SPIRIT: A Semantic Transparency and Compliance Stack

Patrick Westphal[1], Javier D. Fernández[2,3], Sabrina Kirrane[2], and Jens Lehmann[4]

[1] Institute for Applied Informatics (InfAI), University of Leipzig, DE
patrick.westphal@informatik.uni-leipzig.de
[2] Vienna University of Economics and Business, AT
{javier.fernandez, sabrina.kirrane}@wu.ac.at
[3] Complexity Science Hub Vienna, AT
[4] Enterprise Information Systems, Fraunhofer IAIS, DE
jens.lehmann@iais.fraunhofer.de

Abstract. The European General Data Protection Regulation (GDPR) sets new precedents for the processing of personal data. In this paper, we propose an architecture that provides an automated means to enable transparency with respect to personal data processing and sharing transactions and compliance checking with respect to data subject usage policies and GDPR legislative obligations.

1 Introduction

The European General Data Protection Regulation (GDPR) came into effect in May 2018. Although Recital 26 of the GDPR clearly states that the obligations set forth in the GDPR do not apply to anonymous data, it is often impossible to guarantee that data is truly anonymous. Especially in a Big Data context many companies find themselves in a state of uncertainty with respect to current business intelligence operations. Another issue faced by many companies is the fact that the desired outcome can not always be achieved over anonymised data (e.g. personalised recommendations). In such cases, there is a need not only to obtain consent for personal data processing from the data subject, but also to provide transparency with respect to the processing and demonstrate compliance with respect to the data subjects consent. Towards this goal, in this paper, we propose a personal data processing transparency and compliance architecture that employs existing Big Data processing techniques.

2 The SANSA Stack

The current "Big Data landscape" provides a plethora of tools and frameworks covering a variety of methods and techniques for processing huge amounts of data via a distributed cluster of machines. However, none of the general purpose Big Data processing frameworks provide built-in support for processing *big semantic*

data e.g. to load and store RDF data, which, as a uniform data format supports dealing with heterogeneity of Big Data. This gap is tackled by the *Semantic Analytics Stack (SANSA)*[5] [1] which is an open source semantic data processing framework built on top of Apache Spark[6] and Apache Flink[7] SANSA provides a stack of functional layers ranging from RDF/OWL data representation to machine learning algorithms working on semantic data.

The *Knowledge Distribution and Representation* layer provides a means to read and write RDF and OWL files. In terms of data structures and programming interfaces SANSA follows the common and accepted representations of Apache Jena[8] and the OWL API[9]. Hence, the RDF and OWL data is provided as distributed collections of Apache Jena triples and OWL API axioms, respectively. On top of this, the *Query* layer comprises functionality for searching, exploring and extracting information from big semantic data through the SPARQL query language. SANSA supports executing SPARQL queries within an Apache Spark/Flink program, or via an HTTP SPARQL endpoint. In both cases the actual queries are translated into lower level Apache Spark/Flink data processing instructions and executed on the Knowledge Distribution and Representation layer. The next layer in the SANSA Stack is the *Inference* layer which builds on the layers mentioned so far. Besides actual data-level assertions, the Semantic Web technology stack also provides a means to express schema or ontological knowledge. Parts of the inherent semantics of the respective W3C standards, RDFS and OWL, may be encoded as rules which can be applied to infer new knowledge. With this *forward chaining* process all rule-based inferences may be materialized. In contrast *backward chaining* techniques infer new knowledge starting at a given 'goal', which can be a (set of) RDF triple(s). SANSA supports different existing reasoning profiles for rule-based forward/backward chaining. Apart from these profiles, SANSA is able to compute an efficient execution plan for arbitrary sets of rules. Hence, users can adjust the trade-of between expressivity and performance, and furthermore introduce custom rules, e.g. to represent business policies. On top of the SANSA Stack the *Machine Learning* layer provides a collection of machine learning algorithms that can directly work on RDF triples or OWL axioms. The algorithms implemented thus far cover knowledge graph embeddings [2] (e.g. for link prediction), graph clustering and association rule mining techniques.

3 SPIRIT: Leveraging the SANSA Stack for Transparency and Compliance

In this paper, we introduce our transparency and compliance checking implementation of the SANSA stack, which is depicted in Figure 1. The SANSA-based

[5] SANSA Stack home page, `http://sansa-stack.net`
[6] Apache Spark, `https://spark.apache.org`
[7] Apache Flink, `https://flink.apache.org`
[8] Apache Jena, `http://jena.apache.org/`
[9] OWL API, `https://owlcs.github.io/owlapi/`

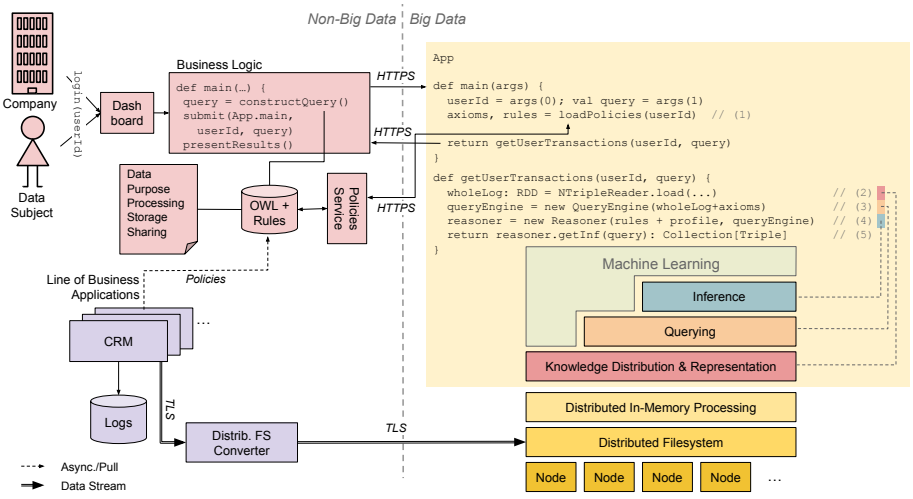

Fig. 1. SPIRIT architecture exemplifying the transparency use case

transparency and compliance checking application (right) is used to analyse log information concerning personal data processing and sharing that is output from line of business applications on a continuous basis (bottom left), and to present the information to the user via the SPIRIT dashboard (top left).

Ingesting Transaction Logs into SPIRIT: When it comes to personal data processing there is a need for a general mechanism to verify compliance with existing usage policies and legal obligations. One such mechanisms is the recommissioning of existing application and system logs such that they can be used to verify that data processing and sharing complies with usage policies specified by the data subject. Considering the sheer volume of data generated when application logs are used for personal data processing and sharing auditing, there is a need for a file system that is able to handle Big Data, is fault tolerant, and is capable of supporting parallel processing. The *Hadoop Distributed File System (HDFS)*[10] fulfills all those criteria and is the default choice for Apache Spark and Apache Flink. Moreover, there is a stable and mature solution to transfer log data to HDFS, called Apache Flume[11], which provides a means to transform log content, e.g. obtained from an application log, before it is passed along to the HDFS. This allows heterogeneous transaction logs to be translated to RDF on the fly.

SPIRIT Transaction Log Processing with SANSA: Our SANSA-based architecture allows storage and access to all log data in a Big Data processing environment. Semantic web technologies ease *data integration* across several heterogeneous line of business applications, enabling *interoperability* across platforms and providing a simple way to link user data and policies. As sketched

[10] HDFS, `http://hadoop.apache.org/`
[11] Apache Flume, `http://flume.apache.org/`

in Figure 1 the main steps that need to be performed include: (1) loading the policies from the policy store and dividing them into rules that are used in the reasoning step, and schema/ontology axioms added to the log data later; (2) loading the RDF log data stored in the distributed file system; (3) initialising a query engine with the log and schema/ontology data; (4) creating a reasoner which works on the query engine, considers the rules from the policy store and a set reasoning profile; and eventually (5) invoking the backward chaining on the given query goal. Our SPIRIT architecture offers transparency for data subjects, and means to verify that all business processes comply both with the consent provided by the data subject and relevant obligations from the GDPR by: (i) encoding user data policies in (subsets of) OWL 2 DL; and (ii) providing a compliance checking mechanisms on the basis of the the SANSA inference rule-engine. As for the former, we allow policies to define restrictions in terms of five data categories related to the GDPR regulation (as depicted in Figure 1): *Data* reflects which personal data is governed by the policy. *Processing* lists the operations (e.g. anonymisation, aggregation, etc.) performed on the personal data. *Purpose* describes why data are collected/processed. *Storage* concerns where data are stored and for how long. *Sharing* specifies the potential use of the personal data by third parties. In addition to the personal data policies, the SPIRIT architecture holds rules that provide means to check compliance of data processing and sharing transactions according to the data policies and GDPR regulations. Acknowledging that GDPR compliance checking cannot fully automated (given the generality, vagueness and subjectivity inherent in the regulation), we focus on verifying minimal sets of conditions (*"if condition X holds then the data policy Y is violated"*) to assist the stakeholders in charge of providing evidence of GDPR compliance.

The SPIRIT Dashboard: The SPIRIT dashboard provides a means for data subjects, companies and supervisory authorities to obtain transparency with respect to the processing of personal data and compliance with respect to the data subjects usage policies. A user request is converted into a query which is passed to the SANSA application, together with a user identifier. The results of the respective processing task is then passed back to the dashboard to be presented to the user.

References

1. J. Lehmann, G. Sejdiu, L. Bühmann, P. Westphal, C. Stadler, I. Ermilov, S. Bin, N. Chakraborty, M. Saleem, and A.-C. N. Ngomo. Distributed semantic analytics using the sansa stack. In *Proceedings of the 16th International Semantic Web Conference (ISWC)*. Springer, 2017.
2. M. Nickel, K. Murphy, V. Tresp, and E. Gabrilovich. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33, 2016.