SPECIAL

Scalable Policy-awarE Linked Data arChitecture for prIvacy, trAnsparency and compLiance

Deliverable D2.1

Policy Language V1

Document version : 1.0

SPECIAL DELIVERABLE

Name, title and organisation of the scientific representative of the project's coordinator:

Mr Philippe Rohou t: +33 4 97 15 53 06 f: +33 4 92 38 78 22 e: philippe.rohou@ercim.eu

GEIE ERCIM, 2004, route des Lucioles, Sophia Antipolis, 06410 Biot, France

Project website address: http://www.specialprivacy.eu/

Project	
Grant Agreement number	731601
Project acronym:	SPECIAL
Project title:	Scalable Policy-awarE Linked Data arChitecture for prIvacy, trAnsparency and compLiance
Funding Scheme:	Research & Innovation Action (RIA)
Date of latest version of DoW against which the assessment will be made:	17/10/2016
Document	
Period covered:	M1-M12
Deliverable number:	D2.1
Deliverable title	Policy Language V1
Contractual Date of Delivery:	31/12/2017
Actual Date of Delivery:	26/12/2017
Editor (s):	P.A. Bonatti, L. Sauro, I. Petrova (CeRICT)
Author (s):	P.A. Bonatti, S. Kirrane, I. Petrova, L. Sauro, E. Schlehahn
Reviewer (s):	W. Dullaert, U. Milosevic, P. Raschke, R. Wenning
Work package no.:	2
Work package title:	Policy and Transparency Framework
Work package leader:	Sabrina Kiranne (WU)
Distribution:	PU
Version/Revision:	1.0
Draft/Final:	Final
Total number of pages (including cover):	56

Disclaimer

This document contains description of the SPECIAL project work and findings.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any responsibility for actions that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of the SPECIAL consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 28 Member States of the Union. It is based on the European Communities and the Member States cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors (http://europa.eu/).

SPECIAL has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 731601.

Contents

List of Figures

1 Summary

The goal of this deliverable is introducing the *usage policy language* of SPECIAL. The usage policy language is meant to express both the data subjects' consent and the data usage policies of data controllers in formal terms, understandable by a computer, so as to automatically verify that the usage of personal data complies with data subjects' consent.

What is in this deliverable

Chapter 1 introduces the core of the usage policy language – which is based on the MCM, the Minimal Core Model introduced in D1.3 – while Chapter 2 focusses on the vocabularies for expressing each of the MCM's elements (data categories, purposes, etc). This deliverable contains only a first draft of such vocabularies; an improved version is being developed jointly with the pilot leaders, based on the pilots' needs. Finally, the formal ontologies defining the meaning of the terms employed in the policy language and related vocabularies are reported in the Appendix.

What is *not* in this deliverable

Here we do not deal with the languages for describing business policies and processes, that are needed to check compliance w.r.t. the GDPR. This is the subject of another deliverable (D2.2). In particular, obligations will be dealt with there.

Similarly, we do not (yet) deal with any issue related to consent *requests*, such as splitting the policy into mandatory and optional parts, and specifying whether the default selection modality is opt-in or opt-out.

In both cases (business policies and consent requests) the usage descriptions described in this deliverable will be part of the specification. Following an incremental design strategy, usage policies will be extended with suitable attributes to encode obligations, opt-in/out policies and any further elements needed in those contexts.

Finally, since more joint work with the pilot leaders is needed to detail the vocabularies for the policy language, the examples currently included in the paper are still generic. They are only meant to illustrate the policies' structure, and are not suitable for legal analysis.

Chapter 1

SPECIAL's Usage policy Language

This chapter starts by introducing policies at a conceptual level, in simple mathematical terms, based on the notions of sets and tuples (Sec. 1). Section 2 explains how to encode those sets and tuples using the standard ontology language OWL 2. However, this deliverable can be read (and the policy language used) without knowing much about OWL 2, because policy expressions are described by a simple ad-hoc grammar (Fig. 1.1) that needs no logical prerequisites to be understood. Nonetheless, encoding usage policies in OWL 2 has several advantages:

- We have designed the policy encoding in OWL 2 so that the available reasoning engines for OWL 2 can directly execute a variety of policy validation checks. Such engines can also check whether a controller's policy complies with data subjects' consent, however in forthcoming deliverables we are going to develop an ad-hoc compliance engine for scalability (cf. Sec. 4).
- The correctness of this realization of validation and compliance is based on sound formal results, leveraging the formal semantics of OWL 2 and a large body of theoretical results.

1 What is a Usage Policy?

Conceptually, a *usage policy* is meant to specify a *set of authorized operations*. In the *minimum core model* (MCM) specified in deliverable D1.3 such authorized operations are characterized by:

- the data processed by the operation;
- the purpose of the operation;
- a description of the operation itself (e.g. "*query*", "*classification*", "*disclosure*", etc.);
- a description of where the result is stored and for how long;
- the entities that can access the result of the operation (recipients).

That is, in abstract, mathematical terms a usage policy is just a set of tuples like

$$\langle \textit{data item}, \textit{purpose}, \textit{operation}, \textit{storage}, \textit{recipient} \rangle$$

that will be called *authorizations*, each of which specifies a permitted operation.

Example 1 *A policy P allows to disclose the file* John_Smith_profile.xml *to company ACME for commercial purposes iff a corresponding authorization*

$$\langle \textit{John\_Smith\_profile.xml, CommercialPurpose, disclose, null, ACME} \rangle .$$

belongs to P. ∎

A data subject's *consent* comprises a usage policy; the authorizations contained in that policy are the operations allowed by the data subject.

Dually, the usage policy enforced by a data controller contains the operations that are permitted within the data controller's organization.

Therefore, the usage policy adopted by the data controller (call it P_c) *complies* with the usage policy in the data subject's consent (call it P_s) if and only if all the authorizations in P_c are also authorized by P_s, that is, P_c complies with P_s if and only if

$$P_c \subseteq P_s . \tag{1.1}$$

Now we have to define a *policy language* where this kind of policies can be modelled precisely and unambiguously, possibly by representing large sets of authorizations (tuples) in a concise way. Such policy language should be equipped with an *inference engine* capable of checking reliably and exhaustively whether policy containments such as (1.1) hold, whenever P_c and P_s are expressed with the policy language. These are the goals of the next section.

2 SPECIAL's Usage Policy Language

SPECIAL's usage policies are encoded in OWL 2.[1] This standard supports both an XML format and so-called *functional syntax*.[2] The ontology editor *Protégé*[3] supports both. In order to improve readability, here we use the functional syntax which is less verbose.

The sets of authorizations that conceptually consititute the policy (see Sec. 1) are naturally represented by OWL 2 *classes*, and single authorizations by class *instances*. In the following we illustrate the language and provide some examples. The impatient reader can find a grammar of policy expressions in BNF format in Figure 1.1.

2.1 Basic Usage Policies

A usage policies is composed of one or more *basic usage policies*, each of which is an OWL 2 expression of the form:

```
ObjectIntersectionOf(                                              (1.2)
    ObjectSomeValuesFrom(spl:hasData SomeDataCategory)
    ObjectSomeValuesFrom(spl:hasProcessing SomeProcessing)
    ObjectSomeValuesFrom(spl:hasPurpose SomePurpose)
    ObjectSomeValuesFrom(spl:hasRecipient SomeRecipient)
    ObjectSomeValuesFrom(spl:hasStorage SomeStorage)
)
```

[1] https://www.w3.org/TR/owl2-primer/
[2] https://www.w3.org/TR/owl2-syntax/
[3] https://protege.stanford.edu/

The important parts in this expression are the policy's attributes highlighted in bold; the policy author shall decide for each of them a suitable range, that in the above text is highlighted in italics. The above policy authorizes all the operations that

1. fall within the specified *SomeProcessing* category,
2. operate only on data that belong to *SomeDataCategory*,
3. have any purpose covered by the *SomePurpose* category,
4. disclose the results to any member(s) of the *SomeRecipient* category,
5. store the results in any place belonging to the *SomeStorage* category.

The classes *SomeDataCategory, SomeProcessing, SomePurpose, SomeRecipient* shall be terms defined in the vocabularies discussed in Chapter 2, while the structure of *SomeStorage* is illustrated later on in this section.

Note on OWL 2: An expression

`ObjectSomeValuesFrom(` *AttributeName AttributeRange* `)`

denotes the class of all objects that have (at least) an attribute *AttributeName* whose value belongs to the class *AttributeRange*. By enclosing several of these expressions into an `ObjectIntersectionOf` we obtain the class of all objects that have (at least) all the specified attributes. Therefore, (1.2) encodes the set of all authorizations that have the specified attributes.

2.2 General Usage Policies

A basic usage policy like (1.2) cannot specify diverse conditions for different data categories, or different purposes. In order to state – for example – that personal data can only be used for non-commercial purposes and shall neither be stored nor disclosed to third parties, while pseudonymised data can be used freely, one can use a *general usage policy* like the following (we assume here that the auxiliary vocabularies define the terms *PersonalData, NonCommercial,*

PseudonymizedData):

```
ObjectUnionOf(
  ObjectIntersectionOf(
    ObjectSomeValuesFrom(spl:hasData PersonalData)
    ObjectSomeValuesFrom(spl:hasProcessing spl:AnyProcessing)
    ObjectSomeValuesFrom(spl:hasPurpose NonCommercial)
    ObjectSomeValuesFrom(spl:hasRecipient spl:Null)
    ObjectSomeValuesFrom(spl:hasStorage ŝpl:Null)
  )
  ObjectIntersectionOf(
    ObjectSomeValuesFrom(spl:hasData PseudonymizedData)
    ObjectSomeValuesFrom(spl:hasProcessing spl:AnyProcessing)
    ObjectSomeValuesFrom(spl:hasPurpose spl:AnyPurpose)
    ObjectSomeValuesFrom(spl:hasRecipient spl:AnyRecipient)
    ObjectSomeValuesFrom(spl:hasStorage ŝpl:AnyStorage)
  )
)
```

(1.3)

A general usage policy may contain any number of basic policies as in:

$$\texttt{ObjectUnionOf(}\ \mathit{BasicPolicy}_1\ \ldots\ \mathit{BasicPolicy}_n\ \texttt{)}$$

where each $\mathit{BasicPolicy}_i$ is of the form (1.2). The resulting policy is conceptually the *union* of all the authorizations supported by the basic policies, that is, an operation is authorized by the general policy if and only if the operation is authorized by *at least one* of its basic policies. The complete syntax of usage policies is specified by the BNF grammar in Figure 1.1.

General policies whose basic policies differ only in one attribute can be factorized using a logical equivalence supported by OWL 2. For example, the following general usage policy

```
ObjectUnionOf(
  ObjectIntersectionOf(
    ObjectSomeValuesFrom(spl:hasData NavigationData)
    ObjectSomeValuesFrom(spl:hasProcessing spl:AnyProcessing)
    ObjectSomeValuesFrom(spl:hasPurpose NonCommercial)
    ObjectSomeValuesFrom(spl:hasRecipient spl:AnyRecipient)
    ObjectSomeValuesFrom(spl:hasStorage ŝpl:AnyStorage)
  )
  ObjectIntersectionOf(
    ObjectSomeValuesFrom(spl:hasData LocationData)
    ObjectSomeValuesFrom(spl:hasProcessing spl:AnyProcessing)
    ObjectSomeValuesFrom(spl:hasPurpose NonCommercial)
    ObjectSomeValuesFrom(spl:hasRecipient spl:AnyRecipient)
    ObjectSomeValuesFrom(spl:hasStorage ŝpl:AnyStorage)
  )
)
```

can be abbreviated by the equivalent OWL 2 expression

```
ObjectIntersectionOf(
  ObjectSomeValuesFrom(spl:hasData
                       ObjectUnionOf( NavigationData LocationData ))
  ObjectSomeValuesFrom(spl:hasProcessing spl:AnyProcessing)
  ObjectSomeValuesFrom(spl:hasPurpose NonCommercial)
  ObjectSomeValuesFrom(spl:hasRecipient spl:AnyRecipient)
  ObjectSomeValuesFrom(spl:hasStorage spl:AnyStorage)
)
```

2.3 Storage Expressions

The `hasStorage` policy attribute is a structured object itself, with two attributes, and is specified as follows:

```
ObjectIntersectionOf(                                              (1.4)
   ObjectSomeValuesFrom(spl:hasLocation SomeLocation)
   ObjectSomeValuesFrom(spl:hasDuration SomeDuration)
   DataSomeValuesFrom(spl:durationInDays Interval)
)
```

where *SomeLocation* and *SomeDuration* are expressed in terms of the corresponding location and duration vocabularies, respectively (cf. Chapter 2). The *Interval* limiting storage duration in days[4] is expressed with the integer *facets* of OWL 2, that is:

```
DatatypeRestriction( xsd:integer                                   (1.5)
   xsd:minInclusive min duration                    (optional)
   xsd:maxInclusive max duration                    (optional)
)
```

where min and max durations follow the syntax of *xsd:integer*, that is,

"*sequence of digits*"^^xsd:integer

It is not necessary to include in the policy both `hasDuration` and `durationInDays`. However at least one of them should be specified.

Example 2 *The following policy permits to store the result of the processing in a server located*

[4]Here we are choosing days as the finest granularity in order to make examples easier to understand. Technically, we only need to map time points onto integers. So we could as well express dates and time up to the second, and use a standard mapping from dates to integers like the one used in Unix-like systems.

in Europe for at most one year:

```
ObjectIntersectionOf(
  ObjectSomeValuesFrom(spl:hasData LocationData ))
  ObjectSomeValuesFrom(spl:hasProcessing spl:AnyProcessing)
  ObjectSomeValuesFrom(spl:hasPurpose NonCommercial)
  ObjectSomeValuesFrom(spl:hasRecipient spl:Null)
  ObjectSomeValuesFrom(spl:hasStorage
    ObjectIntersectionOf(
      ObjectSomeValuesFrom(spl:hasLocation Europe)
      DataSomeValuesFrom(spl:durationInDays
        DatatypeRestriction( xsd:integer
                             xsd:maxInclusive "365"^^xsd:integer)
      )
    )
  )
)
```

SPECIAL's usage policy namespace and semantics: All policy attributes are *functional*, e.g. every single authorization in a policy has one data element, one purpose element, one processing element etc. The semantics of these attributes – as well as the semantics of all symbols in the `spl` namespace, that contains the specific symbols of SPECIAL's policy language – is defined in the OWL 2 ontology illustrated in Appendix 1.

3 Policy Annotations

OWL 2 supports annotations that can be attached to virtually every element of the language. In SPECIAL such annotations can be used to add comments and explanations to policies. Similar annotations can be exploited to construct policy visualization and policy explanation tools that automatically produce a human-readable text from the internal OWL 2 format of the policy. A more detailed treatment of annotations is deferred to the next versions of this deliverable.

4 Compliance Checking and Policy Verification

Recall from Section 1 that a policy P_c complies with another policy P_s if and only if (1.1) holds, that is, $P_c \subseteq P_s$. In OWL 2 terminology, checking whether a general usage policy Pol_1 complies with another general usage policy Pol_2 amounts to checking whether the following axiom is *entailed* (implied) by the combined ontology $\mathcal{O}$ containing the language ontology in Appendix 1 plus the vocabularies discussed in Section 2:

$$\texttt{SubClassOf(}\ Pol_1\ Pol_2\ \texttt{).} \tag{1.6}$$

Note on OWL 2 semantics: The above OWL 2 axiom is entailed if and only if all the instances of the class Pol_1 are also contained in Pol_2, therefore it matches perfectly the inclusion test (1.1).

Figure 1.1: SPECIAL's Usage Policy Language Grammar

UsagePolicy :='ObjectUnionOf' '(' **BasicUsagePolicy BasicUsagePolicy** { **BasicUsagePolicy** } ')'
| **BasicUsagePolicy**

BasicUsagePolicy :='ObjectIntersectionOf' '(' **Data Purpose Processing Recipients Storage** ')'

Data :='ObjectSomeValueFrom' '(' 'spl:hasData' **DataExpression** ')'

Purpose :='ObjectSomeValueFrom' '(' 'spl:hasPurpose' **PurposeExpression** ')'

Processing :='ObjectSomeValueFrom' '(' 'spl:hasProcessing' **ProcessingExpression** ')'

Recipients :='ObjectSomeValueFrom' '(' 'spl:hasRecipient' **RecipientExpression** ')'

Storage :='ObjectSomeValueFrom' '(' 'spl:hasStorage' **StorageExpression** ')'

DataExpression :='spl:AnyData' | **DataVocabExpression**

PurposeExpression :='spl:AnyPurpose' | **PurposeVocabExpression**

ProcessingExpression :='spl:AnyProcessing' | **ProcessingVocabExpression**

RecipientsExpression :='spl:AnyRecipient' | 'spl:Null' | **RecipientVocabExpression**

StorageExpression :='spl:AnyStorage' | 'spl:Null' |
'ObjectIntersectionOf' '(' **Location Duration** ')'

Location :='ObjectSomeValueFrom' '(' 'spl:hasLocation' **LocationExpression** ')'

Duration :='ObjectSomeValueFrom' '(' 'spl:hasDuration' **DurationExpression** ')'
| 'DataSomeValueFrom' '(' 'spl:durationInDays' **IntervalExpression** ')'

LocationExpression :='spl:AnyLocation' | **LocationVocabExpression**

DurationExpression :='spl:AnyDuration' | **DurationVocabExpression**

IntervalExpression :='DatatypeRestriction' '(' 'xsd:integer' **LowerBound UpperBound** ')'

LowerBound :='xsd:minInclusive' **IntegerLiteral**

UpperBound :='xsd:maxInclusive' **IntegerLiteral**

IntegerLiteral := **stringOfDigits** '^^' 'xsd:integer'

stringOfDigits := *a sequence of digits enclosed in a pair of " (U+22)*

DataVocabExpression := *as specified in Chapter 2 Section 4*

PurposeVocabExpression := *as specified in Chapter 2 Section 5*

ProcessingVocabExpression := *as specified in Chapter 2 Section 6*

RecipientVocabExpression := *as specified in Chapter 2 Section 1*

LocationVocabExpression := *as specified in Chapter 2 Section 2*

DurationVocabExpression := *as specified in Chapter 2 Section 3*

Off-the-shelf engines: The general inference engines for OWL 2 – such as HermiT and FaCT++,[a] just to name a few – are able to check whether a `SubClassOf` axiom is entailed by an OWL 2 knowledge base like $\mathcal{O}$. Such entailment queries can be programmed using the OWL API, that supports methods like `isEntailed(OWLAxiom axiom)` and interface `OWLSubClassOfAxiom` to handle axioms like (1.6). It is also possible to evaluate `SubClassOf` queries through the GUI of Protégé.

[a] Both engines have been integrated in Protégé.

SPECIALized engine: The general inference engines for OWL 2 have a very high asymptotic complexity, so they might not be performant enough over a very large number of policies. For this reason we are developing an engine that can answer compliance queries in polynomial time. The complete description of this engine and its experimental assessment and comparison with the aforementioned general engines will be included in the next versions of this deliverable.

By means of entailment checking of `SubClassOf` axioms one can also identify some errors in policy writing, such as type mismatches (e.g. using a data category in the `hasPurpose` slot) and out-of-range errors (e.g. negative storage durations). Such errors cause the policy to be *inconsistent*. In that case, the ontology $\mathcal{O}$ entails the axiom:

```
SubClassOf( PolicyExpression owl:Nothing ).        (1.7)
```

5 Foundational aspects

The OWL 2 formalization of usage policies has been crafted so as to preserve the conceptual view of policies as sets of authorization tuples.

Tuples are encoded in OWL by *reifying* them. "Reification" is a technical term meaning that each tuple $t = \langle d, pu, pr, s, r \rangle$ is represented by an object o whose attributes match the tuple's elements (in our case, `hasData`, `hasProcessing`, `hasPurpose`, `hasRecipient`, `hasCategories` have values d, pu, pr, s, and r, respectively).

Reification requires some care. Two different objects o and o' may represent the same tuple t (and there is no way to prevent this in OWL 2); as a consequence some set-theoretic operations (notably intersection and complement) may be incorrectly represented by the corresponding OWL 2 operations `ObjectIntersectionOf` and `ObjectComplementOf`.

The encoding of general policies specified in Figure 1.1 correctly preserves the set theoretic meaning of policies (i.e. union). We are working at more general guarantees, covering also policy intersection and complement. Our current conjecture is that all the operations of the *policy composition algebra*[5] are correctly represented by their direct OWL 2 counterpart. This theoretical work is in progress and will be included in the future versions of this deliverable.

[5] Piero A. Bonatti, Sabrina De Capitani di Vimercati, Pierangela Samarati: An algebra for composing access control policies. ACM Trans. Inf. Syst. Secur. 5(1): 1-35 (2002)

Chapter 2

Auxiliary Vocabularies V1

In this chapter we introduce the first version of the vocabularies for specifying the attributes of SPECIAL's usage policies. This version is meant as a tool for developing the first version of the pilots, while "more standardized" vocabularies are being developed with the contribution of a wider range of stakeholders (cf. D6.3, Sec. 5). The temporary nature of the first vocabularies led us to re-use extensively some previously defined privacy-related vocabularies (such as those introduced in P3P and ODRL) with minimal modifications and extensions needed to support SPECIAL's pilots and the relevant GDPR concepts. Some examples of the use and extension of these vocabularies, inspired by SPECIAL's pilots, are illustrated in Sec. 8.

1 Recipient categories

The set of possible recipients in P3P fits rather well the legal classification adopted in the GDPR, therefore we adopt the same model in SPECIAL's recipient vocabulary. We start by recalling P3P's vocabulary.

Recipients in P3P

<ours>
Ourselves and/or entities acting as our agents or entities for whom we are acting as an agent: An agent in this instance is defined as a third party that processes data only on behalf of the service provider for the completion of the stated purposes. (e.g., the service provider and its printing bureau which prints address labels and does nothing further with the information.)[1]

<delivery>
Delivery services possibly following different practices: Legal entities performing delivery services that may use data for purposes other than completion of the stated purpose. This should also be used for delivery services whose data practices are unknown.

<same>
Legal entities following our practices: Legal entities who use the data on their own behalf under equable practices. (e.g., consider a service provider that grants the user access to collected

[1] This description matches closely the GDPR's notion of *data processor* and currently we suggest to use this term to express data disclosure to data processors.

personal information, and also provides it to a partner who uses it once but discards it. Since the recipient, who has otherwise similar practices, cannot grant the user access to information that it discarded, they are considered to have equable practices.)[2]

<other-recipient>
Legal entities following different practices: Legal entities that are constrained by and accountable to the original service provider, but may use the data in a way not specified in the service provider's practices (e.g., the service provider collects data that is shared with a partner who may use it for other purposes. However, it is in the service provider's interest to ensure that the data is not used in a way that would be considered abusive to the users' and its own interests.)

<unrelated>
Unrelated third parties: Legal entities whose data usage practices are not known by the original service provider.

<public>
Public fora: Public fora such as bulletin boards, public directories, or commercial CD-ROM directories.

Recipient Expressions in OWL 2

Recipient elements are mapped to OWL 2 classes. If `<R>` is a recipient element, then the corresponding class is `svr:R`, where `svr` abbreviates the namespace

`<http://www.specialprivacy.eu/vocabs/recipients#>`.

All classes in the recipients vocabulary *must* be subclasses of `<spl:AnyRecipient>`.

Syntax and the logical semantics of the current recipient vocabulary are specified in Figure 2.1 and Appendix 5, respectively.

Specializing the recipients' vocabulary: The recipients' vocabulary may be extended by a data controller by introducing new classes that correspond to application-specific recipient categories. *Such additional classes should be subclasses of SPECIAL's recipient vocabulary's classes.* Specific recipients (individual parties) can be specified as instances of recipient classes, using appropriate vocabularies for describing legal and physical persons.

[2] This definition and the associated example are not precise enough to understand the scope and meaning of this term, in view of a legal evaluation of a policy. This limitation is probably due to the fact that P3P has not been conceived to address compliance with respect to any regulation, so its definitions are not tied to any particular set of legal definitions. As a consequence, for SPECIAL's purposes, this term may need to be specified more precisely, and possibly split into different complementary terms.

Figure 2.1: SPECIAL's Recipient Expression Grammar

RecipientVocabExpression := *an expression* `<svr:R>` *where* `R` *is a P3P recipient*

2 Location

From a legal perspective, the main aspects related to location are (i) whether the information is stored in the EU or in countries with similar data protection legislation, and (ii) whether the information is kept by the data controller or stored outside its boundaries. SPECIAL's first location vocabulary embraces this level of granularity.

Let `svl` abbreviate the namespace `<http://www.specialprivacy.eu/vocabs/locations#>`. The location vocabulary contains a first group of mutually disjoint classes:

`svl:EU, svl:EULike, svl:ThirdCountries.`

The class `<svl:EULike>` currently comprises all countries in the European Economic Area (e.g. Iceland, Liechtenstein and Norway) plus other officially recognized contries such as Andorra, Argentina, Canada, Switzerland, Faeroe Islands, Guernsey, Israel, Isle of Man, Jersey, New Zealand and Uruguay.[3]

Additionally, the location vocabulary comprises the following group of mutually disjoint classes:

`svl:ControllerServers, svl:ProcessorServers, svl:ThirdParty .`

For convenience we introduce an additional superclass `svl:OurServers` that includes both `svl:ControllerServers` and `svl:ProcessorServers`.

Two classes belonging to different groups may overlap. For example, if data are kept in the controller's servers that are located in France, then location may be specified with the class:

`ObjectIntersectionOf( <svl:ControllerServers> <svl:EU> ).`

All classes in the location vocabulary *must* be subclasses of `<spl:AnyLocation>`.

Syntax and the logical semantics of the current location vocabulary are specified in Figure 2.2 and Appendix 6, respectively.

Specializing the location vocabulary: The location vocabulary may be extended by a data controller by introducing classes and instances that correspond to specific servers or groups of servers. This may facilitate the mapping between the controller's systems and its usage policies. Similary, the vocabulary may be extended with classes corresponding to countries and other geographic areas, so as to later relieve the policy author from knowing the appropriate classification of non-EU countries into `EULike` and `ThirdCountries`. *All of such additional classes should be subclasses and instances of the location vocabulary's classes.*

[3] The current list is based on the current Data Protection Directive 95/46 EC and may have to be revised when the GDPR comes into force.

Figure 2.2: SPECIAL's Location Expression Grammar

LocationVocabExpression :='ObjectIntersectionOf' '(' **LocationClass LocationClass** ')'
| **LocationClass**

LocationClass :='svl:EU' | 'svl:EULike' | 'svl:ThirdCountries' | 'svl:ControllerServers'
| 'svl:ProcessorServers' | 'svl:ThirdParty' | 'svl:OurServers'

3 Storage duration

Duration can be expressed in abstract terms as in P3P's `RETENTION` element. We first recall P3P's original retention vocabulary, then we introduce our OWL 2 encoding.

Retention in P3P

The following list contains the possible values of P3P's `<RETENTION>` element. Their descriptions are quoted from P3P1.1's specification.[4]

<no-retention/>
Information is not retained for more than a brief period of time necessary to make use of it during the course of a single online interaction. Information MUST be destroyed following this interaction and MUST NOT be logged, archived, or otherwise stored. This type of retention policy would apply, for example, to services that keep no Web server logs, set cookies only for use during a single session, or collect information to perform a search but do not keep logs of searches performed.

<stated-purpose/>
Information is retained to meet the stated purpose. This requires information to be discarded at the earliest time possible. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

<legal-requirement/>
As required by law or liability under applicable law: Information is retained to meet a stated purpose, but the retention period is longer because of a legal requirement or liability. For example, a law may allow consumers to dispute transactions for a certain time period; therefore a business may for liability reasons decide to maintain records of transactions, or a law may affirmatively require a certain business to maintain records for auditing or other soundness purposes. Sites MUST have a retention policy that establishes a destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

<business-practices/>
Determined by service provider's business practice: Information is retained under a service provider's stated business practices. Sites MUST have a retention policy that establishes a

[4] `http://www.w3.org/TR/P3P11/#RETENTION`

Figure 2.3: SPECIAL's Duration Expression Grammar

DurationVocabExpression := *an expression* `<svdu:R>` *where* `R` *is a P3P retention element different from* `no-retention`

destruction time table. The retention policy MUST be included in or linked from the site's human-readable privacy policy.

<indefinitely/>
Information is retained for an indeterminate period of time. The absence of a retention policy would be reflected under this option. Where the recipient is a public fora, this is the appropriate retention policy.

Retention in OWL 2

P3P's retention elements may turn out to be too generic for SPECIAL's purposes, still they may work as an *upper ontology* (that is, a set of general classes to be extended by specialized, possibly controller-specific vocabularies). Moreover, duration can be described more precisely with day intervals, using the `durationInDays` storage attribute.

The `no-retention` element is not necessary in SPECIAL's usage policy language, because it should be rather specified with the attribute specification

```
ObjectSomeValueFrom( <spl:hasStorage> <spl:Null> ).
```

The other retention elements of P3P are mapped to OWL 2 classes. If `<R>` is a retention element, then the corresponding duration class is `svdu:R`, where `svdu` abbreviates the namespace

```
<http://www.specialprivacy.eu/vocabs/duration#>.
```

All classes in the recipients vocabulary *must* be subclasses of `<spl:AnyDuration>`.

Syntax and the logical semantics of the current recipient vocabulary are specified in Figure 2.3 and Appendix 7, respectively.

4 Data categories

P3P describes what a statement is about by means of data types and categories. Data types represent which specific classes of data a service is actually or potentially collecting. Categories are optional descriptors that provide hints to the intended use of the data. They typically express more abstract classes of data and may be used to define more generalized preferences and rules over the exchange of their data.

Data types in P3P

Data types are structured in data schemas which are hierarchically organized in elements of increasing levels of granularity. Policies may use either the Base Data Schema provided by the P3P specification, or may create custom data schemas. An element high up in the hierarchy thus contains all the meaning of the allowed subtypes, unless the collected subtypes are declared by inserting a child element. For example:

```
<datatype>                                                    (2.1)
   <user/>
</datatype>
```

means: "the service may collect any of the allowed subtypes of user data" (name,birthdate,login, cert etc...). Furthermore this property is carried through to all the subtypes of these elements. So collecting user data also means collecting login-id and login-password etc ... If instead the following is specified:

```
<datatype>                                                    (2.2)
   <user>
      <cert>
         <key/>
      </cert>
   </user>
</datatype>
```

this means that the service may only collect the user's certificate key and nothing else. Basic Data Schema consists of the following data types:

<dynamic/>
Data elements that do not have fixed values that a user might type in or store in a repository. In the P3P base data schema, all such elements are grouped under the class of dynamic data. Sites may refer to the types of data they collect using the dynamic data set only, rather than enumerating all of the specific data elements. Subtypes: clickstream ,http, clientevents, cookies, searchtext, interactionrecord, miscdata.

<user/>
General information about the user. Subtypes: name, bdate, login, cert, gender, jobtitle, home-info, business-info.

<third-party/>
The thirdparty data set allows users and businesses to provide values for a related third party. This can be useful whenever third party information needs to be exchanged, for example when ordering a present online that should be sent to another person, or when providing information about one's spouse or business partner. Such information could be stored in a user repository alongside the user data set. User agents may offer to store multiple such thirdparty data sets and allow users to select the appropriate values from a list when necessary. The allowed subtypes of thirdparty data element are identical to those of the user data set. Subtypes: name, bdate, login, cert, gender, jobtitle, home-info, business-info.

<business/>
The business data type features a subset of user data relevant for describing legal entities. In P3P1.1, this data element is primarily used for describing the policy entity, although it should also be applicable to business-to-business interactions. Subtypes: orgname, department, cert, contact-info.

<orgname/>
Organization Name.

<name/>
User's Name. Subtypes: prefix, given, family, middle, suffix, nickname.

<bdate/>
User's Birth Date. Subtypes: ymd.year, ymd.month, ymd.day, hms.hour, hms.minute, hms.second, fractionsecond, timezone.

<login/>
User's Login Information. The login element and its children refer to information (IDs and passwords) for computer systems and Web sites which require authentication. Note that this data element should not be used for computer systems or Web sites which use digital certificates for authentication: in those cases, the cert element should be used. Subtypes: id, password.

<cert/>
User's Identity Certificate. The cert element and its children refer to identity certificates (like, for example, X.509). Subclasses: key, format.

<gender/>
User's Gender (Male or Female).

<employer/>
User's Employer.

<department/>
Department or Division of Organization where User is Employed.

<jobtitle/>
User's Job Title.

<home-info/>
User's Home Contact Information. Subclasses: postal, telecom, online.

<contact-info/>
Contact Information for the Organization. Subclasses: postal, telecom, online.

<business-info/>
User's Business Contact Information. Subclasses: postal, telecom, online, employer, department.

<clickstream/>
The clickstream element and its children refer to information typically stored in Webserver access logs. The clickstream element is expected to apply to practically all Web sites. It represents the combination of information typically found in Web server access logs: the IP address or hostname of the user's computer, the URI of the resource requested, the time the request was made, the HTTP method used in the request, the size of the response, and the HTTP status code in the response. Web sites that collect standard server access logs as well as sites which do URI path analysis can use this data element to describe how that data will be used. Web sites that collect only some of the data elements listed as allowed children of the clickstream element MAY choose to list those specific elements rather than the entire dynamic-clickstream element. This allows sites with more limited data-collection practices to accurately present those practices to their visitors. The resource in the HTTP request is captured by the uri field. The IP address of the client system making the request is given by the clientip field. Subclasses: uri, timestamp, clientip, other.httpmethod, other.bytes, other.statuscode

<http/>
The http element contains additional information contained in the HTTP protocol. The http element and its children refer to information carried by the HTTP protocol which is not covered by the clickstream element and its children. Subclasses: referer, useragent.

<clientevents/>
The clientevents element represents data about how the user interacts with their Web browser while interacting with a resource. For example, an application may wish to collect information about whether the user moved their mouse over a certain image on a page, or whether the user ever brought up the help window in a Java applet. This kind of information is represented by the clientevents data element. Much of this interaction record is represented by the events and data defined by the Document Object Model (DOM) Level 2 Events `[https://www.w3.org/TR/P3P/#ref_DOM2-Events]`. The clientevents data element also covers any other data regarding the user's interaction with their browser while the browser is displaying a resource. The exception is events which are covered by other elements in the base data schema. For example, requesting a page by clicking on a link is part of the user's interaction with their browser while viewing a page, but merely collecting the URL the user has clicked on does not require declaring this data element; clickstream covers that event. However, the DOM event DOMFocusIn (representing the user moving their mouse over an object on a page) is not covered by any other existing element, so if a site is collecting the occurrence of this event, then it needs to state that it collects the dynamic.clientevents element. Items covered by this data element are typically collected by client-side scripting languages, such as JavaScript, or by client-side applets, such as ActiveX or Java applets. Note that while the previous discussion has been in terms of a user viewing a resource, this data element also applies to Web applications which do not display resources visually - for example, audio-based Web browsers.

<cookies/>
The cookies element should be used whenever HTTP cookies are set or retrieved by a site. Please note that cookies is a variable category data element and requires the explicit declaration of usage categories in a policy.

<miscdata/>
The miscdata element references information collected by the service that the service does not

reference using a specific data element. Categories have to be used to better describe these data: sites MUST reference a separate miscdata element in their policies for each category of miscellaneous data they collect.

<searchtext/>
The searchtext element references a specific type of solicitation used for searching and indexing sites. For example, if the only fields on a search engine page are search fields, the site only needs to disclose that data element.

<interactionrecord/>
The interactionrecord element should be used if the server is keeping track of the interaction it has with the user (i.e. information other than clickstream data, for example account transactions, etc).

<ymd.year/>
The following elements refer to data connected with dates. Since date information can be used in different ways, depending on the context, all such information is tagged as being of variable category. For example, schema definitions can explicitly set the corresponding category in the element referencing these elements, where soliciting the birthday of a user might be "Demographic and Socioeconomic Data", while the expiration date of a credit card might belong to the Purchase Information category.

<ymd.month/>
Month.

<ymd.day/>
Day.

<hms.hour/>
Hour.

<hms.minute/>
Minute.

<hms.second/>
Second.

<fractionsecond/>
Fraction of Second.

<timezone/>
Time Zone.

<prefix/>
Name Prefix.

<given/>
Given Name (First Name).

<family/>
Family Name (Last Name).

<middle/>
Middle Name.

<suffix/>
Name Suffix.

<nickname/>
Nickname.

<id/>
The "id" element represents the ID portion of the login information for a computer system. Often, user IDs are made public, while passwords are kept secret. IDs do not include any type of biometric authentication mechanisms.

<password/>
The "password" element represents the password portion of the login information for a computer system. This is a secret data value, usually a character string, that is used in authenticating a user. Passwords are typically kept secret, and are generally considered to be sensitive information.

<key/>
Certificate Key.

<format/>
The "format" element is used to represent the information of an IANA registered public key or authentication certificate format, while the "key" field is used to represent the corresponding certificate key.

<postal/>
The following 3 elements and their children refer to contact information. Services can specify precisely which set of data they need, postal, telecommunication, or online address information. The postal element and its children refer to a postal mailing address.

<telecom/>
The telecom element and its children refer to the characteristics of telephone, fax, mobile and pager numbers.

<online/>
The online element and its children refer to online information about a person or legal entity.

<intcode/>
International Telephone Code.

<loccode/>
Local Telephone Area Code.

<number/>
Telephone Number.

<ext/>
Telephone Extension.

<comment/>
Telephone Optional Comments.

<street/>
Street Address.

<city/>
City.

<stateprov/>
State or Province.

<postalcode/>
Postal Code.

<country/>
The "country" element represents the information of the name of the country (for example, one among the countries listed in [ISO3166]).

<organization/>
Organization Name.

<telephone/>
Telephone Number.

<fax/>
Fax Number.

<mobile/>
Mobile Telephone Number.

<pager/>
Pager Number.

<email/>
Email Address.

<uri/>
Home Page Address. The uri element and its children refer to Universal Resource Identifiers (URI), which are defined in [URI]. Since URI information can be used in different ways, depending on the context, all the child elements of the uri element are tagged as being of variable category. Schema definitions MUST explicitly set the corresponding category in the element referencing this data structure.

<authority/>
URI Authority.

<stem/>
URI Stem.

<querystring/>
Query-string Portion of URI.

<hostname/>
Complete Host and Domain Name. The hostname element is used to represent collection of either the simple hostname of a system, or the full hostname including domain name. The partialhostname element represents the information of a fully-qualified hostname which has had at least the host portion removed from the hostname. In other words, everything up to the first '.' in the fully-qualified hostname MUST be removed for an address to quality as a "partial hostname".

<partialhostname/>
Partial Hostname.

<fullip/>
Full IP Address. The fullip element represents the information of a full IP version 4 or IP version 6 address. The partialip element represents an IP version 4 address (only - not a version 6 address) which has had at least the last 7 bits of information removed. This removal MUST be done by replacing those bits with a fixed pattern for all visitors (for example, all 0's or all 1's). Certain Web sites are known to make use not of the visitor's entire IP address or hostname, but rather make use of a reduced form of that information. By collecting only a subset of the address information, the site visitor is given some measure of anonymity. It is certainly not the intent of this specification to claim that these "stripped" IP addresses or hostnames are impossible to associate with an individual user, but rather that it is significantly more difficult to do so. Sites which perform this data reduction MAY wish to declare this practice in order to more-accurately reflect their practices.

<partialip/>
Partial IP Address.

<timestamp/>
The time at which the server processes the request is represented by the timestamp field. Server implementations are free to define this field as the time the request was received, the time that the server began sending the response, the time that sending the response was complete, or some other convenient representation of the time the request was processed. Subclasses: ymd.year, ymd.month, ymd.day, hms.hour, hms.minute, hms.second, fractionsecond, timezone.

<clientip/>
The clientip element and its children refer to IP addresses and Domain Name System (DNS) hostnames. Subclasses: hostname, partialhostname, fullip, partialip.

<other.httpmethod/>
The HTTP method (such as GET, POST, etc) in the client's request.

<other.bytes/>
The number of bytes in the response-body sent by the server.

<other.statuscode/>
The HTTP status code on the request, such as 200, 302, or 404 (see section 6.1.1 of `[http://www.ietf.org/rfc/rfc2616.txt]` for details).

<referer/>
The referer element represents the information in the HTTP Referer header, which gives information about the previous page visited by the user. Note that this field is misspelled in exactly the same way as the corresponding HTTP header. Subclasses: authority, stem, querystring.

<useragent/>
The useragent field represents the information in the HTTP User-Agent header (which gives information about the type and version of the user's Web browser), and/or the HTTP accept* headers.

Categories in P3P

Categories can be used to better specify the intended meaning or usage of collected data. More than one category may be associated with a fixed data element. However, each base data element is assigned to only one category whenever possible. Generally, each data type is associated with a list of allowed categories. The mapping between basic data types and the relative allowed categories can be found in the P3P 1.1 Specification (`https://www.w3.org/TR/P3P11/`).

<physical/>
Physical Contact Information: Information that allows an individual to be contacted or located in the physical world – such as telephone number or address.

<online/>
Online Contact Information: Information that allows an individual to be contacted or located on the Internet – such as email. Often, this information is independent of the specific computer used to access the network. (See the category "Computer Information")

<uniqueid/>
Unique Identifiers: Non-financial identifiers, excluding government-issued identifiers, issued for purposes of consistently identifying or recognizing the individual. These include identifiers issued by a Web site or service.

<purchase/>
Purchase Information: Information actively generated by the purchase of a product or service, including information about the method of payment.

<financial/>
Financial Information: Information about an individual's finances including account status and activity information such as account balance, payment or overdraft history, and information about an individual's purchase or use of financial instruments including credit or debit card

information. Information about a discrete purchase by an individual, as described in "Purchase Information," alone does not come under the definition of "Financial Information."

<computer/>
Computer Information: Information about the computer system that the individual is using to access the network - - such as the IP number, domain name, browser type or operating system.

<navigation/>
Navigation and Click-stream Data: Data passively generated by browsing the Web site – such as which pages are visited, and how long users stay on each page.

<interactive/>
Interactive Data: Data actively generated from or reflecting explicit interactions with a service provider through its site – such as queries to a search engine, or logs of account activity.

<demographic/>
Demographic and Socioeconomic Data: Data about an individual's characteristics – such as gender, age, income, postal code, or geographic region.

<content/>
Content : The words and expressions contained in the body of a communication – such as the text of email, bulletin board postings, or chat room communications.

<state/>
State Management Mechanisms: Mechanisms for maintaining a stateful session with a user or automatically recognizing users who have visited a particular site or accessed particular content previously – such as HTTP cookies.

<political/>
Political Information: Membership in or affiliation with groups such as religious organizations, trade unions, professional associations, political parties, etc.

<health/>
Health Information: information about an individual's physical or mental health, sexual orientation, use or inquiry into health care services or products, and purchase of health care services or products.[5]

<preference/>
Preference Data: Data about an individual's likes and dislikes – such as favorite color or musical tastes.

<location/>
Location Data: Information that can be used to identify an individual's current physical location and track them as their location changes – such as GPS position data.

[5] It is questionable whether all of these aspects (especially sexual orientation) should be included in the `health` term. For SPECIAL's purposes, such topics should better be represented with a separate term.

<government/>
Government-issued Identifiers: Identifiers issued by a government for purposes of consistently identifying the individual.

<other-category> string </other-category>
Other: Other types of data not captured by the above definitions. (A human readable explanation should be provided in these instances, between the <other-category> and the </other-category> tags.)

Representing data categories in SPECIAL

P3P is a rich source of data categories that SPECIAL is going to re-use for its first vocabulary. However, differently from P3P where two distinct vocabularies (for data types and categories) are combined, in SPECIAL all kinds of data are uniformly represented as OWL2 classes. In other words, the P3P data types and categories that are imported in our specification are conflated in a single taxonomy. If `<D/>` is a P3P data type or category, the corresponding OWL 2 class is `<svd:D>`, where `svd` abbreviates the namespace

`<http://www.specialprivacy.eu/vocabs/data#>`

Service providers may need to provide a detailed description of which data is gathered. This means that the relative data are largely described at the level of granularity of P3P *data types*. On the other hand, users may prefer more general data classes for two reasons. First, a fine-grained descriptions of data such as `other.statuscode` may results somewhat obscure to the average data subject. Secondly, users may want to express flexible policies which adapt to different contexts. Therefore, we start by supporting data classes at the level of granularity of P3P's *categories*.

For this reasons we extend the set of P3P's categories. In particular, we introduce the following new concepts:

<svd:derived>
Any additional data that has been produced by processing/combining data directly provided b the user. Subclasses: `<svd:profile>`, `<svd:statistical>`.

<svd:profile>
Classifying Data derived from personal characteristics or behavior patterns.

<svd:statistical>
Statistical data and analytics which make use of the user data.

<svd:anonymized>
Data that has been anonimized by removing identifying details from user data.

<svd:social>
User's data concerning her social status, e.g. job descriptors, social relationships.

<svd:judicial>
Data pertaining to judgment in courts of justice.

<svd:activity>
Data concerning user's activities. Subclasses: `<svd:physical-activity>`, <svd:online-activity>, <svd:telecom-activity>, <svd:audiovisual-activity>.

<svd:physical-activity>
Data describing the activities of a user in the *real world*, e.g. travels, sports, concerts, etc.

<svd:online-activity>
Data describing online activities such as browsing, liking on social networks, posting, etc.

<svd:telecom-activity>
Data concerning telecomunication activities such as mobile calls or messages.

<svd:audiovisual-activity>
Data on watching TV programs, subscribing to youtube channels, etc.

In P3P categories are meant to be mutually exclusive and hence specifications recommend to use only one category in any specific occurrence of a data type. Clearly, some classes such as `<svd:physical-activity>` and `<svd:online-activity>` are assumed to be disjoint also in SPECIAL. However, some classes may overlap. For instance, subscribing to youtube channels can be considered both a online activity and an audiovisual activity. Such multi-faceted data classes can be expressed with class *intersections*, e.g.:

```
ObjectIntersectionOf( <svd:online-activity> <svd:audiovisual-activity> ).
```

If necessary, multi-faceted data classes of common use can be abbreviated using *equivalence axioms* to define new terms from the ones listed above. With reference to the previous example, data about YouTube usage could be captured by a new class `<svd:YouTube-activity>` defined by adding the following OWL 2 axiom to the data category ontology (Appendix 2):

```
EquivalentClasses(
  <svd:YouTube-activity>
  ObjectIntersectionOf( <svd:online-activity> <svd:audiovisual-activity> )
)
```

All classes in the data vocabulary *must* be subclasses of `<spl:AnyData>`.

Syntax and the logical semantics of the current data vocabulary are specified in Figure 2.4 and Appendix 2, respectively.

5 Purposes

P3P provides a rich set of purposes that can be used also in ODRL. Our initial purpose vocabulary is modelled around the same concepts, so we start by recalling P3P's terminology.

Figure 2.4: SPECIAL's Data Expression Grammar

DataVocabExpression := **DataClass** |
'ObjectIntersectionOf' '(' **DataClass DataClass {DataClass}** ')'

DataClass := *an expression* `<svd:D>` *where* `R` *is a P3P data category or one of the novel data classes introduced in this section (e.g. profile, statistics, etc.)*

Purposes in P3P

The following list contains the possible values of P3P's `<PURPOSE>` element. Their descriptions are quoted from P3P1.1's specification.[6]

<current/>
Completion and Support of Activity For Which Data Was Provided: Information may be used by the service provider to complete the activity for which it was provided, whether a one-time activity such as returning the results from a Web search, forwarding an email message, or placing an order; or a recurring activity such as providing a subscription service, or allowing access to an online address book or electronic wallet.[7]

<admin/>
Web Site and System Administration: Information may be used for the technical support of the Web site and its computer system. This would include processing computer account information, information used in the course of securing and maintaining the site, and verification of Web site activity by the site or its agents.

<develop/>
Research and Development: Information may be used to enhance, evaluate, or otherwise review the site, service, product, or market. This does not include personal information used to tailor or modify the content to the specific individual nor information used to evaluate, target, profile or contact the individual.

<tailoring/>
One-time Tailoring: Information may be used to tailor or modify content or design of the site where the information is used only for a single visit to the site and not used for any kind of future customization. For example, an online store might suggest other items a visitor may wish to purchase based on the items he has already placed in his shopping basket.

<pseudo-analysis/>
Pseudonymous Analysis: Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals for purpose of research, analysis and reporting, but it will not be used to attempt to identify specific individuals. For

[6] `http://www.w3.org/TR/P3P11/#PURPOSE`
[7] The `<current/>` element can be further specified with the *primary purpose* element illustrated after this list.

example, a marketer may wish to understand the interests of visitors to different portions of a Web site.

<pseudo-decision/>
Pseudonymous Decision: Information may be used to create or build a record of a particular individual or computer that is tied to a pseudonymous identifier, without tying identified data (such as name, address, phone number, or email address) to the record. This profile will be used to determine the habits, interests, or other characteristics of individuals to make a decision that directly affects that individual, but it will not be used to attempt to identify specific individuals. For example, a marketer may tailor or modify content displayed to the browser based on pages viewed during previous visits.

<individual-analysis/>
Individual Analysis: Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data for the purpose of research, analysis and reporting. For example, an online Web site for a physical store may wish to analyze how online shoppers make offline purchases.

<individual-decision/>
Individual Decision: Information may be used to determine the habits, interests, or other characteristics of individuals and combine it with identified data to make a decision that directly affects that individual. For example, an online store suggests items a visitor may wish to purchase based on items he has purchased during previous visits to the Web site.

<contact/>
Contacting Visitors for Marketing of Services or Products: Information may be used to contact the individual, through a communications channel other than voice telephone, for the promotion of a product or service. This includes notifying visitors about updates to the Web site. This does not include a direct reply to a question or comment or customer service for a single transaction – in those cases, <current/> would be used. In addition, this does not include marketing via customized Web content or banner advertisements embedded in sites the user is visiting – these cases would be covered by the <tailoring/>, <pseudo-analysis/> and <pseudo-decision/>, or <individual-analysis/> and <individual-decision/> purposes.

<historical/>
Historical Preservation: Information may be archived or stored for the purpose of preserving social history as governed by an existing law or policy. This law or policy MUST be referenced in the <DISPUTES> element and MUST include a specific definition of the type of qualified researcher who can access the information, where this information will be stored and specifically how this collection advances the preservation of history.

<telemarketing/>
Contacting Visitors for Marketing of Services or Products Via Telephone: Information may be used to contact the individual via a voice telephone call for promotion of a product or service. This does not include a direct reply to a question or comment or customer service for a single transaction – in those cases, <current/> would be used.

<other-purpose> string </other-purpose>
Other Uses: Information may be used in other ways not captured by the above definitions. (A human readable explanation MUST be provided in these instances).

The <current/> element can be further described through the extension <PPURPOSE/> (*primary purpose*) whose possible values are listed below.

<account/>
Account and/or Subscription Management: Information may be used for managing an account. A common example is updating account information. This may also include creation and/or termination of an account or subscription.

<arts/>
Arts and Entertainment: Information may be used for delivering the arts. Examples include such interests as music, literature, drama, movies, and the visual arts.

<browsing/>
Web Browsing: Information may be exchanged automatically for the purpose of browsing web pages.

<charity/>
Charitable Donations: Information may be used for a charitable donation.

<communicate/>
Communications Services: Information may be used for facilitating communication between users. Examples include communication conducted over email, telephone, facsimile, videophone, instant messaging, or any other communications medium.

<custom/>
Customization: Information may be used to customize the user's online experience as explicitly requested by the user. This element should not be used to represent purposes that can be described by tailoring, pseudo-decision, or individual-decision. This element might be used, for example, at a site that allows the user to change the language in which content is presented.

<delivery/>
Delivery: Information may be used for delivering a product or products.

<downloads/>
Software Downloads: Information may be used to allow the user to download an executable program. This element should not be used to describe downloads of web pages, multimedia files, scripts run by a web browser, and plugin content. This element might be used, for example, at a site that offers a downloadable media player. However, it would not be used at a site that offers only music files playable by the media player.

<education/>
Education: Information may be used for educational purposes; examples include teaching, grading, testing, and interactions between educators and students.

<feedback/>
Responding to User: Information may be used for the purposes of responding to the user. This can vary from responding to a query to simply providing the user with feedback.

<finmgt/>
Banking and Financial Management: Information may be used for bank transactions or financial management. Examples include opening, closing, and managing financial accounts, as well as trading securities.

<gambling/>
Online Gambling: Information may be used for online gambling where wagers for money are placed on games of chance.

<gaming/>
Online Gaming: Information may be used for online games that do not involve gambling.

<government/>
Government Services: Information may be used for online government services. Examples include voter registration, vehicle registration, and citizen information services.

<health/>
Healthcare Services: Information may be used to offer the user products or services that relate to their physical and/or mental health.

<login/>
Authentication and Authorization: Information may be used for the purpose of online identity verification. Usernames and passwords are often exchanged to confirm online identity and/or grant access to protected content.

<marketing/>
Advertising, Marketing, and/or Promotion: Information may be used for marketing and promotional purposes.

<news/>
News and Information: Information may be used for delivering news or other information.

<payment/>
Payment and Transaction Facilitation: Information may be used to facilitate a financial transaction. This is different from sales as the payment is sent or received by a third party.

<sales/>
Sales of Products or Services: Information may be used as part of a business transaction with the user. Information is provided for the purpose of completing a sale.

<search/>
Search Engines: Information may be used for querying a search engine.

<state/>
State and Session Management: Information may be used to keep track of sessions. Examples

Figure 2.5: SPECIAL's Purpose Expression Grammar

PurposeVocabExpression := *an expression* `<svpu:P>` *where* `P` *is a P3P purpose not occurring in (2.3)*

include unique identification numbers or information to identify the previous pages viewed. Other uses include serving the user dynamic content.

<surveys/>
Surveys and Questionnaires: Information may be used to conduct surveys and questionnaires.

Purpose Expressions in OWL 2

P3P's purpose and primary purpose elements are regarded as OWL 2 classes. Informally speaking, each of those purposes can in principle be further refined and detailed, so the class corresponding to a purpose contains as instances all of the purpose's variants. For example, one might specialize the class of contact purposes by introducing a subclass corresponding to "contact from our company", that in turn can be further refined to distinguish contact modality (e.g. email, SMS, etc).

Not all P3P purposes are regarded as such in SPECIAL's policies. In particular, in SPECIAL, the construction of user profiles modelled by P3P's `*-analysis` is regarded only as a means to an end that constitutes the actual purpose of the processing. The profile construction in itself is just a possible kind of processing. Similarly, the `*-decision` elements indicate that a decision is made with an algorithm, but it does not specify which purpose that decision is related to. For this reason, the following terms are *not* regared as purposes in SPECIAL:

```
pseudo-analysis, pseudo-decision,
individual-analysis, individual-decision.
```
(2.3)

Concerning notation, if `<P/>` is a purpose or primary purpose element of P3P other than the four elements in (2.3), then the corresponding class is `<svpu:P>`, where `svpu` abbreviates the namespace

`<http://www.specialprivacy.eu/vocabs/purposes#>`.

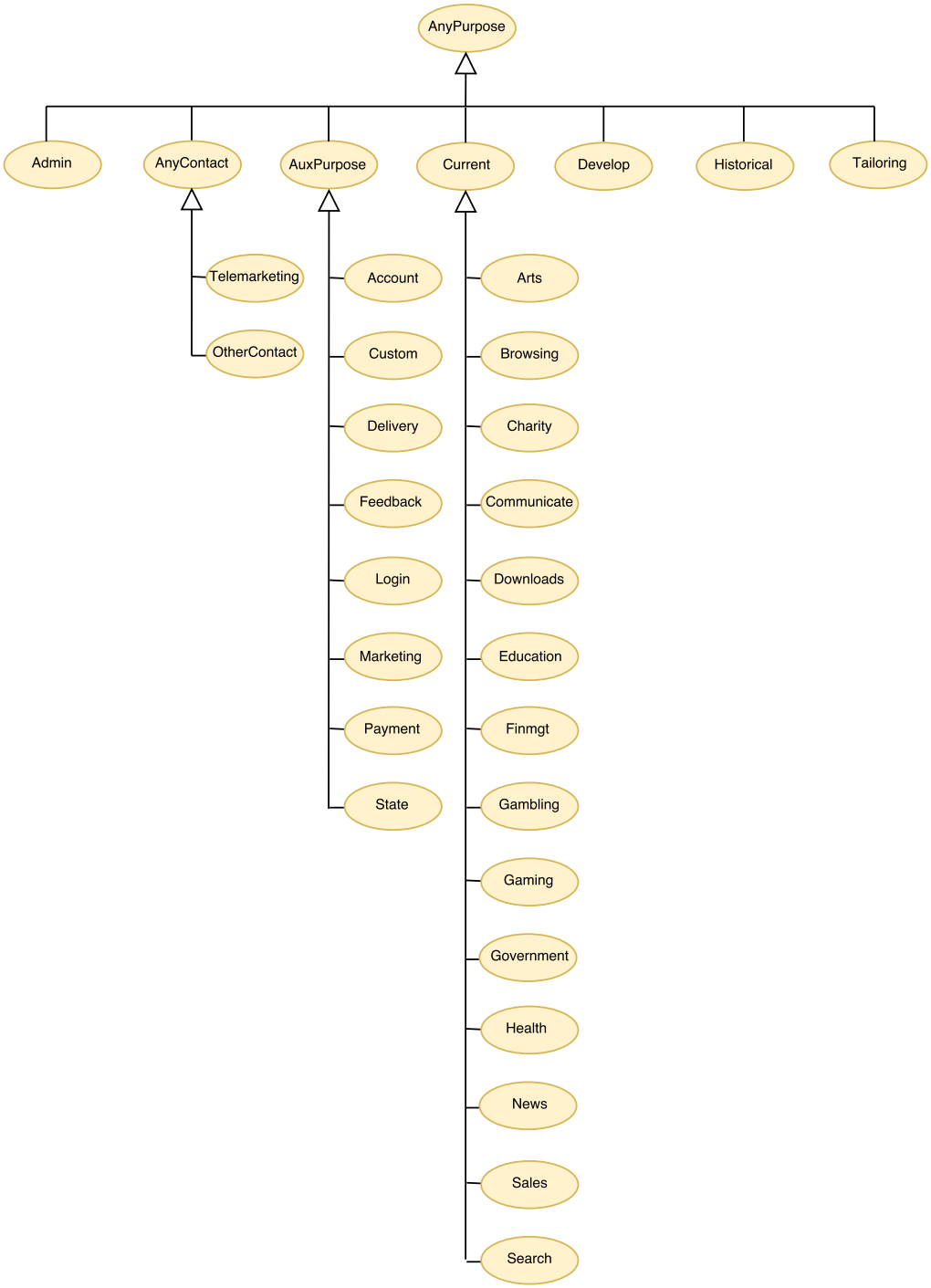
The syntax of purpose terms is specified in Figure 2.5. Their taxonomy is illustrated in Figure 2.6, and their complete logical meaning is specified in the ontology illustrated in Appendix 3.

All classes in the purpose vocabulary *must* be subclasses of `<spl:AnyPurpose>`.

6 Processing

ODRL has a rich vocabulary for describing operations. We model our temporary vocabulary of processing categories around similar concepts, with particular regard to those that are more closely related to data protection.

Figure 2.6: Taxonomy of purpose categories



ODRL's vocabulary of Actions

This section recalls ODRL's action categories and related definitions, as specified in `http://www.w3.org/ns/odrl/2/ODRL21`. Here we report only the actions that are directly relevant to usage control and SPECIAL's pilots. The correspondences and mismatches between ODRL vocabularies and SPECIAL's use cases are discussed in footnotes. In the following – and from SPECIAL's perspective – *the term "asset" should be understood as "data" unless explicitly stated otherwise*.

use
Definition: The Assigner permits/prohibits the Assignee to use the Asset as agreed. More details may be defined in the applicable agreements or under applicable commercial laws. Refined types of actions can be expressed by the narrower actions.[8]

grantUse
Definition: The Assigner permits/prohibits the Assignee to grant the use the Asset to third parties. This action enables the Assignee to create policies for the use of the Asset for third parties.

acceptTracking
Definition: The Assigner requires that the Assignees accepts that the use of the Asset[9] may be tracked. The collected information may be tracked by the Assigner, or may link to a Party with the role function "trackingParty".

aggregate
Definition: The Assigner permits/prohibits the Assignees to use the Asset or parts of it as part of a composite collection.[10]

anonymize
Definition: The Assigner permits/prohibits the Assignees to anonymize all or parts of the Asset. For example, to remove identifying particulars for statistical or for other comparable purposes, or to use the asset without stating the author/source.

archive
Definition: The Assigner permits/prohibits the Assignees to store the Asset (in a non-transient form). Constraints may be used for temporal conditions.[11]

copy
Definition: The act of making an exact reproduction of the asset.

derive
Definition: The Assigner permits/prohibits the Assignees to create a new derivative Asset from this Asset and to edit or modify the derivative. A new asset is created and may have significant

[8] In SPECIAL's policy language this concet corresponds to `<spl:AnyProcessing>`.

[9] From SPECIAL's perspective, this ODRL term is relevant if we interpret "asset" as a service provided by the data controller, and "tracking" applies to the data subject's use of that service.

[10] In SPECIAL, this term should be intended as an information aggregation such as a join or similar operations, possibly operating over multiple data sources.

[11] In SPECIAL's policy language this is handled by the *Storage* attribute of usage policies.

overlaps with the original Asset. (Note that the notion of whether or not the change is significant enough to qualify as a new asset is subjective). To the derived Asset a next policy may be applied.[12]

digitize
Definition: The Assigner permits/prohibits the Assignees to produce a digital copy of (or otherwise digitize) the Asset from its analogue form.[13]

distribute
Definition: The Assigner permits/prohibits the Assignees to distribute the Asset.[14]

give
Definition: The Assigner permits/prohibits the Assignees to transfer the ownership of the Asset to a third party without compensation and while deleting the original asset.[15]

move
Definition: The Assigner permits/prohibits the Assignees to move the Asset from one digital location to another including deleting the original copy. After the Asset has been moved, the original copy must be deleted.

read
Definition: The Assigner permits/prohibits the Assignees to obtain data from the Asset. For example, the ability to read a record from a database (the Asset).[16]

secondaryUse
Definition: The act of using the asset for a purpose other than the purpose it was intended for.[17]

sell
Definition: The Assigner permits/prohibits the Assignees to transfer the ownership of the Asset to a third party with compensation and while deleting the original asset.[18]

transfer
Definition: The Assigner transfers/does not transfer the ownership in perpetuity to the Assignees.[19]

[12] The notion of "next policy" in ODRL is related to sticky policies and GDPR's binding corporate rules: namely, it is a policy that applies to the asset released to a third party.

[13] This term may be useful to descibe the collection of documents and IDs from data subjects.

[14] SPECIAL needs to express data disclosure operations. ODRL's *distribute*, although semantically related to disclosure, has different connotation, rooted in ODRL's focus on licensing.

[15] While semantically related to information transfer, the deletion of the Assignee's copies of the Asset makes this ODRL concept unsuitable to some typical use cases for SPECIAL.

[16] This action seems suitable to model *queries* in general.

[17] This concept is too generic for legal purposes. According to the GDPR, repurposing requires a specific, full-fledged consent from the data subject (i.e. a complete usage policy).

[18] Same obstacle to SPECIAL's adoption as for `give`: in SPECIAL's use cases the Assignee typically retains its copy of the data.

[19] This concept is difficult to apply in SPECIAL. Ownership of data is a complex matter, where data subjects play a particular role. SPECIAL's use cases are rather concerned with *data transfers* that are a kind of communication act.

share
Definition: The act of the non-commercial reproduction and distribution of the asset to third-parties.

Interestingly, some further ODRL actions (`delete`, `inform`, `obtainConsent`) correspond to obligations mentioned in the GDPR. Obligations are dealt with in deliverable D2.2.

OWL 2 encoding for SPECIAL

Let `svpr` abbreviate the namespace

`<http://www.specialprivacy.eu/vocabs/processing#>`.

The vocabulary of processing categories consists of the following concepts.

All classes in the processing vocabulary *must* be subclasses of `<spl:AnyProcessing>`.

These processing categories match their ODRL definition:
<svpr:Aggregate>,
<svpr:Anonymize>,
<svpr:Copy>,
<svpr:Derive>,
<svpr:Move>.

The remaining processing categories are specifically SPECIAL's (and do not exactly match any of ODRL's actions).

<svpr:Analyze>
Definition: The act of analysing data, for example through data mining algorithms, or by performing statistical analyses. It is a subclass of `<svpr:Derive>`.

<svpr:Collect>
Definition: The act of collecting the data subject's information (as specified in the *Data* attribute of the usage policy).

<svpr:Query>
Definition: The act of querying the asset (possibly storing the result, if so specified in the *Storage* attribute of the usage policy).

<svpr:Transfer>
Definition: The act of communicating the information specified in the *Data* attribute of the policy to the recipient specified in the corresponding policy attribute. Differently from ODRL, the transferring party does not delete its copy of the data. No ownerhip transfer is implied.

Similarly to data classes, the processes that fall under multiple categories can be expressed by means of the `ObjectIntersectionOf` operator. For example, a privacy-preserving data mining algorithm falls within the processing category:

`ObjectIntersectionOf( <svpr:Analyze> <svpr:Anonymize> ).`

Syntax and the logical semantics of the current data vocabulary are specified in Figure 2.7 and Appendix 4, respectively.

Figure 2.7: SPECIAL's Processing Expressions Grammar

ProcessingVocabExpression := **ProcessClass** |
'ObjectIntersectionOf' '(' **ProcessClass ProcessClass {ProcessClass}** ')'

ProcessClass := *one of the processing categories* `<svpr:P>` *introduced in this section*

7 Vocabulary Annotations

OWL 2 supports annotations that can be attached to virtually every element of the language. In SPECIAL such annotations can be used to add explanatory comments to vocabulary terms. Similar annotations can be exploited to construct policy visualization and policy explanation tools that automatically produce a human-readable text from the internal OWL 2 format of the policy. A more detailed treatment of annotations is deferred to the next versions of this deliverable.

8 Examples

This section illustrates some examples directly inspired by the pilots. They are meant to show the pilot leaders how to approach the formalization of their usage policies. The need for more details about the pilots and the preliminary nature of the vocabularies prevent these examples from being detailed enough for a legal evaluation. More specific examples will be included in the next version of the deliverable.

8.1 A policy for collecting public information

The policy

> *The data controller will collect financial and judicial information from public sources and analyse it for "know your customer" purposes. This information will be stored on the controller's servers and released to third parties.*

can be formalised with the following factorised general policy:

```
ObjectIntersectionOf(
  ObjectSomeValueFrom( spl:hasData
    ObjectUnionOf( svd:Financial svd:Judicial ))
  ObjectSomeValueFrom( spl:hasProcessing
    ObjectUnionOf( tr:Collect-public svpr:Analyze ))
  ObjectSomeValueFrom( spl:hasPurpose tr:KYC )
  ObjectSomeValueFrom( spl:hasStorage
    ObjectIntersectionOf(
      ObjectSomeValueFrom(spl:hasLocation spl:ControllerServers)
      DataSomeValuesFrom( spl:durationInDays
        DatatypeRestriction( xsd:integer
          xsd:mininclusive "0"^^xsd:integer ))
    )
  )
  ObjectSomeValueFrom( spl:hasRecipient svr:AnyRecipient )
)
```

In the above policy, the processing class `tr.Collect-public` is an application-specific extension of the basic process class `svpr:Collect` defined by adding to the purpose ontology the new class's declaration and the axiom:

```
SubClassOf( tr.Collect-public svpr:Collect ).
```

Similarly, the purpose class `tr.KYC` is an application-specific extension of the basic purpose class `svpu:finmgt` defined by adding to the purpose ontology the new class's declaration and the axiom:

```
SubClassOf( tr.KYC svpu:finmgt ).
```

8.2 Public disclosure of anonymised location data

The policy

> *Location data are analysed anonymously with a privacy-preserving data mining algorithm for the purpose of issuing traffic alerts. The results of the processing can be disclosed and stored with no restriction on location, time, and recipients.*

can be formalised with the following basic policy:

```
ObjectIntersectionOf(
  ObjectSomeValueFrom( spl:hasData svd:Location )
  ObjectSomeValueFrom( spl:hasProcessing
    ObjectIntersectionOf( svpr:Analyze svpr:Anonymize ))
  ObjectSomeValueFrom( spl:hasPurpose dt:TrafficAlerts )
  ObjectSomeValueFrom( spl:hasStorage
    ObjectIntersectionOf(
      ObjectSomeValueFrom( spl:hasLocation spl:AnyLocation )
      DataSomeValuesFrom(  spl:durationInDays
        DatatypeRestriction( xsd:integer
          xsd:mininclusive "0"^^xsd:integer ))
    )
  )
  ObjectSomeValueFrom( spl:hasRecipient svr:AnyRecipient )
)
```

The class `dt:TrafficAlerts` is application-specific and shall be added to the basic purpose vocabulary. A reasonable place in the purpose taxonomy is under the `AnyContact` class. This extension be implemented by adding the class's declaration and the following axiom to the ontology in Appendix 3:

```
SubClassOf( dt:TrafficAlerts svpu:AnyContact ).
```

8.3 A recommendation policy

The policy

> *Demographic data, navigation data, TV program choices, and location are analysed to create a user profile for the purpose of issuing recommendations. Profiles are stored indefinitely in the EU by the data controller and its data processors, who are the only recipients of this information.*

can be formalised as follows with a factorised general policy:

```
ObjectIntersectionOf(
  ObjectSomeValueFrom( spl:hasData
    ObjectUnionOf(
      svd:Demographic svd:Navigation px:TV svd:Location ))
  ObjectSomeValueFrom( spl:hasProcessing px:Profiling )
  ObjectSomeValueFrom( spl:hasPurpose px:Recommendation )
  ObjectSomeValueFrom( spl:hasStorage
    ObjectIntersectionOf(
      ObjectSomeValuesFrom( spl:hasLocation
        ObjectIntersectionOf( svl:OurServers svl:EU ))
      DataSomeValuesFrom(   spl:durationInDays
        DatatypeRestriction( xsd:integer
          xsd:mininclusive "0"^^xsd:integer ))))
  ObjectSomeValueFrom( spl:hasRecipient svr:Ours )
)
```

In this example, the basic vocabularies need to be extended with three new classes:

1. `px:TV` – the class of TV program choices – that naturally extends the data category `svd:AudiovisualActivity`;
2. `px:Profiling` – the class of processes that create a user profile by analysing the user's behavior – that is a subclass of `svpr:Analyze`;
3. `px.Recommendation`, that is a natural subclass of `svpu:Marketing`.

The basic vocabularies can be extended with these new classes simply by adding to the respective ontologies (of data categories, processes and purposes) their declarations and the axioms

```
SubClassOf( px:TV svd:AudiovisualActivity )
SubClassOf( px:Profiling svpr:Analyze )
SubClassOf( px.Recommendation svpu:Marketing )
```

9 Further extensions

The vocabularies are still under development, based on the elaboration of concrete policies for the pilots, and on feedback from all partners. The issues marked in footnotes are going to be addressed along this process. Further terms may have to be added to the current lists. For future reference, we recall that it has been suggested to add some data categories related to vcards.

Policies need to be associated to further metadata, too, such as the policy' version, its validity period, etc., for policy management and correct compliance checking along extended time windows. Such extensions are going to be addressed in other deliverables, where the usage policies described here in D2.1 will be encapsulated into consent expressions, company policies, and transparency records. Syntactically, such extensions should be expected to consist in further elements like:

```
ObjectSomeValueFrom( AttributeName AttributeClass ) .
```

Chapter 3

Appendix

1 The Usage Policy Language Ontology

```
# NAMESPACE DEFINITIONS

Prefix(spl:=<http://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

# ONTOLOGY IRI AND ITS VERSION

Ontology( <http://www.specialprivacy.eu/langs/usage-policy-ontology>
<http://www.specialprivacy.eu/langs/usage-policy-ontology/1.0>

# LIST OF ALL CLASSES AND PROPERTIES

Declaration(Class(spl:AnyData))
Declaration(Class(spl:AnyDuration))
Declaration(Class(spl:AnyLocation))
Declaration(Class(spl:AnyProcessing))
Declaration(Class(spl:AnyPurpose))
Declaration(Class(spl:AnyRecipient))
Declaration(Class(spl:AnyStorage))
Declaration(Class(spl:Authorization))
Declaration(Class(spl:Null))
Declaration(ObjectProperty(spl:hasData))
Declaration(ObjectProperty(spl:hasDuration))
Declaration(ObjectProperty(spl:hasLocation))
Declaration(ObjectProperty(spl:hasProcessing))
Declaration(ObjectProperty(spl:hasPurpose))
Declaration(ObjectProperty(spl:hasRecipient))
Declaration(ObjectProperty(spl:hasStorage))
Declaration(DataProperty(spl:durationInDays))


# HERE WE SAY:
# 1) THAT ALL PROPERTIES ARE FUNCTIONAL
# 2) WHICH ARE THEIR DOMAIN AND RANGE

FunctionalObjectProperty(spl:hasData)
ObjectPropertyDomain(spl:hasData spl:Authorization)
ObjectPropertyRange(spl:hasData spl:AnyData)

FunctionalObjectProperty(spl:hasDuration)
ObjectPropertyDomain(spl:hasDuration spl:AnyStorage)
ObjectPropertyRange(spl:hasDuration spl:AnyDuration)

FunctionalObjectProperty(spl:hasLocation)
ObjectPropertyDomain(spl:hasLocation spl:AnyStorage)
ObjectPropertyRange(spl:hasLocation spl:AnyLocation)
```

```
FunctionalObjectProperty(spl:hasProcessing)
ObjectPropertyDomain(spl:hasProcessing spl:Authorization)
ObjectPropertyRange(spl:hasProcessing spl:AnyProcessing)

FunctionalObjectProperty(spl:hasPurpose)
ObjectPropertyDomain(spl:hasPurpose spl:Authorization)
ObjectPropertyRange(spl:hasPurpose spl:AnyPurpose)

FunctionalObjectProperty(spl:hasRecipient)
ObjectPropertyDomain(spl:hasRecipient spl:Authorization)
ObjectPropertyRange(spl:hasRecipient ObjectUnionOf(spl:AnyRecipient spl:Null))

FunctionalObjectProperty(spl:hasStorage)
ObjectPropertyDomain(spl:hasStorage spl:Authorization)
ObjectPropertyRange(spl:hasStorage ObjectUnionOf(spl:AnyStorage spl:Null))

FunctionalDataProperty(spl:durationInDays)
DataPropertyDomain(spl:durationInDays spl:AnyStorage)
DataPropertyRange(spl:durationInDays xsd:positiveInteger)

# THE FOLLOWING CLASSES ARE MUTUALLY DISJOINT

DisjointClasses(
   spl:AnyData
   spl:AnyDuration
   spl:AnyLocation
   spl:AnyProcessing
   spl:AnyPurpose
   spl:AnyRecipient
   spl:AnyStorage
   spl:Authorization
   spl:Null )



# AN AUTHORIZATION IS ANYTHING THAT HAS THE
# DATA, PROCESSING, PURPOSE, RECIPIENT, STORAGE ATTRIBUTES

EquivalentClasses(<spl:Authorization>
   ObjectIntersectionOf(
      ObjectSomeValuesFrom(<spl:hasData> owl:Thing)
      ObjectSomeValuesFrom(<spl:hasProcessing> owl:Thing)
      ObjectSomeValuesFrom(<spl:hasPurpose> owl:Thing)
      ObjectSomeValuesFrom(<spl:hasRecipient> owl:Thing)
      ObjectSomeValuesFrom(<spl:hasStorage> owl:Thing)
   )
)

)# end of ontology
```

2 SPECIAL's Data Categories V1

```
Prefix(svd:=<http://www.specialprivacy.eu/vocabs/data#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(spl:=<https://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(svd:=<http://www.specialprivacy.eu/vocabs/data#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.specialprivacy.eu/vocabs/data>
Import(<http://www.specialprivacy.eu/langs/usage-policy/1.0>)

Declaration(Class(svd:Activity))
Declaration(Class(svd:Anonymized))
Declaration(Class(svd:AudiovisualActivity))
Declaration(Class(svd:Computer))
Declaration(Class(svd:Content))
Declaration(Class(svd:Demographic))
Declaration(Class(svd:Derived))
Declaration(Class(svd:Financial))
Declaration(Class(svd:Government))
Declaration(Class(svd:Health))
Declaration(Class(svd:Interactive))
Declaration(Class(svd:Judicial))
Declaration(Class(svd:Location))
Declaration(Class(svd:Navigation))
Declaration(Class(svd:Online))
Declaration(Class(svd:OnlineActivity))
Declaration(Class(svd:Physical))
Declaration(Class(svd:PhysicalActivity))
Declaration(Class(svd:Political))
Declaration(Class(svd:Preference))
Declaration(Class(svd:Profile))
Declaration(Class(svd:Purchase))
Declaration(Class(svd:Social))
Declaration(Class(svd:State))
Declaration(Class(svd:Statistical))
Declaration(Class(svd:TelecomActivity))
Declaration(Class(svd:UniqueId))

# CLASS HIERARCHY / TAXONOMY

SubClassOf(svd:Activity spl:AnyData)
SubClassOf(svd:Anonymized spl:AnyData)
SubClassOf(svd:AudiovisualActivity svd:Activity)
SubClassOf(svd:Computer spl:AnyData)
SubClassOf(svd:Content spl:AnyData)
SubClassOf(svd:Demographic spl:AnyData)
SubClassOf(svd:Derived spl:AnyData)
SubClassOf(svd:Financial spl:AnyData)
SubClassOf(svd:Government spl:AnyData)
```

```
SubClassOf(svd:Health spl:AnyData)
SubClassOf(svd:Interactive spl:AnyData)
SubClassOf(svd:Judicial spl:AnyData)
SubClassOf(svd:Location spl:AnyData)
SubClassOf(svd:Navigation spl:AnyData)
SubClassOf(svd:Online spl:AnyData)
SubClassOf(svd:OnlineActivity svd:Activity)
SubClassOf(svd:Physical spl:AnyData)
SubClassOf(svd:PhysicalActivity svd:Activity)
SubClassOf(svd:Political spl:AnyData)
SubClassOf(svd:Preference spl:AnyData)
SubClassOf(svd:Profile svd:Derived)
SubClassOf(svd:Purchase spl:AnyData)
SubClassOf(svd:Social spl:AnyData)
SubClassOf(svd:State spl:AnyData)
SubClassOf(svd:Statistical svd:Derived)
SubClassOf(svd:TelecomActivity svd:Activity)
SubClassOf(svd:UniqueId spl:AnyData)

# DISJOINT CLASSES

DisjointClasses( svd:Government svd:UniqueId )
DisjointClasses(svd:OnlineActivity svd:PhysicalActivity)

)# end of ontology
```

3 SPECIAL's Purpose Ontology V1

```
Prefix(svpu:=<http://www.specialprivacy.eu/vocabs/purposes#>)
Prefix(spl:=<http://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)


Ontology(<http://www.specialprivacy.eu/vocabs/purposes>
Import(<http://www.specialprivacy.eu/langs/usage-policy/1.0>)

Declaration(Class(svpu:Account))
Declaration(Class(svpu:Admin))
Declaration(Class(svpu:AnyContact))
Declaration(Class(svpu:Arts))
Declaration(Class(svpu:AuxPurpose))
Declaration(Class(svpu:Browsing))
Declaration(Class(svpu:Charity))
Declaration(Class(svpu:Communicate))
Declaration(Class(svpu:Current))
Declaration(Class(svpu:Custom))
Declaration(Class(svpu:Delivery))
Declaration(Class(svpu:Develop))
Declaration(Class(svpu:Downloads))
Declaration(Class(svpu:Education))
Declaration(Class(svpu:Feedback))
Declaration(Class(svpu:Finmgt))
Declaration(Class(svpu:Gambling))
Declaration(Class(svpu:Gaming))
Declaration(Class(svpu:Government))
Declaration(Class(svpu:Health))
Declaration(Class(svpu:Historical))
Declaration(Class(svpu:Login))
Declaration(Class(svpu:Marketing))
Declaration(Class(svpu:News))
Declaration(Class(svpu:OtherContact))
Declaration(Class(svpu:Payment))
Declaration(Class(svpu:Sales))
Declaration(Class(svpu:Search))
Declaration(Class(svpu:State))
Declaration(Class(svpu:Tailoring))
Declaration(Class(svpu:Telemarketing))

SubClassOf(svpu:Account svpu:AuxPurpose)
SubClassOf(svpu:Admin spl:AnyPurpose)
SubClassOf(svpu:AnyContact spl:AnyPurpose)
SubClassOf(svpu:Arts svpu:Current)
SubClassOf(svpu:AuxPurpose spl:AnyPurpose)
SubClassOf(svpu:Browsing svpu:Current)
SubClassOf(svpu:Charity svpu:Current)
```

```
SubClassOf(svpu:Communicate svpu:Current)
SubClassOf(svpu:Current spl:AnyPurpose)
SubClassOf(svpu:Custom svpu:AuxPurpose)
SubClassOf(svpu:Delivery svpu:AuxPurpose)
SubClassOf(svpu:Develop spl:AnyPurpose)
SubClassOf(svpu:Downloads svpu:Current)
SubClassOf(svpu:Education svpu:Current)
SubClassOf(svpu:Feedback svpu:AuxPurpose)
SubClassOf(svpu:Finmgt svpu:Current)
SubClassOf(svpu:Gambling svpu:Current)
SubClassOf(svpu:Gaming svpu:Current)
SubClassOf(svpu:Government svpu:Current)
SubClassOf(svpu:Health svpu:Current)
SubClassOf(svpu:Historical spl:AnyPurpose)
SubClassOf(svpu:Login svpu:AuxPurpose)
SubClassOf(svpu:Marketing svpu:AuxPurpose)
SubClassOf(svpu:News svpu:Current)
SubClassOf(svpu:OtherContact svpu:AnyContact)

DisjointClasses(svpu:OtherContact svpu:Telemarketing)

SubClassOf(svpu:Payment svpu:AuxPurpose)
SubClassOf(svpu:Sales svpu:Current)
SubClassOf(svpu:Search svpu:Current)
SubClassOf(svpu:State svpu:AuxPurpose)
SubClassOf(svpu:Tailoring spl:AnyPurpose)
SubClassOf(svpu:Telemarketing svpu:AnyContact)


DisjointClasses(
   svpu:Account
   svpu:Custom
   svpu:Delivery
   svpu:Feedback
   svpu:Login
   svpu:Marketing
   svpu:Payment
   svpu:State
)

DisjointClasses(
   svpu:Admin
   svpu:AnyContact
   svpu:AuxPurpose
   svpu:Current
   svpu:Develop
   svpu:Historical
   svpu:Tailoring
)

DisjointClasses(
   svpu:Arts
```

```
    svpu:Browsing
    svpu:Charity
    svpu:Communicate
    svpu:Downloads
    svpu:Education
    svpu:Finmgt
    svpu:Gambling
    svpu:Gaming
    svpu:Government
    svpu:Health
    svpu:News
    svpu:Sales
    svpu:Search
)
)# end of ontology
```

4 SPECIAL's Processing Ontology V1

```
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(spl:=<http://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
Prefix(svpr:=<http://www.specialprivacy.eu/vocabs/processing#>)


Ontology(<http://www.specialprivacy.eu/vocabs/processing>
Import(<http://www.specialprivacy.eu/langs/usage-policy/1.0>)

Declaration(Class(svpr:Aggregate))
Declaration(Class(svpr:Analyze))
Declaration(Class(svpr:Anonymize))
Declaration(Class(svpr:Collect))
Declaration(Class(svpr:Copy))
Declaration(Class(svpr:Derive))
Declaration(Class(svpr:Move))
Declaration(Class(svpr:Query))
Declaration(Class(svpr:Transfer))

SubClassOf(svpr:Aggregate spl:AnyProcessing)
SubClassOf(svpr:Analyze svpr:Derive)
SubClassOf(svpr:Anonymize spl:AnyProcessing)
SubClassOf(svpr:Collect spl:AnyProcessing)
SubClassOf(svpr:Copy spl:AnyProcessing)
SubClassOf(svpr:Derive spl:AnyProcessing)
SubClassOf(svpr:Move spl:AnyProcessing)
SubClassOf(svpr:Query spl:AnyProcessing)
SubClassOf(svpr:Transfer spl:AnyProcessing)

)# end of ontology
```

5 SPECIAL's Recipient Ontology V1

```
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(spl:=<http://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(svr:=<http://www.specialprivacy.eu/vocabs/recipients>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.specialprivacy.eu/vocabs/recipients>
Import(<http://www.specialprivacy.eu/langs/usage-policy/1.0>)

Declaration(Class(svr:Delivery))
Declaration(Class(svr:OtherRecipient))
Declaration(Class(svr:Ours))
Declaration(Class(svr:Public))
Declaration(Class(svr:Same))
Declaration(Class(svr:Unrelated))

SubClassOf(svr:Delivery  spl:AnyRecipient)
SubClassOf(svr:OtherRecipient  spl:AnyRecipient)
SubClassOf(svr:Ours  spl:AnyRecipient)
SubClassOf(svr:Public  spl:AnyRecipient)
SubClassOf(svr:Same  spl:AnyRecipient)
SubClassOf(svr:Unrelated  spl:AnyRecipient)

DisjointClasses(
   svr:Delivery
   svr:OtherRecipient
   svr:Ours svr:Public
   svr:Same svr:Unrelated )
)
```

6 SPECIAL's Location Ontology V1

```
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(spl:=<http://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(svl:=<http://www.specialprivacy.eu/vocabs/locations#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.specialprivacy.eu/vocabs/locations>
Import(<http://www.specialprivacy.eu/langs/usage-policy/1.0>)

Declaration(Class(svl:ControllerServers))
Declaration(Class(svl:EU))
Declaration(Class(svl:EULike))
Declaration(Class(svl:ThirdCountries))
Declaration(Class(svl:OurServers))
Declaration(Class(svl:ProcessorServers))
Declaration(Class(svl:ThirdParty))

SubClassOf(svl:ControllerServers spl:OurServers)
SubClassOf(svl:EU spl:AnyLocation)
SubClassOf(svl:EULike spl:AnyLocation)
SubClassOf(svl:ThirdCountries spl:AnyLocation)
SubClassOf(svl:OurServers spl:AnyLocation)
SubClassOf(svl:ProcessorServers spl:OurServers)
SubClassOf(svl:ThirdParty spl:AnyLocation)

DisjointClasses(svl:OurServers svl:ThirdParty)
DisjointClasses(svl:ControllerServers svl:ProcessorServers)
DisjointClasses(svl:EU svl:EULike svl:ThirdCountries)
)
```

7 SPECIAL's Duration Ontology V1

```
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(svdu:=<http://www.specialprivacy.eu/vocabs/duration#>)
Prefix(spl:=<http://www.specialprivacy.eu/langs/usage-policy#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)

Ontology(<http://www.specialprivacy.eu/vocabs/duration>
Import(<http://www.specialprivacy.eu/langs/usage-policy/1.0>)

Declaration(Class(svdu:BusinessPractices))
Declaration(Class(svdu:Indefinitely))
Declaration(Class(svdu:LegalRequirement))
Declaration(Class(svdu:StatedPurpose))

SubClassOf(svdu:BusinessPractices spl:AnyDuration)
SubClassOf(svdu:Indefinitely spl:AnyDuration)
SubClassOf(svdu:LegalRequirement spl:AnyDuration)
SubClassOf(svdu:StatedPurpose spl:AnyDuration)

DisjointClasses(
   svdu:BusinessPractices
   svdu:Indefinitely
   svdu:LegalRequirement
   svdu:StatedPurpose )
)
```