
# Initial Specification of BPR4GDPR architecture

## Deliverable D2.3

**Editor**

Nikolaos Dellas (SLG)

**Reviewers**

Lorenzo Bracciale (URM)

Adrián Juan-Verdejo (CAS)

**Date**

28th February 2019

**Classification**

Public

BPR4GDPR has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 787149

## Contributing Author

| Name | Partner |
|---|---|
| Nikolaos Dellas | SLG |
| Nikolaos Dellas, Mariza Koukovini, Georgios Lioudakis | SLG, ABOVO |
| Rashid Zaman, Azadeh Mozafari, Marwan Hassani | TU/e |
| Lorenzo Bracciale | URM |
| Nikolaos Dellas | SLG |
| Adrián Juan-Verdejo | CAS |
| Nikolaos Dellas | SLG |
| Adrián Juan-Verdejo | CAS |

## # Version History

| # | Description |
|---|---|
| 1 | 02/02/2019 — Structure and template |
| 2 | 20/02/2019 — First integrated version of SLG and ABOVO contributions |
| 3 | 23/02/2019 — TU/e contribution |
| 4 | 25/02/2019 — URM contribution |
| 5 | 26/2/2019 — Prefinal version |
| 6 | 27/02/2019 — Description of the data subject support asset |
| 7 | 28/2/2019 — Final version |
| 8 | 01/03/2019 — CAS case improvement |

## Table of Contents

# 1 Introduction

This deliverable reports the specification of BPR4GDPR system-as-a-whole architecture (Task T2.4), taking into account the regulatory analysis (D2.2), the identified functional and non-functional requirements (D2.1) as well as input from the initial specification of the individual components. The specification includes the high-level functional structure of the identified BPR4GDPR components as well as their interaction patterns. The detailed specification of the identified components will be provided by the work packages WP3, WP4 and WP5, taking as input the hereunder specified architecture.

Section 2 provides a high level architecture overview. Starting from the basic system functionalities emerging from the requirements analysis, the key components of the architecture and their interaction patterns have been identified. In this direction, the BPR4GDPR architecture is divided in four “quadrants”/blocks, reflecting different groups of functionalities: governance, planning, monitoring and runtime.

Section 3 details policies and their management. The Information Model, along with the ontology implementing it, is described. Further, the rule-based policy framework, also implemented as a semantic ontology and built on top of the aforementioned Information Model, is also elaborated. Finally, the logical architecture of the governance component is defined.

Section 4 deals with the management of processes. The BPR4GDPR Compliance Metamodel ontology is defined enabling the detailed specification of processes and their verification and adaptation, when necessary.

Section 5 describes the process mining and monitoring modules. The process mining module aims at initially discovering the process model out of collected event logs while in the process monitoring module a repair of the process models is performed based on the degree of conformance with business rules and rules reflecting the GDPR regulation.

Section 6 elaborates the run-time BPR4GDPR toolkit. First, the internal structure and the interfaces of Data Management Middleware, that acts as a data proxy providing access and usage control and being regulated by the Policy Framework, is presented. Then, tools that access the company resources through Data Management Middleware and offer an easy and fast deployable solution to cope with specific GDPR aspects are described, focusing on their structure and deployment model. Finally, the identified core BPR4GDPR tools are shortly described.

# 2 High level architecture overview

This section introduces the basic architectural concepts of BPR4GDPR. It begins by identifying fundamental aspects pertaining to specification and execution of organisational processes, reflecting the points of intervention that the project approach will address. It then proceeds with an overview of the components and interfaces that make up the architecture in order for the latter to fulfil the corresponding requirements and functional needs.

## 2.1 Functional view

A fundamental characteristic of the BPR4GDPR approach is that it addresses GDPR compliance in a *holistic* manner, in the sense that the proposed solutions aim at covering the full process lifecycle, from its initial identification or specification to its enactment and execution, as well as its post-analysis for mining data related to privacy. All phases are under the control of a comprehensive framework of GDPR-oriented privacy policies, to which processes shall comply. In addition, BPR4GDPR anticipates a toolkit fostering the enforcement of compliant processes at execution time.

As illustrated in Figure 1, there are six main stages comprising the BPR4GDPR process lifecycle, numbered 1 – 6, respectively dealing with its specification by an administrative user or its discovery based on event logs, its analysis and re-design, implementation, execution and monitoring, resulting eventually in possibly updated process models, adapted to real-time circumstances and other evolutionary factors. Furthermore, BPR4GDPR considers two additional phases, vertical to the process lifecycle and devised, respectively, for the initial actions that should take place in order for an organisation to become BPR4GDPR-ready (phase 0), and for the operations that are either horizontal, or process-independent (phase 7). The eight phases are summarised in the following.

**Phase 0: Set-up** This phase consists in all tasks that concern setting up the base elements of the BPR4GDPR operation and performance of all activities that are preparatory in nature, in the sense that they are fundamental for the whole system to work. These include, for instance, the specification of the, possibly varying per organisation, information models, the classification of data, systems and other resources, the assignment of roles and attributes to the different entities —human and others— participating in the system, the definition of purposes behind data collection and processing, and the specification of policies and underlying rules that should govern the operation of the system components.

**Phase 1: Process identification** This phase concerns the definition of process models through two possible mechanisms: i) automated process discovery, in order to eventually depict, in a formal way, the procedures and associated information flows taking place within an organisation through the resulting process models; ii) manual definition of such procedures by administrative users using the appropriate graphical tool. Either way, the outcome of this phase will be BPR4GDPR process model specifications, to be, at later stages, subject to incorporation of sophisticated privacy constraints enforceable at run-time, as required.

**Phase 2: Process analysis** This concerns the analysis of a process model in order to identify the risks, flaws and points of non-compliance, on the basis of well-defined policies. This way, process models shall be evaluated and verified as regards their compliance with the GDPR and provisions thereof. This phase entails a highly expressive policy framework, specified during Phase 0, and considering a variety of aspects and parameters,

such as attributes, context, dependencies between actions and participating entities therein, as well as separation and binding of duty constraints.

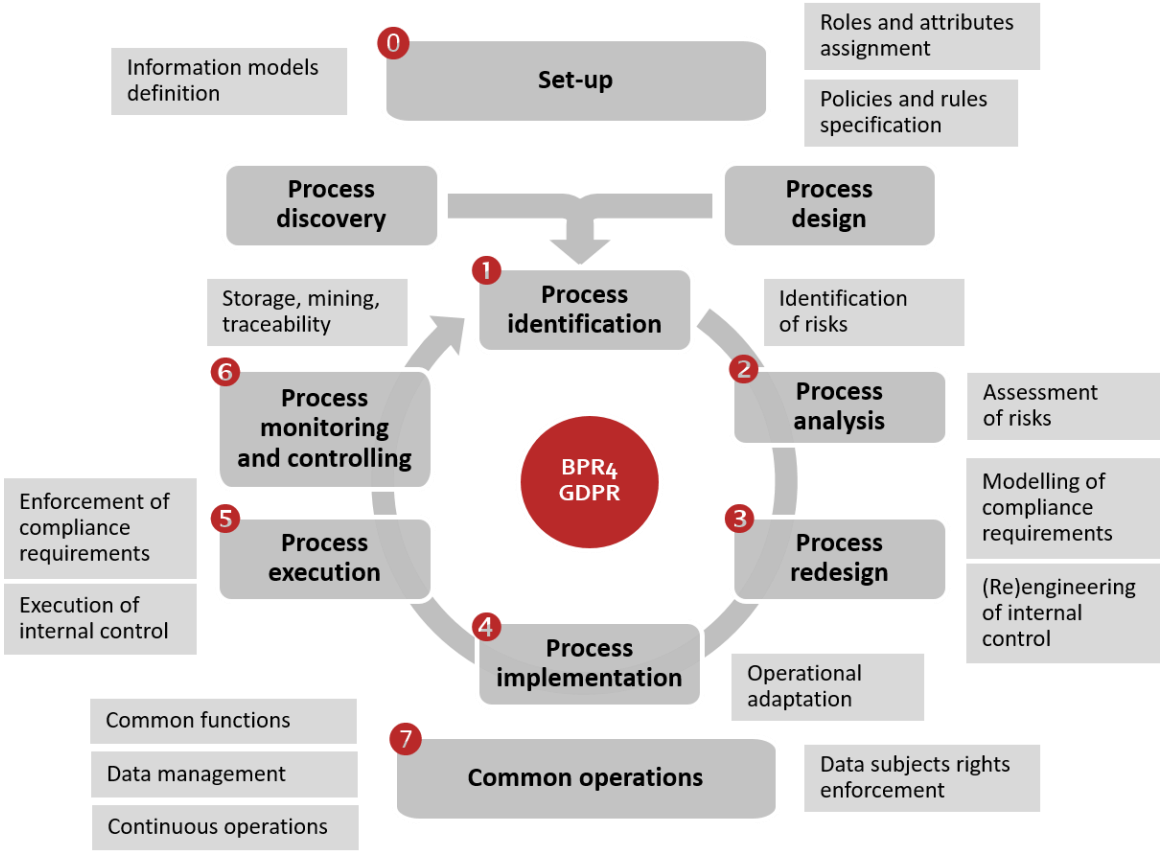


**Figure 1: BPR4GDPR operational phases**

**Phase 3: Process redesign** This phase complements process analysis, by providing for the automatic transformation of non-compliant process models, so that they are rendered inherently privacy-aware before being deployed for execution. It is supported by a Compliance Metamodel, a comprehensive process modelling technology able to capture advanced privacy provisions, constituting a fundamental goal of the project.

**Phase 4: Process implementation** This concerns the effective enactment of GDPR-compliant processes within each specific organisation, mainly as regards two aspects. The first essentially reflects the requirement for availability of those mechanisms necessary for the enforcement of privacy provisions dictated by the GDPR-compliant process models. To this end, a comprehensive set of tools able to support all diverging requirements that may arise from GDPR, related to data handling, data subjects' involvement, various PETs, etc., is needed, so that even organisations with currently no such infrastructure in place can readily have such mechanisms at their disposal. The second aspect is related to the structural and semantic alignment of the modelled processes with the actual infrastructure of the organisation; this shall be grounded primarily on the semantic foundations of the project, which will enable the refinement and adaptation of the BPR4GDPR models to each organisation's reality through the corresponding tools.

**Phase 5: Process execution** This extends process implementation by ensuring the execution of processes in accordance to compliant process and following the appropriate configuration set forth during the process

implementation phase. In other words, it is mainly during this phase when the mechanisms towards real-time privacy protection are applied and respective provisions are enforced.

**Phase 6: Process monitoring and controlling** This phase concerns the use of process mining for the ex post analysis of processes, in order to ensure that specified policies are indeed enforced, fostering accountability. Furthermore, and apart from verifying compliance, such techniques will offer the added value of automatically improving process models over time towards optimised fulfilment of both legal and business goals and requirements.

**Phase 7: Common operations** This refers to operations that are not (necessarily) part of a process lifecycle, but are rather executed asynchronously to processes or are independent thereof. They fall in different categories, including:

i. Functions that are supportive to all phases and other organisational activities (e.g., authorisation mechanisms).
ii. Enforcement of certain data subject rights, such as synchronous and asynchronous consent and management of privacy preferences, access to data, erasure, portability, etc.
iii. General data management functions, such as secure data storage and enforcement of retention provisions.
iv. Continuous operations, such as risk estimation, operations logging, etc.

## 2.2 Components view

In order to cover its functional needs towards GDPR compliance and cope with the operational phases described in Section 2.1, BPR4GDPR has specified the system architecture highlighted in Figure 2. As illustrated, the BPR4GDPR architecture is divided in four "quadrants", reflecting different groups of functionalities.

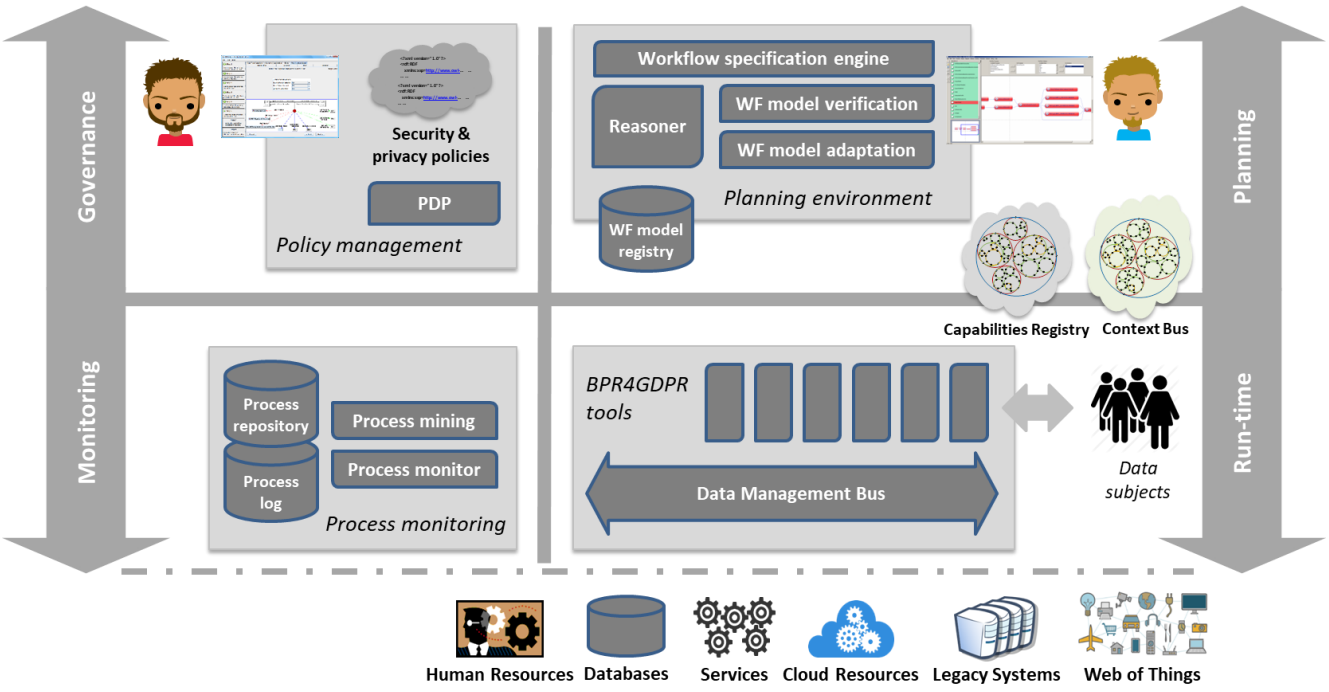


Figure 2: BPR4GDPR system architecture

The **Governance** block provides all functions related to the specification of regulation-driven policies and reasoning thereof, thus representing the Policy Decision Point (PDP) of the system. **Planning** concerns the specification of workflow models and, based on the appropriate Compliance Metamodel, their verification as regards compliance with the GDPR and their subsequent transformation, if needed, so that they become compliant *by design*. **Run-time** provides the means for the run-time system operation, particularly in terms of policy enforcement, data management, privacy-enhancing tools, and interaction with the data subjects. Finally, the **Monitoring** group deals with process mining and monitoring with the aim to identify discrepancies between compliant and actual behaviour.

Below, the main principles and technical ideas per block are summarised, while the corresponding functionalities are further elaborated upon in the Sections that follow.

### 2.2.1 Governance

Policies are spotlighted at the core of the BPR4GDPR framework, as they comprise the drivers for the compliance-aware process verification and re-engineering, as well as for the run-time operation, providing the behavioural norms of underlying entities. Privacy and security policies are incorporated in the processes already during their specification through the Compliance Metamodel (cf. Section 3.2) or during the verification and transformation phase as a result of the compliance checks. Run-time enforcement is achieved through the Compliance Toolkit (cf. Section 3.3), with policies regulating access to and usage of the underlying resources and prescribing the employment of privacy-enhancing mechanisms.

In that respect, BPR4GDPR focuses on providing a **comprehensive framework for the specification of sophisticated security and privacy policies**, able to capture all complex concepts stemming from the GDPR and other related legislation, and the needs and requirements of all associated stakeholders. Eventually, and fostering legislation-awareness, BPR4GDPR is specifying a **Compliance Ontology** (cf. Deliverable D3.1) providing a high-level codification of GDPR into concepts that need to be taken into consideration by the policy framework, as well as in the context of privacy-aware process re-engineering.

The Compliance Ontology includes, for instance, the types under which personal data fall, roles of the entities requesting and processing personal data, operations and services performed over personal data, attributes of all the involved entities, purposes of requesting/processing data, among others. It also considers the interrelations among the identified concepts and provides for their thorough semantic structuring, by specifying hierarchies reflecting generalisation/particularisation and the inclusion of some types to another.

The Compliance Ontology is extended with policies, formalised as sophisticated and fine-grained access and usage control rules. For the specification of the latter, BPR4GDPR will deliver a **comprehensive Policy-based Access and Usage Control framework** tailored to the needs of highly distributed environments, involving multiple stakeholders, even in cross-border scenarios (cf. Section 3.2).

### 2.2.2 Planning

The Planning part of the architecture offers dedicated components intended to facilitate a) the formal specification of the operational processes of an organisation, as either defined manually by end-users or automatically discovered (cf. Section 5.1), and b) the subsequent automatic verification and, if needed, transformation of such processes, so that they are rendered inherently privacy-aware. To this end, processes, expressed in an appropriate format, are validated and refined by the Workflow Model Verification and

Adaptation modules according to constraints imposed by the Policy Model (cf. Section 3.2) through the **Reasoner**, provided that the capabilities required are made available by the underlying infrastructure.

Given that a process can be roughly seen as coordination of tasks, i.e., operational steps, towards the fulfilment of a more complex business objective, said policies must consist in accurately specifying who performs which operation on which resource during each such step, but also on the information exchanged towards their coordination. Going a step further, in order to achieve adequate expressiveness, additional concepts need to be reflected in the resulting process models, complementing the basic features of tasks and information flow; these include, the use of attributes in the description of all participating entities, as well as context- and situation-awareness, that have been recognised as key enablers in the enforcement of security policies in general [1], providing for the required flexibility and adaptability levels in view of emerging threats. Finally, specific security aspects often pertaining to process execution, like SoD/BoD [2], the data state at each process step (e.g., encrypted vs. unencrypted) and various other kinds of interrelations among process elements need to also be considered.

In order to achieve the above, BPR4GDPR is grounded upon a Compliance Metamodel (cf. Section 3.2), with a view to formally incorporating sophisticated GDPR-oriented provisions. This is based on prior academic work of BPR4GDPR researchers [3] [4], and presents a number of innovative features, including [5]: i) it enables the comprehensive specification of workflow elements, providing extensive coverage of core workflow perspectives [6]; ii) it introduces the novel concept of *assets*, as a means for representing the entities being subject to the execution of workflow tasks; iii) it allows the explicit modelling of both control and data flows, thus being suitable for applications based on either of them or both of them combined; iv) its expressiveness supports the expression of complex and varying security and privacy constraints.

Every process model must be rendered GDPR-compliant before being enacted within the organisation, and this must take place transparently to the user, i.e., the policies being part of its specification, as described above, must be automatically incorporated as part of its design. Therefore, apart from the comprehensive definition of process models, the BPR4GDPR approach towards compliance involves **sophisticated means for the evaluation of a process specification against a number of compliance aspects**. Their main aim is, on the one hand, to control access to, usage of, and flow of information and prevent illegitimate activity, e.g., disclosure of data to unauthorised entities, while, on the other, to determine whether critical tasks are properly included and, if not, impose their execution, referring, for example, to cryptographic operations that must be performed on data before their transfer or storage, approval that must be granted before privacy-sensitive operations, etc. Only after a process has been successfully evaluated against said compliance aspects, may be available for future deployment and execution.

### 2.2.3 Run-time

In order to facilitate the deployment of appropriate technical measures, as required by the Regulation, BPR4GDPR is developing a set of **functional components addressing common needs of stakeholders**, such as cryptographic tools and access control infrastructures. This so-called **Compliance Toolkit** consists of **modular functions** that, fostering “plug and play” to the extent possible, will be **easy to deploy, easy to configure and easy to integrate** within an organisation’s ICT environment, while they will be automatically incorporated to process chains, as a result of re-engineering. The toolkit is complemented by a **Capabilities Registry**, providing information regarding availability of the capabilities (such as functionalities and attributes) of the underlying tools, and a **Context Bus**, that keeps track of the real-time contextual parameters and events, enabling their

efficient circulation. The modules of the Compliance Toolkit fall into three broad families, described in the following.

#### 2.2.3.1 Privacy-enhancing technologies (PETs)

These refer particularly to cryptographic tools, devised for anonymisation and pseudonymisation, data and communications confidentiality, message and information integrity, non-repudiation, as well as enforcement of access rights by cryptographic means.

BPR4GDPR leverages open state-of-the-art tools, as well as prior research results of its partners, such as the advanced cryptographic functions developed in the context of the ReCRED project[1]. To this end, BPR4GDPR employs plethora of cryptographic primitives, both symmetric and asymmetric, together with data-centric techniques, particularly Attribute-Based Encryption (ABE) [7]; this is a novel family of encryption schemes that allows ciphering data with one or more *attributes* representing the characteristics that the recipient has to possess. That is, it is suitable for the cryptographic enforcement of data disclosure policies by leveraging the attributes assigned to entities, being people or systems. A suitable revocation policy is investigated; indeed, revocation of ABE keys is a complex issue often circumvented using time-related attributes on the cryptographic level.

As regards anonymisation and pseudonymisation, and apart from legacy mechanisms, e.g., hashing, BPR4GDPR leverages state-of-art open mechanisms, such as *l*-diversity, *t*-closeness and *k*-anonymity, as well as pseudonyms-based identity management.

#### 2.2.3.2 Data management tools

These are devised for controlling data handling, by means of data access and usage management, including management of retention and storage, pre- and post-processing, etc.

A prominent position is held here by the **Data Management middleware** (cf. Section 6.2), comprising the core Policy Enforcement Point (PEP) of the run-time environment. It is a policy-driven data management and messaging middleware, able to control data collection, processing, storage and dissemination in a fine-grained way, and to handle information flows in a compliant manner. Further, the Data Management middleware constitutes the interoperation Bus among all other tools of the Toolkit, also orchestrating their inter-working towards the application of data protection means. It provides a unified solution for accessing information stored in heterogeneous systems, under a common interface, while providing common, semantic abstractions of underlying components' operational aspects.

#### 2.2.3.3 User-centred tools

The GDPR includes a wide range of existing and new rights for the data subjects, while requiring controllers to provide significantly more information to data subjects about their processing activities and to take into consideration data subjects' preferences as regards data handling. To this end, this type of tools provides for the enforcement of data subjects' rights, including: information and notification; consent management and consideration of own data handling preferences; rights of access, erasure ("right to be forgotten") and rectification; right to data portability.

[1] H2020 ReCRED, From Real-world Identities to Privacy-preserving and Attribute-based CREDentials for Device-centric Access Control, https://www.recred.eu/

### 2.2.4 Monitoring

In the last two decades the focus on process-orientation (e.g., process-aware information systems or BPM systems) has increased, while, with the incredible growth of event data (cf. “Big Data”), it has become possible to use process mining, i.e., a posteriori analysis techniques exploiting the information recorded in the event logs, to discover models and check the conformance of existing ones. Indeed, most organisations have very limited knowledge about the reality happening throughout their day-to-day operation; process mining focuses on this kind of problem, with a view to assessing the organisational reality and reduce the gap between what is supposed to happen and what actually happens. The key facets of process mining are discovery, monitoring and improvement of real processes by extracting knowledge from the organisation’s available data. Previous research has pointed large discrepancies between the idealized model and the process in reality. Moreover, process mining has shown that different models are possible for different and particular views on the process at hand.

In light of the above, BPR4GDPR implements a **Privacy-Aware Process Mining Framework**, based on mature technology brought by the Analytics for Information System Group at Eindhoven University of Technology, particularly ProM[2]. By using ProM, BPR4GDPR seeks to meet requirements related to: *(i) transparency*-- being able to discover and integrate interpretable business procedures into a process model, i.e., to generate process models reflecting, as precisely as possible, an organisation’s current modus operandi; *(ii) compliance*-- automatically identifying “business rules” for different perspectives; and (iii) accountability-- spotting non-conformant executions (cf. Section 5). While checking the conformance between a process model and events in reality, two main concepts should be considered: *real-time data* and *concept drift*.

Streaming process mining techniques can process real-time data in reasonable time. As a result, the modelled process can be used to find the differences between designed and actual models, detecting problems, anomalies and potential frauds. The end result of this effort will be a real-time process mining technique to enable, for example, early warning in automated processes or finding errors or misuse regarding the defined policies.

For non-stationary domains, business rules may become less accurate over time (a concept drift problem) or new factors/requirements may arise, so that the process model will be out-of-date and in need to be adapted/improved. To this end, both active and passive solutions will be provided. The former type will define a **change-detection system** that will update the statistics about the data-related behaviours and will establish rules to integrate recent information to improve the model. The latter will offer **continuous update**, frequently retraining the model based on the most recent observations.

Checking *a posteriori* the compliance of running processes will help to identify discrepancies between modelled and observed behaviour, but also follow and guide process evolution over time, for the benefit not only of legality but also of the affected businesses per se. This becomes even more important considering that such findings may be correlated with other data sources, various context and KPIs, in order to expose vulnerabilities not foreseen at the model level, but also extract information that can only occur through day-to-day practice.

[2] http://www.promtools.org/

# 3 Policy management

In BPR4GDPR, policies and management thereof hold a dual role. On the one hand, they provide the means for system governance, in the sense that they set the rules that regulate the operation of BPR4GDPR components. On the other hand, they comprise the knowledge base that feeds the procedure of process re-engineering, towards compliant by design process models.

In this context, rule-based reasoning takes place upon a semantically rich Information Model, that constitutes the basic pillar of the BPR4GDPR Compliance Ontology (cf. Deliverable D3.1); upon the Information Model, the rules are specified, leveraging a comprehensive framework that allows complex rules to be defined in order to regulate the actions taking place during the system operation.

In the following, we first describe the Information Model, along with the ontology implementing it, followed by the framework for the specification of rules. Finally, the logical architecture of the respective components is overviewed.

## 3.1 Information Model

### 3.1.1 Overview of the Information Model

The day-to-day operation of an organisation involves a variety of entities, like machines, users and data. BPR4GDPR considers two representation levels; the *concrete level* refers to well-specified entities, e.g., named humans, while the *abstract level* enables referring to entities by using abstractions, especially their semantic type and attributes. The main entities of the two levels are summarised in Table 1.

More specifically, at a concrete level, the set of *Users (U)* represents human entities, while this of *Organisations (Org)* describes internal divisions (e.g., departments) or external parties (e.g., sub-contractors). The various machinery comprise the *Machines (M)* set, providing hosting to *Operation Containers (OpC)* that offer *Operation Instances (OpI)*. The latter correspond to actual implementations of functionalities, while Operation Containers bundle collections of Operation Instances provided by the same functional unit. Finally, information comprises the set of *Data (D)*, whereas *Events (E)* take place and may lead to actions for responding thereof.

All above elements constitute instantiations of their semantic equivalents described at the abstract level. Users are assigned with *Roles (R)*, Operation Instances provide implementations of *Operations (Op)*, while data, organisations, machines, operation containers, and events have types, reflecting the semantic class they fall under; thus, sets of *Data Types (DT)*, *Organisation Types (OrgT)*, *Machine Types (MT)*, *Operation Container Types (OpCT)* and *Event Types (ET)* are defined. The semantic model also includes *Context Types (ConT)*, enabling the definition of contextual parameters, *Attributes (Att)*, leveraged for describing properties and characteristics of other elements, and *Purposes (Pu)* justifying access requests, as well as any other type of action that takes place during the system operation.

All concepts summarised in Table 1 comprise graphs of elements that are characterised by relations; the latter are implemented by predicates defining AND- and OR-hierarchies and enabling the inheritance of attributes and rules, as well as the specification of dependencies. For instance, and with respect to the *DT* graph, three partial order relations are defined: *isA*($dt_i$, $dt_j$), *moreDetailedThan*($dt_i$, $dt_j$) and *isPartOf*($dt_i$, $dt_j$), where $dt_i$, $dt_j \in$ *DT*, reflecting the particularisation of a concept, the detail level and the inclusion of some data types to

another, respectively. Figure 3 provides a simple example of the *DT* graph hierarchies, highlighting all three relations.

Moreover, the model specifies the necessary predicates in order to link concepts from different graphs; for example, the predicate *mayActForPurposes*($r$, $\langle pu \rangle^k$), where $r \in R$, $\langle pu \rangle^k \subseteq \mathcal{P}(Pu)$, indicates the legitimate purposes $\langle pu \rangle^k$ for which the users assigned with the role $r$ may act.

| Abstract Level | Concrete Level | Description |
|---|---|---|
| Data Types (*DT*) | Data (*D*) | Data being collected and/or processed, organised according to their semantic types |
| Roles (*R*) | Users (*U*) | Human users assigned with roles reflecting their responsibilities inside an organisation |
| Operations (*Op*) | Operation Instances (*OpI*) | Operations reflect all actions that can take place in the context of the system's operation |
| Operation Container Types (*OpCT*) | Operation Containers (*OpC*) | Components or other functional structures that typically offer a set of operations together |
| Machine Types (*MT*) | Machines (*M*) | Hardware (in the typical case) components hosting operation containers |
| Organisation Types (*OrgT*) | Organisations (*Org*) | The various domains within which actions are performed |
| Event Types (*ET*) | Events (*E*) | Expected or unexpected events that may affect the operation of an organisation, or may call for actions in response |
| Context Types (*ConT*) | Context keys and values | Real-time parameters that should be considered in decision making, such as spatial, temporal, environmental values |
| Purposes (*Pu*) | (no concrete representation) | Purposes for which actions take place, processes are executed, and access to resources is requested |
| Attributes (*Att*) | Attribute keys and values | Characteristics further describing members of the other sets |

**Table 1: Concepts of the Information Model**

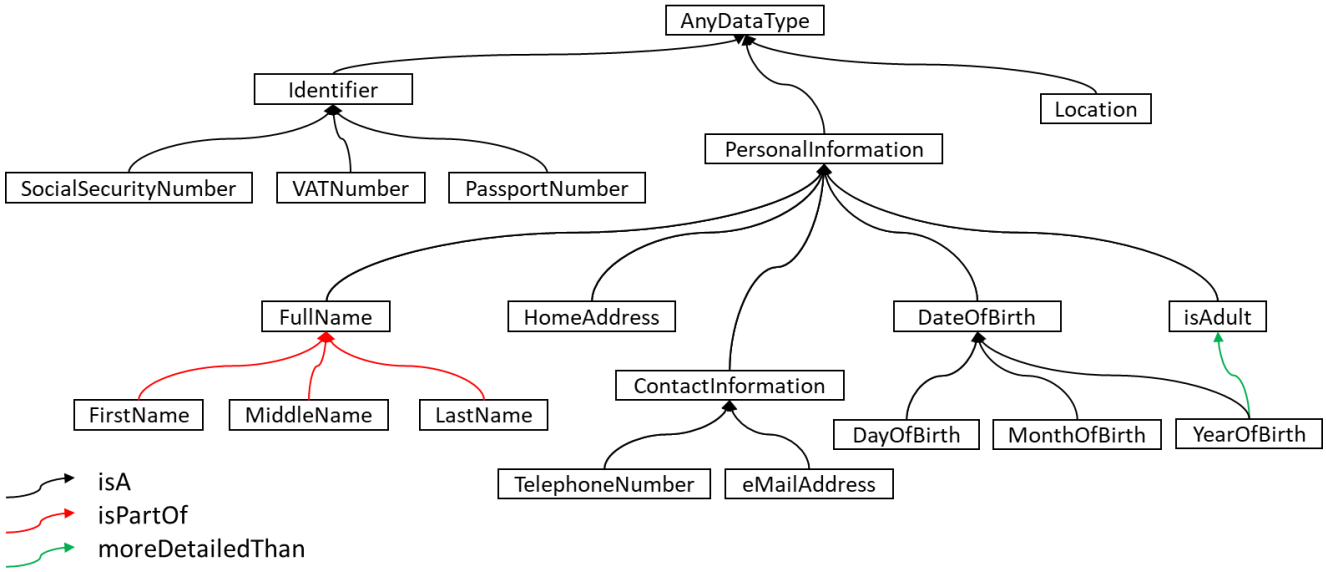


**Figure 3: Information Model hierarchy example**

### 3.1.2 Information Model Ontology

Figure 4 illustrates the Information Model Ontology (IMO), providing the ontological implementation of the Information Model. All abstract concepts described in the previous Section comprise classes, whereas their intra- and inter-class relations are implemented as OWL object properties. In this context, the main classes comprising the BPR4GDPR Information Model Ontology are the following:

- `DataTypes`, which comprises all the types of data that can be used during the system's operation.
- `Roles`, which includes the roles that the system's users may hold.
- `Operations`, which contains all the operations that may take place.
- `OrganisationTypes`, reflecting the different types of organisations, external or internal, actual or virtual, that may be involved in operations.
- `OperationContainerTypes`, representing components that offer a set of operations together.
- `MachineTypes`, containing the types of machines that are used during the system's operation.
- `EventTypes`, reflecting the types of events that may affect the operations and/or may require some action to take place in response.
- `ContextTypes`, serving for the specification of run-time constraints.
- `Attributes`, providing for the description of properties that characterise the entities.

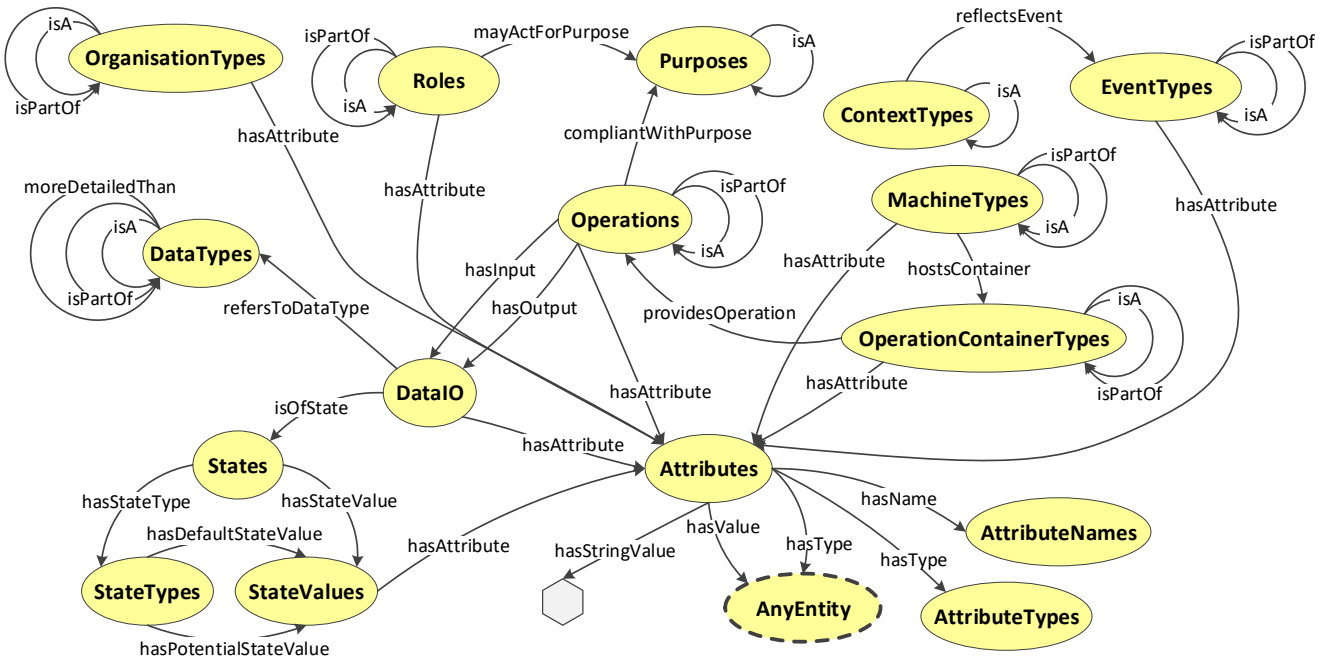


**Figure 4: Information Model Ontology**

These classes are complemented by additional classes holding a rather “auxiliary” role; these enable, for instance, to define the inputs and output of an operation, the state of data as regards some of their properties (e.g., anonymised or not), etc.

On the other hand, the main intra-class properties are `isA`, `isPartOf` and `moreDetailedThan` that, along with their inverses[3], essentially comprise AND- and OR- hierarchies, enabling inheritance, as well as dependencies specification. Associations between concepts of different classes are implemented by mean of inter-class OWL object properties; for instance, the roles that may act for a purpose are indicated by the `mayActForPurpose` property, whereas the attributes characterising a concept are related to the concept by means of the `hasAttribute` property.

## 3.2 BPR4GDPR Policy Framework

### 3.2.1 Actions

Similar to the *subject—verb—object* linguistic pattern, anything that takes place during the function of a system, can be seen as an *operation* of an *actor* over a *resource*. This applies at any granularity level: at the highest level of a business process, a set of actors performs an aggregated super-operation, consisting of elementary operations, over a set of resources; at a low level, a barcode reader executes a "read" operation over a barcode. Given that a series of such activities, such as a business process, may be executed across different organisations, as in the case of cross-domain processes, the *organisation* within which something takes place comprises an important contextual aspect.

Thus, security and privacy policies are intuitively centred around conceptual quadruples of {*actor*, *operation*, *resource*, *organisation*}; in BPR4GDPR, each quadruple as such is characterised as an *action*, providing the fundamental element for the definition of rules. Actions are formed leveraging the entities of the Information Model; therein, operations and organisations are explicitly defined, whereas different types of entities may play the role of actors and resources, thus be members of the corresponding *Actors* (*A*) and *Resources* (*Res*) sets. Although there are some obvious patterns (e.g., users are typically actors and data are always resources), which entities can be actors and/or resources strongly depends on each organisation, and operational aspects and modelling choices thereof.

More specifically, an action $act \in Act$ is a tuple $\langle a, op, res, org \rangle$, such that: $a \in A$ is an actor; $op \in Op$ is an operation; $res \in Res$ is a resource; and $org \in Org$ is the organisation within which an action takes place.

An action can be either *atomic* or *composite*, depending on whether the associated operation can be decomposed to more elementary operations or not, following the hierarchical relations in *Op*. Actions are also categorised to *abstract*, *concrete* and *semi-abstract*, depending on whether actors and resources are defined at abstract, concrete or mixed level.

Further, it should be stressed that the elements of an action can be specified as *enhanced entities* that include, apart from the entity's semantic type, expressions over its attributes and/or sub-concepts, thus refining the concept definition, towards specifying attribute-based constraints and access and usage control rules.

### 3.2.2 Access and usage control rules

Upon the concept of actions, access and usage control rules are specified; they are defined as *permissions*, *prohibitions* and *obligations* over actions and, since actions can be abstract, concrete or semi-abstract, rules

[3] Inverse properties are explicitly defined for all object properties in the ontology, in order to ease navigation from one ontological element to another.

are also specified at these three levels. The format of rules in BPR4GDPR is illustrated in Figure 5; as shown, a BPR4GDPR rule consists of the following elements:

- *action* describes the core of the rule, i.e., what action the rule by definition permits, prohibits or obliges to take place.
- *purpose* reflects the overall objective behind data collection and/or processing; in fact, an operational decision cannot be taken regardless this parameter, which can significantly differentiate an entity's behaviour.
- *pre-action* reflects actions that should have previously taken place in order for the rule to be activated; as an example, there can be the case where a patient should have provided explicit consent (pre-action) in order for her medical record to be processed for research purposes.
- *post-action* similarly implies anything that needs to take place after the enforcement of a rule; for instance, a rule may permit reading some data for providing a service, but may additionally require that the data are deleted immediately after.
- *context* describes conditions defined over "environmental" properties and states, as well as events.

In other terms, an *access and usage control rule* is a structure:

$$\left.\begin{matrix} Permission \\ Prohibition \\ Obligation \end{matrix}\right\} (pu, act, preAct, cont, postAct)$$

where $act \in Act$ is the action that the rule applies to; $pu \in Pu$ is the purpose for which $act$ is permitted/prohibited/obliged to be executed; $cont \in \mathcal{P}(ConT)$ is a structure of contextual parameters; $preAct \in Act$ is a structure of actions that should have preceded; $postAct \in Act$ refers to the action(s) that must be executed following the rule enforcement.

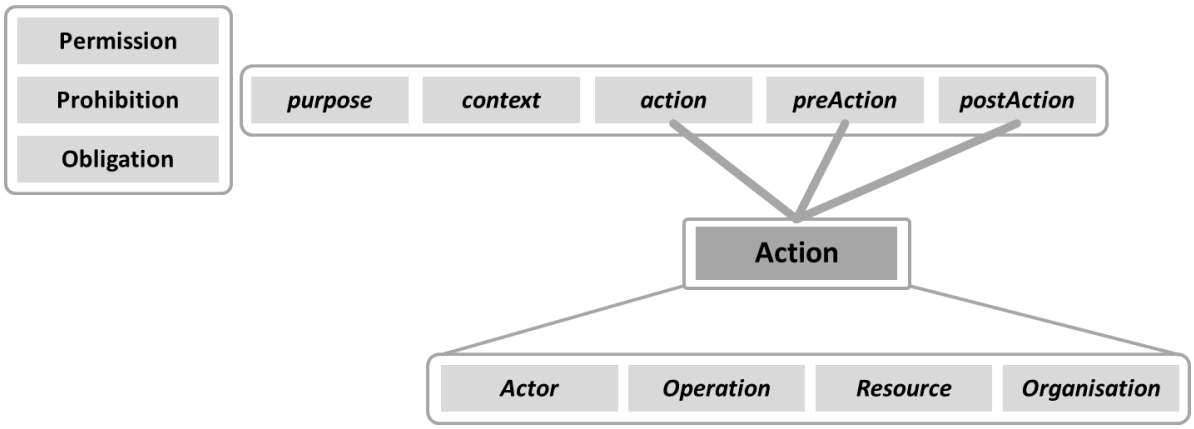


**Figure 5: BPR4GDPR rules format**

### 3.2.3 Policy Model Ontology

The rule-based framework described above is implemented as a semantic ontology, referred to as the Policy Model Ontology (PMO). It is built on top of the Information Model Ontology, and comprises ongoing work of project Task 3.2 "Rule-based access and usage control".

Rules are implemented as instances of the `Permissions`, `Prohibitions` and `Obligations` classes (sub-classes of the `Rules` class), whereas actions are implemented as `Actions` class instances, with $\langle a, op, res, org \rangle$ being reproduced by means of the corresponding object properties. Within an action, the

actor, operation, resource and organisation are defined at either the abstract or the concrete level; in that respect, for the representation of an action's elements at the abstract level, instances of the `EnhancedEntities` class are leveraged, constraining the referenced IMO semantic type with respect to its attributes and/or sub-concepts, while for the concrete level, the aforementioned properties point at instances of the class `ConcreteEntities`.

Two useful mechanisms for defining constraints upon actions' elements, and also achieving rich expressiveness in general, are *expressions* and *logical relations*. The latter, implied by thick lines in Figure 6, allow specifying logical structures of concepts. Expressions enable the definition of contextual conditions and constraints on concepts (e.g., on an actor's attributes); they comprise ternary relations assigning a value to a subject through an operator, or logical structures of such triples.

`Actions` instances are used for specifying the main action and pre- and post-actions of a rule; in the latter cases this association is indirect, with the `RequiredActions` class mediating and enabling the specification of time and sequence constraints. Beyond such constraints, the BPR4GDPR approach incorporates a mechanism for combining actions, so as to form complex structures thereof, referred to as *skeletons*, following various sequence patterns. Finally, dependencies among all the entities comprising the actions of a rule enable the specification of advanced SoD and BoD constraints, instead of relying only on role-/user- centric constraints.

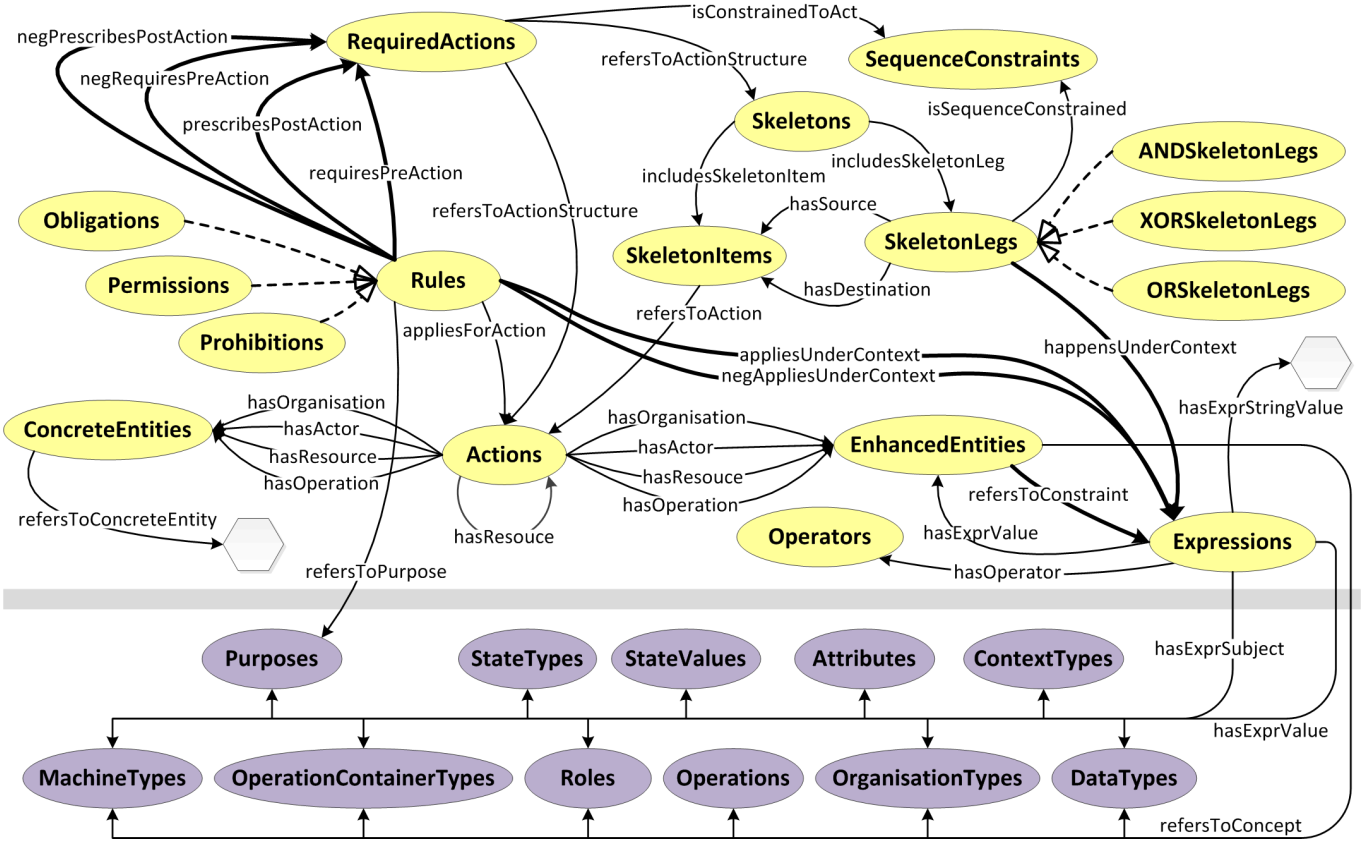


**Figure 6: Policy Model Ontology**

## 3.3 Governance architecture

The rule-based framework presented in the previous section is the core BPR4GDPR component as regards governance of the whole system. In order to enable the exploitation of the full potential anticipated by the framework, appropriate reasoning mechanism is necessary that will allow extracting the knowledge contained in the rules. Indeed, much of the knowledge codified in the BPR4GDPR rules is not readily available, but has to be inferred, even in real-time. Therefore, the governance module of the system is being designed with a dual focus: on the one hand, to be able to extract precise information from complex —sometimes even conflicting—rules, and, on the other hand, to be efficient for coping with stringent performance needs of real-time operations.

In Figure 7, the logical architecture of the governance component is sketched. As illustrated, it internally consists of the following modules:

**Models Repository**. This module provides hosting to the two fundamental ontological models, i.e., the Information Model Ontology and the Policy Model Ontology. In this context, it constitutes an RDF Triplestore, undertaking all functions regarding the management of ontologies.

**Model Handlers**. This module encapsulates the ontologies, providing a convenient API that makes the underlying models accessible by the other components, particularly the Reasoning Beans and the Management Interface (see below). In this context, it offers all functions necessary for the management of the models, such as graphs' configuration and rules' administration.

**Reasoning Beans**. Reasoning constitutes the most essential part of a rule-based system, as it implements the knowledge extraction and decision making functions. In BPR4GDPR, the underlying intelligence is offered by the Reasoning Beans, that implement the algorithms for answering all queries regarding authorisation and compliance. They constitute the Policy Decision Point (PDP) of the system, comprising a dynamic set of stateless engines —for improved performance— that extract knowledge from the ontological models and feed the other components with instructions, addressing either authorisation decisions or compliance patterns and guidelines.

**Offline Policies**. Motivated by the need for improved performance during real-time operation, BPR4GDPR leverages the paradigm of offline reasoning, i.e., proactive extraction of knowledge contained in the access and usage control rules. Through the Offline Reasoning procedure (to be documented in the Deliverable D3.2 "Initial specification and prototyping of the policy framework"), all the required knowledge becomes available already by the request time, thus reducing the number of queries to the ontologies and offering performance gains. In other words, all the heavy processing tasks are performed offline and only when the PMO is updated, for instance, when new rules are added or existing ones are revoked.

In order to interact with the other BPR4GDPR components, the governance module provides three interfaces, devised for, respectively, answering authorisation queries, responding to compliance-related queries, and management aspects of the ontological models.

**Authorisation interface**. This interface serves the fundamental need of providing knowledge related to authorisation to other components of the BPR4GDPR ecosystem, particularly the Data Management Bus that holds the role of being the primary Policy Enforcement Point.

**Compliance interface**. This interface is devised to provide other components, particularly the planning environment (cf. Section 2.2.2), with instructions as regards the adaptation of processes and operations in order to become compliant, as part of the verification procedure.

**Management interface**. Through this interface, the appropriate functions are provided for the management of the Information Model and the Policy Model. This includes the configuration of the graphs and the specification of the rules. In order to support different user interfaces that could provide the front-end for humans, it is anticipated to be accordingly implemented, so that different solutions can be built and interact in a loosely-coupled manner.

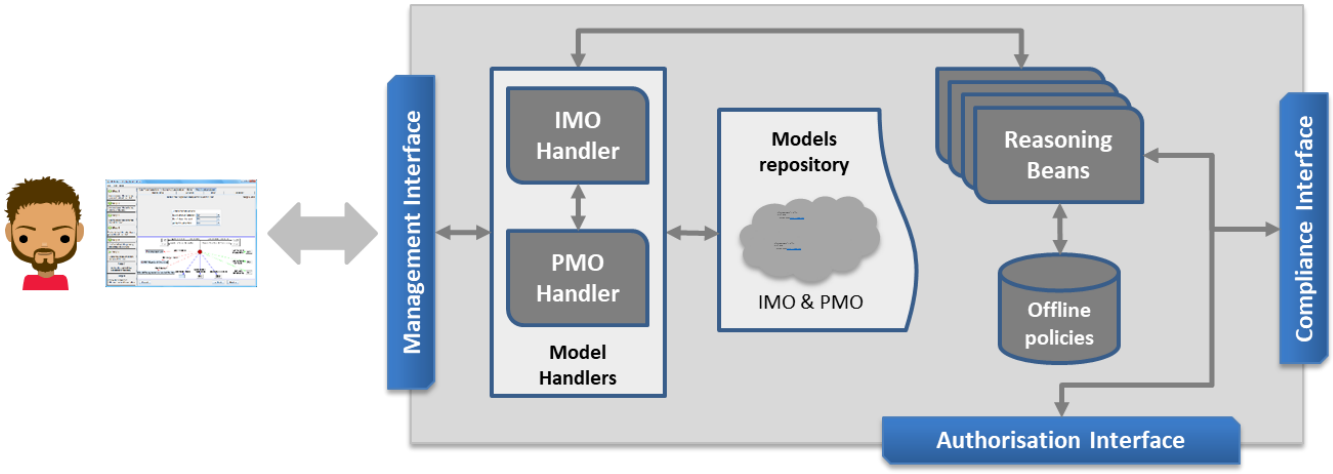


Figure 7: Governance component logical architecture

# 4 Process management

## 4.1 Process modelling

A vital step towards any organisation's regulatory compliance is the accurate modelling of privacy-aware processes, offering precise insight into what is being or must be executed as part of the corresponding operational procedures. For this purpose, the BPR4GDPR Planning Environment offers the appropriate tools (cf. Section 2.2) that will, on the one hand, allow their description in a way that will effectively guide their execution and integration into the organisational environment, and, on the other, be expressive enough, in order to be able to capture the associated provisions and incorporate comprehensive security and privacy policies in their design.

These tools, comprising the Workflow specification engine, provide for formalising each operational process based on a **Compliance Metamodel**; the latter is defined with the use of ontologies in a way suitable for formally incorporating sophisticated GDPR-oriented provisions *in-design*.

At a high level, the most fundamental artefacts of the Compliance Metamodel are *tasks* and *flows*. The former represent actions to be executed within the workflow, each describing the *operation* performed by an *actor* on an *asset*. Flows express dependencies between tasks, are represented through directed edges and are of two types: *control* and *data*. A control flow dependency $t_A \xrightarrow{f_C} t_B$ between two tasks $t_A$ and $t_B$ means that $t_B$ is executed only after the execution of $t_A$ is completed; what the edge transfers is the thread of control, potentially accompanied by the necessary control parameters. On the contrary, a data flow dependency $t_A \xrightarrow{f_d} t_B$ assumes both tasks continuously under execution, with $t_B$, however, being dependant on the stream of data produced by $t_A$. Further, a workflow model is complemented by the operational *purposes* it is meant to serve, and the potential *initiators*, denoting entities authorised to initiate the workflow.

The core part of the Compliance Metamodel Ontology (CMO) is shown in Figure 8. All concepts participating in a process specification are represented by instances of the corresponding classes, while OWL object properties model their relationships with each other and with IMO elements (cf. Section 3.1). Every process is represented as an instance of the WorkflowModels class, comprising its reference semantic entry; this is associated to sets of Initiators and `WFPurposes` individuals. Tasks and edges are represented through the classes `TaskNodes` and `Edges`, respectively, while the latter is further subclassed by `DataEdges` and `ControlEdges`, denoting the two types of flows, $F_D$ and $F_C$.

The core constituents of tasks are actors, operations and assets, while for flows, the exchanged information is essential for edges definition. Despite the semantic and structural differences of actors, assets, operations and information, the corresponding entities share common features and therefore, to some extent, a uniform representation; thus, the associated CMO classes are all subclasses of `EnhancedEntities`. Each instance indicates the entity's semantic type and the constraints that describe said abstract entity, through the `refersToConcept` and `hasConstraint` properties, while if the entity is defined concretely, the `refersToConcreteEntity` property is used instead. Information entities, used for flows specification, are further enriched with *states*, serving as indicators of the effect that the execution of preceding tasks has had on each information entity. A data state is characterised by a *type* and a *value*, incorporated in the

information model as the sets *ST* and *SV*, being the instances of IMO classes `StateTypes` and `StateValues`.

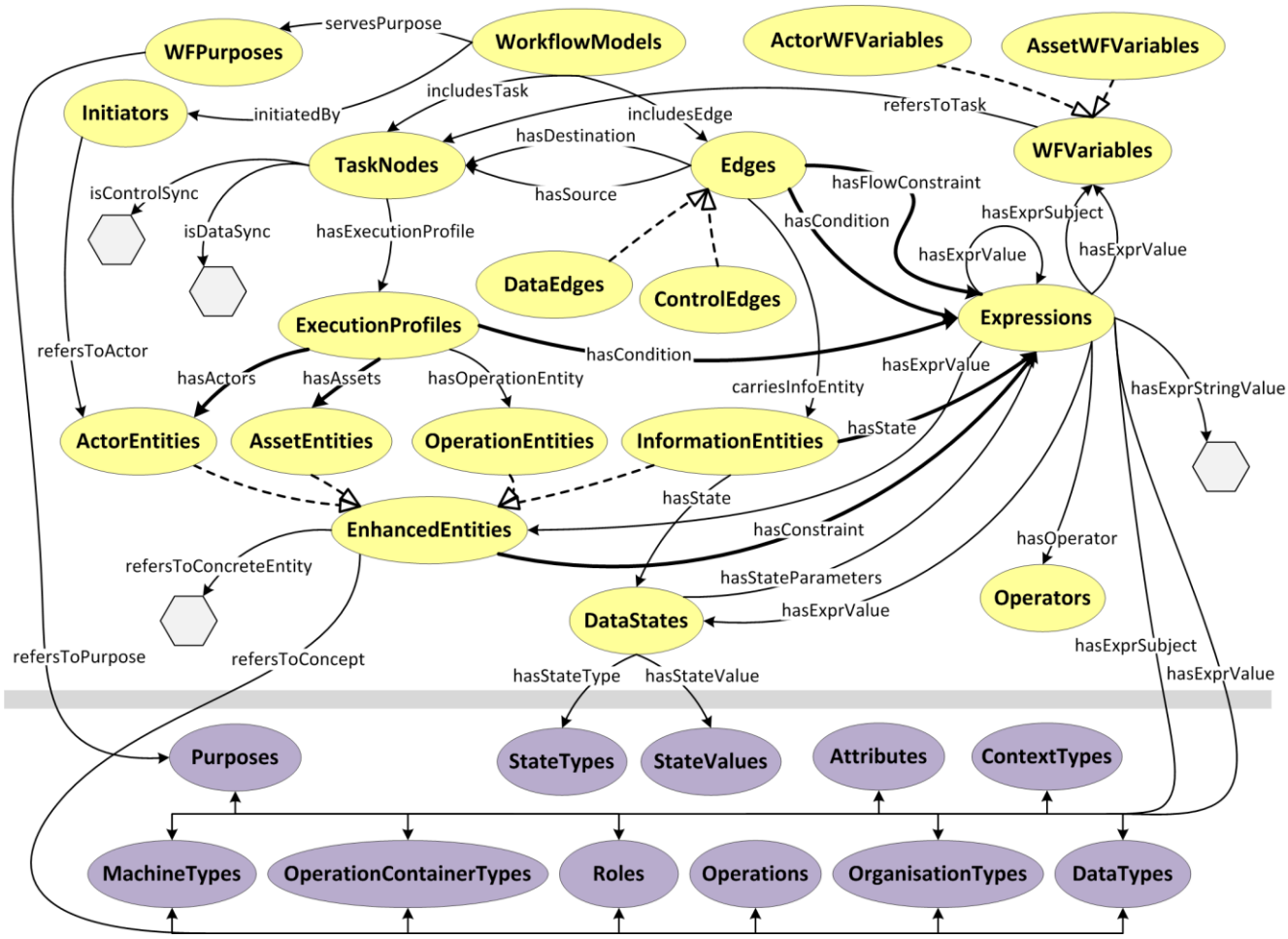


**Figure 8: Compliance Metamodel Ontology**

A particularity of the proposed modelling approach is the concept of *execution profiles*, enabling the specification of variations regarding the execution of a task. This concerns two aspects: differentiated execution based on some *conditions*, and capturing the dependencies between the task's actors, assets and operation constraints, that is, precisely defining their valid combinations. Task conditions describe real-time constraints external to the workflow specification (e.g., contextual factors), or spanning beyond task boundaries, that cannot be expressed on the basis of referenced entities' attributes alone. Execution profiles are modelled through individuals of the `ExecutionProfiles` CMO class, appropriately linked to `EnhancedEntities` (sub-classes) instances. Task conditions are indicated through the `hasCondition` property, while the `hasExecutionProfile` property associates a `TaskNodes` instance with its `ExecutionProfiles`. Finally, `isControlSync` and `isDataSync` properties indicate, where necessary, the synchronisation behaviour of the task. Regarding modelling of flows, data and control edges share the same characteristics: each connects two tasks and denotes the flow direction, the information exchanged, the underlying conditions, and other flow properties; the distinction between them stems from the semantics of the connected operations regarding the manner they receive and consume information.

As a final remark, aforementioned conditions (e.g., contextual) and constraints on concepts are defined through *expressions*; these comprise ternary relations assigning a value to a subject through an operator, or logical structures of such triples. Logical structures in general are modelled through *logical relations* and can be used to cover more complex cases. For instance, a task may not be assigned to one type of actor; its definition may include a set of heterogeneous entities that must jointly undertake its execution (AND), a set of alternative actors, inclusive (OR) or exclusive (XOR), or combinations thereof. In Figure 8 possible usage of logical relations is indicated through thick lines. Additionally, *workflow variables* add to the overall expressiveness, providing reference "handles" serving as abstractions of the underlying objects, thus allowing to describe relative relations and dependencies. For instance, a constraint such as "the actor of task $t_B$ should be the same as the one of $t_A$" can be defined, whatever the actors of $t_A$ and $t_B$ are.

## 4.2 Process re-engineering

Process specification aside, the other fundamental duty of the Planning Environment is the verification and adaptation, when necessary, of the processes, as they are expressed through the Compliance Metamodel. To this end, appropriate interaction with the Reasoner (cf. Section 3.3) exploits the knowledge encoded in the Policy Model; this interaction results in a set of compliance Directives that are tailored to each specific process at hand and control a number of aspects that have to do with the fact that, in the context of a process, if and how individual tasks are executed is not adequate, calling for an approach that will be able to guarantee compliance at a larger scale. Such aspects can be indicatively summarised in the following:

- Purpose compliance: It must be ensured that the process as a whole is relevant and consistent with a purpose, while the purpose itself should not contradict with the access rights of the person initiating the process.
- Each task is further evaluated in more detail regarding the enforcement of rules that control task sequence, access on operations and resources, input and output data of each task and the flow of data between any subsequent tasks. In that respect, important aspects that need to be taken into consideration during task verification include the following:
    - the system must be able "by definition" to offer the respective capability
    - the initiator must have the right to include the task in the process
    - each actor must have the right to execute the corresponding operation to the associated resource
    - the task must not conflict with precedent and subsequent tasks
    - the task may require the prior or subsequent execution of other tasks

Needless to say that through all of these checks, all elements constituting a task must be considered, i.e., since a task comprises an actor, an operation and a resource, task sequences must be evaluated in terms of allowed combinations among all these elements. When through any of these checks violations are discovered, the system must be able, whenever possible, to come up with modifications to the original process, that will render it acceptable. Such modifications can be the removal or substitution (e.g., in the case a task is prohibited), or the addition (e.g., in the case of complementary actions needed) of certain tasks. Furthermore, they may concern more complex alterations, such as entire task sequences (paths) or patterns to be followed in parts of or the entire process. Notably, such modifications are especially important with respect to privacy preservation, as they may concern the addition of tasks that perform data transformations necessary, so that the data flow in the user-defined process is compliant with the corresponding access control requirements.

Apart from information available in the process under verification, access control decisions often depend also on contextual information, that can only be evaluated at run-time. In that respect, the process must be designed in such a way that it is ensured that the values of the corresponding contextual parameters will be checked during execution and the process will proceed accordingly. This can be achieved, e.g., through execution profiles or conditional branching of the affected parts of the process, based on the parameters that the policy model demands, whereby all alternative paths are pre-specified and validated at design-time. This way, dynamic constraints can also be taken into account during the design phase, so that the chances that the process is executed correctly are maximised. Such constraints can be included in the process design as policies that are generated through reasoning over the policy model and attached to the related tasks. Towards further enhancing *flexibility by design*, alternatives are also being investigated, in order to address more complex cases without the need for over-specification and subsequent burden on readability and maintainability of process models.

# 5 Process monitoring

In this section, we explain the basic architectural concepts of the process monitoring component of BPR4GDPR. We begin by explaining the process mining module that aims at initially discovering the process model out of the event logs collected through the interface with the data management component. In a following step, process mining performs a conformance checking between the discovered process models and the event logs by including additionally the rules through the interface with the policy management component. In the process monitoring module, a repair of the process models is done based on the degree of conformance with the rules that represent both the business rules and also the GDPR regulations.

## 5.1 Process Mining

Process mining is performed to discover, monitor and improve real processes (not assumed or modelled processes) and is based on the knowledge extracted from the event logs of information systems. Most organisations have very limited knowledge about the reality happening throughout their day-to-day operation. Process mining focuses on this kind of problem, with a view to assessing the organisational reality and reduce the gap between what is supposed to happen and what actually happens.

In fact, process mining caters to the Phase 1: Process identification of the process lifecycle explained in Section 2.1. Referring to Figure 9, the two main "first class citizens" of process mining [8] are (business) process models and the event log recorded in information systems of the organizations. On the basis of the event log of an organization, process discovery techniques [8] of process mining discover the way different operations (processes) are performed in reality (usually in contrast to the desired way). Formally speaking, let $L$ represent a log file that is a sequence of events $e_i$ each belonging to a particular case $c_j$ of the business process. An event $e_i$ represent an activity performed as part of the business process $P$. Usually, such an event is represented by the activity $a$ performed, the timestamp $t$ of the event and the case $c_j$ to which the event belongs. Process discovery algorithm $\alpha$ takes the sequence of events $e$ as input and discovers a process model $M$, i.e., $\alpha: e \rightarrow M$.

The conformance checking techniques [8] [9] of process mining confronts the organizational event log with the process model. These process models are either discovered through the process discovery techniques or chalked out by process experts or stakeholders. The aim of this comparison is to discover the deviations existing in the business operations between what stakeholders think they are doing and that what they are *really* doing. The discovered deviations are attributed to discrepancies in the operational process of the organization in case the business model is considered normative. Conversely, if the considered model is descriptive then the discovered deviations presumably indicate discrepancies in the business model. Formally, let's consider the event log to be a multiset of traces $\sigma$ such that each trace represents a sequence of related events in the log, representing a particular case $C_i$ of the process $P$. State of the art alignment-based conformance checking techniques calculate the individual fitness score by calculating optimal alignment $\lambda^M_{opt}$ and worst-case alignment $\lambda^M_{worst}$ for each trace $\sigma_i$ in model $M$through function $fitness(\sigma, M)$ as:

$$fitness(\sigma, M) = 1 - \frac{\delta(\lambda^M_{opt})(\sigma)}{\delta(\lambda^M_{worst})(\sigma)}$$

and extending the trace-level fitness to log L level fitness $fitness(L, M)$by

$$fitness(L, M) = 1 - \frac{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda^{M}_{opt})(\sigma)}{\sum_{\sigma \in L} L(\sigma) \times \delta(\lambda^{M}_{worst})(\sigma)}$$

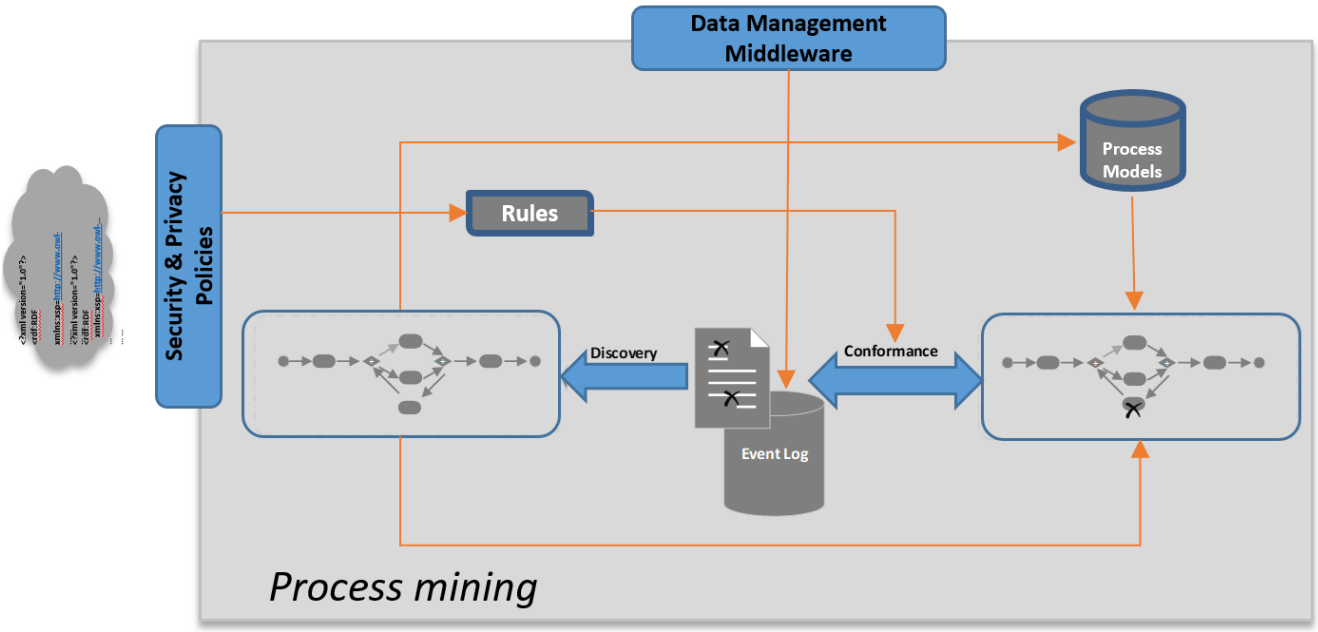


**Figure 9: Process Mining Tool Architecture**

The third important category of process mining techniques, are targeting process enhancement [8]. Such techniques take into account the data recorded in event logs in addition to the activity name $a$ and timestamp $t$, for instance, the organizational resource $r$ executing the activity, and provides useful insights for enhancement of the business process in terms of efficiency and efficacy.

Positioned in the Monitoring block of the BPR4GDPR system architecture of Figure 11, and essentially based on the mature techniques developed as part of the PROM framework [10] [11], process mining tools will provide two main functionalities. The process discovery component will provide mechanism for discovering the real modus operandi, in contrast to the documented process or the process perceived by the stakeholders also referred to as *to-be* model. These are extracted from the event logs of the organization in the form of process model(s) also referred to as *as-is* model (cf. Figure 9). Process conformance checking component on the other hand will provide mechanisms for conformance checking of the business process model, either discovered by the process discovery component or devised by process experts, against a variety of business rules, enforced either by the organization itself or security and privacy policies devised by policy management module on the basis of GDPR regulations and accordingly highlight deviations in the execution of the process with respect to the business rules or security and privacy policies.

In context of the BPR4GDPR framework architecture, process mining tools will in essence be provided as plug-in components as is the case with other Privacy-Enhancing tools and Data Management tools. Process mining tools will be having interface with the Data Management Middleware for communication with rest of the framework. The Data Management layer, being the bridge between the process mining tools and the organizational data sources, will provide the extracted (events) data from the relevant data sources to the data mining tools, as demonstrated in Figure 9.

The resulting output models in case of process discovery or conformance analysis statistics in case of conformance checking will provide insights on the actual process to the stakeholders for discussion and accordingly adaptation if needed, as well as take notice of critical deviations detected by the tool. Additionally, the results of conformance checking steps will be used for model repair in the process monitoring phase (cf. Section 5.2).

## 5.2 Process Monitoring

In light of the introduction to process mining as mentioned in Section 5.1, a process monitoring framework will be implemented in the BPR4GDPR context based on various process mining techniques.

A short description of the functionality of the Process Monitoring core modules is given below.

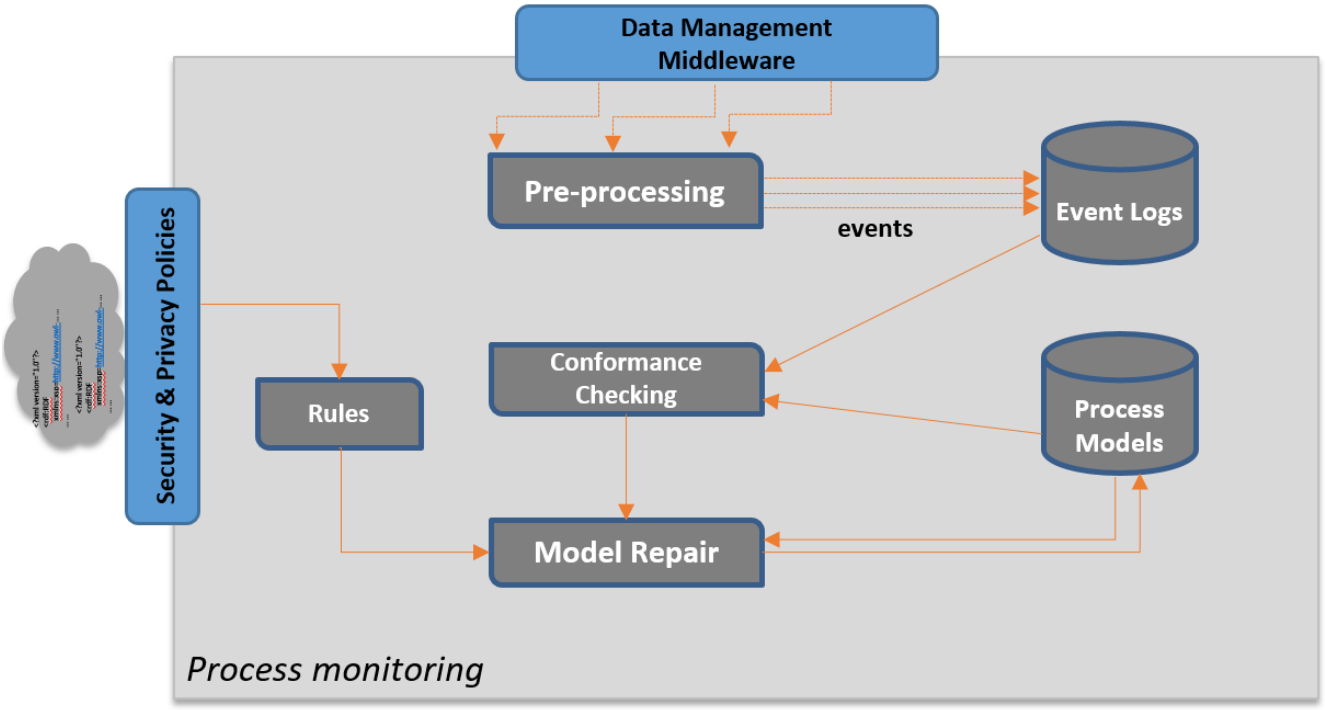


Figure 10: Process monitoring high-level architecture

Figure 10 shows the interaction of Process Monitoring module with various BPR4GDPR components and its main components. This module has mainly four subcomponents, these include pre-processing, conformance checking, rules and model repair.

- The **pre-processing subcomponent** constantly receives the data from *Data Management middleware* (cf. Section 6.2) as an input and updates and produces standard event logs for process mining.
- The **conformance checking subcomponent** is fed with the updated logs and the previous discovered process models. In conformance checking, an existing model is compared with the model discovered from the event log. This technique assesses whether a real process discovered from the log conforms to the assumed and predetermined process. Conformance checking can be applied to different aspects of the business process. This subcomponent provides deviation results that are essential for the model repair subcomponent.

- The **rules subcomponent** gets the policies from *the Security & Privacy Policies module* (cf. Section 3) and converts these policies to a set of access control rules.
- The **model repair subcomponent** receives the to-be-repaired process models, the converted rules and the results of conformance checking as inputs. With this, the subcomponent is able to identify which parts of the process model are not compliant with the rules. The goal is to find the minimal set of changes to repair the model such that the repaired model is now compliant with the rules and, ideally, still represents the previous observed behaviours.

The end result of this module will be the output a set of process mining techniques [12] [13] [14] to enable, for example, an early warning in automated processes and/or finding errors or misuses regarding the defined rules and policies.

# 6 BPR4GDPR toolkit

The co-existence of many different technical aspects regulated by the GDPR makes the definition of the architecture of BPR4GDPR toolkit not an easy task since it needs to present an extreme flexibility in order to fit the different needs coming from heterogeneous use cases.

For this reason, in the toolkit architecture definition, we focused on these three major aspects: abstraction, integration and deployment.

Abstraction should be done in a way to be enough generic to address cross-sectorial GDPR-related issues but at the same time that can provide an easy-to-use solution to practical problem without any major adaptation to cast in specific application domain.

Integration is another fundamental aspect as some tools will severely impact the way existing system run (e.g. how a company stores their assets in the cloud in a privacy preserving manner). Typical production environment is a complex and delicate ecosystem, with company proprietary technology that reflects company culture and its choices during years. Therefore, to foster potential integration in the operating workflow, tools should be as much as possible *self-contained*, *embeddable*, able to working in a *standalone* mode as well as *safely be integrated* with company assets such as storage or database. Company assets should be abstracted in order to design tools that are not bound to a specific technology (e.g. a specific type of database); conversely, an adapting layer is defined to cope with this issue, allowing tools to be generic and portable in different operating environment.

Finally, fast and easy deployability is a key issue. Building tools that are easy to deploy allows rapid test and foster usage inside companies' premises.

## 6.1 Overview & System requirements

We evaluated technical requirements with respect to an initial set of tools that deliberately cover different and heterogenous technical aspect and thus involve different system requirements.

From this bottom-top approach, we design these different tools (detailed in section 6.4), whose motivation is coming from different use cases:

- Risk assessment tool
- Cloud data management tool
- Anonymization/pseudonymization tool

The tool architecture should thus provide:

- Fast deployability for demo/studying purposes
- Modularity: subset of network/service should be easily separated to be integrated in different environment
- A way to develop distributed infrastructure
- Machine-to-machine service interface (APIs) to interconnect with company legacy service
- An interface to system administration
- An optional interface to be chained with other service

Moreover, all the tools should offer capability and interfaces in order to be used as a standalone service, and/or being integrated by a workflow orchestrator, and/or to allow third party integration.

Tools must access company assets in a standard form, abstracting, for instance, whether a file is accessed on a local storage or on a SAN/NAS and being controlled in what they can access in terms of company resource.

At the same time, tools should share common assets.

To this aim we rest in an architecture composed by two entities:

- A Data Management Middleware bus that act as a data-proxy, providing access control and being regulated by PDP. This part is described in section 6.2.
- The BPR4GDPR tool that accesses the company resources through the previous system and offers an easy and fast deployable solution to cope with some specific GDPR aspect. This part is described in section 6.3.

## 6.2 Data Management middleware

### 6.2.1 Overview

Data Management is a generic policy-driven middleware for handling data access and usage requests employing the necessary mechanisms for achieving this goal, like, e.g., data encryption, anonymisation, pseudonymisation and aggregation; control data collection, pre- and post-processing, storage and dissemination in a fine-grained way. This middleware provides application-independent support, being sufficiently flexible to support a range of concerns and environments, through the transformation of the Platform Independent Model (PIM), as defined by means of semantic ontologies, to a Platform-Specific Model (PSM), while it acts as Policy Enforcement Point (PEP) enforcing the fine-grained access and usage control policies specified by means of the Policy Model Ontology. For achieving these purposes it utilises knowledge extracted through reasoning, policy decisions as well as privacy-enhancing mechanisms and user-centric tools expressing data subjects' rights. For the encompassed Policy Engine, the BPR4GDPR provided engine can be use, although the solution allows usage of different engines providing equivalent functionality.

Figure 11 and Figure 12 show the interaction of Data Management middleware with the rest of core BPR4GDPR components. More specific, Figure 11 illustrates the case a Data Access tool requests access to some data stored, for example, in a database, where no pre- or post-processing is needed (as result of PDP decisions).

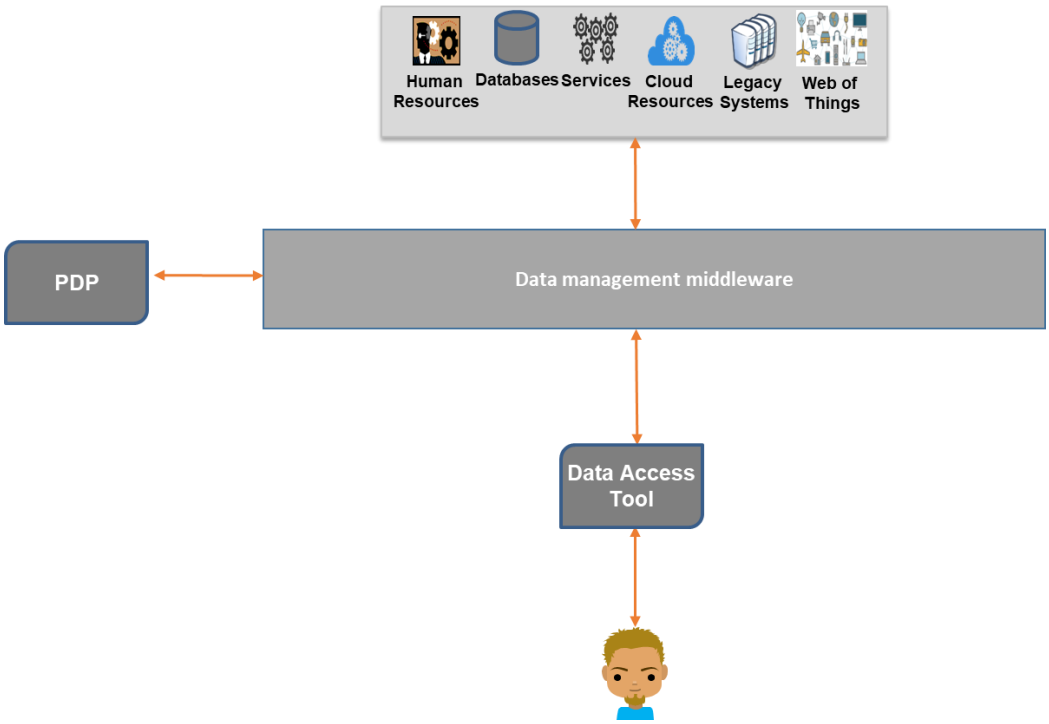


Figure 11: Data access use case

Figure 12 shows the case of an application request data that, according to Policy Engine decisions, can only be provided anonymized. Thus, the middleware uses the appropriate tool in order to provide the requested anonymization functionality.

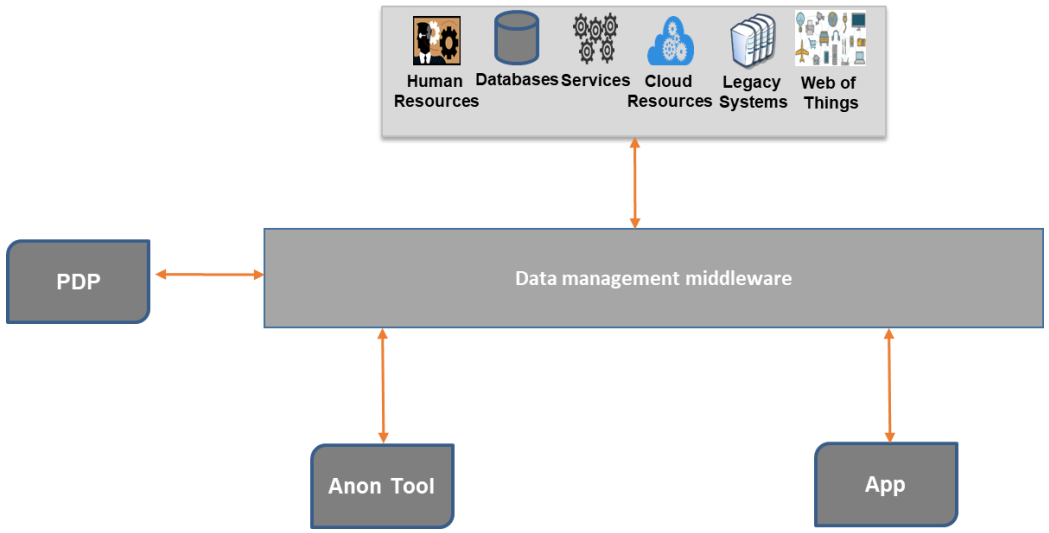


Figure 12: Data anonymization use case

In order to support the aforementioned functionality the Data Management middleware is built upon the ontology-based Semantic Message Broker brought to BPR4GDPR by Singular Logic, and it enables handling information flows in a compliant manner. Figure 13 shows its high-level internal structure.

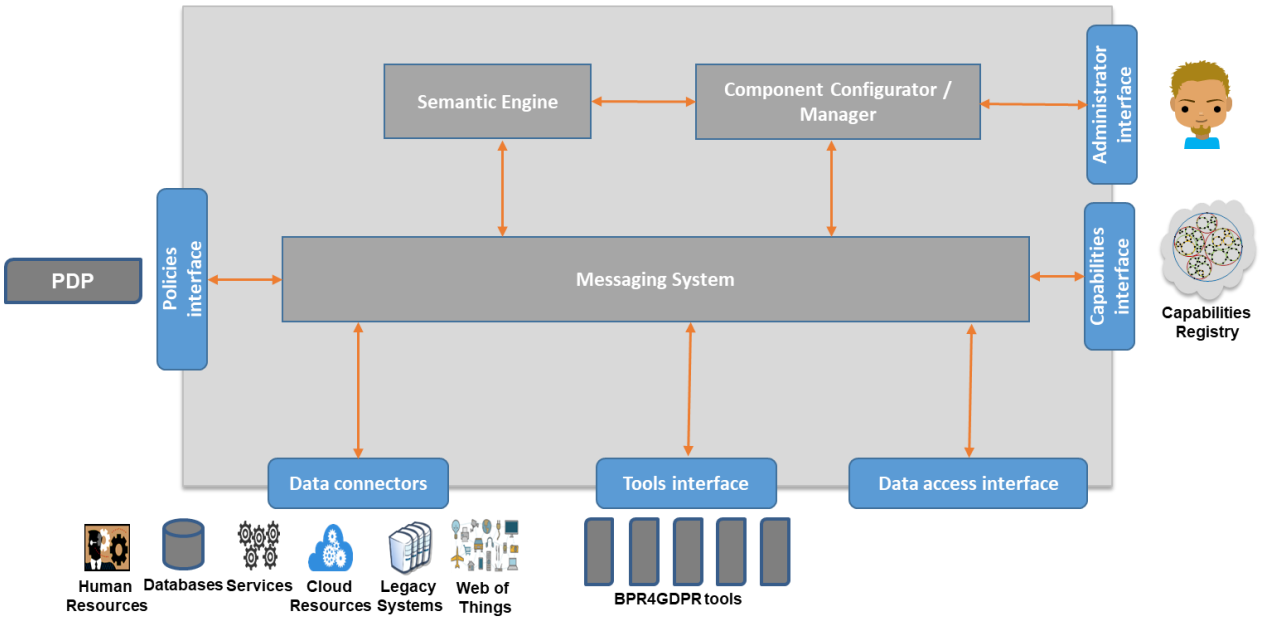


Figure 13: Data Management high-level architecture

The Data Management internal structure as well as the provided interfaces will be specified into greater detail in deliverable D2.4 (Final Specification of BPR4GDPR architecture) as well as deliverables of Task 5.2 (Data management tools).

### 6.2.2 Data Management core modules

A short description of the functionality of the Data Management core modules follows.

**Semantic Engine**. This module is used for extracting knowledge out of semantic information.

**Component Configurator / Manager**. This module is assigned with a variety of general tasks. It manages the specification of mapping information between the platform specific and platform independent models. It is also in charge of handling tools registration to the Capabilities Registry. Moreover, it provides management functionality of Data Connectors.

**Messaging System**. The Messaging System is the entity responsible for real-time delivery of data between BPR4GDPR components and tools as well as external entities (e.g. databases, legacy systems, etc.). A set of appropriate interfaces have been identified (see section 6.2.3). Moreover, the Messaging System constitutes an effective Policy Enforcement Point (PEP), by providing the mechanisms for enforcing the specified access and usage control policies, while it supports the transformation between Platform-Specific Models (PSM) and Platform Independent Models (PIM) by means of semantic ontologies. Based on the instructions received by the Policy Decision Point (PDP), this module is responsible for invoking needed tools (for example for data processing), by utilizing the appropriate information stored into the Capabilities Registry.

### 6.2.3 Data Management interfaces

The Data Management provides a set of interfaces in order to support integration with the rest of the BPR4GDPR components as well as data sources and external applications.

**Data connectors**. Data connectors support the Message Endpoint enterprise integration pattern[4] by providing a unified solution for accessing information stored in heterogeneous systems or collected on the fly, under a common interface.

**Tools interface**. This interface provides a common way to instruct the tools to perform a specific operation (e.g. to anonymize data).

**Data access interface**. This interface is used for accessing data provided by various data sources. On one hand, it provides a publish/subscribe interface for enabling tools to receive information based on semantic types and tool specified criteria. On the other hand, the Data access interface provides external querying functionality based on semantic search standards, like SPARQL[5].

**Policies interface**. This interface provides integration with the Policy Decision Points (PDP) in order to enforce the specified fine-grained access and usage decisions and instructions.

**Capabilities interface**. This interface is used for querying the Capabilities Registry for the set of available capabilities provided by the various tools integrated into the system. This interface also provides the tools registration management functionality.

**Administrator interface**. This interface is used for administrating the Data Management component, for example registering new tools and data connectors.

## 6.3 Generic tool description

To cope with all the different requirements presented so far, we resort to a standard stack rather than a common single system model and we propose the use of a container system for this reason.

The technologic solution chosen for this project is Docker[6], currently a market leader in providing containerization. This choice will prevent tools design and implementation from technology or infrastructure lock-in.

Before introducing the initial BPR4GDPR tool architecture we briefly introduce the Docker technology that we will use underneath.

### 6.3.1 Tool Deployment model

Docker is a technology for providing container software.

A Docker container is a standardized unit of software that includes software code, all its dependencies, settings and system tools to make portable software package. Differently from virtual machines, Docker containers share the same machine's operating system kernel, resulting in lightweight, standalone package solution that can be deployed on different platforms (Linux, Mac and Windows).

Docker is an industry standard, adopted in many cloud system and datacenter, as it separate application dependencies from infrastructure.

[4] https://www.enterpriseintegrationpatterns.com/patterns/messaging/MessageEndpoint.html
[5] https://www.w3.org/TR/sparql11-overview/
[6] https://www.docker.com/

Containers isolate software from the host operating system environment, making run application safe and fast and thus matching the project system requirements with respect to virtual machine. For this reason, it seems to be suitable for BPR4GDPR tools developing.

Docker also provides some architectural elements that we want to use for the definition of the BPR4GDPR tool. In particular these elements are four (presented in increasing hierarchical order):

**Docker containers**: are standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably, in an isolated and safe environment.

**Docker services**: are "containers in production". A service runs one image and codifies the way that image runs such as what ports it should use and how many replicas of the container should run.

**Docker swarm**: a group of machines that are running Docker and joined into a cluster.

**Docker stack**: A stack is a collection of services that make up an application in a specific environment.

Stack is described by a *docker compose file*, standard file expressed in YAML, whose an example is represented in Figure 14.

These files describe the services to be deployed, the port needed by services and the resource shared/accessed from the real machine (such as volumes, real/virtual network interfaces).

```yaml
lb:
  image: dockercloud/haproxy
  links:
    - web
  ports:
    - "80:80"
  roles:
    - global
web:
  image: dockercloud/quickstart-python
  links:
    - redis
  target_num_containers: 4
redis:
  image: redis
```

**Figure 14: Example of docker stack configuration file (docker-compose.yml)**

### 6.3.2 BPR4GDPR tool architecture

The architecture of a single tool is based on a Docker stack, represented in the red box in Figure 15.

The tool may present multiple containers, thus fulfilling the need of easily prototyping a distributed system. The tool can access external resource of the company through a bidirectional data channel toward the Data Management Middleware element, that enforces the policy and offer an abstraction layer of the actual company resources.

Each tool can run in standalone mode. Docker simplifies the deployment of each tool both in demonstration and in production environment.

Each tool presents an API whose role is twofold: on one hand it can be used from third parties of from an external orchestration, on the other hand it allows the construction of responsive web interface for administration.

These API can be defined in a cross-platform tool such Swagger[7] that help in the design of your API development.

Tools can be provided with a web interface, allowing user (admin) to access, configure or display the various tools functionality.

Being a Docker stack, each tool will be described by a Docker compose file. The image of each single container can be provided either from private repository or, from Docker Hub to simplify distribution and provisioning.

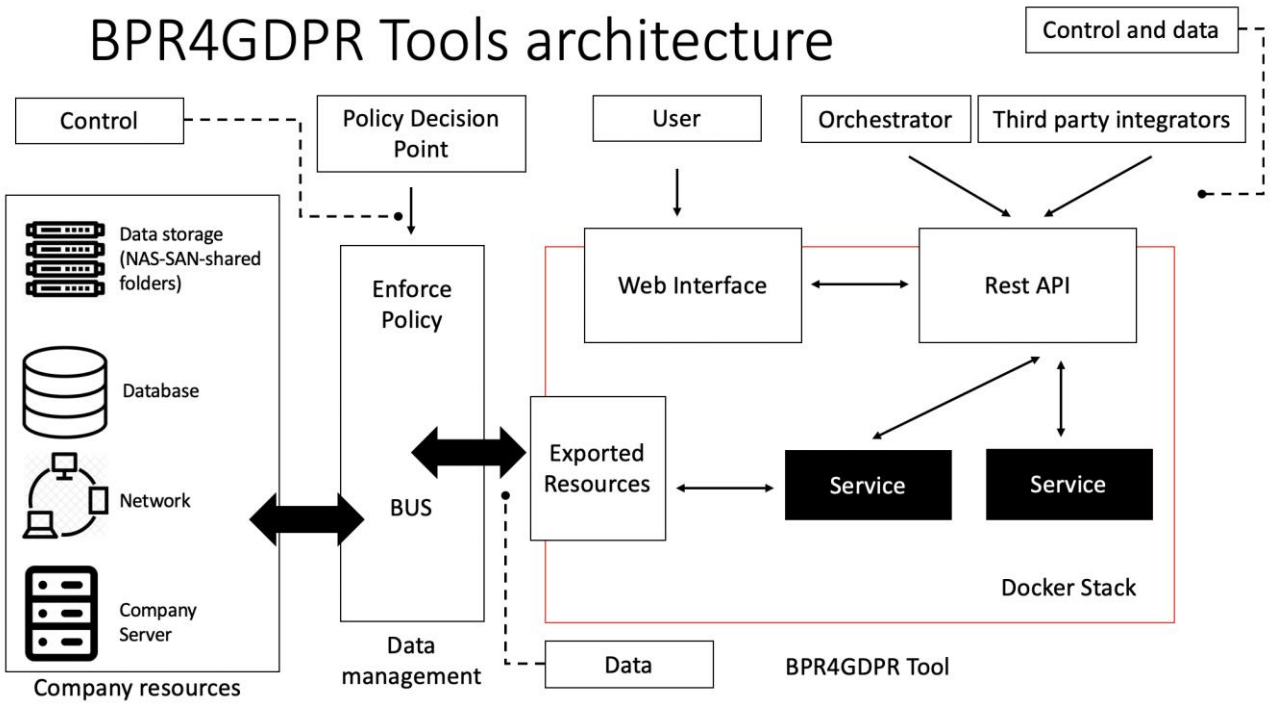


Figure 15: BPR4GDPR tool architecture

## 6.4 BPR4GDPR Tools

### 6.4.1 Risk assessment tool

Every company that needs to be complaint with GDPR (and in particular with recital 77) has to conduct regular risk assessments as it is an essential part of cyber security. However, there is an apparent dichotomy: if on one side a correct risk assessment will be possible only by the collaboration of several security experts, on the other hand company's risks should not be disclosed to many actors as represent a very sensitive information.

[7] https://swagger.io

BPR4GDPR, using Privacy Enhanced Technologies, can help in this sense, providing tools for collaborate in a privacy preserving manner, also to accomplish collaborative risk assessment.

This tool requires distributed deployment. Indeed, to effectively use techniques such Shamir Secret Share, we need to develop a distributed system running on multiple non-colluding peer nodes so that they do not normally combine their secrets share.

### 6.4.2 Cloud data management tool

An important aspect issue of GDPR is about the use of cryptography. This aspect is tightly and inextricably related to the problem of key management. Many cloud providers indeed offer solutions that make use of cryptography but they deal the key management in a centralized manner. Even though this policy helps the management of many case (e.g. the client loses its key), this centralized key management likewise severely limits the effects of cryptography on enhancing data. Techniques in the field of Secret Sharing e.g. (Linear Secret Sharing - ABE) could help providing solutions in case of key loss, and allow more sophisticated and partially decentralized cryptographically-enforced data access policy management.

Being a core part of IT infrastructure, a tool that deal with cloud data management pose challenging requirement in terms of expositing machine-to-machine interface to integrate. We think that a key point is also the easy ability to deploy the tool in a demonstration environment in order to show its capability, fostering companies to take only just part of the proposed solution for replication/integration in it in their network.

### 6.4.3 Anonymization/pseudonymization tool

The anonymization techniques cited in the GDPR for data masking need a special attention as it should deal with modern deanonymization attacks such [15]. Indeed, even if anonymized, data always release a fingerprint about user information and it is not always easy, also in case of case specific problems, understand if anonymized data is disclosing sufficient information to represent a problem for privacy. The goal then is to find a tradeoff among the usability of the anonymized data (efficiency), and the amount of privacy revealed (secrecy).

Notably, using PETs, the same problem can be addressed in some cases more efficiently by moving the query within the data storage, using the so called “secure queries” [16]. The result is the inversion of the operations: export the result of the data process (which often aggregates data consistently), instead of exporting anonymized data to be further processed.

### 6.4.4 Data subject support

The services group together as data subject support intend to provide adopting organisations with reusable services and software components to help data subjects exert their rights in accordance with the GDPR regulation with regards to data erasure, data rectification, right of access to their data, and data portability. That is, the data subject support implements the basic reusable mechanisms necessary to be combined to ensure compliance with multiple articles of the GDPR regulation. *Article 15*, *right of access by the data subject*, within *Section 2* on *information and access to personal data* and the *Articles 16, 17*, and *20* as gathered in *Section 3* on *rectification and erasure: right to rectification*, *right to erasure (‘right to be forgotten’),* and *right to data portability*.

The data subject support will deliver a service devised for data erasure in heterogeneous software landscapes with multiple persistence software components and distributed business logic. This specific scenario of data erasure stems from real-life systems such as automotive CRM systems[8] whereby the data about data subjects are stored and processed across a complex cross-organisational ecosystem with many collaboration software entities processing their data—at the left-hand side of Figure 16. The first steps towards the implementation of cross-organisational right to erasure involve finding the actual data of the data subject exerting the right to erasure of her data. The data subject support asset analyses the multiple data sources, at the left-hand side of **Error! Reference source not found.**, to identify and extract the data and data location of the data of the data subject requesting to erase her data, in this case Müller. This step includes either continuous updating the Data subject support asset on data changes when appropriate interfaces are implemented to the external data sources or data exports (e.g. CSV, XSL) upon requests. Further, the Data subject support asset carries out data integration mechanisms to discover data discrepancies and uses dictionaries to match the data in different databases to the same data subject regardless of the differences between those data—for example, the fact that Müller, Mueller, and Muller might be the same person. This information is stored in the database on the top right-hand side of **Error! Reference source not found.** to notify the data processors of which data they should erase. The notification could be stepwise turned into automated erasure upon databases of which the pilot demonstrator disposes of total control.

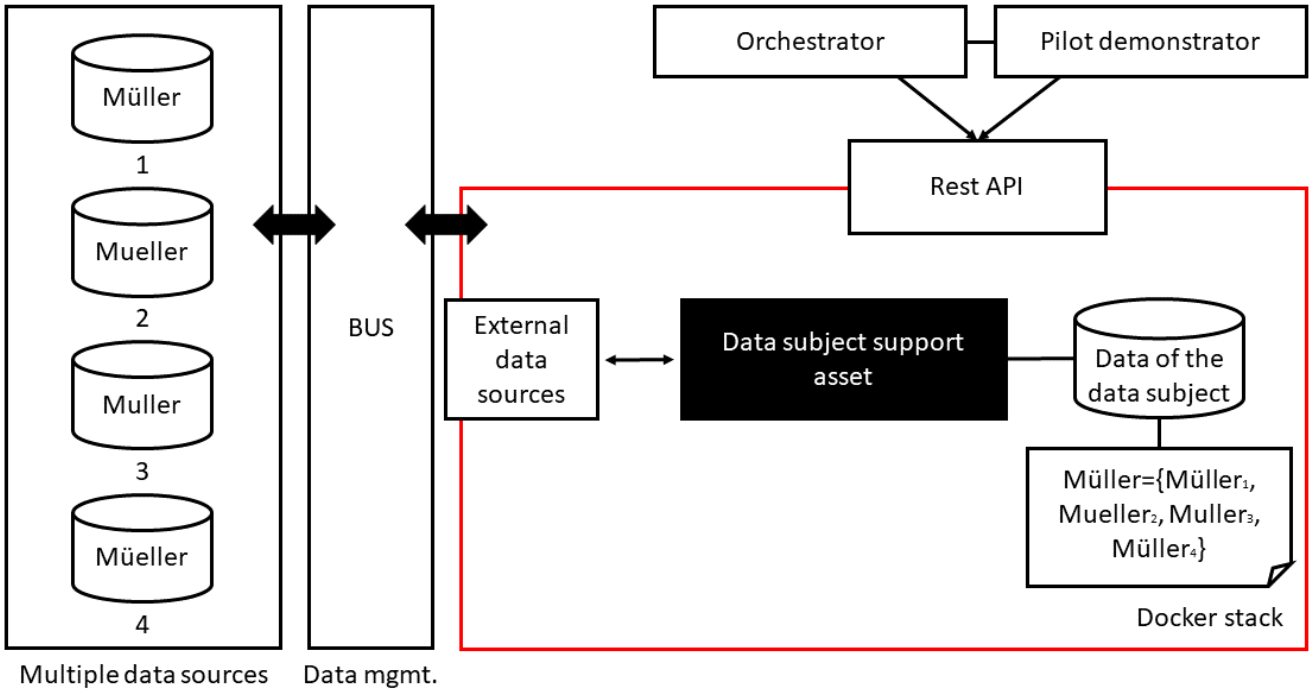


**Figure 16: Conceptual depiction of the data subject support asset**

[8] This case stems from Section 5.5—Use case 4: cross-organisational right to erasure—within the pilot demonstration for Compliance-as-a-Service in cross-organisational automotive CRM in the BPR4GDPR Deliverable 2.1, use cases and requirements.

# 7 Conclusions

This deliverable provides the initial specification of the BPR4GDPR architecture. The proposed architecture is divided in four “quadrants”/blocks, reflecting different groups of functionalities: governance, planning, monitoring and runtime. For each block the functionality of the components has been specified, along with their internal functional architecture and provided interfaces.

Whereas the BPR4GDPR system architecture in terms of components and interfaces has been clarified and is not expected to change in the remainder of the architecture specification work, the fine-grained specification is current work in progress. More specifically, the detailed specification of each BPR4GDPR component is the subject of the dedicated technical work-packages WP3, WP4, and WP5. Moreover, the detailed specification of the framework, components and related interfaces will be included in the final architecture specification deliverable (D2.4), due at project month 25.

## References


[1] Cuppens, F., Cuppens-Boulahia, N.: Modeling Contextual Security Policies. International Journal of Information Security 7(4), 285{305 (2008)

[2] Botha, R.A., Elo, J.H.P.: Separation of duties for access control enforcement in workflow environments. IBM Systems Journal 40(3), 666{682 (March 2001)

[3] Koukovini, M.N.: Inherent privacy awareness in service-oriented architectures. Ph.D. thesis, National Technical University of Athens (2014)

[4] Koukovini, M.N., et al.: Towards inherent privacy awareness in workflows. In: 9th International Workshop on Data Privacy Management (DPM 2014) (September 2014)

[5] Koukovini, M.N., et al.: An Ontology-Based Approach towards Comprehensive Workflow Modelling. IET Software 8(2), 73{85 (2014)

[6] Jablonski, S., Bussler, C.: Workflow management: modeling, concepts, architecture and implementation. International Thomson Computer Press (1996)

[7] Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy (SP '07) (May 2007)

[8] Van Der Aalst, W. (2011). Process mining: discovery, conformance and enhancement of business processes (Vol. 2). Heidelberg: Springer.

[9] Carmona, J., van Dongen, B., Solti, A., & Weidlich, M. (2018). Conformance Checking: Relating Processes and Models. Springer

[10] Van der Aalst, W. M., van Dongen, B. F., Günther, C. W., Rozinat, A., Verbeek, E., & Weijters, T. (2009). ProM: The process mining toolkit. BPM (Demos), 489(31), 2.

[11] Kalenkova, A. A., De Leoni, M., & van der Aalst, W. M. (2014, September). Discovering, Analyzing and Enhancing BPMN Models Using ProM. In BPM (Demos) (p. 36).

[12] Hassani, M., et al.: Efficient process discovery from event streams using sequential pattern mining. In: IEEE Symposium Series on Computational Intelligence, SSCI 2015, Cape Town, South Africa, December 7-10, 2015. pp. 1366–1373 (2015), https://doi.org/10.1109/SSCI.2015.195.

[13] Drik Fahland, Will M.P.van der Aalst: Model repair — aligning process models to reality. Information Systems Vol(47): 220-243- https://www.sciencedirect.com/science/article/pii/S0306437913001725

[14] van Zelst, S.J., et al.: Online conformance checking: relating event streams to process models using prefix-alignments. International Journal of Data Science and Analytics (Oct 2017), https://doi.org/10.1007/s41060-017-0078-6

[15] A. Narayanan and V. Shmatikov, 2008. Robust De-anonymization of Large Sparse Datasets. In 2008 IEEE Symposium on Security and Privacy, doi: 10.1109/SP.2008.33, ISSN 1081-6011

[16] Jelena Mirkovic. 2008. Privacy-safe network trace sharing via secure queries. In Proceedings of the 1st ACM workshop on Network data anonymization (NDA '08). ACM, New York, NY, USA, 3-10