# SPECIAL

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

## Deliverable D1.7

**Policy, transparency and compliance guidelines V2**

Document version: V1.0

# SPECIAL DELIVERABLE

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

Ms Jessica Michel t: +33 4 92 38 50 89 f: +33 4 92 38 78 22 e: jessica.michel@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: | M01-M17 |
| Deliverable number: | D1.7 |
| Deliverable title | Policy, transparency and compliance guidelines V2 |
| Contractual Date of Delivery: | 31-05-2018 |
| Actual Date of Delivery: | 31-07-2018 |
| Editor (s): | S. Kirrane, P. Bonati |
| Author (s): | P. Bonatti, S. Kirrane, R. Wenning, A. Corazza, C. Galdi, A. Apicella, W. Dullaert, P. Raschke, L. Sauro |
| Reviewer (s): | P. Raschke, U. Milošević |
| Participant(s): | CeRICT, WU, ERCIM, TF, TUB |
| Work package no.: | 1 |
| Work package title: | Use Cases and Requirements |
| Work package leader: | CeRICT |
| Distribution: | PU |
| Version/Revision: | 1.0 |
| Draft/Final: | Final |
| Total number of pages (including cover): | 102 |

# 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.

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 Use Case 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 use case 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.

**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.

[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/

**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 it 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 [108].

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 [155] 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 [155] 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 [25] and Schneier and Kelsey [140] 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 [25] 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 [140] also uses MACs. However, the log is composed of hash chains as opposed to cipher block chains. Whereas Holt [82] proposes an alternative that combines public key cryptography with hash chains. These approaches are further enhanced by Ma and Tsudik [109] 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. [137] 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 [25] and Holt [82] evaluate the performance and scalability of the proposed logging and verification algorithms, while Ma and Tsudik [109] 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 [3] 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. [164] 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. [80], Peeters et al. [132], Pulls et al. [133] 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. [132], Pulls et al. [133], 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. [132], Pulls et al. [133] 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 [140] 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. [162] 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 [141] 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. [174]. 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 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

| | Local Ledger | Global Ledger + TTP | Global Ledger + P2P |
|---|---|---|---|
| Completeness | - | - | - |
| Confidentiality | MAC [25, 82, 137, 140],<br>FssAgg [109],<br>PKI [82, 109] | MAC [3, 80, 132, 133],<br>PKI [164],<br>unlinkability [80, 132, 133] | MAC [140],<br>PKI[141],<br>compound identities [140, 174] |
| Correctness | - | - | - |
| Immutability | cipher chains [25],<br>hash chains [82, 140] | hash chains [82, 140] | network of peers [141, 162]<br>blockchain [174] |
| Integrity | forward integrity [25, 82, 109, 137, 140]<br>MAC security proof [25] | forward integrity [3, 80, 132, 133] | forward integrity [140] |
| Interoperability | - | - | - |
| Non-repudiation | - | - | - |
| Rectification & Erasure | - | - | - |
| Traceability | - | event trails [164] | - |

Table 1: Candidate architectures and ledger functionality gap analysis.

| | Local Ledger | Global Ledger + TTP | Global Ledger + P2P |
|---|---|---|---|
| Availability | - | - | - |
| Performance | logging & verification [25, 82],<br>signature generation & verification [109] | logging [132, 133],<br>throughput [132, 133] | - |
| Scalability | encrypting records [82, 109] | - | - |
| Storage | key & signature [109] | resource restricted devices [3] | - |

Table 2: Candidate architectures and ledger robustness gap analysis.

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 [140] 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 [69] and references therein. Additionally there exists a number of general event vocabularies such as the *Event*[9] ontology and the *LODE*[10] ontology [136] 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 [39]). 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 [84] 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. A range of available fair exchange protocols is illustrated in Section 4. 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, see Section 4 for more details.

### 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 of log entries in distributed environments 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. [48, 158, 159]. 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.

**Software architecture** The above sections highlight a number of nontrivial research issues that call for a pragmatic strategy to guarantee the timeliness of the first software release. Summarizing, because a malicious data controller might simply chose to not log specific events, we rely on the goodwill of data providers to want to provide this transparency. The use of fair-exchange protocols to get stronger guarantees for multi-party transactions can be investigated at a second stage, as such protocols (that in SPECIAL reside at the application level) are largely independent from the underlying big data architecture. In this context a lot of the cryptographic guarantees given by the ledger infrastructure presented in the literature is only of small value. Moreover, adopting popular ledger technology such as blockchains would be risky due to its functional conflicts with some of the GDPR's requirements, such as the right to be forgotten.

Therefore, for the first iteration of the special platform, it is advisable to focus on another major aspect, which is pervasive in SPECIAL's approach, that is, the RDF representation of events and the scalability of the compliance checking.

For the log infrastructure and faceted browsing, well performing off-the-shelve components such as Apache Kafka have been selected. More information is available in D3.2. Cryptographic features for the ledger infrastructure and fair-exchange protocols will be investigated at a later stage.

# 4 Fair exchange methods

As discussed in the previous section, fair exchange protocols constitute a promising way of improving the correctness and completeness of transparency ledgers. In particular, they provide additional guarantees to the information transfers that involve two or more non-colluded parties. This section reviews the main available fair exchange protocols and their properties. In a first subsection we recall the basic terminology needed for classifying fair exchange approaches. In a second subsection we survey the available methods.

## 4.1 Basic terminology

Informally speaking, in a fair exchange, two parties $A$ and $B$ want to exchange two objects $O_A$ and $O_B$ such that $A$ receives $O_B$ if and only if $B$ receives $O_A$. There exist in the literature extensions to multiparty fair exchange, in which multiple parties interact to exchange digital objects, with different security definitions. Fair exchange protocols can be classified according to different parameters. In this section we remark the

### 4.1.1 Communication Channel Classification

In the literature different assumptions have been made on the guarantees provided by the communication channels used by the entities. It is possible to classify channels as follows [99].

- *Unreliable Channels.* An unreliable channel provides no guarantee on data transmission. Data might be corrupted, or lost. No authentication is provided. Furthermore, data may be arbitrarily delayed by an adversary.
- *Resilient Channels.* A resilient channel always delivers messages correctly after a finite, but unknown, period of time. This type of channels typically model asynchronous networks. Notice that, on resilient channels, the order of messages might be altered.
- *Operational Channels.* An operational channel always delivers messages correctly within a finite and known *a priori* period of time.

### 4.1.2 Trusted Third Party Involvement

Most of the solutions to the fair exchange problem require the presence of a Trusted Third Party, or TTP for short. Fair Exchange protocols can be essentially classified as on-line/in-line and off-line (or optimistic) protocols. In the first class, a Trusted Third Party (TTP for short) has a central role in the protocol in the sense that *each* exchange involves the TTP. In the optimistic protocols, the TTP comes into play only if at least one of the players misbehaves while, in the other cases, the users run the protocols by themselves.

It is clear that the latter class of protocols has a number of advantages with respect to the former one. In-line protocols are usually simpler than optimistic ones but have the drawback that the TTP could become a bottleneck for the system. On the other hand, optimistic protocols typically do not allow simple *centralized* accountability. Indeed, whenever the players behave properly, the TTP does not even know a pair of players exchanged messages. Since TTPs are typically run by service providers, the lack of accountability might become an issue.

It is possible to classify different protocols based on the extent to which the TTP participates to the protocol.

- *Inline TTP*. In *inline* protocols the TTP is involved in every *message transmission.* In other words, every message is either sent to the TTP or is sent by the TTP to another party.

- *Online TTP.* In an *online TTP* protocol, the TTP is involved in every *protocol execution*, i.e., in every run of the protocol there exists at least one message that is either sent to the TTP, but there might exist messages exchanged directly by the other parties.

- *Offline TTP.* In a protocol, the TTP is said to be offline if the TTP participates in the protocol only in case one of the parties misbehaves. This type of protocols are also called *optimistic*.

- *Transparent TTP.* An optimistic protocol in which the TTP produces evidences that are indistinguishable from the ones produced by the parties in a correct execution of the protocol is said to be *transparent*.

### 4.1.3 Protocol Properties

Because of its generality, the fair exchange problem can be instantiated in different contexts for solving different problems. Just to name a few, certified email, e-commerce of digital goods or contract signing are different instantiations of the this problem. Clearly, each instantiation context might pose requirements that are non-existing or irrelevant for other contexts. Consider for example the problems of digital contract signing and certified email. A basic requirement in the former is that both/all parties know in advance the content of the message to be signed, the contract. This requirement is non-existing in the latter context as, clearly, the recipient of an email (typically) does not know the content of the messages she is going to receive.

Following we list (an almost comprehensive set of) properties that might appear as requirements in different instantiations of the fair exchange problem. Properties are stated for the case of Alice-Bob two-party protocols but they can be extended to the multiparty case.

- *Fairness:* The protocol should be *fair* in the sense that neither Alice nor Bob should be able to obtain an advantage on the other player. In other words, either Bob receives the message $O_A$ and Alice the corresponding $O_B$ or none of them receives useful information.

- *Non-repudiation of origin:* Alice should not be able to deny the fact that she sent the object $O_A$. This means that Bob, at the end of the protocol, should have enough information to prove the sender's identity for the object $O_A$.

- *Non-repudiation of receipt:* Bob should not be able to deny the fact that he received an object. Alice should get a receipt for the messages she sends that can be used as a proof in a court of law.

- *Authenticity:* The players should be guaranteed of their reciprocal identity.

- *Integrity:* The parties should not be able to corrupt the messages and/or their receipts, e.g., Alice should not be able to obtain a receipt for a message different from the one received by Bob and *vice versa*.

- *Confidentiality:* The protocol should be such that only Alice and Bob will be able to read the content of the message. Notice that this property also holds for the

TTP in the sense that it should not be able to infer useful information about the message.

- *Timeliness:* The protocol terminates within a finite and known *a priori* time.
- *Temporal Authentication:* Some applications, e.g., patent submission, require the possibility to verify the time at which the message was sent. The timestamp should be observable by the players and should be ensured by a trusted authority.
- *Sending Receipt:* Some fair exchange protocols might involve human interaction, e.g., certified email ones. It might be desirable that the sender obtains an evidence of the fact that he has *started* the process of exchanging messages. Notice that this receipt may not contain any information generated by the recipient, e.g., it is produced by a third authority.

Almost all of the above properties are desirable in the context of SPECIAL. This is pretty obvious for all but the last two properties. Concerning temporal authentication, the time of a data exchange is needed to determine the applicable policy (that may change over time). Receipts of the start of a data exchange, instead, currently seem less relevant to privacy concerns, but they may be useful to prove that contractual obligations with a business partner have been fullfilled.

#### 4.1.4 Different notions of fairness

In 1980, Even and Yacobi in [66] proved that no deterministic fair exchange protocol exists, in which there is no participation of a third party. In other words, this means that it is impossible to ensure a *perfect* fairness without using a TTP.

Historically, the first solutions to the fair exchange problems appeared at the beginning of the 80s. Since their introduction, a number of different definitions for the concept of *fairness* have been proposed.

**Computational fairness.** First definitions of fairness in [37, 65, 67] considered the problem from a computational perspective. For example, in [67] the authors proposed a first notion of fairness in the context of contract signing, which they called *Concurrency*, stated as follows: *if one party X executes the protocol properly, then his counterpart Y cannot obtain X's signature to the contract without yielding his own signature to it.* Despite this definition, in [67] the authors provide a protocol for a weaker notion of fairness, which they call *Approximate Concurrency*, stated as follows: *If one party X executes the protocol properly then, with very high probability, at each stage during the execution, X can compute his counterpart's signature to the contract using approximately the same amount of work used by Y to compute X's signature to the contract.* The approximate fairness of the protocol is proved under standard cryptographic assumptions and under the *equal resource assumption*, that is, *The computational capabilities of both parties are (approximately) the same.*

**Probabilistic fairness.** The equal resource assumption has been immediately recognized a strong assumption. A way to remove it from the definition of fairness is the usage of probabilistic arguments as done in [27, 135]. In [27] the authors define fairness as follows: *A contract signing protocol is* $(v, \epsilon)$*-fair for A if the following holds: for any contract C, when A follows the protocol properly, at any step of the protocol in which the probability that B is privileged is greater than v, the conditional probability that "A is not privileged" given that "B is privileged" is at most* $\epsilon$. *A protocol is* $(v, \epsilon)$*-fair if it is* $(v, \epsilon)$*-fair for both A and B.*

**General definition of fairness.** Computational or probabilistic fairness typically imply that protocols in which parties *gradually disclose* the information. This type of protocols do require a lot of resources both in terms of time and communication. Currently, the widely (e.g., [9, 11, 13, 14]) accepted definition of fairness can be stated as follows [99]: *at the end of the fair exchange protocol run, either all involved parties obtain their expected information or none of them receives anything.*

**Weaker notion of fairness.** A weaker notion of fairness has been presented in [149]. In this paper the author defines *weak fair exchange* as the property that *guarantees to a participant that she receives either the agreed upon goods or evidence that the other principal has access to the goods she sent.*

## 4.2 Available approaches

### 4.2.1 Protocols without TTP

A number of solutions that do not use TTP have been presented in the past.

The first approach to fair exchange without TTP is the gradual release of secrets. The underlying idea is to exchange the objects in a way that, at each round, both players have approximately received the same percentage of the desired data or, more technically, have approximately the same advantage. Examples of this approach are the ones in [151, 152], in which the author describes a method for exchanging "arbitrary fractions" of bits.

Another solution that does not use TTP has been presented in [149]. In this paper the author first defines a weak notion of fairness. He then shows how to achieve it by using a primitive called *Weakly Secret Bit Commitment*. Roughly speaking, this primitive is a variant of bit commitment schemes in which the hiding property can be broken within a known-a-priori amount of time. This protocol cannot be considered a fair exchange one since it allows the possibility that one player gains advantage w.r.t. the other player. Indeed, one player might receive the commitment to the object and decide not to send her commitment back. She then pays this advantage by using the computational power to break the bit commitment and extract the committed object. Clearly, this type of behaviour is reasonable only if the *value* assigned by the party to the object is greater than the amount the party has to pay to gain the advantage. For this reason this type of protocols are called *rational exchange protocols.*

In [114] the authors present the first fair exchange protocol that also guarantees the non-repudiability of messages. Non repudiation is guaranteed by using standard signature schemes by embedding identities of sender and receiver, along with a timestamp in each exchanged message. The goal of this protocol is to avoid the intervention of a TTP at the price of accepting the probabilistic version of fairness. The protocol has to be parameterised on the basis of the most powerful entity's computing power.

### 4.2.2 Protocols with inline TTP

Inline protocols assume that each message is sent to/received by the TTP. This means that the TTP easily becomes a bottleneck because of their computation/space/communication involvement. On the other hand, (centralized or distributed) inline TTPs greatly simplify service accountability.

Examples of inline protocols can be found in [19, 168]. Sometimes these type of protocols may require multiple TTPs, each providing a different service in a trusted manner. For example, the authors in [50] present a protocol that requires an inline *trusted non-repudiation* server and an inline *trusted timestamping* server in order to provide a non-repudiation fair exchange protocol.

### 4.2.3 Protocols with online TTP

A third class of protocols uses TTPs in an on-line fashion. Specifically, in every execution of the protocol, the TTP receives at least one message from at least one player. This type of involvement of the TTP aims at reducing this player overhead while keeping the possibility of easy accountability. An examples of this class of protocols is [2], a protocol specifically designed for certified email systems. In this case the TTP is used to guarantee the fairness of the transaction. The message sender sends to the receiving party the email message encrypted using a random key, along with the random key encrypted using the public key of the TTP. The receiving party sends the encrypted key to the TTP who sends back the random key to the receiving party and a receipt to the message sender.

In [52] the authors present protocols for purchasing digital goods. The TTP, called the Netbill server, is used to guarantee fairness. The basic protocol requires the client to identify herself for successfully executing the purchase, e.g., in case of software licences. Authors also explicitly consider the possibility of replacing identities with pseudonyms in order to guarantee clients' anonymity while preserving linkability of transactions on the merchant side.

In [57] the authors proposed a certified e-mail protocol. Their major contribution is that the protocol was designed to be integrated into an existing e-mail protocol.

Online TTPs have been used in different ways. For example, in [135], at each round the TTP broadcasts messages containing public keys and one private key corresponding to one public key broadcasted in the previous round. In this case the same TTP might be used for multiple instances of the protocol in parallel. Furthermore, the TTP does not need to store any data about transactions.

Sometimes the TTP is used to properly manage encryption keys. For example in [167, 169] the TTP receives from the parties the keys they have used to encrypt the messages they exchange. In [167] the TTP is responsible for pushing the proper key to the appropriate party. In this case she is thus responsible for the delivery of the messages. Instead, in [169], the TTP simply publishes the keys in a publicly available location and the responsibility of its correct delivery is delegated to the party and it assumes the reliability of the channel.

The idea reducing the trustworthiness of the third party has been proposed in [71] for fair-exchange and used, for example in the context of certified email in [14]. In [71] a third party is *semi-trusted* if, essentially, she is honest-but-curious. She does not collude with any party but she may try to gain information from the messages exchanged during the execution of the protocol.

### 4.2.4 Protocols with offline TTP

A TTP is said to be offline if she participates in the protocol only in the case in which one of the party aborts or misbehaves. From this perspective, offline TTPs are *optimistic* in the sense that they behave like the other parties will behave in the best possible way by following the protocol. Hence, protocols in this class are also called *optimistic*. Since TTP involvement is not required at each protocol execution, optimistic protocols are typically described by defining the algorithm for the honest parties along with the recovery procedures that the parties and the TTP have to execute in case of failures/timeouts.

Among the first optimistic protocols for contract signing we mention [27]. In this case, the TTP, called the *judge* does nothing until invoked by one of the party. Intuitively, the parties exchange signed messages stating that *the contract $C$ will be valid with probability $p$ at time $D$*, where the value of $p$ is randomly incremented at each round. The protocol terminates either when $p$ is equal to 1, at the deadline $D$ or in case one of the party stops the protocol. If the protocol does not terminate with $p = 1$, one of the party may invoke the judge that randomly selects a number in $p_C$ in $[0, 1]$. The judge decides that the contract $C$ is binding for both parties is $p \geq p_C$. Clearly, the trustworthiness of the judge in this protocol is crucial as its role is to decide on the validity of a contract based on a random number she selects. Given the above description, it is clear that the TTP does not *act* on the messages exchanged by the parties but takes a decision based on their content. For this reason, sometimes the protocol in [27] is considered as *not having* a TTP.

The idea of *optimistic* protocols introduced in [120] in the context of certified email and in the setting efficient fair exchange protocols for *generic items* in [9, 11, 22].

The degree of fairness guaranteed by the protocol depends on certain properties of the items to be exchanged: if the third party can undo a transfer of an item (so called revocability) or if it is able to produce a replacement for it (so called generatability) the protocol achieves true fairness. The idea of revocable items was given in [9]. However, they only proposed protocols that require a synchronous setting, i.e. operational channels. Only in 2003, [157] designed a new protocol that works on resilient channels. The new protocol was motivated by the fact that revocability is easier to implement for most popular payment schemes than generatability. For (strong) fairness to be possible [9], at least one of the exchanged items must either be generatable or revocable by the TTP. The TTP might replace/generate one protocol message either because she resends the same message she has received earlier, e.g., an encryption key, or by issuing an affidavit by herself, e.g., a message receipt. Clearly the TTP can be transparent only in the first case.

In [9] the authors present protocols that can guarantee strong fairness in presence of operational channels. In [170] the assumption on the communication channel is relaxed. Specifically, the authors guarantee strong fairness over unreliable channels. In [11] the authors propose protocols that achieve fairness in presence of resilient channels. In order to achieve this, their protocols are composed of a main protocol, for the faultless case, an abort protocol that can be executed by Alice, and two recovery protocols, which can be executed by either Alice or Bob. This scheme, which involves one main protocol and three sub-protocols, serves as a “template” for most optimistic fair exchange protocols based on generatability. Although the general scheme is correct, it is not easy to give a correct instance avoiding flaws. For instance, authors in [171] found several flaws in the certified e-mail protocol given in [11].

In [102] the authors consider a sort of *possibly-asymmetric* fair exchange problem. Motivated by the exchange of files in p2p networks, the authors consider the following variation of fairness. Informally, an exchange is fair either when both parties receive the requested file or when one party receives a payment for a file she provides. The key idea is to amortize the use of a (computationally intensive) cryptographic primitive, namely a verifiable escrow of a payment, over a sequence of faultless protocol executions. The protocol works as follows. One party, say Alice, sends a verifiable escrow of a payment to the receiver. At this point, the two parties exchange encrypted files. Alice then sends a signed escrow of her key. Bob sends Alice the keys to his files and receives from Alice the key for her files. If Alice does not send back her keys, Bob requires from the TTP the opening of the payment. Notice that the verifiable escrow of the payment can be computed and sent once and it will be usable until it is opened by a trusted agent.

**On the selection of cryptographic primitives.** Special care needs to be considered when selecting cryptographic signatures. Historically, the security of public key cryptographic primitives is defined in the so-called *single-user* setting, that can be seen as follows: there is a user that holds a private key, public key pair; the knowledge of the latter enables the creation of ciphertexts which the user can decrypt using corresponding secret key. This model ignores the fact that, in real life, there exist multiple users, each with her own keys, interacting by exchanging encrypted/signed/committed messages. In the *multiple-user* setting the interactions among parties give more power to the adversary that can collect and correlate different messages processed by different entities. In [26] and [72] the authors show that security in the single-user setting implies security in the multi-user setting for the case of encryption and signature schemes, respectively.

This does *not* hold in the case of optimistic fair encryption schemes. In [58, 172] the authors show that secure protocols in the single-user setting may not be secure in the multi-user setting *if the cryptographic primitives are not properly selected.* Specifically, in [58] the separation is obtained by presenting a fair exchange scheme that is secure in the in the single-user setting but insecure in the multi-user one. The authors then define the multi-user security model fair exchange and provide a generic setup-free (i.e., roughly, a scheme that does not require any *user key-registration* at the TTP) construction of optimistic fair exchange secure in the multi-user setting. This scheme is of theoretical interest but, may be very inefficient in practice.

Independently, in [172] showed such a separation by proving that a verifiably committed signature scheme, secure in the single-user model, becomes insecure in the multi-user setting. They defined the security notions of verifiably committed signature in the multi-user setting and proposed a concrete construction of multi-user secure stand-alone and setup-free verifiably committed signatures.

**Distributed Trusted Third Party** As for inline protocols the idea of reducing the trust over a *single* third party has been developed also in the case of off-line protocols.

In [15] the authors first introduce the possibility of distributing the role of TTP among a set of *honest neighbours* in the network. Intuitively the protocol initiator uses a (publicly) verifiable secret sharing scheme to generate $n$ shares of her message $m$. She then encrypts each share with the public keys of $n$ other parties in the network and sends all such encrypted shares to the receiving party. The receiving party sends back the expected message and receives the original message $m$. If sender and receiver act properly, the protocol terminates without the need of any external intervention. If any of the player tries to deviate from the protocol, if a sufficient number of honest players is present, the protocol fairness is guaranteed. This solution implicitly assumes the presence of timeouts and some bounds on the number of untrusted parties. In [101], the authors show that timeouts, i.e., the existence of loosely synchronised clocks, are essential for guaranteeing fairness, unless complex (and costly) cryptographic tools like secure multiparty computation are deployed.

In [16] the authors present a solution in which perfect fairness can be achieved if a majority of parties are honest but, whenever the majority of parties are dishonest, it is possible to achieve probabilistic fairness with arbitrary low probability of error.

**Transparent TTPs.** In the previous protocols with offline TTP, when the TTP intervenes, in case of problems during the communication between Alice and Bob, the TTP digitally signs some pieces of information which will be used as an affidavit. These evidences have the same effect to an adjudicator as those produced by Alice and Bob in a faultless case. The aim of the protocols described hereunder is to have a transparent TTP. At the end of the protocol, by only looking at the produced evidences, it is impossible to decide whether the TTP did intervene in the protocol execution or not.

The use of a transparent, or invisible, TTP, was first proposed by [120] in the framework of certified e-mails. The authors in [12] and [22], proposed fair exchange protocols allowing to recover, in case of problem, the original client's signature rather than affidavits produced and signed by the TTP, by means of verifiable convertible signatures. In [13] the author presented a scheme based on verifiable encryption, that is more efficient than [12]. A further improvement has been presented in [115].

**Protecting Privacy of parties.** One issue that is considered in the last years is related to the privacy of parties. Specifically, variants of the fair exchange problem have been developed to protect the identity. Typically, fair exchange protocols use signature schemes because of their non-repudiability.

In Ambiguous Optimistic Fair Exchange (AOFE), defined in [85, 87], one party receives partial signatures from the other party and she is not able to convince any other agent of the identity of the sender without the interaction with the original sender or the TTP. Intuitively, the received signatures do not endanger the identity of their sender but, at the same time, can be used to guarantee protocol's fairness.

In the context of fair exchange of digital signatures, Perfect Ambiguous Optimistic Fair Exchange (PAOFE)[160] fulfils all traditional requirements of cryptographic fair exchange of digital signatures and, in addition, guarantees that the communication transcript cannot be used as a proof to convince others that the protocol is in progress.

Both AOFE and PAOFE protect user identity from every adversary. Only the TTP is able to identify relevant information from the transcript.

A further evolution of fair exchange is Privacy-Preserving Optimistic Fair Exchange (P2OFE) [86], in which other than Alice and Bob, no one else, including the arbitrator, can collect any evidence about an exchange between them even after the resolution of a dispute.

All such schemes are typically based on the composition of cryptographic primitives like designated confirmer signatures.

### 4.2.5 Fair exchange protocols via Cryptocurrencies

In recent years, a number of cryptographic currencies or *cryptocurrencies* have been coined and used worldwide, with Bitcoin [34] being the most known and diffuse one. Intuitively, spending a cryptocoin corresponds to execute an electronic transaction in which the ownership of the cryptocoin is transferred from one user to another. Each cryptocurrency can be used to implement more than simple money transfers by providing a tool to enforce the so called *smart contracts* [33]. Such type of contracts typically have financial consequences, i.e., some amount of money is transferred between users. Intuitively, it is possible to include in each transaction a description of the conditions under which the transferred amount will be spendable. Such contracts are self-enforcing in the sense that once the parties entered the protocol they cannot withdraw their participation, unless the contract explicitly allows it. Such enforcement is guaranteed by the underlying rules of the cryptocurrencies and do not require any external authority.

Each electronic transaction with a given currency is written into a *write-only public ledger*. The impossibility of modifying/deleting a transaction, guarantees the users against double-spending. The main property that all crypto-currencies meet is that the public ledges is distributively maintained and verified by a number of peers. Informally, the set of peers managing the ledger can be seen as a distributed trusted third party.

From the technical point of view, cryptocurrencies provide the possibility of writing a program, that is attached to the transaction, and that is used to define the contract conditions. Such programs are distributively executed by a number of parties in the network. Each currency provides its own *scripting language* that is interpreted by the peers in the blockchain network. Bitcoin scripting language has been purposely designed *not* to be Turing complete [35], by not allowing loops. The main advantage of this design choice is that it is always possible to estimate the running time of a script as it cannot have infinite loops. Furthermore, such restriction gives stronger guarantees that *malicious* contracts cannot be written. On the negative side, not-being Turing complete limits the complexity of the contracts that can be specified by using the Bitcoin scripting language. Other platforms, like Ethereum [63], Hyperledger [89] or Ubiq [153] do support Turing-complete scripting languages. On the one side this design choice allows the specification of arbitrary (computable) contracts, with obvious effects on the expressiveness of the language. On the negative side, such scripting languages do have to solve issues related to malicious contract specifications. One example is an attack related to a vulnerability in the DAO scripting language emerged in recursive contracts [62]. Another issue to be managed is the running time needed to evaluate a contract that, for Turing-complete programs, is unpredicatble. The latter issue is typically solved by different platforms by assigning a maximum amount of resources for the evaluation of a contract.

Fair exchange protocols can be implemented by using a smart contract called the

*Zero Knowledge Contingency Payment* [32]. Informally, let us assume a player, the Buyer, wants to buy a element $x$ she does not know and pay for it a given amount $t$ only if the element $x$ meets some property described by a (computable function) $f$. As an example, assume that $x = (p, q)$ is the factorization of some number $N$ and the function $f$ simply verifies such property, i.e, $f_N(p, q) = true$ iff ($N = pq$ and $p \neq 1$ and $q \neq 1$). This is a typical example of the fair exchange problem in which the Buyer cannot compute the factorization of $N$ and she is willing to pay $t$ whoever is able to provide $N$'s prime factors. Clearly, the Buyer will never pay the Seller before being sure of obtaining the solution to her problem. Similarly, the Seller will never provide $N$'s factors before being sure of getting the payment. Informally, this problem can be solved by publishing on the ledger a contract $C$ that states *"The Buyer puts aside \$t. This money can be claimed by whoever provides x such that f(x)=true. If, after T time units, nobody provides a solution, this amount of money goes back to the Buyer".*

This type of transaction is feasible only if the specific scripting language is able to verify that $f(x)$=*true*, i.e., if the scripting language is Turing-complete. Since some cryptocurrencies do not provide some features, a way of implementing this smart contract is the so-called *hash-locked transaction* that can be described as follows. Seller and Buyer engage in an offline protocol in which the seller encrypts $x$ using a random key $k$, generating $\hat{x} = E_k(x)$ and computes $y = Hash(k)$. She then sends to the buyer $f, y$ and a zero knowledge proof that $f(E^{-1}_{Hash^{-1}(y)}(\hat{x})) = true$. Notice that, this interaction occurs *outside* the blockchain and, thus, buyer can verify the zero-knowledge proof regardless the restriction of the scripting language. If the proof is correct, the buyer can publish on the ledger a transaction like the following *Pay \$t to whoever provides a preimage of y and provides a signature that can be verified with the public key of the Seller. If after T time units, nobody provides a solution, this amount of money goes back to the Buyer.*

In [49] the authors focus on the fair exchange for digital services. They consider the scenario in which a Seller provides a file storage service and a Proof of Retrievability (PoR) service [93], that is, upon request, the system generates a proof that the requested file is stored by the seller. The PoR service can be seen as a instance of the fair exchange where the Buyer provides a payment only if the seller provides the proof of retrievability. If the scripting language is rich enough, the standard ZKCP can be used to implement such fair exchange. On the other hand, if the language is restricted, the hash-locked transaction technique described above cannot be used. The key observation is that the offline transaction executed by the parties should include itself a proof of retrievability that has to be correctly verified by the buyer *before* the payment transaction is started. Once the proof is verified, the buyer may simply stop the protocol, not paying the seller, as she already obtained the service. This example can be generalized to every service in which the seller does not provides a digital good $s$ but it sells the *proof that she knows* $s$ such that $f(s) = true$, for an efficiently computable function $f$.

In this work the authors define the notion of *Zero-Knowledge Contingent Service Payment (ZKCSP)* and provide two protocols for this new notion. The solutions are extensions of the hash-locked transaction mechanism described above, in which the hash-locking value can either can either be the pre-image of one of two different hash functions having the same range. They also show an attack to ZKCP that is of independent interest.

In [21] the authors start from the observation that whenever the scripting language is restricted, it is extremely hard to design non-trivial smart contracts. They thus consider the problem of creating efficient non-trivial scripts in Bitcoin. They provide a technique for generating ZKCP protocols, using only *standard* (Bitcoin) transactions, for every $f$ for which the language $\{x : f(x) = true\}$ has an efficient zero-knowledge proof of knowledge.

From a theoretical point of view, the authors in [29] define new primitives that allow *fair* computations and that can be executed on top of the BitCoin ecosystem. Although the paper is based on the BitCoin ecosystem, it is independent from this specific technology. Specifically, the authors introduce an ideal primitive they call the *claim-or-refund*, that allows a sender to make a deposit that can be conditionally transferred to a receiver. They further present a multi-party extension that allows what they call multiparty with penalties. Intuitively, in an $n$-party setting, one malicious party might try to stop the protocol after having received her expected output but before other parties have received theirs. In this case, each honest party will receive a pre-specified payment.

### 4.2.6 Multi-party fair exchange

In multi-party fair exchange, multiple parties need to exchange items in a fair way. Multi-party fair exchange protocols can be mapped to graphs with each party being a vertex and each exchange between two parties being an edge. Clearly, different protocols induce different graph *topologies*.

In [70], Franklin and Tsudik propose a classification, mainly depending on the topology of the exchange. Bao et al. [23] and Franklin and Tsudik [70] concentrated on a ring topology.

Another topology is the more general matrix topology, where each entity may desire items from a set of entities and offer items to a set of entities. Such protocols have been proposed by Asokan et al. on synchronous networks in [8].

A fundamental difference between non-repudiation and fair exchange protocols is the following. In a fair non-repudiation protocol, the originator sends some data with a non-repudiation of origin evidence to a recipient, who has to respond with a non-repudiation of receipt evidence. The sent data is generally not known to the recipient a priori. In a fair exchange protocol each entity offers an a priori known item (i.e. a kind of specification of the item is known a priori but not its precise content) and receives another item, also known a priori. Therefore, in a multi-party fair exchange protocol one can imagine sending an item to one entity and receiving an item from a different one. In non-repudiation it does not make sense that one entity receives some data and a distinct entity sends the corresponding receipt. Thus a ring topology is not sound. The most natural and here considered generalisation seems to be a one-to-many protocol on a star topology, where one entity sends a message to $n > 1$ receiving entities who respond to the sender. In [100, 113] the authors proposed the first multi-party non-repudiation protocols, with both online and offline TTP. The main motivation for these protocols is a significant performance gain with respect to n two-party protocols. Afterwards, the authors in [127] extended our work with online TTP, in order to permit the sending of different messages to each entity.

| | TTP | Ch | Fair | NRO | NRR | Aut | Int | Con | Time | T. Aut | Context |
|---|---|---|---|---|---|---|---|---|---|---|---|
| [169] | ON | U | S | Y | Y | Y | | | N | N | General |
| [170] | OFF | U | W | Y | Y | Y | | | N | N | General |
| [9] | OFF | O | S | Y | Y | Y | | N | Y | N | General |
| [9] | OFF | R | S | Y | Y | Y | Y | | Y | | General |
| [157] | OFF | U | S | N | N | | | N | Y | N | General |
| [2] | ON | R | S | Y | Y | | Y | Y | N | N | Email |
| [27] | OFF* | U | P | N | N | N | N | N | Y | N | Contract signing |
| [71] | ON | R | S | N | N | N | N | P | N | N | General |
| [115] | OFF | R | W | Y | Y | Y | Y | N | | N | Item vs signature |
| [114] | NO | U | P | Y | Y | Y | | | Y | N | General |
| [52] | ON | R | S | Y | Y | Y | Y | | Y | Y | Payments |
| [15] | OFF | O | S | N | N | N | Y | Y | Y | N | General |
| [16] | OFF | O | S/P | N | N | N | Y | Y | Y | N | General |

Table 3: Two-party methods

### 4.2.7 Schematic representation

Table 3 reports the main results for two-party fair exchange presented in this section. For each paper, we specify the following:

- TTP: TTP involvement. On-line, Off-line, NO TTP, INline. For [27] see the discussion above.
- Ch: Communication Channel type: Unreliable, Reliable, Operational
- Fair: Fairness. Strong, Weak, Probabilistic

The following are mainly yes/no properties.

- NRO: Non-repudiation of origin
- NRR: Non-repudiation of receipt
- Aut: Authentication of parties
- Int: Integrity.
- Con: Confidentiality. For [71], the confidentiality can be defined *Partial* in the sense that it is not preserved if one party misbehaves. Notice that, [71] is an online protocol.
- Time: Timeliness. Timeliness over unreliable channels is obtained via timeouts.
- T. Aut: Temporal authentication (timestamping).

Table 4 summarizes the main results for multi-party fair exchange presented in this section.

| | TTP | Ch | Fair | NRO | NRR | Aut | Int | Con | Time | T. Aut | Context | Top. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [10] | OFF | U | S | Y | Y | | | | Y | | General | Ring |
| [70] | IN | | S | | | | | Y | | | General | Ring |
| [113] | OFF | R | S | Y | Y | | | | Y | N | General | 1-N |

Table 4: Multi-party approaches

### 4.3 Conclusions

Only a fraction of the fair exchange methods is applicable in SPECIAL use cases. The methods that offer only probabilistic guarantees are not completely satisfactory in a context where failure to achieve fairness makes it possible to insert in the transparency ledger events that appear to be certified while they are not. This could be used to create false evidence that a data controller has violated the GDPR, thereby exposing the controller to serious sanctions.

Adopting non-probabilistic fairness, however, rules out the approaches without TTP. Of the remaining methods, those with online TTP all require a number of additional messages to/from the TTP, that would not scale to the amount of data processing events expected in the application domains addressed by SPECIAL.

According to the above considerations, the first approach investigated in SPECIAL adopts fair exchange methods with offline TTP. It is an adaptation of Micali's optimistic protocol aimed at supporting very large data exchange. This approach is introduced and illustrated in D2.4.

Having multiple, distributed TTPs may improve robustness and scalability, so it is an interesting topic for further research in SPECIAL.

Of course, should the approaches with offline TTP fail to scale to the big data scenarios of SPECIAL, it would be necessary to focus again on fair exchange protocols with weaker fairness guarantees.

## 5 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 [130] 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 ready-made, set in stone static consent forms, there is a need to develop the technologies necessary to facilitate a novel way of requesting and giving consent that we call *dynamic consent*. The main features of dynamic consent are: (i) Only the relevant information for the *specific situation* should be presented; (ii) 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.

## 5.1 Requirements

The challenge for *dynamic consent* is to marry such a system with the legal requirements. Fortunately, the Art. 29 Working Party (recently replaced by the European Data Protection Board, or EDPB) 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 [131] which examines the necessity for a reform of Directive 2002/58EC [64].

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) [51]. 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.

[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

*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.*[12]

*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.

*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.

## 5.2 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

[12] Some of the data mining techniques used for this purpose produce anonymized data, hence they have different implications on consent. Such privacy preserving data mining techniques are illustrated in Section 7.

- 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* 6.

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 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.

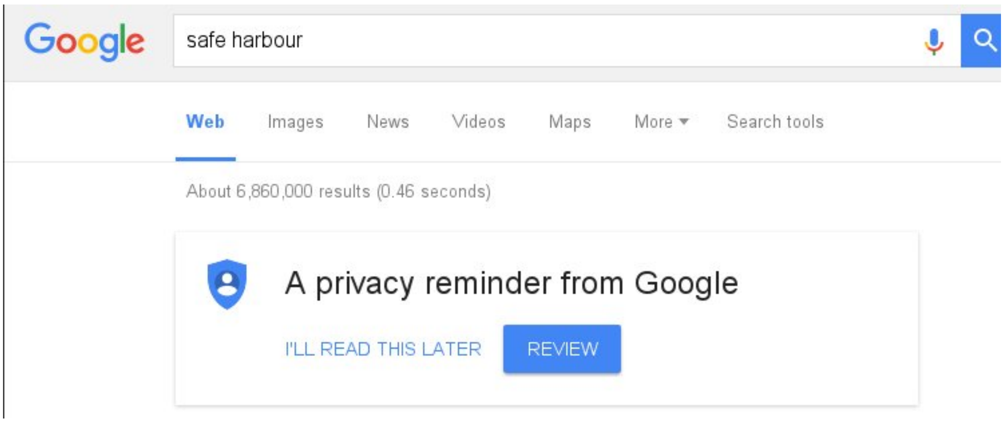
## 5.3 Candidate Consent Mechanisms and Limitations

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.

### 5.3.1 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



safe harbour

About 6,860,000 results (0.46 seconds)

A privacy reminder from Google

Figure 1: A classic way to grab an affirmative action

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 [117] 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. [4] and Borgesius [44] 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. [4] 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 [44] 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 [44] 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.

### 5.3.2 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 [125] *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-Schönberger [116] and most prominently by Stephen Hawking[13]. 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 [163] 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[14]. 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 [81] 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.

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

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

### 5.3.3 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 [47]. According to Steinsbekk et al. [145] *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 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.

### 5.3.4 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 [68] 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[15], a tool for increasing web browsing privacy through control of cross-site requests [138], 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, 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.

### 5.3.5 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. [68], Acquisti et al.[4] and Borgesius [44] 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 [129] 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 [128], ODRL [90], KAoS [46], Rei [94] and Protune [38], 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

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

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. [156] and the research done in PrimeLife [83] 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 [44] highlights the fact that companies rarely use privacy as a competitive advantage. Recent articles by Gürses and del Alamo [76] and Hansen [79] 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. When possible, the chicken-and-egg issue can be mitigated with *privacy preserving data mining techniques*, that is, a kind of analytics that produces anonymized results. The expected advantages are twofold: on the one hand, the anonymized nature of the output may encourage data subjects to opt in. On the other hand, anonymous information can be more easily shared. The main privacy preserving data mining approaches are illustrated in Section 7. This survey is also useful to verify whether the goals of the pilots necessarily need consent, and to what extent.

## 5.4 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 [28]. 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. [98] 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 [90] currently in the process for standardisation by the W3C's Permissions and Obligations working group still suffer partially from semantic ambiguities [146] 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. [97] 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* 6 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 [143] 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 [45] 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 [144] 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.

### 5.5 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. [4] and Borgesius [44]. 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 SPECIAL components (e.g. the transparency log, compliance checking algorithms etc...).

## 6 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. 6.1. 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. 6.2.

### 6.1 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.

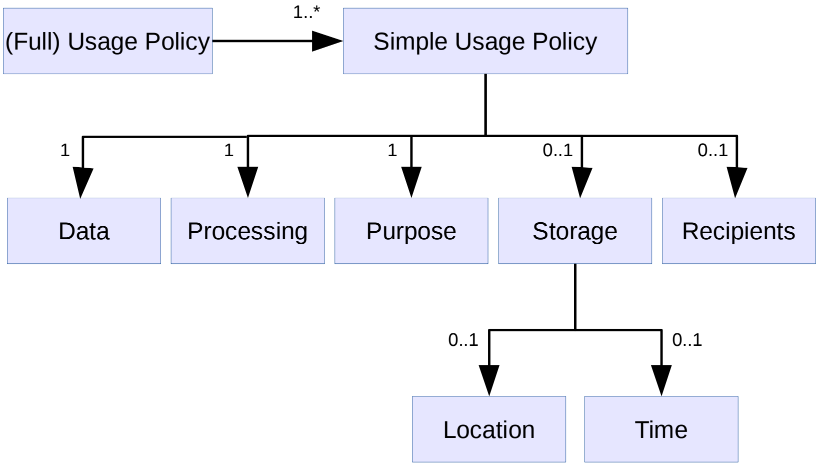


Figure 2: The minimum, core usage policy model (MCM)

### 6.1.1 Usage policy model

The reference scenario as well as the pilots described in Deliverable D1.1 involve a set of 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[16] contains a nontrivial vocabulary of data categories that includes also dynamic data, collected 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[17] 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 the existing legislation (e.g. note the attention to 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 order to derive and reach consensus on vocabularies to be used to represent personal data, processing of personal data and related policies and regulations. The first activities along this line have been a workshop[18] and the constitution of a W3C community group[19].

**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

[16] P3P, https://www.w3.org/P3P/
[17] http://xmlns.com/foaf/spec/
[18] https://www.w3.org/2018/vocabws/
[19] https://www.w3.org/community/dpvcg/

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 such descriptions are undecidable (e.g. algorithm equivalence and output properties). Furthermore, an algorithmic description of data processing is of little meaning to most 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 and its second version D1.5); 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).

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. For this reason, the privacy preserving data mining techniques illustrated in Section 7 are particularly interesting for SPECIAL: they combine the analyses sorely needed by companies with anonymization, and as such fit the above vision and may be relevant to the pilots where the results of the analyses describe global (non-individual) behavioral trends (cf. the old and new DT pilots, where the analyses are used to plan public infrastructures and improve network quality). 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.[20] 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[21], whose purpose categorisations can be exploited as a basis for developing SPECIAL's purpose-related 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 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 [30]). 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

[20] 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.

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

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).

### 6.1.2 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[22] 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 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 functionality assertions and datatype restrictions (hence storage time constraints). We expect the ontologies that describe data, processing, purposes and recipients to fit completely within OWL2 EL. This is important because inference over OWL2 EL knowledge bases is tractable and there exist well-engineered, scalable engines tailored to this profile.

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

In order to deal with the features that are not covered by OWL2 EL we are developing specialized engines.

Theoretical results say that if a datatype (called *concrete domain* in the description logic jargon) enjoys a so-called *p-admissibility* property then reasoning in OWL2 EL remains tractable [18]. 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 [18] (in particular, the concrete domain $\mathsf{Q}$ supports only $>$, not $<$). Indeed, it has been proved that non p-admissible domains as well as functional roles make reasoning intractable. Also the interplay between intervals and policy sets makes reasoning intrinsically expensive (NP-hard). So we have to develop ad-hoc solutions for scalable reasoning about policies; the first results have recently been published.[23]

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 [56]. 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.

### 6.1.3 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.

### 6.1.4 Relationships with existing vocabularies and policy languages

*P3P* [53]. 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* [77]. 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

[23] P. A. Bonatti, *Fast Compliance Checking in an OWL2 Fragment*, Proc. of the Int. Joint Conf. on Artificial intelligence (IJCAI), 2018.

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* [154]. 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* [122] that make complete reasoning undecidable [17, 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* [95]. 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* [41]. 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.

### 6.1.5 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 [148] 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.

### 6.1.6 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 [165]), and an encoding in description logics [148] (compatible with the encoding of the MCM in OWL2). Only a fragment of ODRL has been given a formal semantics [24]. 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*.[24]

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,

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

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*) [54]. 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.

## 6.2 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.

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.[25] 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

### 6.2.1 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

[25] 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.

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[161].

### 6.2.2 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)[26]. 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.

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.

### 6.2.3 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.

[26] 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.

*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".

### 6.2.4 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.

### 6.2.5 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 [41] 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).

### 6.2.6 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 [6], 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 [42, 73]. 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 [40, 43] 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.

## 6.3 Policy synthesis for derived data

One of the policy-related computational tasks of interest for SPECIAL's industrial partners is the automated identification of the policies to be associated to *derived data*, that is, the data resulting from the processing of personal data.

The initial idea was using policy templates (i.e. parameterized policies), from which the policies applying to derived data would be obtained by instantiating the template's parameters with the derived dataset and its metadata.[27] Templates (a.k.a. *parametric polymorphism*) are now being replaced with a different approach based on *inclusion polymorphism*, that is, subclasses.

Recall that policy are specified as classes, and authorize all the data usage actions that are instances of the policy (D2.1, Sec. 1). This naturally allows a single consent policy to cover the range of all (similar) data uses that can be formalized as subclasses of the consent policy. For example, a single *consent policy authorizing pseudonymized mobility habits to be analyzed for the purpose of planning public services and infrastructures*,[28] naturally covers – as subclasses – a set of similar data uses such as

- anonymized location mining for infrastructure planning,
- pseudonymized location mining for public office placement,

and so on. The variety of derived data (resulting from different location analysis algorithms tailored to different specific purposes) are all covered by the super-policy, that includes the more specific, ad hoc uses such as those exemplified in the above bullet list.

Note that this is a natural way of dealing with data re-purposing, subject to the limitations of the GDPR, Art. 6, point 4, and recital (50). A consent policy allows the data controller to change the specific purpose of processing as long as it belongs to the purpose class specified in the policy. This formalizes the requirement that there must be a clear link, or similarity, between the new purpose and the one for which data had initially been collected (as they must belong to the same class).

We conclude this paragraph with a technical remark on the two forms of polymorphism discussed here. Parametric polymorphism (templates) – in the context of programming languages – has the advantage of allowing static type checking, while the advantage of inclusion polymorphism (subclassing) is greater flexibility. In the context of OWL-based policy languages (like SPECIAL's), the advantages of parametric polymorphism are less significant, since the behavior of *all* policies is decidable, hence statically analyzable.

## 6.4 Policy synthesis as an authoring aid

The process of policy authoring might require additional aid in order to support users to express semantically correct policies, which reflect their data privacy preferences. This process of creating policies is complicated by the involvement of derived data, since it can be difficult to comprehend how this data will be produced, which external information sources will be queried for it, and what other information might be inferred from this produced data. Allowing users to express restrictions for this kind of information requires the expression of statements about data the controller is allowed to infer from certain

[27] A similar approach was followed in *role based acccess control)* (RBAC), by introducing *role templates*. Roughly speaking, also in that case templates were used to define functions from entity descriptions to authorization sets (i.e. policies).

[28] This example has been inspired by the first pilot of DT. The new pilot provides similar examples, where the the planning of public services is replaced by the improvement of the quality of network connections and mobile services.

information and data sources that might be used for this. These statements can be very complex to express precisely and correctly in natural language, so it is even more difficult with a policy language.

To emphasize the need for additional guidance during the process of policy authoring, we further take the users into consideration, who are supposed to express their privacy preferences. Some users can be overwhelmed just by specifying rules that regulate who is able to access their personal data and who is not. With derived data the complexity is increased, since users are now supposed to determine who is able to produce which data based on already disclosed data. Further restrictions with regard to the Big Data application that processes the data in question are thinkable. However, these applications are also complex itself. For all these reasons, It is clear that users need to be guided by the user interface when creating policies. Therefore, reasonable limitations need to be made, which are set by the user interface that is designed to author these usage policies.

The user interface also offers the review and modification of already created or existing policies, thus the policy synthesis is important to look at. In particular, users must be able to interpret the visualization of a policy and extract its semantic meaning. This further includes the consequences and the impact of the policy in different scenarios. Here, predefined policies and templates can aid users in the process of authoring. It is conceivable that users simply select and use a policy template that is close to their own privacy preferences and just adjust certain aspects of it. Descriptions of these policy templates in natural language might be helpful to users in order to chose the right template as basis. These descriptions could include information on the impact and consequences in exemplary scenarios.

It would be helpful to investigate how users prioritize the different aspects of a policy. Do they put emphasis on the storage location or duration, on involved third parties, the purpose of processing, or on the data itself? Does this vary among different user groups? Is this different from use case to use case? Is this dependent on the data and its context? A user study to answer all these questions could be difficult to realize, however a set of common privacy concerns and preferences can be helpful to define policy templates that are useful to users.

A first set of alternative UIs has been designed in deliverable D4.1, to investigate the effectiveness of different interaction/description approaches in achieving the above goals. Deliverable D4.2 provides a first empirical assessment of those interfaces.

# 7 Privacy-preserving Data Mining

In this section we review the main privacy-preserving data mining methods introduced in the literature. These methods are interesting in SPECIAL for the reasons illustrated in the previous sections, e.g. data subjects may be encouraged to opt in for data analysis and the results of the analysis can be more easily shared (as it could happen, for instance, in the new DT use case, where DT could carry out the analysis and transmit the anonymized result to Motionlogic).

Keeping anonymity in the age of big data is a difficult task, due to the huge quantities of information that are collected and analyzed at every moment, and can be linked together. A concrete case is given by [124], where it is shown how anonymous data could be enough to identify people, using a case study based on Netflix's public data.

The researchers show that information usually believed to be benign can be combined with other data and used to discover private information. From these cases, it is easily deducible that anonymity could not be enough anymore, making necessary appropriate methods for managing information that can guarantee confidentiality and privacy.

This section aims at summarizing the current state-of-the-art in Privacy preserving data analysis, focusing on *clustering*, a very common set of techniques used to select and group together data that have similar characteristics. Clustering applications are numerous, including, but not limited to market research and segmentation, consumer opinions, sales analysis and so on. The most famous clustering algorithm is *k-Means*, used in almost every area of data analysis, both in the academic and business fields. Because of how it is built and implemented, the *k-Means* algorithm works and releases data without any guarantee of privacy on the data used: even if the algorithm releases only data that are the result of a processing, advanced algorithms and statistical inference techniques could allow a potential attacker the reconstruction of the original data [91], [119]. Already from this we can deduce the importance of an adequate use of information not only during the data collection but also during their treatment.

This section is organized as follows:

- In Section 7.1, we briefly define the clustering problem and the k-Means algorithm;
- In the first part of Section 7.2, we illustrate the relationship between k-means and privacy, together with the problems that this requirement may bring;
- In 7.2.1, we describe some of the classic approaches used in past years to resolve privacy issue
- In Section 7.3, we talk about more recent approaches used today and which are being studied to guarantee data privacy, e.g. differential privacy.
- In Chapter 7.4, a final discussion on these new privacy-preserving methods, showing some results in literature.

## 7.1 k-Means Clustering

$k$-Means [110] is probably the most known among the flat clustering algorithms [112], aiming at creating a partition of a set composed by $n$ objects into $k$ sets. In particular, given a set $D = \{x_1, x_2, \dots, x_n\}$ of points in a $d$-dimension space, we want to find a $D$ partition of $k$ sets $\{\omega_1, \omega_2, \dots, \omega_k\}$ (clusters) such that the Residual Sum of Squares (RSS) defined by:

$$\text{RSS} = \frac{1}{N} \sum_{j=1}^{k} \sum_{\mathbf{x}_l \in \omega_j} \|\mathbf{x}_l - \mu_j\|^2 \tag{1}$$

is minimum.

The partition is generated by an iterative algorithm of progressive adjustments; a first set of $k$ seeds $\mu_1, \mu_2, \dots, \mu_k$ is chosen among the points in $D$; every remaining point in $D$ is then associated to the closer seed using some metric distance defined in the point space (e.g. Euclidean distance). A first clustering is then obtained, and the *centroid* is then computed for every cluster:

$$\mu_j \leftarrow \frac{1}{|\omega_j|} \sum_{\mathbf{x}_l \in \omega_j} \mathbf{x}_l \tag{2}$$

Two steps can then be distinguished, the former where each point is re-assigned to the closest centroid and the latter where the centroids are recomputed. These two steps are then repeated in an iterative manner until convergence or another stop criteria (e.g. fixed repetition number). The algorithm is summarized in Algorithm 1.function and has several local minima; therefore, although the algorithm convergence is guaranteed when the Euclidean distance is adopted, it can converge to any of the local minima. From this point of view, the choice of the initial seeds is crucial for the final solution. On the other hand, different seeds sets can result in different clustering solutions. One of the most known k-Means version was proposed by Lloyd [107].

k-Means clustering is used in many real-life applications, including statistical analysis on consumers, biological research, etc., that in many cases use very large datasets with millions of points.

**Algorithm 1** k-Means

```
random choice of $o_1, o_2, \ldots, o_k$ centroids;
while there are changes in some centroid: do
    for all $x_i$ do:
        compute distance between $x_i$ and every centroid $o_1, \ldots, o_k$
        assign $x_i$ point to the cluster with closer centroid;
    end for
    for all cluster $O_j$ do:
        update centroid $o_j \leftarrow \frac{\sum_{x_l \in O_j} x_l}{|O_j|}$
    end for
end while
return computed assignments and centroids
```

## 7.2 k-Means and privacy

Privacy requirement must be enforced whenever $k$-Means is applied to a database containing sensible data. *Data subjects* can choose to keep them hidden, but there could be situations where other entities (parties), possibly also other data subjects, want to share cluster analysis. The purpose is then to avoid access to private data to any other party during algorithm execution.

Different approaches can be devised depending on the type of privacy we want to attain:

**General** : we want to obtain a database from which an attacker can not infer the original data. In this case, dataset perturbation approaches are usually considered: data controllers make a noisy dataset version adding noise to the original dataset.

Noisy data can be viewed from every other parties, so that every single party can collect the whole dataset and then perform the clustering process.

As a special case we also consider *Secure Multi-Party Computation (SMC)*: a given number of participants obtain the result of a function by collaborating in the use of data, but without sharing data in their specific possession.

**Differential Privacy** : we want to obtain similar responses with or without the data regarding a single case.

The following sections consider in detail each one of these cases, together with the main applicable approaches.

### 7.2.1 Traditional approaches

When we are considering the general case in which we aim to hide sensibile data from users, clustering using approaches based on dataset transformation are usually applied. Such transformations can be either deterministic or random. In the former case, a matrix $T \in \Re^{N \times M}$, where $N$ is the number of points and $M$ the size of each vector, is multiplied to the input data matrix: $Y' = Y \cdot T$. A projection technique which belongs to this class of approaches is PCA. Although it has been shown in [74] that it gives the best performance, PCA is impracticable for computational reasons when the dataset dimensionality is large. A cheaper projection method is presented in [31], but it is less accurate with respect to PCA.

When the perturbation is random, on the other hand, dataset perturbation approaches can be divided in two different categories depending on the type of introduced perturbation:

**Additive Data Perturbation (ADP) methods** : a random variable $\epsilon$ with a given probability distribution (e.g. Uniform, Normal, etc.) with expected value 0 and given variance is added to data $Y$ resulting in $Y' = Y + \epsilon$; it should be noted that additive perturbation does not preserve the distance between points and this can have a negative effect on clustering.

**Multiplicative Data Perturbation (MDP) methods** : the data perturbation are obtained by multiplying the input data $Y$ with a random variable $\epsilon$ sampled from a given probability distribution with expected value equal to 1, obtaining $Y' = Y \cdot \epsilon$.

### 7.2.2 Secure Multi-Party Computation

As discussed above, Secure Multi-Party Computation (SMC) refers to a situation in which each party wants to maintain his/her data hidden from the others, but all of them want to share the clustering solution. As it can be expected, the main drawback of the approaches trying to solve this problem is the high communication overhead: parties need to continuously exchange messages between each others, and this could represent a problem in real settings.

Among all possible approaches dealing with this setting, we briefly describe Distributed $k$-Means, which is also necessary to the Privacy-Preserving $k$-Means algorithm described
in [91].

### Distributed $k$-Means

The Distributed $k$-Means algorithm aims at giving a solution to the following problem. Two parties $A$ and $B$ want to make a partition in $k$ clusters $\{\omega_1, \omega_2, \dots, \omega_k\}$ of the join of their data. $A$ owns the points $\{x_1, x_2, \dots, x_z\}$ and $B$ the points $\{x_{z+1}, x_{z+2}, \dots, x_n\}$. $A$ and $B$ want to cluster the points without disclosing their own data to the other. Let's assume that there is a *Trusted Third Party* (TTP) that can be involved in the process. $A$ and $B$ can make a first clustering iteration of their data; we will call $\{\omega_1^A, \omega_2^A, \dots, \omega_k^A\}$ the clusters computed by $A$, and with $\{\omega_1^B, \omega_2^B, \dots, \omega_k^B\}$ the results from $B$. For every cluster, $A$ and $B$ send to TTP:

- the sum of all points in every cluster, i.e. $s_j^p = \sum_{x \in \omega_j^P} x$ with $p \in \{A, B\}$
- total number of elements in every cluster, i.e. $|\omega_j^P|$ with $p \in \{A, B\}$

with these data, TTP can compute the centroids for every cluster $\mu_j = \frac{s_j^A + s_j^B}{|\omega_j^A| + |\omega_j^B|}$ which will be sent to $A$ and $B$; $A$ and $B$ can use these data in the following clustering iteration.

The main problem with Distributed $k$-Means is that it assumes a TTP which needs to know sensible data.

### Privacy Preserving (PP) $k$-Means

The problem described above is usually known as *Weighted Average Problem* (WAP) and can be formalized as follows: there are two parties $A$ and $B$; $A$ owns $c$ pairs $(x, n)$ with $x \in \mathbb{R}$ and $n \in \mathbb{N}^+$; similarly, $B$ is associated to an anlogous pair $(y, m)$. We need a protocol to compute

$$((x, n), (y, m)) \to \left( \frac{x + y}{n + m}, \frac{x + y}{n + m} \right) \tag{3}$$

preserving privacy. With this notation we want to emphasize that $A$ and $B$ give $(x, n)$ and $(y, m)$ respectively and both receive $\frac{x+y}{n+m}, \frac{x+y}{n+m}$.

There is a weak version of the same problem where both parties know $n$ and $m$; this version is not taken into account because less interesting for us.

The literature offers many solutions to the WAP problem, for example [111] or [91] in which WAP problem is seen as an instance of *Private Rational Polynomial Evaluation* (PRPE) Problem, which can be stated as:

Let $F$ a finite field; $A$ owns two polynomials $P$ and $Q$ with coefficients in $F$. $B$ owns two points $\alpha$ and $\beta$ in $F$. Both parties want to compute $\frac{P(\alpha)}{Q(\beta)}$; so, we want to compute the following function:

$$((P, Q), (\alpha, \beta)) \to \left( \frac{P(\alpha)}{Q(\beta)}, \frac{P(\alpha)}{Q(\beta)} \right) \tag{4}$$

preserving privacy.

PRPE problem can be resolved with through another problem known as *Oblivious Polynomial Evaluation* (OPE), which can be stated as:

$A$ has a polynomial $P$ of degree $d$ over the finite field $F$, while $B$ has an element $\gamma \in F$. We want a private protocol such that $B$ receives just $P(\gamma)$ and A nothing.

**Algorithm 2** [123] protocol for OPE

```
    $A$ computes a random polynomial $T$ with degree $r = sd$ (with $s$ parameter) such that $T(0) = 0$;
    $A$ compute another polynomial $G(x, y) = T(x) + P(y)$ such that $G(0, y) = P(y)$;
    $B$ computes a random polynomial with degree $s$ such that $S(0) = \gamma$. $B$ can compute $P(y)$ interpolating $R(x) = G(x, S(x))$ (it should be noted that $G(0, S(0)) = P(S(0)) = P(\gamma)$);
    $B$ learns $r + 1$ pair $(x_i, R(x_i)) \; \forall \; 1 \leq i \leq r + 1$;
    $B$ interpolates $R(0) = P(\gamma)$ using values computed in the previous step;
```

In [123] a protocol to resolve OPE problem is described, that we briefly summarize in algorithm 2.

The main issue of the algorithm described above is when it requires $A$ and $B$ to communicate with each other.

In [123] a secure method to do the transaction described in Algorithm 3 is described.

**Algorithm 3** [123] transaction method

```
    $B$ chooses a point form a set $C$ of $m(r + 1)$ (with $m$ parameter) random points over $F$ all different from zero;
    $B$ chooses a random subset $T \subset C$ of $d + 1$ points. for every point $x_i \in C$ is defined:
        $y_i = \begin{cases} S(x_i) & \text{if } x_i \in T \\ \text{random value} & \text{otherwise} \end{cases}$
    $B$ sends to $A$ the set of pairs $\{(x_i, y_i)\} \forall_{i=1}^{m(r+1)}$;
    $A$ sends to $B$ all values $G(x_i, y_i)$;
    $B$ uses from all $G(\cdot)$ values received $A$ only these that have $x_i \in T$ to computer $R$;
```

Other methods and informations can be found in [123].

In [91] it is shown a privacy preserving solution for PRPE problem using OPE as an oracle. This is briefly described in Algorithm 4.

**Algorithm 4** [91] PRPE solution

```
    $A$ choices a random point from $z \in F$ ;
    $A$ computes two new polynomial $zP$ and $zQ$;
    $B$ compute $zP(\alpha)$ and $zQ(\beta)$ using OPE protocol;
    $B$ computes $\frac{P(\alpha)}{Q(\alpha)}$ using $\frac{zP(\alpha)}{zQ(\alpha)}$.
```

Using PRPE, is then possible to give a solution to the WAP problem defining for $A$ the polynomials $P(w) = w + x$ and $Q(w) = w + n$ and forms $\alpha = y$ and $\beta = m$. The output given to both parties $A$ and $B$ will then be $\frac{x+y}{n+m}$. The correctness proof is shown in [91]. We can resume Privacy-Preserving $k$-Means in Algorithm 5.

**Algorithm 5** PP$k$-Means

```
assign $x_i$ to the closer centroid;
for every cluster $j$ do
    compute $s_j^p$ and $|\omega_j^p|$;
    update the centroid $\mu_j$;
end for
repeat from 1. until convergence;
return $\mu_1, \mu_2, \ldots, \mu_k$.
```

### 7.2.3 Attacks

These methods have been quite extensively explored in the past years. However, it has also been shown that, when some assumptions are verified, original data (or a potential useful approximation) can be recovered.

**Eigen-analysis attack** [88]: which aims to remove the noise using data correlation;

**Bayesian Attack** [88]: which tries to estimate the probability $P(Y|Y')$ of an hypothesis of original data given the perturbed data.

**Distribution Analysis Attack** [5]: using an estimate of the data original distribution computed through $Y$, the attacker could infer the real data;

**Known input-output attack** [105]: assumes that an attacker knows a part of the original data and the corresponding noise version; these informations, together, can be used to construct a system of equations and recover the original dataset. Knowing the distances between the points in the original space.

**Known Sample Attack:** the attacker has a set of indipendent samples taken from the same data distribution; using a PCA-based attack [105], the attacker can obtain an estimate of data. A similiar type of attack is AK-ICA, described in [75].

For further details, [106] show a a survey of the techniques discussed.

## 7.3 Differential Privacy

Intuitively, Differential Privacy (DP) goal is to avoid that a single element of the dataset could influence the result too much so that a potential attacker could recover original data by inference on results.

More formally, given an algorithm $A$ that takes as input a dataset $D$, the algorithm is called $\epsilon$-*Differentially Private* if, for every possible pair of datasets $(D_1, D_2)$ that differ for a single element, and for every possible subset $S \subseteq \mathbf{im}A$, we have

$$Pr(A(D_1) \in S) \leq \exp(\epsilon) Pr(A(D_2) \in S). \tag{5}$$

Another widely adopted definition is that of $(\epsilon, \delta)$-*Differentially Private* algorithm; it differs from the previous in terms of an additive factor, that is:

$$Pr(A(D_1) \in S) \leq \exp(\epsilon) Pr(A(D_2) \in S) + \delta. \tag{6}$$

So, Differentially-private algorithms need an additional parameter, named *privacy budget* (usually indicated with $\epsilon$), that is a trade-off measure between results accuracy and the level of privacy wanted.

The privacy budget is usually computed as a function of some query *sensitivy* indexes, that are a measure of how much the results of a query change as a consequence of "little changes" in the data. The sensitivity indexes for a query $f$ most widely used in literature are the *Global Sensitivity* $GS_f$ and the *Local Sensitivity* $LS_f$. They can be formally defined as:

$$GS_f = \max_{x,y:d(x,y)=1} \|f(x) - f(y)\| \tag{7}$$

that is the maximum difference between the query results when the query is applied on every possible pair of databases differing for exactly one row.

Instead, *Local Sensitivity* is defined as:

$$LS_f = \max_{y:d(x,y)=1} \|f(x) - f(y)\| \tag{8}$$

the difference between local and global sensitivity is that the former is computed on *every possible pair of databases* with distance 1, the latter on all databases $y$ that differ from our *real data* $x$ for at most one row. This is an important point because many procedures used to make data Differentially-Private employ a different amount of noise based on the type of sensitivity index used; $GS$ can be too "meticulous", taking into account also pairs of highly unlikely databases, while $LS$ needs the real database and can be too hard to compute.

In this section, we briefly discuss the main variations of $k$-Means preserving differential privacy.

### 7.3.1 SuLQ $k$-Means

A first version of the Lloyd algorithm that is Differentially-Private is shown in [36, 118]; in this version, Laplacian noise is added to every iteration and the number of iterations is decided *a priori* so that the total of noise is fixed.

Formally, SuLQ framework uses the following support function:

$$SuLQ(q, S) = \sum_{i \in S} q(d_i) + N(0, R) \tag{9}$$

where $q$ is a query function, $S$ a dataset, which in our case is composed by a set of points, (that can be input for $q$), $N$ a random number sampled from a Normal distribution with average 0 and variance $R$. The resulting modification of the $k$-Means algorithm is described in algorithm 6.

When the number of iterations is too large, the resulting clustering could be too noisy, while, when it is too small the final solution can be less than optimal.

A critical factor in every $k$-Means algorithm is the choice of the initial seeds; a bad centroids initialization can lead to a result far from the global optimum. Probably the most usual choice, as discussed in [118], is to generate seeds randomly. However, when the generated points are too close each other the resulting clustering could be poor.

**Algorithm 6** SuLQ $k$-Means

```
Make a dataset partition $\{S_1, S_2, \dots, S_k\}$ associating every element $x_i$ with the nearest $\mu_j$
for all $1 \le j \le k$ do
    Approximate the number of points in each cluster:
        $n_j = SuLQ(q, S_j)$
    with
        $q(x_i) = \begin{cases} 1 & \text{se } x_i \in S_j \\ 0 & \text{otherwise} \end{cases}$
    Approximate the sum of points in each cluster
        $m_j = SuLQ(q, S_j)$
    with
        $q(x_i) = \begin{cases} x_i & \text{se } x_i \in S_j \\ 0 & \text{otherwise} \end{cases}$
    Update centroids, $\mu_j = \frac{m_j}{n_j}$
end for
```

### 7.3.2 Subsample-and-aggregate $k$-Means

In [126] the *subsample-and-aggregate* framework is proposed, initially used to obtain a differentially-private smoothed version of a given function $f$. The idea is to make a partition of data in $h$ blocks of size $n/h$; the function $f$ is then used on every block so that the results can be used as input of a DP function $B$, i.e.:

$$B(f(x_1, x_2, \dots, x_{\frac{n}{h}}), \dots, f(x_{(k-1)\frac{n}{h}+1}, \dots, x_n)). \tag{10}$$

In [126] it is shown that this composition is always DP, regardless of function $f$ structure. Eventually, $B(\cdot)$ result is input to a differentially private *aggregation* function. An implementation of this framework is shown in [121]. [126] and [142] show how to use this framework to obtain a DP version of $k$-Means algorithm that can be viewed in algorithm 7. A key parameter is now $h$; smaller $h$ implies bigger noise, while greater $h$ implies less block information in the cluster. In [121] is set to $h = N^{0.4}$ (with $|D| = N$) while in [147] is used $h = \frac{N}{3k}$.

The main problem in this type of algorithms is that it requires the input dataset to be well-separable, meaning that the data in each computed cluster are far from each other [150].

### 7.3.3 PrivGene $k$-Means

Another differentially-private $k$-Means version is proposed in [166], based on genetic algorithms.

In this version, centroids are seen as a vector of size $kd$. Initially, 200 of these vectors are randomly chosen (i.e. 200 sets of $k$ centroids) sampled from data space. By means

**Algorithm 7** [126] and [142] DP-$k$-Means

```
given a dataset $G$, make a partition in $h$ block $\{G_1, G_2, \ldots, G_h\}$
for every block $y$ do
    compute a set of $k$ centroids$\{\mu_1^y, \mu_2^y, \ldots, \mu_k^y\}$;
end for
for every group of centroids of a cluster $1 \leq j \leq k$ do
    compute a final centroid averaging on all the centroids and adding Laplacian noise in function of $\epsilon$.
end for
```

of an iterative process and exponential mechanism, a subset of $m$ vectors is used to generate a set of candidate vectors. Briefly, for every pair of vectors in the subset, a number $< u < kd$ is randomly chosen and every vector is divided in two parts, of size respectively $u$ and $kd - u$. Pairs of vectors in the second part are then swapped and noise is added to a random element in these vectors. The resulting set of vectors is eventually used as seeds for $k$-Means.

These steps are then repeated for a given number of iterations that changes as a function of $\epsilon$. However, the number of iterations is still an open problem: few iterations could lead to poor solutions, while too many iterations could affect the candidate choice.

### 7.3.4 Extended Uniform Grid $k$-Means

The approaches described so far have two big limitations:

- the output is only constituted by centroids (together with, only in some cases, also in the number of points for every cluster) without other informations about clusters;

- the privacy budget is consumed at every algorithm iteration, so that it is not possible to further analyse the data without risking to disclose private information (and then break differential privacy).

Non-iterative methods avoid these problems working, rather than on the original data, on an artificially generated version that preserves the essential features of the original data. Authors of [134] and, more recently, [147] propose the *Extended Uniform Grid $k$-Means*, a non-iterative approach to generate a synopsis of a dataset; the procedure is briefly described in algorithm 8. The released synopsis is represented as a set of cells each containing a number of points with added noise. However, the exact location of every point can not be directly recovered. Classic $k$-Means algorithm can then be used on these points.

This method needs the parameter $M$ that can be critical: too large a value of $M$ can lead to low counts, so that the noise has a greater impact, while too low a value of $M$ can lead to a greater area covered by every cell and then to a poor solution (assuming that points are not uniformly distributed). On the other hand, [134] suggests to set $M = \frac{N\epsilon}{10}$ for a 2-dimensions dataset while [147] shows that, when $M = \left(\frac{N\epsilon}{\sqrt{\frac{\alpha}{2\beta^2}}}\right)^{\frac{2d}{2+d}}$ (where $\alpha, \beta$ are input parameters), the quadratic error on the generated dataset is minimized.

**Algorithm 8** Extendend Uniform Grid $k$-Means

```
given a dataset $D$, make a partition in $M$ cells of equal size;
for all cell do
    counts how many points it contains;
end for
for all counter do
    add Laplacian noise.
end for
.
```

The choice of initial seeds is addressed by [147] in the following manner: given a radius $r$, $k$ random seeds are generated such that every center has a distance $\geq 2r$ from each other, and a distance $\geq r$ from the edges of the domain. Every time a generated point does not respect this condition, it is discarded and a new point is generated. If the number of generated wrong points is too large, then $r$ is too large, and therefore the process is repeated with a lower $r$. This process is data-independent and then it can not directly release any information about data.

[147] extends previous results, showing an hybrid approach between EUG$k$-Means and DPLloyd: first, EUG$k$-Means is executed on the data (using a first portion of $\epsilon$) and then the output centroids are used as initial seeds for DPLoyd. This strategy seems to perform well for high $\epsilon$ values, but for smaller values EUG$k$-Means outperforms such hybrid approach because DPLloyd may worsen the centroids.

### 7.3.5 High-Dimensional $k$-Means

Grid-based algorithms (as [147]) are too complex in time and space; authors of [20] show am efficient $k$-Means version in high-dimensional space, where we call high-dimensional space a space with dimension $d = \omega(polylog(|dataset|))$. This method seems better than others in terms of achieving the objective function in high-dimensional space, while in non-high-dimensional space [147] seems to achieve better results. The procedure can be summarized as described in Algorithm 9.

**Algorithm 9** High-Dimensional $k$-Means

```
for all $x_i \in D$ do
    Project $x_i$ in a space with dimension $p$, with $p < d$; this step is achieved;
end for
Divide recursively the space in hypercubes until every hypercube contains few points;
Take the set $C$ of the hypercube centers obtained in the previous step;
Use an ad hoc clustering algorithm (described in [20]) to find a good set of centers $T \subseteq C$;
for all $z_i \in T$ do
    Project $z_i$ in the original $d$-dimensional space.
end for
```

## 7.4 Discussion on Differential Privacy

Among all methods that we have studied, the approaches based on Differential Privacy (DP) actually seem the best compromise in terms of mathematical formalization and privacy preserving thanks to a strong mathematical foundation.

Whenever a new approach preserving DP is proposed, two things have to be provided:

- a prove that the proposed approach guarantees $\epsilon$-Differential privacy with a given $\epsilon$.
- a bound on the *loss*, that is the distance between the solution given by the proposed algorithm and the *optimal* solution, given by an algorithm optimizing the same target function without any limitation (as computational or privacy terms) that could influence the results.

These two points are provided in very different ways, that we briefly describe in the following of this section.

### 7.4.1 Proving Differential Privacy

This kind of proofs is usually given basically in two steps: the former proving that the "core" algorithm is differentially private, the latter that the remaining part of the algorithm is still differentially private; the first step is usually obtained adding noise sampled from specific distributions (e.g. Laplacian distribution) that are proven to make data that satisfy differential privacy; the latter is usually obtained with the help of support theorems as *composition theorems* that we will briefly state. In particular, we give three versions of the Serial composition theorem, relying on the subsequent refinements over the years and summarized in [96].

#### Parallel Composition Theorem

Given a set of algorithms $F = \{M_1, M_2, \ldots, M_l\}$ such that every algorithm is $\epsilon_i$-Differentially Private (with $1 \leq i \leq l$) on a disjoint subset of the whole dataset, $F$ results to be, on the whole, $\max\{\epsilon_i\}$-Differential privacy.

All in all, the Parallel Composition Theorem is used when every $M_i$ is used on a disjoint subset of the data.

#### First Serial Composition Theorem

A first composition theorem can be seen in [59],[60]; let $F$ be a set of algorithms $F = \{M_1, M_2, \ldots, M_l\}$; if every algorithm is applied to the whole dataset in sequence and every algorithm is $(\epsilon_i, \delta_i)$-differentially private (with $1 \leq i \leq l$), then $F$ is $(\sum_{i=1}^{l} \epsilon, \sum_{i=1}^{l} \delta)$-differentially private.

In other words, when several algorithms are used in sequence on the same dataset, the final differential privacy budget is given by the sum of the budgets of the single algorithms. Obviously, if all algorithms are $\epsilon$-differential privacy with $\epsilon$ constant, then $F$ will be $(l\epsilon)$-DP, with $l$ number of algorithms.

### Second Serial Composition Theorem (or k-Fold Composition Theorem)

This theorem needs the definition of **$\delta$-Approximate Max-Divergence**: given two discrete random variables $Y$ and $Z$ with output values in the same space $S$, we define the $\delta$-Approximate Max-Divergence as:

$$D_{\infty}^{\delta}(Y||Z) = \max_{S} \ln \frac{P(Y \in S) - \delta}{P(Z \in S)}. \tag{11}$$

Notice that a mechanism $M$ is $(\epsilon, \delta)$-Differentially Private $\iff$ for every pair of datasets $D_0$,$D_1$ that differ for one element, we have $D_{\infty}^{\delta}(D_0||D_1) \leq \epsilon$.

Now, we can begin to describe the $k$-fold composition Experiment used to state the $k$-fold Adaptive Composition theorem.

**$k$-fold composition Experiment.** Let $b \in \{0, 1\}$ and $A$ an Adversary; in every moment $i$, a database $D_b^i$ has data based on $i$ and $b$; for example, $D_0^1$ is a medical database with data of a specific person (that we call *Bob*), $D_1^1$ is the same database without *Bob*'s data, while $D_0^2$ could be a database with poll results and *Bob* vote, while $D_1^2$ is the same without *Bob*'s data. $A$ can choose:

- which query to make;
- the $(\epsilon, \delta)$-Differentially private algorithm to use;
- which database to consult using $b$.

$A$ can change these parameters using previous outputs. Outputs set $V^b = (y_1^b, y_2^b, \dots, y_k^b)$ given to $A$ is called *Adversary View*. The aim is to guarantee that $A$ can't use his View to know if *Bob*'s data are used to produce the results.

A differentially Private algorithms sequence with parameters $(\epsilon_1, \delta_1), (\epsilon_2, \delta_2), \dots, (\epsilon_k, \delta_k)$ is $(\epsilon_g, \delta_g)$-Differentially Private under *k-Fold Adaptive Composition* if, for every possible adversary $A$, we have

$$D_{\infty}^{\delta_g}(V^0, V^1) \leq \epsilon_g. \tag{12}$$

In [60], Theorem III.3, it is shown that the class of algorithms $(\epsilon, \delta)$-Differentially privacy satisfies $(\epsilon_r, k\delta + \delta_r)$-differential privacy with $\delta_r \in (0, 1]$ under *k-Fold Adaptive Composition*, with

$$\epsilon_r(\delta_r) = k\epsilon(\exp(\epsilon) - 1) + \epsilon\sqrt{2k \log \frac{1}{\delta_r}}$$

This result is an important improvement compared to the first composition theorem for $\epsilon \to 0$.

### Third Serial Composition Theorem

Another improvement to the composition theorem is discussed in [96]; in this work the theorem 3.3 shows that the class of algorithms $(\epsilon, \delta)$-Differentially private is $((k - 2i)\epsilon, 1 - (1 - \delta)^k(1 - \delta_i))$-differentially private under *k-fold adaptive composition* for every $i = \{0, 1, \dots, \lfloor k/2 \rfloor\}$ with

$$\delta_i = \frac{\sum_{l=0}^{i-1} \binom{k}{l} (e^{(k-l)\epsilon} - e^{(k-2i+l)\epsilon})}{(1 + e^{\epsilon})^k} \tag{13}$$

The same theorem can also be stated as follows: the class of algorithms $(\epsilon, \delta)$-differentially private satisfies $(\epsilon_r, 1-(1-\delta)^k(1-\delta_r))$-differential privacy with $\delta_r \in (0,1]$ under *k-fold adaptive composition* with

$$\epsilon_r = \min\left\{k\epsilon, \frac{(e^\epsilon - 1)\epsilon k}{e^\epsilon + 1} + \epsilon\sqrt{2k\log(e + \frac{\sqrt{k\epsilon^2}}{\delta_r})}, \frac{(e^\epsilon - 1)\epsilon k}{e^\epsilon + 1} + \epsilon\sqrt{2k\log\frac{1}{\delta_r}}\right\}. \quad (14)$$

### 7.4.2 Proving bounded *Loss*

The aim of a $k$-means algorithm is to find a set of $k$ clusters of points that minimize a given target function that usually is:

$$\sum_{j=1}^{k}\sum_{i=1}^{n}\min_j ||x_i - \mu_j||^2. \quad (15)$$

However, the exactly minimization of this function is an NP-Hard problem [55]. If we call the minimal value $OPT$, a good candidate solution must give a value of target function near to the $OPT$ value.

An example of *loss* measure for a clustering solution can be the *$(\alpha, \beta)$-Approximate Candidate Set* (used in [20]) that is defined as: given a set of points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$, a set $C \subseteq \mathbb{R}^d$ is said $(\alpha, \beta)$-Approximate Candidate Set if $\exists z_1, z_2, \ldots, z_k \in C$ such that the clustering target function using these points gives a result $\leq \alpha \cdot OPT + \beta$.

So, showing that the solutions given by an algorithm are contained in a $(\alpha, \beta)$-Approximate Candidate Set with bounded $\alpha, \beta$ values, can give an acceptable measure of the loss given by the proposed algorithm.

### 7.4.3 Possible issues with Differential Privacy

Differential-privacy approaches seem to be the actual State-of-Art for Privacy-preserving methods in fields as data analysis and clustering. However, the differential privacy framework suffers from a few possible problems on which the community is working.

#### Attacks

In addition to the attacks discussed in Section 7.2.3 for systems guarateeing more traditional forms of privacy, also systems based on Differential Privacy are subject to potential attacks showing that, even if strictly formalized, differential privacy is not free of dangers. Among the others, [61] accurately describes some types of differential privacy attacks; [78] and [104] show a set of possible attack techniques:

**Timing attacks:** a query can need a different amount of time depending on the presence or not of a specific data; an attacker could infer the real query result simply observing the query completion time;

**State attacks:** query results could influence the state of the machine (e.g., a system variable); if an attacker can access the database machine, he/she could infer results watching the value of this variable;

**Privacy budget attacks:** an attacker can build an ad-hoc query that "consumes" less or more privacy budget based on the presence of a data; the attacker could infer information observing the remaining budget.

**Inference attacks:** if data contains any kind of dependence between data, an attacker could use it to infer real query results.

As all these classes of attacks, with the only exception of the last one, depend on how much an attacker is able to access the framework machine, so they concern the execution environment rather than the formal definition of differential privacy. Although PINQ [118] is an example of framework that suffered of all these problems, it should be considered that it was one of the first differential privacy frameworks; moreover, [78] proposed Fuzz, a framework that seems to resolve some of these issues.

The *inference attack*, on the other hand, is more tricky to resolve: differential privacy imposes independence assumptions on data, and these assumptions do not always hold in real-world where users can have relationships between each other. Two possible solutions are given by [173] and [104] proposing different sensitivity measures ( *Correlated sensitivity* and *Dependent Sensitivity*, respectively) which try to calibrate noise's amount in presence of dependence between data. A problem with these approaches is that they need a quantification of the dependence degree, which depends on the real probabilistic models of the data. Therefore, a reliable estimation needs a complete knowledge about dependence relationship or data generation.

### Parameters

There could be issues with materially obtaining differential privacy; one example, as already said in Section 7.3, is computing parameters as Global or Local Sensitivity(GS or LS). However, [92] proposes a new Sensitivity measure (called *Elastic Sensitivity*) and shows that this new value is an upper bound for $LS$ and it is easier to compute respect to $LS$. Furthermore, there is a possible semantic misunderstanding in the privacy budget meaning: $\epsilon$ affects how much an item could influence the result, but nothing is said about the actual disclosure of the data. This could be confusing for non-technicals, including managers and users who must decide whether to concede their data. Such issue is also related to the more general problem for a company to choose a good value of $\epsilon$ based on their privacy policy; however, solutions for this last point have been proposed, for example by [103], which introduces the concept of *Differential Identifiability*, a model based on Differential Privacy, that tries to overcome the problem of the choice of privacy parameters by data publishers.

### Queries

Queries can be considered a critical issue in differential privacy for at least three reason:

**Noise:** As discussed in [139], the uncertainty in query answer is an intrinsic, but necessary, feature of differential privacy caused by noise.

**Type:** traditional frameworks (e.g. PINQ) allow to make just a subset of all possible queries available on databases; these could be enough for tasks as clustering, but not for more general data analysis. In recent years the scientific community is working on these limitations producing new works to overcome this problem; The State-of-Art, in our knowledge, is [92] that proposes a framework that enormously enlarges the number of possible queries that can be done on data, moreover it doesn't seem to require any change on existing databases.

**Number:** In real-world applications, the number of queries that can be materially performed is a crucial point in the differential privacy framework: if the number of queries is too large, the privacy budget needs to be divided into many pieces, and, consequently, a large amount of noise is required.

### 7.4.4 Final considerations

Literature gives a set of results about differential private clustering that we briefly report: [147] shows the comparison reported in Figure 3 between their results (in terms of inter-class variance and other methods on different datasets, as a function of their dimension $d$ and the number of clusters $k$. From these results, the method they propose outperforms all the others. On the other hand, the method proposed in [20] also shows a good behavior (Figure 4) respect to other frameworks, but, when the space dimension of data is low, the method of [147] still shows better results, as shown in 5. Another point that is often taken in consideration by authors (e.g. [20]) is the computational complexity of their algorithms that seems to be a critical issue especially when dataset size and dimensionality are large.

# 8 SPECIALising Company Systems

Considering the SPECIAL framework introduced in *D2.3 Transparency Framework V1*, which is depicted in *Figure* 6, there are two distinct high level connection points: (i) enhancing existing Line of Business (LOB) applications with transparency and compliance checking abilities; and (ii) supporting data protection aware business intelligence or data science. The question arises what is the intersection between the SPECIAL transparency and compliance platform and existing LOB applications. Given that the objective of SPECIAL is to help companies comply with the General Data Protection Regulation in terms of obtaining consent and providing transparency with respect to the processing of personal data, it is important to describe how companies could potentially benefit from the toolstack developed by SPECIAL. Unfortunately, this cannot simply be achieved in an Extract, Transform, Load (ETL) fashion as providing the necessary transparency and compliance checking will more than likely involve changes to the existing company infrastructure.

In the presented framework components that are coloured in green are assumed to exist already, while components in blue will be developed by SPECIAL and/or the know how to develop said components will be provided by SPECIAL. It is worth noting that the RDF symbol is used to denote RDF data, HDFS and Spark symbols are used to highlight big data and big data processing respectively, and a simple reasoning symbol is used to denote components that could potentially require some form of reasoning capabilities[29]. In this chapter, we highlight several considerations with respect to the

[29] The HDFS and Spark symbols do not signify a technology choice but are rather used to denote the need for big data storage and processing

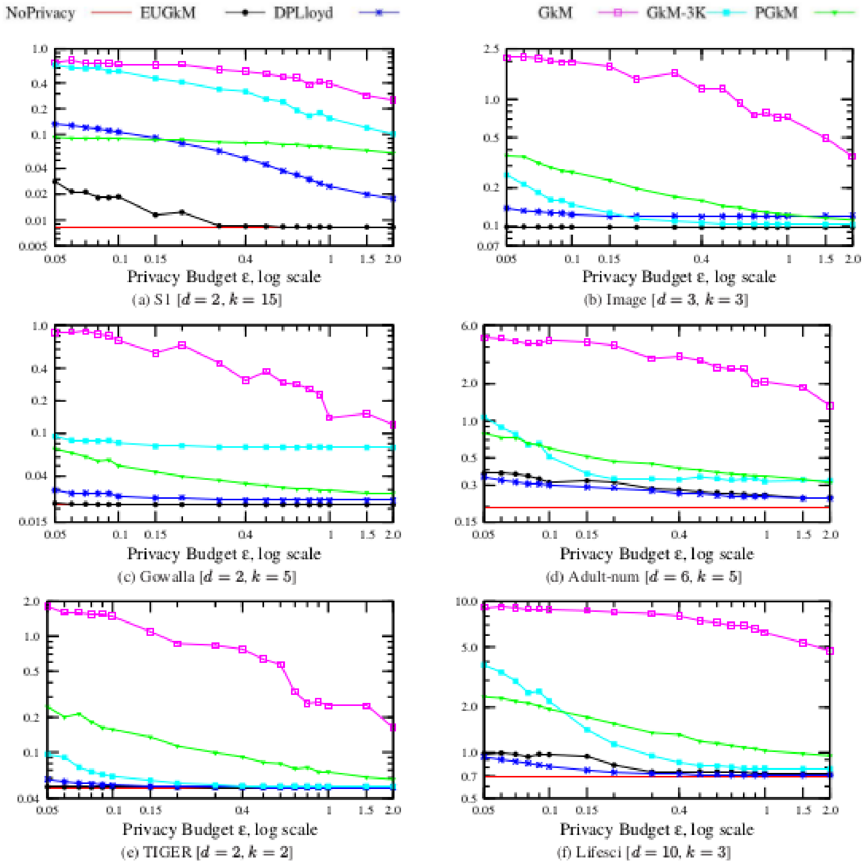


Figure 3: Comparison between: simple k-Means, EUGk-M [147], DPLloyd [36], G-kM [134], G-kM with fixed block size (GkM-3K [147]), PGk-M [166] - figure taken from [147]

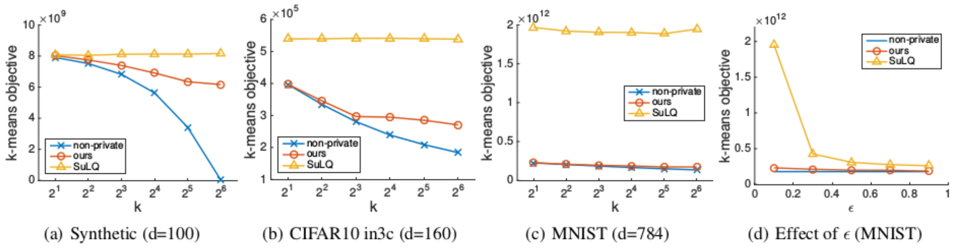
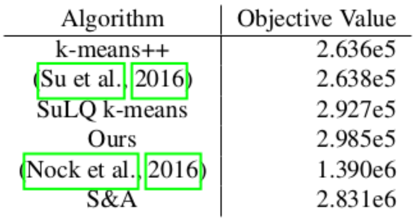


Figure 4: Comparison between SulQ [36] and [20] (in figure referred as "ours") - figure taken from [20]

intersection between existing company systems and the SPECIAL components.

## 8.1 LOB Application Transparency and Compliance Checking

Clearly there is a tight coupling between SPECIAL and existing LOB applications both in terms of usage policy specification and enforcement. Firstly, the data which is processed by the LOB application, needs to form part of the consent request that is pre-



| Algorithm | Objective Value |
|---|---|
| k-means++ | 2.636e5 |
| (Su et al., 2016) | 2.638e5 |
| SuLQ k-means | 2.927e5 |
| Ours | 2.985e5 |
| (Nock et al., 2016) | 1.390e6 |
| S&A | 2.831e6 |

Figure 5: $d = 3$, $k = 4$ (in figure [20] is referred as "ours") - table taken from supplementary material of [20]

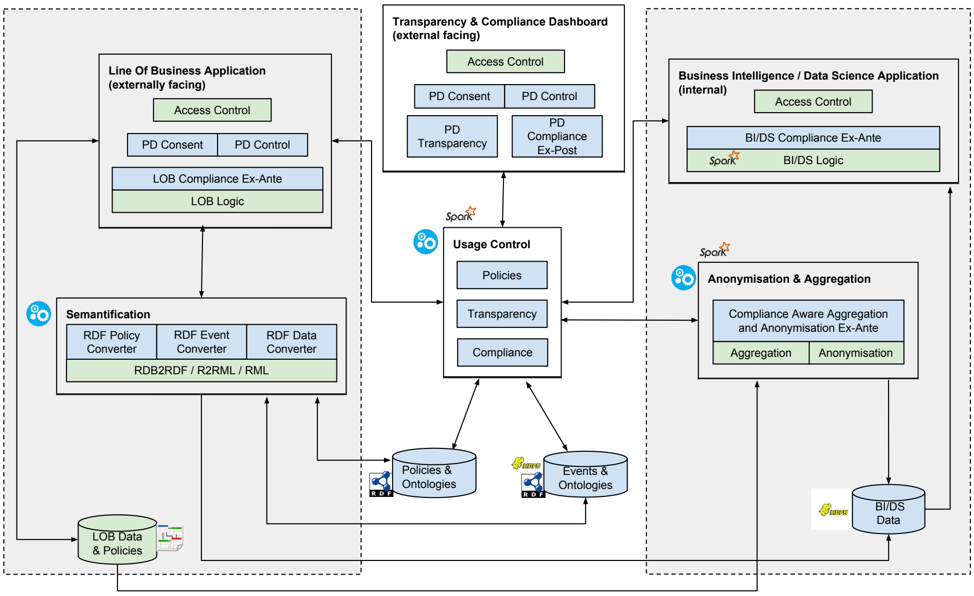


Figure 6: SPECIAL Landscape

sented to the user. Once the users provide their consent this information could be used to construct usage policies that could be used for transparency and compliance checking purposes.

In order to better understand the interplay between SPECIAL and existing company systems, in this section we provide some insights into the following tasks: (i) identifying personal data digital assets within a company; (ii) adding support for consent requests that are necessary in order to generate usage policies; (iii) providing transparency with respect to personal data processing; and (iv) enabling data protection aware data processing via compliance checking.

### 8.1.1 Personal Data Processing Inventory

Assuming that a company wishes to use the SPECIAL infrastructure to ensure that existing LOB applications comply with consent and transparency requirements stipulated in the GDPR, a necessary first step is to do some preliminary analysis in order to determine:

- What data is collected and which data points would be classified as personal data?
- 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?

Although the aforementioned questions seem relatively straight forward, for larger companies and in complex environments where data is often stored in multiple locations and also shared between applications this task could be quite time consuming. In such cases a systematic approach that is suitable for several business domains would be highly desirable. Considering the potential complexity of the task it would be highly beneficial if a set of guidelines or best practices that could be used by companies to audit their personal data processing practices were available. In the following we identify primary considerations with respect to performing a personal data processing inventory:

*Engaging domain experts:* As with any systems analysis project it is important to engage the help of both business and technical professionals who possess expert knowledge of the system(s) under analysis. For instance, business experts could provide background on the system and how it is used in practice, whereas technical experts would be able to provide technical details on the data sources, business logic, and current usage policies.

*Identification of systems that process personal data:* In order to determine what personal data is processed it is first necessary to identify the systems that process personal data, were these systems are located (i.e. on premise or in the cloud) and who has access to the data. Although in an ideal world the systems and the connections between them are documented in the form of system architectures in many cases such documentation is not available.

*Analysis of existing personal data processing:* Considering that the goal is to understand what the applications are doing as opposed to simply analysing the inputs and outputs this task may involve analysing documentation, workflows, code, and system logs, in order to determine how personal data is processed, if the system derives new data from this personal data, where the processed and/or derived data is stored, if the system enables the sharing of data inside the company or externally, and for how long is the data stored.

*Classification of personal data:* The classification step builds on the personal data and processing identification by examining the different categories of data and processing and by describing the data formats used and the type of metadata attached to the personal data (e.g. provenance, temporal, usage constraints). This step is a necessary precursor for developing / extending the SPECIAL vocabularies to cater for different types of applications and domains.

*Managing personal data assets:* In order to ensure that all information is kept up to date there is also the need for (collaborative) interfaces that can be used to manage data assets, taxonomies (data attributes, risks, applications, processing purposes, tags etc.) and data flows.

### 8.1.2 Consent Requests and Usage Policies

One of the expected outputs of a personal data processing inventory is a personal data catalog that can be used to develop SPECIAL consent requests (based on the processing performed by the system and the personal data required to support this processing) and usage policies (based on the consent provided by the data subject). Additional information on informed consent and policy models and languages can be found in *Chapter 5* and *Chapter 6* respectively.

Assuming that a company does not already have a means to request consent at the level of granularity needed to create usage policies based on the Minimum Core Model (MCM) presented in *Chapter 6* the company could develop consent and control connectors for their existing LOBs based on the consent user interfaces developed in SPECIAL or to re-engineer their existing user interfaces. Additional details on said interfaces and the results of our initial usability tests can be found in deliverable *D4.1 Transparency dashboard and control panel release V1* and *D4.2 Usability testing report V1*, respectively. Clearly, the content of the consent request and subsequently the usage policy would need to be tightly coupled to the data gathered by and the processing performed by the LOB application, thus the personal data processing inventory (discuss previously) would play a critical role in this context. In the following we provide some pointers with respect to obtaining consent:

*Consent retrieval.* The key objective is to create consent requests for the processing, which needs to be performed by the application. Here the various consent user interfaces developed by SPECIAL and/or the feedback obtained from the user studies could be used as a basis for developing connectors to LOBs or for re-engineering existing systems such that consent could be gathered from the user. Deliverable *D1.6 Legal requirements for a privacy-enhancing Big Data V2* exemplifies how this could potentially be done in a legally compliant way.

*Usage policies.* On receipt of the consent the system should automatically create the corresponding usage control policy. If a company wishes to use the SPECIAL transparency and compliance platform in order to ensure they are meeting their legal obligations under the GDPR, the usage policies will need to be represented using the SPECIAL policy language and vocabulary. Depending on the data format used by the LOB application the RDF Mapping language (RML) could be used to map data in heterogeneous structures and serializations to the RDF data model.

*Storage and indexing.* Although SPECIAL authorisations contain data, processing, purpose, recipient, storage attributes, it may be desirable from an analytical perspective to be able to retrieve usage policies based on the user, the governed data, the type of processing, where the data is stored or with whom it is shared, thus indexing of usage constraints based on each of these attribute would be advisable.

### 8.1.3 Transparent Data Processing

In *Chapter 3* we discussed the role of system logs when it comes to providing transparency with respect to personal data processing. Irrespective of where the log resides, the data that needs to be recorded in the log is dependent on what information is needed in order to automatically check compliance with both usage policies and relevant regulations. Our assumption is that existing LOB logging mechanisms could be recommissioned such that it was possible to check that all personal data processing performed by an application complies with the data subjects usage policy (i.e. ex-post compliance checking). In the following we provide some recommendations that could be used to guide this activity:

*Analysis of existing logging mechanisms.* Irrespective of the logging infrastructure employed by the company a necessary first step is to examine what additional attributes need to be recorded in order to satisfy personal data processing transparency and ex-post compliance checking. This could be done by comparing the schema of the existing application logs to that of the SPECIAL Policy Log presented in *Deliverable D2.3 Transparency Framework V1*, in order to identify if any additional attributes need to be recorded.

*Logging events.* In order to benefit from the SPECIAL transparency and compliance engine as described in *D3.2 Policy & events release* and *D4.1 Transparency dashboard and control panel release V1*, the LOB application logging mechanisms may need to be amended/extended such that it is possible for the additional attributes identified during the log structure analysis to be recorded by the application. The degree of re-engineering required will highly depend on the existing logging mechanisms employed by the company.

*Transforming events.* The SPECIAL transparency engine expects events to be represented using the SPECIAL Policy Log. Although the LOB application is not expected to serialize their event data as RDF, it is expected that there will be a one to one mapping between the information stored in the LOB application and the information that is necessary for verifying data processing adheres to the data subjects consent. Here the RDF Mapping language (RML) could potentially be used to map data in heterogeneous structures and serializations to the RDF data model.

*Interfacing to the SPECIAL engine.* The initial release of the SPECIAL platform, uses Kafka to consume logs and usages policies generated by existing LOB applications, represented as / transformed into the SPECIAL log vocabulary and policy language. The system takes as input an application log topic, a policies topic and a base ontology. The reasoning engine subsequently performs the compliance checking and outputs the results to a ex-post compliance topic.

### 8.1.4 Compliance Checking

In order to help companies to comply with the GDPR, in addition to providing data subjects with transparency with respect to the processing of their personal data it is also necessary to check that the processing is permitted in advance of performing the actual processing, akin to traditional access control enforcement (i.e. ex-ante compliance checking). In the case of ex-post compliance checking (as described in the previous section) relevant information is recorded in a log, in a manner that it is possible to for the SPECIAL compliance engine to automatically check compliance. Whereas, for ex-ante compliance checking the strategy may differ depending on the system and the needs of the business. Possible approaches discussed to date include using the log vocabulary to generate a processing request that is checked by the SPECIAL compliance engine and representing the business logic in the form of a company business policy that should be checked against the data subjects usage policy. In the following we examine the high level activities that should be performed when it comes to compliance checking:

*Analysis of existing business logic.* The first major release of the SPECIAL platform focused on providing transparency via ex-post compliance checking, the second major release of the SPECIAL platform will extend the platform with ex-ante compliance checking capabilities. Here again the personal data processing inventory will be needed in order to determine the most effective way of transforming business rules into RDF. Considering the tight coupling between ex-ante compliance checking and traditional access control one potential opportunity would be to hook into the existing access control mechanisms, alternatively it may be necessary to re-engineer the system.

*Transforming business logic into requests.* In order to benefit from the SPECIAL transparency and compliance platform the LOB application will need to generate a processing request that contains the attributes necessary to check compliance. Considering the type of processing varies depending on the domain, it is important that the SPECIAL vocabularies can easily be extended. Here again RML could potentially be used to map data in heterogeneous structures and serializations to the RDF data model, such that it is possible to benefit from the SPECIAL compliance checking mechanisms.

*Interfacing to the SPECIAL engine.* The initial release of the SPECIAL platform, uses Kafka to consume logs and policies represented using the SPECIAL log vocabulary and policy language. Considering that Kafka provides support of stream processing, it is particularly suitable for ex-ante compliance checking. In this instance processing requests could be submitted to a processing request topic. As per the transparency scenario the system would also need to submit usage policies to the policies topic, and take as input the extended base ontology. The reasoning engine will subsequently perform the ex-ante compliance checking and output the results to a ex-ante compliance topic.

## 8.2 Data Protection Aware Business Intelligence

One of the primary objectives of SPECIAL is to enable data protection aware personal data value processing. In this context the question arises, is it possible to perform business intelligence or data science activities over personal data in light of the new General Data Protection Regulation? Further more, is it possible to share personal data and related policies, between companies according to business relationships, while at the same time providing guarantees to data subjects that their usage preferences will be adhered to. In order to better understand the interplay between SPECIAL and existing company systems, in this section we provide some insights into the following tasks: (i) policy aware personal data analytics; and (ii) policy aware data sharing.

### 8.2.1 Policy Aware Personal Data Analytics

According to the GDPR the principles of data protection do not apply to anonymous information, *namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable.* Although SPECIAL focuses on demonstrating how companies can comply with the GDPR in terms of obtaining consent and providing transparency there is a partial overlap with anonymisation in the sense that helping companies and data subjects to understand the risk associated with anonymised data could actually be used as a means to obtain consent for business intelligence / data science activities. As such, in *Chapter* 7 this deliverable includes a comprehensive survey of the main privacy-preserving data mining methods found in the literature. In the following we examine the high level activities relating to the anonymisation or aggregation:

*Anonymisation and aggregation.* One of the challenges with existing anonymisation and aggregation strategies is the fact that in a big data environment it is difficult to truly ensure that it is not possible to identify a data subject. Clearly stronger anonymisation techniques give stronger personal data protection guarantees, however said approaches can destroy the utility of the data. The SPECIAL project is currently exploring machine learning techniques that carry out analytics while provably preserving privacy to a specified degree, e.g. by guaranteeing $\epsilon$ differential privacy for a given parameter $\epsilon$ (that is used to quantify the risk posed by releasing the analysis computed on personal data).

*Storage and indexing.* Although SPECIAL authorisations contain data, processing, purpose, recipient, storage attributes, it may be desirable from an analytical perspective to be able to retrieve usage policies based on the user, the governed data, the type of processing, where the data is stored or with whom it is shared, thus indexing of usage constraints based on each of these attribute would be advisable.

*Clustering according to policies.* Considering that SPECIAL enables companies to obtain very granular consent, it is possible that the data itself could be clustered according to usage constraints specified in policies. This would allow companies to maintain the utility of the data, while at the same time adhering to the data subjects usage policies. Here, the utility of the data when carved according to usage policies is still an open research question. In this context the SPECIAL project is exploring the combination of symbolic reasoning and machines learning techniques. Open research questions relate 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.

*Interfacing to the SPECIAL engine.* The initial release of the SPECIAL platform, uses Kafka to consume logs and policies represented using the SPECIAL log vocabulary and policy language. In the case of business intelligence Kafka could potentially be used as an intermediary between data sources from which features are extracted, the model building environment and the prediction environment. At present SPECIAL envisages two types of business intelligence services: (i) analysing data with parametric privacy guarantess, so that the data subject have more transparency with respect to the risks associated to the analysis they have consented to; and (ii) enabling companies to perform analytics over data that are constrained by usage policies. Thus inputs are expected to include both data, anonymisation parameters and policies.

### 8.2.2 Data Protection Aware Personal Data Sharing

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). In the following we examine the high level activities relating to the data protection aware personal data sharing:

*Fair exchange.* When it comes to data sharing, there is the option to use fair exchange protocols to guarantee that the operation is completed (e.g. data are transferred). A comprehensive survey of the predominant fair-exchange protocols and their properties is presented in *Chapter* 4. A modified version of Micali's protocol, designed for large data transfers, is described in D2.3.

*Sticky policies.* The term sticky policy is used to refer to policies that are tightly coupled to personal data. Considering that SPECIAL looks to support policy aware data sharing, cryptographic mechanisms could be used to strongly associate policies with the data. One of the challenges with respect to existing approaches is that there is a need for a trusted third party to ensure that obligations specified in the policy are fulfilled. 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. In this context the SPECIAL project is exploring alternative trust mechanisms.

*Interfacing to the SPECIAL engine.* The initial release of the SPECIAL platform, uses Kafka to consume logs and policies represented using the SPECIAL log vocabulary and policy language. Considering, the fact that Kafka is the big data platform of choice for many companies when it comes to big data processing, is a very positive indicator of our choice of platform.

## 8.3 Summary

The goal of this chapter was to identify the intersection between the SPECIAL transparency and compliance platform and existing personal data processing in corporate environments in the context of both Line of Business and Business Intelligence / Data Science applications. While other chapters both identified requirements and examined the state of the art, considering the applied nature of the tasks described in this chapter it focuses solely on requirements. The aim is to develop a set of guidelines, best practices, lessons learned, etc... as we progress through the project, which will feed into *D5.4 Development of Guidelines*.

# 9 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 is fairly extensive at this stage, and will form a roadmap for the research carried out in workpackage two. Still, considering the agile nature of the project, parts of this report may be further 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 checkers.

# 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] M. Abadi, N. Glew, B. Horne, and B. Pinkas. Certified email with a light on-line trusted third party: design and implementation. In *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii*, pages 387–395, 2002. doi: 10.1145/511446.511497. URL `http://doi.acm.org/10.1145/511446.511497`.

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

[4] 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.

[5] D. Agrawal and C. C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 247–255. ACM, 2001.

[6] 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`.

[7] N. Asokan. *Fairness in electronic commerce.* PhD thesis, University of Waterloo, 1998.

[8] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for multiparty fair exchange. Technical Report 90840, IBM Research, December 1996.

[9] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *CCS '97, Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, April 1-4, 1997.*, pages 7–17, 1997. doi: 10.1145/266420.266426. URL `http://doi.acm.org/10.1145/266420.266426`.

[10] N. Asokan, B. Baum-Waidner, M. Schunter, and M. Waidner. Optimistic synchronous multi-party contract signing. Technical Report RZ 3089, IBM Zurich Research Lab, December 1998.

[11] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Security and Privacy - 1998 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 3-6, 1998, Proceedings*, pages 86–99, 1998. doi: 10.1109/SECPRI.1998.674826. URL `http://dx.doi.org/10.1109/SECPRI.1998.674826`.

[12] N. Asokan, V. Shoup, and M. Waidner. *Optimistic fair exchange of digital signatures*, pages 591–606. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-69795-4. doi: 10.1007/BFb0054156. URL `http://dx.doi.org/10.1007/BFb0054156`.

[13] G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *CCS '99, Proceedings of the 6th ACM Conference on Computer and Communications Security, Singapore, November 1-4, 1999.*, pages 138–146, 1999. doi: 10.1145/319709.319728. URL `http://doi.acm.org/10.1145/319709.319728`.

[14] G. Ateniese, B. de Medeiros, and M. T. Goodrich. TRICERT: A distributed certified e-mail scheme. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2001, San Diego, California, USA*, 2001. URL `http://www.isoc.org/isoc/conferences/ndss/01/2001/papers/ateniese.pdf`.

[15] G. Avoine and S. Vaudenay. *Optimistic Fair Exchange Based on Publicly Verifiable Secret Sharing*, pages 74–85. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. ISBN 978-3-540-27800-9. doi: 10.1007/978-3-540-27800-9_7. URL `http://dx.doi.org/10.1007/978-3-540-27800-9_7`.

[16] G. Avoine, F. Gärtner, R. Guerraoui, and M. Vukolić. *Gracefully Degrading Fair Exchange with Security Modules*, pages 55–71. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005. ISBN 978-3-540-32019-7. doi: 10.1007/11408901_5. URL `http://dx.doi.org/10.1007/11408901_5`.

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

[18] 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`.

[19] A. Bahreman and D. Tygar. Certified electronic mail. In *Symposium on Network and Distributed Systems Security*, pages 3–19, 1994.

[20] M.-F. Balcan, T. Dick, Y. Liang, W. Mou, and H. Zhang. Differentially private clustering in high-dimensional euclidean spaces. In *International Conference on Machine Learning*, pages 322–331, 2017.

[21] W. Banasik, S. Dziembowski, and D. Malinowski. Efficient zero-knowledge contingent payments in cryptocurrencies without scripts. In I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, editors, *Prooceedings of 21st European Symposium on Research in Computer Security (ESORICS 2016)*, pages 261–280. Springer International Publishing, 2016. ISBN 978-3-319-45741-3. doi: 10.1007/978-3-319-45741-3_14. URL http://dx.doi.org/10.1007/978-3-319-45741-3_14.

[22] F. Bao, R. H. Deng, and W. Mao. Efficient and practical fair exchange protocols with off-line TTP. In *Security and Privacy - 1998 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 3-6, 1998, Proceedings*, pages 77–85, 1998. doi: 10.1109/SECPRI.1998.674825. URL http://dx.doi.org/10.1109/SECPRI.1998.674825.

[23] F. Bao, R. Deng, K. Q. Nguyen, and V. Varadharajan. Multi-party fair exchange with an off-line trusted neutral party. In *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*, pages 858–862, 1999. doi: 10.1109/DEXA.1999.795294.

[24] 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.

[25] 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.

[26] M. Bellare, A. Boldyreva, and S. Micali. *Public-Key Encryption in a Multi-user Setting: Security Proofs and Improvements*, pages 259–274. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-45539-4. doi: 10.1007/3-540-45539-6_18. URL http://dx.doi.org/10.1007/3-540-45539-6_18.

[27] M. Ben-Or, O. Goldreich, S. Micali, and R. L. Rivest. A fair protocol for signing contracts. *IEEE Trans. Information Theory*, 36(1):40–46, 1990. doi: 10.1109/18.50372. URL http://dx.doi.org/10.1109/18.50372.

[28] E. Benda, H. Simon, K. Hesse, D. Katzenstein, G. Niemeyer, H. Heußner, and J. F. Henschel. Bverfge 65, 1. 65:1–71, 1983.

[29] I. Bentov and R. Kumaresan. How to use bitcoin to design fair protocols. In *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, pages 421–439, 2014. doi: 10.1007/978-3-662-44381-1_24. URL https://doi.org/10.1007/978-3-662-44381-1_24.

[30] 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.

[31] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 245–250, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X. doi: 10.1145/502512.502546. URL http://doi.acm.org/10.1145/502512.502546.

[32] Bitcoin Wiki. Zero Knowledge Contingent Payment. https://en.bitcoin.it/wiki/Zero_Knowledge_Contingent_Payment, 2016.

[33] Bitcoin Wiki. Contact. https://en.bitcoin.it/wiki/Contract, 2017.

[34] Bitcoin Wiki. Bitcoin wiki. https://en.bitcoin.it/wiki/Main_Page, 2018.

[35] Bitcoin Wiki. Script. https://en.bitcoin.it/wiki/Script, 2018.

[36] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The sulq framework. In *Proceedings of the Twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '05, pages 128–138, New York, NY, USA, 2005. ACM. ISBN 1-59593-062-0. doi: 10.1145/1065167.1065184. URL http://doi.acm.org/10.1145/1065167.1065184.

[37] M. Blum. How to exchange (secret) keys. *ACM Trans. Comput. Syst.*, 1(2):175–193, May 1983. ISSN 0734-2071. doi: 10.1145/357360.357368. URL http://doi.acm.org/10.1145/357360.357368.

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

[39] 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.

[40] 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.

[41] 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.

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

[43] 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.

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

[45] 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.

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

[47] I. Budin-Ljøsne, H. J. Teare, J. Kaye, S. Beck, H. B. Bentzen, L. Caenazzo, C. Collett, F. D'Abramo, 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.

[48] 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.

[49] M. Campanelli, R. Gennaro, S. Goldfeder, and L. Nizzardo. Zero-knowledge contingent payments revisited: Attacks and payments for services. *IACR Cryptology ePrint Archive*, 2017:566, 2017. URL http://eprint.iacr.org/2017/566.

[50] T. Coffey and P. Saidha. Non-repudiation with mandatory proof of receipt. *SIGCOMM Comput. Commun. Rev.*, 26(1):6–17, Jan. 1996. ISSN 0146-4833. doi: 10.1145/232335.232338. URL http://doi.acm.org/10.1145/232335.232338.

[51] 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.

[52] B. Cox, J. D. Tygar, and M. Sirbu. Netbill security and transaction protocol. In *Proceedings of the 1st Conference on USENIX Workshop on Electronic Commerce - Volume 1*, WOEC'95, pages 6–6, Berkeley, CA, USA, 1995. USENIX Association. URL http://dl.acm.org/citation.cfm?id=1267185.1267191.

[53] 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.

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

[55] S. Dasgupta. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, San Diego, 2008.

[56] 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.

[57] R. H. Deng, L. Gong, A. A. Lazar, and W. Wang. Practical protocols for certified electronic mail. *Journal of Network and Systems Management*, 4(3):279–297, 1996. ISSN 1573-7705. doi: 10.1007/BF02139147. URL http://dx.doi.org/10.1007/BF02139147.

[58] Y. Dodis, P. J. Lee, and D. H. Yum. *Optimistic Fair Exchange in a Multi-user Setting*, pages 118–133. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-71677-8. doi: 10.1007/978-3-540-71677-8_9. URL http://dx.doi.org/10.1007/978-3-540-71677-8_9.

[59] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32731-2, 978-3-540-32731-8. doi: 10.1007/11681878_14. URL http://dx.doi.org/10.1007/11681878_14.

[60] C. Dwork, G. N. Rothblum, and S. Vadhan. Boosting and differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 51–60. IEEE, 2010.

[61] C. Dwork, A. Smith, T. Steinke, and J. Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 2017.

[62] Ethereum. The DAO Vulnerability. https://blog.ethereum.org/2016/06/17/critical-update-re-dao-vulnerability/, 2016.

[63] Ethereum. Ethereum Project. https://www.ethereum.org, 2018.

[64] 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.

[65] S. Even. A protocol for signing contracts. *SIGACT News*, 15(1):34–39, Jan. 1983. ISSN 0163-5700. doi: 10.1145/1008908.1008913. URL http://doi.acm.org/10.1145/1008908.1008913.

[66] S. Even and Y. Yacobi. Relations among public key signature systems. Technical Report 175, Comput. Sci. Dept., Technion, Haifa, Israel, 1980.

[67] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[68] 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.

[69] 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.

[70] M. Franklin and G. Tsudik. *Secure group barter: Multi-party fair exchange with semi-trusted neutral parties*, pages 90–102. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-53918-6. doi: 10.1007/BFb0055475. URL `http://dx.doi.org/10.1007/BFb0055475`.

[71] M. K. Franklin and M. K. Reiter. Fair exchange with a semi-trusted third party (extended abstract). In *CCS '97, Proceedings of the 4th ACM Conference on Computer and Communications Security, Zurich, Switzerland, April 1-4, 1997.*, pages 1–5, 1997. doi: 10.1145/266420.266424. URL `http://doi.acm.org/10.1145/266420.266424`.

[72] S. Galbraith, J. Malone-Lee, and N. Smart. Public key signatures in the multi-user setting. *Information Processing Letters*, 83(5):263 – 266, 2002. ISSN 0020-0190. doi: http://dx.doi.org/10.1016/S0020-0190(01)00338-6. URL `http://www.sciencedirect.com/science/article/pii/S0020019001003386`.

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

[74] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

[75] S. Guo and X. Wu. Deriving private information from arbitrarily projected data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 84–95. Springer, 2007.

[76] 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.

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

[78] A. Haeberlen, B. C. Pierce, and A. Narayan. Differential privacy under fire. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 33–33, Berkeley, CA, USA, 2011. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=2028067.2028100`.

[79] 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.

[80] 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.

[81] 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/`.

[82] 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.

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

[84] 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.

[85] Q. Huang, G. Yang, D. S. Wong, and W. Susilo. *Ambiguous Optimistic Fair Exchange*, pages 74–89. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-89255-7. doi: 10.1007/978-3-540-89255-7_6. URL `http://dx.doi.org/10.1007/978-3-540-89255-7_6`.

[86] Q. Huang, D. S. Wong, and W. Susilo. *P2OFE: Privacy-Preserving Optimistic Fair Exchange of Digital Signatures*, pages 367–384. Springer International Publishing, Cham, 2014. ISBN 978-3-319-04852-9. doi: 10.1007/978-3-319-04852-9_19. URL `http://dx.doi.org/10.1007/978-3-319-04852-9_19`.

[87] Q. Huang, G. Yang, D. S. Wong, and W. Susilo. Ambiguous optimistic fair exchange: Definition and constructions. *Theoretical Computer Science*, 562:177 – 193, 2015. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/j.tcs.2014.09.043. URL `//www.sciencedirect.com/science/article/pii/S0304397514007336`.

[88] Z. Huang, W. Du, and B. Chen. Deriving private information from randomized data. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 37–48, New York, NY, USA, 2005. ACM. ISBN 1-59593-060-4. doi: 10.1145/1066157.1066163. URL `http://doi.acm.org/10.1145/1066157.1066163`.

[89] Hyperledger. Hyperledger home page. `https://www.hyperledger.org`.

[90] 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.

[91] S. Jha, L. Kruger, and P. McDaniel. Privacy preserving clustering. In S. de Capitani di Vimercati, P. Syverson, and D. Gollmann, editors, *Computer Security – ESORICS 2005*, pages 397–417, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31981-8.

[92] N. M. Johnson, J. P. Near, and D. X. Song. Practical differential privacy for SQL queries using elastic sensitivity. *CoRR*, abs/1706.09479, 2017. URL `http://arxiv.org/abs/1706.09479`.

[93] A. Juels and B. S. Kaliski, Jr. Pors: Proofs of retrievability for large files. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 584–597, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-703-2. doi: 10.1145/1315245.1315317. URL `http://doi.acm.org/10.1145/1315245.1315317`.

[94] 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.

[95] 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.

[96] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017.

[97] 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.

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

[99] S. Kremer. *Formal Analysis of Optimistic Fair Exchange Protocols.* PhD thesis, Université Libre de Bruxelles, 2004.

[100] S. Kremer and O. Markowitch. Fair multi-party non-repudiation protocols. *International Journal of Information Security*, 1(4):223–235, 2003. ISSN 1615-5270. doi: 10.1007/s10207-003-0019-3. URL `http://dx.doi.org/10.1007/s10207-003-0019-3`.

[101] A. Küpçü and A. Lysyanskaya. *Optimistic Fair Exchange with Multiple Arbiters*, pages 488–507. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-15497-3. doi: 10.1007/978-3-642-15497-3_30. URL `http://dx.doi.org/10.1007/978-3-642-15497-3_30`.

[102] A. Küpçü and A. Lysyanskaya. Usable optimistic fair exchange. *Computer Networks*, 56(1):50 – 63, 2012. ISSN 1389-1286. doi: http://dx.doi.org/10.1016/j.comnet.2011.08.005. URL `//www.sciencedirect.com/science/article/pii/S138912861100301X`.

[103] J. Lee and C. Clifton. Differential identifiability. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1041–1049, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339695. URL `http://doi.acm.org/10.1145/2339530.2339695`.

[104] C. Liu, S. Chakraborty, and P. Mittal. Dependence makes you vulnerable: Differential privacy under dependent tuples. In *Proc. of the Network and Distributed System Security Symposium 2016*, San Diego, California,USA, 2016.

[105] K. Liu, C. Giannella, and H. Kargupta. An attacker's view of distance preserving maps for privacy preserving data mining. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 297–308. Springer, 2006.

[106] K. Liu, C. Giannella, and H. Kargupta. A survey of attack techniques on privacy-preserving data perturbation methods. In *Privacy-Preserving Data Mining*, pages 359–381. Springer, 2008.

[107] S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, March 1982. ISSN 0018-9448. doi: 10.1109/TIT.1982.1056489.

[108] 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.

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

[110] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[111] D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay—a secure two-party computation system. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*, SSYM'04, pages 20–20, Berkeley, CA, USA, 2004. USENIX Association. URL `http://dl.acm.org/citation.cfm?id=1251375.1251395`.

[112] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008. ISBN 0521865719, 9780521865715.

[113] O. Markowitch and S. Kremer. A multi-party optimistic non-repudiation protocol. In *Information Security and Cryptology - ICISC 2000, Third International Conference, Seoul, Korea, December 8-9, 2000, Proceedings*, pages 109–122, 2000. doi: 10.1007/3-540-45247-8_9. URL `http://dx.doi.org/10.1007/3-540-45247-8_9`.

[114] O. Markowitch and Y. Roggeman. Probabilistic non-repudiation without trusted third party. Technical report, In Second Conference on Security in Communication Networks 1999, Amalfi, Italy, September 1999., 1999.

[115] O. Markowitch and S. Saeednia. Optimistic fair exchange with transparent signature recovery. In *Financial Cryptography, 5th International Conference, FC 2001, Grand Cayman, British West Indies, February 19-22, 2002, Proceedings*, pages 329–340, 2001. doi: 10.1007/3-540-46088-8_26. URL `http://dx.doi.org/10.1007/3-540-46088-8_26`.

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

[117] 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`.

[118] F. D. McSherry. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*, SIGMOD '09, pages 19–30, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-551-2. doi: 10.1145/1559845.1559850. URL `http://doi.acm.org/10.1145/1559845.1559850`.

[119] F. Meskine and S. N. Bahloul. Privacy preserving k-means clustering: a survey research. *International Arab Journal of Information Technology*, 9(2):194–200, 2012.

[120] S. Micali. Certified e-mail with invisible post offices. Invited presentation at RSA '97 conference, 1997.

[121] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler. Gupt: Privacy preserving data analysis made easy. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pages 349–360, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1247-9. doi: 10.1145/2213836.2213876. URL `http://doi.acm.org/10.1145/2213836.2213876`.

[122] 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.

[123] M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, STOC '99, pages 245–254, New York, NY, USA, 1999. ACM. ISBN 1-58113-067-8. doi: 10.1145/301250.301312. URL `http://doi.acm.org/10.1145/301250.301312`.

[124] A. Narayanan and V. Shmatikov. How to break anonymity of the netflix prize dataset. *CoRR*, abs/cs/0610105, 2006. URL `http://arxiv.org/abs/cs/0610105`.

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

[126] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 75–84, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-631-8. doi: 10.1145/1250790.1250803. URL `http://doi.acm.org/10.1145/1250790.1250803`.

[127] J. A. Onieva, J. Zhou, M. Carbonell, and J. Lopez. *A Multi-Party Non-Repudiation Protocol for Exchange of Different Messages*, pages 37–48.

Springer US, Boston, MA, 2003. ISBN 978-0-387-35691-4. doi: 10.1007/978-0-387-35691-4_4. URL http://dx.doi.org/10.1007/978-0-387-35691-4_4.

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

[129] 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.

[130] 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.

[131] 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.

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

[133] 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.

[134] W. Qardaji, W. Yang, and N. Li. Differentially private grids for geospatial data. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 757–768, April 2013. doi: 10.1109/ICDE.2013.6544872.

[135] M. O. Rabin. Transaction protection by beacons. *Journal of Computer and System Sciences*, 27(2):256 – 267, 1983. ISSN 0022-0000. doi: http://dx.doi.org/10.1016/0022-0000(83)90042-9. URL http://www.sciencedirect.com/science/article/pii/0022000083900429.

[136] 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.

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

[138] 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.

[139] R. Sarathy and K. Muralidhar. Evaluating laplace noise addition to satisfy differential privacy for numeric data. *Trans. Data Privacy*, 4(1):1–17, 2011.

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

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

[142] A. Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 813–822, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0691-1. doi: 10.1145/1993636.1993743. URL `http://doi.acm.org/10.1145/1993636.1993743`.

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

[144] 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`.

[145] 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.

[146] 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.

[147] D. Su, J. Cao, N. Li, E. Bertino, and H. Jin. Differentially private k-means clustering. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, CODASPY '16, pages 26–37, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-3935-3. doi: 10.1145/2857705.2857708. URL `http://doi.acm.org/10.1145/2857705.2857708`.

[148] 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`.

[149] P. Syverson. Weakly secret bit commitment: applications to lotteries and fair exchange. In *Proceedings. 11th IEEE Computer Security Foundations Workshop*, pages 2–13, Jun 1998. doi: 10.1109/CSFW.1998.683149.

[150] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005. ISBN 0321321367.

[151] T. Tedrick. *How to Exchange Half a Bit*, pages 147–151. Springer US, Boston, MA, 1984. ISBN 978-1-4684-4730-9. doi: 10.1007/978-1-4684-4730-9_13. URL http://dx.doi.org/10.1007/978-1-4684-4730-9_13.

[152] T. Tedrick. Fair exchange of secrets. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 434–438, New York, NY, USA, 1985. Springer-Verlag New York, Inc. ISBN 0-387-15658-5. URL http://dl.acm.org/citation.cfm?id=19478.19512.

[153] Ubiq. Ubiq. https://ubiqsmart.com.

[154] 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.

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

[156] 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.

[157] H. Vogt. *Asynchronous Optimistic Fair Exchange Based on Revocable Items*, pages 208–222. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-45126-6. doi: 10.1007/978-3-540-45126-6_15. URL http://dx.doi.org/10.1007/978-3-540-45126-6_15.

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

[159] 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.

[160] Y. Wang, M. H. Au, and W. Susilo. Perfect ambiguous optimistic fair exchange. In *Information and Communications Security - 14th International Conference, ICICS 2012, Hong Kong, China, October 29-31, 2012. Proceedings*, pages 142–153, 2012. doi: 10.1007/978-3-642-34129-8_13. URL http://dx.doi.org/10.1007/978-3-642-34129-8_13.

[161] *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.

[162] 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.

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

[164] 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.

[165] 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.

[166] J. Zhang, X. Xiao, Y. Yang, Z. Zhang, and M. Winslett. Privgene: Differentially private model fitting using genetic algorithms. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 665–676, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2037-5. doi: 10.1145/2463676.2465330. URL http://doi.acm.org/10.1145/2463676.2465330.

[167] N. Zhang and Q. Shi. Achieving non-repudiation of receipt. *Comput. J.*, 39(10): 844–853, 1996. doi: 10.1093/comjnl/39.10.844. URL http://dx.doi.org/10.1093/comjnl/39.10.844.

[168] J. Zhou and D. Gollmann. Certified electronic mail. In *Computer Security - ESORICS 96, 4th European Symposium on Research in Computer Security, Rome, Italy, September 25-27, 1996, Proceedings*, pages 160–171, 1996. doi: 10.1007/3-540-61770-1_35. URL http://dx.doi.org/10.1007/3-540-61770-1_35.

[169] J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *1996 IEEE Symposium on Security and Privacy, May 6-8, 1996, Oakland, CA, USA*, pages 55–61, 1996. doi: 10.1109/SECPRI.1996.502669. URL http://dx.doi.org/10.1109/SECPRI.1996.502669.

[170] J. Zhou and D. Gollmann. An efficient non-repudiation protocol. In *10th Computer Security Foundations Workshop (CSFW '97), June 10-12, 1997, Rockport, Massachusetts, USA*, pages 126–132, 1997. doi: 10.1109/CSFW.1997.596801. URL http://dx.doi.org/10.1109/CSFW.1997.596801.

[171] J. Zhou, R. Deng, and F. Bao. *Evolution of Fair Non-repudiation with TTP*, pages 258–269. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-540-48970-2. doi: 10.1007/3-540-48970-3_21. URL http://dx.doi.org/10.1007/3-540-48970-3_21.

[172] H. Zhu, W. Susilo, and Y. Mu. *Multi-party Stand-Alone and Setup-Free Verifiably Committed Signatures*, pages 134–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-71677-8. doi: 10.1007/978-3-540-71677-8_10. URL http://dx.doi.org/10.1007/978-3-540-71677-8_10.

[173] T. Zhu, P. Xiong, G. Li, and W. Zhou. Correlated differential privacy: Hiding information in non-iid data set. *IEEE Transactions on Information Forensics*

*and Security*, 10(2):229–242, Feb 2015. ISSN 1556-6013. doi: 10.1109/TIFS.2014.2368363.

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