics.

Source of Assessment \ Consistency	Informal (Level 1)	Structured (Level 2)	Automated (Level 3)
Self-assessment (Level 1)	**0**	**1**	**1**
Third party assessment (Level 2)	**1**	**2**	**2**
User/Publicly Verifiable (Level 3)	**1**	**2**	**3**

Figure 4. Metrics confidence matrix (D:C-5.2).

The next version of this deliverable (D:D-2.4) will apply the elicitation methodology described in D:C-5.2 to revisit the accountability metrics shown in Table 4, and extend their coverage within the AMM. Future versions of this document will discuss in further details the trade-offs associated to the automated evaluation of the accountability metrics, from the AMM perspective, and taking into account the final version of the proposed A4Cloud architecture. The next section provides an initial analysis of the AMM usage both from the (i) A4Cloud accountability life cycle, and (ii) cloud reference architecture perspectives.

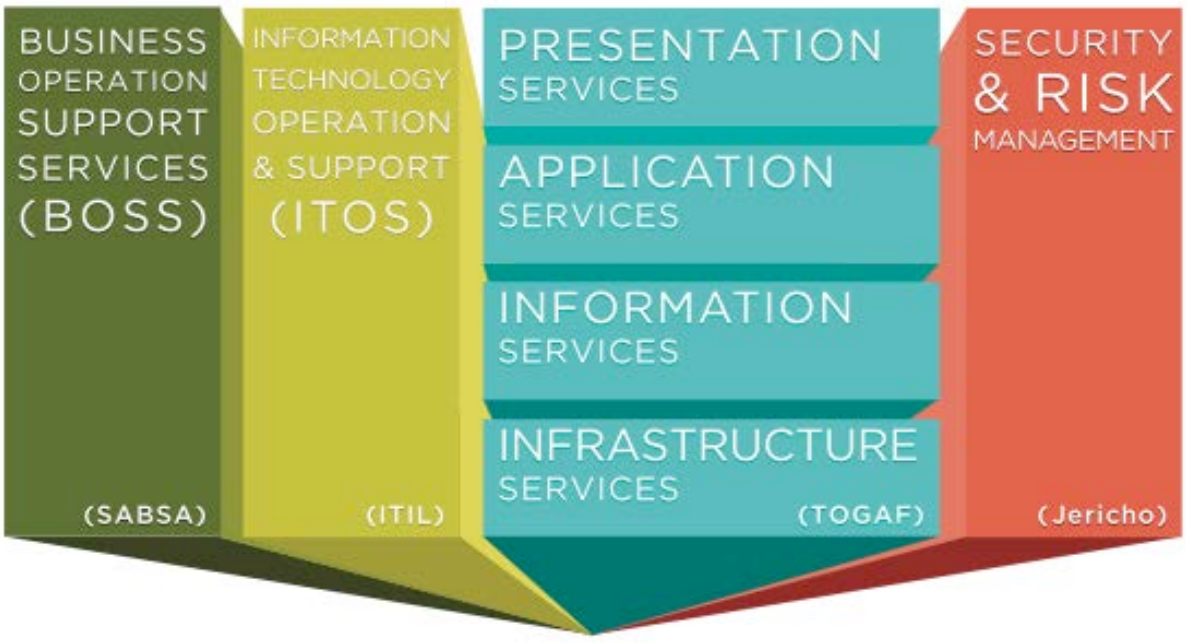

Figure 5. CSA Enterprise Architecture

3.3.2 Accountability Assessment - Architectural Perspective

This section is analyses the usage and applicability of the AMM (and associated metrics) to an abstract cloud reference architecture and the A4Cloud accountability life cycle. The presented analysis also discusses open research challenges to be further elaborated in D:D:2.4

3.3.2.1 Applying the AMM to a Cloud Reference Architecture

In order to provide *useful* information about the accountability level achieved by an organization, both the AMM and metrics (as presented in Section 3.3.1) must be applied to a specific Cloud context. This context might refer for example to some particular Cloud deployment/service model, possibly resulting from a preliminary risk analysis or an existing set of security/privacy requirements. In any case, the proposed AMM/metrics cannot be applied on an isolated manner. This section elaborates on using the AMM[14] for the quantification of organizational accountability levels, based on the A4Cloud toolbox. The selected approach directly leverages the abstraction known as Cloud Reference Architecture (CRA).

A CRA is typically comprised of a framework (i.e., methodology and tools) that enables security architects, enterprise architects, and risk management professionals to leverage a common set of solutions (patterns). These solutions fulfil a set of common requirements that risk managers must assess regarding the operational status of internal IT security and CSP controls (e.g., from AMM). The controls are expressed in terms of "security capabilities" and designed to create a common roadmap to meet the security needs of their business. NIST Special Publication 500-299 (draft) [22] and CSA Enterprise Architecture (CSA EA, formerly known as Trusted Cloud Initiative[15]) are two commonly referenced CRA's at the state of the art. For the sake of A4Cloud standardization, the discussion presented in the rest of this section will be focused on CSA EA.

The CSA EA (shown in Figure 5) is structured in a hierarchical manner. Eight domains exist at the top level, which are composed of containers, and in turn these are comprised of one or more capabilities. As mentioned early on this section, the CSA EA capabilities can be mapped to the AMM controls, just as shown in Table 5.

A4Cloud AMM		CSA EA		
Control name	**Control code**	**Domain**	**Container**	**Capability**
Business Continuity Testing	BCR-02	BOSS	Operational Risk Management	Business Continuity
Impact Analysis	BCR-09	ITOS	Service Delivery	Information Technology Resiliency - Resiliency Analysis
Management Program	BCR-10	SRM	Policies and Standards	Operational Security Baselines
Unauthorized Software Installations	CCC-04	ITOS	Service Support	Configuration Management - Software Management
Production Changes	CCC-05	ITOS	Service Support	Release Management
Classification	DSI-01	BOSS	Data Governance	Data Classification
Data Inventory / Flows	DSI-02	BOSS	Data Governance	Handling / Labelling / Security Policy
Asset Management	DCS-01	ITOS	Service Support	Configuration Management - Physical Inventory
Entitlement	EKM-01	SRM	Cryptographic Services	Key Management
Key Generation	EKM-02	SRM	Cryptographic Services	Key Management

[14] In this section, the term AMM will also refer tot he associated accountability metrics.
[15] Please refer to https://cloudsecurityalliance.org/research/eawg/

A4Cloud AMM		CSA EA		
Control name	**Control code**	**Domain**	**Container**	**Capability**
Management Support/Involvement	GRM-05	SRM	Governance Risk & Compliance	Compliance Management
Credential Lifecycle / Provision Management	IAM-02	SRM	Policies and Standards	n/a
Trusted Sources	IAM-08	Information Services	User Directory Services	Active Directory Services, LDAP Repositories, X.500 Repositories, DBMS Repositories, Meta Directory Services, Virtual Directory Services
User Access Authorization	IAM-09	SRM	Privilege Management Infrastructure	Identity Management - Identity Provisioning
User Access Reviews	IAM-10	SRM	Privilege Management Infrastructure	Authorization Services - Entitlement Review
User Access Revocation	IAM-11	SRM	Privilege Management Infrastructure	Identity Management - Identity Provisioning
User ID Credentials	IAM-12	SRM	Policies and Standards	Technical Security Standards
Utility Programs Access	IAM-13	SRM	Privilege Management Infrastructure	Privilege Usage Management - Resource Protection
Audit Logging / Intrusion Detection	IVS-01	BOSS	Security Monitoring Services	SIEM
Incident Management	SEF-02	ITOS	Service Support	Security Incident Management
Incident Reporting	SEF-03	BOSS	Human Resources Security	Employee Awareness
Incident Response Legal Preparation	SEF-04	BOSS	Legal Services	Incident Response Legal Preparation
Data Quality and Integrity	STA-01	SRM	Governance Risk & Compliance	Vendor Management
Incident Reporting	STA-02	ITOS	Service Support - Incident Management	Cross Cloud Incident Response
Provider Internal Assessments	STA-04	SRM	Governance Risk & Compliance	Vendor Management
Supply Chain Agreements	STA-05	BOSS	Legal Services	Contracts
Supply Chain Governance Reviews	STA-06	SRM	Governance Risk & Compliance	Vendor Management
Supply Chain Metrics	STA-07	ITOS	Service Delivery	Service Level Management - Vendor Management
Third Party Assessment	STA-08	SRM	Governance Risk & Compliance	Vendor Management
Third Party Audits	STA-09	BOSS	Compliance	Third-Party Audits
Anti-Virus / Malicious Software	TVM-01	SRM	Infrastructure Protection Services	Anti-Virus

Table 5. Mapping the AMM to CSA's Cloud Reference Architecture (CSA EA)

Based on Table 5, it is possible to use the AMM to quantitatively assess the accountability level of an organization through the following sequence of steps:

1. Map the organization's security architecture components to the capabilities shown on Table 5. Additional guidance to perform this mapping can be found on the CSA EA specification.
2. Based on the previous mapping, select the AMM controls (also from Table 5) that correspond to each component's capability.
3. Using the information from Table 4, classify the AMM controls from Step 2 into "Quantifiable" (C_Q, if at least one accountability metric is associated to them) and "No Quantifiable" (C_{NQ}, if the control is not associated to any accountability metric).
4. Evaluate the accountability controls in the following manner:
    a. The C_Q controls should be measured according to the respective metric definition (cf., Deliverable D:C-5.2).
    b. The C_{NQ} controls should be assessed (e.g., by a human auditor) according to the applicable practice (for example, in the case of CSA CCM please refer to [19]).
5. Aggregation of results (out of scope in Deliverable D:D-2.3):
    a. The measurement results associated to the controls C_Q can be aggregated by using the proposal in D:C-5.2, or a state of the art techniques like [23] or [24].
    b. The assessment results obtained for C_{NQ} can be scored to an overall maturity level, following the rules presented in Section 3.3.1.1.

The aggregated result from Step 5, is the actual maturity level associated to the architectural component being evaluated. The accountability quantification process described above is shown in Figure 6. Evaluating the accountability level (architectural approach).. A preliminary usage of the proposed process will be presented in Section 5.

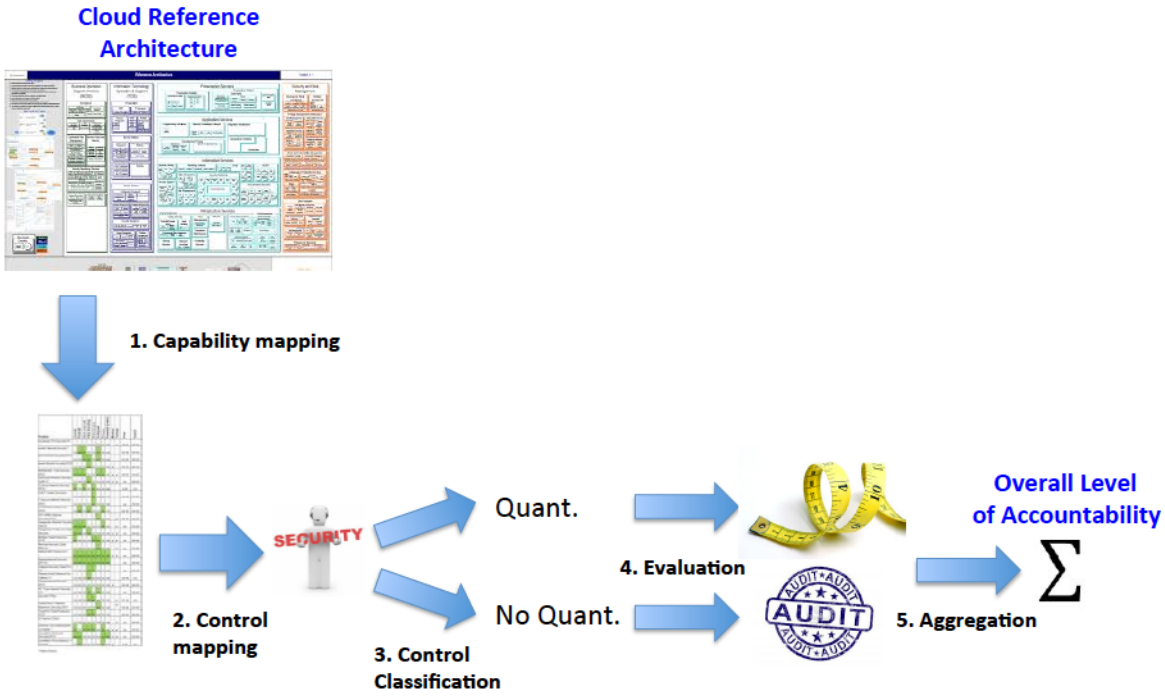

Figure 6. Evaluating the accountability level (architectural approach).

3.3.2.2 Example: applying the AMM to the COAT tool

The assessment process described in Section 3.3.2.1 can be applied to any of the A4Cloud tools presented in Section 5. Next, we show a step-by-step example to illustrate the use of AMM to the Cloud Offerings and Advisory Tool (COAT cf., Section 5.1.2).
As a first step, we proceed to identify the COAT components namely User Requirement, Matchmaker – Service, Matchmaker – Contract, Comparison, Information/Explanation/Guidance, and Logging. These components are shown in Figure 23.

The second step consists in mapping CSA EA capabilities to COAT components. This process adds a degree of subjectivity that can be reduced if expert knowledge about the tool is available. As COAT is basically used by (prospective) Cloud Customers during the procurement/planning stage, then we can expect that CSA EA capabilities related to risk management, and service/capacity/business planning will be mapped. The result of this second step is shown in Table 6.

COAT Component	CSA EA		
	Domain	Container	Capability
User Requirement	BOSS	Operational Risk Management	Risk Management Framework
	BOSS	Operational Risk Management	Business Continuity
	BOSS	Operational Risk Management	Independent Risk Management
	Security and risk management	Polices and Standards	Best practices & regulatory
	ITOS	Service Support	Configuration Management
Matchmaker - Service	ITOS	Service Delivery	Service Management
Matchmaker - Contract	BOSS	Legal	Contracts
Comparison			
Information/Explanation/Guidance	BOSS	Legal	eDiscovery
	Information services	Reporting services	Reporting tools
	Information services	Reporting services	Business intelligence
	Security and risk management	Governance, Risk and Compliance	Vendor Management
Logging	Information services	Service Support	Service Events

Table 6. Mapping COAT components to CSA EA capabilities

Next, we need to sequentially apply the mapping shown in Tables 5 and 4 to realize the accountability controls and accountability metrics (respectively) associated to the COAT components. The result of this mapping has been illustrated with a tree-like structure in Figure 7.

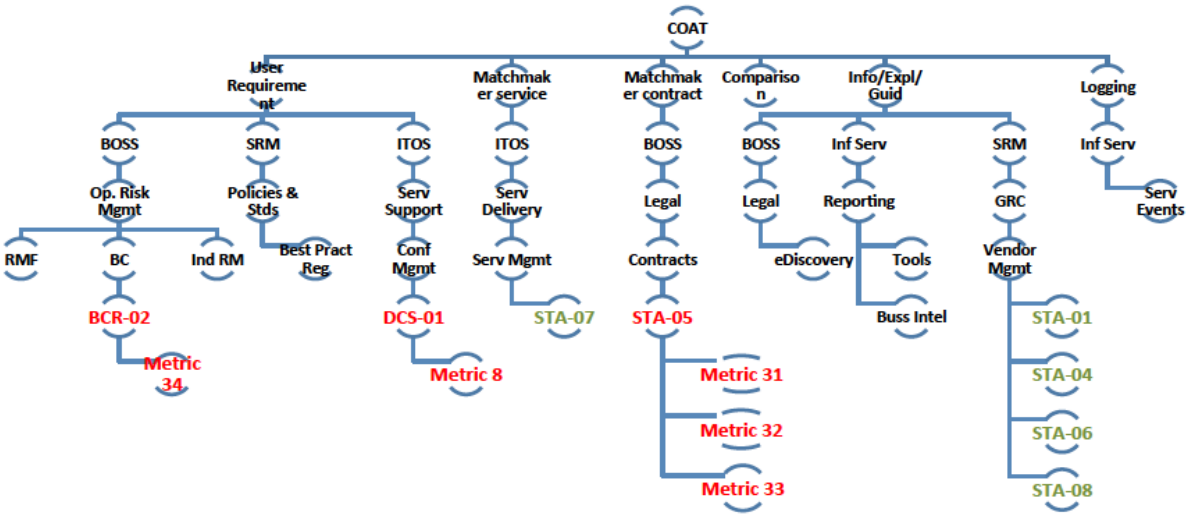

Figure 7. AMM controls and metrics related to COAT.

The first level of the tree shown in Figure 8 refers to COAT's components then subsequent levels show the CSA EA's Domain, Container, Capability mappings followed by the corresponding AMM Control and Metric. As discussed in Section 3.4.2.1, not all COAT components are related to accountability e.g., the Comparison component. Furthermore, some components do not have a metric at all (like in the case of the Matchmaker – Service) whereas some others must be assessed through more than one metric (e.g., the Matchmaker – Contract). The overall accountability level provided by COAT can be computed by *composing* the following two assessments C_Q and C_{NQ}:

$$C_Q = Metric_{14} + Metric_8 + Metric_{31} + Metric_{32} + Metric_{33} \ldots.(1)$$

$$C_{NQ} = STA_1 + STA_4 + STA_6 + STA_7 + STA_8 \ldots. (2)$$

$$AccLevel_{COAT} = C_Q + C_{NQ} \ldots. (3)$$

The final $AccLevel_{COAT}$ shown in expression (3) represents the overall accountability level that can be provided to an organization adopting the COAT tool, which is the *total maturity level of the tool*. It has to be noticed that in expressions (1), (2), and (3) the "+" operator denotes composition and not an arithmetic addition.

Thanks to the proposed approach, it is possible for COAT to guide customers in designing the actual thresholds to expect from a CSP Service Level Agreement (SLA), in particular related to notification of privacy incidents (Metrics 31 and 32). Even for AMM controls not associated to accountability metrics, it is possible to leverage tools like CSA CAIQ[16] to aid in the assessment of CSP offers with COAT. For example, even though AMM control STA-07 (Figure 7), is not associated to any accountability metric, it is related to the following CSA CAIQ assessment questions:

- Are policies and procedures established, and supporting business processes and technical measures implemented, for maintaining complete, accurate and relevant agreements (e.g., SLAs) between providers and customers (tenants)?
- Do you have the ability to measure and address non-conformance of provisions and/or terms across the entire supply chain (upstream/downstream)?
- Can you manage service-level conflicts or inconsistencies resulting from disparate supplier relationships?
- Do you review all agreements, policies and processes at least annually?

As mentioned in Section 3.3.2.1, the composition of $\boldsymbol{C_Q}$ (measurement results) and $\boldsymbol{C_{NQ}}$ (answers to CSA CAIQ), has been explored in the past (e.g., **[23]** or **[24]**) and will be further scoped in the next version of this deliverable.

[16] Please refer to https://cloudsecurityalliance.org/research/cai/

3.4 Demonstrating Accountability – The Account

As discussed above in sections 2, 3.1.5 and 3.1.7, provision of *accounts* is an important part of the organisational lifecycle, and the means of demonstrating accountability. In this section it is explained what accounts are (in 3.4.1: general concepts); who produces them (in 3.4.2); how their properties can vary (in 3.4.3); how this analysis relates to the functional elements of accountability corresponding to section 3.1 (in 3.4.4) and what the relationship is between accounts and certification (in 3.4.5). Furthermore, examples of the two main types of account are given, namely evidence about compliance (in 3.4.7) and about data breaches or policy violations (in 3.4.8). Within this discussion, ways of improving account provision are given, and the process around the provision of accounts and their verification is clarified.

3.4.1 General Concepts

In this section it is explained what an account is, and the varying properties of accounts are considered. As stated in [3], "*an account can simply be defined as 'a report or description of an event'.* Such an event can either be expected as "prescribed" by one of the organization's obligations or it can be an incident such as a breach or failure and thus unexpected. The previous definition can also be extended in order to cover *proactive* reports: an accountable organization can also generate some reports prior to making the service available: such a report may indicate the "quality" and "security" level of the services and may include certifications of practices or may include a data protection impact assessment. Another essential element of the proactive account is of course the policy describing the normative, regulatory and contractual obligations agreed between the Cloud Provider and its customer. By analysing the policy only, the Auditor, the Data Protection Authority or even the Data Subject may discover some issues or inconsistencies.

In addition to proactive reports, the Accountable Organization should also report while its services are operational. In this case, the Accountable Organization will either validate its operations or inform on an incident. As explained in [3], while describing such an event, be it expected (legitimate event) or unexpected (incident), " *the account should generally include the answers to what are traditionally referred to as the reporter's questions […] backed up with as much evidence as possible to validate the account".* These questions are listed in the following (we refer the reader to Section 4.3.3 for more information on evidence):

- *Who?* The account should provide information on all cloud actors involved in the actual event. This information will especially be very helpful for the Auditor or the Data Protection Authorities to identify the responsible or liable actor.
- *What?* The report should describe all actions taken with respect within this event or provide the details on the incident.
- *Where?* The answer to such a question is especially helpful while verifying the compliance with respect to data transfer policies.
- *When?* The account should mention the time (preferably a timestamp) and duration of the actual event.

While these four questions should definitely be answered in the case of both expected and unexpected events, the account on legitimate events may also include some more details on the process in order to demonstrate the compliance to the corresponding policy rule by answering the following two additional questions:

- *Why?* The answer to this question will simply be the obligation or policy rule the accountable organization is aiming at enforcing.
- *How?* The report should include as much details as possible on the means used to achieve the corresponding action. For example, to demonstrate that a Cloud Provider implements security and privacy measures, it should provide details of the underlying functions such as the encryption algorithm, the size of the encryption key, etc.

On the other hand, although an account describing an incident cannot easily answer the previous two questions, it should nevertheless provide some information on remediation and hence answer the following question:

- *What Next?* In [3] authors note that an account is used in a *"prospective function"*; hence together with the description of the incident the account should ideally contain some additional information on future remedial actions and the adopted measures to prevent the recurrence of such an undesired event.

Although the description of the event or process is an essential element, the account should also carry the following attributes:

- The account recipient: This is the actor who receives the account. Depending on the recipient, the level of detail in the description of the event may change.
- Evidence: Relevant information to support explanation and justification about assertions (for further discussion see section 4.3.3).
- Timestamp and signature: The accountable organization is of course responsible for producing the account and therefore should sign the entire report including the date. Accounts on legitimate events may be periodic and could sometimes be used as evidence for prior events whenever an incident happens in the future. A timestamp in the report hence becomes mandatory.

3.4.2 Interactions between Cloud Actors Related to Accounts

As discussed further within D:32.1 [3], a cloud actor (accountor) is accountable to certain other cloud actors (accountees) within a cloud ecosystem for:

- **Norms:** the obligations and permissions that define data practices; these can be expressed in policies and they derive from law, contracts and ethics.
- **Behaviour:** the actual data processing behaviour of an organisation.
- **Compliance**: entails the comparison of an organisation's actual behaviour with the norms.

For the project scope, the accountors are cloud actors that are organisations (or individuals with certain responsibilities within those) acting as a data steward (for other people's personal and/or confidential data). The accountees are other cloud actors, that may include private accountability agents, consumer organisations, the public at large and entities involved in governance.

Contracts express legal obligations and business considerations. Also, policies may express business considerations that do not end up in contracts. Enterprise policies are one way in which norms are expressed, and are influenced by the regulatory environment, stakeholder expectations and the business appetite for risk. By the accountor exposing the norms it subscribes to and the things it actually does, via an *account*, an external agent can check compliance.

Generally speaking, the sort of information that an organisation needs to measure and demonstrate in such an account include: policies, executive oversight, staffing and delegation, education and awareness, ongoing risk assessment and mitigation, program risk assessment oversight and validation; event management and compliance handling; internal enforcement; redress [25] [9]. Existing organisational documents can often be used to support this analysis [10]. Measurement of the achievement needs to be done in conjunction with the organisation and the external agents that judge it, which is dependent upon the circumstances, and to other entities that may need to be notified. Some examples of accounts that may be provided to cloud actors fulfilling certain data protection roles in a given context are shown in Table 7.

Type of Account	Data Protection Roles	Example Cloud Actor producing the Account
Account for self-certification/verification	Data Controller (DC), for Data Protection Authorities (DPAs) and their customers	Organisational Cloud Customer
Periodic internal reviews (to check that mechanisms are operating as needed and update if required)	DC or Data Processor (DP), for themselves or auditors	Organisational Cloud Customer, Cloud Provider
Evidence provided by risk analysis, PIAs and DPIAs (including assessment along the CSP chain and how this was acted upon)	DC, for DPAs and their customers	Organisational Cloud Customer
External certification e.g. BCRs, CBPRs, CSA OCF level 3, privacy seals, accountability certifications, security certifications	DC or DP, for certification bodies (evidence for certification) or for customers (evidence of certification)	Organisational Cloud Customer, Cloud Provider
External audit (ongoing)	DC or DP, for auditors (evidence) or customers (audit output)	Organisational Cloud Customer, Cloud Provider
Verification by accountability agents	DC to agent, output to DPA	Organisational Cloud Customer
Evidence about fault if data breach	DC to DS, DC to DPA, DP to DC, DP to DP	Organisational Cloud Customer, Cloud Provider

Table 7: Accounts provided by whom to whom and in what circumstances

It is not just a question of interaction between actors in the provision of accounts, but also in the verification of accounts. Verification methods may differ across the different forms of account in the cloud, as considered further below. Nymity has provided an example structure for evidence and associated scoring mechanism for accountability based on existing documentation that can form some of these types of accounts – but some organisations may want to take a different approach and so this should not be regarded as a standard. The Nymity accountability evidence framework is intended for collecting evidence in a single organisation and for demonstrating accountability that is structured around 13 privacy management processes [10].

There are different levels of verification for accountability, as proposed by Bennett (Bennett, 1995), which correspond to policies (the level at which most seals programmes operate), practices and operations. It is very weak to carry out verification just at the first of these levels – instead, mechanisms should be provided that allow verification across all levels. Most privacy seal programmes just analyse the wording in privacy policies without looking at the other levels, and thus provide verification only at this first level (of policies). The second level relates to internal mechanisms and procedures, and verification can be carried out about this to determine whether the key elements of a privacy management framework are in place within an organisation. Few organisations however currently subject themselves to a verification of practices, and thereby being able to prove whether or not the organisational policies really work and whether privacy is protected in the operational environment. To do this, it seems necessary to involve regular privacy auditing, which may need to be external and independent in some cases.

In terms of the verification process, there are various different options about how this may be achieved. There could for example be a push model in terms of the account being produced by organisations or else a pull model from the regulatory side; the production of accounts could be continuous, periodic or triggered by events such as breaches. In general, there should be spot checking by enforcement agencies (properly resourced and with the appropriate authority) that comprehensive programmes are in place in an organisation to meet the objectives of data protection.

There could in some cases be certification based on verification, to allow organisations to have greater flexibility in meeting their goals.

It is often regarded as underpinning an accountability-based approach that organisations should be allowed greater control over the practical aspects of compliance with data protection obligations in return for an additional obligation to prove that they have put privacy principles into effect (see for example [26]). Hence, that whole approach relies on the accuracy of the demonstration itself. If that is weakened into a mere tick box exercise, weak self-certification and/or connivance with an accountability agent that is not properly checking what the organisation is actually doing, then the overall effect could in some cases be very harmful in terms of privacy protection. As Bennett points out ([27] p. 45), due to resource issues regulators will need to rely upon surrogates, including private sector agents, to be agents of accountability, and it is important within this process that they are able to have a strong influence over the acceptability of different third party accountability mechanisms.

In particular, it is important that the verification is carried out by a trusted body that does not collude with the accountor, and that it is given sufficient resources to carry out the checking, as well as there being enough business incentive (for example, via large fines) that organisations wish to provide appropriate evidence to this body and indeed implement the right mechanisms in the first place. The overall process around verification of an account is summarised within Figure 8. Sanctions might be applied at several points, notably if the organisation does not provide an account in the first place, if it fails to respond adequately to the dialogue with the assessor, or if the assessor is not satisfied in respect to the accounts produced. The process of providing an account could be quite complex, and this is just a generic overview of that process. There could be multiple documents that in the form described here provide an account, but each of which may be viewed as an individual account, and perhaps even have a slightly different process flow. Furthermore, accountability agents could be used to provide verification of accounts, and serve as an intermediary to the ultimate accountees, some of whom impose the sanctions. If, as considered within D:C-2.1, there is a good trust relationship between such an agent and the accountee, then the agent's account is likely to be directly accepted by the other accountees.

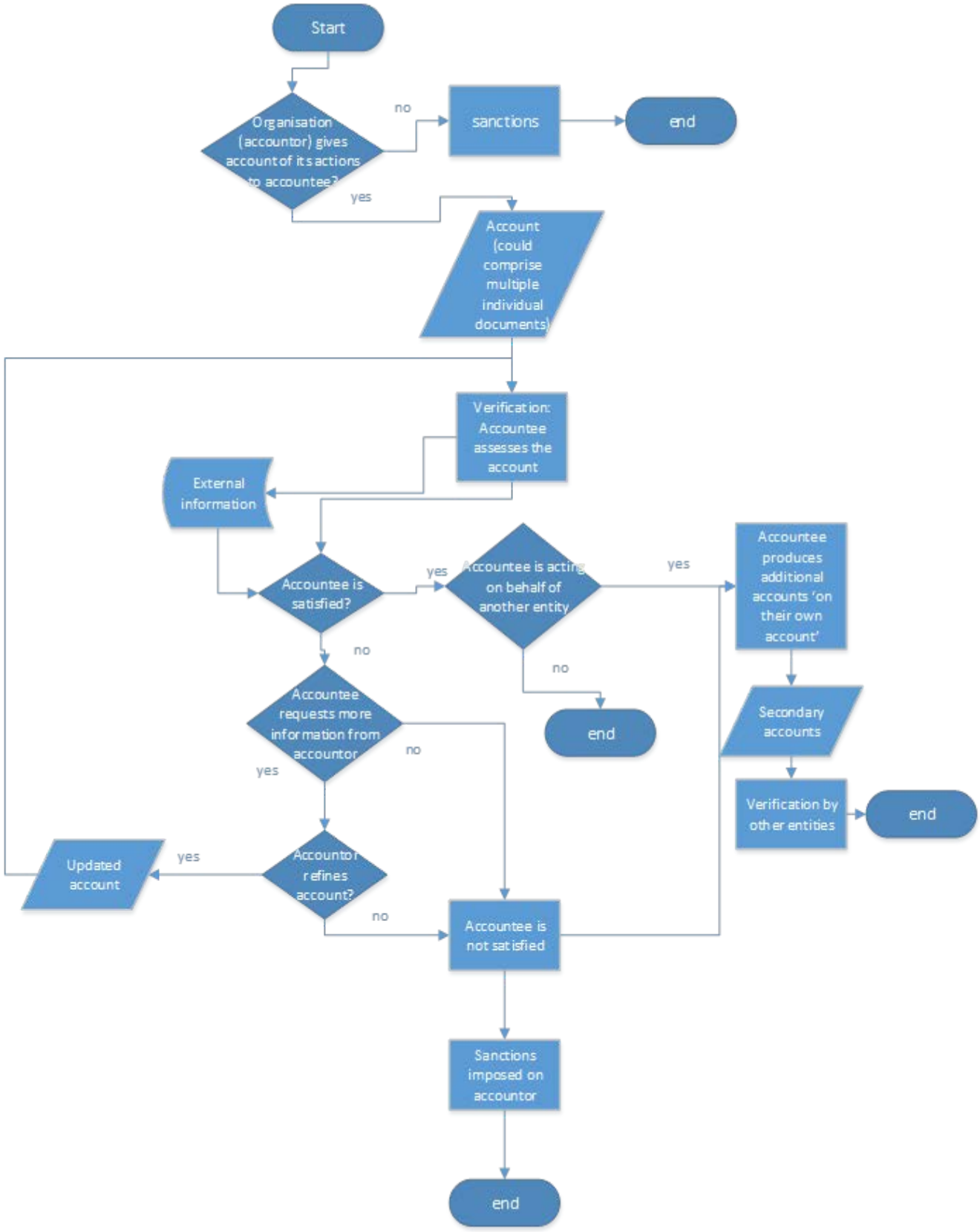

Figure 8: High level view of the provision and verification of an account

3.4.3 Properties of Accounts

In this section we examine what an account should contain from a practical perspective, beginning with the obligations arising from legal and regulatory norms, contractual obligations and the opinions of legal or academic commentators about the content of an account. From these sources, we identify what could or should be the content of an account in more detail than already considered in 3.4.1.

Legal or regulatory norms about the content of the account

There is no formal legal standard on the content of an account. Guidance on the content of the account given by regulatory bodies is typically very high level with little specificity provided by regulators as to what accounts must contain. For example, very little direction is provided in the Data Protection Directive, or even the proposed General Data Protection Regulation on this point.

The Article 29 Working Party ('Article 29 WP') has published its own opinion highlighting the importance of the notion of accountability in the field of personal data protection [28]. In its Opinion 3/2010 on the principle of accountability, the Article 29 Data Protection Working Party highlighted the importance of a concrete proposal for a general accountability principle. Specifically, the Article 29 Working Party found that accountability should focus on two main elements: "(i) the need for a controller to take appropriate and effective measures to implement data protection principles;" and "(ii) the need to demonstrate upon request that appropriate and effective measures have been taken. Thus the controller shall provide evidence of (i) above." ([28] at 28) From a data protection point of view, the account is the method of presenting such evidence and demonstrating such measures. The Article 29 Working Party also explained how the use of accounts will lead to greater enforcement by data protection authorities, and perhaps for our discussion, increased accountability:

> Furthermore, putting the accountability principle into effect will provide useful information to data protection authorities to monitor compliance levels. Indeed, because data controllers will have to be able to demonstrate to the authorities whether and how they have implemented the measures, very relevant compliance related information would be available to authorities. They will then be able to use this information in the context of their enforcement actions. Moreover, if such information is not provided upon request, data protection authorities will have an immediate cause of action against data controllers, independently of the alleged violation of other underlying data protection principles.([28] at 60)

A similar approach is taken in the non-binding 2009 Madrid international privacy standard, which also addresses the need for organisations to provide an account:

> The Responsible person shall: a) Take all the necessary measures to observe the principles and obligations set out in this Document and in the applicable national legislation, and b) Have the necessary internal mechanisms in place for demonstrating such observance both to data subjects and to the supervisory authorities in the exercise of their powers, as established in section 23 (Monitoring).

These documents are important to the process of devising accountability mechanisms in that they indicate how accounts could be used by data protection authorities, for example, to monitor compliance and to demonstrate to the relevant authorities that such measures are in place in the organisation. Therefore they implicitly indicate what types of information an account should contain: evidence of compliance with legal norms and information that demonstrates to authorities that relevant compliance has taken place. Nevertheless, these statements do not give a template of what exactly an account should contain. Rather they are good practice guidance for accountability at quite a high level.

Contracts in the Cloud and practical accountability

Contracts between data controllers and cloud users, and, to a lesser degree, contracts between data controllers and data processors also, do not shed much light on the notion of the account. Contractual obligations essentially take regulatory obligations, which may be at a high level, and translate them

into specific binding obligations between the parties. It is also important to note that contractual obligations are not only based on regulatory obligations. Non-legislative obligations such as industry standards and certifications or even accepted industry norms can be included into agreements, which turn such obligations into legal contractual obligations. And even then, data controllers largely try to further limit their obligations, particularly their liability, in their contracts and/or terms of service. [29]

As between data controllers and data processors, Article 17 of the Data Protection Directive requires data controllers to impose on data processors the same obligations regarding the implementation of security measures as those imposed on data controllers. The relationship between data controllers and data processors will normally be established via the prior conclusion of a contractual agreement (or other legal act).[17]The initial draft of the Proposed Regulation stipulated in Article 26(2) that such a contract or legal act should be obligatory and should require the processor to "make available to the controller and the supervisory authority all information necessary to control compliance with the obligations laid down in this Article", or in other words to provide at least a partial account.

Finally, the one area where one would most expect an account to be provided would be where there has been a security breach, yet, even in negotiated contracts, as opposed to the standard, non-negotiated contracts which currently dominate the cloud computing landscape, "many providers' standard terms did not require reporting of security incidents and so on to users." [30]

It is noteworthy that even where accounts are imposed by law through legislation and contracts, there is mostly little to no express provision as to what the account must specifically include. If accountability is to be built into the Cloud, an important element will be the inclusion of terms in contracts which require a proper account to be given.

It is not possible to draft model contract clauses for this purpose because what is proper in an account will vary substantially depending on the nature of the relationship between the giver and the recipient of the account. It is, however, possible to suggest some overriding principles which might guide the drafting of such clauses:

(a) The recipient of the account should be entitled to appropriate information about how its data will be stored and processed, updated as storage and processing methods change. The level of detail will depend on the nature of the relationship and the data. Thus a consumer user of a "free" cloud service should be content with quite general information, whereas a financial institution will require far more detail.
(b) There should be a suitable mechanism for checking that the actual operations on data match the information given under (a). Mechanisms might range from tools that allow customers to generate their own reports, through independent audit reports, to a right to inspect and audit a provider's systems.
(c) There should be an appropriate mechanism for reporting breaches to those whose interests are engaged, primarily customers, data subjects and regulators. What level of reporting, at what seriousness of breach, and to whom, again will depend on the nature of the relationships.
(d) The account should include explanations of the reasons for any failings, and the measures which will be taken to prevent future failure. The frequency, granularity and addressees of this part of the account are also relationship-dependent.

The content of an account

Since we have no clear stipulation from legal or regulatory sources about the content of the account, we therefore have to turn to statements by academic commentators who have studied or assessed the notion of the account or accountability and analyse what they suggest an account should contain.

[17] Data Protection Directive Article 17.3 *The carrying out of processing by way of a processor must be governed by a contract or legal act binding the processor to the controller and stipulating in particular that: the processor shall act only on instructions from the controller,*

As the oft-cited Charles Raab noted:

> To 'give an account' – *rendre des comptes* – is to tell a story, and there are three levels that can be distinguished. First, on a weak definition, it means the obligation of an organization to report back, to 'give an account of its actions'. Second, on a stronger definition, it means that, plus the implication that the audience can interrogate the account and produce other accounts 'on their own account'. Third, on the strongest definition, it means the previous two plus the implication that sanctions can be brought to bear where there is a general agreement that the organization has 'given a bad account of itself', either (a) through its inactions, or (b) through its own unsatisfactory production of an account. The audience, which may be the public, can thus 'hold the organization to account', and that might have real consequences. [31]

And, as Raab further noted:

> But the account must also, and essentially, include descriptions and explanations of the actions, for two reasons. First, so that we can better understand the organisation's intentions and its understanding, or theory, of its own situation or how it might act in it. Second, because most of a steward's actions are invisible to the principal, and therefore have to be re-presented, through stories or accounts, explanations, and justifications. [31]

Importantly, especially for an organisation to be accountable, an account is not provided only when something has gone wrong, but rather can be presented at any time upon request. As one commentator opined:

> Accountability does not wait for a system failure; rather, it requires that organizations be prepared to demonstrate upon request by the proper authorities that it is securing and protecting data in accordance with the essential elements. [32]

3.4.4 Mapping Different Kinds of Account to Functional Elements of Accountability

Provision of an account could be *proactive*, in the sense that the choice of accountability mechanisms and tools needs to be justified to external parties, and this could happen before any processing takes place (perhaps as part of a third party assurance review), when processing is particularly risky (e.g. before such processing, with documentation generated via Data Protection Impact Assessments), or using ongoing certification to provide flexibility (for example, as is the case with Binding Corporate Rules). There may also be a reactive, *retrospective* element, for example when a data protection breach has occurred, or when provision of accounts is triggered by a spot check by a regulator.

In terms of the organisational lifecycle described above in Section 3.1, provision of an account may take place in different phases, as shown in Figure 9. Example accounts corresponding to these four stages are shown in Table 8: Mapping of different kinds of account to functional elements.

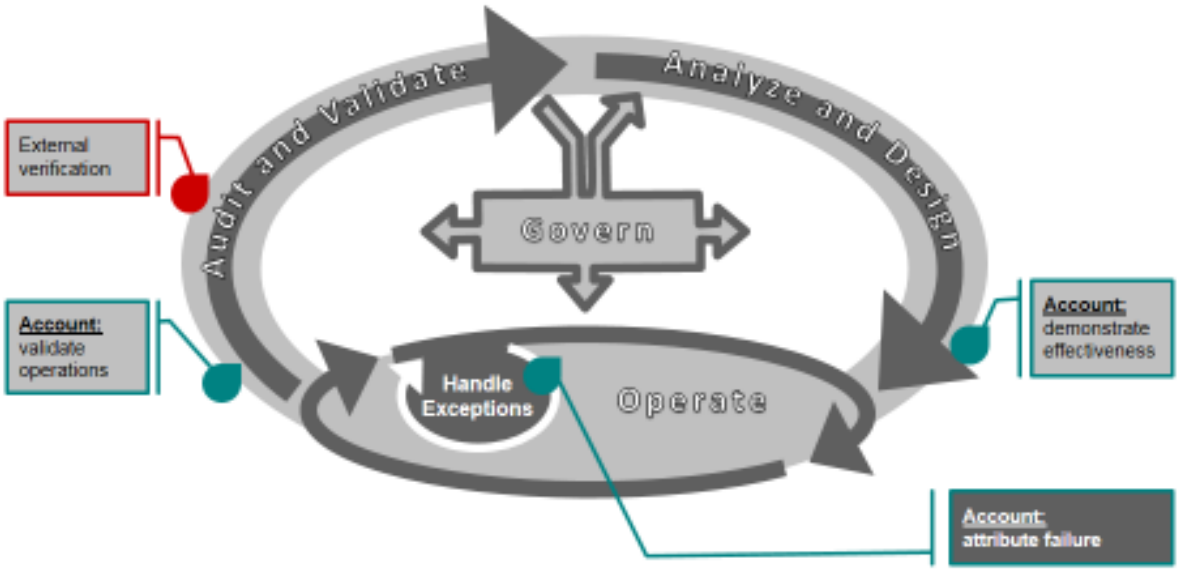

Figure 9: Functional elements of organisational account provision

Functional Element	Types of Account
Demonstrate effectiveness	Data Protection Impact Assessment Notice to supervisory authorities (before processing) Documentation obtained, created and maintained by DC & DP
Validate operations (organisation)	Contractual compliance verification
Attribute failure (exception cycle)	Notification of data breach to data subjects Notification of data breach to supervisory authorities Notification from cloud provider to other cloud provider/organisation
Output of third party checking (to be shared)	Certification & seals, e.g. OCF level 3 Audit reports Verification by third party accountability agent

Table 8: Mapping of different kinds of account to functional elements

In our further analysis we will not focus on records used for internal use (for example, risk reduction and self-improvement) within the security lifecycle, but instead those for external use, and in particular some cases of evidence provided for compliance and data breach notifications.

Furthermore, an interesting distinction can be made between what may be regarded as *static* accounts, as opposed to *dynamic* accounts. The former do not vary over time, whereas the latter take into account parameters that may change over time. For instance, an example of a dynamic account would be a CSA Open Certification Framework (OCF) level 3 account, which is an example of a dynamic certification. Indeed, it could be argued that yearly or monthly audits are irrelevant in an environment that changes completely on a daily or hourly basis, as is often the case with cloud computing. Continuous compliance monitoring is essential to securely delivering cloud services and ensuring compliance. Cloud services are inherently dynamic, because the dynamic provisioning and de-provisioning of resources is a key part of the Cloud value proposition and business model. Hence,

automation for operations and asset management are essential in this dynamic environment and verification of compliance with policy and legislation – such as the EU Data Protection Directive, GLBA, HIPAA, and Export compliance controls like ITAR – requires continuously running automation.

3.4.5 Relationship with Certification

In addition to producing an account during the operational phase, the accountor can deliver a proactive report to demonstrate the effectiveness of the provided service. The foremost evidence included in such a proactive account are the policy or the binding corporate rules (BCR). For example the Data Protection Authority can verify whether the actual policy rules, terms and conditions are governed by law.

In terms of the role of certifications, from a high-level perspective the certification of a service, product or process involves the verification of the following steps:

1) Define the scope of the certification.
2) Conduct a risk analysis to identify potential security, governance or general compliance risks.
3) Mitigate the risks by selecting and implementing a set of controls (some risks can also be transferred to another party or accepted if they are minor).
4) Monitor and update periodically the whole process, potentially starting again from step 1.

These steps can be conducted internally by an organisation in the form of a self-certification. However a higher level of assurance is always obtained when the assessment of these steps is conducted by an independent third party (auditors).

Well-known certification schemes that relate to security and privacy include:

- ISO 27001, which certifies information system management practices.
- PCI DSS, which focuses on the secure processing of bankcard data.
- Service Organisation Control Reports (SOC 1, 2 or 3), which is also about information system management.
- CSA CAIQ, which is a self-assessment regarding best practice governance, risk and compliance.
- CSA STAR Certification, which applies ISO 27001 to a set of cloud specific controls.
- BCR (Binding Corporate Rules): A set of rules that apply to intra-organizational data transfer, in order o assure compliance with EU data prot
- EuroPriSe: Certifies compliance with data EU protection rules for products and services.

Certification can be used to support the “account” as defined in A4Cloud in two ways.

Certification as an account

First, when an organisation obtains a certification to demonstrate compliance with a set of rules, whether they relate to security, privacy or governance in general, they are fully applying the notion of a “proactive account” (see 3.4.1).

With the notable exception of CSA CAIQ, which is a form of self-certification, whereby all findings are published in a public repository, the certification schemes listed above all require the evaluation of the different steps of the certification to be conducted by an independent third party (i.e. an auditor or a public authority). The “account” of all 4 steps of the process is therefore not always available to the customer of the cloud service, but only to a trusted party. BCRs mandate that the result of the evaluation is made available to “data subjects”, but most other certifications scheme have no such requirement. In fact, there are also arguments against providing the same information to auditors and customers (or data subjects):

- The documented account contains descriptions of security measures and processes that could provide information to adversaries.
- The documented account contains intellectual property or other confidential business data.

The result is that in many cases, certified companies will merely highlight that they hold a certification, and will not disclose details of the process. The “proactive account” is provided by “proxy”: the account is provided to the auditor and the customer is expected to trust the auditor.

While this principle is defensible form the reason stated above, it contradicts the principle of transparency that is central to accountability. There should be therefore a middle ground that distinguishes a “detailed proactive account”, which is only provided to the auditors, and a “public proactive account”, which is provided to all interested stakeholders (customers or data subjects). The example of the CSA CAIQ Registry[18], which currently has reached 100 entries, shows that companies are willing to provide information detailing on a high-level how they implement certain controls addressing risk, compliance and governance issues in cloud services. There is therefore room for the notion of a “public proactive account” in certified cloud services.

Certification of the account

Second, in order to build the account, a certain number of processes need to be in place. For example, incident reporting processes need to be defined tested and implemented, evidence collection procedures need to be verified. Like any other organizational and technical processes, these elements are by nature certifiable, provided that a reference organisational standard exists. These elements would have a natural place within an “accountability management standard” as we suggested in 3.2.2.

3.4.6 Summary of Core Properties

From the analysis above, we can identify what an account should include. An account can perhaps best be defined, though simply, as “a report or description of an event.” This means that an account should have a story or narrative that can be easily understood. This account or report can be presented at any time, not just when there has been a system failure. The report or description should sometimes include reasons or explanations, for example, if the event should not have occurred. It should sometimes also explain consequences, for example, what action will be taken to remedy a situation or what action will be taken in the future. It may also include justifications for actions taken or for omissions.

In light of the foregoing, an account, when required and/or provided, usually consists of the accountable actor providing a report or description of an event or process. The account should generally include the answers to what are traditionally referred to as the ‘reporters’ questions’, i.e. who, what, where, when, why and how. Often, an account will also include the measures being taken to remedy a breach or failure. Still, the form and content of the account are contextually dependent and may be specifically dictated under the specific circumstances. Forms of the account may include Data Protection Impact Assessments, notifications to supervisory authorities, notifications to data subjects, contractual compliance verifications, audit reports, and even certifications and seals obtained by data controllers and/or data processors from third party certification agencies such as Cloud Security Alliance.

Applying these principles in practice perhaps best demonstrates the notion of the account and what would be encompassed in an actual account. So next, examples of giving an account are considered further.

3.4.7 Key Examples: Accounts Relating to Compliance

While an account is usually expected and more useful in the event of security breach or policy violation, an accountor, i.e. the organisation acting as a data steward, may report on events or operations that have taken place in accordance with the pre-defined rules. For example, the process for providing information on the use of third parties in the cloud service chain is not adequately addressed in current practices, resulting in the bulk of end users not being aware of the complete cloud service delivery chain and their rights on data handling processes. In DB3.2 [33], several

[18] https://cloudsecurityalliance.org/star/?r=4615#_registry

different obligations on informing different actors (Data Subjects or DPAs) about data processing practices have been enumerated (O1-O4, O13, O16).

Therefore, an account relating to compliance should demonstrate that the accountor fulfils the expected requirements regarding data processing practices usually defined through the obligations expressed in policies or in law. Such a report should generally describe the event in question and include the following information:

- involved actor(s): the account should list all relevant actors who were involved in the event;
- list of actions with **time** and **location** information: the report should describe all relevant actions taken with respect to the event to be reported. Such description should include all information on time and location.
- justification of the event: the account should describe the reason of the event. This usually refers to the identification of the policy rule or obligation for which the accountor should comply with,
- contact details of the person or group who is responsible for the event in case further information is needed or an unexpected problem occurs.

The description of the event should further be completed with some evidences in order to achieve a certain level of confidence, indeed, as also stated in [10], it is more difficult to fully demonstrate compliance than to report on a security breach. Therefore an account on compliance should regroup all appropriate evidence.

In [10], Nymity divides their proposed *privacy compliance attestation methodology*, into two main steps: identify the rules that require appropriate evidences and further demonstrate accountability. We propose to follow the same approach and to enrich this methodology by providing examples of forms of account regrouping a list of potential evidences with respect to specific obligations derived from D23.2 [33].

Account of secure data deletion

An accountability policy includes obligations on data retention which includes information on data storage period. According to Obligation 7 in D23.2 **[33]**, "the *data controller must make sure that all personal data are deleted (…) after the data collection purpose has been fulfilled*". Secure deletion of data is not straightforward and cannot be 100% guaranteed. Existing solutions either remove the link of the data to be deleted or overwrites the content with random data. Some other solutions use cryptography and for example propose to encrypt data while storing it and discard the decryption key for deletion. It is therefore important to describe how data is deleted. Therefore, an account on secure data deletion should include the following information:

- description of the deletion method (unlinking, overwrirting, etc.);
- log traces on delete queries including information on time and location both from primary storage and backup servers;

In addition to these evidences, the account may also include the contact details of the person responsible for this action in case a further problem occur.

Account of correct data storage

Obligation 8 in D23.2 states that "*The controller must (..) ensure that appropriate security and privacy preservation measures have been implemented throughout the service delivery cha*in". Since the main service supplied by a cloud provider is data outsourcing, the cloud provider should ensure data availability and integrity. In order to prove the correct storage of data, the cloud provider can provide the following evidence:

- information about data handling practices;
- log traces with respect to the handling of the particular data;
- cryptographic proofs about the storage and integrity of the data.

Account of data location

Cloud adoption raises serious privacy concerns with respect to data residency. An accountability policy should express rules about the location of the data and the accountor should provide some evidences on the location of personal data either upon receiving a request or automatically whenever data is transferred. An account on compliance with respect to data location rules can regroup the following evidences:

- BCR approval: the number of multinational companies adopting Safe Harbor,Binding Corporate Rules [34] which define the rules with respect to international data transfer is increasing. Therefore, a BCR certification can be considered as essential evidence for an account on compliance.
- information on the physical location of the servers: the accountor can provide such information with a third party audit report for example;
- log traces: data transfer logs can be obtained with a monitoring tool like the A4Cloud's DTMT.

3.4.8 Key Examples: Handling a Data Breach

The most common situation where an account is required and provided is a data breach scenario. As part of an accountability policy, legal and normative obligations, with respect to the information of associated parties in case of abnormal behaviour, such as data breach or policy violation, should be expressed in the form of rules. Currently, the requirement for information about such events is not explicitly derived from the regulatory framework, resulting in a lack of proper information about data leakage and violations happening in the cloud service supply chain.

In D23.2 [33], we have presented examples for the expression of the account on information about the resulting notification for an abnormal event. A4Cloud introduces the obligation for generating notifications on abnormal events (Obligation O18). This type of account should be verified through the following information:

- The actor sending the notification
- Type of incident, detailing also which PII was affected
- The actor in which the incident was raised
- Evidence in the form of logs traces, explaining the incident history
- Timestamp of generating the notification
- Contact details of the actor responsible to answer notification response
- Potentially the contact details for the responsible DPA

In this section several different cases are illustrated in which an account is given in the event of a data breach. These examples also differ from each other with respect to the recipient of the account. The first example is where an account about unauthorized data access is provided to a data subject. Next, an example is provided in relation to a regulatory investigative process. Finally, we consider some examples of data handling within a service provision chain.

Account to data subject

This scenario below gives an example of how a data breach could be reported to a data subject. It hypothesizes a breach of a cloud provider where data has been accessed and downloaded without authorisation. This breach notification is not required by law. Neither the Data Protection Directive, nor its implementation in national legislation in Member States of the EEA requires a notice of a security breach to be provided to either the Data Protection Authority or the data subjects. Regardless, we are giving an example of a scenario in which an accountable Cloud provider, the provider here desires to provide an account to the user and the Data Protection Authority.

How the breach notice should be communicated

The question of how the breach should be communicated to data subjects is entirely at the discretion of the cloud provider since there is no legal obligation to provide an account. That said, it is most likely that the cloud provider would send the account by email to data subjects, in the first instance at least,

but depending on the severity of the breach, notice could and quite possibly should also be sent by mail to ensure proper notice and receipt.

What should be included in the breach notice

As noted above in Section 3.4.3, there is no legal or regulatory template for such a communication but the account here should encompass answers to the fullest extent possible of the reporters' questions, i.e. who, what, when, where, how and why, as well as measures being taken to prevent such breaches in the future.

More specifically, the cloud provider will want to do the following in its communication:

1. explain who committed the breach, if known, or that further investigation is being undertaken to ascertain who committed the breach;
2. what the breach consisted of and the extent of the information that might have been accessed, i.e. health information, financial information, etc.;
3. when the breach occurred and was discovered;
4. where the breach occurred;
5. how and why the breach occurred, if known, what security measures in place, whether those security measures were properly working at the time of the breach, and how the breach generally circumvented such measures;
6. what measures were taken to ascertain the extent of the breach;
7. what measures are being taken to prevent such breaches in the future;
8. contact information for a department or person to respond to any further enquiries regarding the breach; and
9. perhaps a link to a web page where further information, if any, will be disseminated regarding the breach and any further investigation.

Thus, hypothetically and a its basic form, an account by a Cloud customer and/or Cloud provider to Cloud subjects after a data breach may look like the letter or email shown in Figure 10.

Dear Data Subject:

We write to you regarding a recent unfortunate incident involving an unauthorized access to our servers in which your personal data may have been accessed.

On February 1, 2015, we believe that an outside intruder circumvented our security measures and was able to access the personal information of some of our users. We realized the access almost immediately and were able to minimize the access. The full extent of the breach is not known, or whether your information was accessed and/or otherwise obtained by the intruder. What we do know at this time is that our security measures were operating properly, but the intruder was able to circumvent such measures through illegal means. We have since closed the means through which the access occurred and are re-examining all of our security measures to ensure the fullest protection available moving forward. We are also continuing to investigate the situation and further exploring the extent of the information which may have been accessed.

We will release further pertinent information regarding our investigation on our website at www.cloudprovider.com/01022014breach, so we invite you to regularly check that page for any updates regarding this situation. Should you desire to contact us for further information, please do so at [email] or [telephone number], where we will be standing by to respond to any enquiries as quickly as possible.

We thank you for your continued patronage and your confidence in us preventing these unfortunate incidents in the future.

Sincerely,

Cloud Provider

Figure 10: Example data breach account (notification to end user)

To the data subject, the account will be general and use simple and non-technical language, without much of the technical information that would otherwise be available to the cloud provider. The cloud provider may decide to include more technical information on its website or upon request by the data subject, but the overriding objective to the end user should receive a clear explanation of the account.

What should not be included in the breach notice

The notice to the data subject is in contrast to the account of the same breach to the Data Protection Authority, discussed in the section about investigations above, Section [xxx], where the account should contain more technical information, for example, the extent of the breach, a more technical overview of the breach, and the number of persons impacted by the breach. In addition, the account to the Data Protection Authority would also include relevant evidence regarding the breach, i.e. any applicable logs, audit trails, system maintenance records, and any other technical evidence regarding the proper operation of the cloud provider's security measures and the extent of the breach. Providing such information to a data subject, however, would be counterproductive since such detail could confuse them about the nature and extent of any breach. Therefore excessive technical detail or evidence should not be included in the initial breach notice to the data subject.

Updating and providing additional information after the breach notice

As more information is obtained by the Cloud provider and/or business, such information could continue to be provided through updated accounts to the data subject. An example of this is the handling of the data breach by U.S. company Target referenced above (in section 3.4.2) provides an example of such accountability in practice. Target established a webpage containing rather detailed information after its credit card processing systems were compromised. (https://corporate.target.com/about/shopping-experience/payment-card-issue-FAQ last accessed on 29 January 2015). It continued to update that page providing its customers with information about the

extent of the breach, measures that were being taken to prevent such breaches in the future, and other precautions end users should take to avoid damages and/or further damages. The account and updated accounts by Target provide an excellent template for companies facing similar data breaches and/or circumstances in the future.

Generating Accounts during Cloud Investigations by European Data Protection Authorities

In this subsection, we analyse how various accounts are produced during a specific regulatory process, namely, when European data protection authorities ('EU DPAs') exercise their regulatory power of investigation in the context of the cloud. EU DPAs are the statutory independent public regulatory bodies which have various functions including applying and enforcing data protection laws in European member states. Investigations refer to the one of the enforcement powers of EU DPAs, namely, their power to investigate 'data controllers', such as companies which offer cloud computing services or technologies ('Cloud Providers'), in specific circumstances (e.g. when an individual complains). This analysis is generated from the qualitative socio-legal research as part of D4 where we interviewed fifteen respondents including EU DPAs which have investigated Cloud Providers, and Cloud Providers which have been investigated by EU DPAs.

Our data analysis suggests that multiple accounts are generated by various actors during the different stages of an investigation of a Cloud Provider by an EU DPA ('Cloud Investigation'). Cloud Investigation can be approached as a three-stage process which consists of the pre-investigative, investigative and post-investigative stage. The pre-investigative stage includes a plethora of circumstances, practices, and routines which lead to the investigative stage (e.g. email exchanges and conference calls between the EU DPA and the Cloud Provider). The investigative stage starts when the EU DPA initiates the Cloud Investigation (e.g. by sending a 'letter of intention to audit' to the Cloud Provider) and ends when the investigation report is finalised and/or published (depending on whether the report is published). The post-investigative stage refers to the stage following the publication (whether internal or external) of the investigation report.

During the pre-investigative stage, multiple accounts of compliance can be generated by different actors depending on the investigation in question. For example, an EU DPA that is unfamiliar with the data processing operations and business model of a Cloud Provider may engage in substantial discussions with various teams of the Cloud Provider (e.g. management, engineering, and legal) to know more about the entity it will regulate later on. Such requests for information also generate multiple accounts from the Cloud Provider such as account of compliance through internal and external policies. Here the types of account take various form such as exchanging relevant information through conversation or email or documents.

During the investigative stage, other accounts of compliance are generated by various actors. As with the pre-investigative stage, such accounts and the actors involved in generating these accounts are context-dependent. For example, subject to several factors such as financial pressures faced by EU DPAs, scope and aim of Cloud Investigations, different forms of accounts may be sought such as an account of how different technical functions operate in practice Here the Cloud Provider often has to provide the EU DPA either with access to the algorithmic codes which implement these technical functions so that the EU DPA can test whether the algorithmic codes operate in the manner set out by the Cloud Provider in its policies (e.g. a cookie is deleted within a period of time specified in the cookie policy or the encryption methods used by the Cloud Provider operates in the manner specified in its privacy policy). Technical testing here often include other actors such as sub-contractors employed by EU DPAs that face financial constraints. Here, the account of compliance generated by the sub-contractor when s/he tests the relevant data processing operation of the Cloud Provider has to be compiled with other accounts of compliance generated by other employees of the EU DPAs (e.g. through analysis of privacy policies etc).

Other compliance accounts can also be sought and produced such as accounts of compliance with the relevant data protection laws by providing the EU DPA with access to specific computer terminals when it inspects the premises of the Cloud Provider. Such accounts emanate from various sources and have to be managed at the Cloud Provider level before being passed on to the EU DPA for its review to ensure that these multiple accounts do not provide conflicting views of the compliance of the

Cloud Provider with existing data protection laws. Here the generated accounts are questioned by the EU DPAs and can often be clarified by the Cloud Provider in cases of confusion.

These multiple accounts are examined by the EU DPA at the end of the investigation to determine to what extent the Cloud Provider complies with the relevant data protection laws. Here, there is evidently a very close link between the accounts produced during the Cloud Investigation and the outcome of the Cloud Investigation (e.g. the recommendations of the EU DPA to bring the operations of the Cloud Provider in line with the relevant data protection laws). This does not mean that accounts of compliance cannot be constructed in specific ways so that a particular version of compliance is generated especially when the report produced at the end of the Cloud Investigation is published. We have explored this point further in the deliverable D: D-4.11.

Finally at the post-investigative stage, other accounts of compliance are sought and generated by specific actors. For example, the EU DPA seeks account of how the Cloud Provider is implementing its recommendations. Additionally, the Cloud Provider can also seek advice from the EU DPA about the compliance of its proposed future innovations with existing data protection laws. Here accounts of compliance are generated through informal interactions such as face-to-face meetings.

Accounts within the service provision chain

A number of incident scenarios are being studied by work package D4. Some incident categories can be identified by the A4Cloud detective tools, other categories would fall outside the scope of the description of work, but standard techniques and tools can be used to detect them. For instance, non-compliance with respect to data location constraints can be detected by the Data Transfer Monitoring Tool (considered within section 5), but a machine infected with malware allowing a malicious external agent to have unauthorized access, would not be the main focus of our tools. Once a potential data breach has been notified to the data controller (via e.g. the A-PPL Engine), evidence needs to be analysed to confirm the breach. This later step can be achieved with the help of the techniques devised in the Work Package C-8 and the Audit Agent System (AAS), for instance, which will help to identify where failures occurred in the cloud service provisioning chain. Finally, the data subject notification can be handled as described above, possibly with the support of the Incident Response Tool, currently being designed in the context of the D-4 Work Package.

Remedial actions can be imposed by the Data Protection Authorities upon data subject complaint fillings. For a detailed legal analysis of remediation and redress mechanisms, please see MS:D4.1 [35]. The search of an arrangement to remediate damages caused by a breach can be facilitated by the Remediation and Redress Tool, also under consideration in D-4.

For a more precise incident please consider the following scenario from D-4:

"Misconfiguration of services and failing to patch software quickly can lead to severe security problems such as being vulnerable to exploits and violating security requirements. Recent SSL vulnerabilities such as POODLE, BEAST and Heartbleed are prime examples for the need to patch as soon as fixes become available. However, patching may not be enough in some cases. For instance, to mitigate the Heartbleed vulnerability, certificates need to be replaced, old certificates revoked and private keys changed. Besides that, problems can arise from service misconfiguration. The recently discovered POODLE vulnerability is closely linked to obsolete protocols being allowed (which is an SSL configuration problem). Also, in cases where strong cryptography is required, specific SSL configuration is required (protocol versions, available cipher suite, cipher order, algorithms, key length, certificate status…)."

An attacker can gain access to personal data by exploiting this kind of vulnerability. In some cloud supply chains, it can be complex to identify whose responsibility is to apply the necessary patches and updates to the software. This will depend on the service model and on the contractual agreements in place. The A4Cloud approach and tools help to clarify these situations: from the detection towards the remediation phase.

Another example from D4 of a data breach has to do with a data holder's right to have access to a set of data (right to know). In such a case, individuals are granted access to specific data, but in case of a

contextual change in the individual's behaviour over such access rights (such as a large number of access requests in a short period of time), this may imply a potential violation (need to know property). An example like this is common in today's security systems, which adopt intrusion detection mechanisms or perform log and error analysis to monitor malicious intruders and discover misbehaviours, which can result from e.g. the loss of credentials from the data subject's side. If an intrusion is detected, then the responsible system administrator is informed of the timestamp of the event and the details of intrusion attempt (e.g. who, what, reason for alarm, etc.). The relevant Data Subject will occasionally be informed, but this is subject to the responsible behaviour of the service provider, while the notification of the breach will be mainly handled in a manual way (such as by mail). As happens with the previous example, in this example, the handling of the breach involves multiple notification recipients, each of whom should receive a different level of informed actions to respond to. In case of the Data Subject, as mentioned earlier, this actor will occasionally be notified that their credentials have been compromised, in a way that could enable hackers to enter the system with their digital identity. On the other hand, the DPA may also be informed in case of these events, since this actor has to be told the details of this breach if the damage is severe, the affected personal data and the respective data subjects and the actions undertaken to mitigate the risks from the exposure of the breach.

4 Implementing Accountability Across the Supply Chain

4.1 Challenges in Implementing Accountability across the Supply Chain

Cloud computing describes a model for enabling ubiquitous, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [36]. Its key characteristics are on-demand self-service, broad network access, resource pooling, rapid elasticity, multi-tenancy (of users and/or applications) and measured service. Cloud computing can be provided via different service models, such as *Software-as-a-Service* (SaaS), *Platform-as-a-Service* (PaaS) and *Infrastructure-as-a-Service* (IaaS), as well as deployment models such as *private, hybrid* and *public* cloud [36].

Looking at the resulting cloud environment from an accountability perspective, it consists of four major elements. The first is the cloud *infrastructure*. Not to be confused with the Infrastructure-as-a-Service model, infrastructure encompasses everything that makes possible the provision of computing resources under the cloud computing paradigm described above, such as networks, data centres and the software systems necessary for operating them (e.g. virtualisation systems, management systems, etc). *Data* is all information that enters the cloud to be processed and/or stored. *Applications* describe the software made possible by the cloud infrastructure that manipulates the data entering the cloud in useful ways. These are the software services that make the cloud useful for its users, whether they are platform services and middleware used to support other applications or end-user-facing applications themselves. Finally, *people* describe the humans who support and operate the infrastructure, develop and manage the applications and handle or are responsible for the management of the data lifecycle. People are members of specific organisations with clearly-defined roles and should not be confused with the cloud "actors" identified in Section 2.1

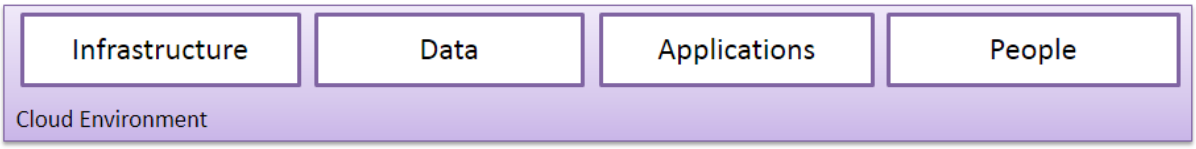

Figure 11: Elements of the cloud environment.

A major benefit of cloud computing is that it enables the flexible composition of powerful applications by chaining together functions provided by different cloud services and providers. For example, end-user-facing cloud applications may be composed from different service components packaged as Software-as-a-Service offerings, themselves utilising cloud resources provided by different Infrastructure-as-a-Service providers. Furthermore, the use of standard interfaces and technologies means that cloud services[19] along the service provisioning chain may be substituted with others of similar specification without radically altering the way the application is composed.

An implication of this model however is that separate, independent entities assume control, ownership and responsibility for different parts of the service provision chain, the latter constituting separate domains of control. This is illustrated in Figure 12 below, which presents a typical cloud service supply chain. A cloud service provider is operating a datacentre to provide a public IaaS cloud offering. Numerous tenants utilise the cloud resources made available to provide applications to the general public in the SaaS model. Each tenant's virtual environment is isolated from all others' by means of the IaaS provider's virtualisation and management infrastructure. Finally, customers access tenant applications over the public Internet to support various business functions.

Clearly, different parts of this supply chain belong to different control domains. The IaaS cloud operator has administrative control (and responsibility) over the IaaS support infrastructure. This is indicated in the figure by the area marked by the red dotted line. Employees of the cloud operator may need to access privileged administrative interfaces to manage parts of the IaaS support infrastructure (for example to apply security patches). Naturally, access to those privileged interfaces will only be

[19] This is especially common for services operating at the IaaS layer, as the functions supported by different vendors at this layer are generally uniform.

given to a select, vetted and authorised subset of the IaaS operator's staff and will never extend to outside its area of control due to their potential for catastrophic misuse.

Similarly, each of the IaaS tenants only controls the virtual environment made available to them, such as the green or blue areas in the diagram. IaaS tenants have privileged access to configuration and management controls of their SaaS applications which they will not make available to entities outside their own domains for security, liability or business confidentiality reasons.

Finally, an application user may be responsible for the handling of data processed by tenant applications as part of some business function. Such a user, especially if they are processing regulated data, may configure controls (such as at-rest encryption) to prevent upstream providers (e.g. the SaaS or IaaS provider in this example) from having non-authorised access to the raw data.

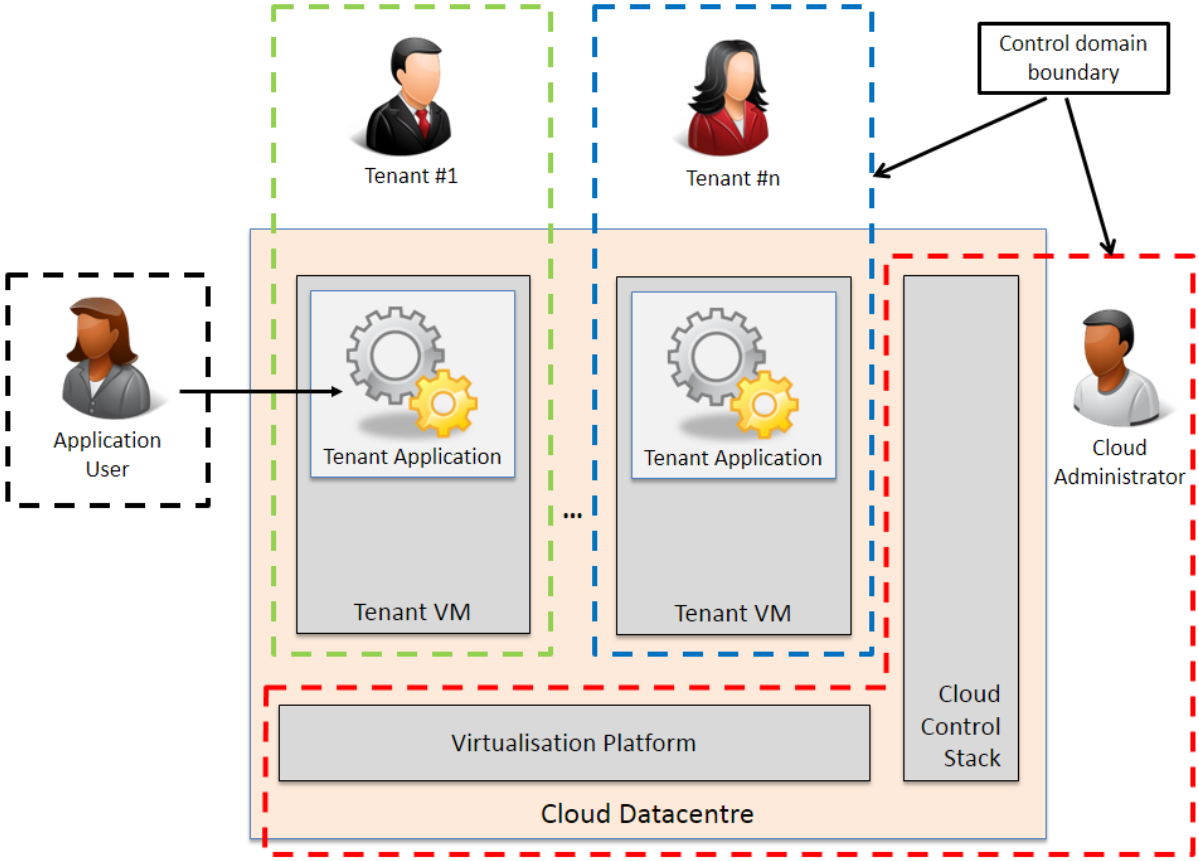

Figure 12: Separate domains of control in cloud service provisioning chains.

Based on the fundamental accountability practices identified in the A4Cloud conceptual model of accountability and discussed earlier in the text, being accountable largely means for an organisation to implement the controls appropriate for the service offered, demonstrate that obligations stemming from policies and regulations are met, and handle exceptions appropriately, remedying failures when applicable.

Even in the relatively simple example of a cloud service provisioning chain illustrated earlier, the challenges involved with achieving accountability end-to-end, across the entire chain, are evident. Since every part of the chain is separated by technical and organisational boundaries between domains, it is hard for any actor to establish whether the processes and operations executed beyond its own domain are according to the agreed rules and obligations. Thus, even if a number of actors have individually implemented accountability-supporting mechanisms inside their own domains (e.g. by implementing the accountability governance process described in the previous section) there are no obvious means for accountability to be extended beyond the various domain boundaries to cover the entire supply chain.

The A4Cloud Cloud Accountability Reference Architecture (RA) was developed to provide a method to tackle these challenges by designing mechanisms to support accountability both within an organisation and across cloud service provision chains. The accountability process described in Chapter 3 focuses on the former task while the rest of this chapter addresses the latter. The rest of the chapter is structured as follows: In the next section, the types of information artifacts that need to flow across the supply chain to support accountability are identified. Next, a high-level view of the service-oriented approach for accountability in the cloud promoted by the RA is presented. This is followed by separate sections looking into each of the various types of accountability support services in detail, before discussing the cloud adoption patterns that emerge for this approach.

4.2 Flow of Accountability Information

As discussed in the previous section, control domains may be formed because of any combination of architectural, technical, organisational or economic reasons. A pragmatic approach to extending accountability beyond domain boundaries must reflect the way clouds and cloud services are architected and operated, and thus focus on enabling the various domains to exchange the information necessary to establish, evaluate and exercise accountability while maintaining their structural separation.

At the highest level, the exchange of accountability-supporting information between two accountable[20] actors[21] in the cloud supply chain may be viewed as supporting one of two purposes: the communication of the service consumer's *objectives* to the service provider as they pertain to the handling of data as part of the service provided and, in response, the provision of *assurance* that those objectives are appropriately met by the provider to the consumer. Figure 13 below illustrates this concept.

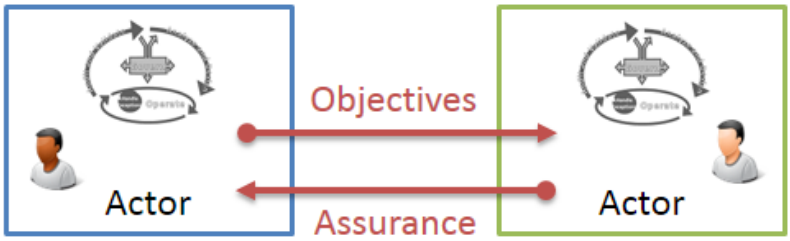

Figure 13: Exchange of accountability-supporting information between two actors.

The communication of objectives and provision of assurance encompass a number of different processes which result in the generation of various types of accountability-related information, called *accountability artifacts*. It is the exchange of these accountability artifacts between actors at various phases of the service lifecycle that facilitates the establishment (and continuous evaluation) of accountability. Figure 14 illustrates how objectives are gradually decomposed into accountability artifacts, and the latter recomposed to provide "the account" and ultimately assurance. We will now look into this process in more detail.

Objectives describe the high-level goals an organisation wishes to achieve through the use of an IT service. These objectives express the business (i.e. functional) needs of the organisation as a prospective cloud customer and may also contain elicited preferences of the cloud subjects it represents. Before a particular service is procured from the cloud, the prospective cloud customer needs to refine and reformulate these objectives so that they clearly capture its privacy and security requirements for the service. These requirements will encompass any requirements stemming from law and regulation, such as limitations on how personal data are collected and handled. The formulation and compilation of these requirements is part of the "analyse & design" phase of the accountability process lifecycle described in Chapter 3, and comprises the necessary first step towards defining the scope for accountability in the sought-after service relationship. These

[20] An organisation or actor defined as "accountable" hereafter is one that has implemented the accountability process described in Chapter 3.
[21] For the list of actors and roles identified in the RA, see Section 2.1.

requirements are (implicitly or explicitly) communicated to the selected cloud provider[22] as part of the service procurement process.

The cloud provider has requirements of its own, stemming from how its business is run and the constraints it has itself from law and regulation. These requirements inform the provider's service specification, which may be published in various forms ranging from documents outlining "terms & conditions" to machine-readable service description documents. In an accountability-based approach the process of publishing a cloud provider's service specification takes a more prominent role. The cloud provider is expected to *advertise* the capabilities of the service it offers with regards to handling of data, including providing information on what security and privacy controls it offers, what mechanisms it has deployed internally to preserve the security and privacy characteristics of the data as well as any certifications or other documents supporting these assertions. The advertisement of these **capabilities** comprises the first accountability artifact to be produced, which enables the service procurement phase between a customer and a provider of cloud services.

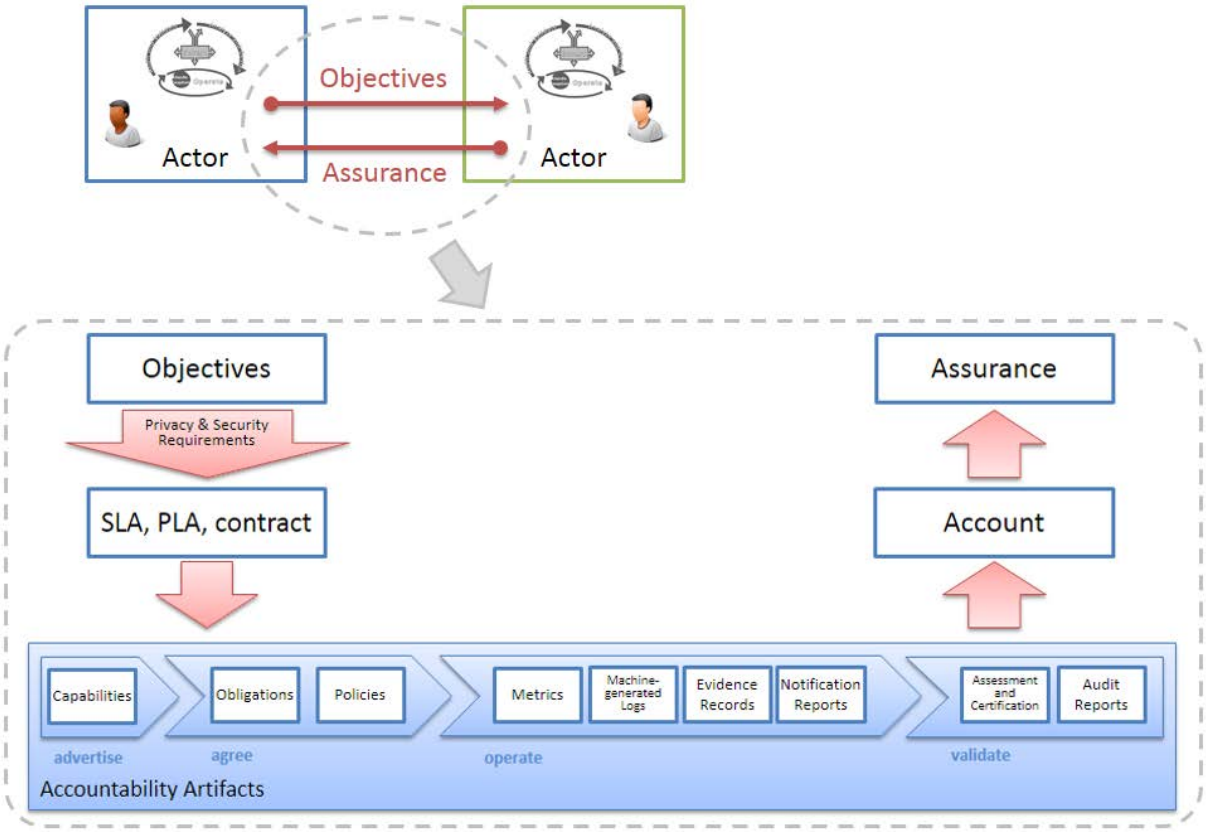

Figure 14: Exchange of accountability artifacts.

Typically, if the procurement process deems that the cloud customer's requirements are compatible to the cloud provider's service description (capabilities), a contract is formed between the two parties describing the service agreement and responsibilities of each party to the other. Service and Privacy Level Agreements (SLAs and PLAs respectively) are particular forms of such contracts or may be produced to supplement a general contract to more precisely define particular aspects of the service. In an accountability-based approach this step is particularly important because it is where the cloud

[22] To avoid unnecessarily complicating this example, we describe a process between two parties, a single cloud customer and a single cloud provider, which results in a straightforward agreement. Obviously, while searching the market for the most appropriate service offering, the cloud customer may engage with multiple providers in parallel, expressing its requirements to each of them and evaluating all responses before selecting one. Additionally, it may engage in negotiation of terms, which will imply a number of out-of-band iterations before agreement is reached. These actions do not affect the ultimate outcome of this process, which is the acceptance of agreed obligations by the provider.

provider's obligations to the customer are defined and agreed upon with regards to the handling of data.

In general terms, obligations can be legal, contractual or normative [3]. Legal obligations cover accountability obligations imposed by law and regulation, such as those which arise from the Data Protection Directive and the proposed Data Protection Regulation (currently undergoing the EU legislative process)[23]. Contractual obligations are derived from contractual agreements, such as the aforementioned contracts, SLAs and PLAs. Finally normative obligations are standards with which legal and/or natural entities are expected to conform because they are moral agents. Obligations belonging to these three categories are combined to produce the full binding set of obligations agreed upon by the cloud customer and provider. These obligations define precisely for what a cloud provider is to be held accountable by the cloud customer (and by extension by regulatory authorities) for the particular service relationship. As such, the set of **obligations** comprises the second type of accountability artifact that must be generated.

The agreed obligations at this stage may be expressed in a variety of forms, with natural language being the most common. These obligations need to be translated into specific policies to be enforced by the cloud provider. This necessitates a procedure for the translation of obligations into machine-readable policies that can be automatically monitored and enforced so that all handling of data is performed as per the agreed set of obligations. These (enforceable) policies, called accountability **policies**, comprise the third type of accountability artifact that must be generated. Accountability policies may describe operations more verbosely or at a more granular level compared to obligations and their exact form may depend on the architecture and implementation specifics of the service provided.

During the operation of the cloud service, various elements of it will access, process or otherwise handle personal data at some capacity. While regular operations may be expected to handle the data as prescribed by the relevant policies, an accountability-based approach requires the explicit provision of evidence to demonstrate compliance and meeting of obligations. Furthermore, information on the internal workings of the service, such as machine-generated logs, may need to be provided for the purposes of auditing or to support transparency reports. Both **machine-generated logs** and **evidence records** are thus important accountability artifacts that need to be generated and exchanged between different actors[24]. In addition, as discussed in Section 4.3.2, the evidence collection process drives the development of the "account", which is the principal means for supporting accountability at various phases of the accountability governance lifecycle in its own right.

While logs and evidence provide information on discrete actions and events that took place at specific instances during the operation of the service, the provider must also demonstrate how well it is meeting various accountability-related criteria during the continuous operation of the service, as these are determined by its stated obligations. The assessment of a provider's performance with respect to such criteria is performed through the measurement of service-specific subjective and objective metrics over defined periods of time. The provision of **metrics** to enable the evaluation of how well a provider meets various accountability-related criteria thus comprises another important accountability artifact.

It is also important for the provider to demonstrate global compliance to best practice standards. This is typically achieved through an auditing process, typically involving external auditors. The auditors produce detailed **audit reports** which are mostly used internally. The auditors also deliver more concise, summary-level **assessments** on the performance of the provider. The provider may also elect to be **certified** (or attested) against formal criteria defined in e.g. CSA Star Certification or Attestation [37]. These documents have historically been issued as paper reports, but are increasingly delivered as structured electronic documents protected against tampering.

[23] Section 8.1 lists the obligations stemming from the Data Protection Directive to which Cloud actors must adhere.

[24] Although in many cases evidence may include machine-generated logs, they can have different uses as well as different requirements for creation and handling. As such we consider them as two distinct classes of artifacts.

If an incident occurs or otherwise a failure to meet the agreed obligations is detected, the cloud provider must notify the affected parties, take steps to remedy the problem, and potentially offer redress. Thus, the construction of the **notification report** comprises an essential accountability artifact, containing besides a description of the incident, information on corrective actions or the proper remedial steps that have been taken.

Table 9 provides a summary of the various types of accountability artifacts discussed.

Accountability Artifact	Brief description
Capabilities	Document containing a description of the service in terms of the capabilities and controls it makes available to its user. The document may be presented in a machine-readable form to enable easier processing by software systems for analysis and comparison of service offerings.
Obligations	Document enumerating the binding legal, contractual and normative obligations of each party engaging in a service relationship, represented in a human-readable (natural language) form.
Accountability policy	Document or set of documents expressing the obligations of a service provider to a service consumer with regards to data handling in machine-readable form for automated processing.
Machine-generated logs	Machine- or human-readable objects, which are collected from various components of the cloud provider infrastructure (such as the network, hardware, the host operating system, hypervisor, virtual machines and cloud management systems, applications, etc.), detailing the actions and events that occurred during the execution of a service.
Metrics	Measurements of various service-specific objective and subjective performance characteristics over defined periods of time.
Evidence record	Structured information object which aggregates information from logs, documents and other sources with other metadata to demonstrate the occurrence of particular actions or events.
Audit report	Document which contains evidence records and related objects (i.e. logs, policies) to demonstrate compliance.
Notification report	Document or message meant to alert affected parties on the occurrence of an incident. It may contain relevant information on the incident, along with any potential corrective actions to be undertaken.
Assessment and Certification	Document which attests to the assessment of compliance to good practice (e.g. performed by an external auditor) or to the certification or attestation against a formalized criteria (e.g CSA Star Certification [37])

Table 9: Accountability artifacts.

The exchange of the aforementioned types of accountability artifacts at various phases of the service lifecycle enables the establishment and continuous evaluation of accountability between any two accountable cloud actors directly engaged in a service relationship. This model can be extended to cover cloud service supply chains of arbitrary length as long as every direct service relationship along the chain is fully documented by the exchange of corresponding accountability artifacts. Figure 15, illustrates how the process of exchanging accountability artifacts between pairs of actors in the supply chain can lead to accountability for the agreed context across the entirety of a given cloud service supply chain.

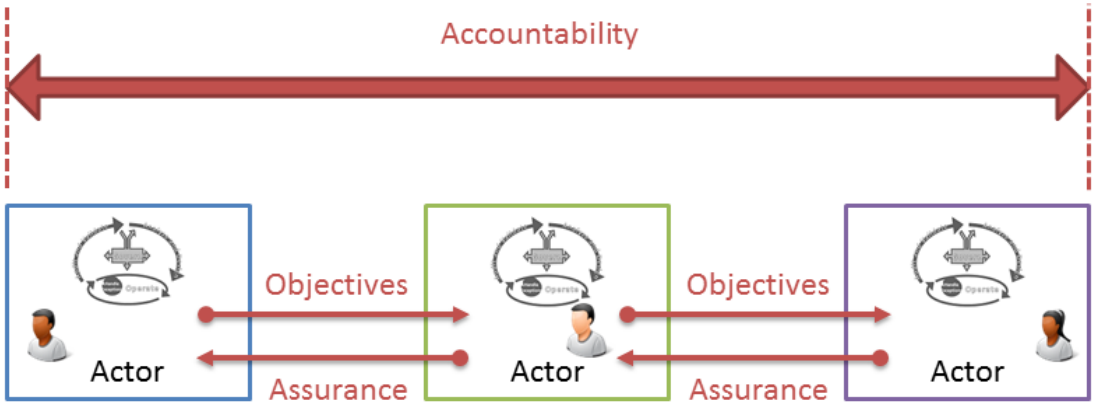

Figure 15: A model for end-to-end accountability in cloud service supply chains.

4.3 Service-Oriented Approach for Accountability in the Cloud

In the introduction to this chapter it was established that cloud service provision chains are composed from heterogeneous elements which reside in separate control, trust and ownership domains. This creates challenges to establishing accountability across the chain, as strong accountability cannot be achieved without cross-domain access to certain controls and views of information typically only available to the administrative entities inside each domain. To overcome this problem we have identified a narrow set of accountability artifacts that can be exchanged across domain boundaries to support accountability in a controlled manner without interfering with the structural properties of the boundaries themselves.

This approach achieves two critical goals:

1. It promotes a model for accountability across cloud supply chains which is pragmatic, technology-agnostic and does not make unrealistic assumptions about the cloud environment. By developing a framework for evaluating accountability based only on the exchange of information in the form of accountability artifacts instead of requiring changes in the way clouds are architected, the desirable properties of cloud computing that emerge from the ability to flexibly compose services and abstract lower-level complexity are fully preserved. Overall accountability is achieved by establishing accountability over each direct service relationship formed.
2. It does not dictate a “one-size-fits-all” approach. Every cloud actor can individually apply the accountability governance process in the way that makes sense to their business domain, organisational setup, capabilities and resources, as long as they can provide (and operate on) the required accountability artifacts in an interoperable way.

Accountability attributes can be generated via multiple combinations of potentially already-in-place and purpose-built systems and workflows of operations (e.g. existing logging systems re-configured to provide accountability-supporting machine-generated logs). In the A4Cloud Accountability Reference Architecture we propose a technology-agnostic service-oriented approach for the generation (or compilation) and exchange of accountability artifacts. As illustrated in Figure 16 which mirrors the example used earlier in the chapter, sets of services collectively designated as “accountability support services”, are operated internally by the various actors involved in a cloud service supply chain to support each actor’s primary operations by generating (or compiling) at the various phases of the lifetime of a service the necessary accountability artifacts that can be allowed to traverse control domain boundaries.

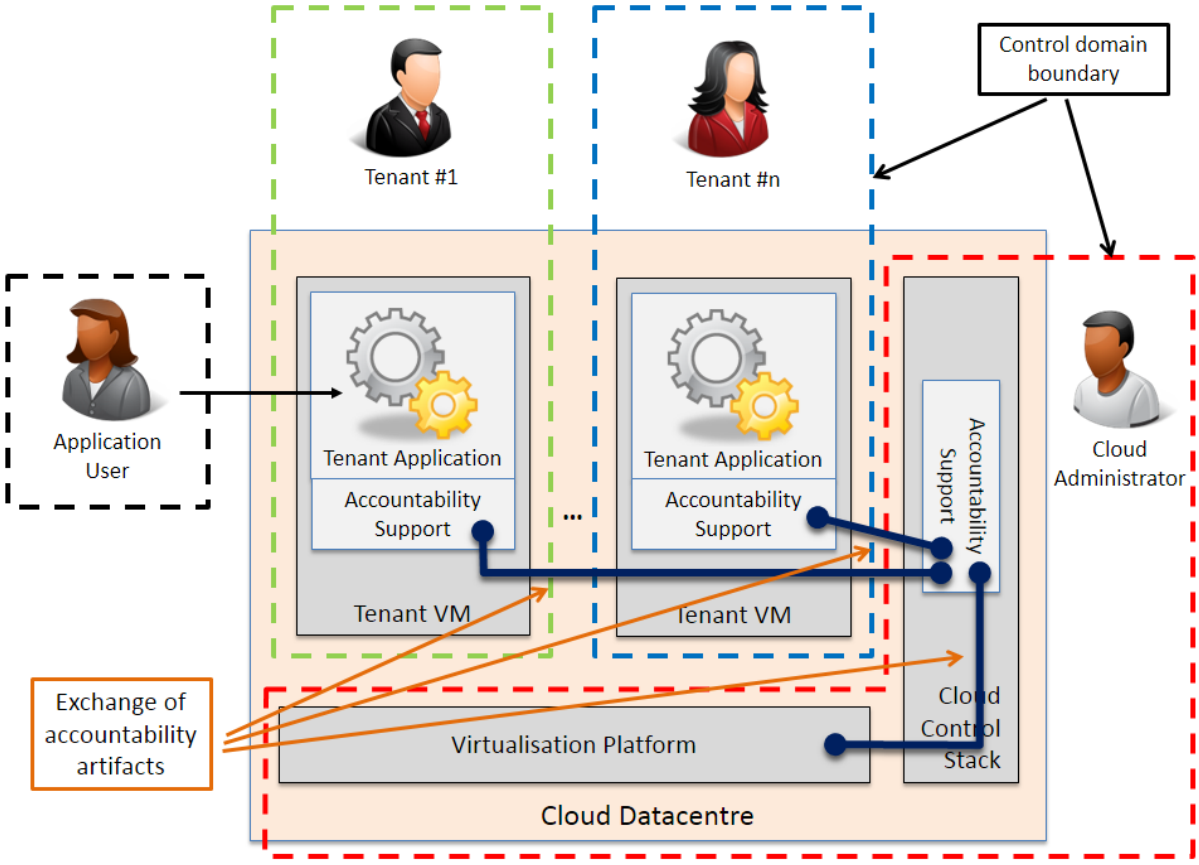

Figure 16: Supporting accountability via a service-oriented approach.

The A4Cloud RA does not dictate specific implementation requirements for the accountability support service model. Instead, we have identified a set of high-level functions which participants must implement and integrate with their systems in a context-specific manner to develop the accountability support service suite that is suitable for their environment. As illustrated in Figure 17, the RA Accountability Support Services are:

- Policy definition and compliance: systems that enable and facilitate the definition and configuration of policies and compliant extraction of policy terms from obligations.
- Policy enforcement: systems that ensure operations (such as handling of data) are performed exclusively according to defined policies.
- Environment state information: systems that collect information and monitor the state of the various systems and components that comprise a particular cloud service.
- Collection & management of evidence: systems that collect and compile evidence records about the operation of a particular cloud service and manage their full lifecycle according to specific integrity, confidentiality and access control requirements.
- Notification: systems that enable the formation, population and transmission of notification reports to authorised parties.
- Remediation: systems that assist in compiling and communicating remediation options to affected parties.
- Data subject enablement: services to ensure data subjects are provided with the ability to exercise individual rights regardless of direct participation to particular cloud service relationship.

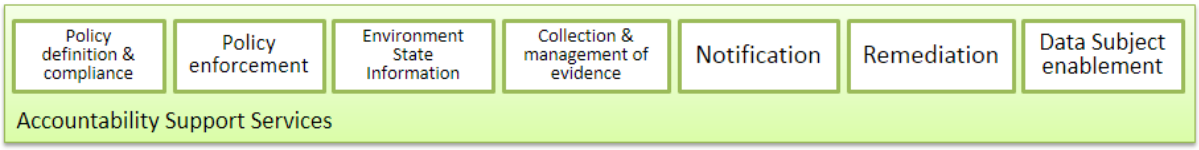

Figure 17: Accountability Support Services.

The rest of this chapter will focus on analysing these classes of accountability support services in more detail.

4.3.1 Policy Definition and Compliance

In general terms, policies in IT systems specify sets of rules related to a particular purpose, such as defining the security credentials one must possess to access a particular data object and the actions to be taken under various conditions. In the scope of the A4Cloud project the obligations an organization has in regards to how it must handle personal data are also expressed through policies.

Organizations that are subject to obligations need not only to meet their obligations, but also to ensure that their business partners and sub-contractors do not invalidate them. In particular, an accountable organization needs to make sure that their obligations to protect personal data are adhered to all across the service provisioning chain.

An important aspect of governance in an accountable organization is to define and deploy policies for their data processing practices and to make sure that they are followed by all the involved service providers. The policies should ideally travel with the data, and they should be used as input to monitoring of data processing practices, to generate evidence that policies are fulfilled, to correct policy violations that may occur and in general to demonstrate policy compliance.

Therefore, an accountability-based approach requires services via which to express accountability obligations using a common policy specification language (or standard interoperable policy specification languages) and to distribute them throughout the cloud supply chain. An additional component of policy definition support is the checking of compliance between the original obligation or law and the actual set or rules and actions defined in the policy.

The A4Cloud project has developed a cloud accountability policy representation framework [38] (see Figure 18 from the same source) based on the following design requirements:

- Access and Usage Control rules - express which rights should be granted or revoked regarding the use and the distribution of data in cloud infrastructures, and support the definition of roles as specified in the Data Protection Directive, e.g. data controller and data processor.
- Capturing privacy preferences and consent - to express user preferences about the usage of their personal data, to whom data can be released, and under which conditions.
- Data Retention Periods - to express time constraints about personal data collection.
- Controlling Data Location and Transfer - clear whereabouts of location depending on the type of data stored and on the industry sector processing the data (subject to specific regulations) must be provided. Accountability policies for cloud services need to be able to express rules about data localization, such that accountable services can signal where the datacentres hosting them are located. Here we consider strong policy binding mechanisms to attach policies to data.
- Auditability - Policies must describe the clauses in a way that actions taken upon enforcing the policy can be audited in order to ensure that the policy was adhered to. The accountability policy language must specify which events have to be audited and what information related to the audited event have to be considered.
- Reporting and notifications - to allow cloud providers to notify end-users and cloud customers in case of policy violation or incidents for instance.
- Redress - express recommendations for redress in the policy in order to set right what was wrong and what made a failure occur.

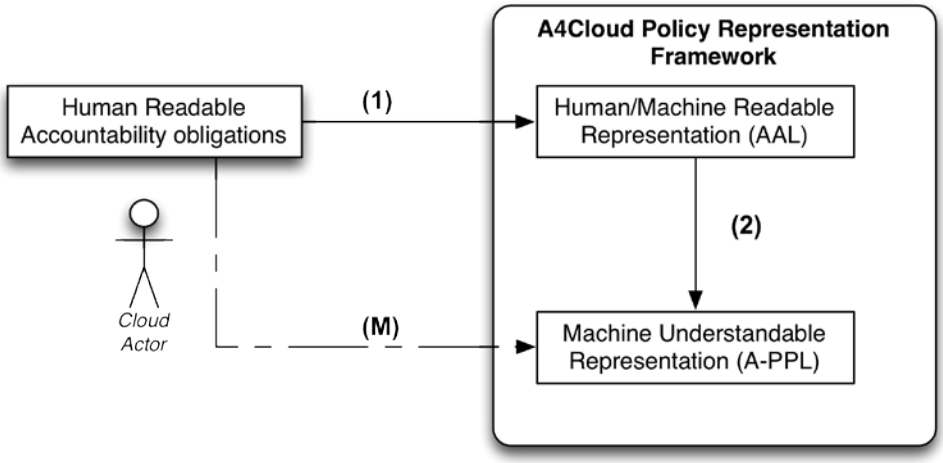

Figure 18 - Overview on the accountability policy representation framework.

While there exist languages for privacy and theoretical models for accountability, we proposed AAL, a domain specific language (DSL), to express abstract obligations in a language close to sentences in laws, data directives and contracts. This language is equipped with a formal logical background and is the first stone towards policy enforcement trough accountable design and verification [39]. We also define a concrete policy enforcement language, called A-PPL, as an extension of the PPL language [40]. The proposed framework offers the means for a translation from abstract obligations expressed in AAL to concrete policies in A-PPL.

In the A4Cloud policy framework, the cloud customer must define the high-level data-handling requirements and obligations in AAL, a representation which is both human- and machine-readable. These policies can be communicated and agreed with potential cloud providers. Since AAL has an associated formal semantics and there are tools to check its properties using a model checker which is embedded in the AccLab tool[25] it is possible for cloud customers to use the tool to check cloud provider policies for conformity and compliance with their own data governance practices and regulations.

If the cloud provider uses subcontractors, then it must check the conformance of the cloud customer policy with the practices of the actors further down the chain. In Figure 19 we illustrate a cloud service chain. The SaaS provider must propagate *Policy 1* from the Cloud Customer to an adapted form to manage the resources it uses from its subcontractor. AAL and AccLab have the necessary features to help in the definition and verification of these dependencies. Since AAL is abstract, it acts as a pivot model between different accountability obligation representation models and such promotes the interoperability of heterogeneous systems.

[25] http://www.emn.fr/z-info/acclab/

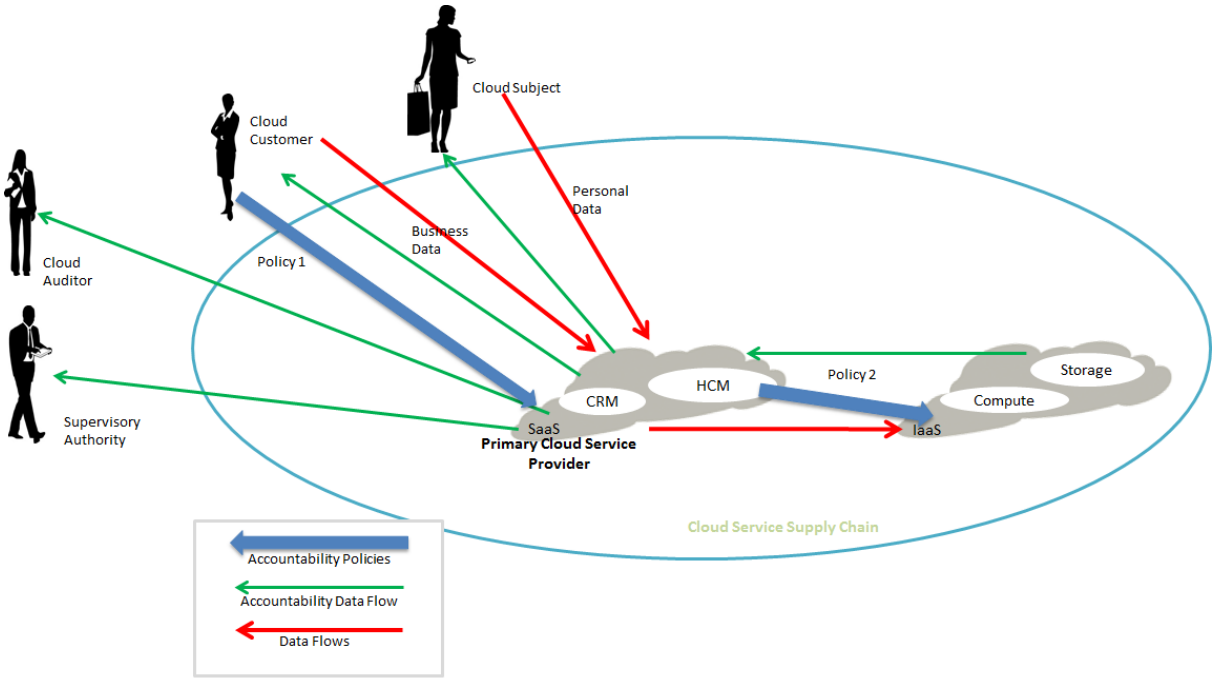

Figure 19 - Accountability Policy Distribution and Data Flows

At a later phase, the accountability obligations expressed in AAL are (semi-)automatically translated into a machine understandable policy (Step 2 in Figure 18). The main target of the mapping step is A-PPL, for expressing usage control and data handling obligations [41]. However the framework is flexible enough to accommodate further source and target languages. For instance, Privacy Level Agreement (PLA[26]) is emerging as a federating initiative to make privacy statement declarations clearer and to allow to assess what is the level of privacy disclosure to a given service. The work package D-3 is working on support for transforming PLA's into enforceable policies in A-PPL.

4.3.2 Policy Enforcement

Once policies are expressed into machine readable format, an accountable system should implement an automated enforcement framework in order to fulfil the requirements defined through policy rules. According to the nature of the obligations, different enforcement mechanisms can be integrated within this framework. This section first discuss the main components of the enforcement framework and further suggests some relevant enforcement techniques.

A policy enforcement framework should at least define two main components: the Policy Control Point (PCP) where policies are defined and decisions are taken and, the Policy Enforcement Point (PEP) which enforces policies through different technologies. The PCP first maps each policy rule to a specific set of actions/operations and takes decisions on when and how to perform such operations. The PEP executes the actions upon PCP's decisions (such as data handling operations based on access control decisions) or continuously monitor some events (logging).

PEP can implement different enforcement techniques: while preventive solutions such as privacy based access control solutions are implemented to meet data handling obligations, detective solutions help in order to verify the compliance with such data handling obligations. Specifically:

- **Privacy enhanced data access control**: personal data can be protected through well configured privacy-enhanced solutions. While encryption techniques become mandatory for the protection of the data stored in the cloud, a well-defined access control framework combined with a secure identity management system will help to meet data handling obligations. The new enforcement framework should also allow data subjects to have full access over their data (read, update, delete). Another way to enforce such rules is to

[26] https://cloudsecurityalliance.org/research/pla/

implement sticky policies whereby rules and constraints travel with the data. This is especially beneficial in a cloud environment where data can travel.

- **Monitoring/logging solutions**: Ensuring compliance with respect to data handling obligations is not always an easy task; the enforcement framework should therefore implement some dedicated logging solutions. There is a specific need for data transfer monitoring since controlling the location of data can remain difficult; data transfer monitoring tools may help to discover unexpected events. The integrity of logs is considered as the critical functionality of a secure logging solution.

Reliable policy enforcement requires a number of trust assumptions to be made:

- The cloud service provider wants to demonstrate accountability at a reasonable cost, therefore it would have no interest to tamper with the accountability enforcement engine.
- Access to personal data will not circumvent the accountability enforcement engine. Notice that the engine by itself cannot guarantee this, since it cannot control the entire environment it is part of (operating system, network, etc).
- Further providers in the cloud service chain provide assurance about the security and privacy procedures and controls such that data subject rights can be guaranteed.

4.3.3 Collection and Management of Evidence

The provision of evidence is an essential element of an accountable system, enabling demonstration, verification and the construction of the account, while motivating and guiding a range of other functions and procedures important for accountability. Evidence contribute to the detective controls inherent to accountable systems. In particular, they support the account by providing the arguments to show whether policies, norms and regulations are satisfied at any stage of the service delivery. Therefore, the A4Cloud project proposes an account-oriented definition of accountability evidence: "*a collection of data, metadata, and routine information and formal operations performed on data and metadata, which provide attributable and verifiable account of the fulfilment of relevant obligations with respect to the service and that can be used to convince a third party of the veracious (or not) functioning of an observable system*" [42].

As shown in Figure 20: Accountability Evidence, the project determines three verification levels where accountability evidence should be provided in order to verify the correct operations of the system:

- Organizational policies: At this layer, evidence support the correct definition of policies for the context.
- Mechanisms and procedures: This involves the provision of evidence that appropriate mechanisms and controls are deployed in accordance with the obligations defined at the policy layer.
- Operational practices: The evidence from this level should reflect that operations (what is actually happening) satisfy the requirements expressed in the policies (what is supposed to be happening). This may need continuous monitoring, recording and analysis of the activities in the service delivery.

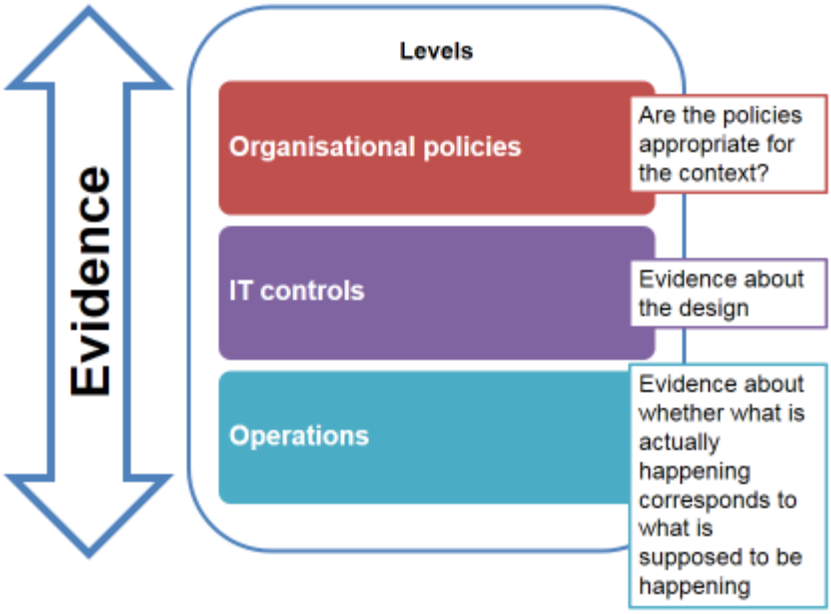

Figure 20: Accountability Evidence

In line with the above levels, the project identifies the following five major types of evidence relevant for accountability:

- **Data processing practices**: evidence on operational practices such as replication, storage, deletion, copy, access, optimization, consent, security, segregation, proofs of retrievability, etc.
- **Data collection practices**: evidence on data collection practices such as policies compliance, privacy issues, security breaches, etc.
- **Notification**: evidence that notifications were sent to the interested stakeholders in case of privacy issues (unauthorized access, etc.), policy violations, security breaches (data leakage, data lost, corrupted or tampered, etc.) and services or policy modifications, as well as service practices and users rights;
- **Remediation**: evidence on remediation to their customers in case of security breaches, privacy issues and policy violations.
- **Organisational practices**: evidence on requirements related to employee's training, system certifications, privacy policies, etc.

The provision of evidence is a complex process that entails identification, association and collection of information that will constitute evidence from various sources; processing and use of cryptographic techniques to ensure integrity; secure storage; and presentation for the various different stakeholders, auditors and regulators. Specifically, the process must be designed in such a way as to:

- Give account of events and occurrences within the services of cloud providers in a non-invasive manner to the customers' privacy and without exposure of sensitive material from the inside of organisations;
- Ensure information is collected securely in a tamper-evident process, which however may allow verification checks from different services, tools and trusted third parties (e.g. auditors);
- Avoid excessive data collection, minimising privacy issues and ensuring scalability of evidence storage with time, even with growing user bases and multi-tenancy scenarios.

To address this challenge the A4Cloud project has designed a Framework of Evidence (FoE) consisting of a set of mechanisms for extracting evidence in typical scenarios encountered in cloud services and managing their full lifecycle [42]. The FoE provides an automation support to all the phases of the evidence lifecycle: monitoring, collection, storage, verification and presentation. The FoE considers four main types of sources:

- internal data as logs, metadata SIEM outputs, etc., collected from monitored operations and services;

- data from end-users or from a cloud service to another, or internal data as customers' or employees' lists, training certificates, seals, etc.;
- documentation of the service's practices, contracts and organisational policies that relate to accountability and for which implied obligations can be expressed (manually or by other A4Cloud tools like AccLab or COAT – discussed later in the document) in machine-readable policies;
- cryptographic proofs that are generated upon request at a particular point of time.

Using the FoE, the collected and processed evidence are assembled and stored as *records*. These records contain elements supporting a claim such as the actions that the evidence records, a timestamp, the identity of the authenticated agent that performed the recorded action and a reference to the policy that the action may or may not comply with. Among the supporting elements included in the records, audit logs and cryptographic proofs are of particular interest in the proposed Framework of Evidence. Audit logs provide documentary evidence of a sequence of events, actions and changes observed in the accountable service. They support the occurrence (or not) of an operation and demonstrate compliance or non-compliance with the policies. Audit logs must be protected against adversarial tampering or deletion to provide a non-disputable record of events. Logs also enforce the correct system behaviour since the latter is held accountable for the events, changes and actions that are logged in the audit logs. Cryptographic proofs provide an instantaneous assurance of correct (or not) behaviour of the accountable system, verifiable at any point of time. They are intended to convince any entity verifying them that the system provides the appropriate safeguards when processing and storing data. As an example of such proofs, in the project we focus on a particular cryptographic proof of storage called proofs of retrievability. They ensure with high probability that the accountable system stores the outsourced data as expected in policies, contracts and regulations and that this data is retrievable at any point of time.

Evidence contribute to account definition, attribution of responsibilities and policy violation reports.

Therefore, to support accountability across the supply chain, services providing the following high-level functions need to be provided for evidence provision:

- Support for the extraction of evidence targets (i.e. what to monitor, when, etc.) from policies, ensuring compliance and full coverage (i.e. no obligations described in policies are left unmonitored);
- Support for collection of logs from different sources and parts of the infrastructure;
- Support for full lifecycle of evident management (i.e. mechanisms to extract, assemble, secure, store evidence, etc.), ensuring privacy, integrity, verifiability;
- Support for enforcing access rights on evidence;
- Support for context-specific presentation of evidence to different parties possessing different access-rights and privileges;

Evidence are also supporting elements for the provision of the account. The C-2 conceptual framework [3] defines three types of accounts that can be provided. We outline here some examples of evidence that can be provided to support these three different types of accounts:

- *Proactive account*: This kind of account may report the quality and security levels of the services provided by the cloud. Evidence in this case may consist of certifications of the compliance of the offered data processing and storage services with regulations and contracts. The certificates should designate the party that provides the service, the issuer of the certificate, the validity period of the certificate, the level of security guaranteed by the certified service, etc.
  Additionally, proactive accounts can be supported by consistency checking reports that act as evidence of contractual obligations agreed between the Cloud Provider and its customer.
- *Account on compliance*: Evidence should be collected to provide an account of a legitimate event to demonstrate that the Cloud Provider complies with regulations and contracts. Logs are the most relevant sources of evidence to proof policy compliance. However, the integrity of logs may not be always easy to verify. Hence, there may also be additional cryptographic proofs for some cases such as proofs on data storage: for example, Proofs of Retrievability, such as the one presented in [42], are cryptographic proofs that verify whether or not a data storage service actually stores the outsourced data. Collected on a periodic basis, these proofs of retrievability enable an auditor to check the correct storage of the data, meaning that

the service stores the data as expected by regulation and contracts. Therefore, the proofs of retrievability can support this kind of account.

- A*ccount on security breach*: In case of the occurrence of an incident, evidence should demonstrate that the fault actually occurred, identify the actors and the environmental settings that lead to the incident, assign time and date to the event, and have a reference to the particular policy rule(s) that the reported event infringe. Logs can be an example of such inculpatory evidence to provide an account in case of an incident and to attribute the responsibility of that incident to a particular actor in the cloud. Indeed, logs report the actions performed by a particular entity and record the identity of that entity and the timestamp corresponding to the reported event.

The RA does not prescribe a specific method for providing an account, recognizing that in many cases the form and contents of the account as well as the context of its provision are specific to the particular circumstances of the actors involved. For example, a particular account may consist of highly structured and annotated information allowing it to be transmitted electronically in an automated fashion, or may be an e-mail containing textual information that is not organized in a specific way.

However, given that the construction and provision of the account requires the provision of evidence, the RA notes that in order to support accountability across the supply chain via the construction and provision of the account, actors need to implement the services supporting evidence provision outlined above, as well as implement services supporting context-specific presentation of the account to different parties possessing different access-rights and privileges, in cases where the account can be processed automatically by computers.

4.3.3.1 The role of evidence in metrics for accountability

As previously discussed, the notion of evidence plays a central role in accountability frameworks (cf., Figure 20). Evidence can be used to assess the suitability of organizational policies, IT controls and operations. In this sense, the usage of quantifiers (metrics) capable of performing such assessment brings light on the existing relationship between accountability metrics and evidence. From the perspective of the accountability framework, metrics are a means for demonstrating accountability, through the provision of *quantifiable evidence* of the application of proper practices and the performance of operational processes. If *evidence is treated as the data collected to support a metric* (including the parameters necessary to calculate and repeat the measurement according to the metric definition), then improvements in the implementation of accountability practices can be justified and traced in a quantitative way. The notion of evidence is broad enough to encompass not only experimental observations, but also system logs, certifications asserted by a trusted party, or textual descriptions of a procedure. Evidence contributes to the S.M.A.R.T. characteristic of a metric, by:

- S - Specific (or Significant) – Evidence allows metrics to be specific and targeted to the area being measured. For example, evidence provides information about the actors involved on the measurement, the purpose or benefits of the metrics, etc.
- M - Measurable (or Manageable, Meaningful) – Evidence supports the elicitation of “meaningful metrics”, by providing clear information about the input parameters.
- A - Achievable (or Appropriate, Attainable) – Metrics should not be developed if one cannot collect accurate or complete data. Evidence in this case helps stakeholders to assess the quality of the inputs, and ultimately to evaluate the assurance associated to the measurement result.
- R - Repeatable – Provided evidence is essential to trace the measurement process applied by the assessor to obtain the result. Evidence should provide enough information to repeat the measurement, as many times as required, in order to validate the obtained measurement result.
- T - Timely metrics are those for which data are available when needed, and this feature is directly related to the quality of provided evidence.

The five characteristics discussed above directly relate the assurance associated to the measurement result, to the quality of the provided evidence. This fact is partially represented by Figure 4, where the listed “Source of Assessments” is related to different qualities of provided evidence. In order to support accountability, it is necessary to discuss in further detail two main challenges associated to the joint notion of evidence and metrics, namely:

1. The (quantitative) assessment of the evidence's quality.
2. The use of evidence to trace a measurement result.

The first challenge (quality of evidence), relates to model evidence in such a way that it can be evaluated and associated to the actual definition of the accountability metric. Sharing a common understanding for representing and using metrics/evidence, also allows actors to manage (e.g., comparing CSPs) and standardize them (as part of metrics models like NIST CSM [20]). As mentioned in a previous paragraph, the quality of evidence is closely related to the notion of metric assurance.
The second challenge is quite related to the actual definition of the accountability metric and enables the analysis of a measurement result into its different components (including underlying measurements). If properly designed, this analysis might help to improve the levels of transparency achieved by organizations and CSPs by realizing the evidence associated to the elements involved on the actual measurement process.
Both challenges will be further elaborated in the next (and final) version of this deliverable.

4.3.4 Environment State Information

The ability to collect evidence of the events and actions that take place inside a cloud environment is absolutely reliant on the ability to observe and monitor the state of all systems that operate in this environment. Clearly, this requires that all systems are instrumented and monitored so that information about their state and inner workings is extracted and made available for analysis at various levels of granularity. Because most cloud services are implementation-specific and involve customisations, we do not propose a specific architecture for environment state information collection services. Nonetheless, we assume that capabilities to monitor the state of the cloud environment are in place.

Using the Cloud Trust Protocol (CTP)

The Cloud Trust Protocol (CTP) is a project currently being developed by CSA (a partner in A4Cloud) which aims to create a RESTful API that will allow cloud customers to query cloud providers about the security/privacy/compliance level of their service in near real-time. In the next paragraph, we analyse the potential use of this new tool for the purpose of providing the "account".

It should be highlighted that CTP does not define a monitoring architecture or framework, it only defines an API to present the result of the monitoring in a standardised way.

In CTP the level of security of a cloud service is expressed through the measurement of "security attributes", which apply to "cloud resources". More precisely, CTP has adopted the following data model:

- A cloud service is divided into a set of resources (e.g. a VM, an API call, a database)
- Each resource has a set of attributes (e.g. "uptime", "confidentiality at rest", "incident response performance")
- Each attribute has a set of measurements, where a measurement is a process that enables to quantify or qualify an attribute, according to a specific metric (e.g. "percentage of failed requests per hour").
- Each measurement produces a value called "measurement result" (e.g. "99.98637 % / hour")
- Each measurement can also be associated with an objective, which represents what is typically described in a SLA by describing a constraint on the measurement result (e.g. "result > 99.95 % / month")

This data model enables cloud customers to get precise information about the security level of a service, and compare these levels with the security objectives that have been defined by the provider (provided of course that the provider is willing to provide a valuable set of attributes and measurements). In addition to these elements, the CTP API also proposes:

- Triggers: these are conditions (like objectives) that will generate an alert, sent to the customer.
- Logs: logs that are stored by the provider, for each trigger event.

With the high-level presentation of CTP we can examine how closely it can be related to the notion of the account. First, we can observe that:

- By construction, CTP triggers provide a mechanism to be notified of events, whether these events are "legitimate" or "incidents" depends solely on the choice of the trigger conditions

formulated by the customer. CTP triggers and logs therefore provide a vehicle for *'a report or description of an event',* which defines the notion of the account (see 3.4).

- CTP "objectives", which describe obligation of the organisations, are closely linked to the notion of "proactive report".
- The account is expected to answer a certain number of question: notably who, what, where and when. By construction, the CTP data model provides:
    - Clear identification of "Who" (via service-units).
    - Describes partially "what" through measurement results, but does not identify "root cause"
    - Describes "where" in the sense of identifying the "resources" to which attributes apply. Moreover "location" can be an attribute of a resource in itself.
    - Describes when through timestamps.
- The CTP standards envisions explicitly the possibility to add a digital signature to measurement results, though this features is not standardized as of this date.

Yet, we can also argue that CTP is not designed to present all the features that are expected of an "account" in the sense defined in A4Cloud:

- CTP does report "evidence" in the strict sense but only "results". Logs in CTP only materialize the fact that a result failed to match a condition but they do not provide support for the results themselves.
- CTP does not describe actions taken to deal with an incident, but only that it occurred.

One last aspect to consider is that CTP targets mainly security/privacy attributes that can be minored in an automated way. This does not cover all elements that an organisation should monitor as part of an accountable process.

In summary, the CTP API can be used by organisations as a tool to provide the "account", of technical and measurable attributes related to security, privacy and compliance in general. However, the CTP API does not provide a mechanism to support the presentation of evidence in relation with incidents but only a mechanism to report that a specific incident occurred.

4.3.5 Notification

Notification is an essential element of accountability. A strong accountability-based approach requires cloud providers to notify all affected parties of the occurrence of an incident or discovered policy violation within a reasonable timeframe. Notifications may be provided through common means such as e-mail or letters to the relevant parties, or dedicated communication channels designated for the purpose (usually between cloud providers).

The RA does not prescribe particular methods for notification recognizing the fact that the circumstances around each incident rarely are the same and flexibility should be allowed in which mechanisms to mobilize during response. However, the following functions should be implemented for a strong accountability-based approach:

- Obligations with regards to providing notification within a predefined, reasonable timeframe should be reflected in the policies enforced. As such, any policy-support services implemented (discussed in section 4.3.1) should ensure that this element is explicitly supported.
- An automated approach to transmission of notifications can reduce the burden of operating this part of the accountability lifecycle. As such, organizations may opt to implement services supporting the exchange of machine- and human-readable notifications based on a predefined protocol that has provisions for the inclusion of information about the incident (including evidence of which subset of a subject's personal data were involved in the incident) and, where possible, links to an automatic remediation management system, discussed next.

4.3.6 Remediation

Like notification, remediation is also an essential element of accountability, referenced directly by the fourth and final accountability practice in the accountability model presented in Chapter 2. Again, the specifics of a particular remedial action in response to a specific violation depend on the

circumstances of the violation itself, and many may be enacted ad-hoc. As such, the RA does not propose a particular mechanism for remediation. We do, however, note that a framework for the systematic addressing of violations and provision of remedies depends on the proper implementation of the accountability-support services described in the previous sections. Specifically, service functions should be in place to facilitate:

- the ability to detail the origin of policy violations in order to provide appropriate responses. Customers need to know whether the policy violation occurred as the result of an attack, a deliberate action by the provider, an unintended alteration or any other mean, in order to make an educated decision about the efficacy of the proposed remediation or request additional redress.
- the ability to suggest response actions to ease the process for customers responding to the event. Customers could get assistance from the service provider in performing any necessary step on their part to handle the event. This would include any remediation action deemed appropriate.

Accountability support tools facilitate notification and remediation processes in the cloud. Starting from the filling of complaints towards incident detection (AAS, DTMT), incident handling (IRT), automated notification (A-PPL-E, DT) and remediation and redress, the workflow across these phases and tools can provide much more assurance that corrective actions have been taken in accordance with contracts and regulations. As the work package D-4 is currently working on the remediation and redress processes, the final architecture will contain a much clearer vision of how remediation can be achieved in the cloud.

4.3.7 Data Subject Enablement

The final necessary element for end-to-end support of accountability across cloud supply chains is the provision of services aimed at enabling data subjects to consent, control, review and correct their personal data held in the cloud.

Specifically, the Data Subject should be provided with facilities to:

- provide consent on the use of their data;
- request from a data controller access to their data stored in the cloud for review;
- request from a data controller to correct or delete their data stored in the cloud;
- view detailed information about how the data has been shared and used by the data controller;
- receive notifications of incidents affecting them;
- receive assistance in requesting remediation and redress.

To support these functions, actors along the chain (excluding data subjects) should therefore implement services that implement the following functions:

- track and produce evidence of all data uses;
- provide means for data subjects to view their personal data held, along with meaningful metadata (e.g. time of data disclosure, etc.);
- provide means for data subjects to amend or request deletion of their personal data held;
- since almost all data subject controls will only interact with the data controller and not the actual data processors along the supply chain, services should be in place to facilitate the passing of data subject requests in an appropriate form to the relevant processors. Clearly, to support this capability the functionality provided by the accountability-support services described in previous sections (such as services to exchange enforceable policies and provide evidence) will be required.

One tool to consider in order to receive information about incidents occurring in the supply chain is CTP (the Cloud Trust Protocol). As described in 4.3.4, CTP can report alerts regarding the measured security level of a service, or more precisely the measured attribute of a specific resource. For example, if the availability of a disk resource falls below a user defined threshold, an alert can be sent to a user. This works also across a supply chain where one cloud provider can send an alert to another provider, which in turn sends an alert to another provider. In the case of a supply chain, the

alerts might change in nature: an availability issue with a disk might become an availability issue with a database for the next provider in the chain. Importantly, CTP is agnostic to the notion of “data subject” and only reports security levels and the resources they affect. CTP is a protocol designed to be used between cloud customers and cloud providers, not between data subjects and providers. As such it is the responsibility of the “customer facing” provider to identify which cloud subjects are affected by an incident and to notify them appropriately.
.

4.4 Integration and Adoption Patterns

This section aims to offer a practical approach to the adoption of the Accountability Framework by identifying patterns, which respect to the integration of A4Cloud approach to the existing capabilities of a cloud environment and the adoption of this approach by the identified stakeholders.

4.4.1 Integration Patterns

In this section, we introduce the concept of integration patterns for accountability, which is inspired by the existing attempts of several software vendors and IT system and solution providers to provide some common ways to facilitate integration in multi-scaled and multi-oriented systems. From an accountability perspective, the integration patterns refer to the interaction of the accountability mechanisms with each other and with the external world. By saying so, we emphasise on the need for interoperability among the different layers of a cloud ecosystem in order to achieve accountability, as this is described in the four phases, namely agreement, reporting, demonstration and remediation. The integration patterns may go beyond the strict boundaries of the technical details of an accountability solution and analyse the integration requirements along processes (or even other non-technical mechanisms, such as legal contracts) as well.
In the context of the A4Cloud Reference Architecture, we will introduce the following integration patterns:

- Agreement patterns: this family of patterns analyses the integration patterns to serve the agreement practices of the Accountability Framework and involve:
    - Policy Specification pattern; this pattern integrates the various types of data handling procedures that reflect the accountability dimensions of a data protection problem (for example, the specification of data access or data transfer rules).
    - Policy enforcement pattern; this pattern describes the common functions for a policy decision point.
- Reporting Patterns
    - Logs communication pattern; this pattern describes the way that the collected logs are communicated with a specific transformation model to serve an integrated approach for log reasoning and analysis
    - Incident Messaging pattern: this pattern is used to communicate incidents via a point to point approach.
- Demonstration Patterns
    - Evidence building pattern; this pattern provides the way to build an evidence shared repository from multi-source log listeners and collectors
- Remediation Patterns
    - Incident Response pattern; this pattern defines a communication path for enabling the exchange of incident management and remediation actions.

As an example of the logs communication pattern, we consider the actions happen of the cloud provider, which offers the infrastructure. In other case, this infrastructure is based on OpenStack **[43]** and, thus, the relevant pattern refers to the integration of the OpenStack services with the accountability ones to collect the appropriate logs. This integration includes the monitoring and analysis of the events referring to the traffic realised in the OpenStack network and, especially, the Controller Node. An accountability service should implement a pattern to log the events collected in this part of the OpenStack infrastructure and parse these logs to identify the type of actions happened and filter them, based on policies. The pattern should, then, implement a protocol so that the collected logs can be communicated to other referring accountability services (e.g. evidence).

4.4.2 Adoption Patterns

This section introduces the concept of the adoption patterns of the A4Cloud Reference Architecture for the different cloud service models and cloud computing and data protection roles. This approach is inspired by the IBM handbook on the Cloud Computing Reference Architecture (CCRA) version 4.0[27]. As reflected there, an adoption pattern is "is a collection of commonly observed functions and features that customers desire in their solution, where a customer starts to solve a specific business problem, typically driven by the same business motivation". In IBM CCRA 4.0, the adoption patterns are solely driven by the adopted cloud service model that a specific cloud provider wants to adopt. In A4Cloud, the adoption pattern takes the form of a guidance for the different actors in the cloud service supply chain in order to follow the accountability lifecycle and be accountable to their collaborating providers.

In more details, the A4Cloud adoption patterns do not focus only on the cloud provider perspective, but they try to capture the operational needs and the respective accountability requirements for the end-to-end chain in the cloud service provision. As such, these patterns should investigate on the responsibilities and obligations of the actors, according to their position in the cloud computing / data protection role matrix and offer a guided roadmap for the adoption of the Accountability Framework in their cloud-based business model. From an accountability perspective, the position of an actor in data processing is very important, which raises various security and privacy requirements that should be addressed and a number of legal and normative obligations that should be implemented. In that respect, we can distinguish among the following A4Cloud adoption patterns:

- The Accountable Cloud Customer and Controller (ACCC) Adoption Pattern; this pattern refers to cloud customers, who act as data controllers and regulate the type of data to be collected from data holders and the purpose for doing so.
- The Accountable Cloud Provider and Controller (ACPC) Adoption Pattern; this pattern refers to cloud providers, who act as data controllers and regulate the type of data to be collected from data holders and the purpose for doing so.
- The Accountable Cloud Provider and Processor (ACPP) Adoption Pattern; this pattern refers to cloud providers, who act as data processors.
- The Accountable Cloud Customer and Processor (ACCP) Adoption Pattern; this pattern refers to cloud customers, who act as data processors.
- The Cloud Subject Enabling (CSE) Adoption Pattern; this pattern refers to cloud subjects, who are enabled with guidance on how to track the disclosure of their personal data in the cloud environments.

Such adoption patterns aim to address the principal question for the cloud actors about "how am I going to be accountable to my customers?", while providing informed guidance to the data owners on how they can take control over the results and the impact of the data processing procedures followed by the cloud providers to their data that has been disclosed in the cloud service supply chain. In general, the A4Cloud Adoption Patterns should provide a roadmap for the adoption of the Accountability Framework in all the respective lifecycle phases. Each pattern should, then, instantiate these general principles for the case of each target stakeholder, by detailing on the following items:

- The security and privacy requirements that drive the definition and the specification of the intended cloud-based business;
- The actors that are involved and how they relate with the primary actor of the adoption pattern;
- The accountability use cases and the respective accountability functions for this pattern;
- The roadmap to adopt the Accountability lifecycle;
- The instantiation of the A4Cloud Reference Architecture, highlighting the intended use of each A4Cloud tool and any external tools that serve application specific functions, making references to target AMM;

27 https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Wf3cce8ff09b3_49d2_8ee7_4e49c1ef5d22/page/IBM%20Cloud%20Computing%20Reference%20Architecture%204.0

- The realization of the operational model for the A4Cloud components and the external tools and the respective information models

5 The A4Cloud Toolset

The tools comprising the A4Cloud toolset have been designed to support accountability for data governance in the cloud by specifying certain functions identified during the development of the A4Cloud accountability framework. The tool design process was motivated by the goal to provide stakeholders with tools supporting those elements of accountability for which little or no support was found to exist, while complementing existing privacy and security mechanisms. Each A4Cloud tool addresses different elements of accountability, and may operate over different timescales and interact with data at different points of its lifecycle compared to others. The descriptions included in this section are preliminary and will be updated prior to the final integration of the tools.

The A4Cloud toolset is composed of the following eleven (11) tools and a plug-in:

- The Data Protection Impact Assessment Tool (DPIAT): This tool is used by Small-Medium Enterprises (SMEs) to identify the risks in a given configuration and environment of carrying out a certain business transaction, which involves the processing of personal or confidential data.
- The Cloud Offerings Advisory Tool (COAT): This tool is designed to assist potential cloud customers (SME organizations and individuals) in assessing and selecting cloud offerings, with respect to certain security and privacy requirements.
- The Accountability Lab (AccLab): This tool translates human readable accountability obligations expressed in the Abstract Accountability Language (AAL) into a lower level machine-readable accountability policy language called Accountable Primelife Policy Language (A-PPL).
- The Accountable Primelife Policy Engine (A-PPL Engine): This tool enforces data handling policies expressed in A-PPL and generates logs with respect to the actions enforced in compliance to these policies.
- The Audit Agent System (AAS): This tool enables the automated audit of multi-tenant and multi-layer cloud applications and cloud infrastructures for compliance with custom-defined policies, using software agents.
- The Data Transfer Monitoring Tool (DTMT): This tool automates the collection of evidence describing how data transfers within a cloud infrastructure comply with data handling policies.
- Data Track (DT): This tool is used by data subjects to get a user-friendly visualization of all personal data they have disclosed to cloud services, with the additional capability to rectify data if necessary.
- The Transparency Log (TL): This cryptographic tool provides a secure and privacy-preserving unidirectional asynchronous communication channel, typically between a cloud subject and a cloud provider. Messages can be stored on untrusted system and can be still be securely retrieved asynchronously by recipients.
- The Remediation Tool (RT): This tool assists cloud customers (individuals or SMEs) in responding to real or perceived data handling incidents.
- The Incident Management Tool (IMT): This tool is the entry point for managing anomalies and violations that occur in cloud services and should be notified to the cloud subjects, such as privacy violations or security breaches. The tool enables cloud subjects receive incident notifications and takes the initial steps to respond to these incidents, by gathering comprehensive and structured information related to these incidents.
- The Assertion Tool (AT): this tool ensures the validation of the A4Cloud tools through a test case-based validation methodology, during the development and deployment phases.
- The Plug-in for Assessment of Policy Violation (PAPV): This is a plug-in component to A-PPLE that provides an assessment on the criticality of previously detected policy violations. By using it, the cloud actors can check which policy violations are the most relevant ones to their data processed in the cloud service supply chain.

At a high level, accountability functions can be separated into preventive, detective and corrective. Preventive functions focus on mitigating the occurrence of an unauthorized action. In the RA, preventive functions include assessing a risk, identifying and expressing appropriate policies to mitigate it, and enforcing the latter via mechanisms and procedures put in place. Detective functions are used to identify the occurrence of an incident or risk that goes against the policies and procedures in place. In the RA, these centre on monitoring and identifying policy violations via detection and

traceability measures such as audit, tracking, reporting, and monitoring. Finally, corrective functions are those that are used to fix an undesired result that has already occurred. In the RA, these focus on managing incidents, providing notifications and facilitating redress.

Using this categorization, we can separate the A4Cloud tools into five functional areas:

- Contract and Risk Management: the A4Cloud tools in this area aim to address the need for supporting the management of risks and the selection of suitable cloud service contracts in the context of accountability for data governance in the cloud. All tools in this category implement preventive accountability functions.
- Policy Definition and Enforcement: the A4Cloud tools in this area aim to address the functionalities needed for defining and enforcing accountability policies, as well as their maintenance within the data lifecycle of a cloud service provision chain. The tools in this category implement preventive accountability functions.
- Evidence and Validation: the A4Cloud tools in this area deal with the collection and provision of evidence and the validation of the proper execution of the accountability tools in a specific setting. The tools in this category implement both preventive and detective accountability functions.
- Data Subject Controls: the A4Cloud tools in this area target the needs of data subjects by providing controls for the proper management and protection of their personal data in a cloud service ecosystem. The tools in this category implement detective accountability functions.
- Incident Management and Remediation: the A4Cloud tools in this area provide corrective accountability functions facilitating remediation and redress.

Figure 21 illustrates the A4Cloud tools according to this classification. The rest of this chapter presents a detailed description of each A4Cloud tool, examining these functional areas in succession. The description of the tools is structured, so that, for each tool, the following information is provided:

- An overview of the tool, describing its main functions and the envisaged target stakeholders;
- The high level architecture of the tool, identifying its major functional components;
- The description of both the user and machine/application programming interfaces (UI and API) provided, indicating the input and output for each interface and the expected consumers, as well as the main required machine interfaces from other A4Cloud or external tools.

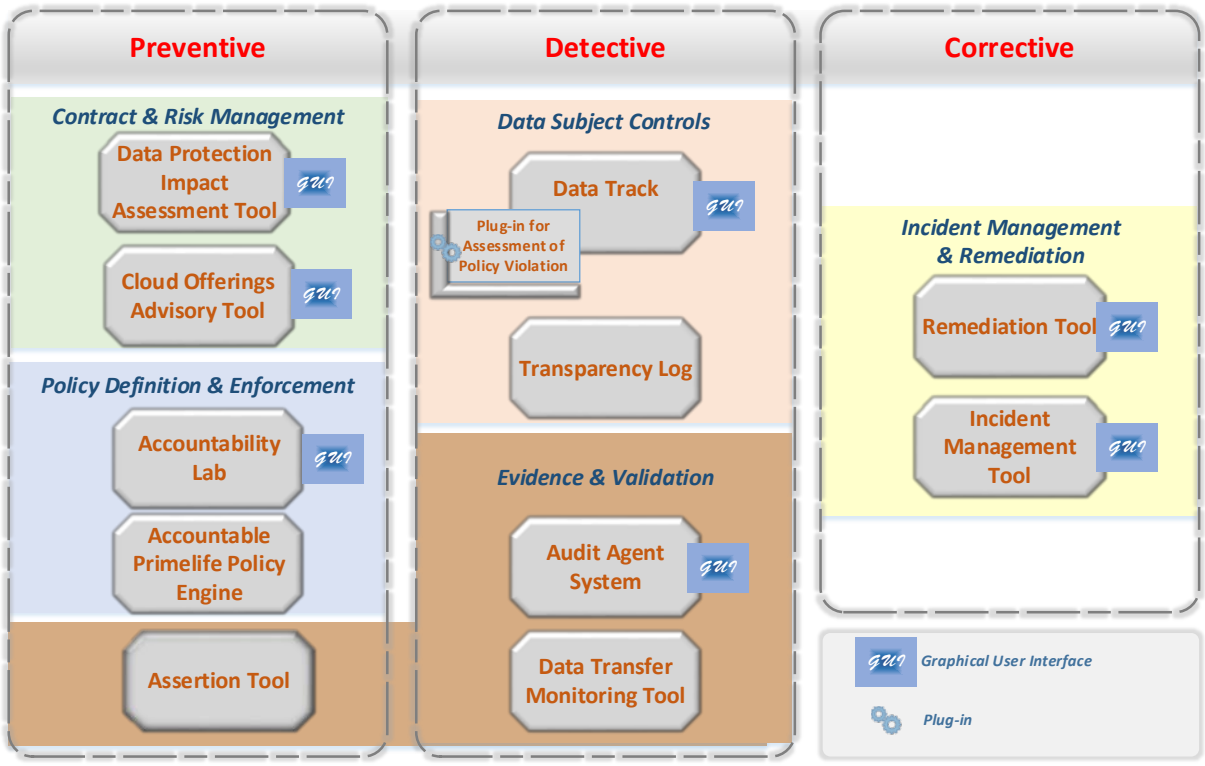

Figure 21: High level view of the A4Cloud Toolset.

5.1 Contract and Risk Management

The contract & risk management area of the A4Cloud toolkit architecture contains tools which address the need for support in managing risk and cloud service contract selection in the context of accountability for data stewardship in the cloud. As a result, tools in this area serve a *preventive* role. Prevention is facilitated via two separate but complementary mechanisms, namely:

i. Assessment of the risks associated with various facets of the cloud service consumption process, involving personal and/or confidential data and elicitation of actionable information and guidance on how to mitigate them;
ii. Evaluation of cloud offerings and contract terms with the goal of enabling a more educated decision making on which service and service provider to select.

For the instantiation of this part of RA, these two mechanisms are being developed as distinct A4Cloud software tools: the Data Protection Impact Assessment Tool (DPIAT) and the Cloud Offerings Advisory Tool (COAT) respectively.

5.1.1 Data Protection Impact Assessment Tool

As mentioned above the DPIAT is used to identify the risks, related to the data protection domain for a given configuration and environment when carrying out a certain business operation, which involves processing of personal or confidential data. The tool is used by Small-Medium Enterprises (SMEs) to assess the classification of the data involved in this business operation (whether they are personal or sensitive data) and how they can be secured and protected in the cloud. Furthermore, DPIAT reports on the risks with respect to data breaches and the privacy of cloud service users and provides insight to the potential threats. Through this tool, the users are guided through questions, getting explanations for both the questions and the respective answers. As a result, the user gets educated on risks and threats to ensure the ethical aspect of accountability.

The output of DPIAT is delivered in two phases. At a first step, DPIAT offers a set of easy-mode questions to enable the cloud customer determine on the requirement to run the expert-mode questions.
The final outcome of DPIAT is a report that includes the data protection risk profile, an advice on whether to proceed or not with the specific business operation (in the context of the given configuration and the given details of the project), and the suggested mitigations in cases of risk exposure. It, also, logs the offered advice and the user's decision for accountability purposes..

The risk profile produced by DPIAT contains a) a set of potential data protection issues and corresponding scores (according to DPIA questionnaire answers) and b) a list of risks associated to the adoption of a given cloud service grouped into 3 categories: service, security and privacy, in case that the user has selected a cloud service provider.

Figure 22 shows the high level architecture of DPIAT.

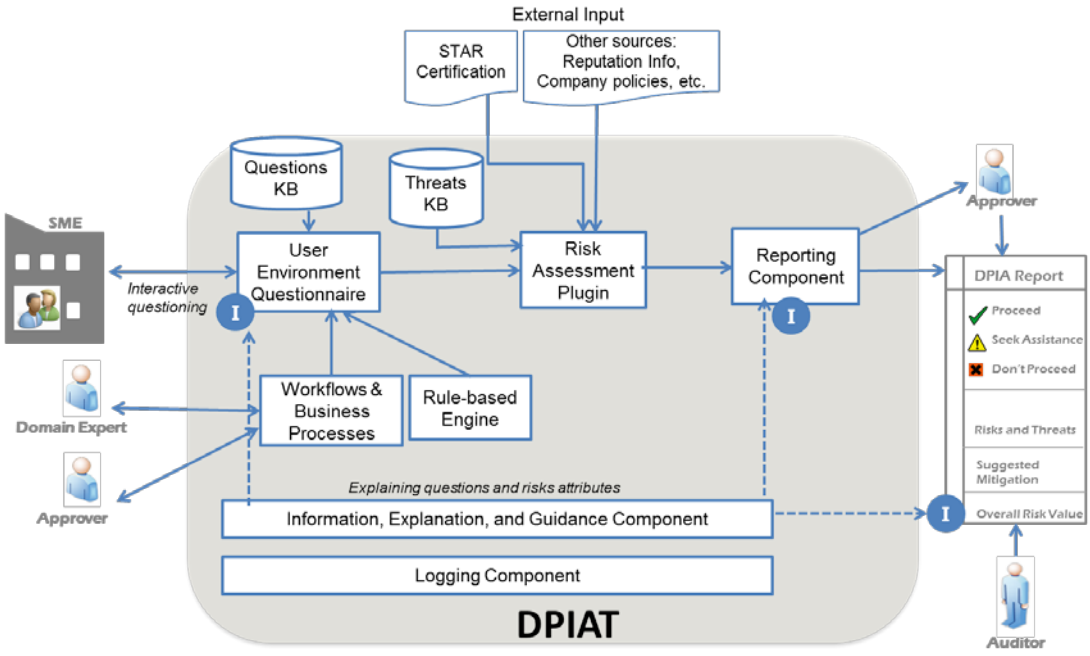

Figure 22: The high level architecture of the Data Protection Impact Assessment Tool

As shown in Figure 22 DPIAT is decomposed to the following components, implementing respective process level mechanisms:

- The User Environment Questionnaire, which offers interaction with the users to collect information about the project (s)he will be involved in the intended transaction (or the type of data that are going to be used in the transaction).
- The Rule-based Engine, which sets up and processes the rules to the path taken when the user answers with specific answers.
- The Risk Assessment Plugin, which assesses the risks associated with the information collected from the User Environment Questionnaire.
- The Reporting Component, which is responsible for presenting the user with a complete assessment report in a comprehensible and user-friendly format.
- The Information and Explanation Component, which provides support on the meaning of each question and the implications of the responses.
- The Logging Component, which logs the values of the answers to the questionnaire and the final report given to him.

These components are fed with user related information (such as the data location, the roles involved in the transaction, contact details of those responsible for defining the purpose of use for the involved data), contextual information on the environment setting, the respective trust and risk modelling, external certification systems (with emphasis for our case the CSA STAR Certification program, which is to manually be adopted), reputation information with respect to the agents involved in the trust modelling, the organisational policies for the protection of the classified data and the knowledge base (KB) of the threats and their associated mitigation control actions.

Through this architecture, the DPIAT will be able to assess a set of use cases, including the support to decision making on competing cloud service offerings, the action on moving data into the cloud (or processes) and the determination on what services in the cloud are the best ones to choose.

DPIAT offers a Web User Interface, which enables interaction with the SMEs. During this interaction, SME representatives can feed the questionnaire with certain input, while the tool gives back a report with three colours (green colour means “Okay to move forward”, yellow colour means “Possible to move forward with issue resolution” and red colour means “Do not move forward - contact Approver”), the relevant threats and risks, and the suggested mitigations/actions and tracking.

DPIAT offers a single interface, named *Idpiat*, which has the following two methods, as shown in Table 10.

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Idpiat / Retrieve Questionnaire	Returns a single instance of a questionnaire	The UI of DPIAT (target all envisaged users of the tool)	JSON	RESTful
Idpiat / List Questionnaires	Returns a list of all available questionnaires (currently, there are two questionnaires available, the pre-screening and the screening ones)	The UI of DPIAT (target all envisaged users of the tool)	JSON	RESTful

Table 10: Interfaces and respective API methods provided by the Data Protection Impact Assessment Tool

Although these interfaces are currently used internally by the UI of DPIAT, it is examined whether they can be public so that they will be consumed by other tools as well in the future, through a web service messaging and transport layer (such as a RESTful service). A detailed analysis of the offered interfaces will be provided in a later stage.

DPIAT consumes the APIs provided by other tools and environments, as shown in Table 11.

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Countries List	Retrieve a list of countries that are available to Service Providers when adding their services. Used as options for the end user to choose from where to select their location	Cloud Providers	JSON	RESTful
Service Types List	Retrieve a list of service types that are available to Service Providers when adding their services. Allows the end user to filter offers by service type when selecting their questionnaire	Cloud Providers	JSON	RESTful
Questionnaire List	Retrieve a list of Questionnaires that can be chosen by the user to complete.	Questionnaire Provider, Certification Authorities	JSON	RESTful

Table 11: Interfaces and methods needed by the Data Protection Impact Assessment Tool

5.1.2 Cloud Offerings Advisory Tool

The purpose of the Cloud Offerings Advisory Tool (COAT) is to assist potential cloud customers (SME organisations and individuals) in assessing and selecting cloud offerings with respect to security and privacy requirements by providing information and guidance on:

- How to understand and assess what a cloud service provider is offering from a privacy and security perspective;
- How to compare offerings (from a data protection compliance and provider accountability point of view)
- The meaning of the comparison attributes.

The tool generates a guided comparison of the cloud service offerings, along with an explanation of the (selected) security and privacy attributes. The tool, also, logs the offered advice and the user's decision for accountability purposes.

COAT is primarily used by cloud customers, both SMEs and individuals. The tool assists such stakeholders in assessing the offerings of a cloud service provider offering, from a privacy and security perspective and compare offerings from various providers, from a data protection compliance and provider accountability point of view. It, also, provides guidance on the meaning of the comparison attributes and the education of users on security aspects, while it implements mechanisms, so that the offered advice and the user's decision are logged.

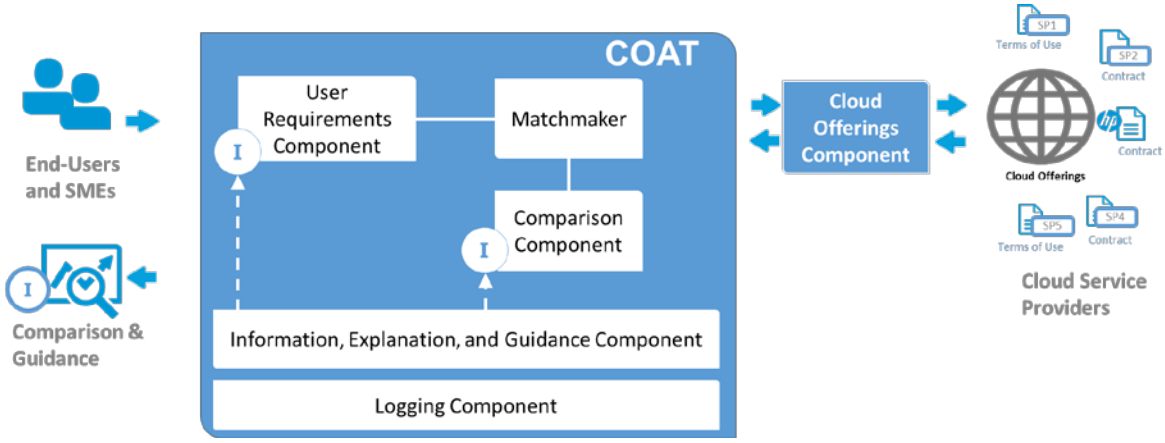

Figure 23: The high level architecture of the Cloud Offerings Advisory Tool

Figure 23 shows the high level architecture of COAT. As shown there, COAT is decomposed to the following main components, implementing respective process level mechanisms:

- The User Requirements Component, which gathers all the explicit (such as user requirements, based on criteria) and implicit (such as user location and contextual information) input for this tool.
- The Logging component, which logs the values of the user selections and the final report with the cloud offerings given to him/her.
- The Information, Explanation and Guidance Component, which supports the guidance of the users on the explanation to each term or criterion used in the comparison process.
- The Matchmaker, which is further split into the Service Matchmaker module (matching the user requirements to service functional features) and the Contract Analysis Component (matching the user requirements to contract offerings).
- The Comparison Component, which produces a report with the best option for each criterion, by facilitating grouping and ranking of the offerings.

The tool gets as input user related information such as the data location, the roles involved in the scenario to be built on the cloud, contact details of those responsible for defining the purpose of use for the involved data), contextual information on the environment setting, the user needs and requirements, the cloud service offerings in structured form, models of cloud contracts and points of attention, reputation information with respect to the agents involved in the offering process and the knowledge base of the threats and their associated mitigation control actions. The outcome of COAT is a report, including guidance on things to pay attention to, when exploring and comparing the terms of service offerings, overview of compatible service offerings, a list of requirements that are communicated to the cloud service providers and relevant guidance for SME-specific security and privacy issues.

Through this architecture, COAT will be able to implement a set of use cases, including the support for the decision making about cloud service offerings and the understanding of the contract terms of these service offerings.

COAT offers a Web User Interface, which enables interaction with the target users. During this interaction, end users that want to be eventually cloud customers can provide as input to the graphical interface a collection of answers to a questionnaire, while the tool gives back a report and a guidance information, comparing the different offers matching the user requirements.

Furthermore, COAT offers a single interface named *Icoat,* which implements the following API methods, as shown in Table 12.

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Icoat/Retrieve Questionnaire	Returns a single instance of a questionnaire	The UI of COAT (target all envisaged users of the tool)	JSON	RESTful
Icoat/Manage Questionnaire	Stores, updates or deletes a single instance of a questionnaire for future reference	The UI of COAT (target all envisaged users of the tool)	JSON	RESTful
Icoat/List Questionnaires	Returns a list of all available questionnaires in the system	The UI of COAT (target all envisaged users of the tool)	JSON	RESTful
Icoat/Matches	Accepts a JSON representation of the MatchCriteria object and returns a list of matching Service offerings	The UI of COAT (target all envisaged users of the tool)	JSON	RESTful
Icoat/SOLRClient	Search for and retrieve records from a SOLR instance based on the match criteria created.	The UI of COAT (target all envisaged users of the tool)	XML /JSON implemented through SOLR structure	RESTful

Table 12: Interfaces and respective methods provided by the Cloud Offerings Advisory Tool

Although these interface methods are currently used internally by the UI of COAT, it is examined whether they can be public so that they will be consumed by other tools as well in the future, through a web service messaging and transport layer (such as a RESTful service). A detailed analysis of the offered interface methods will be provided in a later stage.

COAT consumes the APIs provided by other tools and environments, as shown in Table 13.

Name of the API/methods	*Purpose of use*	*Should be offered by*	*Data format*	*API format*
Countries List	Retrieve a list of countries that are available to Service Providers when adding their services. Used as options for the end user to choose from where to select their location	Cloud Providers	JSON	RESTful
Service Types List	Retrieve a list of service types that are available to Service Providers when adding their services. Allows the end user to filter offers by service type when selecting their questionnaire	Cloud Providers	JSON	RESTful
Questionnaire List	Retrieve a list of Questionnaires that can be chosen by the user to complete	Questionnaire Provider, Certification Authorities	JSON	RESTful
Offer Detail	Shows a detailed view of an offer in html	Cloud Providers	JSON	RESTful

Table 13: Interfaces and methods needed by the Cloud Offerings Advisory Tool

5.2 Policy Definition and Enforcement

This functional area supplements the previous one of Section 5.1 in the preventive role of the A4Cloud tools to support accountability. The set of tools belonging to this category facilitate prevention of the loss of data governance in complex cloud service provision chains through the implementation of mechanisms for the provision and support of enforceable accountability policies. Such mechanisms facilitate the definition, validation, storage, enforcement and overall management of accountability policies.

For the instantiation of the RA, two software tools are provided: AccLab (Accountability Laboratory) and A-PPL Engine (Accountable Primelife Policy Engine).

5.2.1 Accountability Lab

The purpose of the Accountability Lab (AccLab) tool is to bridge the gap between the human-readable, but abstract, accountability obligations expressed in a policy specification language, which in our case is the Abstract Accountability Language (AAL) and the concrete, but machine-readable, accountability policies expressed in Accountable Primelife Policy Language (A-PPL). AccLab provides facilities to write abstract accountability obligations in AAL via a “smart wizard” user interface and (semi-) automatically generate A-PPL policies from them to be enforced by the A-PPL Engine.

This tool is used by the privacy officers of the data controllers to express the accountability policies in an abstract form and the data subjects to express their data-handling preferences, attached to a specific policy. The tool gets as input the abstract textual definition of the policy in AAL format, expressing obligations (for a privacy officer) or the user preferences (for a data subject). The tool internally processes the obligations and identifies the mapping between these obligations to policy terms and rules, in order to generate concrete A-PPL policies for the enforcement of these obligations during the execution of cloud services. Along this process, AccLab checks the AAL statements for consistency and correctness, while compliance checking is applied to determine whether one obligation is stronger than another.

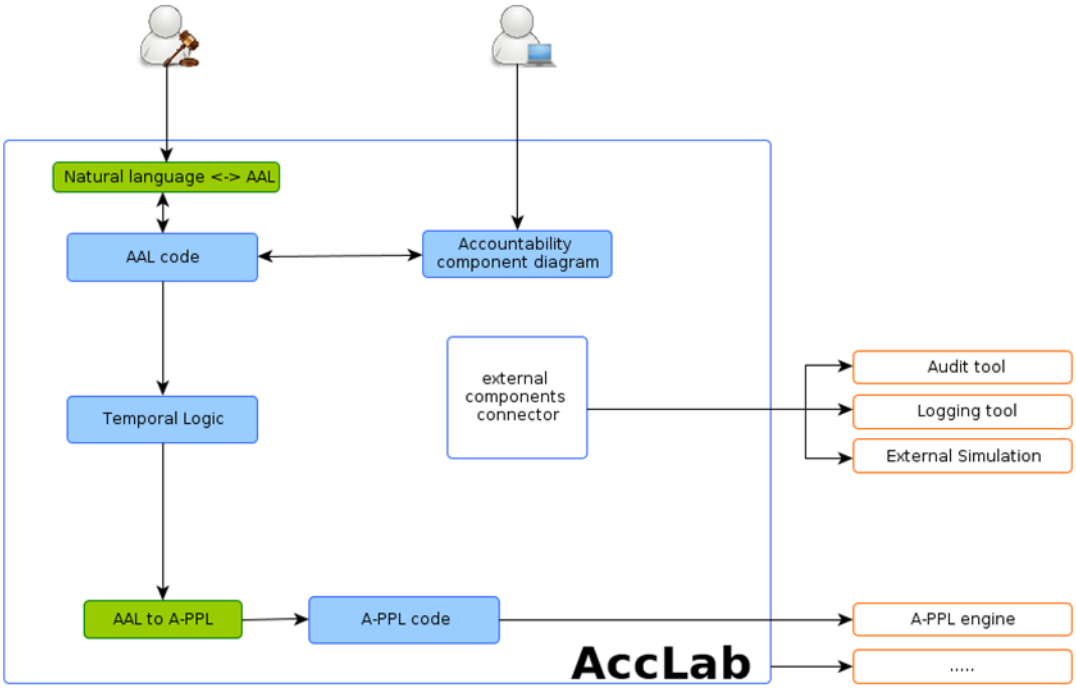

Figure 24: The high level architecture of the Accountability Laboratory tool

Figure 24 shows the high level architecture view of the AccLab. The natural obligations, i.e. textual sentences about laws, regulations and norms for data privacy preferences, are integrated with the components of a system that is designed using an accountability components diagram. The kernel of the AccLab is the AAL language (AAL code) and its semantics (Temporal logic). AAL expressions are interpreted, checked, and mapped into A-PPL code. The second part of the kernel is a connector, which can interact with a set of other tools to enable monitoring, logging, a policy engine connection, or potentially any external auditing.

Through this architecture, the AccLab produces a set of A-PPL policies (i.e. policies which conform to the A-PPL schema) and that can be fed to the policy enforcement environment.
AccLab offers a Web User Interface, which enables interaction with the target users. During this interaction, end users that want to define obligations (for privacy officers) or express privacy and security preferences (for a data subject) can use the tool and especially the UI components shown below:

- The AAL Editor, which is used to write AAL policies with editing facilities. The ALL editor targets data subjects, data controllers, auditors and data processors, who write the policies in plain AAL text and view them in HTML forms.
- The AAL Checker, which is used by data subjects, data controllers, auditors and data processors to parse and check an AAL program. This UI level functionality can also be exposed as an API, as it is shown in Table 14.
- The AAL component designer, which is used by the data controllers and the privacy officers to parse and check an AAL program (provided in JSON)

The AccLab tool offers an API relevant to the functionality exposed by the AAL Checker, as shown in Table 14.

Name of the API	*Purpose of use*	*Consumed by*	*Data format*	*API format*
AAL Parser	Parses an AAL program and performs some checks	The UI of AccLab (target all envisaged users of the tool)	JSON	RESTful

Table 14: Interfaces provided by the Accountability Laboratory

Currently, AccLab does not consume any external API by other tools and environments.

5.2.2 Accountable Primelife Policy Engine

The Accountable Primelife Policy Engine (A-PPL Engine) is an extension of the PPL engine initially designed in the PrimeLife project[28] with additional modules that enable accountability features. The main role of the original PPL engine was to enforce privacy policies related to personal data handling. A-PPL Engine extends the PPL engine with functionality that enables the enforcement of accountability obligations defined in the A-PPL language.

The tool receives the accountability policy in A-PPL format and associates the policy provisions to the piece of personal data disclosed and stored together with the data in a Personally Identifiable Information (PII) repository. This policy is referred to as a Sticky Policy in the A-PPL Engine specification. From the time that this PII is created and onwards, and for as long as a data controller holds a copy of those data, any access requests to the data are regulated by the engine, which enforces all data handling rules and obligations specified in the sticky policy. The A-PPL Engine relies on data transfer logs and evidence repositories populated by third-party tools in order to identify operations on data covered by a sticky policy that occur outside of its reach and ultimately determine whether a policy violation has occurred.

The A-PPL Engine is used by Cloud Providers who are Data Controllers, Data Processors and Cloud Auditors. Both roles use this tool to enforce the accountability policies, defined through an AAL policy specification tool, like the AccLab tool (see 5.2.1). Along with the A-PPL policies, the engine gets as input events (signifying trigger actions as described in obligations), relevant evidence from the environment with respect to actions occurred and data access request. The tool analyses the policies by retrieving access and usage control rules and enforces the relevant obligations described in them, through analysing triggers and actions. It, also, provides configuration of handlers for events monitoring and obligations, collection of evidence (such as the logs created by the Engine) and message exchanges related to event handling and action executions.

[28] http://primelife.ercim.eu/

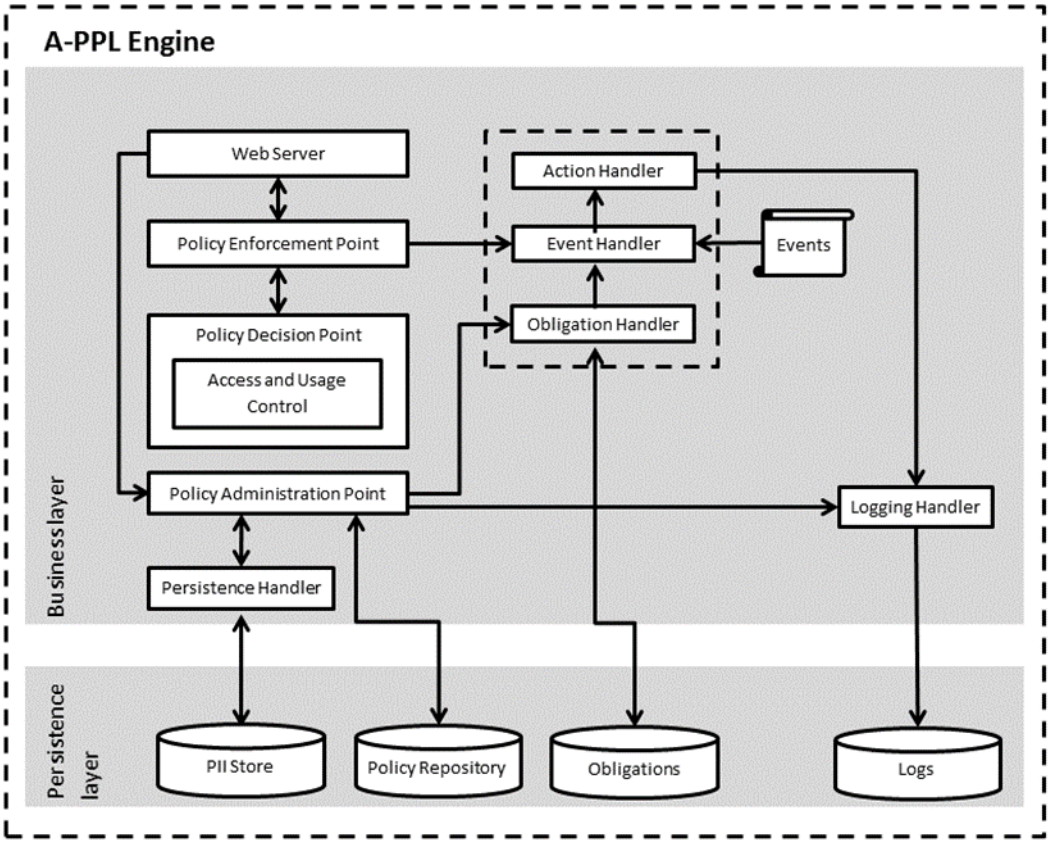

Figure 25: High level architecture of the Accountable Primelife Policy Engine

Figure 25 shows the high level view of the A-PPL Engine. As shown there, the specification of the A-PPL Engine adopts a two-layer high-level architecture to enforce isolation of engine components and data: The core elements of the policy engine providing the enforcement functionality reside in the Business layer. All personal data and their associated sticky policies stored in the PII Store reside in the Persistence layer. Access to the persistence layer is mediated through a Persistence Handler component, which abstracts the underlying location and storage data model to the Business layer functions above. The architecture defines the Policy Decision Point (PDP) as the central element of the A-PPL Engine, responsible for taking access control decisions, with regards to a data access request.

More specifically, the A-PPL Engine consists of the following main components:

- The Policy Decision Point (PDP), which is responsible for taking access control decisions.
- The Policy Enforcement Point (PEP), which the PDP decisions with respect to allowing access to a piece of data. The PEP acts as an orchestrator of the enforcement process, orchestrating two modules, namely the Action and Obligation Handlers.
- The Policy Administration Point (PAP), which records the obligations associated with the PIIs.
- The Obligation Handler, which executes A-PPL obligations related to the personal data handling and is complemented by the Event Handler (serving as the entry point to trigger the events which are responsible for the event based triggers) and the Action Handler (facilitating actual action execution)
- The Logging Handler, which provides an extensible secure logging interface to support all logging related functionality during the policy enforcement and personal data handling process.
- The Evidence Handler, which is responsible for the compilation of evidence, based on engine operations.
- The Audit Handler, which provides a central component for handling audit requests by facilitating the process of retrieving the necessary information from the various sub-components.

These components are complemented by the Policy Repository, which provides storage for generic A-PPL policies that the Engine will retrieve and possibly modify based on the context of the particular data disclosure before turning into a "sticky policy". It is assumed that the Policy Repository is separate from the PII Store.

Through this architecture, the tool generates evidence that are captured and/or created by the engine itself (evidence can include logs, messages, events, etc.). An audit report can also be produced to facilitate requests from cloud auditors.

- The A-PPL engine provides customised APIs, which enable interaction with the operations of the business applications. During these interactions, end users can indirectly access the relevant API calls offered by the A-PPL Engine and the respective *IAppEngine* Interface, as shown in Table 15.

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
IAppEngine / storePii	Stores PII inside the PII repository, together with attached A-PPL policy that defines how the data can be processed	The application (for the Data Controller)	XML-based A-PPL specification	RESTful
IAppEngine / deletePii	Deletes PII from the PII repository	The application (for the Data Controller)	JSON	RESTful
IAppEngine / getPii	Retrieves specific PII given a set of attributes	The application (for the Data Controller)	JSON	RESTful
IAppEngine / getAllPii	Retrieves all PII given a specific owner	The application (for the Data Controller to be finalised exposed to Data Subjects through DT)	JSON	RESTful
IAppEngine / updatePii	Updates/corrects specific PII's value given a set of attributes	The application (for the Data Controller/Data Subject)	JSON	RESTful
IAppEngine / deleteAllPii	Deletes all PII of a user from the PII store	The application (for the Data Controller/Data Subject)	JSON	RESTful
IAppEngine / storePolicy	Stores a policy template	The application (for the Data Controller/ potentially for Data Subject)	XML-based A-PPL specification	RESTful
IAppEngine / getPolicy	Retrieves a policy template	The application (for the Data Controller/ potentially for Data Subject)	XML-based A-PPL specification	RESTful
IAppEngine / deletePolicy	Deletes a policy template	The application (for the Data Controller)	JSON	RESTful
IAppEngine / requestPii	Used for downstream usage. A third party	The application (for the Data	XML-based A-PPL specification	RESTful

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
	Data Processor can request access to a piece of personal data.	Processor)		
IAppEngine / triggerUserRegistration	A-PPLE is notified when personal data of a data subject is collected for the first time	The application (for the Data Controller)	JSON	RESTful
IAppEngine / triggerPolicyViolation	A-PPLE is notified when policy violation is detected by DTMT	DTMT and AAS	JSON	RESTful

Table 15: Interfaces and the respective methods provided by the Accountable Primelife Policy Engine

A-PPL Engine consumes the APIs provided by other tools and environments, as shown in Table 16.

Name of the API	*Purpose of use*	*Should be offered by*	*Data format*	*API format*
Encrypt Data	The API of TL sender module of Transparency log	TL	JSON	RESTful
Send Incidents	This API is used to communicate the incidents detected from the evidence tools and the A-PPL Engine to the relevant stakeholders of the cloud providers	IMT	JSON	RESTful
IEvidence	This API is used by the A-PPL engine to send the evidence corresponding to logs collected by the Engine	AAS	JSON	RESTful

Table 16: Interfaces and methods needed by the Accountable Primelife Policy Engine

5.3 Evidence and Validation

The Evidence and Validation functional area offers accountability support in both a preventive and detective manner. The set of tools belonging to this category facilitate the assertion over the prevention mechanisms against the loss of data governance in complex cloud service provision chains and the monitoring mechanisms of the appropriate software resources to control and verify the accountability policy-based operation of these chains. More specifically, the tools in the Evidence and Validation area fulfil two main purposes:

- They provide means for the audit of cloud applications and infrastructures during their execution. This objective principally requires that appropriate auditing policies can be defined, data be monitored during execution and evidence be stored, and that auditing properties can be verified at appropriate times during execution.
- They enable the validation of A4Cloud tools. Properties to be validated include, for example, that actions of one tool do not invalidate hypotheses needed for the application of another and that information is passed correctly between tools.

The respective mechanisms to meet these objectives are being developed within the scope of the A4Cloud software tools, namely the Audit Agent System (AAS), the Data transfer Monitoring Tool (DTMT) and the Assertion Tool.

5.3.1 Audit Agent System

The Audit Agent System (AAS) tool enables the automated audit of multi-tenant and multi-layer cloud applications and cloud infrastructures for compliance with custom-defined policies, using software agents. Agents can be deployed at different cloud architectural layers (i.e., network, host, hypervisor, IaaS, PaaS, SaaS) with the purpose of i) collecting and processing evidence, ii) generating audit reports and iii) aggregating new evidence. AAS uses audit tasks that specify the data collection sources and tools to use to collect data, and policies to specify the thresholds and constraints, against which the evidence is examined to generate the audit results.

An audit component is central to any approach to accountability. Remediation actions that have to be performed after some policy has been violated, for instance, often rely on fine-grained monitoring facilities and extensive analysis capabilities of the resulting evidence. The AAS tool provides suitable means for the runtime monitoring of cloud applications and infrastructures, the verification of audit policies against the collected data, and the reporting of policy violations along with the evidence supporting it.
AAS is used by auditors, who may act on behalf of the cloud customer/data subject (external view) or the cloud provider (internal view) to perform continuous and periodic audits. The goals and nature of policies which are audited may differ depending on the view. The view may also differ in case of a trusted third-party auditor (TPA), who is independent from the customer and provider, but acting on behalf of any of those.

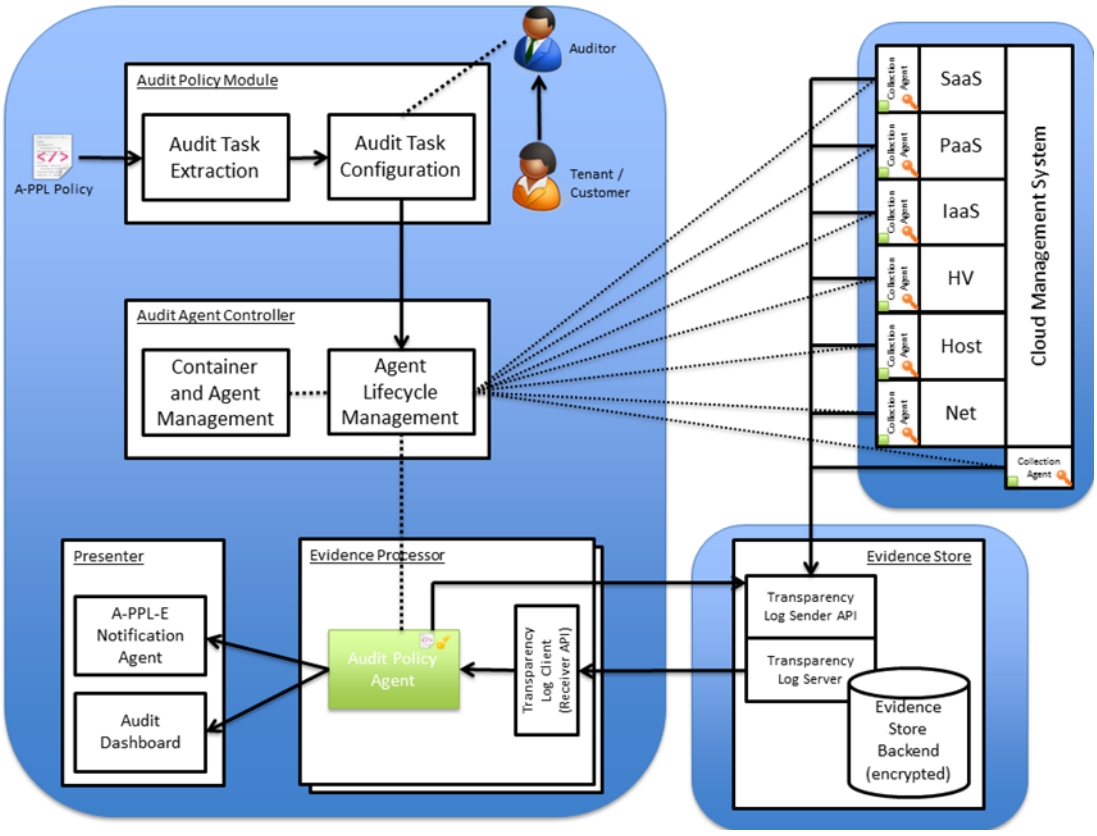

Figure 26: The high level architecture of the Audit Agent System tool

Figure 26 shows the high level view of the AAS architecture, with the following components:

- Audit Policy Module (APM): The APM is used by the auditor to define audit policies. On the basis of these policies (describing the goal of what needs to be checked - tasks which shall be performed during audits and thresholds for compliance and failure respectively), a suitable set of audit tasks is selected by the Auditor and configured properly. These Audit Tasks are processed by the AAC. The APM is located at the Primary Service Provider (PSP).

- Audit Agent Controller (AAC): The AAC prepares audit agents according to the audit task description and manages their life-cycle from configuration, deployment to deletion. Evidence collected by agents is stored in the evidence store.
- Evidence Sources: The cloud stack consists of several architectural layers, on which evidence can be collected (i.e., network, host, hypervisor, IaaS, PaaS, SaaS and the cloud management system). For each evidence source (e.g., log, CMS API etc.), a specialized Collection Agent is implemented.
- Evidence Store (ES): The Evidence Store is used to store data collected by the agents for audit processing and report generation by the Evidence Processor and Presenter which logically groups audit evaluation and reporting agents. The Evidence Store is encrypted, located at the PSP and there is isolation between tenants. The Evidence Store is implemented using Transparency Log. The TL recipient is the evaluation agent. Therefore, the evaluation agent can decrypt records collected for it. Every tenant has its own TL pile.
- Evidence Processor & Presenter (EPP): The Evidence Processor & Presenter is a logical component running audit evaluation and reporting agents. Audit results are generated using evidence stored in the ES (after verifying its integrity). The results are presented to the auditor in the form of reports and a web-based dashboard. Additionally, supporting evidence records can be requested via the dashboard.

The tool receives as input the policy to monitor and an audit task description (describing the data collection sources and tools to use to collect data). Both these descriptions are stored in a machine-readable format. The specification of audit tasks is done by an auditor using a Web-based User Interface. A cloud provider may also provide several pre-defined audit tasks to choose from and customise.

AAS takes advantage of this input to configure the software agents specific to the collection tool, which needs to be used, and deploys them in agent runtime environments, located at different cloud architectural layers. Which agents are being deployed and therefore which data are being collected depends on the audit task. The requirements expressed as policy terms are evaluated against the collected data (which is the evidence). The evaluation results are used to generate audit reports containing audit results and supporting evidence. For evaluation of the cloud service delivery chains, intermediate results are passed out for further evaluation.

The output of the AAS tool is a report containing compliance status and supporting proof/evidence for the compliance claim. Additional output may also be (aggregated) evidence from the data gathered during the audit process (e.g., output of intermediate results for further evaluation). The report is presented to the auditor in the form of Web dashboard. Furthermore, AAS provides policy violation reporting.

AAS provides a web-dashboard, which enables interaction with the target users for easy management of the platform, definition of new audit policies and presentation of audit results-. During this interaction, the auditors can use the dashboard to configure (e.g., data collection tool configuration) and control the software agents (e.g., agent lifecycle) and the audit process, as well as to check the compliance status.

The interaction of the auditors, as the end users of the AAS tool and the tool itself, is based on a set of UI functions that are the outcome of relevant API calls offered by the AAS tool, as shown in Table 17. These interfaces can be broadly categorised into being audit/AAS-specific and evidence-specific. Evidence-specific (*IEvidence* Interface) is closely related to functionality described in the Framework of Evidence (please refer to the work in WP:C-8 for more details) and the collection and storage of evidence, whereas audit/AAS-specific (*IAudit* Interface) provides an audit-focused layer on top of the Framework of Evidence functionality, which also includes the definition and evaluation of audit policies as well as the presentation of compliance results.

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
		IAudit		
SetAuditTask	Definition of audit task to be performed for checking compliance with audit policy	The UI of the AAS (for Auditor)	JSON	RESTful
RequestAuditReport	Requests the audit results of a single or multiple audit tasks	The UI of the AAS (for Auditor, Cloud Provider, Cloud Customer)	JSON	RESTful
RequestRunningContainers	Requests the running JADE agent containers (core and clients) inside an AAS platform	The UI of the AAS (for Auditor)	JSON	RESTful
RequestRunningAgentsByContainer	Requests the agents running in a given container	The UI of the AAS (for Auditor)	JSON	RESTful
RequestRunningAgents	Requests all agents running on the platform (i.e., all containers)	The UI of the AAS (for Auditor)	JSON	RESTful
RequestRunningAgentsByAuditTask	Requests the running agents for a given audit task	The UI of the AAS (for Auditor)	JSON	RESTful
RequestRunningAuditTasks	Requests the currently executed audit tasks	The UI of the AAS (for Auditor)	JSON	RESTful
RequestAuditTaskConfiguration	Requests the configuration for a given audit task	The UI of the AAS (for Auditor)	JSON	RESTful
ExtractFromPolicy	Set the policy file to extract the audit tasks	The UI of the AAS (for Auditor)	XML	RESTful
		IEvidence		
RequestEvidence	Requests a record, or a chain of records respectively, from the evidence repository, given an identifier of the record	The UI of the AAS (for Auditor)	XML	RESTful
StoreEvidence	It enables the A4Cloud tools to store their evidence in the AAS repository to facilitate for audits	A-PPLE	JSON	RESTful

Table 17: Interfaces and respective API methods provided by the Audit Agent System

Furthermore, AAS consumes the APIs provided by other tools and environments, as shown in Table 18.

Name of the API	*Purpose of use*	*Should be offered by*	*Data format*	*API format*
Dispatching of policy violation alerts	Forwarding of policy violations to reach stakeholders and notify them about violation	A-PPL Engine	JSON	RESTful

Table 18: Interfaces and methods needed by the Audit Agent System

5.3.2 Data Transfer Monitoring Tool

The Data Transfer Monitoring Tool (DTMT) aims to enable cloud service providers to demonstrate compliance with personal data protection and other regulations regarding where and by whom the processing occurs. The tool automates the collection of evidence about how data transfers comply with the obligations, concerning personal data handling procedures, and are being carried out properly, by generating logs regarding the operations performed on personal data involving transfer at the infrastructure level (for example when performing load-balancing or creating backups and storing them in different hosting machines). It, then, analyses these logs using rules, helping an auditor to identify whether all transfers were compliant with the authorisations from the Data Protection Authority being obtained beforehand by the Data Controller. In this way, policy violations can be identified.

Part of the necessary information to monitor data transfers can be obtained from machine-readable policies specifying accountability obligations, such as A-PPL policies. Furthermore, the tool enables manual configuration for constraints about which parties, and geographical locations are allowed for a given data set.

Upon identification of a potential violation (by the tool), further data can be collected to provide supporting evidence of an incident. Notifications can be triggered by the auditor. The tool relies on the A-PPL Engine to carry out actions for policy enforcement related to the events it generates and to other tools to act upon the detected violations (e.g. for notification or remediation).

The DTMT is used by the privacy officers of data controllers and the internal and external auditors hired by cloud providers and/or customers. The tool is configured with information for the geographical location of host machines in order to determine the current physical location of the data. This information is not part of the A-PPL policy statement, but must be provided by the data processors (such as IaaS providers), so that the monitoring part of the DTMT to work properly.

The data controller needs to feed the DTMT with the authorisations made to data processors and other third parties that handle personal data. A configuration file containing these facts is filled by the data controller with information obtained from the data processors (who can further delegate the processing to other parties with the prior permission of the data controller and consent from the data subjects). Thus, if there is a transfer of the personal data to a party outside this list, it is deemed as a violation of privacy obligations. Furthermore, DTMT is fed with queries related to data location.

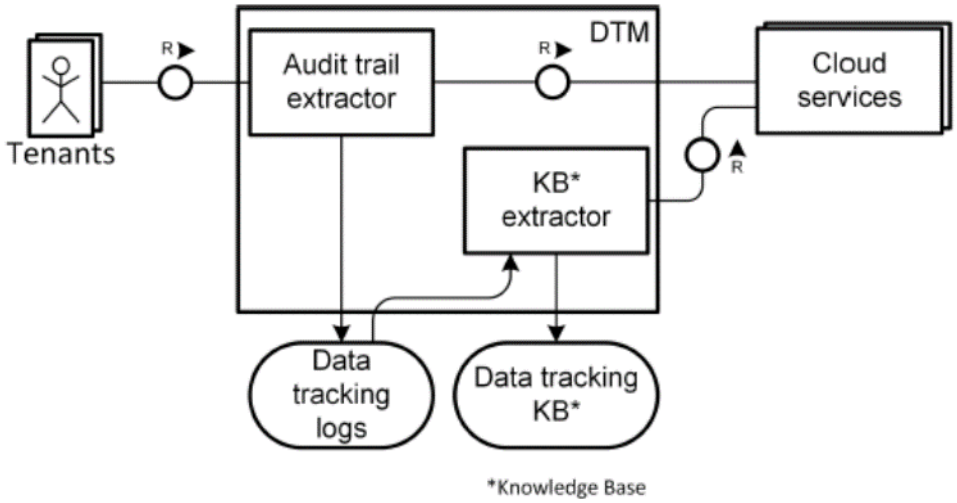

Figure 27: The high level architecture of the Data Transfer Monitoring Tool

Figure 27 shows the high level view the DTMT internal architecture. The tool continuously logs any activity on the cloud infrastructure that implies data transfers. It, then, performs inference operations on the logs to identify whether transfers were compliant to the defined A-PPL policies and/or constraints manually configured in the tool. Thus, DTMT is able to answer specific queries about the location of the processing and the parties involved in the processing for the given data set, by producing logs related to data transfers.

As shown in Figure 27, the DTMT tools is mainly composed of two main parts. Initially, a proxy module is used to monitor the machine interface calls from different tenants (e.g. data controllers or cloud platform administrators) to the cloud services and extract the audit trail. The latter is exploited to construct a data tracking knowledge base that represents operations on personal data as logical facts, suitable for analysis. The knowledge base involves an inference engine that can assert where a given data set is located in the virtualised environment. It can also answer to audit queries that help an auditor to check whether previous data transfers where compliant.

Currently, the DTMT tool is designed as a standalone tool, which is accessible through command line to run queries over the collected logs. However, the tool will be evolved to interface with the A-PPL Engine to trigger notifications to the users, to collect logs about data transfers at the upper layer (SaaS or PaaS), and to obtain relevant information from the accountability policy in place, about the authorised transfers.

In that respect, the DTMT tool will implement the *Idtmt* Interface as shown in Table 19.

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Idtmt				
fc_check_data_location	It returns all locations where the entered personal data reference are and were stored	DTMT users and potentially by A-PPL Engine	XML	RESTful
fc_check_instances	It returns the infrastructure resources where the entered personal data reference are and were stored	DTMT users and potentially by A-PPL Engine	XML	RESTful
fc_check_processors	It returns all parties that processed a given data set	DTMT users and potentially by A-PPL	XML	RESTful

Name of the API/method	*Purpose of use*	*Consumed by*	*Data format*	*API format*
		Engine		
fc_check_violations	It returns all unauthorized transfers (to location, or to party) detected	DTMT users and potentially by A-PPL Engine	XML	RESTful
show_authorized_transfers	This function simply shows what are the authorizations set for a data set	DTMT users and potentially by A-PPL Engine	XML	RESTful

Table 19: Interface and the respective methods provided by the Data Transfer Monitoring Tool

Furthermore, DTMT consumes the APIs provided by other tools and environments, as shown in Table 20.

Name of the API	*Purpose of use*	*Should be offered by*	*Data format*	*API format*
Dispatching of policy violation alerts	Forwarding of potential policy violations with respect to data transfers	A-PPL Engine	JSON	RESTful

Table 20: Interfaces and methods needed by the Data Transfer Monitoring Tool

5.3.3 Assertion Tool

The Assertion Tool (AT) is part of the Assertion Framework, a framework that provides assurance to the cloud service provider of the valid combination of the A4Cloud tools. Within the scope of the Assertion Framework, AT ensures the validation of the A4Cloud tools, using a test case-based validation methodology that tests the interactions held among two or more tools.

The assertion tool ensures the validation of the A4Cloud tools through a test case-based validation methodology. This validation might involve data coming from other tools (e.g. collecting logs through AAS). AT is responsible for gathering those data and their application on the validating scenario. Because AT is an aspect oriented programming-based tool, it could also be used to extract, analyse and transmit data that are not directly accessible from other tools.

The validation of the A4Cloud tools takes place before their delivery to the tools' users. In that respect, AT offers specific functionalities for the validation by making specific functionalities available for the A4Cloud developers, prior to the final release of the tools. This means that the AT may not be available for non A4Cloud users.

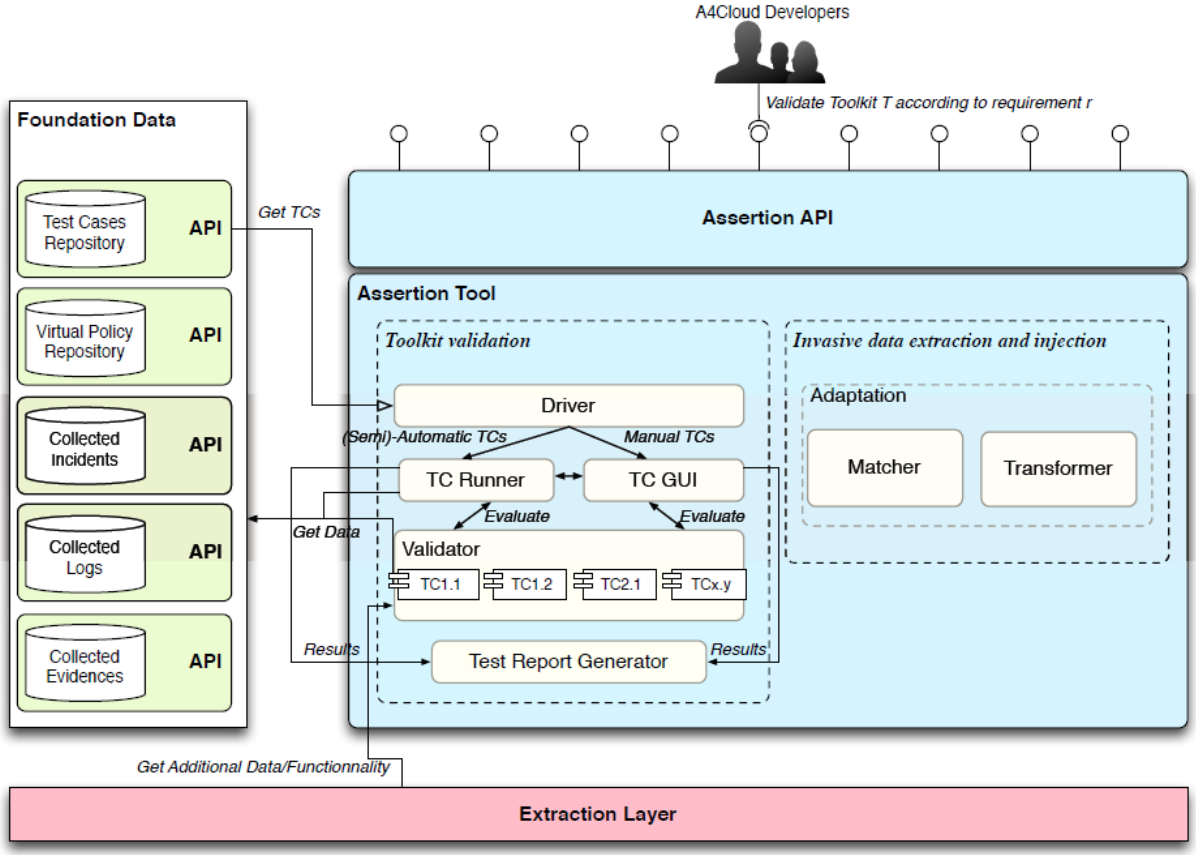

Figure 28: The high level architecture of the Assertion Tool

Figure 28 shows the high level view of the AT architecture. The tool gets as input a combination of the A4Cloud, a set of accountability requirements to be validated and a set of associated test-cases through which validation will be performed. Then, the AT automatically executes the test cases on the selected A4Cloud tools to accomplish the validation tasks. The well execution of these test cases may require direct information coming from the tools (e.g., logs, audit reports, ...) or information that might be not directly offered by the tools (e.g., timestamp whenever missing). To get this additional information AT uses Aspect Oriented Programming techniques, which enable invasive transformations at specific points in the control flow of the service application. The outcome of this tool is a report stating whether or not the tool combination is accountable. In case that the result shows that the combination is not accountable, the report contains requirements that have not been validated.

AT partially supports a Web User Interface, which enables interaction with the target users, in the sense that the developers can work with two Domain Specific Language (DSLs) for adapter definitions. The first DSL is a protocol based language for the description of (automata of) events. It enables writing queries that specify where a modification has to take place. The second DSL is used for the definition of modifications. The DSL is based on Java and enables the description of extraction of information and the modification of service calls and implementations.

Table 21 shows the APIs provided by the AT.

Name of the API	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Assertion API	Test an Assertion for a given toolkit. Instantiates an evaluation for a given toolkit over an accountability attribute.	The UI of the AT (for A4Cloud developers)	-	RESTful

Table 21: Interfaces provided by the Assertion Tool

Furthermore, AT consumes the APIs provided by other tools and environments, as shown in Table 22.

Name of the API	*Purpose of use*	*Should be offered by*	*Data format*	*API format*
Extraction API	Operations to get or generate the logs, evidences, incidents and policies associated to a given tool	A4Cloud tools	JSON	RESTful
Foundation Data API	Allows access to data (test cases, policies, incidents, logs and evidences) needed for an A4Cloud tools validation	Ideally by the repositories provided by the A4Cloud tools	JSON	RESTful

Table 22: Interfaces needed by the Assertion Tool

5.4 Data Subject Controls

The Data Subject Controls functional area offers accountability support in a detective manner. The set of tools belonging to this category facilitate the need of the data subjects (i.e. individuals whose personal data are collected and/or processed by cloud service providers) to verify the correct treatment of their data. The tools implement runtime mechanisms to follow the proper execution of data handling accountability policies and provide data subjects with notifications on how their data are exploited along cloud service chains.

In the context of the A4Cloud architecture, these mechanisms are being developed within the scope of the A4Cloud software tools, namely the Data Track and the Transparency Log. We also include in this section a separate description of the Plug-in for Assessment of Policy Violation, but this plug-in is considered to be an integral part of the Data Track tool.

5.4.1 Data Track

Data Track (DT) is a tool used by data subjects to get an overview of all personal data they have disclosed. The tool allows them search through their history of data disclosures to see what personal data they have disclosed, to whom they have disclosed these data to (i.e. which data controller), and under which privacy policy. Furthermore, the DT tool enables users to directly assert their right to access and rectify the data concerning them held by the data controller. The tool can be used by both active and passive data subjects.

Specifically, DT enables data subjects to:

- Request access to their data stored at a data controller;
- Request to correct their data;
- Request that their data should be deleted or blocked;
- View detailed information about how the data has been shared and used by the data controller;
- View prior data disclosures and associated metadata like privacy policy, time, etc.;
- Detect policy violations by matching log data;
- Seek redress (via handing-over to relevant tool).

DT is consumed by data subjects, through a front end visualisation module. It gets as input the user's data disclosures, the data handling policy, describing relevant operations and obligations for the particular data disclosure, the credentials to remotely access user data at the cloud service providers'

side, and the notifications received from the providers. DT uses an indexer to index and categorise all data disclosures. Upon a request from the user, DT remotely accesses the user's data located at different providers to gather the additional personal data collected by the service provider about the user. The tool can also support correcting and requesting deletion of personal data at cloud service providers. Last, but not least, DT analyses log data about how personal data has been processed and notifications to detect policy violations.

The output of the tool is a list of data disclosures for both explicitly disclosed personal data and implicitly collected data by service providers, potentially stored remotely, presented via different visualisations. Users are also presented with notifications from service providers, for example to inform them about policy violations or privacy policy updates from the service provider.

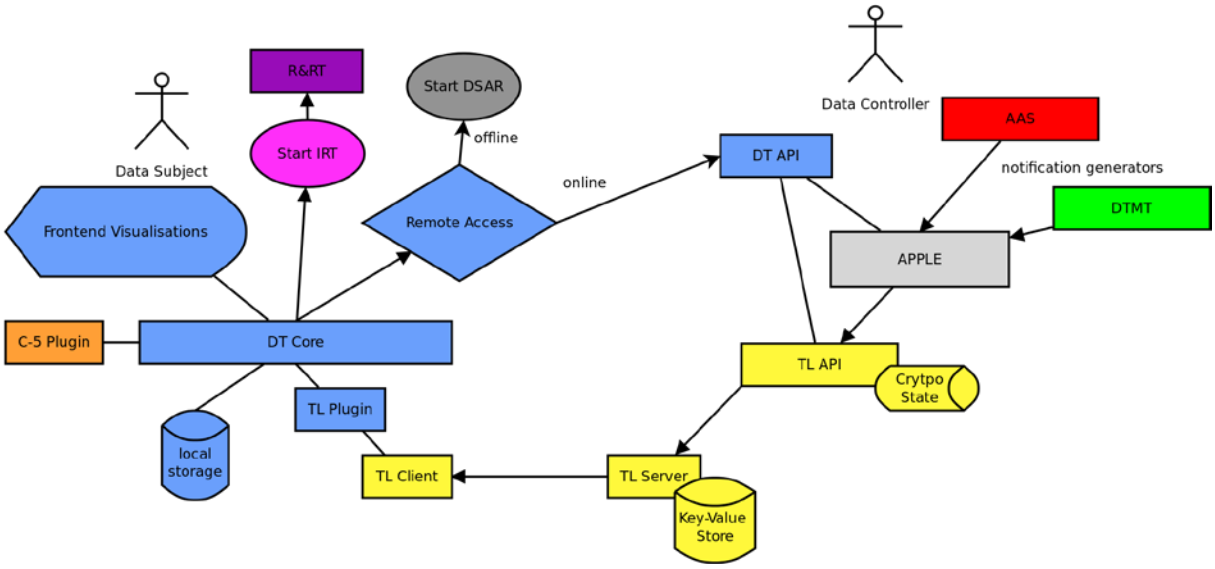

Figure 29: High level view of the Data Track and Transparency tools

Figure 29 illustrates the DT tool and the interactions with other A4Cloud tools. The tool consists of the blue highlighted functional boxes. The core component of the Data Track is the DT Core. The DT Core acts as a backend with local storage and support for plugins that get data from remote sources. The core provides access to data from its plugins in a normalised form to the Frontend Visualisations. An example of such a plug-in is shown in Figure 29 for the interaction with the Transparency Log (TL) tool, which provides notifications from the A-PPL engine (and potentially also logging data) to DT. Possible generators of notifications are indirectly the AAS and the DTMT. It must be noted that we consider to integrate the data track logic assigned to the data controller into the A-PPL Engine, so that DT is only installed on the data subject side.
In case of notifications concerning incidents or violations, the DT Core is using a policy violation plug-in to assess the severity of these notifications. The severity will impact how the notifications are presented to the data subject by the Frontend Visualisations component. For example, a high severity may cause a notification to the data subject as soon as the tool is launched, while less severe notifications are only shown on request. If the data subject wishes to act on a notification concerning an incident or a violation, the DT Core can launch the Incident Response Tool.

If the data subject wishes to perform actions that involve remote access to a cloud service provider (i.e. a data controller), there are two cases:

- In case of online access being possible, the DT Core uses the *DT API* component running at the data controller. The DT API interacts with A-PPL Engine (and potentially other components at the data controller) to accomplish the remote access request.
- When online case is not possible, the DT Core can start the Data Subject Access Request tool.

DT offers a Web User Interface to serve the front end visualisation module. This UI provides different visualisations of the user's data disclosures. One of the visualisations shows traces indicating what attributes a user has disclosed to which providers, by visually linking cloud service providers and

attributes. Another visualisation shows a timeline of the user's data disclosures. The users will have the possibility to filter out displayed services or attributes.

Furthermore, DT provides a set of API methods that are offered by the *Idatatrack* Interface and can be consumed both by the UI of the DT and other A4Cloud tools. These interface methods provided by DT are shown in Table 23.

Name of the API	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Idatatrack / SetupTL	Setup the data subject at the Transparency Log	The UI of the DT (for data subjects)	JSON	RESTful
Idatatrack / Authenticate	Authenticate a data subject using the Transparency Log	The UI of the DT (for data subjects)	JSON	RESTful
Idatatrack / GetPersonalData	Get all data associated to the authenticated data subject	The UI of the DT (for data subjects)	JSON	RESTful
Idatatrack / RequestDataCorrection	Requests that a provided attribute should be corrected (changed) to the provided value	The UI of the DT (for data subjects)	JSON	RESTful
Idatatrack / RequestDataDeletion	Requests that all data associated to the authenticated data subject should be deleted	The UI of the DT (for data subjects)	JSON	RESTful

Table 23: Interfaces and respective methods provided by the Data Track tool

In order to better understand the APIs exposed by the DT tool and since this tool is strongly coupled with the TL tool, we present the following scenario to explain the use of the APIs. For an active data subject, the data subject generates his or her own asymmetric key-pair. For a passive data subject, the data controller generates the key and stores all key material for the data subject (with the obvious loss of for example confidentiality if you do not trust the data controller). Regardless, we assume the following:

- The data subject has a public key, as defined by Curve25519[29].
- The data controller has an identifier for the data subject. This may be an account (like alice@example.com) or a transaction pseudonym (like a session cookie).

Given a data subject's public key and an identifier, the data controller uses the API method *SetupTL* to setup the Transparency Log. In case of an active data subject, the reply is returned directly to the data subject. In case of a passive data subject, the reply is stored by the data controller. After setup, the active data subject (or passive data subject that retrieved their key material) has the possibility to authenticate using the API method *Authenticate*. In A4Cloud, we use the TL key to authenticate. The reply from a successful *Authenticate* is an authentication key used to authenticate calls to the other API methods. The authenticated API methods are:

- *GetPersonalData* – returns all data associated with the authenticated data subject. Associated metadata should describe the data.
- *RequestDataCorrection* – requests that a provided attribute should be corrected (changed) to the provided value.
- *RequestDataDeletion* –requests that all data associated with the authenticated data subject should be deleted.

Furthermore, DT consumes the APIs provided by other tools and environments, as shown in Table 24.

[29] http://cr.yp.to/ecdh.html

Name of the API	*Purpose of use*	*Should be offered by*	*Data format*	*API format*
TL Client	Retrieve all data sent to the data subject through the Transparency Log. Also provides access to other Transparency Log functionality such as cryptographic proofs of authenticity, time, inconsistency, etc.	Transparency Log	Plain text for data (that is, retrieved data is in the same format as written), JSON or Google Protocol Buffers for cryptographic proofs	RESTful
Start IRT	Enable a data subject to start the Incident Response Tool for assistance with responding to a particular incident	Incident Response Tool	-	RESTful
Policy Violation Plugin	Enable the data subject to assess the severity of one or more incidents	Plug-in for Assessment of Policy Violation	-	Might be integrated with DT
TL API	Enable the DT API to interact with the Transparency Log to authenticate data subjects	Transparency Log	Google Protocol Buffers (internal API between DT and TL), that is base64 encoded in JSON (to fit the generic external DT API)	RESTful
A-PPL API	Enable the DT API to interact with A-PPL Engine for API calls GetPersonalData, RequestDataCorrection, and RequestDataDeletion	A-PPL Engine	-	RESTful

Table 24: Interfaces and methods needed by the Data Track tool

5.4.2 Plug-in for Assessment of Policy Violation

The Plug-in for Assessment of Policy Violation (PAPV) provides an assessment on the relevance of previously detected policy violations. By using it, data subjects (or their representatives) can check which policy violations are the most relevant ones to the data subject itself.

PAPV is used by data subjects to assess policy violations. The plug-in gets as input a collection of instances of the policy violations from the DT tool, where an instance of a policy violation is any piece of evidence that describes an occurrence of a policy violation event, and a machine-readable policy description, detailing the obligations of the data controller, regarding the data handling procedures treatment for the personal data of the data subjects, which will serve as a reference for evaluating the detected violations. The plug-in can also be fed with a document describing the data subject's preferences with respect to the treatment of their data, which will guide the assessment.

By receiving a list of the latest policy violations, together with their associated policies, PAPV produces an assessment for the relevance of each reported violation. It, then, prepares an ordered list of policy

violations, sorts them by the level of importance. The output of the plug-in is an ordered measurement (either qualitative or qualitative) of the relevance of the violation event, for each instance of the policy violation. This enables the list of policy violations to be sorted with respect to their relevance.

Figure 30 shows the high level architecture of PAPV.

PAPV does not support any UI, but the functionalities provided by it are realised through the DT UI.

The plug-in will be integrated with the DT tool development and there is no need to consume any API provided by other tools. However, the API shown in Table 25 will be offered by PAPV.

Name of the API	*Purpose of use*	*Consumed by*	*Data format*	*API format*
IPapv	Asses the relevance of the provided policy violation	Data Track	XML	Go library

Table 25: Interfaces provided by the plug-in for Assessment of Policy Violation

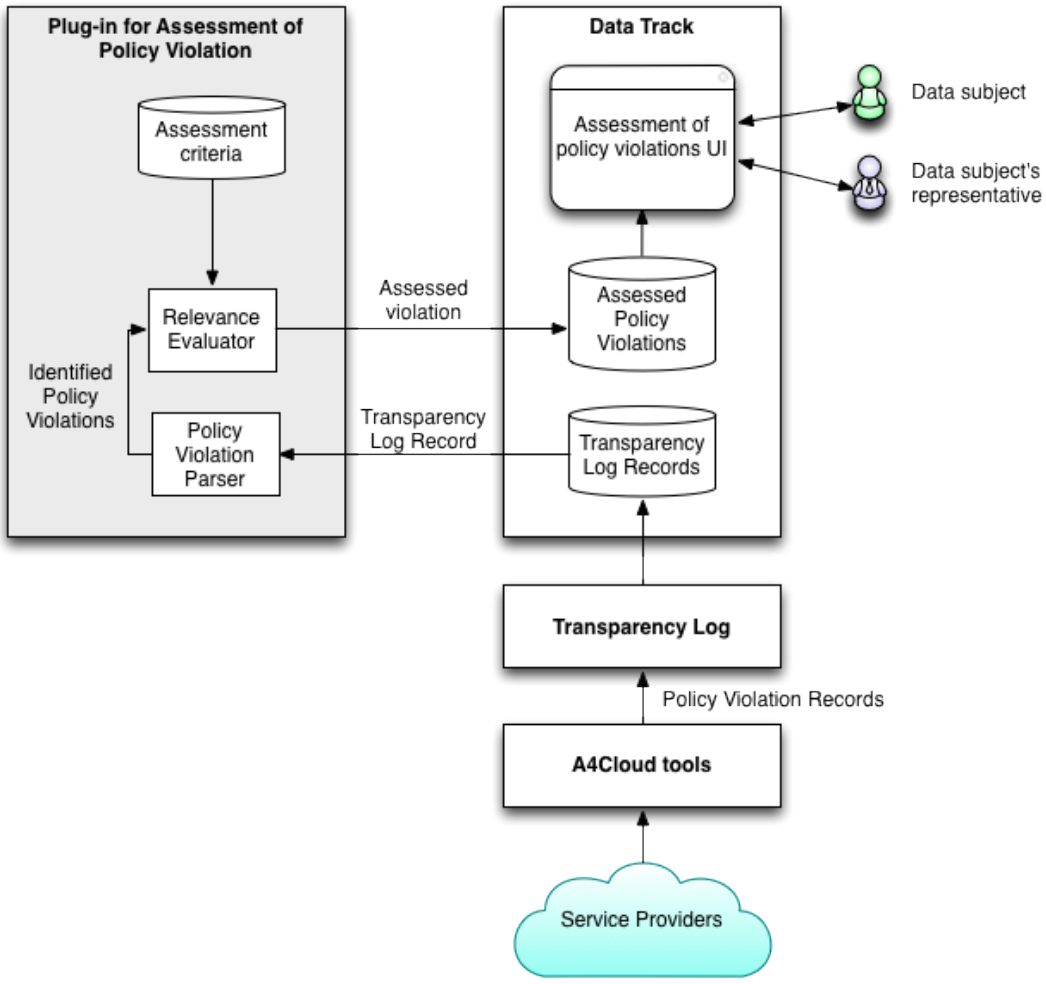

Figure 30: High level architecture of the plug-in for Assessment of Policy Violation

5.4.3 Transparency Log

The Transparency Log (TL) is based on the cryptographic scheme Insynd [44], which is used for secure and privacy-preserving unidirectional asynchronous communication in settings, where intermediate servers are considered active adversaries. TL uses Insynd to provide a one-way communication channel between service providers and data subjects. This enables a reliable channel

for sending notifications to data subjects (who cannot always be assumed to be online to receive data) even for privacy-sensitive data that normally cannot be sent by email, for example.

By using this cryptographic scheme, TL has the following properties:

- Secrecy: all stored data are encrypted and can only be decrypted by the recipient.
- Fully untrusted outsourced storage: except from availability (i.e. the storage provider can deny service), an untrusted party can store all data without loss of any of the properties provided by TL.
- Forward integrity and deletion detection: in case that the service provider sends data to data subjects through TL and this provider becomes compromised, then an attacker cannot modify those data sent using TL prior to the provider being compromised.
- Forward unlinkability of events and recipients: A third party cannot tell which recipient, out of all possible data subject recipients, received what data (stored in events) from the sender.
- Publicly verifiable proofs (i.e. that any third party can verify) of:
    - The time an event (or data inside) was sent, as certified by a time stamping authority.
    - Who the recipient of a particular event is.
    - Who the sender of a particular event is.
    - The consistency of all data stored at the untrusted party.
- Support for distributed settings, where the ability to send data to a data subject (or any recipient) can travel with the data dynamically.

TL is used by data subjects, whose personal data are being processed by cloud service providers. The tool gets as input any data a service provider wishes to send to specific data subjects. TL, in turn, provides a number of cryptographic operations on the provided input, as part of a cryptographic scheme, which, among other things, prevent information leaks and anyone from modifying the data. The outcome of this tool is the data sent by the cloud service provider together with cryptographic proofs of correctness.

As presented in section 5.4.1, the TL tool is tightly coupled with DT. Figure 29 illustrates the interactions of this tool with other A4Cloud tools, in which the TL components are coloured yellow. This summarises the intended role of TL for DT. TL is, however, a more general solution consisting of three components:

- The first is the *sender*, depicted in the diagram as the TL API. The sender holds some crypto state to be able to send data.
- The second is the *untrusted server*, depicted in the diagram as the TL Server. The server provides storage of data sent by the sender to recipients.
- The third is a *recipient*, depicted in the diagram as the TL Client. The recipient receives data from the sender by querying the server.

As a cryptographic scheme, the TL tool does not provide any Web User Interface.

The tool provides a set of APIs to communicate mainly with the data controllers. Thus, the public APIs defined here mainly refer to the sender component, while the API between sender, untrusted server, and recipient remain internal to the TL tool.

TL provides the *Itlog* API shown in Table 23, in which the methods for GET, POST, and DEL do the expected operations on logs and recipients. TL does not need any interfaces provided by other tools.

Name of the API/methods	*Purpose of use*	*Consumed by*	*Data format*	*API format*
Itlog / ListLogs	List all logs	DT (for Data Controllers)	JSON	RESTful
Itlog / SetupLog	Setup a new log to log data for recipients	DT (for Data Controllers)	JSON	RESTful
Itlog / CloseLog	Close a log, preventing further data from being logged	DT (for Data Controllers)	JSON	RESTful
Itlog / ListRecipients	List all valid recipients setup for a log	DT (for Data Controllers)	JSON	RESTful
Itlog / SetupRecipient	Setup a recipient for a log	DT (for Data Controllers)	JSON	RESTful
Itlog / RemoveRecipient	Remove a recipient for a log, preventing further data from being logged for the recipient in question	DT (for Data Controllers)	JSON	RESTful
Itlog / Log	Record data for a particular recipient in a particular log	DT (for Data Controllers)	JSON	RESTful
Itlog / GetState	Enables a recipient to query its state at the sender	DT (for Data Controllers)	JSON	RESTful

Table 26: Interfaces and respective API methods provided by the Transparency Log tool

5.5 Incident Management and Remediation

The Incident Management and Remediation functional area offers accountability support in a corrective manner. The set of tools belonging to this category facilitate the need for providing mechanisms for remediation of accountability failures and incident response.

In the context of the A4Cloud architecture, these mechanisms are being developed within the scope of the A4Cloud software tools, namely the Remediation and Redress Tool and the Incident Management Tool.

5.5.1 Remediation and Redress Tool

The Remediation and Redress Tool (RRT) aims to assist individuals or small SME cloud customers in responding to (perceived) incidents in their cloud arrangement. It is activated as a result of certain incidents reported by the Incident Management Tool (see Section 5.5.2 about IMT) or can be invoked by the user on the basis of information coming from other sources (such as newspaper reports).

If the tool is triggered by the IMT, then RRT knows what type of incident has occurred and what possible actions can be undertaken. Then, it will guide the user through these actions, which may involve composing requests/questions to the cloud provider or DPAs to take action, formal complaints or other legal actions. In case the tool is consulted by the user without being triggered by the IMT, it will engage in a dialogue with the user to establish their concern and next guide the user through appropriate actions. Where appropriate, the tool will take automatic action (through potential APIs provided by cloud service providers and/or the DPAs) to communicate requests/complaints etc.

RRT gets as input the incident data (in the form of type of incident, time and scope), some user related information (e.g. about the location, the allocated roles, the contact details, etc.), some information about the cloud service provider (e.g. about the location, the allocated roles, the contact details, etc.), any contextual information that can assist in making proper decisions on remediation and the incident response model retrieved from a knowledge base.

The tool integrates mechanisms for rendering user concerns, analysing the incoming information and providing explanation, mapping the concern or incident to the knowledge base, synthesizing the response and producing the relevant report. The output of the tool is a list of potential actions and completed (standard) forms for complaints/request etc. It also educates the respective stakeholders on incidents and potential actions and procedures.

The respective functionalities are offered through the components depicted in Figure 31. This figure integrates the view of both the RRT and the IMT.

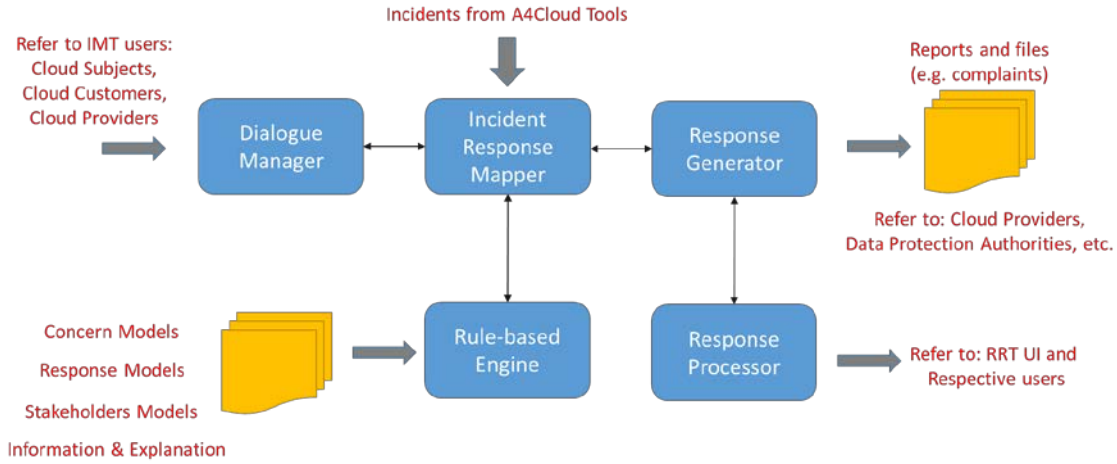

Figure 31: The high level architecture of the Incident Management Tool and the Remediation and Redress Tool

RRT offers a Web User Interface, which integrates the cases planned to be addressed by the tool, thus composing a remediation request upon invocation from the IMT and preparing the remediation request when the end use manually wants to submit one.

Currently, the tool does not consume any interface offered by other tools. As shown in Table 27, the tool offers the *IRmt* Interface, which enables interaction with the IMT to implement remediation.

Name of the API	*Purpose of use*	*Consumed by*	*Data format*	*API format*
IRrt / prepareRequest	Invokes RRT to produce the remediation request, when an incident is identified	IMT	JSON	RESTful

Table 27: Interfaces and respective methods provided by the Remediation and Redress Tool

5.5.2 Incident Management Tool

The Incident Management Tool (IMT) is the entry point for handling anomalies and detected violations in cloud environment scenarios, such as privacy violations or security breaches. The tool receives incident signals and takes the initial steps to respond to these incidents, by sending alerts to the user when a relevant incident has occurred based on different parameters. More specifically, IMT takes input from the A-PPL Engine that has collected the incidents from the evidence tools, as well, and filters them so that they can be reported and presented to the user in a comprehensive way. The tool, also, provides appropriate options to respond to the incident through the RRT.

In that respect, IMT runs on the data processor and data controller side to provide the referring business actors of the respective providers and/or customers a realisation of which incidents have been detected on their cloud arrangement. The actors responsible for handling incidents can, subsequently use IMT to verify the severity of the incident notifications and generate relevant alerts for their customers. The latter can be the providers to which this specific provider operating IMT is accountable for (in case of data processors) or the data subjects (in case of a data controller). Potentially, IMT can be used to facilitate for the need of a Cloud Auditor to receive comprehensive information about the incidents occurred in the territory of the cloud provider. In our words, of the incidents that can be reported, some, but not all, may need to be reported to entities outside the cloud service provider (that is Cloud Customer, Cloud Subject and Cloud Supervisory Authority). The tool filters out the incidents that potentially need to be reported to the tool's user.

The tool relies on incidents as they are detected at the cloud provider side. Incidents that can occur at the providers' side include a wide range of phenomena, such as hardware failure, data breach due to hacking, policy infringements, interception/surveillance by security agencies, use of data in breach of established policies, etc. Not all incidents can be detected automatically. For instance, it may not be possible to detect that a systems administrator has made a copy of the data in their system and sold it to an outsider. These types of incidents will (obviously) not be handled by the IMT.

IMT targets cloud providers and customers. The high level view of the IMT architecture is depicted in Figure 31.

IMT develops a Web User Interface, which is used by cloud customers and providers to receive alerts with respect to generated notifications. In order to do so, IMT needs to interface with other A4Cloud tools and receive information about policy violations and any incident happening in the cloud service chain, along with potential evidence on the specific incident. This is done through the A-PPL Engine.

In that respect, IMT offers the *IImt* Interface, which delivers the methods shown in Table 28. Currently, IMT does not need any external interfaces, offered by other tools.

Name of the API	*Purpose of use*	*Consumed by*	*Data format*	*API format*
IImt / getIncident	Receive a set of anomalies and violations	A-PPL Engine	JSON	RESTful
IImt / sendAlert	Communicates alerts	The UI of IMT / RRT	XML	RESTful

Table 28: Interfaces and respective methods provided by the Incident Response Tool

5.6 Summary of the tool interactions

After presenting the tools and their architectural design, in this section we make an attempt to allocate the A4Cloud tools to the cloud and data protections roles, as they were described in Section 2.1 and the relevant aspects of the accountability-enabled cloud service value chain, as introduced in the different paragraphs of Section 4.

Figure 32 shows the interactions between the A4Cloud tools to accomplish the functions allocated to the four interaction paths identified in the accountability practices (as they have been introduced in WP:C-2 and WP:C-3), namely agreement, reporting, demonstration and remediation. For each component, a respective interaction is shown by projecting the offered and required interfaces, as well as any potential communication with user interfaces.

In this Figure 32, we have used the light beige colour shows the A4Cloud tools, the cyan indicates the relevant UIs for these tools, while green has been used to showcase the involvement of external to the A4Cloud framework tools and software, such as the software application that will be developed to be follow the accountability lifecycle.

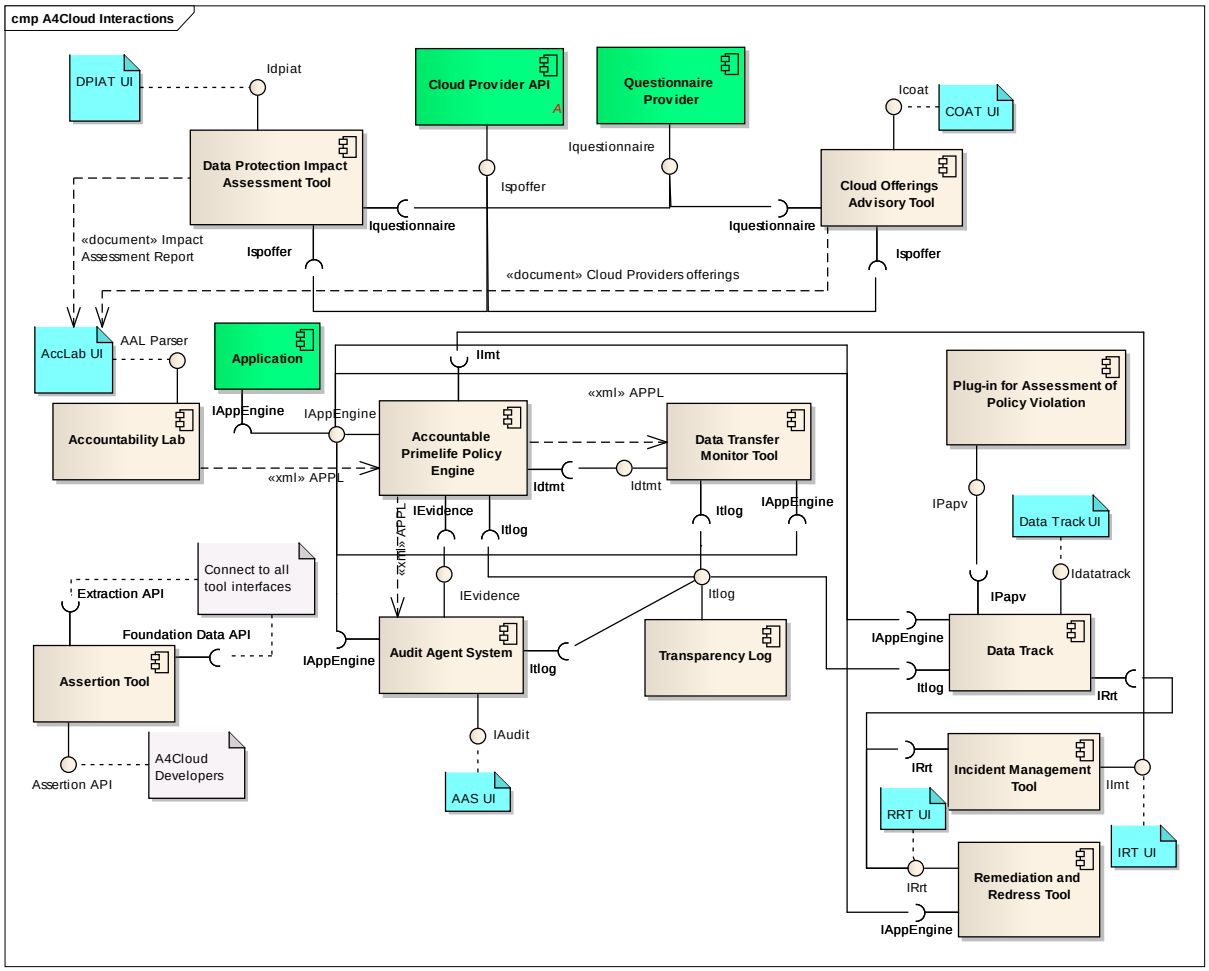

Figure 32: The interaction diagram of the A4Cloud tools

We analyse more this Figure 32 to showcase how the A4Cloud tools contribute to the development of the relevant accountability artifacts, as they are presented in Figure 14 and Table 9. More specifically, the **obligations** are built by the data controllers and they enrich them by using COAT and DPIAT to respectively select the appropriate cloud providers and perform a risk assessment on processing the involved personal data along a specific cloud service supply chain. These obligations are fed into the AccLab, along with other parameters, to develop the **accountability policies**. The latter can take the form of human readable policies or machine readable ones, which can then be enforced in a runtime environment to control the actions happen, according to the provisions of these policies. The A-PPLE exploits the accountability policies to enforce the runtime conditions for the execution of business scenarios. Part of the rules in the accountability policies are, indirectly, enforced by AAS and DTMT to configure their runtime behaviour.

The enforcement of the policies drives the collection of **machine-generated logs**, which are collected by A-PPLE, AAS and DTMT to enforce the privacy and data protection scenarios defined in the accountability policies. All these tools collect and propose logs from different layers of the cloud environment. According to the execution of the business actions and the enforceable policy rules, the collected logs are aggregated together to provide the runtime **evidence records** on how data procedures are operated in practice. The runtime evidence are mainly stored at the AAS side, but both A-PPLE and DTMT contribute to them, through the respective logs, generated or collected on their side.

Evidence can refer to the design or deployment time as well. At this point, evidence can refer to the establishment of processes and the deployment of mechanisms that support accountability. The relevant evidence artifact for the design time is based on the definition of the appropriate accountability policy and is, then, generated by the COAT, DPIAT and AccLab. The evidence at the

deployment part is built through the configuration and deployment of A-PPLE, AAS, DTMT, IMT and RRT (including the TL for the secure communication of specific functions), while AT, also, contributes to provide the verification of the proper deployment of these tools.

The analysis of logs provides a realisation of the incidents that can emerge during the operation of accountability rules. These incidents drive the generation of **Notification Reports**, which are produced by, mainly, the IMT, with the contribution of the tools generating or collecting logs (thus A-PPLE, AAS and DTMT), which deliver notification events about potential violations and incidents. Along the development of the accountability lifecycle, the relevant roles can ask for **Audit Reports**, which are produced by AAS to demonstrate the compliance to agreed policies and the legal framework.

In the following, we allocate the tools and their user interfaces to the roles of Section 2.1, excluding the AT, which is used by A4Cloud developers to validate the proper operation of the A4Cloud tools. This is depicted in Table 29, in which we analyse which tools are used by each cloud role and the relevant data protection role. The analysis considers both the user interfaces of the tools that are accessible by the indicated roles and the backend tool software that needs to be deployed on the specific role machine to provide the associated accountability function. It must be noted that, compared to the list of cloud roles shown in Section 2.1, we excluded the cloud carriers and brokers, who can mainly act as data processors (and in some case as controllers), since they are attributed in the same situation as for cloud providers.

Extended NIST cloud role / Data Protection Role	Which User Interface Tool is used	Which Tool is hosted at this role's side	Main Tool Input	Main Tool Output	Intention of use
Cloud Subject / Data Subject	AccLab UI	AccLab	Privacy and security preferences	A-PPL policy	This role can potentially use AccLab to express specific preferences to be included in the policy published by the Data Controller
	DT UI	DT	Personal data	List of data disclosures, violations from providers	Used to control the data cloud disclosures, receive data handling violations and request for actions on this data
	DT UI	PAPV	Notification on violation	Ranked violation	Providing ranking on the received violations to highlight severity levels
	DT	TL	Data recipient	Encrypted Logs	Used to securely communicate with the relevant Data Controller
	RRT UI	RRT	Notification on violation	List of potential Remediation Actions	Suggest remediation, such as to complete and submit complaints form to a DPA
Cloud Customer / Data Controller	COAT UI	COAT	Privacy and security requirements, obligations	Guidance on cloud provider selection	SMEs follow COAT to get a guided selection of a cloud provider
	DPIAT UI	DPIAT	Personal Data, Privacy and security requirements, obligations, Risk and trust models for cloud providers	Impact Assessment Report	Used to guide SMEs cloud customers to determine on the requirement to follow specific privacy, security and functional steps
	AccLab UI	AccLab	Privacy and security requirements, obligations	A-PPL policy	Express human readable obligations and requirements to machine readable accountability policies

Extended NIST cloud role / Data Protection Role	Which User Interface Tool is used	Which Tool is hosted at this role's side	Main Tool Input	Main Tool Output	Intention of use
	The UI of the business application	A-PPL Engine	A-PPL policy	Notification	Store accountability policies and handle incidents and violations
	AAS UI	AAS	Audit Tasks	Audit Report, Evidence	Receive Evidence
	IMT UI	IMT	Incident	Notification	Enable assessment on the incident types and generate notification alerts for Data Subjects and Auditors
Cloud Customer / Data Processor	The UI of the business application	A-PPL Engine	A-PPL policy, PII	Incidents	Manage PII, based on accountability policies and handle incidents and violations
	AAS UI	AAS	A-PPL policy, Audit Tasks	Audit Report, Evidence	Receive Evidence
	IMT UI	IMT	Incident	Notification	Enable assessment on the incident types and generate notification alerts for Data Subjects and Auditors
Cloud Provider / Data Controller	COAT UI	COAT	Privacy and security requirements, obligations	Guidance on cloud provider selection	SMEs follow COAT to get a guided selection of a cloud provider
	DPIAT UI	DPIAT	Personal Data, Privacy and security requirements, obligations, Risk and trust models for cloud providers	Impact Assessment Report	Used to guide SMEs cloud customers to determine on the requirement to follow specific privacy, security and functional steps
	AccLab UI	AccLab	Privacy and security requirements, obligations	A-PPL policy	Express human readable obligations and requirements to machine readable accountability policies
	The UI of the business application	A-PPL Engine, TL	A-PPL policy, PII	Encrypted logs, incidents	Enforce accountability policies, manage PII, based on them, create logs (and maintain them through TL) and handle incidents and violations
	AAS UI	AAS, TL	A-PPL policy, Audit Tasks	Audit Report, Evidence	Collect logs (and maintain them through TL), generate incidents and provide evidence
	-	DTMT, TL	A-PPL policy	Data Transfer Logs, Notifications	Raise incidents of potential violation with respect to data transfers (used in case that this is an IaaS provider)
	IMT UI	IMT	Incident	Notification	Enable assessment on the incident types and generate notification alerts for Data Subjects, Providers and Auditors
Cloud Provider / Data Processor	The UI of the business application	A-PPL Engine, TL	A-PPL policy, PII	Encrypted logs, incidents	Enforce accountability policies, manage PII, based on them, create

Extended NIST cloud role / Data Protection Role	Which User Interface Tool is used	Which Tool is hosted at this role's side	Main Tool Input	Main Tool Output	Intention of use
					logs (and maintain them through TL) and handle incidents and violations
	-	DTMT, TL	A-PPL policy	Data Transfer Logs, Notifications	Raise incidents of potential violation with respect to data transfers (used in case that this is an IaaS provider)
	AAS UI	AAS, TL	A-PPL policy, Audit Tasks	Audit Report, Evidence	Collect logs (and maintain them through TL), generate incidents and provide evidence
	IMT UI	IMT	Incident	Notification	Enable assessment on the incident types and generate notification alerts for Providers and Auditors
Cloud Auditor, Cloud supervisory authority / Supervisory authority (DPA or NRA)	AAS UI	AAS	A-PPL policy, Audit Tasks	Audit Report, Evidence	Used by this role to request compliance of actions on data handling with respect to agreed policies

Table 29: The tools and their user interfaces used by the A4Cloud cloud and data protection roles

From Table 29 and the categorization of the tools to functional areas, as done in the opening paragraphs of section 5, we also show connection of the accountability information and practices with the tools. As such:

- The policy definition and enforcement relates to the functionalities and the accountability information produced by the tools of the respective Policy Definition and Enforcement functional area. These tools handle the requirements, which can be set and/or calibrated through the tools of the Contract and Risk Management functional area, for implementing the preventive accountability mechanisms and can be used by the cloud roles to set the conditions, under which a cloud service that involves the processing of personal and/or business confidential data is operating.
- The account is progressively built along the whole accountability lifecycle. For example, at runtime the account is built by exploiting the logs collected from the cloud providers and the A4Cloud tools, namely the A-PPL Engine, AAS and DTMT. The log sources and types mainly relate to the tools of the Evidence and Validation functional area, which implement the respective detective accountability mechanisms. All these logs are analysed, based on the Framework of Evidence, as it is defined in WP:C-8, and they build the evidence records. The latter are used as part of an account to facilitate the reporting and compliance demonstration functions of the accountability framework.
- The analysis of the logs processed to constitute the evidence and the relevant account are used to raise incidents and generate notifications that drive the corresponding incident management and remediation as part of the functionalities offered by the tools of the respective functional area, in the context of the corrective accountability mechanisms.
- The data subject functionalities are being implemented through the tools of the Data Subject Controls area, which enable relevant actors to take control over how their data disclosed in the cloud is handled

The use of these A4Cloud tools addresses the accountability support services presented in Figure 17. As referred there, the tools aim to implement these services and instantiate them to support accountability. A correlation between the tools and the accountability support services in provided in Table 30

	Policy definition and compliance	Enforcement	Evidence	Account	Notification	Remediation	Metrics & AMM	Data Subject Enablement
DPIAT	√						√	
COAT							√	
AccLab	√			√			√	
A-PPLE		√	√	√	√		√	
AAS	√	√	√	√			√	
DTMT		√	√	√			√	
AT	√			√			√	
DT				√	√	√	√	√
TL			√	√	√	√	√	√
PAPV				√	√		√	√
IMT				√	√		√	
RRT				√		√	√	

Table 30: The association of the A4Cloud tools with the accountability support services

6 Conclusion

This document comprises an intermediate step towards the delivery of the full A4Cloud reference architecture (due month 42 of the project's run) structured to provide a high-level view of the architecture for accountability we are developing. It presents the conceptual basis for the development of the architecture, our view of how accountability governance should be designed, the challenges of achieving accountability across a complex supply chain and a solution based on accountability-support services to overcome them, before focusing on the design and architecture of the tools comprising the A4Cloud toolset as concrete contributions of the project. The document presented:

- A methodology for applying, from the organization's perspective, an accountability-driven governance approach, including the Accountability Maturity Model (AMM), to quantitatively assess their accountability practices;
- An in-depth discussion of the account, which provides one of the principal means of demonstrating accountability;
- A set of accountability-support services proposed to tackle the challenges of extending accountability across complex cloud service provisioning chains;
- A preliminary description of the tools which are developed by the project to address specific accountability functions and an early view on their integration.

7 References

[1] P. Krutchen, The Rational Unified Process: An Introduction, Reading : Addison-Wesley, 2000.

[2] V. Tountopoulos and et al, “Architecture guidelines and principles (internal report),” A4Cloud project, 2013.

[3] S. Pearson, M. Felici and et al., “WP-32 Conceptual Framework,” A4Cloud project, 2014.

[4] F. Liu and et al, “NIST Cloud Computing Reference Architecture,” NIST Special Publication 500-292, 2011.

[5] CIPL - Galway Project, “Data Protection Accountability: The Essential Elements,” 2009.

[6] Office of the Information and Privacy Commissioner of Alberta; Office of the Privacy Commissioner of Canada; Office of the Information and Privacy Commissioner for British Colombia, “Getting Accountability Right with a Privacy,” 2012.

[7] European Commission, “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),” 2012.

[8] CNIL, “Recommendationds for companies planning to use cloud services,” 2012.

[9] Information Commissioner's Office, “Guidance on the use of cloud computing,” 2012.

[10] Nymity Inc., “Privacy Management Accountability Framework,” 2014.

[11] IT Governance Institute, “COBIT: Control Objectives for Information and related Technology,” 2000.

[12] ISO/IEC, “ISO/IEC 27001:2013: Information technology — Security techniques — Information security management systems — Requirements,” 2013.

[13] J. De Clerq and et al, The HP Security Handbook, Hewlett Packard, 2008.

[14] UK Information Commissioner's Office, “Guidance on the use of cloud computing,” 2012.

[15] J. Luna and D. Cattedu, “Report on A4Cloud contribution to standards,” A4Cloud project, 2014.

[16] A. Pannetrat and et al, “The interoperability of A4Cloud Framework,” A4cloud project, 2014.

[17] ISO/IEC/IEEE, “ISO/IEC/IEEE 29119 Software and systems engineering - Software testing,” 2013.

[18] Software Engineering Institute (SEI), “CMMI for development: Improving processes for developing products and services.,” 2010.

[19] Cloud Security Alliance (CSA), “Cloud Controls Matrix,” 2014.

[20] NIST Public RATAX WG, “Cloud Computing: Cloud Service Metrics Description,” 2014.

[21] ISO/IEC, “Information Technology – Security techniques – Information Security Management – Measurement,” 2009.

[22] NIST, “NIST Cloud Computing Security Reference Architecture,” 2013.

[23] A. Taha, R. Trapero, J. Luna and N. Suri, “AHP-Based Quantitative Approach for Assessing and Comparing Cloud Security,” in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, 2014.

[24] J. Luna, R. Langenberg and N. Suri, “Benchmarking Cloud Security Level Agreements Using Quantitative Policy Trees,” in *ACM Cloud Computing Security Workshop*, 2012.

[25] CIPL Paris Project, “Demonstrating and measuring accountability: a discussion document,” 2010.

[26] D. A. H. B.-L. T. F. J. H. J. &. S. G. Weitzner, “Information accountability,” *Communications of ACM 51(6),* no. June 2008, pp. 82-87, 2008.

[27] C. Bennett, “The Accountability Approach to Privacy and Data Protection: Assumptions and Caveats,” in *In Managing Privacy through Accountability*, D. G. e. al., Ed., MacMillan, 2012, pp. 33-48.

[28] European DG of Justice (Article 29 Working Party), “Opinion 3/2010 on the Principle of Accountability (WP 173),” 2010.

[29] S. Bradshaw, C. Millard and I. Walden, “Standard Contracts for Cloud,” in *Cloud Computing Law*, C. Millard, Ed., Oxford OUP, p. 37 – 72.

[30] K. W. Hon, C. Millard and I. Walden, “Negotiated Contracts for Cloud,” in *Cloud Computing Law*, C. Millard, Ed., Oxford OUP, 2013, pp. 73-107.

[31] C. Raab, “The Meaning of ‘Accountability’ in the Information Privacy Context,” in *Managing Privacy through Accountability*, D. e. a. Guagnin, Ed., MacMillan, 2012, pp. 15-32.

[32] Hunton & Williams LLP, “Data Protection Accountability: The Essential Elements – a Document for Discussion,” 2009.

[33] K. Bernsmed and e. al., “D:B-3.2 Consolidated use case report,” A4Cloud Project, 2014.

[34] European DG of Justice (Article 29 Working Party), “Binding Corporate Rules,” [Online]. Available: http://ec.europa.eu/justice/data-protection/document/international-transfers/binding-corporate-rules/index_en.htm.

[35] E. Kosta and e. al., “MS:D-4.1 “Internal report on legal analysis relating to redress mechanisms and remediation”,” A4Cloud Project, 2014.

[36] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” NIST Special Publication 800-145, 2011.

[37] Cloud Security Alliance (CSA), “CSA Security, Trust & Assurance Registry (STAR),” [Online]. Available: https://cloudsecurityalliance.org/star/.

[38] W. Benghabrit and et al, “A cloud accountability policy representation framework,” A4Cloud project, 2014.

[39] W. Benghabrit and et al, “Abstract Accountability Language,” in *8th IFIP WG 11.11 International Conference on Trust Management*, Singapore, 2014.

[40] S. Trabelsi and et al, “PPL: PrimeLife Privacy Policy Engine,” in *IEEE International Symposium on Policies for Distributed Systems and Networks (POLICY)*, Pisa, 2011.

[41] M. Azraoui and et al, “A-PPL: An Accountability Policy Language for Cloud Computing,” in *DPM / SETOP*, Wroclaw (Poland), 2014.

[42] T. Wlodarczyk and et al, “D:C-8.1 Framework of Evidence,” A4Cloud project, 2014.

[43] “OpenStack: Open source cloud computing software,” [Online]. Available: https://www.openstack.org/.

[44] T. Pulls and et al, “Distributed privacy-preserving transparency logging,” in *Proceedings of the 12th annual {ACM} Workshop on Privacy in the Electronic Society*, Berlin, Germany, 2013.

[45] K. Bernsmed and et al., “MSB-3.1 Use Case Descriptions,” A4Cloud project, 2014.

8 Appendices

8.1 List of Obligations

The following list of obligations was extracted from the WP B-3 MSB-3.1 report [45]. We point to that report for a full list of those obligations that provides extended details, including legal perspectives.

List of obligations from the regulatory perspective (Data Protection Directive), to which Cloud actors must adhere:

- **Obligation 1: informing about processing**. Data subjects have the right to know that their personal data is being processed.
- **Obligation 2: informing about purpose.** Data subjects also have the right to know why their personal data is being processed.
- **Obligation 3: informing about recipients.** Data subjects have the right to know who will process their personal data.
- **Obligation 4: informing about rights.** Data subjects have the right to know their rights in relation to the processing of their personal data.
- **Obligation 5: data collection purposes.** Personal data must be collected for specific, explicit and legitimate purposes and not further processed in a way incompatible with those purposes.
- **Obligation 6: the right to access, correct and delete personal data.** Data subjects have the right to access, correct and delete personal data that have been collected about them.
- **Obligation 7: data storage period.** Personal data must be kept in a form that permits identification of data subjects for no longer than is necessary for the purpose for which they were collected.
- **Obligation 8: security and privacy measures.** Controllers are responsible to the data subjects for the implementation of appropriate technical and organizational security measures.
- **Obligation 9: rules for data processing by provider.** Controllers are accountable to data subjects for how sub-providers process their personal data.
- **Obligation 10: rules for data processing by sub-providers.** The controller must also ensure that all sub-providers involved in the service delivery chain do not process the personal data, except on the controller's instructions (unless they are required to do so by law).
- **Obligation 11: provider safeguards.** Controllers are accountable to data subjects for choosing data processors that can provide sufficient safeguards concerning technical security and organizational measures.
- **Obligation 12: sub-provider safeguards.** The previous obligation comprises all processors in a service delivery chain.
- **Obligation 13: informed consent to processing.** Controllers are accountable to the data subjects for obtaining informed consent before collecting personal data.
- **Obligation 14: explicit consent to processing.** Controllers are accountable to the data subjects for obtaining explicit consent before collecting sensitive personal data.
- **Obligation 15: explicit consent to processing by joint controllers.** Controllers are accountable to the data subjects for obtaining explicit consent before allowing joint data controllers to process their sensitive personal data.
- **Obligation 16: informing DPAs.** Controllers are accountable to the data protection authorities to inform that they collect personal data.
- **Obligation 17: informing about the use of sub-processors.** Processors are accountable to the controllers for informing about the use of sub-providers to process personal data.
- **Obligation 18: security breach notification.** Controllers are accountable to data subjects for notifying them of security incidents that are related to their personal data.

- **Obligation 19: evidence of data processing.** Processors are accountable to the controllers for, upon request, providing evidence on their data processing practices**.**
- **Obligation 20: evidence of data deletion.** Processors are accountable to the controllers for, upon request, providing evidence on the correct and timely deletion of personal data.
- **Obligation 21: data location.** Data controllers are accountable to the data subjects for the location of the processing of their personal data.

9 Index of Figures

10 Index of Tables