Deliverable D6.1
Methodology for Decentralized Data lifecycle Management
Release 1.0

The research leading to these results has received funding from the European Community’s H2020 research and innovation programme under grant agreement n° 731678.
<table>
<thead>
<tr>
<th><strong>Project Number</strong></th>
<th>:  731678</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Title</strong></td>
<td>: RestAssured - Secure Data Processing in the Cloud</td>
</tr>
<tr>
<td><strong>Deliverable Type</strong></td>
<td>: External Report</td>
</tr>
</tbody>
</table>

| **Title of Deliverable** | : Methodology for Decentralized Data lifecycle Management |
| **Nature of Deliverable** | : External Report |
| **Dissemination Level** | : Public |
| **Contractual Delivery Date** | : 30 April 2018 |
| **Actual Delivery Date** | : 30 April 2018 |
| **Contributing WPs** | : All |
| **Editor(s)** | : Dhouha Ayed |
## Contributors

<table>
<thead>
<tr>
<th>Name</th>
<th>Organization</th>
<th>Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dhouha Ayed, Cyrille Piatte</td>
<td>Thales</td>
<td>Data Lifecycle Security Methodology, Architecture of the Gatekeeper, Policies and Contracts specification, Data Protection Decision Point, Data Protection Enforcement Point, Obligation Execution and Enforcement, Introduction, Conclusion</td>
</tr>
<tr>
<td>Adam Burns, Paul Mundt</td>
<td>Adaptant</td>
<td>PAYD Use Case</td>
</tr>
<tr>
<td>Nazila GolMohammadi, Ludger Goeke</td>
<td>UDE</td>
<td>Conceptual Model</td>
</tr>
</tbody>
</table>
Contents

List of Figures ........................................................................ 6

1 Introduction ...................................................................... 7

2 Data Lifecycle Security Management : Methodology ................. 8
   2.1 Data Lifecycle Concepts .............................................. 8
   2.2 GDPR requirements ................................................... 9
      2.2.1 The data subject consent and information ................. 10
      2.2.2 Data Removal ................................................... 10
      2.2.3 Privacy by design ............................................. 10
      2.2.4 Limitation of profiling ........................................ 11
      2.2.5 Data Portability ............................................... 11
      2.2.6 Records ......................................................... 11
   2.3 Conceptual Model ....................................................... 11
   2.4 Data Lifecycle Security Process Overview ......................... 14
      2.4.1 Data processor registration phase ......................... 15
      2.4.2 Data subject registration phase ......................... 15
      2.4.3 Data usage phase ........................................... 16

3 Data Protection Policy Specification ........................................ 18
   3.1 Data protection contract ............................................. 18
   3.2 Sticky policies .......................................................... 19
   3.3 Specification of obligations ......................................... 19

4 Data Lifecycle Security Management : General Architecture ......... 23
   4.1 Data Gatekeeper Building Blocks ................................... 23
      4.1.1 Data Protection Manager .................................. 23
      4.1.2 Sticky Policy Manager ..................................... 23
      4.1.3 Data Protection Decision Point ......................... 24
      4.1.4 Data Protection Enforcement Point .................. 24
      4.1.5 Service Registry ........................................... 24
      4.1.6 AuthN ......................................................... 24
   4.2 Data lifecycle security management process .................... 24
      4.2.1 Data processor registration ............................. 24
      4.2.2 Data subject registration ................................. 25
      4.2.3 Data Usage .................................................. 25
4.3 Data protection decision point: DPDP .................................................. 26
4.4 Data protection enforcement point .................................................. 27
4.5 Generic Public Key Encryption ....................................................... 28
4.6 Attribute Based Encryption .............................................................. 28
4.7 Proxy Re-Encryption ................................................................. 29
4.8 Obligation execution and enforcement ................................................. 30

5 End-User Policy Specification in PAYD Use Case ........................................ 32
5.1 Introduction .................................................................................. 32
5.2 Conceptual Data Model and Data Policy Framework ................................ 33
5.3 Use Case Policy Specification Examples ........................................... 33
   5.3.1 Service Provision ........................................................................ 33
   5.3.2 Example Optional Service Offering: Driving Behaviour Risk Assessment .................. 35
   5.3.3 Example Optional Service Offering: Share Journey Data with Telematics Provider ........ 35

6 Conclusion .................................................................................. 36

Bibliography .................................................................................. 37
List of Figures

Figure 1.1: RestAssured Architectural Overview ................................................................. 7
Figure 2.1: Data Lifecycle according to the CSA ................................................................. 8
Figure 2.2: Data Security Lifecycle .................................................................................... 9
Figure 2.3: Conceptual model of the policy specification framework ............................... 12
Figure 2.4: Policy handling overview ............................................................................... 15
Figure 2.5: Data processor registration phase ................................................................. 16
Figure 2.6: Data subject registration ............................................................................... 16
Figure 2.7: Data usage ...................................................................................................... 17
Figure 3.1: Data protection contract ............................................................................... 19
Figure 3.2: Sticky policies ............................................................................................... 20
Figure 3.3: Obligation Ontology ...................................................................................... 21
Figure 3.4: Sticky Policy Ontology including Obligations ........................................... 22
Figure 4.1: General Architecture ..................................................................................... 23
Figure 4.2: Registration Process ....................................................................................... 25
Figure 4.3: Data Usage ..................................................................................................... 26
Figure 4.4: Decision Process .......................................................................................... 27
Figure 4.5: Generic Public Key Encryption ..................................................................... 28
Figure 4.6: Proxy ReEncryption Enforcement ................................................................. 30
Figure 5.1: Simplified PAYD Service Model .................................................................... 32
Figure 5.2: PAYD Conceptual Data Model .................................................................... 34
Figure 5.3: PAYD Conceptual Service Model ................................................................. 34
1 Introduction

One of the main objectives of RestAssured is to provide solutions to specific technical concerns of data protection in the cloud, which are imposed by the dynamic, multi-stakeholder and decentralized nature of federated cloud system. This deliverable (D6.1) is the first deliverable of WP6 whose objective is to facilitate managing the complete data lifecycle in a decentralized cloud setting.

The goal of the deliverable is to define a methodology based on sticky policies that is used by the Data Gatekeeper. As described in D3.2, the Data Gatekeeper manages the data protection policies and the services governing the data life-cycle. It is responsible for deciding, based on the available policies and various constraints (legislation, context, location, etc.), which operations are allowed on which piece of data and to enforce user data protection policies in a decentralized computational environment.

The document is structured as follows:

- Section 2 describes the data lifecycle methodology based on our approach and the conceptual model;
- Section 3 details the structure of the data lifecycle security policies;
- Section 4 describes the architecture of the Data Gatekeeper;
- Section 5 illustrates the methodology with the PAYD use case;
- Section 6 concludes the document;
2 Data Lifecycle Security Management: Methodology

One means of addressing the security of data once it has been released beyond the direct control of its owner is by using sticky policies.

A sticky policy is a security policy, which is associated with (stuck to) a piece of data such that access to and use of that data is only possible if the policy has been complied with. Attempts to un-stick the policy or modify or replace it should render the data inaccessible.

In this section we describe the approach that the Data Gatekeeper uses to manage the data lifecycle security. Section 2.1 introduces some concepts of the data lifecycle and our vision on its security. Section 2.2 describes the concepts we use in our approach based on the requirements of the GDPR. Section 2.3 describes the conceptual model that we propose for data protection policy specification. Section 2.4 gives an overview on the data lifecycle security process used by the Data Gatekeeper.

2.1 Data Lifecycle Concepts

The objective of the Data Gatekeeper is to securely manage the complete data lifecycle (data creation, storage, transfer, processing and deletion) in a decentralized cloud setting. The nodes may be geographically dispersed, across national and EU borders and may have to reflect privacy expectations from various stakeholders (especially data subjects), thereby implying different data protection policies for the different nodes.

According to the CSA (Cloud Security Alliance), the data lifecycle includes six phases (see figure 2.1) [1].

- Create: Creation is the generation of new digital content, or the update of existing content.
- Store: Storing is the act committing the digital data to a storage repository.
- Use: Data is viewed, processed or used in a specific activity.
- Share: Information is made accessible to various actors or organisations.
- Archive: Data is not actively used and enters long-term storage.
- Destroy: Data is permanently deleted.

Data can bounce between these phases without restriction, and may not pass through all stages. For example, not all data is destroyed.

Figure 2.1: Data Lifecycle according to the CSA
Figure 2.2 illustrates the steps and roles that are required to successfully implement the data lifecycle security. Security has to be analysed at the creation of data. This analysis results in a set of security requirements and data protection preferences that are related to all the steps of the data lifecycle (storage, usage, etc.). The data subject shares these security requirements and preferences with the data controller, the entity that manages the data on behalf of the data subject. The data controller specifies or generates the sticky policies that are related to the data and plans the enforcement of these policies and the storage of the data according to these requirements. The data controller relies on a service registry that catalogues the various cloud services (IaaS, PaaS, and SaaS) and their security capabilities in order to select the services that comply with the policies. Once the data is stored, security checks are needed at rest and during data processing to ensure that security enforcement being used is effectively addressing the security policies. If not, data deployment and security enforcement must be revised in order to adopt changes and to address new risks.

Figure 2.2: Data Security Lifecycle

### 2.2 GDPR requirements

The General Data Protection Regulation (GDPR) is the new Regulation for protecting the personal data of European citizens. It will enter into force on the 25th May 2018. The GDPR is a law to which companies, located in the European Union (EU) or offering services to EU nationals, must abide. It supersedes the European Union Data Protection Directive which led to different privacy laws in different European countries. The major changes, comparing to the previous directive are mainly related to the Consent of the Data Owner, the penalties for the companies that do not comply with the GDPR and the Territorial Scope. The changes that are able to impact the RestAssured secure data lifecycle management solution and methodology are detailed in the following.
2.2.1 The data subject consent and information

The data subject consent means “given specific, informed and explicit indication of wishes by which the data subject, either by a statement or by a clear affirmative action, signifies agreement to personal data relating to them being processed.” (GDPR, Article 4). The processing is lawful only if it meets at least one of these criteria:

- the data subject has given consent;
- the processing is necessary for the performance of a contract involving the data subject;
- the processing is necessary for compliance with a legal obligation;
- the processing is necessary for the performance of a task carried out in the public interest.

The controller must be able to provide proof of consent for each treatment (nature, date, etc.) and it should be as simple to withdraw consent as to give it (Article 7).

Any communication about treatment should be done in a concise, transparent, intelligible and easily accessible form, using clear and plain language (Article 12). When data are collected from the data subject, the data subject shall be informed in particular of (Article 13):

- details about the controller;
- a possible transfer of this data abroad;
- the data retention period.

When data is not collected from the data subject, the source of the data must be indicated in addition to the information listed above (Article 14).

2.2.2 Data Removal

Removal of data must be done as soon as possible in the following cases:

- data is no longer needed for the original reason;
- the person has withdrawn their consent to the treatment (Article 7);
- the person exercises his right of objection (Article 21);
- data were the subject of unlawful treatment;
- for compliance with a legal obligation defined by the EU.

2.2.3 Privacy by design

Privacy by design consists in proactively ensuring the processing of personal data from the design of a system. “The controller shall, both at the time of the determination of the means for processing and at the time of the processing itself, implement appropriate technical and organisational measures, which are designed to implement data-protection principles, such as data minimisation, pseudonymisation” [6]. Data minimisation consists in collecting the minimum data necessary for the processing for which they are intended and limit their retention period.
2.2.4 Limitation of profiling

"Profiling means any form of automated processing of personal data consisting of the use of personal data to evaluate certain personal aspects relating to a natural person, in particular to analyse or predict aspects concerning that natural person’s performance at work, economic situation, health, personal preferences, interests, reliability, behaviour, location or movements" [6]; The data subject shall have the right to object, on grounds relating to his particular situation, at anytime to processing of personal data concerning him, including profiling based on those provisions. The controller shall no longer process the personal data unless the controller demonstrates compelling legitimate grounds for the processing which override the interests, rights and freedoms of the data subject or for the establishment, exercise or defence of legal claims.

2.2.5 Data Portability

- "The data subject shall have the right to receive the personal data concerning him, which he has provided to a controller, in a structured, commonly used and machine-readable format and have the right to transmit those data to another controller without hindrance from the controller" [6];
- Data controllers should be encouraged to develop interoperable formats that enable data portability (such as XML, JSON);
- In exercising his right to data portability, the data subject shall have the right to have the personal data transmitted directly from one controller to another.

2.2.6 Records

The controller should maintain a record of processing activities that must be made available to the supervisory authorities. The main information to be recorded is the following:

- all type of needed details about the controller
- the purposes of the processing
- a description of the categories of data subjects and of the categories of personal data where applicable
- transfer of personal data to a third country or an international organisation
- the envisaged time limits for erasure

2.3 Conceptual Model

The conceptual model (see Figure 2.3), that we propose as part of our policy specification framework, represents the relations between main data protection concepts and end-users, service providers as well as data protection policies. In the following the different actors and entities of the policy specification framework are explained.

To determine the roles and requirements related to data protection, we consider two key formal documents that define data protection principles and help identifying the related actors and entities. On the one hand, we consider the specification framework as defined in the international standard ISO/IEC 29100 [2]. On the other hand, we consider the EU GDPR [6] which is the currently binding legal framework in the member states of the European Union.

ISO/IEC 29100 and the EU GDPR partly overlap and do not use the same terms consistently. To avoid ambiguities, we use the following definitions for the most important actors that are relevant in the context of our policy specification framework:
Figure 2.3: Conceptual model of the policy specification framework

- **Data Subject** is an identifiable natural person, who can be identified directly or indirectly, in particular by reference to an identifier such as a name, identification number, location data, online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person [6]. This term is called Personally Identifiable Information (PII) principal in ISO/IEC 29100. In this document, a Data Subject represents a person whose personal data is going to be stored and/or processed in the context of a cloud computing service (Service). The Data Subjects have, with respect to their personal data, the rights stipulated by the GDPR. In this document, we consider an End-User as anyone who interacts with a service.

- **Data Controller** is the natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data [6]. This term is called PII controller in ISO/IEC 29100. In this document, a Data Controller represents a legal entity providing a Service that stores and/or processes personal data. The Data Controller has, with respect to the stored/processed personal data, the obligations stipulated by the GDPR. We consider the cloud provider at the first interaction point with the End-Users as the Data Controller.

- **Data Processor** is any other involved service provider that does not play the role of a Data Controller but processes the Data Subjects’ Sensitive Data. In the context of cloud computing many parties can be involved as Data Processors, but only one of them is responsible for managing the policy as the Data Controller.

- **Sensitive Data** represents personal data that constitutes the focus of GDPR. In this context, personal data means any information relating to an identified or identifiable natural person (Data Subject) [6]. This term is called PII in ISO/IEC 29100. We consider Sensitive Data as the data stored and/or processed in the cloud computing service that needs to be protected in line with the data protection preferences of the affected Data Subject. In particular, we extend the notion of Sensitive Data to also include confidential business data.
Service represents a cloud computing service that stores and/or processes the Data Subjects’ Sensitive Data. The Data Controller is responsible for the legal and compliant operation of the provided Service in form of a cloud computing services.

In the following, further elements of the conceptional model of the policy specification framework are explained.

A Data Subject provides Sensitive Data to a Service that intends to process the Sensitive Data performing special Actions on it. Here, the different Actions define what can be done with the provided Sensitive Data by the Service. The Data Subjects have Data Protection Requirements that satisfy their Data Protection Goals for their provided Sensitive Data. By using the policy specification framework the Data Subjects can present their Data Protection Requirements to the provider of the used Service. To this, a Data Protection Policy is generated from the Data Subjects’ input that specifies the Data Protection Requirements of a Data Subject in a formal way. Such a policy contains Statements that define Obligations to the Actions of the Service that can be applied to the Sensitive Data. These Obligations can for example prohibit the processing of Sensitive Data for marketing purposes.

The Statement is one of the core elements of our conceptual model. It indirectly relates to the Data provided by the Data Subjects and is stuck to this Data by using the approach of data protection policies. Additionally, it refers directly to an Action that is applied on the Data by the Service and the corresponding Obligations for this Actions. These associations to the Data, Action and Obligation enables a Statement to define the set of Actions that are allowed to be applied on the Data.

Accordingly, a Statement implies conditions and rules for the referenced Action. The generic Statement can be aggregated into one or more specific Statements. These specific Statements are considered in the following.

- **Collection Statements** represent Statements that specify which data is allowed to be collected by a Service Provider. In this context, such Statements restrict the Actions regarding the collection of Data by a Service Provider.

- **Disclosure Statements** specify the rules and conditions regarding which Data of the Data Subjects is allowed to be disclosed to which audience.

- **Storage Statement** specifies how long the Data of Data Subjects is allowed be stored.

- **Usage Statement** specifies to which purpose the Data Subjects’ Data is allowed to be used and processed.

- **Notification Statement** specifies the rules and conditions for events about which the Data Subjects have to be informed. Furthermore, the means of notification is specified.

- **Consent Statement** describes where and in which condition the Data Subjects are required to give their consent.

- **Migration Statement** describes the rules and conditions for the migration (location) of Data.

The Actions that are referenced by Statements can be also aggregated into specific Actions. In the following, the different specific Actions are considered.

- **Collect**: Collection of information that is not directly supplied by the Data Subject

- **Store**: Storing of Data that has been supplied by the Data Subject

- **Process/Use**: Processing of Data resulting in new data
- **Disclose/Share**: Sharing Data with other parties
- **Notify**: Notification and consent mechanisms to provide proper transparency to the Data Subject, especially when a data protection violation occurs
- **Transfer**: Transferring Data to other locations

Every Action has a **Purpose** that defines whether an Action is necessary for service provisioning or not. A Purpose can also be aggregated as specific Purposes. In this context, **Service Provision** defines specific Purposes that are necessary for the provision of the Service. Purposes that are **not** necessary for the actual provision of the Service are represented by the particular specific Purposes **Marketing**, **Analytics** or **Advertising**. By using the association of Purposes to Actions, it is possible to decide whether the usage of Data is actually necessary to provide the Service or not. If an Action is not necessary for the service provision, the Data Subjects can decide to not allow this Action.

The Data Controller manages both Data Protection Policies, the one provided by the Service Provider and the Sticky Policy created from the input of the Data Subjects. By judging the Purpose, the Data Controller can decide whether a Statement given by the Service Provider is necessary to provide the Service or whether it is used for additional Purposes. Making it possible to give the user choices on the **Data Protection Policy**, but not allowing strict policies, which would conflict with the data processing needed to provide the Service. The **Data Controller** manages the policies together with the Sensitive Data and controls the involved Service Provider, so that the Services provided by this provider only perform Actions that the Data Controller permits. Thus, it shall be ensured that a Service performs only those Actions to the Data that has been permitted by the Data Subject.

In the following the process of data protection policy handling is described.

Our policy specification framework informs the Data Subjects about processing that is necessary for service provision, e.g. storing of files when using a file sharing service. All other Actions with other Purposes can be either permitted or rejected by the Data Subjects.

The framework finds a trade-off between the unconstrained policy provided by the Service Provider, who wants to process/collect more data than necessary in order to get the most value out of the Data Subject’s input, and the Data Subject’s more strict policy.

All Statements of the Data Protection Policy that are not stated to be necessary for the provision of the Service are considered optional. The Service Providers can specify a processing as necessary by using the Purpose **Service Provision** when specifying the Data Protection Policy for a Service. Optional statements are displayed to the Data Subjects as a modifiable input, allowing the Data Subjects to enter their data protection preferences.

**Fig. 2.4.** shows the relations between the components involved in our proposed policy specification framework. The Privacy Requirements are specified by the Data Subjects (see Fig. 2.4 step 1b)) and translated into Sticky Policies (see Fig. 2.4 step 1c)). Data Protection Contracts are in a first step provided by the Service Providers (see Fig. 2.4 step 1a)) and afterwards agreed and adjusted by the Data Subjects (see Fig. 2.4 step 2a)). The adjustments are then encoded in Sticky Policies (see Fig. 2.4 step 3)) which in the next step are attached to the Sensitive Data (see Fig. 2.4 step 4)) that is going to be processed in the Services provided by the Service Providers.

### 2.4 Data Lifecycle Security Process Overview

The various phases of the data lifecycle security management methodology applied by the Data Gatekeeper are detailed in the following.
2.4.1 Data processor registration phase

A Service Provider or any entity that needs to process the data has to register with the Data Gatekeeper in order to specify the following information:

- the type of data that is needed by the service;
- the planned usage of data (actions and purposes).

This information is stored in a machine readable Data Protection Contract, that is signed before its storage and made available for the Data Subject (see Figure 2.5). It represents a contract of data usage/processing that is negotiated with the Data Gatekeeper provider (see data protection contract details in section 3.1). The Data Gatekeeper has to guaranty the terms of this contract.

A set of security properties can also be collected from the service provider/data processor, describing for example the authentication system, the cryptography, usage of an SGX platform, etc.. This information is stored in a service registry to be used in the selection of services according to the security properties required by data subjects or end-users.

2.4.2 Data subject registration phase

One of the main roles of the Data Gatekeeper is to ensure that the data is accessed and processed according to the preferences of the Data Subject. The Data Gatekeeper makes the Data Protection Contracts defined with the various data processors available over a User Interface. With this user interface, the Data Subject visualises the data types and the kind of processing needed by each service provider (see Figure 2.6). Based on this information, the Data Subject can specify his preferences on the usage of the data. These preferences detail the services that can access the data and the conditions under which this data can be accessed or processed. If the Data Subject preferences are not compliant with the terms of the Data Protection Contracts, he will not be allowed to register with the service. If the requirements are compliant, they are translated into machine readable Sticky Policies that will be enforced each time an entity requests the data (see more details in section 3.2). The generated Sticky Policies are signed and stored in a Sticky Policy storage place that is
managed by the Data Gatekeeper. They are logically bound to the personal data during the whole data lifecycle and even when data is transferred or not managed by the Data Gatekeeper provider. Each lifecycle action on the data (access, deletion, update, transfer, etc.) is enforced to check the compliance to the sticky policies.

2.4.3 Data usage phase

This phase (see Figure 2.7) is about the management of data access and processing when it is requested by an end-user of a service. The end-user starts by authenticating in order to open a session on the service. This authentication process can be delegated to the Data Gatekeeper, in this case the end-user is redirected to the Data Gatekeeper for authentication. The service requesting the usage of the data has also to be authenticated by the Data Gatekeeper.

If one of the Authentication proofs is not valid, access is denied (if the Service authentication fauls, the service cannot be trusted). If the Authentication proofs are both valid, the Data Gatekeeper uses combination algorithms based on the data protection contract, the sticky policies and the organizational access control policies to deliver either a Deny access or a Grant access.
Figure 2.7: Data usage
3 Data Protection Policy Specification

Data Protection policies (sticky policies and data protection contracts) can be based on existing access control models, such as Discretionary Access Control (DAC), Role Based Access Control (RBAC), Attribute Based Access Control (ABAC) \[5\] or Context Based Access Control (CBAC) \[4\].

Recent work on access control policy specification and enforcement focused on representing existing access control models using semantic technology; proposing new access control models suitable for open, heterogeneous and distributed environments \[3\].

ABAC is a flexible model that grants or denies access to resources, based on properties of the subject and/or the resource, known as attributes that can be very easily adapted to the specification of data protection policies but we think that CBAC is more advantageous for sticky policies usage and data protection since it uses properties, pertaining to users, resources and the environment, to grant or deny access to resources.

For more openness, heterogeneity, support of distributed environments, and flexibility, we choose the CBAC (Context Based Access Control) policy model using ontologies and rules to specify access control policies that take context relating to subjects, objects, transactions and the environment into consideration.

Human-understandable ontologies allow us to express the Data Subject’s requirements in sticky policies. The ontologies can also be used to transcribe the legislations, regulation and organizational security policies in addition to the fact that they represent one of the best formalism to specify contextual information pertaining to the user, resources or the environment.

The technologies based on linked data, such as graph reasoning, inference and query languages offer a fine-grained and strong access control mechanisms and ease the automation for querying access control information. In addition to that, the usage of linked data technologies offers interoperability functionalities, in compliance with the portability between Data Controllers, specified by the GDPR. In our approach, we use semantic technologies in two ways: to bind security policies to a data subjects and to filter the data to be protected.

3.1 Data protection contract

The goal of a Data Protection Contract is to define the scope of the access and processing of the data by a Service Provider for a specific service. A Data Protection Contract is composed of 4 main parts:

- A description of the Service that needs to access the data;
- A specification of the data that is needed to be accessed by the service and whether this access is mandatory;
- A list of usages; a usage is composed of a textual description that describes the purpose of this usage (example: statistics, marketing) and the associated action (example: store, read, etc.) as well as a list of needed data types.
- A specification of data that will be published by the service;

Figure 3.1 details the structure of a data protection contract:

This detailed specification of data usage in a contract, that can be specified at the design time of the service and that is made available to the data subject (through an intermediate trusted authority or not) before he uses the service could be used to inform the user about the way that his data is used and processed during its whole lifecycle as required by the GDPR.
3.2 Sticky policies

Each data subject can be subscribed to a list of services that are provided by various service providers. The Data Subject can specify a set of requirements related to the access and usage to his data. The Data Protection Contracts associated to each service can be used by the data subject to be guided in the specification of these requirements. In addition to the informative role that a data protection contract plays, the data subject can use it to visualise each data that is needed by a service, the usage for which it will be exploited, and give his consent or not.

If the data subject gives his consent for a specific usage of data, he has the ability to specify a set of context conditions under which the usage is allowed.

The Sticky Policy Manager of the Data Gatekeeper generates machine readable sticky policy by associating each data mentioned in the data subject preferences to its corresponding path in the ontology that models the data. The Sticky Policy Manager also generates a context, translating the requirements of the Data Subject. Sticky policies are signed and typically stored in a triple store in RDF. Figure 3.2 details the structure of a sticky policy:

3.3 Specification of obligations

An Obligation is an action that has to be performed when an event is triggered [7]. The type of events that we are interested in are, access requests to data (to execute various actions of data lifecycle), and data transfer between entities. We consider three types of Obligations:

- before Obligations: Obligations that need to be enforced before the access to the data is granted (or denied)
- with Obligations: Obligations that need to be enforced at the same time as the access to the data is granted (or denied)
- after Obligations: Obligations that need to be enforced after the access to the data is granted (or denied)
Figure 3.2: Sticky policies

An obligation is specified with three pieces of information:

- the obligation type (‘before’, ‘after’, ‘with’)
- the event that triggers the Obligation
- the action that has to be performed

An obligation can be specified in a sticky policy, associated to a context of usage of a data (which represents the event that triggers an Obligation). A list of obligations can be associated to each data usage context that can be separated into 3 categories, corresponding to their types (‘before’, ‘after’, ‘with’).

The action that is performed by the obligation can for example be specified with XACML3.0 syntax as follows:
Obligations can also be expressed with an ontology where an action is triggered under a usage and a context and that details the same pieces of information specified the XACML specification. Figure 3.3 shows the structure of an obligation and Figure 3.4 details the specification of the obligations in sticky policies.

Figure 3.3: Obligation Ontology

The processing and enforcement of obligations are detailed in Section 4.8.
Figure 3.4: Sticky Policy Ontology including Obligations
4 Data Lifecycle Security Management: General Architecture

4.1 Data Gatekeeper Building Blocks

The Data Gatekeeper is composed of 6 logical components (see Figure 4.1).

4.1.1 Data Protection Manager

The Data Protection Manager is responsible for the registration of the Data Processors (i.e., service providers) that need to use the personal data or data to be protected. It collects the type of data that is needed by the Service Providers and their usage. This information is stored in Data Protection Contract. The Data Protection Contracts between the Data Gatekeeper and the Service Provider are generated, signed, and stored in the Data Protection Manager component. The organizational access control policies to the services can be delegated to the Data Gatekeeper, in order to be enforced.

4.1.2 Sticky Policy Manager

The Sticky Policy Manager component is responsible for the registration of the Data Subject. It collects both the data to protect and its owner security preferences (consent for processing, etc.). The Sticky Policy
Manager component translates the data subject requirements/preferences into Sticky Policies, and bounds them to the personal data. It is responsible for signing and storing the Sticky Policies.

### 4.1.3 Data Protection Decision Point

The Data Protection Decision Point component is responsible for combining the various organizational access control policies and sticky policies in order to grant or deny the access/processing of the data. It consults the other components to create a context around the request, and takes a final decision. The decision is forwarded to the Privacy Enforcement Point component (see below), that will enforce the decision, and forward the data/filtered data/deny response to the Service Provider, requesting the data.

### 4.1.4 Data Protection Enforcement Point

The Data Protection Enforcement Point component is responsible for intercepting the request for personal data from Services (Data Processors), forwarding the request to the Data Protection Decision Point component and applying the decision made by the Data Protection Decision Point component. A Data Protection Enforcement Point is locally deployed for each Service, in front of each DataBases storing personal information. It is possible for a Service to embed several Data Protection End Points. There shall be no way of connecting to the DataBase bypassing the Data Protection Enforcement Point.

### 4.1.5 Service Registry

The Service Registry component is responsible for recording the security properties/mechanisms provided by a service that is registered in the Data Gatekeeper. It records security measures, which are implemented or used by the service. The service registry specifies the goal of the security measure, how it is implemented (i.e. by which mechanism), and by which technology. For example, confidentiality may rely on an encryption mechanism based on an AES encryption algorithm.

### 4.1.6 AuthN

The AuthN component is responsible for authenticating a data consumer requesting data through a service. When a service registers towards the Data Gatekeeper, it may delegate its authentication to the Data Gatekeeper. This delegation of authentication is considered as a security mechanism in the Service Registry, allowing a better level of trust in the Service. When a data consumer logsins on a service with authentication delegation, he is directed to the AuthN component. He logsins on the AuthN component and is redirected to the service, being authenticated. The AuthN component is an authentication service, using an implementation of the OpenID connect protocol for instance.

### 4.2 Data lifecycle security management process

#### 4.2.1 Data processor registration

A Service Provider needs to register to the Data Protection Manager by specifying the information that is necessary for the definition of a data protection contract for each service he provides. The Data Protection Manager generates a machine readable Data Protection Contract that is signed before being stored in a triple store (see Figure 4.2).

During the registration phase, the Data Protection Manager redirects the Service Provider to the authentication component for a potential delegation of authentication.
4.2.2 Data subject registration

The data subject connects to the Data Gatekeeper via a User Interface that allows him to visualize the Data Protection Contracts between the Service Providers and the Data Gatekeeper (see Figure 4.2). The Data Subject is able to know which data a service provider needs, and what kind of processing it does. Based on this information, the Data Subject can specify his preferences on the usage of the data. These preferences specify the services that can access his data and the conditions under which it can be accessed or processed. If the Data Subject preferences are not compliant with the needs of any of the Service Providers, then the data cannot be transferred to the service provider. If the data subject requirements are compliant, the sticky policy manager translates the data subject security preferences into machine readable Sticky Policies (see more details in section 5) that will be enforced each time a Service Provider will request the data. The generated Sticky Policies are signed and stored in a triple store. They are logically bounded to the personal data. The Data Subject can choose to transfer its data to the service provider. This transfer is intercepted and enforced by the Data Protection Enforcement Point in order to check its compliance with the sticky policies.

4.2.3 Data Usage

After a valid authentication, the end-user requests for data, in compliance with the data protection contract of the Service. The Data Protection Enforcement Point is placed as a reverse proxy, in front of the data storage place, where personal data is stored. The Data Protection Enforcement Point intercepts the request and forwards it to the Data Protection Decision Point. A request for personal data is given by the tuple \( r = (s, c, d, u) \), where \( s \) is a proof of authentication of the Service used by the end-user, \( c \) is a proof of authentication of the end-user itself, \( d \) is the data that is requested, and \( u \) is the planned usage of the requested data. In a first step, the Data Protection Decision Point asks the AuthN component to check the authentication of both the Service and the Data Consumer see Figure 4.3. If one of the Authentication proofs is not
valid, access is denied. If the Authentication proofs are both valid, the evaluation process can continue. In a second step, the Data Protection Decision Point uses combination algorithm to provide an access decision to the Data Protection Enforcement Point based on the contract, data subject preferences and organizational access control policies. The Privacy Decision Point will deliver either a Deny access or a Grant access to the enforcement point with a list of paths to data to be applied by the Data Protection Enforcement Point.

Figure 4.3: Data Usage

4.3 Data protection decision point: DPDP

A request for personal data, made by a Data Consumer, is given by the tuple \( r = (s, c, d, u, ctx) \), where \( s \) is a proof of authentication of the Service used by the Data Consumer, \( c \) is a proof of authentication for the Data Consumer, \( d \) is the data that is requested, \( u \) is the planned usage of the requested data, and \( ctx \) is the Context in which the request is made. With a request for personal data, described above, as input, the Data Protection Decision Point will deliver either a Deny access or a Grant access with a list paths for accessing the data ontology (a list of SPARQL Property Paths for our implementation). In order to achieve this goal, the Data Protection Decision Point runs sequentially the following steps see Figure 4.4:

Step 1: check if the authentications are valid. The Data Protection Decision Point relies on an internal or external authentication service (AuthN) to check the authentication of both the service and the end-user. If any of the Authentication proofs is not valid, a deny is returned. If the authentication proofs are both valid, the DPDP goes to step 2.

Step 2: check the validity of the requested usage. The Data Protection Decision Point extracts the Data Protection Contract associated to the authenticated service and requests the Data Protection Manager to check if the requested usage of the data is specified in the Data Protection Contract. If the planned usage is found in the Data Protection Contract, the DPDP extracts the list of Sticky Policies associated to the planned usage \( u \) and goes to step 3. If not, the data access is denied.

Step 3: extract the context. The Data Protection Decision Point fetches a context in the sticky policy that matches the context, \( ctx \) and goes to step 4 by requesting the Sticky Policy Manager to extract the corresponding Sticky Policies and Obligations.

Step 4: extract the Sticky Policies and Obligations.
The Sticky Policy Manager uses ctx as input to extract the list of Sticky Policies to apply and the list of Obligations to enforce. The Data Protection Decision Point then forwards the ‘before’ Obligations to the Obligation Management Service that requests the corresponding trusted service. This obligation service enforces the action and forwards a response to the Data Protection Decision Point about the success or failure of the enforcement. If one of the action specified in the Obligation fails, the Data Protection Decision Point delivers a ‘deny’ response. Otherwise it goes to step 5.

Step 5: combination of the Sticky Policies. The Data Protection Decision Point uses a combination algorithm to combine the various Paths extracted from the Sticky Policies. The combination algorithm is an intersection or union of the sticky policy paths to apply. It outputs either a ‘deny’ response or a ‘grant’ response. The Data Protection Decision Point sends the response, along with the aggregation of sticky policy paths. It also sends the ‘with’ and ‘after’ Obligations.

Once the evaluation process is finished, the Data Protection Enforcement Point applies the Sticky Policies sent by the Data Protection Decision Point (see next section).

### 4.4 Data protection enforcement point

The application of the Sticky Policies on the data to be protected results in a filtering of the authorized data. Then, the Data Protection Enforcement Point applies the original data request from the Data Consumer to the filtered dataset. The Data Protection Enforcement Point sends the filtered data to the end-user.

It is possible to use encryption techniques to enforce sticky policies [8]. In this case, if an untrusted Data Consumer bypasses the Data Protection Enforcement Point, he can only access ciphered data. Encryption techniques include generic Public Key Encryption (PKE), Identity Based Encryption (IBE), Attribute Based...
Encryption (ABE), and Proxy Re-Encryption (PRE).

### 4.5 Generic Public Key Encryption

The Data Subject (DS) wants to grant access to the data \( m \) under condition \( \text{cond} \), to Data Consumer (DC):

- DS validates the public key \( \text{pkDC} \) of the DC
- DS sends \((\text{DCid}, \text{cond}, c)\) where \( c = \text{Encrypt}(m, \text{pkDC}) \) to the Data Protection Enforcement Point (DPEP)

When DC query for the data \( m \), the DPEP:

- validates that \( \text{cond} \) is verified
- if so, sends \( c \) to the DC

With this encryption technique, the DS can easily update the condition, only informing the DPEP to update the new condition. But if the DS wants to grant access to another DC, it has to store the data \( m \) (because \( c \) can only be decrypted with the first DC secret key), and encrypts it with the public key of the new DC. Also, if the secret key of the DC is compromised, a new key pair must be generated for the DC, and the data must be decrypted and encrypted with the newly generated key pair.

![Figure 4.5: Generic Public Key Encryption](image)

### 4.6 Attribute Based Encryption

Attribute-based encryption is a public key encryption scheme. In an Attribute Based Encryption scheme, messages are encrypted with respect to subsets of attributes (key-policy ABE - KP-ABE) or policies are defined over a set of attributes (ciphertext-policy ABE - CP-ABE).

Someone should be able to decrypt a ciphertext only if the person holds a key for "matching attributes" (attributes in the policies or in the encrypting public key). User keys are issued by a trusted party. This encryption mechanism needs a Trusted Third Party that publishes the system public key \( \text{pk} \).

The Data Subject (DS) wants to grant access to the data \( m \) under condition \( \text{cond} \), to Data Consumers (DC) that satisfies an access structure \( \text{ac} \):
• DS validates the public key pk
• sends (ac, cond, c) where c=Encrypt(m, ac, pk) to the DPEP

When DC query for the data m, the DPEP:
• validates that cond is verified
• if so, sends (c, ac) to the DC

With this encryption technique, the DS can easily update the condition, only informing the DPEP to update the new condition. If the DS wants to grant access to another set of attributes, it has to store the data m (because c can only be decrypted with the previous list of attributes), and encrypts it with the new set of attributes. Also, if the secret key associated to an access structure is compromised, the rules associated with any of the attributes in the access structure may be compromised and need to be recomputed. The revocation of attributes associated to a DC is an issue, and the Trusted Third Party needs to update the attribute for each other DC. Attribute Based Encryption offer a fine grained enforcement mechanism.

### 4.7 Proxy Re-Encryption

Proxy re-encryption is a public-key encryption. In a proxy re-encryption scheme, Alice delegates a semi-trusted proxy to translate ciphertexts encrypted under her key into ciphertexts encrypted under Bob’s key. In order to be able to do that, without knowing Alices secret key, it is possible for Alice to generate a re-encryption key, using her secret key and Bobs public key as input. The re-encryption key shall not leak information about Alices secret key. The re-encryption key allows generating, with input a ciphertext encrypting a message with Alices public key, a new ciphertext that be encrypts the message with Bobs public key.

The Data Subject (DS) wants to grant access to the data m under condition cond, to Data Consumers (DC) that satisfies an access structure:
• DS validates the public key pk
• DS sends (DCid, cond, c, rk) where c=Encrypt(m, pkDS) to the DPEP and rk is the re-encryption key from the DS key pair to the DC key pair.

When DC query for the data m, the DPEP:
• validates that cond is verified
• so, sends c to the DC, where c=ReEncrypt(c, rk)

With this encryption technique, the DS can easily update the condition, only informing the DPEP to update the new condition. If the DS wants to grant access to another DC, it only has to publish the re-encryption key pk’ from the DS to the new DC, and update the tuple (DCid, cond, c, rk) to (DCid, cond, c, rk). Also, if the secret key of the DC is compromised, a new key pair must be generated for the DC, and the data must be decrypted and encrypted with the newly generated key pair.

Finally, if the DS private key is compromised, the DS must generates a new key pair, encrypt the plaintext m with the new key pair, and replace every re-encryption key, and sends the new tuples to the DPEP.

The Data Protection Enforcement Point also forwards the Obligations to the Obligation Service Management.

In the case of a ‘with’ Obligation, the Obligations action and the end-user action are simultaneous. The 2 actions should be considered as atomic; if one of the two actions fails, the other should be rolled back to the original state.

In the case of an ‘after’ Obligation, it is possible that the action defined in the Obligation failed (‘fail’), while the personal data is delivered to the Data Consumer.
4.8 Obligation execution and enforcement

When an Obligation is raised, we need a way to enforce it. An obligation can be executed by a trusted service or a micro-service that can be, for example, be deployed on a secure enclave. For instance, an Obligation can send an email to the Data Subject when his data is requested. In order to enforce such Obligation, we need a trusted email service.

For each Obligation type, a corresponding trusted service is defined. In our architecture, an Obligation Management Service can be embedded either in the Data Protection Decision Point or in the Data Protection Enforcement Point, in a trusted Environment (for instance an Intel SGX Enclave).

An Obligation Management Service that is embedded in the Data Protection Decision Point can be responsible for the ‘before’ Obligations. It receives the Obligations from the Sticky Policies, and forwards the specified action to the corresponding obligation service, that sends back a ‘fail’ or ‘success’ response according to the execution of the action.

If the response is ‘success’, the Data Protection Decision Point delivers the decision (‘grant’ or ‘deny’). Otherwise, the Obligation is not enforced, and the access is denied.

The Data Protection Decision Point forwards the ‘with’ and ‘after’ Obligations to the Data Protection Enforcement Point. The Obligation Management Service that is embedded in the Data Protection Enforcement Point is responsible for the ‘with’ and ‘after’ Obligations. It receives the Obligations from the Data Protection Decision Point, and forwards the specified action to the corresponding service, responsible for the execution of the action.

If the Obligation is an ‘after’ Obligation, the Data Protection Enforcement Point gives the Personal Data to the Data Consumer. Then, the Obligation Service forwards the action to the corresponding micro-service. So, it is possible that the action defined in the Obligation failed (‘fail’), while the personal data is delivered to the Data Consumer. In the case of a ‘before’ Obligation, it is possible that the action defined in the Obligation succeed (‘success’), while the personal data is not delivered to the Data Consumer. In the case of a ‘with’ Obligation, the Obligations action and the Data Consumers action are simultaneous. The 2 actions
should be considered as an atomic action. So, if one of the two actions fails, the other should be rolled back/reversed back to the original state.
5 End-User Policy Specification in PAYD Use Case

5.1 Introduction

With access to increasingly detailed automobile telematics journey data, insurance companies are able to shape a more accurate view of an individual drivers overall risk profile based on insights gained from empirical data analysis of their journeys.

As the volume and accuracy of the telematics data grows, an increasing amount of data processing is taking place, which itself becomes subject to privacy and usage control restrictions that up until now have been largely left to the discretion of telematics and insurance companies. With the introduction of EU data protection regulations such as the EU Data Protection Directive and its successor, the General Data Protection Regulation, telematics and insurance companies have to balance competitive innovation with regulatory compliance.

The Pay-as-you-drive automobile insurance use case represents an innovative dynamic service offering to customers, where the premium rate of an insurance offering can be balanced with the level of privacy and data protection any individual customer may choose through privacy and consent settings. They may dynamically maintain through their real-time choices to grant or withdraw informed consent to data processing for explicit purposes. As such, the PAYD use case focusses on end-user consent for the amount and type of automobile telemetry data that is collected and processed.

The PAYD use case also introduces a multistakeholder model in the data value chain wherein that a telematics company such as a 5G network provider that provides secure delivery of end-user telematic streams to the insurance service provider, may have an interest in access to subsets or aggregates of this data for the purposes of 5G network performance monitoring and capacity planning. The insurance service provider can potentially offset telematics provisioning costs by offering discounts to end-user drivers who consent for their data to be used for this purpose.

![Figure 5.1: Simplified PAYD Service Model](https://restassuredh2020.eu//)

The Pay-as-you-drive insurance service utilises the Data Gatekeeper’s Sticky Policy Manager to allow
the driver to grant or withdraw their informed consent for the usage of elements of their automobile journey telematics data for a defined specific purpose. The Data Gatekeeper’s Data Protection Enforcement Points are the control mechanism to enforce user consent and data protection across the data lifecycle.

Firstly, the telematics data stream ingress is strictly controlled by the RestAssured Data Protection Enforcement Point, ensuring that data collection and storage actions with purposes where end-user consent has been withdrawn cannot reach the service provider’s datastores. In some cases the Data Protection Enforcement Point could be implemented in the 5G Edge network hosting infrastructure as well as on board the vehicle.

Secondly, the insurance and telematics analysts’ access to each individual driver’s car journey data is restricted by the Data Protection Enforcement Point to uphold the access specified in the driver’s sticky policy, by querying the Data Protection Decision Point for the current sticky policy for each driver. Each and every analytical query for a pre-defined purpose will only return data of end-users that have (contemporaneously at query time) granted consent for the use of their data for that purpose.

5.2 Conceptual Data Model and Data Policy Framework

As illustrated in Figure 5.2, the pay-as-you-drive data model consists of four main components:

1. Person: required information about the driver
2. Automobile: required information about the automobile
3. Journey: information about each driver journey in an automobile
4. Events: Automobile telemetry readings from each car journey. These can be divided into categories, for instance
   - Geolocational (latitude and longitude)
   - Instrumentational (Speedometer, Accelerator, Brake, etc)

Figure 5.3 represents the PAYD conceptual policy model for the end user, extending the Data Gatekeeper policy specification framework in Figure 2.3.

5.3 Use Case Policy Specification Examples

5.3.1 Service Provision

The Service Provision purpose listed in Table 5.1 represents the minimum information set and actions the PAYD provider requires to provision the core insurance service.

<table>
<thead>
<tr>
<th>Description</th>
<th>Data</th>
<th>Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Provision</td>
<td>Driver</td>
<td>Process</td>
<td>Simple per kilometre insurance premium</td>
</tr>
<tr>
<td></td>
<td>Automobile</td>
<td>Store</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Events.odometer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Service provision use case policy specification example

For end-users who do not wish their location and instrumentation to be collected for any journey, the PAYD service provider offers an insurance option for a given journey based only on the number of kilometers travelled, which can be activated by the end-user consent preferences (effectively withdrawing consent for
all optional service offerings). This allows the driver to choose privacy for any journey by granting or withdrawing consent to use geo-locational and remaining instrumentational automobile telemetry.
Table 5.2: Driving behaviour risk assessment policy specification example

5.3.2 Example Optional Service Offering: Driving Behaviour Risk Assessment

The PAYD service provider offers an insurance option with premiums based on an end-user’s driver behaviour score, where a safer driving history gives reductions to the driver’s insurance premiums. In conjunction with consent for notification, this option allows for gamification of driver safety habits, where feedback can be given back to the end-user for each journey in order to increase their good driver behaviour score and reduce their premiums, which in turn reduces payout risk for the PAYD insurance service provider (see table 5.2).

5.3.3 Example Optional Service Offering: Share Journey Data with Telematics Provider

Table 5.3: Journey data third-party sharing policy specification example

The PAYD service provider may engage a third party mobile network telematics provider for delivery of automobile telemetry data, and may offer premium reductions if end-user drivers consent to having geolocation and other relevant data accessible for analytics to the third party for the purposes of network telematics quality control and network capacity planning (see table 5.3).
6 Conclusion

One of the main results of WP6 in the first period is a methodology for data lifecycle security management that takes into account the requirements of the GDPR and that considers the data subject consent and information as a priority. A conceptual model and an ontology structure for data protection lifecycle policies based on this methodology were detailed. A first architecture of the Data Gatekeeper was defined with a clear process related to the usage (during the lifecycle) of data and its enforcement. A context-based access/usage control model based on ontologies was used for more openness, flexibility and reasoning dynamism as well as support of heterogeneity and distributed environments. A first implementation of the Data Gatekeeper based on the architecture and the methodology described in this deliverable is available.

The second period will be focused on the integration of the adaptation/risk assessment loop with the Data Gatekeeper, in order to take the security properties of the environment and the risk level as a contextual input in the data lifecycle decision process. A priority will be given to the data subject observation of data protection, adaptations and changes as well as the auditability and compliance checking of sticky policies. We plan to further complement the use cases with integration of a UI and API focusing on consent management with the RestAssured Data Gatekeeper Sticky Policy API to provide a seamless and easy mechanism for service end users to dynamically grant or withdraw consent to entities over the usage of data for defined purposes. As part of the Consent Manager UI, standards-based consent receipts are generated in accordance with the Kantara Initiative’s Consent Receipt Specification[1]. Consent Receipts are notable in that they include the codification of purpose (or purposes) to which the data subject has provided consent, which in turn must be addressed by both the policy specification and the Data Protection Contract. From an audit and compliance point of view, consent receipts can be stored by the end-user as an official record of their own consent history, and used as a basis for ensuring that the service provider is acting in accordance with the consent provision. Service providers and associated data processors (to the extent permitted) are also able to store consent receipts for their own purposes, such as for supporting data protection audits and demonstrating that any data processing activities have been carried out in line with the data subject’s explicit consent. As a standard definition, the ability to read and map consent receipts by the policy specification and enforcement points will also facilitate the increased interoperability of RestAssured services within the growing ecosystem of GDPR-enabling technologies.

Bibliography


