Privacy Enforcement at a Large Scale for GDPR Compliance

Ety Khaitzin, Roee Shlomo, Maya Anderson
IBM Research
Haifa, Israel
etyk@il.ibm.com,roeesh@ibm.com,mayaa@il.ibm.com

CCS CONCEPTS

• **Information systems → Cloud based storage**; • **Security and privacy → Access control**; • **Applied computing → IT governance**;

New regulations, such as the upcoming European General Data Protection Regulation (GDPR), specify new and challenging data governance requirements. Specifically, when providing access to the data, the regulations require to take into account new concepts such as the consent given by the individual who provided the data, known as the data subject, and the usage of the data, known as data usage purpose.

Existing access control tools either use compliance checks that do not completely match the new and complex requirements that GDPR introduces, or are limited in their scalability. Most of the existing solutions apply a coarse-grained protection, protecting access to a data object. Tools that provide fine-grained compliance at the granularity of specific cells, do so by either making decision for each row separately, and thus are limited in their scalability in the data lake, or by creating static views for each possible scenario, a solution that will not work for a wide set of request attributes with multiple possible values.

So, how do we enforce GDPR policies, which require taking into account the consent of the data subjects at a very fine granularity, on a large-scale data store? Our goal is to minimize the latency added by policy enforcement. Hence, we start by overcoming the need to communicate with the PAP (Policies Administration Point) and PIP (Policies Information Point) [1] during the in-line query execution. We add a pre-computation phase, in which we compile the policies and parts of the supplementary data (e.g. consents, profiles), keeping only the parts relevant to the policy decisions. Thus, we obtain a compiled representation that can be efficiently used during query-time. The result is stored as close to the data as possible in an accessible form.

This integration between the policies, the data content and profiles requires several steps: Combination of all the policies based on their priorities and target data fields; Evaluation of request-independent data, including data from external data sources; Storage of this data in a compact way, for fast retrieval in run-time phase.

The result of this process is an *intermediate representation* that should be stored, at least partially, near the data. This enables to avoid both access to remote data, and calculation of the logic of the policies during the enforcement time.

The in-line enforcement module handles each user query to the governed data. It is a transparent layer between the query and the data, e.g. acting as a proxy before executing the query on the data. It rewrites the query using *the intermediate representation* computed in pre-computation phase. Now the rewritten query is executed on the combination of the original data and *the intermediate representation*. The solution returns transformed data to the user, e.g. with cell-level filtering, using query rewriting with filters, conditions and functions that are available in the query language.

This approach has several advantages. It enables leveraging the data store query engine and the optimizations available in the data stores, providing us with performance gains. In addition, it improves the security of the solution, due to the fact that the governance action is taken as part of the query engine, so only compliant data leaves the data store, decreasing data leakage risks.

SYSTOR, 2018, Haifa, Israel

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

ACKNOWLEDGMENTS

This work has been supported by the European Commission through the Horizon 2020 Research and Innovation program under contract 731945 (DITAS project).

REFERENCES

[1] eXtensible Access Control Markup Language (XACML) Version 3.0. 22 January 2013. OASIS Standard.