

# Policy Languages

Machine-interpretable policy languages are at the heart of any modern privacy infrastructure. Rather than “hard-coding” fixed privacy policies into the infrastructure, dedicated policy languages provide the flexibility to express and change privacy policies without having to re-implement the software that enforces them. Moreover, if multiple interacting parties agree on the grammar and semantics of a language, or better even, if the language is standardized globally, policy languages can also be used to communicate privacy policies across different interacting entities. Finally, security and privacy policy languages are an important tool to ensure compliance with legal, industrial, and users’ requirements.

## Policy Requirements

PrimeLife collected requirements for data handling, access control, and trust policy languages from the diverse scenarios covered by the project, and analyzed the suitability of existing policy languages to cover the privacy aspects. It quickly became clear that none of the existing policy languages covered all the needs we discovered. It also quickly became clear, however, that satisfying all of the collected requirements was far beyond the available time and budget of PrimeLife. We therefore hand-picked a number of features from the vast collection based on their potential to improve digital privacy in the real world and based on their feasibility within the boundaries of the PrimeLife project. In the following, we provide more details on some of the selected topics and their solutions.

## Downstream Usage Control

The main scenario that we consider (see figure) is one where a private user, or Data Subject, wants to access a resource hosted by a server, or Data Controller. In order to access the resource, the Data Subject has to reveal some personally identifiable information (PII) to the Data Controller. At a later point in time, the Data Controller may want to further forward the PII, e.g., to business partners or advertisers, referred to as Downstream Data Controllers here.

All participants in the interaction specify their proposed or expected treatment of PII by means of policies. The Data Controller and Downstream Data Controller have a policy specifying which information they need

from the Data Subject (access control) and how they will treat this information (data handling). The data handling policy is expressed in terms of authorizations, e.g., to use the PII for a certain purpose, and obligations, e.g., to delete the data after a certain period of time.

The Data Subject’s preferences, on the other hand, express for each piece of PII to which Data Controllers the PII can be released and how the Data Subject expects her information to be treated. These requirements may include downstream usage requirements, meaning the requirements that a Downstream Data Controller has to fulfill in order to obtain the PII from the (primary) Data Controller.

The sticky policy describes the mutual agreement concerning the usage of PII, and is the result of an automated matching procedure between the Data Subject’s preferences and a Data Controller’s policy.

The PrimeLife project developed a simple yet highly expressive language to specify privacy policies and preferences. It gives a clear view on the complex relation between access control and data handling policies, especially in the case where recursive downstream usage is taken into consideration. Two different automated matching procedures have been designed: proactive matching, which already takes the full chain of Downstream Controllers and their policies into account at the moment that PII is revealed, and lazy matching (depicted in the figure), where the downstream policies are only matched at the moment that the PII is forwarded.

## Privacy-Preserving Access Control

Users commonly reveal much more personal data than necessary to obtain access online resources, even though existing cryptographic solutions, in particular

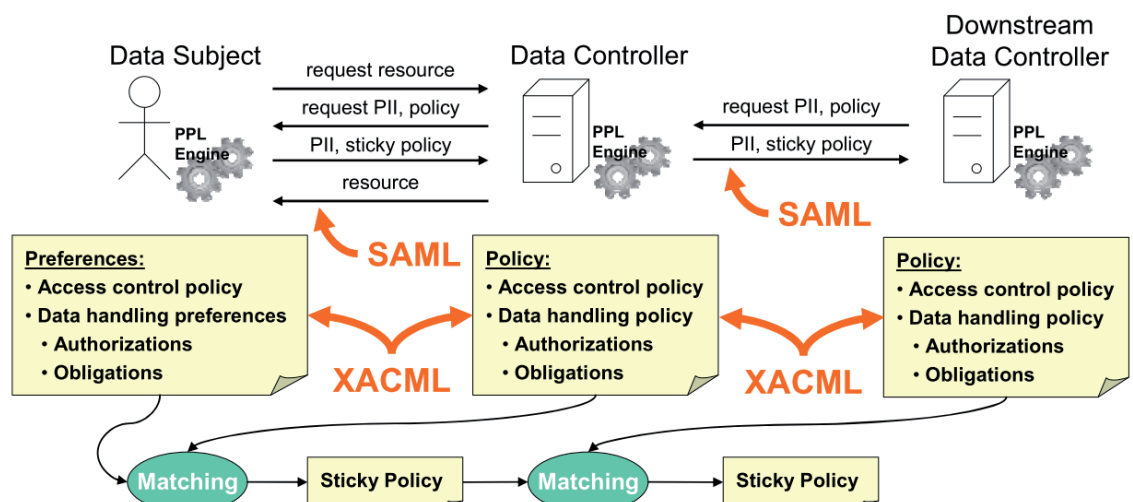
anonymous credentials, offer privacy-preserving alternatives. One of the reasons for the slow adoption is the lack of policy languages that can express the advanced functionalities of anonymous credentials. The PrimeLife project developed an access control language that addresses exactly this need, without however sacrificing compatibility with other, less privacy-preserving technologies such as X.509 certificates or LDAP directories.

The language is based on the generic model of a credential as a bundle of attribute-value pairs signed by an issuer. The decision whether access is granted to a requester is then dependent on the possession of (possibly multiple) credentials that fulfill certain requirements specified in the access control policy. For a user to obtain access to a protected resource, she produces a verifiable claim containing cryptographic evidence that she satisfies the policy.

Basing the access control decision on the possession of credentials is not new. However, the use of anonymous credentials provides support for advanced features such as consumption control, selective disclosure of credential attributes, disclosing attributes to third parties, and proving predicates over attributes without revealing the full attribute values.

## Policy Dialog Management

Classical access control policy languages often require Data Subjects to reveal all of their attributes, so that the Data Controller can evaluate the policy and decide to grant or refuse access. The privacy-preserving access control language described above relies on a dialog management infrastructure allowing requesters to first learn the access control policy they need to satisfy, and, based on this policy, to select an appropriate set of credentials to present.



# Policy Languages

However, the precise access control restrictions may be considered sensitive information by the Data Controller, e.g., because they may compromise the Data Controller or reveal business strategies.

Clearly, there is a trade-off to be made here between the privacy of the Data Subject's PII and of the Data Controller's policy. In PrimeLife, we developed a mechanism called policy sanitization to implement this trade-off directly into the access control language. For each condition appearing the policy, the policy author can specify how that condition will be communicated to the Data Subject: in full detail (e.g., `birthdate < 1993/01/01`), specifying only the predicate (e.g., `birthdate < ?`), the required attribute (e.g., `birthdate`), or the required credential type (e.g., `passport`), or even dropping the condition altogether. Satisfying a sanitized policy will obviously force the Data Subject to reveal more information than strictly necessary; the main advantage is that it offers a more gradual approach to privacy-preserving access control.

## Privacy Extensions Such As The Privacy Dashboard

The Privacy Dashboard developed within the PrimeLife project is part of a group of three extensions for the Firefox browser providing different privacy functionality to the users.

The first instruments the practices used by websites and third parties to collect personal data and track users, as well as offering users the means to set per site preferences.

The second provides a fresh take on P3P, a standardized protocol for privacy protection on the web developed by W3C, using the vocabulary defined by P3P for machine-readable privacy policies covering information collected from HTTP requests. The policies are constrained to make it easier to provide a user interface for setting preferences, and for generating human readable descriptions of the conflicts between the user's preferences and the site's policy. The browser extension looks for a link to the site's privacy policy, which is represented in JSON (JavaScript Object Notation) for ease of processing.

The third explores the potential for privacy-enhancing web authentication using zero knowledge proofs, and is based upon the Java-based Identity Mixer library developed by IBM Research.

## Legal Policy Mechanisms

Transparency is one of the core principles of data protection legislation in Europe, beyond Europe and all around the world. The European understanding is that individuals should be aware of who knows what about them.

Often these principles are hard to enforce and, above all, make understandable to the user. The user is confronted with a multitude of different purposes, often hidden in lengthy legal text of privacy notices, especially when surfing the web.

The multitude of applications and uses of personal data are highly unstructured: no comprehensive ontology exists and no abstractions are apparent. Within the PrimeLife project, we investigated and structured the legal aspects of the processing of personal data in the specific use cases of online shopping and social networks. We concluded that many data controllers act on the assumption that it is precise enough to display the legitimate reason of the data collector on handling data as a legal basis. More precise descriptions of policies are absolutely necessary, and our research has shown that this can be done at least in the selected use cases.

## The PrimeLife Policy Language

The concepts of the above research results have been integrated in a single policy language, the PrimeLife Policy Language (PPL), and a functional engine enforcing PPL has been implemented.

To facilitate real-world adoption, we based PPL on two widespread industry standards for access control and assertion exchange, namely the eXtensible Access Control Markup Language (XACML) and the Security Assertions Markup Language (SAML). As depicted in the figure, we defined extensions to XACML so that the language can express both the Data Subject's preferences and the (Downstream) Data Controller's policies. A matching engine was implemented to generate the sticky policy based on the Data Subject's preferences and the proposed policies. The concepts of privacy-preserving credential-based access control were also embedded in XACML, with support for the Identity Mixer anonymous credential system, as well as support for policy sanitization. For the communication between the different participants, SAML was extended to carry credential-based claims and to attach sticky policies to revealed attributes.

Enforcement of downstream usage was facilitated by using a symmetric language and architecture, so that the Data Subjects use the same engine to protect their PII as the (Downstream) Data Controllers do to protect their online resources. We opted for the "lazy" matching procedure described above because of the complexity of matching XACML access control policies for downstream usage.

The privacy policies associated to PII at the communication layer (e.g., browser version, cookies, etc.) are managed by the Privacy Dashboard. Additional user interfaces were designed to let the PPL engine interact with the user for identity selection, sticky policy inspection, and the editing of preferences.

Further information can be found here: <http://www.primelife.eu/results/documents/>

## PrimeLife at a glance

### Project reference:

216483

### PrimeLife's objective:

Bring sustainable privacy and identity management to the web and develop tools for privacy-friendly identity management

### Project duration:

March 2008 - June 2011

### Partners:

15 partners from industry, academia, research centres and data protection authorities

### Total cost:

About € 15.5 Million

### Total EC funding:

€ 10.2 Million

### Funding:

The PrimeLife project receives research funding from the European Union's 7th Framework Programme.

### Contact:

Marit Hansen

t:+49-431-988-1214

f:+49-431-988-1223

[primelife@datenschutzzentrum.de](mailto:primelife@datenschutzzentrum.de)

### Date of publication of this Primer:

January 2011

### Want more info?

Various deliverables are available online: <http://www.primelife.eu/>

