Towards a Declarative Approach to Stateful and Stateless Usage Control for Data Protection

Francesco Di Cerbo[1], Fabio Martinelli[2], Ilaria Matteucci[2] and Paolo Mori[2]
[1]*Security Research, SAP, France*
[2]*IIT-CNR, Italy*

Keywords: Personal Data Protection, GDPR, Privacy, Security.

Abstract: Virtually any online website or service has a rising need for data protection mechanisms, especially for personal data, considering initiatives such as the new General Data Protection Regulation to operate on the EU economic space, or the Cybersecurity Law for the Chinese market. It seems therefore necessary to dispose of mechanisms that help both users, as well as legal experts and practitioners to automatically manage the processing of personal and sensitive data in a secure and compliant manner, to reduce the probability of human errors. To this aim, we show here our initial proposal for an automatically enforceable policy language, UPOL, for access and usage control of personal information, aiming at transparent and accountable data usage. UPOL extends and combines previous research results, U-XACML and PPL, and it is part of a more general proposal to regulate multi-party data sharing operations. A use case is proposed, considering challenges brought by the new EU's GDPR.

1 INTRODUCTION

The advent of the new European General Data Privacy Regulation (GDPR Regulation (EU) 2016/679) (European Parliament and Council, 2016), entered in force in May 2018, changed the legal framework and the requirements for data processing of European citizens. Such changes are applicable for all entities, anywhere in the world, that process EU personal data. At least the same magnitude of changes are requested to actors operating on the Chinese market, according to the prescriptions recently (Bird and LLC, 2018) published in the framework of the Chinese Cyber Security Law (Standing Committee of the National People's Congress, 2017). This imposes, to all such entities, modifications in their processes and likely in their software to be compliant with the new prescriptions.

In our work, we focussed in particular on a number of key requirements of GDPR that are also similarly prescribed by the Chinese law. These requirements are formalised in Articles 5 and 25as follows: *"lawfulness, fairness and transparency"*, *"purpose limitation"* and *"data minimisation"*.

The *"lawfulness, fairness and transparency"* principle refers to the mandated requirements for a data processing, taking the point of view of the data subject (i.e., the owner of the personal data). The *"purpose limitation"* principle requires data processing to take place only for pursuing predetermined objectives (or purposes), explicitly accepted by the data subject (explicit consent) with a contract prepared under the fairness and transparency principles. *"Data minimisation"* imposes to *data controllers* (e.g., companies collecting personal data of customers) to reduce the amount of collected data to the strict minimal necessary to carry out the requested service, also reducing "the extent of processing, the period of their storage and their accessibility".

In the past, a number of research results had similar objectives. In particular, we observed that the security features of *access* and *usage control* seem to fulfill the data minimisation requirement in the part that refers to computation purpose verification, data retention and data access. Access control refers to the ability of permitting or denying requests to access a resource, according to a predetermined policy or configuration. Multiple access control models exist, from access control lists where authorized requesting subjects, operations and resources are explicitly mentioned, to the more complex Attribute-Based Access Control (ABAC) where rules are expressed on the basis of attributes of such entities (e.g. for requesting subjects, their role in a company). Access control

Di Cerbo, F., Martinelli, F., Matteucci, I. and Mori, P.
Towards a Declarative Approach to Stateful and Stateless Usage Control for Data Protection.
DOI: 10.5220/0006962503080315
In *Proceedings of the 14th International Conference on Web Information Systems and Technologies (WEBIST 2018)*, pages 308-315
ISBN: 978-989-758-324-7

ends its role when access is granted or denied; usage control on the contrary, deals with the observation of the usage of a resource that a requestor does, interrupting or reacting dinamically to requestor's actions according to predetermined directives.

Our activity identified two solutions, whose combination would allow a data controller to achieve a transparent usage of personal data. T hey are PPL (Trabelsi et al., 2010; Di Cerbo et al., 2015) and U-XACML (Lazouski et al., 2012). The novelty of our proposal consists of an approach (policy language and reference architecture) that unifies both benefits and allows to achieve new results, namely:

R1: to meet GDPR's transparency requirement by fully controlling information processing operations. This is achieved through the full support of the Usage Control model proposed by Park and Sandhu (Park and Sandhu, 2004; Zhang et al., 2005), achieved through our contribution as simple extension to a well-known access control standard, XACML (OASIS, 2010)

R2: the control and tracking of processing purpose(s), for the operations requesting an access to the protected pieces of information, both at the moment of the access request and during their consumption. This aims at meeting GDPR's purpose limitation,

R3: the support for GDPR's data minimisation, considering for example data retention conditions.

In summary, we propose a language powerful enough to express legal, data protection and usage control policies in several contexts, such as the compliant collection and usage of a resource in a cloud environment and on mobile devices. Its adoption can be part of an effort to achieve GDPR compliance in an infrastructure.

The paper is structured as follows: next section describes the motivation of this work that is mainly part of a EU FP7 project named Coco Cloud about the sharing of sensitive data through the Cloud, and how parts of this effort are continued in project C3ISP. Section 3 briefly recalls the existing access and usage control policy languages while Section 4 presents the new one, named UPOL able to integrate and enhance in a unique language the expressiveness of both languages. Section 5 presents a use case for an UPOL policy and finally Section 6 draws the conclusion of the paper and discusses about ongoing and future work.

2 USAGE CONTROL IN Coco Cloud AND C3ISP

The Coco Cloud project (Coco Cloud Consortium, 2016) aims at allowing cloud users to securely and privately share their data in the cloud, to increase the trust of users in the cloud services and, consequently, to increase their widespread adoption. Since companies or public bodies are involved in the data sharing, this sharing must be regulated by real digital contracts, called Data Sharing Agreements (DSA), which must be paired with data and enforced every time the data are used (Caimi et al., 2015). Besides enforcing the constraints defined by the sharing parties, a goal among the Coco Cloud project goals, is to address also key challenges for legally compliant data sharing in the cloud. Hence, the project places an early emphasis on understanding and incorporating legal and regulatory requirements into the data sharing agreements which are paired with the data. The European data protection legal framework has been one of the key legal focuses of our work.

Since the factors that are taken into account in the data sharing agreements are mutable, i.e., they can change over time, the policies paired with the shared data follows the Usage Control model (Park and Sandhu, 2004). In this way, the usage control policy is continuously evaluated during the data access time, and the access can be interrupted when the policy is not satisfied any more. For instance, the DSA could state that a subject can read the data only when she is located within a given area (e.g., the building of a company). Hence, a subject could open the document on her mobile phone when she is located within the building but, as soon as she exits the building, the usage control policy (gracefully) interrupts the document reading (for instance, it closes such document saving the unsaved changes in a temporary document copy). An obligation may also be attached, so that, when the user leaves her country, the data is automatically deleted from the device.

Moreover, the DSA also requires that some actions, called obligations, must be executed as a consequence of the execution of other actions or when certain events occur. For instance, the data sharing agreement of a piece of data could include an obligation, which requires that the file is deleted after a given date. This concept has clear application in expressing a personal data retention period, according to the GDPR. These actions can be naturally expressed through obligations by adopting the Usage Control model.

Hence, in this paper we describe *UPOL*, the language we defined for expressing the enforceable version of the usage control policies representing the Coco Cloud DSAs.

The usage control model allows to define very expressive data sharing agreements which can be successfully applied in many scenarios. For example, we applied it to e-government, corporate and healthcare (Caimi et al., 2015) solutions. Moreover, part of the described approach is used in C3ISP[1], an EU-funded H2020 project focussing on cyber security and in particular on the sharing of particularly critical pieces of information, that are necessary for organising an effective defense against online attacks but that may also be used, if in wrong hands, to conduct malicious activities. This is the case of information describing cyber attacks trace logs: other defenders may tune up their countermeasures in order to identify the latest attacks (malware received per email, specially crafted web requests targeting a software's vulnerability etc) but on the other hand, an attacker may get to know where and how a target system is vulnerable and may be successfully breached. Specific extensions are being studied and they will be discussed in the future.

3 BACKGROUND

As mentioned, we aim at achieving the fulfillment of requirements **R1, R2** and **R3**. We identified the usage control model as the theoretical underpinnings for our objective but we observed a number of limitations in the current approaches. Essentially, we looked at a number of declarative solutions (i.e. controllable through a configuration policy) and we concentrated on three technologies that have available implementations. They are:

- **XACML** it is an OASIS standard with several open source and commercial implementations. The main limitation is that it only covers access control models and not usage control. It could only partially fulfill R1, R2 and R3.
- **U-XACML** extends XACML in order to bring some usage control functionalities, most notably the continuous verification of access control conditions during processing of requested resources. It can be used to fulfill R1, partially R2 but not R3.
- **PPL** as well extends XACML with the possibility to verify resource processing purposes against a policy plus it caters the automatic execution of obligations defined by a resource owner. It can be used to implement R3 especially with respect to data retention, R2 but not completely R1.

From these findings, we decided to work on a new concept, UPOL. It extends XACML, combining the advantages brought in by two other extensions, U-XACML and PPL, later described more in depths. From their combination, UPOL achieves new capabilities as detailed in Sections 4. It is the language we used to express DSAs terms and conditions in a machine-enforceable manner. UPOL policies therefore regulate the usage of the data they are paired with: following the sticky policy model (Pearson and Casassa Mont, 2011), each policy (that regulates the access to a resource) get attached to a resource to form a *bundle*, normally protected by means of strong encryption. This imposes that data can be processed only by means of special mechanisms able to decrypt the bundle and to allow its access in accordance to the associated policy. Any attempt to consume arbitrarily a resource once it is protected by such bundle, is destined to fail. The UPOL language is based on the Usage Control model, which extends traditional access control model by dealing with attributes related to the subjects and of the objects which change their values over time. The Usage Control model allows to define policies which are continuously evaluated during the execution of an access, in order to revoke such ongoing access when the corresponding policy is not valid any longer. In particular, the usage control policies define authorization and condition rules which must be satisfied before and/or during the usage of such data (*pre-/ongoing-authorizations* and *pre-/ongoing-conditions*), along with obligations (similarly, *pre-/ongoing-obligations*). UPOL comprises all such categories, extending the XACML capabilities with two new contributions the *asynchronous* and *synchronous* obligations, normally implemented by a trusted third party. The asynchronous obligations are usage control obligations which have to be fulfilled when an event occurs. Events may used to model reactions to mutable attributes as in Park and Sandhu model but, extending it, they may not be connected to an access request, as such as when the retention period for a piece of data expires (as requested for GDPR's data minimization). The synchronous obligations are again usage control obligations. For instance banners that appears while one watches a streaming video, or logging of the exact consumption time of a resource, for accountability purposes. Such kind of obligations may also be considered as *session* obligations and can be used in order to pinpoint when an user starts and terminates to use a resource as well as when the user's access right to the resource is revoked.

[1] homepage: https://c3isp.eu/

In UPOL, the violation of pre-authorization or pre-condition rules prevents the access to the protected data. Instead, when pre-authorization and pre-condition rules are satisfied, the user can access the data, and the ongoing-authorization and ongoing-condition rules are enforced continuously while the access is in progress. In this case, the violation of ongoing rules interrupts the usage of the data. Session obligations may be associated to passed or failed checks, in order to model a desired behavior.

3.1 U-XACML

The U-XACML language is an extension of the XACML language which includes additional constructs to express Usage Control policies. XACML is a standard developed by the OASIS consortium for expressing and managing access control policies in a distributed environment (OASIS, 2010). Briefly, the XACML standard defines a policy meta-model, syntax, semantics, and the related enforcement architecture. The top-level element of a policy is <*PolicySet*>, which includes a set of <*Policy*> elements (or other distinct <*PolicySet*>), each of which, in turn, includes a <*Target*>, which denotes the target of the policy, and a set of <*Rule*> elements which represent the authorization rules. A rule is defined by three main components: the <*Target*>, the <*Condition*>, and the effect of the rule which can be either Permit or Deny. The <*Target*> denotes the target of the rule, i.e., to which authorization requests the rule can be applied. The <*Condition*> elements are predicates evaluating the attributes. Optionally, a rule can include also the <*ObligationExpression*> element. A rule is applicable to an access request if the target of the access request matches the target of the rule and if all the conditions included in the rule are satisfied. If a rule is applicable to an access request, the effect declared for the rule determines whether the access request is permitted or denied, and the related obligation must be executed.

XACML allows to express traditional access control policies dealing with immutable attributes and it does not have specific constructs to express the continuity of policy enforcement. Hence, the U-XACML language extends XACML with some constructs for usage control as follows. To represent the peculiarity of the Usage Control model, i.e., the continuity of policy enforcement, the U-XACML language allows to specify in the <*Condition*> element a clause, called *DecisionTime*, which defines when the evaluation of this condition must be executed. The admitted values are *pre* and *on* denoting, respectively, *pre-decisions* and *on-decisions*. In this way, the conditions whose decision time is set to *pre* are usual XACML conditions. On the other hand, the conditions whose decision time is set to *on* must be continuously evaluated while the access is in progress. We recall that XACML conditions are exploited to represent both UCON authorizations and conditions. In the same way, U-XACML extends the <*ObligationExpression*> element with the *DecisionTime* clause to define when the obligation must be executed. In this case too, the admitted values for the *DecisionTime* clause are: *pre* (*pre-obligations*, i.e., usual XACML obligations) and *on* (*on-obligations*), and *post* (*post-obligations*).

To deal with mutable attributes, U-XACML introduces a new element, <*AttrUpdates*>, which represents the attribute updates in the policy. This element includes a number of <*AttrUpdate*> elements to specify each update action. Each <*AttrUpdate*> element also specifies when the update must be performed through the clause *UpdateTime* which can have one of the following values: *pre* (*pre-update*), *on* (*on-update*), and *post* (*post-update*). U-XACML language, please refer to (Colombo et al., 2010).

3.2 PPL

PPL (Primelife Policy Language) is another XACML extension that allows to express personal data handling and credential capabilities. It achieves this result by providing access and usage control functionalities. In particular it was designed for modelling personal data exchanges between data subjects and data controller , according to definitions provided by the European Data Protection Directive 95/46/EC. PPL adopts the sticky policy approach: once data handling terms have been agreed between data subject and controller, a policy gets associated to the personal data given to the data controller. This policy cannot be detached from the data and regulates each usage. One notable aspect is the expression of usage control obligations. Normally in XACML, an obligation must be fulfilled by the actor that issues an access request. In PPL, they are defined as “a promise made by a data controller to a data subject in relation to the handling of his/her personal data. The data controller is expected to fulfill the promise by executing and/or preventing a specific action after a particular event, e.g. time, and optionally under certain conditions”. PPL obligations may also apply to data processors, i.e., entities authorized by data controller to carry out computations on the data, under the responsibility of the data controller. PPL obligations are expressed as follows:

$$\textbf{Obligation} = \textit{Do}\ \textbf{Action}\ \textit{when}\ \textbf{Trigger} \tag{1}$$

where

$$\textbf{Trigger} = \textbf{Event} \wedge \textbf{Condition} \tag{2}$$

Obligations modelled in this way are not dependent on access requests and therefore differ from access control obligations. They can be used to express a data retention period, e.g., data must be deleted by the data controller after 30 days from their submission. As shown in (Di Cerbo et al., 2015), triggers may also depend on contextual conditions like geographic location.

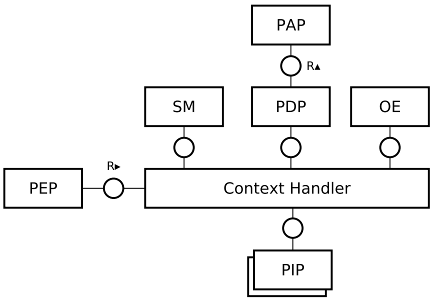

Figure 1: The UPOL Reference Architecture.

4 UPOL

As mentioned, the UPOL language originates from XAMCL, adding statefulness to its interaction model. To cater for that, a number of contributions are proposed, in terms of reference architecture and language syntax, explained in the remainder of this section.

The UPOL reference architecture originates from XACML, mutuating many of the main components for its enforcement engine (depicted in Fig. 1):

PEP Policy Enforcement Point intercepts the requests to perform security relevant actions, triggers the decision process and enforces the results. PEP is normally integrated in a client application.

CH Context Handler coordinates the internal components of the enforcement engine.

PDP Policy Decision Point evaluates requests (also continuously) against UPOL policies.

PAP Policy Administration Point is in charge of retrieving applicable policies for PDP evaluation.

PIP Policy Information Point(s) retrieves all the necessary attributes of any actor (subject, action, resource, environment) for enabling PDP evaluation.

The new components, respectively inherited by U-XACML and PPL are:

SM Session Manager is responsible for keeping track of the ongoing usage sessions, i.e., of the access that are currently in progress, and it is responsible to store the meta-data regarding these sessions. It is the key component of the continuous authorization phase, enabling the PDP to operate continuously.

OE Obligation Engine is responsible for keeping track of obligation triggers and executing the associated action(s).

In UPOL, the CH has also the role to exchange status information among SM and OE, in order to allow session obligations to be triggered.
The UPOL architecture is necessary to implement its new obligations; they go beyond the capabilities of XACML/U-XACML and PPL. Table 1 enumerates UPOL obligations as extension of XACMl/U-XACML, PPL and the newly defined UPOL obligations. U-XACML inherits the standard XACML obligations, they are pre-, or post-obligations associated with an access request.

Instead, PPL obligations are not (necessarily) bound to an access request (see Formula 1). PPL supports triggers (see Formula 2) like proximity to a specific geographic location, but also data deletion or time interval expiration (Di Cerbo et al., 2015). A trusted third party, different from Data Subject and Data Controller and trusted by both actors, is in charge of automatic obligation enforcement. Lastly, UPOL obligations can be defined leveraging the notion of session: UPOL defines three new event types *StartAccess*, *EndAccess* and *RevokeAccess* that can be part of UPOL triggers. Therefore, UPOL obligations may prescribe the execution of actions associated with a session work-flow. A UPOL obligation can start during a data consumption operation and terminate at its end. We recall that a session is created when an access request is approved and the requestor (or an agent) notifies the beginning of a resource consumption. *EndAccess* obligations are triggered when the requestor interrupts a resource consumption operation. *RevokeAccess* on the contrary takes place when a policy violation is detected and a session is interrupted by initiative of the access control mechanism. UPOL obligations also differentiate from other obligations in that they can be used to execute continuous actions, particularly helpful in streaming scenarios: like showing banners in streaming videos or in Big Data streaming Analytics computation. Last but not least, obligations differentiate also with respect to the enforcement actor. XACML/U-XACML obligations are normally enforced at the requestor's end, by the PEP. PPL obligations, instead, are enforced by a third party (for example, a cloud provider) trusted by data subject and data controller. UPOL obligations are triggered by the Obligation Engine run by a trus-

Table 1: Obligations part of an UPOL policy.

Obligation Type	Reference Event	Obligation Action Type	Obligation Enforcement
U-XACML pre- or post-obligations	at the moment of the request	punctual actions	PEP
PPL obligations	dependent or independent from access request	punctual actions	trusted third party
UPOL session obligations	at the beginning, end or during data consumption	punctual or continuous actions	trusted third party or PEP

ted third party but they may be also executed by the PEP, according to the associated actions.

4.1 Comparison with UCON_ABC

The Park and Sandhu model (Park and Sandhu, 2004) defines a number of core usage control models, organised according to their characteristics with respect to:

1. access control constituents, namely authorizations (*A*), obligations (*B*) and conditions (*C*);
2. the continuity of the decision: is the access control decision made when a request is received (like in standard AC, *pre*) or its validity is continuously evaluated during object consumption (typical trait of UC, *ongoing*)?
3. mutability of attributes: are subject or object attributes changing following to a decision? and when? This originates: *immutable*, $pre - update$, $ongoing - update$ or $post - update$ models respectively for models where no attribute changes are foreseen, updates takes place before, during or after an access takes place.

We claim that our UPOL language can describe all control models described by Park and Sandhu, by means of, respectively:

1. the native XACML constructs.
2. the constructs brought in by U-XAMCL.
3. the specific extensions offered by U-XACML combined with PPL: *pre* and *ongoing* conditions can originate specific events to trigger the newly defined UPOL obligations (*synchronous* and *asynchronous*).

Moreover, if used in conjunction with the sticky policy approach, UPOL spans its scope beyond access and usage control, as UPOL obligations can be triggered also without the reception of an access request. Such functionality is particularly helpful in cases like GDPR's data minimisation requirement, where a piece of data may reside on the Data Controller cloud only for a limited amount of time. Provided that a trusted third party runs a UPOL-aware enforcement mechanism to control the UPOL-regulated data on the cloud, such requirement may be fulfilled.

5 USE CASE

As an example, we may consider a simple online commerce scenario involving a company, *ACME* (data controller according to the GDRP), one of its customers, *Customer* as data subject and a third-party, a *Marketing* service provider that profiles customers and suggests marketing campaigns to ACME.

The use case is depicted in Figure 2. The customer submits a set of personal data, necessary for ACME to serve the customer's orders and of course, to keep track of payments, warranties etc. This exchange is regulated by recording the data subject consent to a data sharing agreement (Caimi et al., 2015) stating precisely data controller's rights and obligations, detailed as follows. Data submission and consent recording take place through a specific service, called "Usage Control Service" (the proposed contribution) in the figure, that:

- associates to the personal data, a UPOL policy (using the sticky policy model) that states their access and usage terms (the data sharing agreement);
- enforces the UPOL policy upon incoming personal data access requests;
- monitors usage of personal data, through interactions with the ACME information systems, enforcing continuous authorizations as well as usage control obligations.

When the third party uses the ACME information systems, it interacts with the enforcement system, crea-

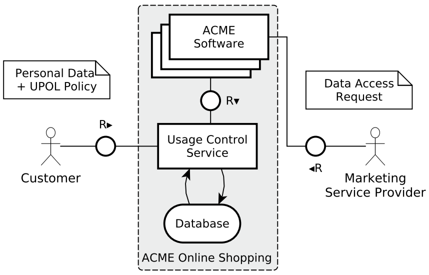

Figure 2: An eHealth use case.

ting requests to access the protected resources. Information systems cater for a number of attributes that allow their requests to be evaluated by the enforcement system, as well as providing indications about the beginning and the end of information processing operations. The interaction protocol between systems and enforcement also allows the interruption of an operation, in case of changes in the evaluation conditions.

It is out of the scope of the current work to include a detailed analysis of the interaction protocol and of the architecture of the enforcement system. Focussing instead on the policy expression, it can be modelled as a (sticky) UPOL *Policy* where a *Rule* with *Target*: *subject role = marketing-third-party, contractor = ACME* may access the data. Two conditions (*pre* and *ongoing*) control the geographic location of the doctor, provided by the information systems, in order to protect against data export clauses (GDPR Article 49). We can model the notification obligation by means of session obligations, triggered by *StartAccess*, *EndAccess* and *RevokeAccess*. In this way, the beginning and the end of sessions can be recorded for future use. Last but not least, each access of the third-party will trigger an email notification to the data subject, in order to fully meet the transparent processing requirements. It might be worth noting that other accesses performed by the data controller will not trigger such notification. Data minimization (with respect to retention directives) is also enforced by means of a specific obligation to delete the data associated to the policy after 3 months from the moment when the personal data is received. The UPOL representation of such policy can be found at Listing 1.

Listing 1. UPOL Use Case.

```

    <!-- DETAILS OMITTED -->
        <!-- XACML Target: ABAC check for
 ↪ attribute 'marketing-third-party' of
 ↪ requestor
          in the authentication system, i.e.
 ↪ role == 'marketing-third-party' -->
    <!-- Continous Authorization: requestor must
 ↪ be in EU to process the information
 ↪ -->
    <upol:Condition DecisionTime="ongoing">
        <xacml:Apply FunctionId="
 ↪ urn:oasis:names:tc:xacml:1.0
 ↪ :function:string-equal">
            <xacml:Apply FunctionId="
 ↪ urn:oasis:names:tc:xacml:1.0
 ↪ :function:string-
        one-and-only">
                <xacml:AttributeDesignator
                    AttributeId="
 ↪ urn:oasis:names:tc:xacml:1.0
 ↪ :subject:subject-location"
                    Category="
 ↪ urn:oasis:names:tc:xacml:1.0:subject-
 ↪ category:
            access-subject"
                    DataType="http://www.w3.
 ↪ org/2001/XMLSchema#string"
                    MustBePresent="true">
                </xacml:AttributeDesignator>
            </xacml:Apply>
            <xacml:AttributeValue DataType="
 ↪ http://www.w3.org/2001/XMLSchema#
 ↪ string">
        EU</xacml:AttributeValue>
        </xacml:Apply>
   </upol:Condition>
    <ob:ObligationsSet
       xmlns:ob="http://www.primelife.eu/ppl/
 ↪ obligation">
       <ob:Obligation>
          <ob:TriggersSet>
        <!-- UPOL Session Obligations -->
            <upol:TriggerRuleEvaluated
 ↪ FulfillOn="StartAccess" />
        <upol:TriggerRuleEvaluated FulfillOn="
 ↪ EndAccess" />
        <upol:TriggerRuleEvaluated FulfillOn="
 ↪ RevokeAccess" />
          </ob:TriggersSet>

          <!-- Action: notify data subject
 ↪ -->
          <ob:ActionNotifyDataSubject>
            <ob:Media>Mail</ob:Media>
            <ob:Address>customer.
 ↪ email@email.provider</ob:Address>
          </ob:ActionNotifyDataSubject>
       </ob:Obligation>
     <!-- Obligation: delete after 3 months
 ↪ from information received -->
     <ob:Obligation>
        <ob:TriggersSet>
           <TriggerAtTime>
              <MaxDelay>
                <Duration>
 ↪ P0Y3M0DT0H0M0S</Duration>
              </MaxDelay>
           </TriggerAtTime>
        </ob:TriggersSet>
```

```
            <ob:DenyAllAndDeleteNow/>
        </ob:Obligation>
    </ ob:ObligationsSet>
```

6 CONCLUSION

Data protection, especially for personal information, has a growing attention and demand. For example, the EU General Data Privacy Regulation require significant changes in the way entities collect personal data of EU citizens, anywhere in the world. In this paper we presented our effort towards the expression of data protection policies to achieve compliance as combination of access and usage control measures. Our policy language, UPOL, was developed in the context of the EU FP7 Coco Cloud project. Its main aim is to obtain a unique language that is powerful enough to express, legal, security, and privacy constraints in automatically enforceable policies, focussed on the sharing and management of (personal or otherwise sensitive) data over the Cloud. Now, our development continues in the EU H2020 C3ISP project, extending data protection also to Big Data analytics scenario, especially considering cyber security information sharing.

Our preliminary results, with obligations enforced automatically by the policy engine are promising especially with respect to the fulfillment of some data controller obligations as stated by the GDPR. We are currently working towards structuring and extending more our language, in order to support more data protection use cases, looking at personal data but also at more in general, confidential information especially in the cyber security domain.

ACKNOWLEDGEMENTS

This work was partly supported by EC-funded projects Coco Cloud [grant no. 610853] and by C3ISP [grand no. 700294].

REFERENCES

Bird and LLC, B. (2018). China cybersecurity law update: Personal information national standards officially published. https://www.twobirds.com. Accessed: 2018-06-20.

Caimi, C., Gambardella, C., Manea, M., Petrocchi, M., and Stella, D. (2015). Legal and technical perspectives in data sharing agreements definition. In Berendt, B., Engel, T., Ikonomou, D., Métayer, D. L., and Schiffner, S., editors, *Privacy Technologies and Policy - Third Annual Privacy Forum, APF 2015, Luxembourg, October 7-8, 2015, Revised Selected Papers*, volume 9484 of *Lecture Notes in Computer Science*, pages 178–192. Springer.

Coco Cloud Consortium (2016). Coco Cloud website. http://www.coco-cloud.eu.

Colombo, M., Lazouski, A., Martinelli, F., and Mori, P. (2010). *A Proposal on Enhancing XACML with Continuous Usage Control Features*, pages 133–146. Springer US, Boston, MA.

Di Cerbo, F., Some, D. F., Gomez, L., and Trabelsi, S. (2015). PPL v2.0: Uniform data access and usage control on cloud and mobile. In Matteucci, I., Mori, P., and Petrocchi, M., editors, *1st IEEE/ACM International Workshop on TEchnical and LEgal aspects of data pRIvacy and SEcurity, TELERISE 2015, Florence, Italy, May 18, 2015*, pages 2–7. IEEE Computer Society.

European Parliament and Council (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation). 27 April 2016 - http://goo.gl/LfwxGe.

Lazouski, A., Mancini, G., Martinelli, F., and Mori, P. (2012). Usage control in cloud systems. In Savage, N., Assad, S. E., and Shoniregun, C. A., editors, *7th International Conference for Internet Technology and Secured Transactions, ICITST 2012, London, United Kingdom, December 10-12, 2012*, pages 202–207. IEEE.

OASIS (2010). eXtensible Access Control Markup Language (XACML) Version 3.0.

Park, J. and Sandhu, R. (2004). The UCON ABC usage control model. *ACM Transactions on Information and System Security (TISSEC)*, 7(1):128–174.

Pearson, S. and Casassa Mont, M. (2011). Sticky policies: An approach for managing privacy across multiple parties. *Computer*, 44(9):60–68.

Standing Committee of the National People's Congress (2017). Cyber security law (draft). http://www.npc.gov.cn/npc/xinwen/lfgz/flca/2015-07/06/content_1940614.htm. Accessed: 2018-06-20.

Trabelsi, S., Njeh, A., Bussard, L., and Neven, G. (2010). Ppl engine: A symmetric architecture for privacy policy handling. In *W3C Workshop on Privacy and data usage control*, volume 4.

Zhang, X., Parisi-Presicce, F., Sandhu, R., and Park, J. (2005). Formal model and policy specification of usage control. *ACM Trans. Inf. Syst. Secur.*, 8(4):351–387.