SPECIAL

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

Deliverable 1.8

Technical Requirements V2

Document version: 1.0

SPECIAL DELIVERABLE

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

Mrs Jessica Michel Assoumou t: +33 4 97 15 53 06 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:	M1-M17
Deliverable number:	D1.8
Deliverable title	Technical Requirements V2
Contractual Date of Delivery:	31/05/2018
Actual Date of Delivery:	31/05/2018
Editor (s):	Bert Van Nuffelen (TF)
Author (s):	Bert Van Nuffelen, Uroš Milošević, Wouter Dullaert (TF)
Reviewer (s):	Rigo Wenning (ERCIM), Javier D. Fernández (WU)
Participant(s):	TF, WU, ERCIM, TR, TLabs, PROXIMUS
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):	32

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.

Table of Contents

1 Introduction

This document reports on the technical requirements and challenges for the SPECIAL platform and represents an update over D1.4 [2] based on the other relevant deliverable updates, namely:

- Deliverable 1.5 [3] describing the updated use-case scenarios,
- Deliverable 1.6 [4] describing the more detailed legal context and analysis of the use-case scenarios,
- Deliverable 3.2 [15], the second ("policy and events") release of the SPECIAL platform, and
- Deliverable 4.1 [17], the first release of the SPECIAL transparency dashboard and control panel.

This deliverable presents an overarching technical perspective of the SPECIAL platform. Based on the findings described in WP2 (D2.1 [6], D2.2 [7], D2.3 [8], D2.4 [9]) and implemented in D3.2, the SPECIAL ecosystem has evolved significantly beyond the initial version delivered in D3.1 [14].

The objective of this deliverable is to facilitate the ongoing and upcoming development and research work by the consortium. The deliverable forms a pair with Deliverable 1.7 [5] which provides an update over the state-of-the art analysis on consent management, policy language and transparency, first presented in D1.3 [1]. This deliverable elaborates the software architectural perspective further and, together, they will feed into an improved project roadmap.

Considering the iterative and agile nature of the project, this deliverable is not meant to serve as a complete list of requirements, but rather as a summary of our current analysis of the technical considerations that will be updated regularly as the project advances. Other findings, namely, the results of research in WP2 (D2.5 [10], D2.6 [11], D2.7 [12], D2.8 [13]) and scalability and robustness testing in WP3 (D3.3 [16]), the feedback received from usability testing in WP4 (D4.2 [18]) and public penetration/hacking challenges in WP5 (D5.2 [21], D5.4 [22]), are all likely to introduce new requirements. Nevertheless, this document aims to describe the key stakeholders and the *essential* set of interactions with the platform (as user stories).

Additionally, we detail the to-be applied approach for privacy threat assessment and the to-be applied risk mitigation strategies.

2 Use Cases and the SPECIAL approach

The SPECIAL project is motivated by the need for simplified personal data management that complies with the General Data Protection Regulation (GDPR). Three use-cases partners, two from the telco industry, Proximus and T-Labs, and one active in financial data services industry, Thomson Reuters Limited, have described their ideas on value adding services in D1.5 *Use Case Scenarios V2*.[1] These business objectives can only be realised if the personal data required to fuel the services is properly managed.

The GDPR grants businesses the ability to create added value from data, including data of personal nature, provided the data subjects (from whom personal data is collected and processed) are given control of their own personal data. The GDPR also states that control over the usage of personal data implies that the purposes for which the data are being acquired are understandable, permission to use the data is obtained in a comprehensive way and that the actual usage is verifiable. For SPECIAL, the control takes the form of **consent** and policy data management to capture the data subject's permissions for personal data processing and sharing, data management of the data usage traces to provide **transparency** on the usage and **compliance** mechanisms to guarantee and prove that the usage is in accordance with the given permissions and the legislation.

Based on the use-case descriptions and the additional provided background information, this deliverable presents an overarching SPECIAL data processing ecosystem with integrated support for consent, transparency and compliance. Each use case corresponds to an instance reusing common parts but differentiating on the used data and the service being implemented.

SPECIAL technical objective is to realise consent, transparency and compliance mechanisms for big data processing. Therefore, the service aspect defined by the use case instances will only be implemented to the level they can be used to demonstrate the consent, transparency and compliance mechanisms.

To obtain the desired personal data management and processing, SPECIAL defines an approach based on policy aware data processing, which is shown in the figure below.

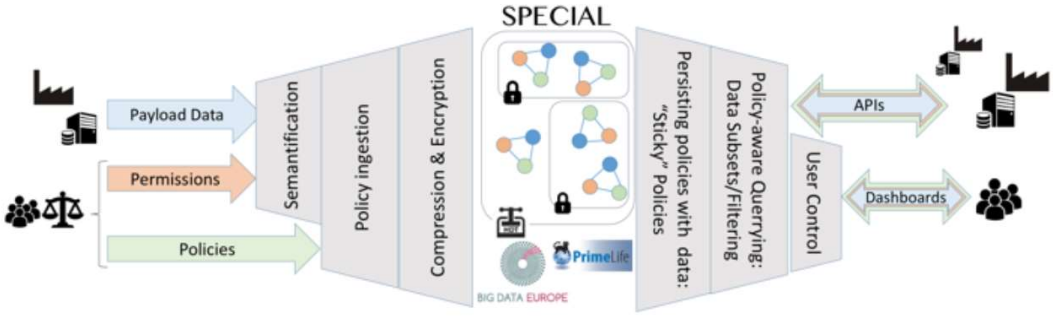


Figure 1:Birds-eye View of the SPECIAL Approach

This approach, from left to right, is defined in short as follows:

1. First, harmonise the data (both payload and the consent data) by making the semantics of the data explicit.

[1] Due to confidentiality no details of the individual usecases are presented.

2. Then, augment the data with consent approval and usage policies.
3. Ensure that the data is securely and efficiently accessible (by applying techniques such a compression, encryption etc.).
4. This creates data with sticky policies[2].
5. Finally, provide Application Programming Interface (API's) and User Interfaces (UI's) to access the payload according to the associated consent and applicable policy.

Using the proposed approach, the payload data processing is integrated with the consent and policy data. While control is awarded to the data subject via transparency and compliance checking mechanisms. If implemented correctly the system has a by-design guarantee that the data subjects consent is honoured.

[2] Sticky policies is the term for the approach to attach the policy to the data in a manner that ensures that the policy is tightly coupled to the data (which is especially important when data transcends company boundaries).

3 SPECIAL Ecosystem

The GDPR defines a data processing ecosystem consisting of various stakeholders (such as, data subjects, data controllers and data processors, supervisory authorities), and legal rights, obligations and constraints with respect to the personal data processing. The SPECIAL ecosystem is an instance of the GPDR data processing ecosystem using the data subject's consent and using the data processor's transparency information, provided by the data controller/processor, to verify compliance with the legislation.

Thus, central in SPECIAL is the management of consent and transparency data. This area is responsible for recording and managing the data subject's consent, administering the policy definitions, providing data for audits, supporting the compliance verification, etc. From now on we will use the abbreviation CTC, referring to Consent-Transparency-Compliance, to denote the area of work to which the SPECIAL project is devoted.

The other area of work in the SPECIAL ecosystem is the data processing which takes the consent into account. Whereas the CTC management is mostly domain neutral and common for each of the use cases, the added value service data processing is specific to the business objectives. In this area of work, SPECIAL will provide a common methodology and several libraries that are required to enable the implementation of the different services use cases. This area will be referred from now on to as the AV, the business Added-Value data processing.

In D2.3 a more low-level view of the SPECIAL ecosystem is presented. The AV refers to the line of business data sources, the line of business applications and the business intelligence / data science applications. Other data sources, middleware and applications from D2.3 can be considered CTC.

In the following we will further elaborate on the SPECIAL ecosystem. The next sections detail the functional/technical requirements more concretely.

3.1.1 Stakeholders

In the context of SPECIAL, the GDPR defines the following key stakeholders:

- **Data Subject**: an identified or identifiable person whose data is being processed.
- **Data Controller**: the organisation which owns the data processing service.
- **Data Processor**: the organisation which actually processes/stores the data. This may be different from the Data Controller, e.g. a cloud service provider.
- **Supervising Authority**: the authority who takes on the *auditor* role, ensuring that the data processing happens according to the legislation.

Without restricting the validity of the SPECIAL outcomes, we can simplify the SPECIAL ecosystem by assuming that the data controller and the data processor are the same entity. In the following, the term *service offering company* is therefore used as alternative term for the data processor or data controller.

Next, we will detail the data subject and the data controller in two distinct roles. The data subject can have the role of a *personal data provider* or as *data service consumer*. Indeed, one can be a personal data provider without consuming the result. For instance, you may grant your telco operator permission to use your communication data for the creation of a traffic pattern knowledgebase, but not consume the service exploiting that traffic pattern. On the other hand, a service consumer might not be a personal data provider. For example, when traffic data is used as the basis to create announcements emitted as radio messages. Obviously within SPECIAL the first role is the most critical one and will be denoted as the data subject.

The data controller is divided in two roles: one is the *policy administrator* and the other is the *service developer*. The policy administrator has the responsibility to maintain and enforce the policies that are associated with the to-be gathered personal data. This is a key role as the policy administrator will translate both the business objectives and the legal obligations into a machine processable format (D2.1). Service developers are responsible for the service implementation. They expect to find within the SPECIAL ecosystem libraries, APIs and guidelines which can be used to build GDPR compliant AV-services.

3.1.2 SPECIAL ecosystem

Figure 2: SPECIAL Ecosystem depicts the SPECIAL ecosystem. The left, coloured in yellow, is the CTC management area. The right side, in blue, is the AV data processing area. They are connected via secure interfaces via which CTC data is exchanged.

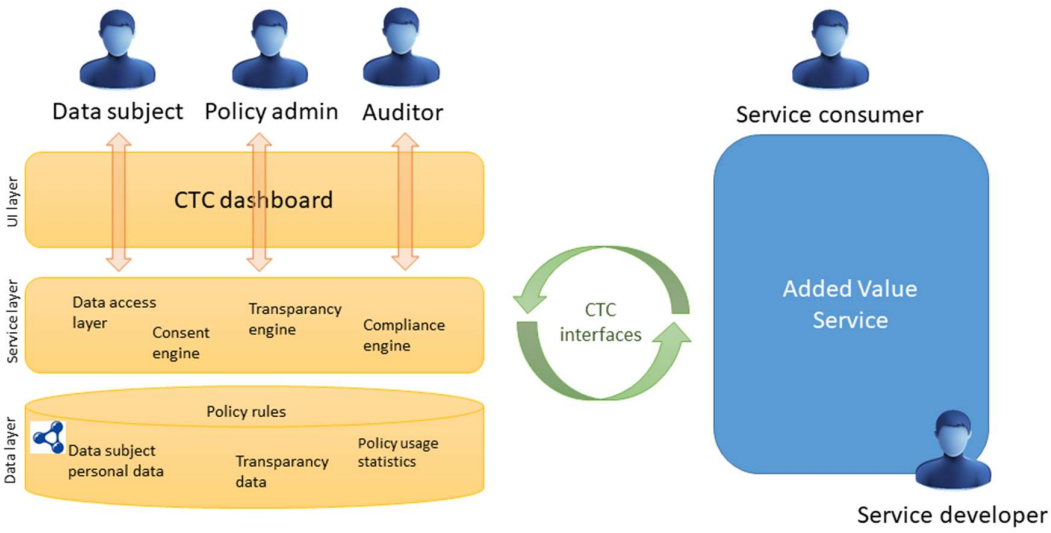


Figure 2: SPECIAL Ecosystem

The figure shows the interaction of the identified roles within the SPECIAL ecosystem. Three roles interact with the CTC dashboard:

- Data subject
- Policy admin
- Auditor

Via the CTC dashboard the data subject can execute the control on the usage of its personal data. Consent can be given or withdrawn, the purpose (policy context) for which consent is requested can be explored, insight to the usage of the data is given, etc. The policy admin is given the power to manage the policies and the power to verify the compliance of the AV service to the policy definitions. For the Auditor, the CTC dashboard provides the necessary verification to ensure compliance with the legislation.

The system architecture for the CTC management follows the multi-tier pattern. In the following subsection more details of each layer are given. In short, from top to bottom: the UI layer implements the UI interaction for the different user roles; the service layer provides the services for accessing data, consent and policy management, transparency and compliance verification; the data layer is responsible for storing the data securely.

The AV data processing area contains the data processing system, developed by the company's service developer, creating the added value data for the company. The service consumer is the party who consumes the AV service. More detail about our vision for the AV data processing is found in subsection 3.3.

The AV data processing will implement the SPECIAL approach for policy aware data processing. For that it must interact with CTC management via CTC service layer. This interaction is denoted in green. Initial thoughts on the interaction been company systems and the SPECIAL components are presented in Deliverable D1.3.

3.1.3 Linked Data centric

The above introduced SPECIAL ecosystem is from a birds-eye perspective comparable to other approaches. It distinguishes from others by the application of Linked Data[3] (or Semantic Web) as the technical foundation.

The following benefits from Linked Data form the basis for our decision to use it:

- it is based upon a domain neutral, flexible, multi-lingual data representation format standardised by W3C,
- it is the most popular data ecosystem supported with automated reasoning capabilities (e.g. OWL[4]) that has been standardised[5],
- it well-balances the human readable aspect with machine readable aspect,
- it is web-enabled by design,
- it is ideal for data integration tasks, and
- it is well-suited for cross-system/cross-organisational data interoperability.

The last item has a not to-be under-estimated value for community adoption. Since the personal data, the consent to use it and the associated policy is going to be used by many different systems within the service offering company, but also across company borders a common, reliable, semantically unambiguous way to reference this data is an important requirement. Otherwise desired properties such as transparency, which requires data processing and sharing events to be associated with the corresponding consent, are hard to achieve.

Consequently, this design requirement influences the component design for the SPECIAL ecosystem. In particular, data processing technology which does not natively support Linked Data has to be extended with it. In the SPECIAL approach this is called Semantic Lifting or Semantification. Vice versa, Linked Data native components[6] might not have the required data privacy or data security properties. It might be necessary to extend those components before they can be part of the SPECIAL ecosystem.

3.2 Consent, Transparency and Compliance Management

The focal point of the SPECIAL project is the consent, transparency and compliance management. *Figure 3: CTC dashboards* highlights this area in more detail.

[3] https://www.w3.org/standards/semanticweb/data, the term Linked Data and Semantic Web are used here as synonyms.

[4] https://www.w3.org/TR/owl2-primer/

[5] To our knowledge the only one.

[6] With components we refer to all aspects: from vocabularies, standards, protocols to software implementations.

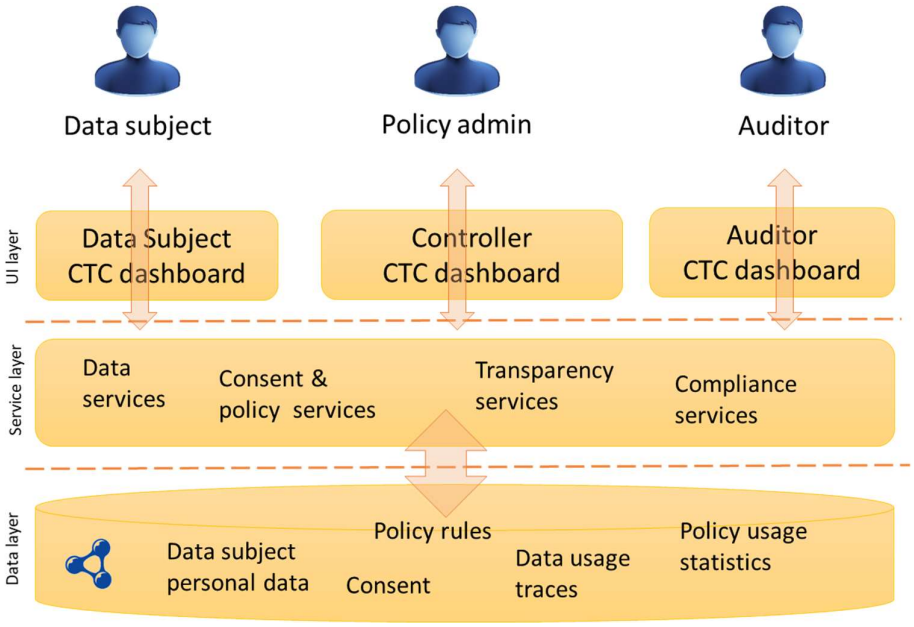


Figure 3: CTC dashboards

Data layer

The *data layer* manages and stores the CTC data which covers among others policy rules (usage constrains, legislative obligations and constraints, business logic), consent of the data subject for the data use, provenance trails about the data processing for transparency, etc. The data layer will be based on Semantic Web technologies (RDF, OWL). We refer to Deliverable 1.7 for a deeper analysis of the data layer covering functional and technical requirements, the concepts to be captured, an overview of existing approaches, the challenges and implementation considerations. Moreover, D3.2 gives an overview of a possible platform architecture in terms of specific storage solutions. Finally, the first version of the policy language itself is given in D2.1, whereas the initial description of the log vocabulary is presented in D2.3.

Service layer

The *service layer* is responsible for facilitating the creation and the access to the CTC data. The base functionality are interfaces assisting the implementing of the UI. More advanced services support the consent interpretation, transparency insights and compliance verification. *Table 1 advanced CTC services* gives an overview of the advanced services we foresee to be implemented. The service layer is also the bridge between the Linked Data based data-layer and the other data representations commonly used in the practice. For instance, the de facto standard for data exchange in UI implementation frameworks is JSON. More variety is expected in the implementation of the added-value services. For the interface two design principles are applied: (a) whenever possible a standard is applied and (b) preference goes to already used standards in the SPECIAL ecosystem. For instance, JSON-API is an industry standard driven by the Ember framework community[7]. The current exchange formats are documented in D3.2.

[7] https://emberjs.com/

Component	functionalities
Transparency engine	• List the data processing and sharing events happened • Find data processing and sharing events by data subject, by consent, by temporal window • Add data processing and data sharing events to the transparency ledger • Export the transparency data in an interoperable format
Consent engine	• List the data subject's consent timeline (when given consent, when retracted, etc.) • Fold/unfold consent into/from groups • Register consent • Revoke consent • Get all contextual information about a consent to create a human readable view • Associated a data processing event with the consent
Compliance engine	• Coherency validation of transparency data and consent data • Can be called by an access control system for ex-ante compliance checking • Can process the transparency ledger for ex-post compliance checking • Get statistics for key parameters (#consents, #revocations, #data sharing events, #data processing events …)

Table 1 advanced CTC services

UI layer

The top layer in *Figure 3: CTC dashboard* is the *UI-layer*. We foresee independent UI's serving the needs for each role. This simplifies the overall access-control mechanism as the interface targets only a single kind of user. Additionally, it creates a separation of concerns reducing the risk of disclosing information. Section 5 gives a more elaborated analysis of the key UI requirements.

3.2.1 Implementation considerations

Besides the above, the following considerations should be taken into account when realising CTC components

- *Storage*: The amount of data that needs to be stored can become easily voluminous. Parameters such as the number of data subjects, the number of consent requests and the number of data processing steps, have a multiplicative effect.
- *Scalability*: Because of the multiplicative effect is it important that the SPECIAL architecture can adapt to larger volumes i.e. via both horizontal and vertical scaling.
- *Responsiveness*: The total volume of data should only marginally impact the responsiveness of the services. Creating a single data store will destroy the data locality for some services, impacting the responsiveness.
- *Robustness & long-term applicability:* Since CTC management is bound to a legal obligation, solutions should be guaranteed to work for many years. For personal data, the GDPR calls for a long-term durable solution. If changed, the new system should be capable of importing the existing CTC data.

- *Security:* In addition to the above requirements, all components in the ecosystem must adhere to a general requirement of data security. More on our approach to identify the privacy threats and the possible mitigation strategies are found in Section 4.

3.3 Added Value Service

A second area of the SPECIAL ecosystem is the AV service for which the data subject's consent has been gathered.

SPECIAL enables the creating of privacy preserving added-value services, that enables data to be combined, aggregated, analysed, etc. The origin of the data may be very diverse: ranging from public open accessible data (e.g. touristic activity statistics from the national statistical office), commercially acquired data (e.g. events happening in a region), to data obtained from other services owned by the company (e.g. location data from the telco network). For companies to comply with informed privacy preferences and legal obligations, the data needs to be connected and combined with both consent (obtained from data subjects) and policy rules (derived from usage constraints and legal obligations) that state how the data can be used.

The use cases described by our use case partners (see Deliverable D1.5) show a wide diversity of services that could leverage our SPECIAL ecosystem. Independent of the privacy aspects the data processing must address several big data challenges because of the characteristics of the data itself. These data characteristics are commonly called the four Vs of Big Data:

- *Volume*: the amount of data being processed,
- *Velocity*: the speed that data is provided,
- *Variety*: the different models/formats in which the data is provided
- and *Veracity*: the trustworthiness of data.

Concerning volume and velocity, the data processor must handle large amounts of data, as the use cases indicate constant data streams in great amounts. Streaming processing support is hence required. But at the same time support could be needed for processing less voluminous, yet complex data having a low change rate.

All use cases indicate the usage of several data sources provided by as many different systems. To address the heterogeneity of the data sources, semantic web technologies will be applied too. This creates a uniform data layer easing the interaction with the policy management data.

In terms of veracity, some use cases provide data that is readily available and easily understood as the data is under the control of the use case partner. However, data may also be collected from "open, uncertain sources". In that case the quality and trustworthiness of the data must be investigated before they can be integrated in the service.

3.4 Big Data Considerations

Because the AV can generate large amounts of data at a very fast rate, the SPECIAL platform should be architected in such a way that it can easily cope with all of this data. Furthermore, the evaluation of the use case scenarios documented in D1.5 and D1.6 has shown a shared need for real time compliance: line of business applications cannot wait hours for a batch job to complete to tell them if their processing is compliant with user consent, nor do data subjects appreciate having to wait until the next business day for their consent and policies to be updated in the platform.

3.4.1 The Lambda Architecture

Within the Big Data community, the lambda architecture[8] is an architecture pattern for handling large quantities of data with low latency requirements. The lambda architecture, which is depicted in Figure 4, is a term given by Nathan Marz for a generic, scalable and fault-tolerant data processing architecture, based on his experience working on distributed data processing systems at Twitter. It distinguishes between three layers: the serving layer, the batch layer and the speed layer.

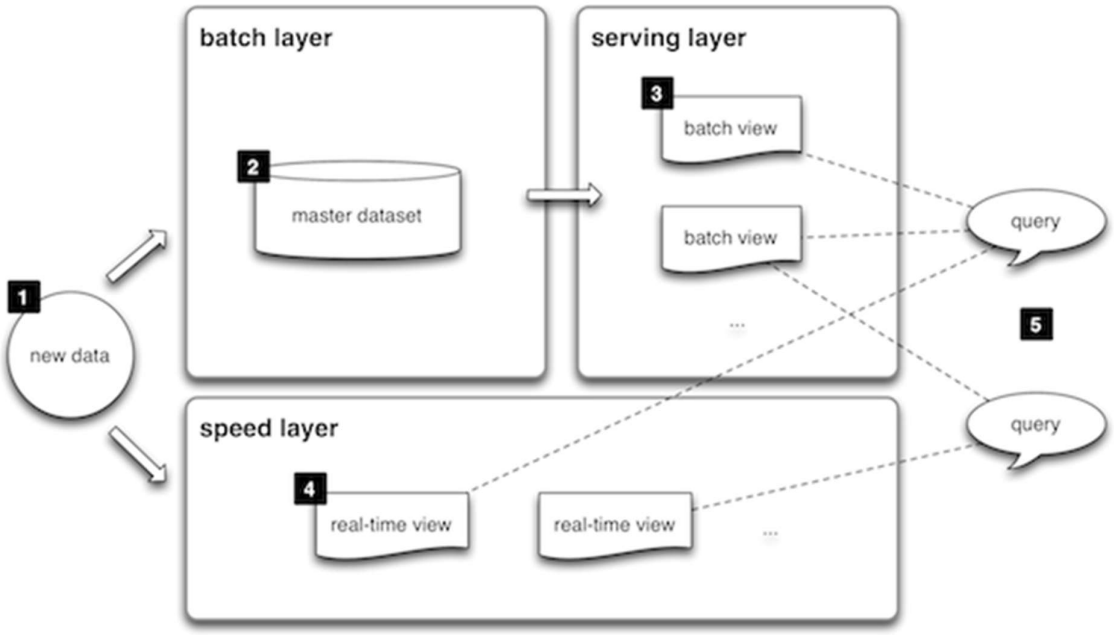


Figure 4 the Lambda Architecture pattern (as defined by Nathan Marz)

The new data is processed by the batch layer and speed layer to realise derived data views. The serving layer is responsible to make the views efficiently queryable for the business applications. The batch layer and speed layer perform data processing, but where-as the batch layer is optimised for performing processing on high amounts of data at once renewed with a low frequency, the speed layer is optimised for performing processing small amount of data given in a high frequency.

Tasks within the batch layer normally require a substantial amount of time to finish. The resulting data view can be a final product to be used in the service layer, but often and it is expected to happen in SPECIAL, it also acts a pre-processing step for the speed layer. Then it lays out the data so that the speed layer (optimised to handle a high volume of messages having a small data payload) can work efficiently.

3.4.2 The SPECIAL Architecture

The reason the lambda architecture relies on both batch and stream processing systems, is because the queues traditionally used within stream processing systems could not be used to store large amounts of data. In SPECIAL however, Apache Kafka is used, which exposes a

[8] http://lambda-architecture.net/

queue-like API on top of a durable storage system, which has no issues storing and processing extremely large amounts of data. The SPECIAL platform therefore does not need to implement a separate batch processing system, all data processing can be done by the speed layer and the results can be persisted in Kafka. This greatly simplifies the architecture without compromising its capacity to process large amounts of data at low latency.

Within SPECIAL, the serving layer will be simplified to deliver the data views on which the desired customer facing service can be built. That means integrating support for associating policies with the payload data, integrating policy enforcement and compliance checking mechanisms.

To integrate with our SPECIAL CTC management, semantic lifting is required. This means the SPECIAL architecture will be augmented with Linked Data processing capabilities[9]. The processing and policy data will be semantically lifted by transforming it to RDF.

The speed layer provides streamed data processing, relying on separate streams per data channel and purpose. The first implementation of the envisioned architecture is given in the SPECIAL platform Policy and Events Release, documented in D3.2, and shown in *Figure **5**: SPECIAL Streaming Data Processing*.

The proposed architecture relies on Apache Kafka. Unlike normal queueing systems, records in Kafka are persisted whether they are consumed or not, making it useful as a data store. It is a special purpose distributed filesystem dedicated for high-performance, low-latency commit log storage, replication, and propagation. When compared with other storage systems, such as Hadoop, the advantage of Kafka is that it has the API of a pub-sub and queuing system. It allows data and data updates to be treated as immutable events and has well defined semantics for how to consume these, while in Hadoop's file-oriented world most of the semantics need to be communicated out of band.

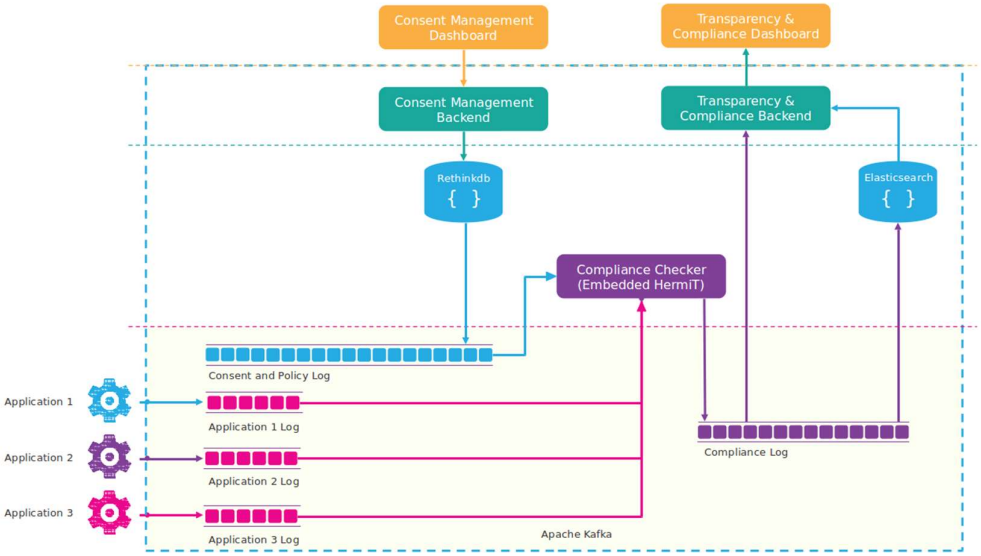


Figure 5: SPECIAL Streaming Data Processing

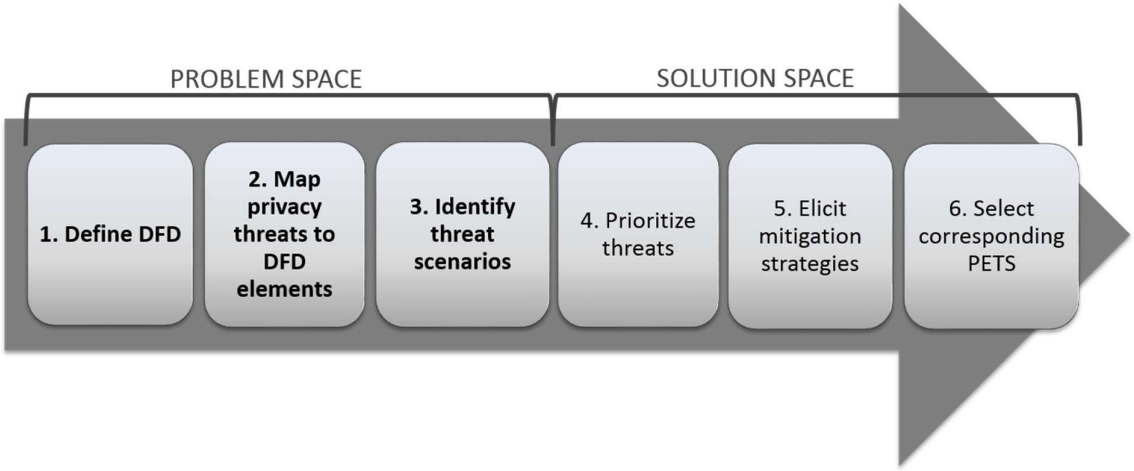
[9] Deliverable D2.3 describes general Ontology-based Data Access frameworks, such as RDB2RDF and R2RML In Deliverable 3.1, the Semantic Data Lake Ontario has been discussed. Some aspects of this might be applicable here too, but that has to be investigated.

4 Assessing privacy threats

The SPECIAL project focusses on building consent-awareness and transparency support for data processing systems. The to-be created components themselves are subject to privacy threats. To assess these threats and take appropriate mitigation actions, all the software will be evaluated using the LINDDUN [10] methodology. LINDDUN refers to the different threat categories the methodology distinguishes: Linkability, Identifiability, Non-repudiation, Detectability, Disclosure of information, Unawareness, Non-compliance.

This methodology is a threat modelling technique which aids in highlighting the possible privacy threats and the mitigation actions that must be taken. It systematises the development process with regards to privacy threats. Applying LINDDUN results in an overview of the threat status of each component.

The approach, illustrated below, consist of 6 steps of which 3 are applied during the problem definition phase, and 3 steps during the solution design phase.



In short, the steps are:

1. Define the data flow diagram (DFD) based on the high-level description of the system. The modelling entities are *external entities, data stores, data flows,* and *processes.*
2. Map the privacy threats to the DFD. When a privacy threat is acknowledged, a short description is given too.
3. Identify the threat scenarios. For each identified privacy threat in the mapping one or more threat exploitation scenarios are designed using a tree representation.
4. Having the scenarios, the next step is to prioritise them using a risk assessment.
5. Next, in order of priority, the threat mitigation approach is defined.
6. Finally, the solutions are detailed by selecting & implementing an appropriate Privacy Enhancing Techniques.

The LINDDUN methodology aids in identifying system wide threats, however some of the threats might be inherent to the chosen technology. In that case, either the technology must be replaced with a better alternative or SPECIAL has to investigate improvements so that the

[10] https://distrinet.cs.kuleuven.be/software/linddun/index.php

threat's impact is reduced. For instance, identity management is such a topic. To relate entities with each other, each data entity needs an identifier. The scope of the identifiers can be either global or local. For Linked Data, the base data representation formalism in SPECIAL, global scoping is normally assumed. Local scoping is possible, but it is usually less supported by the applications. Indeed, global identifiers have the following benefits: they condense data representation, lead to high reuse, and allow easy identification of entities. However, the latter benefit is at the same time a data privacy threat as it makes unlocking sensitive data easy.

4.1 Data privacy threats mitigations

The GDPR not only defines functional requirements (constraints and obligations) for a data processing system, it also states that the software components must be designed with data privacy in mind (called the Privacy by Design principle in GDPR). Software solution providers should apply the best practices at the time and are advised to constantly improve their solutions so that the processed data is handled securely.

Hereunder we present a set of characteristics that will impact the design of the to-be developed software components and pilot setups.

4.1.1 Authentication & Authorisation

Personal data should only be accessible after identity of the data requester is confirmed. Authentication is the process which establishes this identity confirmation. Authorisation is the process to confirm whether the identified user has the right to execute a service or access a particular piece of information.

Authentication and authorisation are a necessary requirement for the externally accessible interfaces such as user interfaces, but also it is important to consider them for the internal data exchange processes. A multi-tier architecture, integrating an authentication & authorisation layer on the internal APIs creates additional security against unwanted penetration. Such a multi-layer approach is decreasing the likelihood that the impact of a data breach is large, but at the same time it may come at an additional operational cost.

As described in D3.2, the SPECIAL platform will rely on the OpenID Connect[11], industry standard for authentication and OAuth2[12] for authorisation.

4.1.2 Encryption

A second measure to increase the data security is the application of data encryption. Encryption is the process of encoding the information so that it is only readable by trusted parties having the key to access it.

Encrypting data addresses scenarios such as:

- Unintended disclosure of the data to other system users, in particular users with high rights such as system admins
- Easy disclosure of the data in case the system has been hacked or if the system is accidently exposed to the public
- Allows to share data over public channels,

[11] http://openid.net/connect/

[12] https://oauth.net/2/

- Reduces the risk of receiving tampered data as tampering requires to break into the encryption

The above scenarios correspond to the following common application areas for encryption techniques:

- The data itself
- The storage medium
- The communication channel

For the latter two, we can mostly rely on the application of existing industry standards and best practices. Encrypting/decrypting on the fly of data being stored in a storage medium is a common offering by cloud providers[13]. Communication channels such as HTTP & telnet, are being replaced with their secure variants HTTPS [14]and ssh[15].

For SPECIAL, encryption of the data itself is more of an open problem. Linked Data is commonly used and exchanged as plain text. The Linked Data ecosystem does not have a built-in approach in which the data represented in RDF is encrypted and stored. Research into the creation of encrypted RDF is therefore part of the research objectives of SPECIAL. Our work on *Self-Enforcing Acccess Control for Encrypted RDF* [16] demonstrates how predicate-based encryption can be applied to realize fine-grained access control on triple patterns over encrypted RDF datasets. In the course of the project, we will investigate how these techniques can be integrated in the SPECIAL platform.

4.1.3 Anonymisation

Anonymisation is a technique turning a source dataset into an equivalent dataset with respect to some properties so that the identifiable real-world data subjects present in the source dataset cannot be derived from the anonymised dataset. According to legal interpretation of the GDPR and related legislation, anonymized data can be used more freely. A discussion on this topic can be found in Deliverable 1.2, from page 13 onwards.

However, based on the use case descriptions and the presented SPECIAL ecosystem, the application of anonymisation will be rather limited in the project. Consent management requires access to the identity of the data subject so that data processing steps can apply the consent as requested.

Moreover, none of the state-of-the art anonymization techniques realises full anonymisation[17], but at most a pseudo- anonymisation, the project will not rely on this risk mitigation technique to be GDPR compliant. At most, the pseudo-anonymisation will be used as an additional obfuscation reducing the impact of a privacy data breach.

[13] A description for the Azure cloud storage is found here: https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption

[14] https://www.w3.org/2001/tag/doc/web-https

[15] https://www.ssh.com/ssh/protocol/

[16] Self-Enforcing Access Control for Encrypted RDF, Javier Fernández, Sabrina Kirrane, Axel Polleres and Simon Steyskal, Proceedings of the 14th European Semantic Web Conference (ESWC2017), 2017

[17] See Deliverable D1.2, p 16.

4.1.4 Purpose based data storage & data access

The GDPR stresses the aspect that the data is only to be stored, used and shared for the purpose consented to. This legal perspective has inspired the technical perspective on how the data should be stored and made accessible and the resulting work in WP2.

Ideally a data processing environment should only request data for which it has the permission, at the time it needs it. Often, still today, application engineers assume that access to the required data is granted all the time, whenever they need it. This simplifies the implementation. Another common activity in software projects is the creation of a developer friendly uniform way to get access to all the possible needed data for the data processing. Usually, the complexity that not all information about a resource is shareable, but only some of its properties when some conditions are met, is ignored. In SPECIAL, we will ensure that neither of the attitudes can risk unwanted disclosure of data.

D2.3 illustrates a complete and tractable structural subsumption algorithm for compliance checking over SPECIAL's policies, taking into consideration purpose and time limits, as expressed by data subjects using the policy language documented in D2.1. This reasoner applies to a fragment of OWL2-DL that slightly generalises the policy languages adopted by SPECIAL (usage policies, business policies, and the partial GDPR formalisation). In particular, it tolerates the creation of new policy attributes and new vocabulary terms, as well as attribute nesting at arbitrary levels. It also includes a consistency checking algorithm for policies, useful for policy validation.

5 User Interface Requirements

D4.1 provides a mind map of the SPECIAL transparency dashboard and control panel (*Figure 6: A mind map of the transparency dashboard and control panel derived from the SPECIAL proposal*). It shows the key project terms as functional components of the dashboard (coloured green) and general attributes or requirements of the dashboard (coloured orange). Because it is by far the hardest problem to solve from a research perspective, the requirements here focus on the data subject dashboard. A data controller and auditor dashboard can be provided by existing business intelligence and reporting tools.

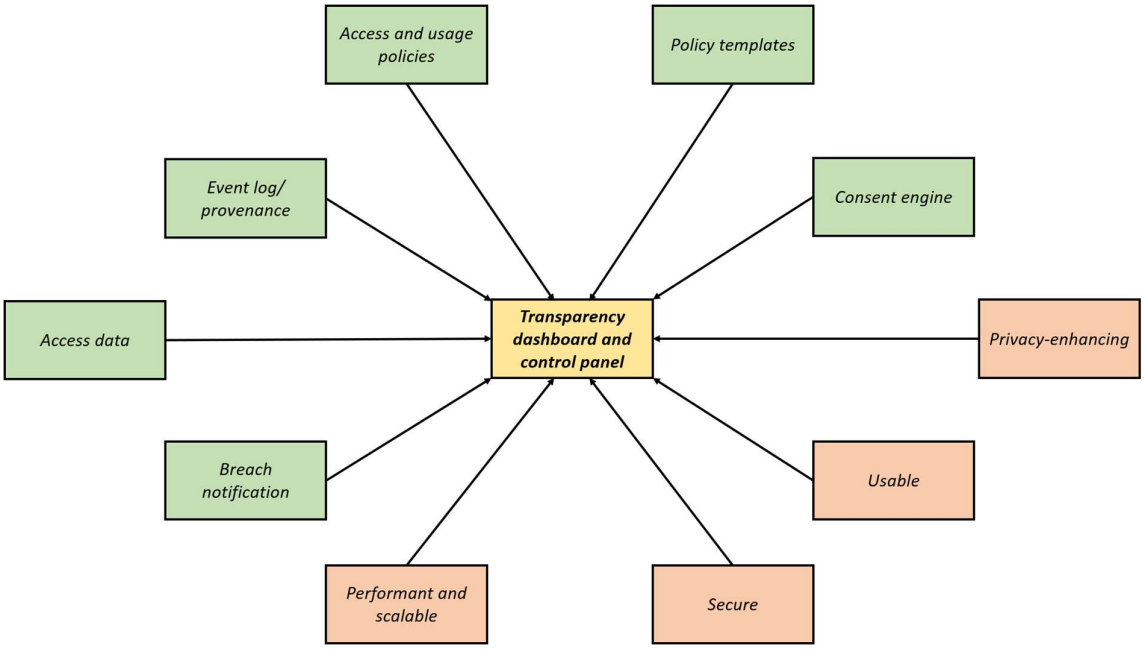


Figure 6: A mind map of the transparency dashboard and control panel derived from the SPECIAL proposal

5.1 Functional components

5.1.1 Access data

The dashboards main purpose is to offer data subjects an interface to access and assess their personal data that is processed by a single or multiple controllers and processors within a specific context for various purposes. While all this information needs to be made accessible to the data subject, it is also of importance to make it digestible for the data subject. Providing access to the data does not necessarily imply transparency, thus a strong focus needs to be put on usability.

5.1.2 Event log/ provenance

In addition, meta information and provenance data are needed to provide full transparency to the data subject. This includes the purpose and the legal basis of the processing, involved processors, context information like time and the physical location of the processing servers,

and which safeguards are applied to protect the data subject's personal data. It will also include information on the compliance of the events with the data subject's consent and policies. The event log's visualization and the identification and presentation of the relevant and necessary information are major challenges addressed in WP4.

5.1.3 Access and usage policies

The expression of access and usage policies is a core functionality of the privacy dashboard. However, the underlying policy language SPECIAL introduces in D2.1 goes beyond conventional access control systems, since legal requirements of the GDPR and data subjects' policies shall be expressed and formulated with it. The SPECIAL UI will avoid complex interfaces with many options, so data subject's will not be required to understand the policy language at all while using it.

5.1.4 Policy templates

To reduce even more complexity, policy templates will be offered to data subjects. The definition of reasonable policy templates is a central challenge of the SPECIAL UI, which will be evaluated in user studies to find out if policy templates really ease the complexity of privacy policies and decisions.

5.1.5 Consent engine

The consent engine is another core component of the dashboard. It is supposed to allow data subjects to review consent that was previously given, to give informed consent for additional purposes offered by the controller, and to withdraw consent if necessary. The SPECIAL UI pursues two main goals: (i) designing and implementing consent interfaces that make consent actually (and measurably) informed, and (ii) finding mechanisms to prevent data subjects being "scared away" by consent requests, for example by informing about the risks and highlighting the benefits of the data disclosure.

5.1.6 Breach notification

The breach notification is a new legal requirement of the GDPR obliging controllers to properly inform data subject's in case of a data breach. In case of a data breach, data subjects can be provided with the most relevant and urgent information and recommendations to react upon. Controllers might benefit from a standardized, uniform, and automated mechanism enabling them to be compliant with the GDPR. The SPECIAL UI aims to identify the relevant information data subjects need and how this can be presented in a usable and user-friendly way.

5.2 General requirements

5.2.1 Performant and scalable

The dashboard must be performant and scalable, this means, it must be capable of handling vast amounts of data, while keeping response times within a reasonable time range. To achieve this, stress tests will be conducted. Additionally, mechanisms will be implemented that limit the amount of data displayed. This also contributes to the usability of the dashboard.

5.2.2 Secure

The dashboard must be secure since it is used to access sensitive personal data. The security risk involved by introducing the privacy dashboard (as an additional mean to access personal data) must be limited to an absolute minimum.

5.2.3 Privacy-enhancing

The dashboard must be privacy-enhancing to an extent that the introduction of a new security risk is justifiable. Data subjects must be able to use it to fulfil tasks that actually enhance their data privacy. These tasks do not only have to be fully implemented, but also the definition of these tasks is crucial.

5.2.4 Usable

It must be usable by a variety of user groups and types to serve the purpose as a transparency-enhancing tool and privacy-enhancing technology. Transparency is enabled by granting access to the data, but still requires a usable and user-friendly presentation so data subjects can interpret and comprehend the impact of the presented information on their data privacy.

6 User stories

In D1.4, we described a collection of user stories defining target characteristics of our SPECIAL platform. Those user stories were translated into an internal, shared, project backlog and have served as a prioritized features list, containing short descriptions of all required functionalities. We have user stories for each of the identified stakeholders, but also for the objectives of the SPECIAL project with specific attention to the impact of the hacking challenge (D5.2). In this deliverable, we revise the stories and group them around common themes.

In an agile spirit, in the remainder of the project, the listed user stories will be further elaborated. It is expected that new ones will be added. The presented order does not express a priority. The prioritisation is always made in collaboration with all project partners, leading to a shared vision and an (improved) project implementation roadmap.

6.1 User stories for stakeholders

6.1.1 Data Subject

As a…	I want to…	Theme
Data subject	browse my personal data	Transparency framework, Data inventory
	be able to see all applied policies on any given piece of data	Transparency framework, Consent management
	be able to see how my data is being processed	Transparency framework
	adapt my personal data	Data inventory
	request to be forgotten (erase my data)	Data inventory
	export my data	Data inventory
	explore the policy definitions	Consent management
	be able to see the policy (change) history for any given policy	Consent management
	adapt my policies	Consent management
	be able to set an expiry date for my policies	Consent management
	have my policies honoured	Consent management, Compliance engine
	export my policies	Consent management
	import policies from a third party	Consent management
	have my data securely stored	Security
	have secure access to the portal	Consent management, Transparency framework, Security

6.1.2 Policy administrator

As a…	I want to…	Theme
Policy administrator	browse the policy definitions	Policy management
	define policy definitions	Policy management
	edit policy definitions	Policy management
	delete policy definitions	Policy management
	explore the usage of policy definitions	Policy management, Transparency
	have all changes propagated to all	Policy management
	have secure access to the portal	Policy management, Security

6.1.3 Auditor

As an…	I want to…	Theme
Auditor	browse the policy definitions	Policy management, Consent management, Transparency framework
	explore the usage of personal data	Transparency framework, Data inventory
	explore data subjects' policies	Consent management, Transparency framework
	investigate policy compliance history	Compliance engine, Transparency framework
	ensure the event log hasn't been tampered with	Transparency framework
	only authorized personnel can access personal data	Transparency framework, Security
	have secure access to the portal	Transparency framework, Security

6.1.4 Service Provider and Developer

As a…	I want to…	Theme

Service provider	be able to create a business valuable service while respecting the law, as well as business and data subjects' policies	Policy management, Consent management, Compliance engine, Security
	share & sell the resulting data for which consent was provided	Consent management, Data inventory
	have a secure data processing solution with minimal personal data disclosure risks	Security
	have a reliable way to implement the right to be forgotten	Policy management, Data inventory
Service developer	have easy, secure & standard access to the consent of a data subject	Security, Consent management
	have easy, secure & standard way to log the data processing provenance trail	Transparency framework, Security
	have the hooks to implement the right to be forgotten	Policy management, Data inventory

6.2 Service consumer

As a…	I want to…	Theme
Service consumer	have the guarantee that the service is based on trustworthy, legally acquired data	All

6.3 User stories for SPECIAL objectives

We add some key user stories covering the perspective of the SPECIAL project. In contrast to the stakeholder user stories, these reflect the research and technical ambitions the project has.

As a…	We want to…	Theme
Project consortium	have a simple development and deployment environment for the platform and its pilots	All
	have a privacy threat analysis for the components of the SPECIAL platform	Security
	have a domain independent consent ontology	Consent management, Compliance engine
	have a domain independent policy ontology	Policy management, Compliance engine

	have a domain independent transparency ledger	Transparency framework
	have a reliable, trustworthy policy engine protected against privacy threats	Compliance engine, Security
	have 3 pilot instances of the SPECIAL platform, each of them corresponding to a use case	All

7 Hacking challenge

One of the objectives of the project is to setup two public hacking challenges (D5.2 and D5.4) to evaluate the SPECIAL platform. Instead of merely creating a public instance and hoping the anonymous internet society finds it and attempts to penetrate it, it is our intent to create a few hacking challenges around the privacy protection measures the platform has.

An example of such a challenge is trying to break into the policy engine with the intent to alter the response on the query if there is consent. Obviously, if that is possible, the policy engine's responses cannot be trusted and hence data processing relying on the consent is untrustworthy.

At this moment, these scenarios are not fixed. This is future work that will be collected during the next phase in the project. These scenarios will become key user stories.

Setting up a hacking challenge imposes an important milestone in the project with respect to the technical readiness of the involved components and data. At the launch of the hacking challenge, the components are to:

- install and deploy easily on the hackers' local infrastructure (since it is not our objective to have the hacker challenge our project's cloud infrastructure),
- have the desired functionality,
- have a documented list of unimplemented features/weaknesses (to avoid reporting issues which we are aware of),
- ensure there is representative syntactic data available.

Aside from the technical requirements the success of a hacking challenge depends on a good communication strategy and expectation management. The communication strategy must initiate enthusiasm in the targeted community.

8 Software & system design principles

Deliverable D3.2 describes the second release of the SPECIAL platform. Whereas the first version mainly focussed on the deployment and implementation aspects of the SPECIAL platform reusing the BDE platform, its extensions and experiences, the Policy and Events release implements some of the key components and gives a clearer outline of the framework the project is to build upon.

Based on what we have learned so far, in the next section, we list a number of technical design principles that will be applied to the development of the SPECIAL platform.

8.1 Operational environment

Principle 1) Automated system rollout.

Using system deployment descriptions such as Terraform (system resources layer) and Docker Compose (services layer) the roll-out of an application becomes reproducible and reliable. Because the description is stored in a source control repository, changes over time and variants can be maintained without the need of having them actively running. The consumption of system resources can then be dedicated to the active developments.

Principle 2) Cloud enabled by design

Our platform should be hardware and Operating System neutral as much as possible. Using a service abstraction layer (i.e. Docker) addresses one part. Additionally, the setup has to be decoupled from the local file system. Only then will the system be completely cloud enabled and runnable independently.

8.2 System architecture

Principle 3) Modular design, preferably following the micro-services pattern

Micro-service design is the idea to create a system from the integration of a collection of services, each with a dedicated purpose. This approach makes it easier for the system to scale: if one service is in high demand, adding new services of the same kind is a straightforward action. In addition, it allows to focus the development effort. The approach has proven results in the design of end-user facing software.

Principle 4) Reuse best practice standards for well-known technical challenges

As already mentioned in section 4, many privacy threats have industry supported mitigation strategies. Therefore, unless they are not sufficiently appropriate or adequate, it is our strategy to apply the best practices as much as possible.

Principle 5) Use an event driven, streaming integration strategy

The system should use an event driven, message passing approach to service integration where possible. This decouples services from one another, making it easier to integrate the system in various environments. Furthermore, this allows data to be streamed through the system, minimizing latency and producing near-real-time results.

8.3 Component interaction

Principle 6) Payload data is preferable in the form of RDF, JSON-LD or JSON.

Although the use-cases indicate that data from various sources with a multitude of formats are processed to create the desired value-adding services, it is our intent to keep the heterogeneity as much as possible under control by using preferable RDF, JSON-LD or JSON as payload data representation. Where-as RDF and JSON-LD are highly compatible with each other, JSON requires additional semantic lifting. This lifting can be defined by adding an LD context to the JSON payload. Thus, although not technically imposed that the data is exchangeable, these 3 data representation formats can form a uniform data landscape.
When a component does not comply with this preference, it may be required to create a dedicated payload translation layer for the component. To some extent, semantic lifting acts as such wrapping.

Principle 7) The data-exchange channels are secure.

The payload data will be exchanged between the services. It is very important that the used channel is secured against penetration: HTTPS, secured database connectors (ODBC, JDBC), secure file access and a secure message bus (Kafka) are the preferred choices.

9 Conclusions

We have provided a high level technical overview of the SPECIAL ecosystem, identifying the key stakeholders with their main user stories, and the high-level software design principles based on our findings and experiences so far. This deliverable, along with the other requirements analysis deliverables, will serve as a reference for all ongoing and upcoming development and research efforts.

The common understanding allows the project to refine the implementation and research roadmap for the following months with the goal of extending and enriching the SPECIAL platform with new or improved functionality and insights. Such improvements will be directly reflected in the upcoming releases, starting with D3.4 [19] and D4.3 [20].

10 References

[1] D1.3 Policy, transparency and compliance guidelines V1, Report, Public, M8

[2] D1.4 Technical requirements V1, Report, Public, M8

[3] D1.5 Use case scenarios V2, Report, Public, M14

[4] D1.6 Legal requirements for a privacy enhancing Big Data V2, Report, Public, M15

[5] D1.7 Policy, transparency and compliance guidelines V2, Report, Public, M17

[6] D2.1 Policy Language V1, Demonstrator, Public, M12

[7] D2.2 Formal representation of the legislation V1, Demonstrator, Public, M12

[8] D2.3 Transparency Framework V1, Demonstrator, Public, M14

[9] D2.4 Transparency and Compliance Algorithms V1, Demonstrator, Public, M14

[10] D2.5 Policy Language V2, Demonstrator, M21

[11] D2.6 Formal representation of the legislation V2, Demonstrator, M21

[12] D2.7 Transparency Framework V2, Demonstrator, M23

[13] D2.8 Transparency and Compliance Algorithms V2, Demonstrator, Public, M23

[14] D3.1 Initial setup of policy aware Linked Data architecture and engine, Demonstrator, Public M6

[15] D3.2 Policy & events release, Demonstrator, M16

[16] D3.3 Backend Scalability and Robustness testing report V1, Report, Public, M18

[17] D4.1 Transparency dashboard and control panel release V1, Demonstrator, Public, M16

[18] D4.2 Frontend Scalability and Robustness testing report V1, Report, Public, M18

[19] D3.4 Transparency & compliance release, Demonstrator, Public, M2

[20] D4.3 Transparency dashboard and control panel release V2, Demonstrator, Public, M25

[21] D5.2 Public challenge report V1, Report, Public, M21

[22] D5.4 Public challenge report V2, Report, Public, M30