# SPECIAL

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

## Deliverable D1.3

## Policy, transparency and compliance guidelines V1

Document version: V1.0

# 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-M16 |
| Deliverable number: | D1.3 |
| Deliverable title | Policy, transparency and compliance guidelines V1 |
| Contractual Date of Delivery: | 31-08-2017 |
| Actual Date of Delivery: | 01-09-2017 |
| Editor (s): | Sabrina Kirrane |
| Author (s): | Piero Bonatti, Sabrina Kirrane, Rigo Wenning |
| Reviewer (s): | Bert Boss, Bert Van Nuffelen, Axel Polleres |
| Participant(s): | Luigi Sauro |
| Work package no.: | 1 |
| Work package title: | Use cases & Requirements |
| Work package leader: | CeRICT |
| 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 Introduction

Although the digital information society has brought about significant economic and societal advances, it has also resulted in a society where all actions and decisions leave digital traces behind. While gossip in a café is ephemeral, the gossip on the social networking platform may remain for a very long time. This digitisation of our life creates huge amounts of data that are often made available in interoperable formats that enable data re-use for new and unexpected purposes. Big data announces great benefits for individuals and society alike, however it is clearly a double edged sword, as it can also be used for unfair discrimination and the shifting of power into the hands of those who control the data.

The data protection legislation is supposed to balance the benefits and dangers brought about by big data. On the one hand, it is necessary to enable the IT industry to mine the gold they hope to find in huge data lakes. Such data can be used not only to uncover new business models but also to bring about societal improvements. On the other hand, citizens should have the right to self determination, a phrase coined by the German Federal constitutional court, which refers to the capacity of an individual to decide how their data is used. Unfortunately self determination finds its limit in the three V's of big data: Velocity, Variety and Volume. It is hard to imagine people clicking OK 12 times a second in a bid to maintain their self determination. Clearly clicking on an OK-button or alternatively reading 50 pages of legalese that nobody understands anymore are ineffective self determination mechanisms.

SPECIAL proposes a solution to this blocking situation by using the power of the machines not only to analyse personal data, but to help people to maintain their autonomy. In fact, it is not easy to fulfill the legal requirements and to still offer a user interface that is non-obtrusive. One of the key challenges is to ask for consent at the right moment while relaying only relevant information. This is only possible if the system uses dynamic interfaces to communicate with the data subject.

But this is not the only challenge. D1.2 already documented requirements for consent. Among those requirements, the data controller has the obligation to demonstrate the existence of consent upon request to justify the processing of personal data. This stems directly from the general principle in data protection that contains a general prohibition of the processing of personal data in Art.6 (1) of the European General Data Protection Regulation (GDPR).

The objective of this deliverable is to examine existing tools, techniques and technologies that could potentially be used for to provide transparency to data subjects concerning the use of their personal data and to obtain consent in an informed non-obtrusive manner. The GDPR, on several occasions, calls for technical means to support the obtaining of consent from data subjects and the provision of transparency with respect to personal data processing and sharing. It thus opens a narrow way to solve the problem of reconciling Big data with data protection. This deliverable builds on *D1.1 Use case scenarios V1*, which describes the Proximus, Deutsche Telekom, and Thomson Reuters pilots, and *D1.2 Legal requirements for a privacy enhancing Big Data V1*, which analyses the legal frame conditions of the European data protection framework for a lawful processing of personal data in the context of Big Data across the European Union.

In terms of scope, we focus our analysis on existing logging mechanisms, techniques to obtain consent and relevant policy languages. In each case we survey the state of the art, perform a gap analysis and highlight open research questions, challenges and opportunities.

Considering the iterative nature of the project, this deliverable is not meant to serve as a complete list of requirements, but rather as a summary of our initial analysis, that will be updated regularly as the project advances. The extended document will subsequently be published as *D1.7 Policy, transparency and compliance guidelines V2* at the end of month seventeen.

We start by providing an exemplifying use case scenario and outlining the characteristics of a data usage policy that will serve as a frame of reference for the discussion that follows in subsequent chapters.

# 2 Motivation

In order to ground the analysis we first present a general use case scenario (that exemplifies the requirements derived from our three use case pilots described in *D1.1 Use case scenarios V1* and *D1.2 Legal requirements for a privacy enhancing Big Data V1*). Additionally, we summarise the features that would characterise a data usage policy.

## 2.1 Exemplifying Usecase Scenario

Sue buys a wearable appliance for fitness tracking from BeFit. She is presented with an informed consent request, comprised of a data usage policy that describes which data shall be collected, and how they will be processed and transmitted in order to give her fitness-related information. The policy says that the device records biomedical parameters such as heart rate; these data are stored in BeFit's cloud; and processed for two purposes: (i) giving Sue feedback on her activity, such as calories consumption; (ii) (optional) creating an activity profile that will be shared with other companies for targeted ads related to fitness. Sue opts in for (ii) in order to get a discount. The usage policy, signed by both Sue and BeFit, is stored in a *transparency ledger*. After one year, the device stops working. After two years, Sue starts receiving annoying SMS messages from a local gym that advertise its activities. Fortunately, all the data collection, processing, and transmission operations have been recorded in the transparency ledger. By querying the ledger, Sue discovers the following facts: (i) the gym has an activity profile referring to Sue, that, due to the appliance's malfunctioning, reports that she is not doing any physical exercise; (ii) the gym received the profile from BeFit, associated with a policy that allows the gym to send targeted ads to Sue based on the profile; (iii) BeFit built the profile by mining the data collected by the appliance; and (iv) all these operations are permitted by the consent agreement previously signed by Sue and BeFit. Using the information contained in the ledger, BeFit and the gym can prove that they used Sue's data according to the agreed purposes. However, Sue can now ask both BeFit and the gym to delete all of her data. The information contained in the ledger indicates precisely which pieces of information she is referring to, so they can be automatically deleted in real time.

## 2.2 Data Usage Policy and Consent Requests

According to the GDPR, informed consent requests shall specify clearly which data are collected, what is the purpose of the collection, what processing will be performed, and whether or not the data will be shared with others. As such, we further elaborate on our usecase scenario by detailing the core features that characterise the data usage policy that would need to be enforced by the company. The type of data collected, purpose of collection, and information concerning data processing and sharing has been derived from our analysis of four smart devices (FITBIT[1], Apple Watch[2], GARMIN Vivomove[3], and GARMIN ForRunner[4] and two cloud based analytics services Runkeeping[5] and Strava[6].

**Collected data** The type of data collected varies depending on the device. The a following is a non complete list based on our analysis of a number of well known brands:

- Steps
- Distance
- Calories burned
- Sleep
- Optical Heartrate sensor
- GPS
- Cardio fitness score (inferred - VO2Max - velocity and time)

**Purpose of data collection and processing** The purpose for which the data is collected and type of processing performed varies depending on the device. The following is a non complete list based on our analysis of a number of well known brands:

- Record and provide access to both the data collected and aggregations of said data.
- Display graphs of activity (e.g. all day heart rate and resting heart rate).
- Derive additional data such as cardio fitness score, calories burned, and race time predictions.
- Display route and pointwise velocity on a map.
- Enable a recovery adviser to advise the owner when they are ready for their next workout.

Additionally, data, inferred data and activity profiles are used to improve the service providers products and services.

[1] FITBIT, https://www.fitbit.com/at/home

[2] Apple Watch, https://www.apple.com/lae/watch/

[3] GARMIN Vivomove, https://buy.garmin.com/en-US/US/p/532348

[4] GARMIN ForRunner, https://explore.garmin.com/en-US/forerunner/

[5] Runkeeping, https://runkeeper.com/

[6] Strava, https://www.strava.com/

**How is data processed** Simple calculations are performed on the device, however in some cases more complex processing that relies on third party data or data belonging to multiple users, happens remotely on the service provider's infrastructure.

**Where are collected data and profiles stored** The data is collected from the device's sensors and is either stored on the device or remotely where the device owner elects to share their data with others, for example via community apps such as Runkeeper.

**For how long are the data stored** The data is typically stored until it is deleted at the request of the device owner.

**Disclosure to third parties** Although not all devices enable data sharing, the following is a summary of the data sharing practices that we came across during our analysis:

- Data, inferred data and activity profiles are shared with third-parties in order to provide targeted fitness advertisements.
- The device enables synchronisation with third party services, such as platforms that enable users to simply backup their data (e.g. Dropbox) and those that enable user to store, analyse and share their fitness activities (e.g. GARMIN Connect, Training peeks, Runkeeper, Strava, Sport Tracks).

**Control mechanisms for data subjects** Depending on the device, control mechanisms range from simply being able to record, view and delete data to being able to store the data remotely and/or share the data with others via third party community applications. Such community applications tend to have a richer set of features that the end user can opt-in and likewise opt-out of.

# 3 Transparency

In order to provide transparency to data subjects with respect to the processing of personal data, companies need to record details of processing activities and personal data transactions (i.e., who shared what data with whom, for what purpose and under what usage conditions).

From a technical perspective there is a need for a transparency architecture that records metadata (i.e., policies, event data, context), that can be used to verify that data is processed according to the wishes of the data subject and the applicable regulations.

Generally speaking such a transparency architecture needs to enable data subjects to verify that data processors are complying with usage policies, and data processors to demonstrate that their business processes comply both with the policies accepted by the data subject and the obligations set forth in the GDPR.

Here, we identify a list of requirements relevant for transparent processing and sharing of personal data; examine the degree of support, with respect to said requirements, offered by the different logging architectures and discuss the open research challenges.

Throughout this report the term "ledger" does not refer to any particular piece of software but is rather analogous to a "log" and thus both terms are used interchangeably in this report.

## 3.1 Requirements

In order to provide transparency with respect to data processing to the data subject, while at the same time allowing companies to demonstrate that they are complying with the regulation the following core functions are required.

### Ledger functionality

*Completeness:* All data processing and sharing events should be recorded in the ledger.

*Confidentiality:* Both data subjects and companies should only be able to see the transactions that involve their own data.

*Correctness:* The records stored in the ledger should accurately reflect the processing event.

*Immutability:* The log should be immutable such that it is not possible to go back and reinvent history.

*Integrity:* The log should be protected from accidental and/or malicious modification.

*Interoperability:* The infrastructure should be able to transcend company boundaries, in the sense that the data subject should be able to easily combine logs that they get from multiple companies.

*Non-repudiation:* When it comes to both data processing and sharing events it should not be possible to later deny that the event took place.

*Rectification & Erasure:* It should be possible to rectify errors in the stored personal data and/or delete data at the request of the data subject.

*Traceability:* In the case of processing it should be possible to know about any previous processing of the data. As such it should be possible to link events in a manner that supports traceability of processing.

### Ledger Robustness

*Availability:* Availability is the process of ensuring the optimal accessibility and usability of the ledger irrespective of whether the log is stored locally or globally. Here there is also a link to security as it is imperative that a breach of security does not hinder ledger operations.

*Performance:* When it comes to the processing of the event data, various optimisations such as parallel processing and/or indexing can be used to improve processing efficiency.

*Scalability:* Given the volume of events and policies that will need to be handled, the scalability of event data processing is a major consideration.

*Storage:* In order to reduce the amount of information stored in the log, the data itself can be stored elsewhere and only a hash of the data and a pointer to the actual data itself needs be be stored in the ledger.

## 3.2 Data to be Captured

The primary objective of the log is to maintain a record of all data processing and sharing activities, so that the log can be used to automatically verify compliance with access and usage control policies specified by data subjects, and legal obligations specified in the GDPR. Based on our initial analysis we have identified two different categories of log entries:

*Agreement entries* are needed to record: (i) the gathering of data from the data subject and the terms and conditions under which said data may be processed and/or shared; and (ii) the transfer of data to others and the terms and conditions under which said data may be processed and/or shared.

*Data Processing entries* are needed to record what processing has been performed including when and where this processing happened.

When it comes to event logging, there is a large body of work in the Business Process Management (BPM) community that focuses on using process execution events for business process compliance monitoring [44].

In the case of SPECIAL, we could potentially use existing logs as a means to verify compliance of existing business processes (that involve personal data) with respect to privacy preferences and legal obligations. Alternatively, there is a need to create a log that will be used specifically for compliance checking and to develop interfaces between the log and existing line of business applications.

Ideally we would need access not only the data processing events but also information concerning how these events are interrelated. Van der Aalst [67] identifies the common data elements that need to be part of an event log in order to support this requirement. Although event attributes vary depending on the application domain, typical attributes include the *process identifier*, the *case identifier* (instances of a process are usually referred to as cases), the *event identifier*, *type of activity* (e.g. details of the processing or sharing), *time* (e.g. when the activity was initiated), *costs* (the cost of the activity), and *resource* (e.g. who or what system initiated the processing). Additionally, Van der Aalst [67] makes the following observations:

- *Processes*, *cases* and *events* should have unique identifiers.
- Each *event* should be associated with a particular *case*.
- In order to support process reconstruct the it should be possible to order events, for example via a *timestamp*.

## 3.3 Candidate Ledgers and Limitations

The analysis presented in this section is based on a survey of the state of the art with regards to logging mechanisms in general and a detailed gap analysis of potential solutions based on the requirements identified in the previous section.

### 3.3.1 The status quo

When it comes to the persistence of event data there are three high level options, that are not necessarily mutually exclusive: Each company maintains a local ledger, which may be backed up remotely; a global ledger could be maintained by one or more trusted third parties; or a global ledger could be distributed across a number of peers.

**Local Ledger** Each peer could store its provenance records locally, including information pertaining to data sharing (both incoming and outgoing). Remote logging to a trusted third party (TTP) could be used to guarantee recoverability of data if the machine where the log is stored is compromised. Bellare and Yee [8] and Schneier and Kelsey [59] demonstrated how a secret key signing scheme based on Message Authentication Codes (MACs) together with a hashing algorithm can be used to generate chains of log records that are in turn used to ensure log confidentiality and integrity. MACs are themselves symmetric keys that are generated and verified using collision-resistant secure cryptographic hash functions. Bellare and Yee [8] discusses how a MAC secret key signing scheme together with evolving MAC keys (whereby each record is encrypted with a different key that is derived from the old key) can be used to ensure: (i) the confidentiality of the log; (ii) that previous log entries cannot be changed; and that (iii) the deletion of a log entry can be detected. In such a scenario the base MAC key, which is needed to verify the integrity of the log is entrusted to a TTP. Schneier and Kelsey [59] also uses MACs. However, the log is composed of hash chains as opposed to cipher block chains. Whereas Holt [35] proposes an alternative that combines public key cryptography with hash chains. These approaches are further enhanced by Ma and Tsudik [45] which demonstrates how individual log entry signatures can be combined into a single aggregate signature that can be used to verify the component signatures and to protect against log truncation. While the previously mentioned works focused on logging in general, Sackmann et al. [57] applies it specifically to data protection by demonstrating how a secure logging system can be used for privacy-aware logging. Additionally, it introduces the “privacy evidence” concept and discusses how such a log could be used to compare data processing to the user’s privacy policy.

When it comes to the robustness requirements, both Bellare and Yee [8] and Holt [35] evaluate the performance and scalability of the proposed logging and verification algorithms, while Ma and Tsudik [45] compares alternative signature generation and verification algorithms.

**Global Ledger and Trusted Third Party** Alternatively, the ledger may contain provenance records that are maintained by one or more TTPs. Accorsi [2] demonstrates how MAC-based secure logging mechanisms can be tailored so that they can be used by resource-restricted devices that may need to log data remotely. Wouters et al. [74] highlights the fact that data often flows between different processes, and as such events cannot be considered in isolation, thus giving rise to the need to store a trail of events. The authors demonstrate how public key cryptography can be used to log events in a manner whereby the data subject can verify the process status. Hedbom et al. [33], Peeters et al. [54], Pulls et al. [55] also provide logging mechanisms that provides transparency to data subjects. The protocol, which is based on MAC secure logging techniques, ensures confidentiality and unlinkability of events and is designed so that it can be distributed across several servers. In the case of Peeters et al. [54], Pulls et al. [55], each log is composed of a user block, a processor block and the encrypted data. A trusted third party is responsible for generating the MAC, encrypting it with the users public key, signing it with their own private key and sending it to the data subject via the data processor. The data processor block is generated in a similar manner. Both the log and the personal data are encrypted in a manner that only the data subject and the processor can access them. In the case of data sharing, a new blinded public key is created (in a manner such that the data subjects private key can decrypt any data encrypted with the blinded public key). The blinded key, which will be used by the second data processor, also serves to ensure the unlinkability of the logs.

Peeters et al. [54], Pulls et al. [55] both evaluate the performance of the proposed algorithms and examine the logging throughput from a local and a remote perspective. The authors conclude that encryption and signing are expensive operations and as such the log entry generation time does not scale linearly with the size of the logged data. They also highlight that the decryption and verification processes are also expensive.

**Global Ledger and Peer-to-Peer network** Alternatively, the ledger may be distributed across several physical ledgers (i.e., a virtual global ledger), whereby provenance records are replicated by each peer. Schneier and Kelsey [59] highlights the vulnerability associated with using a single TTP and discusses how $n$ untrusted machines could be used to replace the TTP, with $m$ untrusted machines required to reproduce the base MAC secret key. Weitzner et al. [72] also discusses how transparency and accountability can be achieved via distributed accountability peers that communicate using existing web protocols. These accountability peers would be responsible for mediating access to data, maintaining audit logs and facilitating accountability reasoning. Unfortunately the authors only touch upon the required features and no concrete architecture is proposed. Seneviratne and Kagal [60] builds on this idea by describing how a distributed network of peers can be used to store a permanent log of encrypted transactions. The replication of log entries at each peer optimises both redundancy and availability. Although the authors describe how a distributed network of peers can be used to store a permanent log of transactions, they focus primarily on helping users to conform to policies by highlighting not only usage restrictions but also the implications of their actions, as opposed to investigating the functional and technical challenges of the proposed transparency architecture itself. An alternative distributed architecture based on blockchain technology, which can be used to manage access to personal data, is proposed by Zyskind et al. [76]. The authors discuss how the blockchain data model and Application Programming Interfaces (APIs) can be extended to keep track of both data and access transactions. Data that is encrypted using a shared encryption key, is sent to the blockchain, which subsequently stores the data in an off-blockchain key value store and a pointer to the data in the form of a hash in the public ledger. Compound identities are used to ensure

| | Local Ledger | Global Ledger + TTP | Global Ledger + P2P |
|---|---|---|---|
| Completeness | - | - | - |
| Confidentiality | MAC [8, 35, 57, 59],<br>FssAgg [45],<br>PKI [35, 45] | MAC [2, 33, 54, 55],<br>PKI [74],<br>unlinkability [33, 54, 55] | MAC [59],<br>PKI[60],<br>compound identities [59, 76] |
| Correctness | - | - | - |
| Immutability | cipher chains [8],<br>hash chains [35, 59] | hash chains [35, 59] | network of peers [60, 72]<br>blockchain [76] |
| Integrity | forward integrity [8, 35, 45, 57, 59]<br>MAC security proof [8] | forward integrity [2, 33, 54, 55] | forward integrity [59] |
| Interoperability | - | - | - |
| Non-repudiation | - | - | - |
| Rectification & Erasure | - | - | - |
| Traceability | - | event trails [74] | - |

Table 1: Candidate architectures and ledger functionality gap analysis.

| | Local Ledger | Global Ledger + TTP | Global Ledger + P2P |
|---|---|---|---|
| Availability | - | - | - |
| Performance | logging & verification [8, 35],<br>signature generation & verification [45] | logging [54, 55],<br>throughput [54, 55] | - |
| Scalability | encrypting records [35, 45] | - | - |
| Storage | key & signature [45] | resource restricted devices [2] | - |

Table 2: Candidate architectures and ledger robustness gap analysis.

that only the user and service providers that have been granted access to the data can decrypt the data. One of the primary drawbacks is the fact that the authors focus on how to repurpose the blockchain as an access-control moderator as opposed to exploring the suitability of the proposed architecture for data transparency and governance.

In comparison to local or global approaches that employ a third party, the robustness of the proposed approaches has not been explored to date. Therefore it is difficult to assess the effectiveness of P2P ledgers or blockchains from a non-functional perspective.

### 3.3.2 Gap analysis

The analysis provided by Tables 1 and 2 enlightens some of the primary technical challenges that are common across all candidate architectures.

*Correctness, Completeness & Non-Repudiation:* Although both *correctness* and *completeness* are very desirable features, irrespective of the choice of architecture, when it comes to data processing events neither can be guaranteed as there is no way to prevent companies from logging incorrect information or not entering the information into the log. Although fair exchange protocols could potentially be used to ensure *non-repudiation* of data transactions (i.e., neither party can deny the transaction took place), to date they have not been used in connection with existing logging mechanisms.

*Confidentiality & Integrity:* The combination of MAC together with cipher or hash chains appears to be the prevailing mechanism used to ensure the confidentiality and forward integrity of logs. Although [59] highlights that it could be feasible to replace the TTP with $n$ untrusted machines whereby any $m$ are required to reproduce the base MAC secret key, no concrete details are provided. Additionally, in the context of our use case the secure logging verification schemes would need to be extended to cater for *rectification & erasure* without affecting the overall integrity of the log.

*Immutability, Rectification & Erasure:* Although it should not be possible for a company to go back and reinvent history, the GDPR stipulates that data subjects have the right to *rectification & erasure* (often referred to as the right to be forgotten). This could potentially be seen as a hard delete whereby the data needs to be erased from both the system and the logs. This would mean that we need to be able to update and delete records from the log without affecting the overall integrity of the log. One potential solution would be to employ a cryptographic delete and to provide support for updates via versioning.

*Interoperability & Traceability:* Another consideration is the interoperability of the log with other logs. Considering that existing logging research has primarily focused on recording operating system and application events it is not surprising that interoperability has received very little attention to date. Although there has been some research on *traceability*, the focus has primarily been on linking processing events in a single log.

*Performance & Scalability:* Considering the potential volume of events that will need to be handled by the transparency ledger, the scalability of existing logging mechanisms will be crucial to their adoption. When it comes to the processing of event data, various optimisations such as parallel processing and/or indexing may improve processing efficiency. Data transfer speed could be improved via exchanging a compressed version of the data payload. Inherently querying and updating logs over distributed databases is a computational challenge.

*Storage:* In practice it may not be feasible for a single log server or each peer in a distributed network to store all provenance records. One possibility is to split the provenance records into multiple ledgers, distributed among TTPs or peers. However, such an architecture would need to be fault-tolerant in the case of peers disconnecting from the network. Relevance criteria and careful forgetting may help too, insofar as storage requirements may be reduced by storing only the information that is needed for compliance checking in the specific domain of interest, and deleting other information.

*Availability:* Clearly from an availability perspective it is important that the best practices are employed in order to protect the security of the log host. Additionally the log should be backed up to a secure location on a regular basis. It is worth noting that when it comes to log recovery, rather than relying on a TTP, a hash of the log could be submitted to a publicly available blockchain (such as Bitcoin). However, unlike trusted third parties, public blockchains do not come with Service Level Agreements (SLAs).

## 3.4 Challenges and Opportunities

Although in this report we primarily focus on transparency, our long term goal is to use the ledger together with access/usage policies in order to automatically verify compliance of existing business processes with the GDPR, to this end it is necessary to model both policies and events in a machine readable manner.

### 3.4.1 The ledger

The Resource Description Framework (RDF), which underpins the Linked Data Web (LDW), is used to represent and link information, in a manner which can be interpreted by both humans and machines. Particularly, the power of RDF is revealed in combination with agreed and extensible meta-data vocabularies to describe provenance and events related to data records in a log as metadata, in semantically unambiguous terms. By employing RDF techniques to represent the provenance events stored in the ledger we will be able to support not only interoperabilily between ledgers, but also traceabiliy between events in a manner that facilitates automatic compliance checking. To this end, there are a number of existing vocabularies that can be adapted/extended. For example the *PROV*[7] and *OWL-Time*[8] ontologies can be used to represent *provenance* and *temporal* information respectively. The former may require extensions to PROV to model particular aspects related to processing of personal data. The latter is particularly relevant if ledger-information is distributed. For example, when tracking audit trails potentially distributed over different systems, synchronisation of timestamps and ensuring sequentiality are major issues. Apart from the actual representation of time, reasoning and querying about time and temporal aspects is still an issue that needs more research in the Semantic Web arena. Different proposals for temporal extensions of RDF and querying archived, temporal information in RDF exist, cf. for instance [28] and references therein. Additionally there exists a number of general event vocabularies such as the *Event*[9] ontology and the *LODE*[10] ontology [56] that could potentially be adapted/extended in order to model our data processing *events*.

An additional benefit of Linked Data is that it provides a simple, direct way of associating policies with data. However, such integration needs to be done in a way that ensures scalability. Several techniques can be exploited for this purpose. As an example, we mention knowledge compilation approaches that 'compile' semantic metadata into a compact but self-contained policy that can be more efficiently enforced, without any further access to the knowledge repository (cf. the approaches based on partial evaluation in [12]). The usage of RDF and URIs will enable the deployment of a linked network of distributed ledgers instead of a single, monolithic (central or P2P ledger). Here it would be interesting to look into efforts for modularising and linking between distributed ledgers such as the recent interledger protocol [37] proposal.

[7] PROV,https://www.w3.org/TR/prov-overview/

[8] OWL-Time,https://www.w3.org/TR/owl-time/

[9] Events,http://motools.sourceforge.net/event/event.html

[10] LODE, http://linkedevents.org/ontology/

### 3.4.2 Ledger integrity and reliability

Ensuring the ledger's integrity and reliability is of course essential for compliance checking and for enhancing the subjects' trust in the transparency architecture. Reliability is partly the result of *voluntary compliance*. In the countries with strong data protection regulations, due to the sanctions and the loss of reputation and customers that may result from data abuse, data processors are willing to comply with the regulations, and feel the need for technical means to ensure compliance. In such scenarios, a correct and complete ledger is an extremely useful tool for the data processors, who can exploit it both for verifying their internal procedures, and for demonstrating compliance to data subjects and data protection authorities. This incentivises the creation and maintenance of a correct and complete ledger. As a further incentive to correctness, the event records should be signed by the parties involved in the recorded operation. In this way, the ledger's records become formal declarations that constitute evidence with legal strength (in the countries where digital signatures have legal value), that may be exploited in case of disputes. As a special case, some of the ledger's records may represent data usage consent declaration, in the form of a usage policy signed by the data subject and the data processor. Such records are very close to a contract that none of the two parties can repudiate, due to the properties of digital signatures.

Creating a reliable record for joint operations, and creating records with multiple "simultaneous" signatures, require the adoption of *fair exchange protocols* to guarantee that the operation is completed (e.g. data are transferred) if and only if all the involved parties sign the record and the record is included in the ledger. An extensive survey of fair exchange protocols can be found in [43]. Ideally, the protocol should not involve centralised nodes such as TTP, but the existing approaches of this kind, based on multiparty computations, currently do not scale to the volume of data expected in the scenarios of interest. There are, however, protocols with *offline TTP*, that involve the trusted third party only in case of malfunctioning (like lost or corrupted messages) or protocol violations. As of today, we regard such protocols as the most promising.

### 3.4.3 Immutability, rectification & erasure

When it comes to transparent personal data processing *immutability* is a very desirable feature as it can be used by companies to prove that they have not gone back and reinvented history. However, said immutability seems to be in direct contention with the right to *rectification and erasure* according to the GDPR. Considering the focus of this report, we restrict our discussion to the rectification and erasure of the log entries and do not give any special consideration to the Line of Business (LOB) application. By only storing a hash of the data and a pointer to the actual data itself in the ledger it is possible to decouple the data from the log and indeed delete data. Another motivation for doing so is the *storage* requirements can be reduced considerably. In the case of rectification it may suffice to update data in the LOB application(s) and enter a new record in the log indicating that the data was updated at the request of the data subject, including a reference to the old – deleted – records hash that confirms that said record was updated in mutual agreement. Likewise, in terms of erasure, we assume that there are scenarios like rectification where it will suffice to delete data from the LOB application(s) and enter a new record in the log indicating that the data was deleted at the request of the data subject. Although this would result in a dangling pointer from the initial log entry by following the audit trail it would be possible to find out that the dangling pointer is the result of an authorised delete. However, there may also be scenarios where delete means a hard delete that needs to be propagated to the log (e.g., where it is possible to identify the individual from the log entry). One option would be to investigate the application of cryptographic deletes (where the old data should not be available anymore) to the ledger. However, it would need to be possible to distinguish between authorised deletes (at the request of the data subject) and log tampering. As such, any delete or update request needs to be strongly coupled with a request from the data subject. So far, cryptographic deletion has been considered only in cloud computing environments, where files are replicated across virtual and physical nodes, and whatever remains of the files after their standard deletion (which is logical) could be later recovered by an attacker, cf. [21, 69, 70]. We propose a novel use of cryptographic deletion as a means to harmonise mandatory preservation requirements and the right to deletion, so as to avoid extreme solutions where one requirement overrides the other.

## 3.5 Implementation Considerations

Irrespective of how the log is implemented there are a set of core functions that are needed in order for the various system actors to interact with the log. In this section, we briefly touch upon several key functions, and refer the reader to D1.4 *Technical requirements V1* for additional details.

**Views on the ledger** The aim of the log is two fold: (i) to enable data controllers and processors to provide transparency to data subjects with respect to the processing of their personal data; and (ii) to provide a means for data controllers and processors to demonstrate compliance with the GDPR. As such we envisage three distinct views on the log corresponding to the data controller/processor view, the data subject view and the supervisory authority view.

**Managing the ledger** Assuming that a company may have one to many logs, from an administration perspective there is a need for functions that enable the creation of a new log and under certain rare circumstance the deletion of an existing log (e.g. obligations based on external constraints, such as domain specific legislation and court rulings). In terms of delete it is not clear at this point whether delete means a hard delete whereby the log must be permanently removed or if it will suffice to make the log inactive. Although we foresee deletion to be instigated manually, in some cases (e.g. where there are temporal constraints specified in domain specific legislation) it may be possible to semi-automate this process.

**Managing ledger entries** The system should have several Application Program Interfaces (APIs) that enable the various actors to interact with the log. Here key functions include: creating a new log entry relating to data processing or sharing events, under certain rare circumstance deleting an existing log entry (here again we imagine that this requirement would be subject to external constraints), providing different views of the log to the various system actors, and checking compliance of log entries based on relevant access and usage policies. In terms of querying, the flexibility offered by faceted browsing over all log attributes would be highly desirable. However, it is worth noting that querying in general will be much more complex when the data required is distributed across multiple sources, especially considering the fact that Federated Semantic query processing is an topic that is still under investigation.

**Policies and ledger entries** Another major consideration is the association of policies with data processing events so that it is possible to know what are the access and usage constraints that are relevant for individual data items, if there are multiple versions of a policy which one was relevant at the time of processing/sharing, and also the management of policies across different logs.

**Interfaces** The SPECIAL transparency service should have several Application Program Interface (APIs) that enable enterprise systems and the SPECIAL dashboard to interface with the SPECIAL ledger.

# 4 Informed Consent

The GDPR introduces a general prohibition with respect to the processing of personal data via Art. 6. Processing is then allowed according to a set of predefined scenarios (e.g. public interest, legal obligations) or via consent from the data subject whose data is processed. According to Art. 4 (11), the *consent* of the data subject needs to be: (i) freely given; (ii) specific; (iii) informed and unambiguous indication of the data subject's wishes; (iv) by a clear affirmative action; (v) by which he or she signifies agreement to the processing of personal data relating to him or her. This definition hasn't changed and still contains the keywords treated in WP187 [52] of the Art. 29 Working Party.

With new technical means, SPECIAL aims to help data controllers and data subjects alike to remain on top of data protection obligations and rights. The intent is to preserve informational self determination by data subjects (i.e., the capacity of an individual to decide how their data is used), while at the same time unleashing the full potential of Big Data in terms of both commercial and societal innovation.

For SPECIAL, the solution lies in the development of technologies that allow the data controller and the data subject to interact in new innovative ways, and technologies that mediate consent between them in a non-obtrusive manner. Instead of a ready-made, set in stone static consent forms there is a need to develop the technologies necessary to facilitate *dynamic consent*. Only the relevant information for the *specific situation* should be presented. Additionally it should be possible to extend or amend consent at any time. While, the necessary interactive interfaces should provide data subjects with the ability to highlight data that is inaccurate and to specify new or update existing access/usage policies. At the same time the systems needs to enable data controllers to capture semantically rich metadata that indicates what they are permitted to do with the data.

## 4.1 Requirements

The challenge for *dynamic consent* is to marry such a system with the legal requirements. Fortunately, the Art. 29 Working Party (and future Data Protection Board under the GDPR) has already addressed many of those challenges while participating in the W3C Do-Not-Track Working Group. Several insights from the technical and legal discussion in the W3C Working Group can be found in Document Nr. 240 [53] which examines the necessity for a reform of Directive 2002/58EC [26].

Meanwhile, the European Commission has issued the proposal for a regulation concerning respect for private life and the protection of personal data in electronic communications, repealing Directive 2002/58/EC (Regulation on Privacy and Electronic Communications) [22]. The Proposal is currently under discussion in the European Parliament and at the time of writing of this document, there are already 827 amendments to the Commission document tabled for discussion in the EU Parliament[11]. Although the ePrivacy regulation is in a state of flux, primary considerations for the SPECIAL project include:

(i) the technical functionality needed for the implementation of *dynamic consent*;

(ii) requirements drawn from existing sources like the GDPR and the opinions of the Art. 29 Working Party; and

(iii) the identification of gaps and obstacles for dynamic consent and their association to the ePrivacy discussion.

Before discussing the status quo with respect to consent we first identify a number of core requirements.

### Dynamic Consent Functionality

*Categorisation:* In order to ensure that the user is not over burdened with consent requests it should be possible to group like requests into categories and ask for consent once per category.

*Customisation:* Rather than offering an all or nothing approach, it is highly desirable that data subjects have more control over which data is processed/shared and for what purpose.

*Innovation:* In order to remain innovative companies need to be able to obtain consent for very general data processing categories, such as *service optimisation* and *business intelligence.*

*Historical Data:* One of the challenges faced by companies is the fact that much of the data they currently possess can't be used because they do not have the consent to do so, as such they need a way to obtain consent for personal data that was gathered at some point in the past.

[11] see 2017/0003(COD) Respect for private life and the protection of personal data in electronic communications `http://www.europarl.europa.eu/oeil/popups/ficheprocedure.do?reference=2017/0003(COD)&l=en` seen on 2017-08-21

*Revocation:* The data subject should be able to revoke consent for future processing and sharing at any time (i.e. to opt out either in part or in whole).

*Understandability:* The consent request should be presented in a manner that is digestible by the customer, so that it is possible for them to understand the implications of the consent. This is especially important in Big Data scenarios.

### Consent Robustness

*Performance:* When it comes to the automatic processing and reasoning over access and usage policies, various optimisations such as parallel processing and/or indexing should be used to improve processing efficiency.

*Scalability:* Given the volume of events and policies that will need to be handled, the scalability of compliance checking is a major consideration.

*Storage:* In order to reduce the amount of information stored differential storage techniques could be used to ensure that only consent updates are stored, however here it is necessary to balance storage on the one side and performance and scalability on the other.

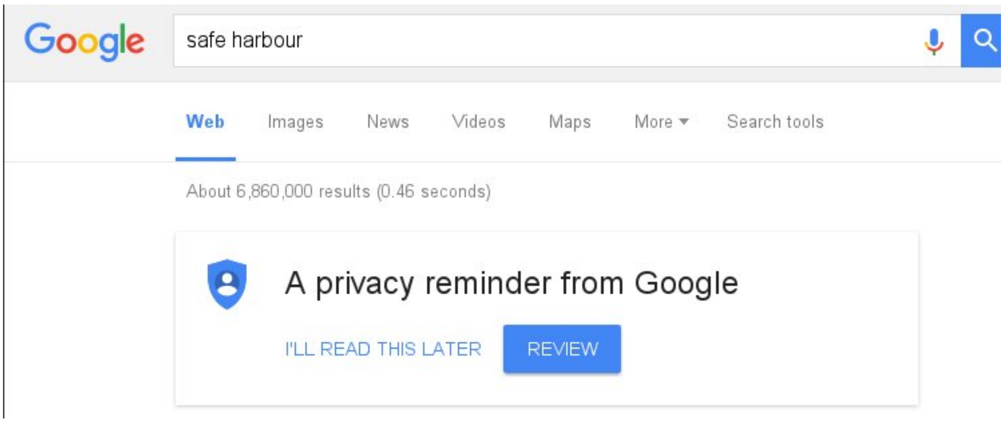
## 4.2 Data to be Captured

As already outlined above and to a larger extent in *D1.2 Legal requirements*, consent requirements create an accountability obligation for the data controller. This accountability obligation guides the collection of the additional data that is needed in order to allow the system to provide an automatic audit trail that contains all necessary information to provide proof of consent. Which information must be presented to the user is highly dependent on the concrete use case, however generally speaking primary considerations include:

- What *data or data category* is collected
- What is the *purpose* of data collection and processing
- Where are collected data *stored*
- For *how long* are the data stored
- With whom is the data *shared*
- What *control mechanisms* are available for the data subjects

For a discussion on possible encodings and vocabularies the reader is referred to *Chapter* 5.

For the system itself, the view is different. The SPECIAL system needs policy contextual information for all items of personal data collected. The awareness of the system is needed to generate the list of things to show to the user. This can be a lot of data that can not be presented to the user in its raw form. This leads to a challenge for the user interface and the type of information presented therein. What remains is that



About 6,860,000 results (0.46 seconds)

A privacy reminder from Google

Figure 1: A classic way to grab an affirmative action

the system then has to record the fact that certain items were shown and followed by affirmative action. Affirmative actions can be very disruptive for the user experience, especially considering that requiring those actions all the time may lead to click fatigue and remove the self awareness of the data subject. One potential solution is to assume that by a first affirmative action, the user agrees to allow for certain choices in the user interface to reduce the amount of annoyance, namely having clickable information that goes away after some time, but remains changeable in the lower levels of the user interface.

## 4.3 Candidate Consent Mechanisms and Limitations

There aren't that many ways to obtain consent. On paper, there is the typical fine print and the courts control the surprising and unfair clauses in such an environment. In the EU, there is a large variety of consumer protection laws but the clauses are harmonised to a certain extent by Directive 93/13EC. This environment works well and is balanced, but it is not fit for the online environment.

### 4.3.1 Classic privacy policies

The classic way is to have a human readable description of the processing where the data collected is described in some very general terms. This is the typical example of the privacy policy of today. But the user can't know whether a certain data item was collected or not and which rights are attached to it. Instead of the concrete operation, we give the data subject a manual that describes how the system normally operates. The art is to write it in such generic terms that the pages of legalese still cover the legal requirements but are almost meaningless beyond. In this case, an affirmative action is recorded in some way, mostly in the form of an *OK-button*, sometimes formed as one of these annoying cookie banners. The *OK-button* or cookie banner usually has a link to a privacy policy page. The recorded click on the *OK-button* then serves as evidence for the fact that the user consented to everything written in the legalese of the privacy policy. A particularly talking example can be seen in the *figure* 1.

In the presented example, the user is even redirected to another landing page with more information that has to be Ok'ed. It records a blanket agreement for **all** processing of data. The criticism here comes from the fact that such consent is neither *specific* nor *informed*, because the data subject is bombarded with too much information and agrees to everything, just to get the service. In fact, even if some nice document would be able to capture **all** purposes and data categories, it would be generic and not specific. One may argue that such consent also has issues with the criteria of *freely given*.

The multipage documents detailing all eventual data collection done by the entire service are there for legal purposes and not for the user. From a users perspective, there is a large body of research that points to the cognitive limitations of users when it comes to informed consent. The most stunning study came from McDonald [47] who explored the question: if website users were to read the privacy policy for each site they visit just once a year, what would the loss of their time be worth? McDonald multiplied the average length of the privacy policies found online (sample) by typical words per minute (WPM) reading speeds. McDonald calculated an average of 201 hours per year for the reading of those policies per individual. This means more than 25 workdays. McDonald also calculated an accumulated time of 44.3 billion hours for the US. As our life is further digitised, we expect this time to increase with the classic solution.

Acquisti et al. [3] and Borgesius [17] point to several recent behavioral studies that cast a shadow of doubt on the effectiveness of notice and choice (i.e. transparency and consent). A study conducted by Acquisti et al. [3] found that more control leads to increased risk taking, an observation that they dubbed the "control paradox". Additionally the authors found that even a short delay between the presentation of a privacy notice and the presentation of irrelevant information was enough to reduce/nullify the effect of the privacy notice. According to Borgesius [17] people tend to agree to anything and many are not aware of the extent to which they are tracked and consequently their consent does not constitute as informed. Borgesius [17] further highlights the strong tendency of system users to accept default settings and to opt for immediate gratification without considering the long term risks.

Another issue with the human readable privacy policies is the fact that they are bundles. Generally speaking, to use a certain service one has to agree to the entire policy. There is no way the user can make any choice other than not using the system. The bundling often means extensive data collection beyond what is necessary. Accepting the all encompassing policy is thus questionable from the point of view of a *freely* given consent.

### 4.3.2 The end of consent

Beyond the mere legal view on data protection, more and more researchers question the value of consent. Either people are lured into giving consent by offering some tiny advantage knowing that consumers typically would not know what it means to agree to the data collection and processing. Or the view is more on data not directly collected from the user, but just tapped from the ambient network that marks the information society. Web and Internet interactions are responsible for the creation of large parts of the Big personal data we deal with. Following a report from the World Economic Forum (WEF) in 2011, Hildebrandt [1] made a data collection categorisation popular that distinguishes between: (i) volunteered data; (ii) observed data; and (iii) inferred data.

The new categorisation dismisses the distinction between personal and non-personal data. The paradigm of data self determination is replaced by Nissenbaums [49] *contextual integrity*. The reign of the algorithms and the upcoming sophistication of artificial intelligence is seen as the main threat. This is decried by Cukier/Mayer-SchÃűnberger [46] and most prominently by Stephen Hawking[12]. To counter the dangers, the limit of data bureaucracy to personal data is given up. A general data administration should control data controllers in order avoid the dangers of data processing in general.

Wenning [73] argues that giving up the distinction between personal and non-personal data may lead to interference with other fundamental rights, namely the freedom of expression and the freedom of information that is so central to the model of the western democracy[13]. While the WEF data categorisation is helpful to understand the origins of Big Data, it isn't helping to align the collection of such data with the GDPR. In fact, Hildebrandt et.al [34] on various occasions doubt about the GDPR system that is still based on the paradigm of informational self determination and thus also doubt about the system and value of consent. They put the decision in the hands of the Data Protection Authorities. This begs the question, are they paternalising the data subjects that they are supposed to protect?

This is especially true when it comes to Big Data and high data volume and velocity. The crisis is sharpened by the ever expanding definition of personal data. Was there a doubt in the past, the GDPR now mentions IP addresses, cookies and other IDs as personally identifiable in consideration Nr. 30. Another factor is that purposes and processing may also evolve over time. Asking the user every 5 minutes whether the processing is still ok may be good from a purist self determination point of view. But it is known that the non-expert people will shy away from such applications and orient themselves towards applications from non-EU juridictions that have a non-existing or much more liberal interpretation of data protection.

### 4.3.3 Dynamic consent

To overcome the cognitive limitations of humans and to allow for a more contextual anchorage of the privacy mediation between data controller and data subject there is a need for *dynamic consent*.

In the biomedical domain *dynamic consent* is a relatively new framework that refers to the use of modern communication mediums to provide transparency, enable consent management and to elicit greater involvement of data subjects from a consent perspective [20]. According to Steinsbekk et al. [63] *dynamic consent* provides greater autonomy and transparency to data subjects with respect to data usage and affords greater involvement than *broad consent*. However, *broad consent* is superior from a ethics perceptive due to well established review and information strategies with opt-out arrangements.

One promising research avenue for SPECIAL would be to classify the actual data collected with the help of some taxonomy, categorisation or ontology. Additionally, machine readable policy information is represented in the same way. It is now sufficient to establish a link between a given data record and related policy information to have

[12] Interview of Stephen Hawking by the BBC on 2014-12-02 `http://www.bbc.com/news/technology-30290540` seen 2017-08-21

[13] Article 11 of the french declaration of Human Rights saying: *La libre communication des pensÃľes et des opinions est un des droits les plus prÃľcieux de l'homme*

a complete picture of the intended processing, including things like retention times, purposes. The system collects all information that is known at collection time. Now any combination of a policy atom and an instance of personal data becomes possible and can be addressed by *dynmic consent*.

*Dynamic consent* has the advantage of allowing a complex system to be *specific*. Instead of the central policy document, the concrete processing intended **now** will be the object of agreement. Several of those agreements can be added to form a larger relationship between data controller and data subject. To be *informed* and *specific*, the dyamic consent needs to present certain information to the data subject and record the *affirmative action* needed into the ledger described in Chapter 3.

A well know approach to data protection is what Hildebrandt [1] and the WEF call *Personal Data Management*. This term is interesting, as it conveys the message that personal data can be managed in a privacy friendly way. But Personal data management, sometimes called *identity management* as a term is also abused to mean a system where data collection is unlimited as long as the user is given some controls on the use of that data. The main difference between the approaches is that the concept of data minimisation is abandoned. The advantage for data controllers is the fact that the aggregation of such data doesn't need further consent, but allows for monetisation of such data. Because even aggregate data can be used to discriminate against people. To remove this ambiguity, the PrimeLife project called its concept *Privacy Enhanced Identity Management*.

Hildebrandt [1] sees a challenge in the fact that there is no sample anymore in Big Data. She calls this «*n=all*», where the sample "n" has all instances of "n" and thus is not a sample anymore, but a complete recording of all the instances. Although, the sheer amount of data makes the enumeration of data items collected impossible, this opens up an opportunity for the SPECIAL engine. If all instances are collected, all instances can be policed and an effective right to be forgotten can be established. Again, volume and variety are a big problem. Using *dynamic consent* and the categorisation of data to allow for innovative user interfaces we may be able to cope with the challenges by volume, variety and velocity.

### 4.3.4 User interaction

As a result of using *dynamic consent*, user interfaces represent a critical component in enabling understandability and customisation of the agreements between data subjects and data controllers. For instance, one of the most widely known existing permission systems is a component of the Android system. However, a survey studying its user attention, comprehension, and behaviour [27] points to the fact that the participants are often unaware of the existence of permissions and just a few (17%) pay attention to them during application installations. RequestPolicy[14], a tool for increasing web browsing privacy through control of cross-site requests [58], proposes using whitelists to protect a user's privacy. It determines blacklists as insufficient as they are not capable of handling new requests that didn't previously exist. In turn, Google provides a solution for visualisation and removal of user data. By logging into the dashboard, stored data can be reviewed within various categories. This includes the user's search history,

[14] RequestPolicy, https://www.requestpolicy.com/

calendars, contacts etc. Additionally, the location history (usually collected by Android smartphones), can be explored via dates and allows for partial deletion. Since the data can often be large, the interface provides summaries and aggregate data where plausible. In terms of more complex policies, few user-friendly schemes go beyond simple access permissions. Creative Commons (CC) provides such a scheme for licenses which can be combined and identified by speaking acronyms and icons (e.g. "BY" for the obligation for attribution, "$" for commercial re-use, etc.). However, no such scheme exists for end-user policies and personal data re-sharing.

#### 4.3.5 GAP analysis

Based on our analysis of the literature it is possible to identify a number of research avenues.

*Categorisation & Understandability:* The low levels of attention, comprehension and behaviour reported by Felt et al. [27], Acquisti et al.[3] and Borgesius [17] indicate that in order to improve usability, user interaction needs to be improved significantly so as to enable users to effectively manage permissions in an understandable manner. For each company the user has data with, permissions should be visualised in recognisable descriptive categories. These categories should be displayed in ways that allow for fine-grained permission modifications. Users should also be presented with specific permission requests that can be issued by companies, detailing the purpose and data required, possibly with specific, easy to understand examples, and let the user agree or disagree. Where appropriate the concepts of layered privacy policies as it is propagated by the Art. 29 Working Party [51] with easy to comprehend icons or short texts complemented by the detailed information may be deployed for further benefit in transparency and clarity.

*Customisation, Historical Data & Revocation:* Existing policy languages such as XACML [50], ODRL [38], KAoS [19], Rei [39] and Protune [11], to name but a few, could be used to formally represent access and usage control policies that support dynamic customisation and revocation of permission to process personal data. One of the primary challenges here is the verification of the suitability of such languages and the corresponding enforcement and administration techniques for mainstream adoption by industry. While, works by Villata et al. [68] and the research done in PrimeLife [36] could serve as a starting point to create and formalise modular, user-understandable modular policy templates, i.e, to establish a 'CreativeCommons-like', easy to remember and understand scheme for end-user formally described machine readable policies supported by acronyms and icons.

*Innovation:* Another major pain point for organisations is the fact that business opportunities themselves are frequently discovered by mining personal data and analysing customer interests. So we run into a chicken-and-egg problem, companies need user consent to analyse personal data, but they are not able to specifically indicate what they need the consent for. Borgesius [17] highlights the fact that companies rarely use privacy as a competitive advantage. Recent articles by Gürses and del Alamo [30] and Hansen [32] provide guidance on privacy engineering, how it can be applied in practice, and highlight some of the challenges that need to be overcome in this emerging field. Primary challenges stem from the fact that existing data protection efforts are scattered and disconnected. Additionally, there are no standardised methodologies, techniques and tools that could serve either as a guide or as a means to assess privacy engineering activities.

## 4.4 Challenges and Opportunities

Although in this report we only provide a high level view of the state of the art, there is a lot that can be learned from recommendations coming from both the legal and the social sciences domains. In particular, SPECIAL aims to implement an architecture that is deeply rooted in the data self determination paradigm as created by the German Federal Constitutional Court in 1984 [9]. In this section we highlight challenges and opportunities regarding machine readable consent requests, dynamic consent, and user interaction.

**Machine readable consent requests** Instead of a ready-made, set in stone static 'consent forms' there is a need to develop the necessary technologies for *dynamic consent*, with a special focus on legal and ethical compliance. Such mechanisms should provide data subjects with the ability to highlight data that is inaccurate and to specify new or update existing access/usage policies. The Resource Description Framework (RDF), which underpins the Linked Data Web (LDW), is used to represent and link information, in a manner which can be interpreted by both humans and machines.Kirrane et al. [42] provide a comprehensive survey of existing access control proposals for RDF, although many of the policy languages presented therein could potentially be used to express and reason over usage policies, regulatory obligations, business rules and provenance events we must first determine the level of expressivity required. Additionally, in order to support automatic verification of compliance it may be necessary to develop (or extend) the formal semantics of the adopted policy language. Indeed, vocabularies for expressing policies such as ODRL [38] currently in the process for standardisation by the W3C's Permissions and Obligations working group still suffer partially from semantic ambiguities [64] or may turn out to be incomplete in practice in terms of expressing policies for personal data handling. Another interesting area for exploration is the reconciliation of machine readable policies with the human readable version. Here we are especially interested in exploring the level of automation that can be achieved from fully manual to fully automated. According to Kaye et al. [41] technical challenges include interfacing with company systems so that the relevant information and feedback it presented to data subjects. Additionally there is a need for compliance checking algorithms that are able to automatically verify that companies are adhering to the access/usage policies specified by data subjects, Given the fundamental nature of machine readable policies to the SPECIAL project a detailed analysis of the state of the art is presented in *Section* 5 of this document.

**Dynamic consent** In the SPECIAL scenario, the backend is capable of adapting in near real time to promises and controls facing the data subject. This includes e.g. the reaction of the backend on receiving a W3C tracking protection signal. In this case, the system could adapt and reduce e.g. the data retention times to the strict necessary and avoid adding the current clickflow to an existing profile. This would also help to establish a system where consent is freely given, because using the system without extensive data collection would still be possible as there is not necessarily a hard bundling like in the traditional privacy policy scenario.

Even within a larger context, e.g. of a complex relation between an ISP or operator and the data subject, the backend would be able to provide specific information about the current intended operation and gain agreement from the data subject. The challenge here is to find new summarising representation of the raw data collected. The linked data world will allow SPECIAL to use ontologies to create high level data taxonomies or ontologies that will allow the data subject to understand what is intended without information overflow. Given that taxonomies can be expanded down to the individual instances, such a system will allow the data subject to drill down into arbitrary detail. This makes the consent achieved very specific without overloading the user and furthers the unambiguous nature of the indications concerning the processing.

A rather difficult issue will be to record a clear affirmative action. But the GDPR itself is rather creative here. It allows to express the affirmative action by setting preferences in the software used e.g. in Art. 21 (5) GDPR. Looking at the cases where the action is missing, it is obvious that by implying consent, one can justify almost all data processing without asking the data subject anything. On the other end is the scenario where the system can not do anything without having the user click Ok all the time, which makes such a system unusable. SPECIAL imagines a two stage process where participating in a SPECIAL system is done by clear affirmative action when subscribing to a certain service. There, the first information given can be very generic and includes the promise of being in control, once further information is collected, used or re-used. This way, the Big Data challenge of changing purposes becomes a matter of ex ante or ex post controls and will prepare the ground for new innovative and non-obtrusive interfaces. *Dynamic consent* results in an adaptive and stateful agreement between the data controller and the data subject.

**Consent and user interactions** Based on our initial analysis there is no clear winner when it comes to obtaining consent. Two interesting avenues for future work include the combination of dynamic and broad consent. Especially in the context of informed Opt-In with exception (whereby the data subject consents to broad data processing categories with the option to Opt-Out of specific processing) and two stage consent request (whereby data subjects Opt-In for preliminary analysis and they have the option to Opt-Out later if they are not happy with the results of the analysis). According to Solove [61] the common cognitive issues include challenges brought about by the fact that people do not read privacy policies, those that do often do not understand them, and those that understand often do not have all information that is needed to make an informed decision. The authors identify the need for privacy self management tools that enable data subjects to manage their privacy preferences globally, for all services providers. When it comes to consent and user interaction there are several challenges that need to be addressed. For example, in terms of personal data management and usability, how do we balance control and cognitive overload? What is the optimal frequency for interaction? How do we ensure that the consent request is both informative yet concise?

The industry persistently complains about the low rate of adoption and consumer reaction in Opt-In systems. Bouckaert and Degryse [18] did a welfare comparison of the three main current policies towards consumer privacy — *anonymity, opt in, and opt out* — within a two-period model of localised competition. They confirmed a finding by Staten and Cate [62] report that only a maximum of 10% of users ever opt out of lists. They also report that an opt-in campaign by a telecom operater resulted in only 5–11% of positive responses. Consequently, Bouckaert finds that, economically, Opt-In performs even below anonymity. One may argue that in 2006 this did not take into account the loss of trust and did not factor in the fact that people stop using the systems. Nevertheless, the challenge is to overcome the 80% difference in economic return between Opt-In and Opt-Out regimes. Bouckaert hints at criteria when finding that «*Consumers never opt out and choose to opt in only when its cost is sufficiently low. Only when opting in is cost-free do the opt-in and opt-out privacy policies coincide.*» GDPR has established an Opt-In regime for most data collection. This means the legislator in the EU has already chosen one side and creating a working but illegal system is not an option. The challenge for SPECIAL will thus be to lower the cost of Opt-In systems by integrating well into peoples communication flows. At the same time, the W3C Do-Not-Track work allows for a much easier Opt-Out in the online environment that is at the origin of most Big personal data. In fact, setting a preference once and for all is sufficient. This means we are approaching the state where the cost of Opt-Out is so low that a new study is needed to see whether the delta of 80% between Opt-In and Opt-Out persists, not taking into the enforcement deficit in the EU.

Although there is much that can be learned from the bio-medical domain concerning dynamic and broad consent, it is still not entirely clear where the boundaries lie in terms of the specificity and understandability of consent requests. One potential avenue for future research is the development of one or more user interfaces that would enable us to empirically evaluate the effectiveness of different categorisation and presentation strategies.

## 4.5 Implementation Considerations

The following provides a high level view of the core functions that are relevant when it comes to managing consent requests. For additional details we refer the reader to D1.4.

**Dynamic consent interface** The dynamic consent user interface should be developed in such a way that it tackles the cognitive limitations reported by Acquisti et al. [3] and Borgesius [17]. Key functions of dynamic consent include: granting consent for processing/sharing, revoking consent for processing/sharing, and updating existing consent.

**Transparency and compliance dashboard** The transparency and compliance dashboard should be developed in such a way that it tackles the users' cognitive limitations. Key functions could include: presenting data processing and sharing events in a easily digestible manner, enabling the user to understand the implications of existing and future consent for processing and sharing.

**Interfaces** The SPECIAL consent service should have several Application Program Interface (APIs) that are necessary to interface with both enterprise systems and other

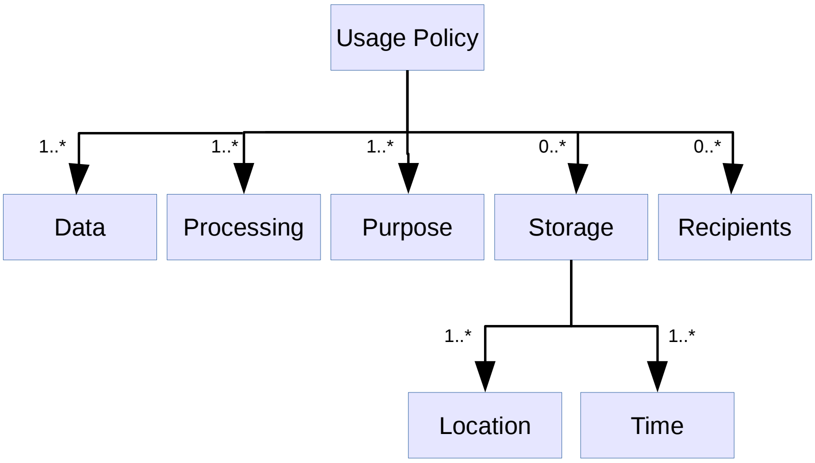


Figure 2: The minimum, core usage policy model (MCM)

SPECIAL components (e.g. the transparency log, compliance checking algorithms etc...).

# 5 Policy Models and Policy Languages

In this section we outline the basic characteristics of the policy models and languages needed in SPECIAL. Consent requests and sticky policies involve *data usage* policies, that are dealt with in Sec. 5.1. Compliance with such policies is meant to be checked automatically, exploiting the knowledge encoded in the transparency infrastructure. The formalisation of the GDPR has different requirements, since the constraints imposed by the data protection regulation are more difficult to assess automatically. Initial guidelines to the formalisation of the GDPR are outlined in Sec. 5.2.

## 5.1 Usage Policies

We are going to describe the requirements on the usage policy language by first introducing an abstract core policy model (focused on SPECIAL's reference scenarios), then discussing its possible encodings with semantic web languages and preexisting policy languages, and finally illustrating how policies are meant to be applied and queried (thereby regarding policies as an abstract data type). These aspects are clearly interrelated and place constraints on each other, that will be discussed in this section.

### 5.1.1 Usage policy model

The reference scenario as well as the pilots described in Deliverable D1.1 involve simple data usage policies, whose main elements are summarised in Figure 2.

- “Data” describes the personal data collected from the data subject.
- “Processing” describes the operations that are performed on the personal data.
- “Purpose” specifies the objective of such processing.
- “Storage” specifies where data are stored and for how long.
- “Recipients” specifies who is going to receive the results of data processing and, as a special case, whom data are shared with.

We will refer to this abstract model as the *minimum core model* (MCM). All the complexity of the usage policy model resides in the description of MCM’s elements, that are illustrated in more detail below.

**The Data element:** In order to describe which categories of data are collected, an ontology of personal data is needed. In the most general case, developing such an ontology is an extremely difficult task, since every possible piece of information that can be attributed to a specific individual is personal information, and as such falls under the scope of the GDPR (and possibly the ePrivacy regulation).

The best approach is to leverage the extensibility and interoperability of semantic metadata, and to develop a core ontology of personal data covering the most common categories of personal data. The core data ontology is meant to cover the most common data categories and should be extended with suitable profiles and/or integrated with further ontologies specialised for particular cases as needed.

Fortunately, the attributes that are commonly used to uniquely identify a person – such as social security numbers and unique fiscal codes, or combinations of name, place and date of birth – constitute a limited space. A quick look at the use cases shows that SPECIAL’s reference scenarios use (a subset of) the information that can be found in IDs like passports, plus telephone numbers, physical and email addresses. Therefore developing a sub-core ontology of identifier data tailored to SPECIAL’s use cases is a feasible task (still one of the ambitions of the project is fostering initiatives to develop a more comprehensive ontology of personal data).

Moreover, a number of use cases based on mobility data (see, for example, the use cases of Proximus and Deutsche Telekom) share common personally identifiable information (PII) that adheres to a small set of telecommunication standards. This facilitates the formalisation (semantification) of the data categories collected and processed in a wide range of applications operating on telephone data. Similarly, it seems feasible to formalise the categories of data more frequently collected by social networks, that provide another wide range of applications with similar data usage modalities.

There have already been some initiatives aimed at categorising personal data that can help in organising the core personal data ontology. For example, P3P[15] contains a nontrivial vocabulary of data categories that includes also dynamic data, collected

[15] P3P, https://www.w3.org/P3P/

by tracking the user's behavior, as well as non personally-identifiable data. P3P's data categories currently do not specifically cover mobility data and television program data, that are needed in SPECIAL's pilots. The available categories constitute a shallow taxonomy that may have to be further articulated in the core data ontology. Within social network applications, it may be interesting to include the FOAF[16] ontology, that although less articulated than P3P's categorisation covers some complementary aspects. A potentially useful classification dimension (currently not covered in a satisfactory way) is the sensitivity of information, which is articulated in some detail in both the existing legislation (e.g. sexual, political, and religious orientation). Deliverable *D6.3 Plan for community group and standardisation contribution* details our plans to engage the web and privacy communities in the form of a workshop that aims to derive and reach consensus on vocabularies to be used to represent personal data, processing of personal data and related policies and regulations.

**The Processing element:** Data processing can be described with at least two approaches: (i) *algorithm oriented*, and (ii) *output oriented*. The former is particularly difficult and ineffective for several reasons. In many cases the same computational task can be carried out with several alternative algorithms, possibly quite different from each other, with complementary properties (e.g. time or memory consumption), but producing exactly the same output. Arguably, such differences are irrelevant in the policy context, since all alternative algorithms produce the same information and distribute it in the same way. Algorithmic descriptions are also intrinsically difficult to process: virtually all interesting properties of – and relations over – such descriptions are undecidable (e.g. algorithm equivalence and output properties). Furthermore, an algorithmic description of data processing is of little meaning to data subjects; this makes algorithm-oriented descriptions unsuitable to the formulation of the usage policies enclosed in informed consent requests.

Due to the unnecessary complications introduced by algorithmic descriptions, we recommend an output-oriented approach to the description of data processing, that is, a categorisation of the data produced by data processing in terms of the information it conveys. For instance, data subjects are interested in knowing which information about themselves can still be found after data have been aggregated or analysed, and possibly the degree and kind of anonymisation of such information.

Therefore, data processing should be described through a suitable ontology of data, similarly to the Data element. However, modelling the results of data processing adds some requirements to the ontology, such as the need to describe aggregates, clusters, and other derivatives, as well as their degree of anonymity. In Proximus' use case, for example, the result of data processing is an interest profile formulated in terms of a vocabulary of keywords extracted from well-defined sources (cf. D1.1); in this case, the description of data processing could be simply collapsed to "an algorithm that produces that type of profile". As far as we know, the existing initiatives such as P3P have not articulated their data taxonomies in sufficient detail to model such aspects. The existence of common needs in important categories of applications encourages the structuring of the data ontology into a core taxonomy plus a set of profiles specific to application categories (as discussed in the Data element paragraph).

[16] http://xmlns.com/foaf/spec/

We expect the level of granularity of the output-oriented approach to be easier to handle for data controllers, too, since it suffices to operate at the service level of the business logic, by introducing an abstract description of the effect of each relevant service, while services need not to be internally analysed.

We conclude the discussion of the output-oriented approach with a comment on the reasons for encoding the type and degree of anonymity of data-processing outputs. Concretely, by "type and degree of anonymity" we mean notions such as *k-anonymity, l-diversity, $\epsilon$-differential privacy* and the like, for specific parameters $k$, $l$, $\epsilon$, etc. As of today, meeting any of such anonymity criteria does not suffice to operate outside the scope of the GDPR, because none of them guarantees that data subjects cannot possibly be (re)identified (and hence data are not considered anonymous in legal terms). Still, providing partial guarantees on anonymity in terms of the above notions does reduce disclosure risks, and may eventually encourage data subjects to release their data. Of course, the corresponding technical definitions are not currently accessible to common users, so in order to experiment with this idea it is necessary to increase the awareness about these anonymisation approaches, and provide understandable description of these methods and their effectiveness in reducing the risk of disclosure (for example, such information may be linked to informed consent requests). Moreover, it is well possible that forthcoming iterations and refinements to data protection regulations will specify that data can be reasonably considered anonymous when they meet some of the above anonymity notions with a specified parameter. Last but not least, highlighting the adoption of the aforementioned anonymisation methods in the policies helps data controllers in demonstrating the adoption of what the GDPR regards as "suggested measures" that improve the robustness of information systems against attacks to confidentiality.

**The Purpose element:** The purpose element shall describe formally why data are collected and/or processed.[17] Not surprisingly, purpose descriptions are to be expressed through a corresponding ontology. Purpose descriptions are part of all of the usage policy languages developed so far, including P3P and ODRL[18], whose purpose categorisations can be exploited as a basis for developing SPECIAL's purpose taxonomy. While personal data and data processing may vary widely across different domains, applications show much less variety in purposes. Objectives such as marketing, service optimisation and personalisation, scientific research, are pervasive across a variety of contexts. Accordingly, we expect the development of an ontology of purposes to be way less problematic than the ontology of data categories.

**The Storage element:** This part of the policy shall describe where data are stored and how long for. Accordingly, the MCM attaches two corresponding subelements to the storage element: Location and Time. The level of granularity of both subelements needs not necessarily be fine-grained, for the reasons outlined below.

The GDPR is mainly concerned with two aspects related to storage location, namely: (i) whether data remains within the company boundaries or is distributed across different organisations (even if they are simply classified as "data processors", or limit their

[17] Recall that the collection of some data may be required by law, and the usage policy may refer to a novel use of those data. In that case the purpose element does not need to justify data collection.

[18] ODRL Information Model,https://www.w3.org/TR/odrl-model/

activity to providing the storage service); (ii) whether data crosses national boundaries, since this may affect the applicable data protection regulations. Thus broad location classes, such as "within/without our company", "our partner's servers" may suffice for point (i), and again the vocabulary adopted by P3P may constitute a useful starting point for developing a location ontology. Nonetheless, in order to monitor and audit company processes, it may be helpful to refine such descriptions by keeping track of the hosts and files where data are stored (e.g. in the form of URIs). This information can be easily refined by adding the nation in which hosts reside, in order to address point (ii).

Concerning storage duration, we do not foresee the need for complex time constraints (unlike some temporal access control policies that support sophisticated periodic constraints, such as [10]). Some laws constrain storage duration by setting a *minimum* storage period (as with some telephone data, cf. deliverable D1.1). The GDPR, on the contrary, requires that storage is strictly bound to the service needs. This implies storage minimisation, hence the need to express *upper bounds* to storage duration, that may be expressed either in terms of the duration of the service that the data have been collected for, or in absolute terms (e.g. in cases where data are stored solely to fulfill the minimum duration requirements specified by some laws, in which case at the end of the required period data must be deleted). Summarising, the storage time element should be able to express a single, possibly open interval, and temporal reasoning collapses to trivial interval membership and interval emptiness checks (for verifying, respectively, that a time point fits within the allowed storage period, and that the allowed storage interval has been correctly specified).

**The Recipients element:** In case of data sharing, the usage policy shall specify the third parties to which data are (or may be) transferred. The GDPR does not clearly state to which level of detail this information has to be specified, and there are opposite needs, such as the companies' desire to keep some of their business relations confidential, and the data subjects' right to trace the flow of their personal information. Some articles mention "categories" of recipients, in which case it is conceivable to adopt a coarse-grained categorisation such as "partners to which services are outsourced", "business partners", "unrelated third parties", possibly "applying our same usage policy". P3P provides a core vocabulary at this level of detail. Should it be necessary to identify data recipients precisely, the ontology may be modelled around the existing standards that describe organizations and possibly their contact persons (e.g. X.509).

### 5.1.2 Guidelines to encoding usage policies in RDFS/OWL2

Given that SPECIAL adopts a semantic layer to obtain a uniform view of all the entities handled in the project, it is natural to encode policies as semantic objects, too. The MCM can be straightforwardly encoded in OWL2[19] by mapping usage policies and each of their elements into classes, and the links between different elements into properties. More specifically, the Time Storage element can be encoded as a data property (where time points are represented as integers in the standard way adopted by operating systems) while all other links are object properties. Then storage intervals can be directly

[19] OWL2, https://www.w3.org/TR/owl2-overview/

encoded through OWL2's numeric *facets*. For example, storage for at least $t$ seconds and no upper bound is expressed by:

$$\texttt{DatatypeRestriction(}\ \texttt{xsd:integer}\ \texttt{xsd:minInclusive}\ t\ \texttt{)}$$

while the time interval $[t, u]$ can be encoded with

$$\texttt{DatatypeRestriction(}\ \texttt{xsd:integer}\ \texttt{xsd:minInclusive}\ t\ \texttt{xsd:maxInclusive}\ u\ \texttt{)}\,.$$

The specification of the above data and object properties fits into the OWL2 EL profile, with the exception of datatype restrictions (hence storage time constraints). Similarly, we expect the ontologies that describe data, processing, purposes and recipients to fit within this profile. This is important because inference over OWL2 EL knowledge bases is tractable and there exist well-engineered, scalable engines tailored to this profile.

In order to exploit such engines one needs to approximate the representation of time intervals with OWL2 EL constructs (for example by replacing datatype restrictions over a single data property with two distinct data properties `from` and `to` ranging over integers). Accordingly, interval emptiness checks do not correspond to the emptiness of the Time element, and must be replaced by ad hoc comparisons of the values of data properties `from` and `to`. Similarly for other reasoning tasks such as policy comparisons.

With this approach, the instances of the class `UsagePolicy` and its elements can be directly represented in RDF. The domain and range of the MCM's data and object properties could alternatively be defined in RDFS, instead of OWL.

The representation of time intervals with two properties `from` and `to` has the drawback of requiring ad hoc inference procedures, as explained above. A cleaner approach consists in extending OWL2 EL engines to support a specific datatype for intervals, with the usual correctness and completeness guarantees. Theoretical results say that if a datatype (called *concrete domain* in the description logic jargon) enjoys a so-called *p-admissibility* property then reasoning remains tractable [6]. The integer domain with both min and max constraints that we exploited in the above datatype restrictions does not match the p-admissible domains illustrated in [6] (in particular, the concrete domain Q supports only $>$, not $<$). Still, we conjecture that a concrete domain of intervals can be formalised in a p-admissible way, and that the engines for OWL2 EL can be extended to support this domain efficiently. This will be the subject of further research.

Note that in principle usage policies could be encoded with a rule language (e.g. some dialect of Datalog). The expressiveness of rule languages is not comparable with the expressiveness of description logics (on which OWL2 is based) and there exist conditions that can be expressed only with rule languages [25]. The reason for focusing on description logics, at this stage, is that (i) the syntax of the usage policy has a structure similar to description logics' syntax, and (ii) some of the reasoning tasks on policies (that will be discussed below) in general are unfeasible for rule-based policy languages.

### 5.1.3 Semantics of the usage policy language

The encoding of usage policies into OWL2 provides a formal semantics to the policy language via the *direct* (model theoretic) *semantics* of OWL2, which is based on the correspondence between the logical operators of OWL2 and the constructs of the description logic $\mathcal{SROIQ}$. The direct semantics, roughly speaking, associates each usage policy with the set of tuples

$$\langle\ \textit{data, operation, purpose,}\ [\textit{storage location, current time}],\ [\textit{current recipients}]\ \rangle$$

that characterise the events permitted by the policy. A more detailed description requires a precise description of policy encoding, so the details will be provided in the forthcoming deliverable devoted to the policy language specification.

### 5.1.4 Relationships with existing vocabularies and policy languages

*P3P* [23]. Each of the MCM's elements has a direct counterpart in P3P. Moreover, as we have already mentioned, P3P provides vocabularies for data categories, purposes and recipients, that may constitute a starting point for developing the ontologies for SPECIAL's usage policy language. These vocabularies have to be extended and structured into more articulated taxonomies for SPECIAL's needs. In particular, P3P does not cover data categories for mobility nor data processing outputs and their anonymity.

*ODRL* [31]. Data categories can be modelled as ODRL's *assets*, that may be described with URIs pointing to RDF graphs formulated in terms of a suitable data ontology, that is currently beyond the scope of ODRL. The closest match for MCM's Processing element is ODRL's Permission element, that may contain actions that partially address the needs of usage policies (e.g. *sell, lend, give, lease* that cover asset/data transfers, *move, duplicate, delete, backup* that have to do with storage handling, and the limited list of elaborations *modify, excerpt, annotate, aggregate*). In its Constraint element ODRL supports the description of actors (by indicating individual and groups) that may play the role of MCM's Recipients. Constraints can also encode temporal intervals, storage locations, and purposes. The details of the specification of actors, locations, and purposes are beyond the scope of ODRL. ODRL provides an element to express obligations.

*KAoS* [66]. This language is based on description logics, and in particular OWL1. The main drawback of KAoS is that it makes also use of operators called *role-value maps* [48] that make complete reasoning undecidable [5, Chap. 5]. Moreover, OWL1 did not support datatype restrictions, so temporal intervals could not be properly modelled. KAoS does not specify ontologies for data categories, purposes, locations, and recipients. Moreover, no complexity and scalability analyses are available. Consequently, encoding the MCM in OWL2 (with an eye to complexity and scalability) may be regarded as a modern evolution of KAoS' approach.

*Rei* [40]. Rei adopts a combination of description logics and rules. This approach increases the expressiveness of the policy language, and using the description logic fragment of Rei it is not difficult to encode the elements of the MCM. Unfortunately, the additional expressiveness provided by rules makes some relevant reasoning tasks undecidable (e.g. policy comparison, cf. the section on reasoning tasks below). Like KAoS, Rei does not provide any specific ontologies or vocabularies for the elements of the MCM (however, it would be straightforward to integrate any such ontologies in Rei).

*Protune* [14]. This is a rule-based language, so it is affected by the aforementioned undecidability of some reasoning tasks. Being a trust negotiation language by design, it does not specifically model the ontologies needed to express MCM's elements. It seems not difficult, however, to model such ontologies with rules. The advanced explanation facility of Protune could turn out to be useful to document policies and automate the generation of dynamic consent requests.

*Conclusions.*The MCM lies in the intersection of several existing languages, such as P3P, ODRL, KAoS, Rei, and Protune, so in principle any of these languages could be used to encode SPECIAL's usage policies, after the necessary auxiliary ontologies have been integrated. Still, there are other relevant considerations that suggest to define SPECIAL's usage policy language around the more recent standard OWL2, and select language constructs carefully in order to achieve an optimal tradeoff between expressiveness and computational complexity. These issue are discussed below, in the paragraphs devoted to language analysis.

### 5.1.5 Reasoning tasks for usage policies

Access control policies are traditionally enforced by submitting all requested operations to a component called *security monitor*, that evaluates the policy and decides whether the given operation is permitted. Thus, the traditional reasoning task on policies boils down to a boolean (yes/no) query over operations and requesters (and possibly context dependent conditions as well). Such boolean queries are not necessarily implemented by deploying and calling a security monitor; a popular alternative consists in *modifying the requested operation* so as to enforce directly the policy. For example, a database query may be rewritten so as to filter out the information that the requester is not allowed to see. In some cases query modification may turn out to be more efficient than security monitors.

Usage policies are more complex and their enforcement may involve proactive actions (such as data deletions), obligations and more. In order to assist data subjects in keeping control on their data, the preferences of data subjects (which are usage policies themselves) may have to be *compared* with the usage policy contained in a consent request. Furthermore, in SPECIAL's scenarios, the main players (subjects, controllers, processors, and officers) should be able to retrieve the policy associated to a given piece of data, and (conversely) collect the data that are subject to a given usage policy. Last but not least, it is important to check policy *correctness* and provide rich policy documentation. All these requirements lead to a richer set of reasoning tasks (or queries) on policies, discussed below.

*Permission checking.* This is the traditional kind of queries submitted to security monitors. In informal terms, this reasoning task answers the questions *can X do Y?* In SPECIAL, this category of queries may occur in different places:

- the business logic of data controllers, where actions may be checked for compliance with the applicable policies before execution;
- audit controls, where the actions recorded in transparency logs are checked for compliance with the applicable policies;

- documentation facilities, that an actor may use to predict whether a possible future action would be allowed by a policy.

Currently it is not clear in which of these different contexts a rewriting approach could be more efficient than a monitor-based approach. This issue requires further research.

*Policy scope.* This reasoning task consists in retrieving all the data that have been collected subject to the policy specified in a previously approved consent request. This may be interesting for a data subject, a data controller, or a data officer who wants to know which of the data encoded in a system can be used according to that policy.

According to the MCM, querying for policy scope requires retrieving all the instances of the class specified in the Data element (which is a term in the ontology of data). If the request is made by a data subject, the answer shall be restricted to her own data. Such instance retrieval, however, does not always suffice to answer policy scope queries, since subsequent modifications of consent may affect the result (data released after the new consent are subject to a different policy, in general).

An interesting generalization of this task takes as input *any* usage policy (not necessarily one previously approved with a consent agreement), and looks for all data that can be used as specified by that policy. In this case the applicable consent declarations shall be *compared* with the given policy to see if the permissions specified by the latter are allowed by the former. Policy comparison is discussed further on.

*Policy consistency checking.* This reasoning task is aimed at policy verification. Consistency checks may be either internal or global, in the following sense:

- A single policy is (internally) inconsistent if any of its elements is inconsistent. For example, the time interval might be empty (the end point precedes the starting point), the class denoting collected data might be the empty class, and so on (see [65] for examples of internally inconsistent P3P specifications). Checking for this kind of inconsistencies helps in detecting errors in the formulation of consent requests.

- A set of policies, possibly applied by different data controllers, is (globally) inconsistent if the policies in the set specify conflicting directives. This kind of consistency checks may help a data subject in detecting personal data that are not equally protected by different data controllers, due to incoherent consent to information usage. Note that global consistency checks address this need only partially; if the policy applied by a data controller is weaker than (although not inconsistent with) the policy applied by another data controller, then data are less protected by the former controller although no inconsistency can be detected. This scenario is addressed by policy comparison (see below).

*Policy comparison.* This reasoning task is aimed at finding the mutual logical relationships between different policies, such as:

- are two policies equivalent?

- does a policy $P_1$ imply a policy $P_2$? (i.e. is $P_1$ stronger than $P_2$)

- are $P_1$ and $P_2$ mutually inconsistent?

Similar comparisons may be applied to sets of policies. There are different reasons for such comparisons. Frequently they are associated to improving data subjects' control on their own data, nonetheless policy comparison may be interesting for data controllers as well. Here is an incomplete list of possible applications of query comparison:

- understanding whether a new consent request, if approved, would strengthen or weaken the previous usage policy; (this may help data subjects in evaluating new consent requests, and may help data controllers in evaluating the correctness of policy modifications);
- understanding whether a novel usage of data is already allowed by previous consent (in that case the data controller needs not to contact the data subject again);
- verifying whether some personal information is protected to different degrees by different data controllers, thereby opening the way to undesired leakage through analogues of record linkage; (see also *policy consistency checking*)
- finding which data can be used in a specified way (cf. the generalized policy scope queries illustrated above);
- determining whether a consent request fits the privacy preferences of a data subject.

*Policy explanation (documentation).* Illustrating the usage policy proposed by a consent request in a dynamic, interactive way, in order to tailor the presentation to the data subject's personal interests and concerns, amounts to answer queries such as "which personal information would be collected?", "who could see my data?", "could anyone see my birth date?" and so on. Such queries, that we will call *documentation queries*, are more general than permission checking, since they do not completely specify the operation, the actor that executes (or may execute) the operation, and the context. Technically speaking, answering these queries amounts to inferring particular kinds of implications.

Another category of queries aims at explaining policy behavior ex post. For instance a data subject may ask "*How could company X get my address? Why can X use it to send me advertisement?*". This kind of query is the policy analogue of inference explanations and of the computation of justifications.

*Policy retrieval.* This is the complement of the *policy scope* query illustrated before. It consists in retrieving the policy that applies to a given piece of data. Policy retrieval may be generalized by returning the policies that applied to the specified data *at a given point in time*. Policy retrieval is an auxiliary task needed in some of the above policy-related queries, for example:

- it is needed for all forms of permission checking, that obviously depend on the applicable policy;
- it is needed to retrieve the previous policy when it has to be compared with a new consent request (cf. policy comparison).

- policy retrieval can be regarded as a form of documentation to answer queries such as "*how can XYZ Ltd. currently use my data?*" and "*how can our company currently use Mr. Smith's data?*"

The association of data with the applicable policies can be encoded and implemented in different ways and the most appropriate choice clearly depends on a number of domain-dependent factors such as the granularity of data, the implementation of business logic, and any constraints originating from legacy systems and standards. Consequently, policy retrieval cannot be handled by SPECIAL's components; its implementation should be part of the SPECIAL-isation of preexisting systems. A suitable API is advocated for interfacing the above reasoning tasks with the policy retrieval facility.

### 5.1.6 Policy language analysis

In this section we evaluate the different policy language options with respect to semantics, expressiveness and complexity.

*Semantics.* A clean declarative semantics is one of the standard desiderata for policy languages. A mathematical account of policy meaning is needed at least (i) as a solid correctness criterion for the implementation of all reasoning tasks, ensuring the mutual coherence of related tasks; (ii) as an unambiguous reference point for all parties, that should interpret a same policy in the same way (which is particularly important as sticky policies are passed along with the data); (iii) as a means for proving formal safety and confidentiality properties of policies.

We have already pointed out that the advocated encoding in RDFS/OWL2 enjoys a natural model-theoretic semantics. A limitation of P3P is that – as a standard – it has not been given any logical semantics. In the technical literature, one can find a non-logical semantics based on three relational tables [75]), and an encoding in description logics [65] (compatible with the encoding of the MCM in OWL2). Only a fragment of ODRL has been given a formal semantics [7]. KAoS and Protune, respectively, inherit the formal semantics of description logics (DL) and rules. Rei, which supports a combination of DL and rules, should specify which of the several alternative formal frameworks for integrating the two families of logics should be applied. Unfortunately, the available papers do not delve into such details, that would be important since complexity and decidability considerations require suitable syntactic restrictions in each of those frameworks; consequently, it is not clear which rule formats are actually allowed.

*Expressiveness (and extensibility).* We have already mentioned that P3P's vocabularies are specifically oriented to privacy and data protection concepts by design, while ODRL was designed with digital rights and licensing in mind. Therefore we should expect P3P to provide particularly good matches to the concepts occurring in SPECIAL's use cases. However, instead of illustrating the pros and cons of the vocabularies natively supported by these two policy languages, here we note simply that P3P and ODRL have extensions mechanisms that can be leveraged to bridge the (potential) gaps in their auxiliary ontologies. For example, P3P's `<EXTENSION>` tag can be inserted virtually everywhere, and in particular it can be exploited to add new data categories, purposes, etc. Consequently, one should not expect to run easily into unresolvable expressiveness issues. Of course, constructs such as the `<EXTENSION>` tag have basically no semantics, since they must provide a fully generic hook to all sorts of extensions. Similar considerations hold for ODRL, whose core vocabulary can be extended by defining suitable *profiles.*[20]

Logical policy languages (based on DL and/or logical rules) can easily represent P3P's and ODRL's vocabularies and, moreover, have several advantages in terms of extensibility. Extensions can be defined axiomatically within the language, and axiomatic definitions give a formal semantics to the corresponding extensions, that can thus be "understood" and processed by inference engines without any ad-hoc integration of the implementation. Axioms can also encode mutual incompatibilities between different terms, thereby enabling sound internal inconsistency checking.

*Reasoning about policies in different languages.* When the policy language is formalized with description logics (DL for short), permission checking is related to instance checking, and policy comparison to subsumption checking (i.e. checking whether a class $C_1$ is contained in class $C_2$). Consistency checking can be reduced to subsumption checking (by verifying whether the given class is contained in the empty class). The computation of justifications has at least the same complexity as subsumption checking (complexity may increase if all minimal justifications are to be computed). In OWL2, instance checking, subsumption checking, and consistency checking are complete for 2-NEXP or its complement. However, if the policies and their auxiliary ontologies for data, purposes, etc. fit within any of the OWL2 profiles (such as OWL2 EL), all of these inferences are tractable. For KAoS (that uses operators not supported by OWL2) complete inference is undecidable, in general, due to *role-value maps.*

In Datalog-based rule languages, permission checking and consistency checking can be mapped on logic program answer computation and the computation of canonical models. These inferences can be computed in polynomial time for any fixed policy (*data complexity*), while over arbitrary policies all these reasoning tasks are EXP-complete (*program* and *combined complexity*) [24]. The computation of justifications can be implemented with so-called *abduction procedures*, that may take exponential time in the worst case, since they may have to produce exponentially many minimal justifications. Policy comparisons such as *does $P_1$ imply $P_2$?* are equivalent to Datalog *query comparisons* that in general are undecidable. The same drawback is inherited by hybrid languages, such as Rei. Moreover, in many examples, Rei makes use of function symbols. Such examples are not Datalog and decidability is not guaranteed (the policy becomes like an arbitrary piece of code).

The above discussion suggests that DL languages are preferable whenever the characteristic expressive capabilities of rule languages are not needed. As of today, they do not seem necessary for SPECIAL's purposes, therefore DL appear particularly appealing. Anyway it should be noted that a nontrivial range of policies can be encoded both as DL classes *and* as Datalog programs.

## 5.2 Regulation Policies

One of SPECIAL's research goals consists in investigating if, how, and to what extent regulations such as the GDPR can be formalized and automatically processed in a way that addresses the needs of SPECIAL's use cases.

[20] `http://w3c.github.io/poe/model/#profile`

The nature of the regulations such as the GDPR is quite different from that of the usage policies dealt with in the previous section. Many articles are about general principles, that are only very loosely and indirectly related to implementations. Other articles are expressed with subjective terms, or terms that admit different interpretations. These features clearly hinder any attempt at fully automated compliance checking.

Still a formalization of the GDPR may enable the development of tools that *assist* data controllers and processors in checking the compliance of their procedures with respect to the regulation.

In this section we discuss the main available approaches at formalizing *vague knowledge* that may be used to turn the GDPR into a partially machine-processable policy.[21] We also address the issues raised by the standard structuring of laws, that heavily exploits exceptions to general norms – a presentation style that cannot be directly handled by the RDFS and OWL2 standards. f

### 5.2.1 Scope of GDPR formalization

Not every aspect of the GDPR is relevant to SPECIAL. For example, the parts related to what member states shall or may do to refine the European regulation and set up supervisory bodies are not in the focus of SPECIAL's goals. Similarly, the goals of the GDPR (cf. Article 1) need not be formalized, as well as any non-normative meta-information about the GDPR itself. Given SPECIAL's focus on the management of informed consent, we will further focus formalization efforts on the norms related to consent, since an extensive formalization covering also the other normative aspects of the GDPR is a major task that would span well beyond the end of the project.

Unfortunately at this stage it is not easy to isolate a small set of relevant articles, due to the abundance of cross references that make the GDPR an almost strongly connected graph[71].

### 5.2.2 Sources of ambiguous, conflicting, and subjective expressions

The legal language sometimes leaves space to different interpretations. We refer to such parts of the regulation as *vague*, as customary in knowledge representation's jargon. Some ambiguity is also inherent in the use of natural language. Since the GDPR is not yet in force, no additional regulations, deliberations, and interpretations are available to disambiguate the "gray areas" of the regulation (with the exception of the recitals associated to the GDPR)[22]. So any attempt at formalizing the GDPR must deal with some sort of vagueness and uncertainty in the interpretation of the regulation.

Some articles involve conditions whose assessment is subjective, in the absence of additional clarifications with binding legal value. For example, Art. 7 states that "*the request for consent shall be presented in a manner which is clearly distinguishable from the other matters, in an intelligible and easily accessible form, using clear and plain language*". None of the underlined part can be assessed objectively based solely on natural language meaning.

[21] Here we are not criticizing the GDPR for being vague. The term *vague* here is used in its technical (logical) acceptation, related to the difficulty of assigning a truth value to some propositions.

[22] Existing judgments concerning the directive and/or existing national legislation may have a role to play in terms of disambiguating the "gray areas" of the regulation.

Further sources of ambiguities result from the need of addressing conflicting requirements. For example, recital (63) recalls that "*A data subject should have the right of access to personal data [...] concerning him or her [...] in order to be aware of, and verify, the lawfulness of the processing.*" But at the same time "*That right should not adversely affect the rights or freedoms of others, including trade secrets [...]*". These opposite requirements raise a question: Should data controllers be obliged to reveal which third parties collected data are transferred to? On the one hand, this information is essential in order to control how the information is being treated; on the other hand, it may reveal trade secrets such as the controller's business relationships. The regulation does not specify which of the two requirements should prevail, and it may well turn out that the conflict should be resolved differently in different cases.

### 5.2.3 Logical modelling (axiomatization)

Logical modelling involves choosing a representation language, which involves both syntactic and semantic choices. In this section we focus on the former and deal with the latter in the next section.

A partial analysis of the GDPR focused on articles 7–17, most directly connected to informed consent and the rights of data subjects, has shown the need for three kinds of axioms that here we call *obligations, constraints*, and *definitions*. We briefly illustrate them in the following.

*Obligations.* These are the statements that describe what data controllers shall do in order to comply with the regulation. An example of obligation, taken from Art. 7 is: "*the controller shall be able to demonstrate that the data subject has consented to processing of his or her personal data*". Deontic logics have been specifically designed to express and reason about obligations and related concepts. Obligation statements may occur within compound statements. For instance, Art. 7 states that the above obligation applies when "*processing is based on consent*", so the obligation is the consequent of an implication whose antecedent formalises the condition that processing is based on consent.

*Constraints.* By "constraint", we mean a logical formula used to restrict or qualify another statement (e.g. an obligation). For instance, by Art. 7, *if* the consent request occurs in a document that concerns also other matters *and* the request is not "suitably formulated" – i.e. it does not meet the distinguishability, intelligibility, accessibility, and clarity requirements reported in the previous section – *then* consent is not legally binding. The above if-then statement is rendered in logic by means of an implication whose consequence is a property of (i.e. qualifies) a consent declaration. Another example of constraint that legally binding consent must satisfy is that it must be "freely given" (Art. 7 point 4).

*Definitions.* The purpose of definitions is defining predicates that succinctly represent complex conditions (i.e. abbreviations). A first example of the usefulness of definitions can be found in the previous paragraph: it is way more readable to define separately what "suitably formulated" means and use this predicate as an atomic expression within the implication that encodes the constraint. This is especially useful if a same abbreviations is used repeatedly in different parts of the regulation.

Definitions are also useful in dealing with the frequent cross-references between articles. Consider Art. 12(1), for example. It obliges data controllers to "*provide any information referred to in Articles 13 and 14 and any communication under Articles 15 to 22 and 34 relating to processing [...]*". Similar conditions occur in points 2 and 3. It is clearly convenient to introduce an abbreviation for all these information categories; replications lead to longer, less readable and more expensive axiomatisations, increase the probability of errors, and make corrections more expensive and error-prone.

Definitions are typically axiomatised with logical equivalences; for instance, the predicate "suitably formulated" would be equivalent to the conditions "distinguishable and intelligible and accessible and clear".

### 5.2.4 Possible semantics

There are different ways of formalizing vague and subjective predicates at the semantic level. Identifying the best approach requires further research, so in this deliverable we simply recall the main approaches.

*Classical/crisp/2-valued semantics.* Under classical model-theoretic semantics, a predicate $p$ with no clear truth value is true in some models of the axioms and false in others, so the axioms imply neither $p$ not $\neg p$. From a philosophical perspective, this presupposes a (possibly unkown) "real world" where the truth value of $p$ is clearly specified. While this seems to be incompatible with subjective information (where there is no agreement on the truth of some sentences), the 2-valued approach is compatible with a "legal" viewpoint: court decisions define a "legal truth" by establishing whether – say – a particular consent request is actually formulated in clear and plain language or not.

*Modal logics.* Modal logics introject the above idea within the language. Models – very roughly speaking – are sets of *possible worlds* (that are classical interpretations). A predicate $p$ is *necessarily true/false* if $p$ is true/false across all possible worlds; $p$ is *possibly true/false* if $p$ is true/false in at least one possible world. A vague or subjective condition would be modelled by a predicate whose truth and falsity are both possible. Note that deontic logics (those that model obligations) are a particular kind of modal logics.

*Three-valued and fuzzy semantics.* These semantics assign more than 2 values to logical statements. The goal is modelling the cases in which it is impossible to establish whether a proposition is true or false (e.g. subjective statements). In 3-valued semantics the truth values are true, false and *undefined* (sometimes represented as 1, 0 and 0.5), while in fuzzy logic there are infinitely many intermediate truth values between false and true, represented by the real interval [0,1]. It may even be possible to give a predicate a specific truth value between 0 and 1, however there is no guidance to the choice of such numbers, and the results may be very confusing.

*Argumentation semantics.* It is based on logical derivations, as opposed to models. Argumentation semantics is analogous to a kind of legal reasoning: it starts by constructing arguments to support $p$ and $\neg p$; then more arguments are constructed to attack the assumptions on which the arguments for $p$ and $\neg p$ are founded, and so on. Eventually, $p$

is concluded only if some argument in favour of $p$ "survives" the attacks, while all the arguments supporting $\neg p$ are successfully attacked.

### 5.2.5 Compliance checking (reasoning)

No matter which semantics is adopted, the formalized GDPR can only be used to compute conditional statements such as "if condition X holds then the GDPR is satisfied", or "compliance implies that conditions X, Y, Z, ... must hold". The actual verification of conditions X, Y, Z etc. cannot be automated because:

1. The GDPR poses general conditions on business processes, e.g. "*The controller shall facilitate the exercise of data subject rights*" (Art. 12(2)). This means – among other things – that there must be a way for the data subject to ask for rectification and deletion of her personal information. This could be done through manual processes, fully automated processes, or intermediate solutions. In practice, no fully automated system can verify whether such a process is in place (the compliance checker would need as an input an impractically detailed and complex description of the business processes of the whole organization). So the requirements for a semi-automatic compliance checker are weaker: With reference to the above example, a formal specification of the GDPR should be able to infer that – in order to be compliant with Art. 12(2) – there must be a process for rectifying personal information and one for deleting it.

2. Subjective and vague conditions cannot be assessed automatically. Again, a reasonable requirement is that the formalized GDPR should be able to infer the subjective conditions that need to be met (e.g. the formalization should entail that consent requests – if any – should be "suitably formulated").

The inferences derivable from the formalized GDPR are context-dependent, e.g. data collection may be justified by other laws (cf. storage obligations applying to telephone and mobility data) so that no consent request is needed in some case (and the related subjective properties need not be verified). More generally, the mutual logical dependencies between different conditions make reasoning on the GDPR more dynamic and useful than a static checklist of conditions for compliance.

The inferences that an automated tool can be reasonably expected to produce call for human intervention, in order to carry out a (reasonably) complete compliance verification. Only humans with suitable knowledge of the organization can assess the existence of the processes required by the GDPR, and only humans with legal competence can assess, in general, whether consent requests and business processes satisfy the subjectively or vaguely formulated constraints imposed by the regulation.

Thus the compliance checking procedure we envision exploits the internal logical dependencies of the regulation to derive minimal, context dependent sets of conditions to be verified; the assessment of those conditions is under the responsibility of human actors, that are in charge of providing *evidence* and possibly *non repudiable declarations* that the required conditions are met. The overall procedure has some analogies with *trust management systems*; see [14] for a brief, informal description of how a reasoning procedure called *abduction* can be used to identify the evidence to be provided and check whether the available, non repudiable evidence is sufficient to prove compliance (in the context of attribute-based access control).

### 5.2.6 Handling exceptions and overriding

Several articles in the GDPR involve what is known as *nonmonotonic reasoning*, that is, inferences based on the *lack* of evidence (as opposed to the *availability* of knowledge and information such as axioms and evidence). In nonmonotonic logics conclusions may be withdrawn when additional information and norms become available. This allows to express and deal with exceptions to general rules in a very natural way. The need for this kind of reasoning is motivated below.

Legislators extensively resort to exceptions in order to refine general directives. In the GDPR we can find many occurrences of this formulation style. Here is a non-exhaustive but representative list of examples:

1. (Art. 9(2)) "*Paragraph 1 shall not apply if one of the following applies: [...]*"
2. (Art. 12(2) "*the controller shall not refuse to act [...] unless the controller demonstrates that [...]*"
3. (Art. 12(3) "*the information shall be provided by electronic means where possible, unless otherwise requested by the data subject*"
4. (Art. 19) "*The controller shall communicate any rectification or erasure of personal data [...] unless this proves impossible or involves disproportionate effort*"
5. (Art. 21(1)) "*The controller shall no longer process the personal data unless the controller demonstrates compelling legitimate grounds for the processing which override the interests, rights and freedoms of the data subject or for the establishment, exercise or defence of legal claims*"
6. (Art. 22(4)) "*Decisions referred to in paragraph 2 shall not be based on special categories of personal data referred to in Article 9(1), unless point (a) or (g) of Article 9(2) applies [...]*"
7. (Art. 34(3)) "*The communication to the data subject referred to in paragraph 1 shall not be required if any of the following conditions are met: [...]*"

Furthermore, there are conditions based on the absence of information (akin to a kind of nonmonotonic reasoning known as *negation as failure*):

7. (Art. 49(1)) "*In the absence of an adequacy decision pursuant to Article 45(3), or of appropriate safeguards pursuant to Article 46, [...] a transfer [...] of personal data to a third country or an international organisation shall take place only on one of the following conditions: [...]*

Last but not least, the conflicts between different requirements (cf. the section *Sources of ambiguous, conflicting, and subjective expressions*) boil down to having one requirement override the other (thus introducing an exception to the latter).

The formulation style based on exceptions and overriding has some practical advantages, e.g. it supports *incremental* and *modular* specifications and updates to the regulation.

Of course, this approach requires the adoption of a nonmonotonic logic. Rule-based languages frequently support nonmonotonic constructs such as negation as failure (see

[4], for a recent example). RDFS and OWL2 are monotonic (i.e. they cannot express exceptions nor overriding) but numerous nonmonotonic extensions has been proposed in the last decades. Unfortunately, in most cases the complexity of nonmonotonic reasoning is significantly more complex than reasoning in the underlying, monotonic Description Logics. For instance, extensions based on Circumscription or the typicality operator make entailment reasoning EXPTIME hard also in the case of OWL2 EL [15, 29]. Moreover, classical approaches such as Circumscription, Autoepistemic Logic, and Default Logic can be unsatisfactory also from a design perspective since, roughly speaking, they tend to repair conflicting overriding rules which instead should be treated as knowledge representation errors.

For this reasons, a new approach, $\mathcal{DL}^N$, has been recently proposed which preserves the tractability of low-complexity Description Logics and generates inconsistent concepts in case of conflicting overriding rules (see [13, 16] for an extensive comparison with competing approaches). Although $\mathcal{DL}^N$ seems to be the most promising proposal, the choice of an appropriate nonmonotonic logic, in formalizing the GDPR, requires further research and lies beyond the scope of this deliverable.

# 6 SPECIALising Company Systems

The initial analysis into potential policy models and languages presented in the previous section provides some pointers as to how usage policies and regulations might be represented in a machine readable manner. In this section, we briefly highlight several considerations and open questions with respect to the intersection between existing company systems and the SPECIAL components. A more detailed analysis including the provision of concrete recommendations are left to D1.7 Policy, transparency and compliance guidelines V2.

## 6.1 Policy Specification and Enforcement

Clearly there is a tight coupling between SPECIAL and existing Line of Business applications in terms of both policy specification and enforcement. Firstly the data that will form part of the consent request and subsequently the usage policy needs to be based on the the type of personal data required by the company in terms of product or service provision, and contextual information relating to the purpose, processing and sharing. Secondly, companies need to ensure that personal data processing and sharing within the orgaisation and by its Information Technology (IT) systems complies with relevant usage policies. In order to better understand the interplay between SPECIAL and existing company systems, in this section we highlight key considerations in terms of associating policies with data, policy enforcement and compliance checking.

## 6.2 Associating Policies with Data

In SPECIAL each consent policy will be given a unique URI. This URI should be used to associate the policy with: the consent obtained from the data subject; the usage policy; relevant data processing and sharing events performed by the company; and possibly even related policies (e.g. if there is a need to maintain a history of policy updates).

Key considerations include how can we associate a URI with personal data stored in existing company systems, according to a variety of data models, possibly at different levels of granularity? How do we ensure that mappings between personal data items and policies are kept up to date? Additionally there is a need for flexibility in terms of policy retrieval. From a navigation perspective it should be possible to navigate from a policy to the data that it governs and also from the data to one or more policies that govern it. Other requirements include the ability to retrieve all policies based on contextual information, such as purpose, type of processing, data subject, to name but a few. Also, where more than one policy governs the data it is necessary to understand the interplay between such policies.

## 6.3 Policy Enforcement

In SPECIAL the sticky policy concept is used to tightly couple data and usage policies. When it comes to the state of the art, sticky policies are usually implemented by using cryptographic means to strongly associate policies with data. However, it is important to highlight that from a practical perspective it is not possible for said policies to be enforced automatically (i.e. it is an honors system whereby data controllers and processors can choose to either obey the policy or not).

Other open questions relate to using technical means to prove that usage policies are being adhered to. For example, if data subjects request that their data is deleted, how do we ensure that this data is in fact deleted and not simply made inactive. Another open research question relates to the inheritance of policies by derivative data. Considering the tight coupling between data and policies, data derivatives (e.g. in the form of aggregated and/or anonymised data) can not be covered by the same sticky policy.

## 6.4 Compliance checking

Irrespective of where the log resides, how much information goes into the log is dependent on what information is needed in order to automatically check compliance with both usage policies and relevant regulations. Given that event logging is a key component of many Line of Business systems, one option would be to re-purpose existing logs so that they can be used to automatically verify compliance of existing systems. However, the suitability of existing logging mechanisms for this purpose requires further analysis. Alternatively it would be possible to have a dedicated log, however the attributes to be recorded would need to align with existing business processes. Ether way, the level of detail required to verify the compliance of existing business processes (that involve personal data) with respect to privacy preferences and legal obligations remains an open question. Likewise further analysis is required in order to determine where is the best place to hook into the existing company systems. Potential considerations here include the data tier, the service tier, or the business tier. Also, considering the complexity of existing business processes that are often only partially automated, a more in depth analysis is required in order to determine how much of the compliance checking can be automated. Key aspects here include the alignment of processing and sharing performed by the systems and what is specified in the consent.

# 7 Conclusions

The overarching goal of the SPECIAL project is to develop an infrastructure that enables companies to comply with consent and transparency obligations specified in the GDPR. Towards this end, the core objective of this deliverable is to identify the high level requirements for the SPECIAL system, to survey the state of the art, and to identify the open challenges that need to be addressed. The analysis presented herein will form a roadmap for the research that will be carried out in workpackage two.

Considering the agile nature of the project this document will be iteratively refined and expanded upon based on further analysis of the uses cases, the development of the policy language and the transparency framework, and subsequently the implementation of the compliance checking. All of which will serve as input into a more detailed requirements deliverable in the form of *D1.7 Policy, transparency and compliance guidelines V2*, which will be delivered at the end of month seventeen.

# Bibliography

[1] *Slaves to Big Data. Or Are We?*, Jun 2013. 9th Annual Conference on Internet, Law & Politics. URL `http://works.bepress.com/mireille_hildebrandt/52`.

[2] R. Accorsi. On the relationship of privacy and secure remote logging in dynamic systems. In *IFIP International Information Security Conference*, 2006.

[3] A. Acquisti, I. Adjerid, and L. Brandimarte. Gone in 15 seconds: The limits of privacy transparency and control. *IEEE Security & Privacy*, 11(4):72–74, 2013.

[4] M. Alviano, F. Calimeri, C. Dodaro, D. Fuscà, N. Leone, S. Perri, F. Ricca, P. Veltri, and J. Zangari. The ASP system DLV2. In M. Balduccini and T. Janhunen, editors, *Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Espoo, Finland, July 3-6, 2017, Proceedings*, volume 10377 of *Lecture Notes in Computer Science*, pages 215–221. Springer, 2017. ISBN 978-3-319-61659-9. doi: 10.1007/978-3-319-61660-5_19. URL `https://doi.org/10.1007/978-3-319-61660-5_19`.

[5] F. Baader, D. L. McGuiness, D. Nardi, and P. Patel-Schneider. *The Description Logic Handbook: Theory, implementation and applications.* Cambridge University Press, 2003.

[6] F. Baader, S. Brandt, and C. Lutz. Pushing the EL envelope. In L. P. Kaelbling and A. Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, July 30 - August 5, 2005*, pages 364–369. Professional Book Center, 2005. ISBN 0938075934. URL `http://ijcai.org/Proceedings/05/Papers/0372.pdf`.

[7] N. Bassiliades, G. Gottlob, F. Sadri, A. Paschke, and D. Roman, editors. *Rule Technologies: Foundations, Tools, and Applications - 9th International Symposium, RuleML 2015, Berlin, Germany, August 2-5, 2015, Proceedings*, volume 9202 of *Lecture Notes in Computer Science*, 2015. Springer. ISBN 978-3-319-21541-9. doi: 10.1007/978-3-319-21542-6. URL `https://doi.org/10.1007/978-3-319-21542-6`.

[8] M. Bellare and B. Yee. Forward integrity for secure audit logs. Technical report, Technical report, Computer Science and Engineering Department, University of California at San Diego, 1997.

[9] E. Benda, H. Simon, K. Hesse, D. Katzenstein, G. Niemeyer, H. Heuï£¡ner, and J. F. Henschel. Bverfge 65, 1. 65:1–71, 1983.

[10] E. Bertino, P. A. Bonatti, E. Ferrari, and M. L. Sapino. Temporal authorization bases: From specification to integration. *Journal of Computer Security*, 8(4):309–353, 2000. URL `http://content.iospress.com/articles/journal-of-computer-security/jcs140`.

[11] P. Bonatti and D. Olmedilla. Rule-based policy representation and reasoning for the semantic web. *Reasoning Web*, pages 240–268, 2007.

[12] P. Bonatti, S. De Capitani di Vimercati, and P. Samarati. An algebra for composing access control policies. *ACM Transactions on Information and System Security (TISSEC)*, 5(1), 2002.

[13] P. A. Bonatti and L. Sauro. On the logical properties of the nonmonotonic description logic dl$^{\mathrm{n}}$. *Artif. Intell.*, 248:85–111, 2017. doi: 10.1016/j.artint.2017.04.001. URL `https://doi.org/10.1016/j.artint.2017.04.001`.

[14] P. A. Bonatti, J. L. D. Coi, D. Olmedilla, and L. Sauro. A rule-based trust negotiation system. *IEEE Trans. Knowl. Data Eng.*, 22(11):1507–1520, 2010. doi: 10.1109/TKDE.2010.83. URL `https://doi.org/10.1109/TKDE.2010.83`.

[15] P. A. Bonatti, M. Faella, and L. Sauro. Defeasible inclusions in low-complexity DLs. *J. Artif. Intell. Res. (JAIR)*, 42:719–764, 2011.

[16] P. A. Bonatti, M. Faella, I. M. Petrova, and L. Sauro. A new semantics for overriding in description logics. *Artif. Intell.*, 222:1–48, 2015. doi: 10.1016/j.artint.2014.12.010. URL `https://doi.org/10.1016/j.artint.2014.12.010`.

[17] F. Z. Borgesius. Informed consent: We can do better to defend privacy. *IEEE Security & Privacy*, 13(2):103–107, 2015.

[18] J. Bouckaert and H. Degryse. Opt in versus opt out: A free-entry analysis of privacy policies. In *WEIS*, 2006. URL `https://pdfs.semanticscholar.org/f86d/af014be6a8581adcd878bd10cfec9ceae82a.pdf;http://dblp.org/rec/conf/weis/BouckaertD06`.

[19] J. M. Bradshaw. *Software agents*. MIT press, 1997.

[20] I. Budin-Ljøsne, H. J. Teare, J. Kaye, S. Beck, H. B. Bentzen, L. Caenazzo, C. Collett, F. DâĂŹAbramo, H. Felzmann, T. Finlay, et al. Dynamic consent: a potential solution to some of the challenges of modern biomedical research. *BMC medical ethics*, 18(1):4, 2017.

[21] C. Cachin, K. Haralambiev, H. Hsiao, and A. Sorniotti. Policy-based secure deletion. In *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13*, 2013.

[22] E. Commission. Proposal for a regulation of the european parliament and of the council concerning the respect for private life and the protection of personal data in electronic communications and repealing directive 2002/58/ec (regulation on privacy and electronic communications), 01 2017. URL `http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52017PC0010&rid=1`.

[23] L. F. Cranor. *Web privacy with P3P - the platform for privacy preferences.* O'Reilly, 2002. ISBN 978-0-596-00371-5. URL `http://www.oreilly.de/catalog/webprivp3p/index.html`.

[24] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Comput. Surv.*, 33(3):374–425, 2001.

[25] O. de Moor, G. Gottlob, T. Furche, and A. J. Sellers, editors. *Datalog Reloaded - First International Workshop, Datalog 2010, Oxford, UK, March 16-19, 2010. Revised Selected Papers*, volume 6702 of *Lecture Notes in Computer Science*, 2011. Springer. ISBN 978-3-642-24205-2. doi: 10.1007/978-3-642-24206-9. URL `https://doi.org/10.1007/978-3-642-24206-9`.

[26] European Parliament and Council Directive. Directive 2002/58/ec of the european parliament and of the council: concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications). Official Journal of the European Communities, 2002.

[27] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the eighth symposium on usable privacy and security*, page 3. ACM, 2012.

[28] J. D. Fernández Garcia, J. Umbrich, M. Knuth, and A. Polleres. Evaluating query and storage strategies for RDF archives. In *12th International Conference on Semantic Systems (SEMANTICS)*, ACM International Conference Proceedings Series, 2016.

[29] L. Giordano, V. Gliozzi, N. Olivetti, and G. L. Pozzato. Reasoning about typicality in low complexity dls: The logics el$^{\perp}$t$_{\mathrm{min}}$ and dl-lite$_{\mathrm{c}}$ t$_{\mathrm{min}}$. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 894–899, 2011.

[30] S. Gürses and J. M. del Alamo. Privacy engineering: Shaping an emerging field of research and practice. *IEEE Security & Privacy*, 14(2):40–46, 2016.

[31] S. Guth and R. Iannella. ODRL V2.0 – core model. `http://odrl.net/2.0/WD-ODRL-Vocab.html`. Accessed: 2010-08-05.

[32] M. Hansen. Data protection by design and by default à la european general data protection regulation. In *Privacy and Identity Management. Facing up to Next Steps*, pages 27–38. Springer, 2016.

[33] H. Hedbom, T. Pulls, P. Hjärtquist, and A. Lavén. Adding secure transparency logging to the prime core. In *IFIP PrimeLife International Summer School on Privacy and Identity Management for Life*, 2009.

[34] M. Hildebrandt. The new imbroglio. living with machine algorithms. In L. Janssens, editor, *The Art of Ethics in the Information Society. Mind you*, pages 55–60. 2016. URL `https://works.bepress.com/mireille_hildebrandt/75/download/`.

[35] J. E. Holt. Logcrypt: forward security and public verification for secure audit logs. In *Proceedings of the 2006 Australasian workshops on Grid computing and e-research-Volume 54*, 2006.

[36] L.-E. Holtz, H. Zwingelberg, and M. Hansen. Privacy policy icons. *Privacy and Identity Management for Life*, pages 279–285, 2011.

[37] A. Hope-Bailie and S. Thomas. Interledger: Creating a standard for payments. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 2016.

[38] R. Iannella, M. Steidl, M. McRoberts, S. Myles, J. Birmingham, and V. Rodríguez-Doncel. ODRL Vocabulary & Expression. W3C Working Draft, available at `https://www.w3.org/TR/2017/WD-odrl-vocab-20170223/`, W3C, 2017.

[39] L. Kagal, T. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*, pages 63–74. IEEE, 2003.

[40] L. Kagal, T. W. Finin, and A. Joshi. A policy language for a pervasive computing environment. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 63–, Lake Como, Italy, June 2003. IEEE Computer Society. ISBN 0-7695-1933-4.

[41] J. Kaye, E. A. Whitley, D. Lund, M. Morrison, H. Teare, and K. Melham. Dynamic consent: a patient interface for twenty-first century research networks. *European Journal of Human Genetics*, 23(2):141, 2015.

[42] S. Kirrane, A. Mileo, and S. Decker. Access control and the resource description framework: A survey. *Semantic Web*, 8(2):311–352, 2017.

[43] S. Kremer, O. Markowitch, and J. Zhou. An intensive survey of fair non-repudiation protocols. *Computer communications*, 25(17), 2002.

[44] L. T. Ly, F. M. Maggi, M. Montali, S. Rinderle-Ma, and W. M. van der Aalst. Compliance monitoring in business processes: Functionalities, application, and tool-support. *Information systems*, 54:209–234, 2015.

[45] D. Ma and G. Tsudik. A new approach to secure logging. *ACM Transactions on Storage (TOS)*, 5(1), 2009.

[46] V. Mayer-Schönberger and K. Cukier. *Big data: a revolution that will transform how we live, work, and think.* Houghton Mifflin Harcourt, 2013.

[47] A. M. McDonald and L. F. Cranor. The cost of reading privacy policies. *ISJLP*, 4:543, 2008. URL `http://heinonline.org/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/isjlpsoc4&section=27`.

[48] L. Moreau, J. M. Bradshaw, M. R. Breedy, L. Bunch, P. J. Hayes, M. Johnson, S. Kulkarni, J. Lott, N. Suri, and A. Uszok. Behavioural specification of grid services with the KAoS policy language. In *CCGRID*, pages 816–823, 2005.

[49] H. F. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1), 2004. URL `http://ssrn.com/abstract=534622`.

[50] E. Parducci, H. Lockhart, and E. Rissanen. Xacml v3. 0 privacy policy profile version 1.0. *Policy*, pages 1–11, 2010.

[51] A. . D. P. W. Party. *Article 29 Data Protection Working Party (2004), Opinion 10/2004 on More Harmonised Information Provisions: Adopted on 25th November 2004. 11987/04/EN.* 2004. URL `http://ec.europa.eu/justice/policies/privacy/docs/wpdocs/2004/wp100_en.pdf`.

[52] A. . W. Party. Opinion 15/2011 on the definition of consent. *Opinions of the Article 29 WP*, 187, Jul 2011. URL `http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2011/wp187_en.pdf`.

[53] A. . W. Party. Opinion 03/2016 on the evaluation and review of the eprivacy directive (2002/58/ec). *Opinions of the Article 29 WP*, 240, Jul 2016. URL `http://ec.europa.eu/justice/data-protection/article-29/documentation/opinion-recommendation/files/2016/wp240_en.pdf`.

[54] R. Peeters, T. Pulls, and K. Wouters. Enhancing transparency with distributed privacy-preserving logging. In *ISSE 2013 Securing Electronic Business Processes*. Springer, 2013.

[55] T. Pulls, R. Peeters, and K. Wouters. Distributed privacy-preserving transparency logging. In *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*, 2013.

[56] M. Rinne, E. Blomqvist, R. Keskisärkkä, and E. Nuutila. Event processing in rdf. In *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188*, 2013.

[57] S. Sackmann, J. Strüker, and R. Accorsi. Personalization in privacy-aware highly dynamic systems. *Communications of the ACM*, 49(9), 2006.

[58] J. Samuel and B. Zhang. Requestpolicy: Increasing web browsing privacy through control of cross-site requests. In *Privacy enhancing technologies*, pages 128–142. Springer, 2009.

[59] B. Schneier and J. Kelsey. Cryptographic support for secure logs on untrusted machines. In *USENIX Security*, 1998.

[60] O. Seneviratne and L. Kagal. Enabling privacy through transparency. In *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*, 2014.

[61] D. J. Solove. Privacy self-management and the consent dilemma. 2012.

[62] M. Staten and F. H. Cate. The impact of opt-in privacy rules on retail credit markets: A case study of mbna. *Duke Law Journal*, 52:745, 2003. URL `https://ssrn.com/abstract=932958`.

[63] K. S. Steinsbekk, B. K. Myskja, and B. Solberg. Broad consent versus dynamic consent in biobank research: Is passive participation an ethical problem? *European Journal of Human Genetics*, 21(9):897, 2013.

[64] S. Steyskal and A. Polleres. Towards formal semantics for odrl policies. In *International Symposium on Rules and Rule Markup Languages for the Semantic Web*, pages 360–375. Springer, 2015.

[65] B. Suntisrivaraporn and A. Khurat. Formalizing and reasoning with P3P policies using a semantic web ontology. In C. Sombattheera, A. Agarwal, S. K. Udgata, and K. Lavangnananda, editors, *Multi-disciplinary Trends in Artificial Intelligence - 5th International Workshop, MIWAI 2011, Hyderabad, India, December 7-9, 2011. Proceedings*, volume 7080 of *Lecture Notes in Computer Science*, pages 87–99. Springer, 2011. ISBN 978-3-642-25724-7. doi: 10.1007/978-3-642-25725-4_8. URL `https://doi.org/10.1007/978-3-642-25725-4_8`.

[66] A. Uszok, J. M. Bradshaw, R. Jeffers, N. Suri, P. J. Hayes, M. R. Breedy, L. Bunch, M. Johnson, S. Kulkarni, and J. Lott. KAoS policy and domain services: Towards a description-logic approach to policy representation, deconfliction, and enforcement. In *4th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY)*, pages 93–96, Lake Como, Italy, June 2003. IEEE Computer Society. ISBN 0-7695-1933-4.

[67] W. M. Van der Aalst. Process mining. In *Process Mining*, pages 95–123. Springer, 2011.

[68] S. Villata and F. Gandon. Licenses compatibility and composition in the web of data. In *Proceedings of the Third International Conference on Consuming Linked Data-Volume 905*, pages 124–135. CEUR-WS. org, 2012.

[69] T. Waizenegger. Secure cryptographic deletion in the swift object store. In *Datenbanksysteme für Business, Technologie und Web (BTW)*, 2017.

[70] T. Waizenegger, F. Wagner, and C. Mega. SDOS: using trusted platform modules for secure cryptographic deletion in the swift object store. In *Proceedings of the 20th International Conference on Extending Database Technology, EDBT*, 2017.

[71] *Modelling the General Data Protection Regulation*, volume Conference Proceedings IRIS 2017, Feb 2017. Weblaw.ch. URL `https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnxzYWJyaW5ha2lycmFuZXxneDozZjNjOWVjMWJlNGQ1Y2Zl`.

[72] D. J. Weitzner, H. Abelson, T. Berners-Lee, J. Feigenbaum, J. Hendler, and G. J. Sussman. Information accountability. *Communications of the ACM*, 51(6), 2008.

[73] R. Wenning. Quo vadis datenschutz? *Berliner Datenschutzrunde*, Aug 2014. URL `https://berliner-datenschutzrunde.de/?q=node/65`.

[74] K. Wouters, K. Simoens, D. Lathouwers, and B. Preneel. Secure and privacy-friendly logging for egovernment services. In *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*, 2008.

[75] T. Yu, N. Li, and A. I. Antón. A formal semantics for P3P. In V. Atluri, editor, *Proceedings of the 1st ACM Workshop On Secure Web Services, SWS 2004, Fairfax, VA, USA, October 29, 2004*, pages 1–8. ACM, 2004. ISBN 1-58113-973-X. doi: 10.1145/1111348.1111349. URL `http://doi.acm.org/10.1145/1111348.1111349`.

[76] G. Zyskind, O. Nathan, et al. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, 2015.