How can DB Systems be ready for privacy regulations

Javam Machado[1], **Paulo Amora**[1]

[1]Laboratório de Sistemas e Bancos de Dados
Departamento de Computação – Universidade Federal do Ceará (UFC)
60.455-760 – Fortaleza – CE – Brazil

{Javam.Machado,Paulo.Amora}@lsbd.ufc.br

Abstract. *Personal data usage and collection are activities that used to grow unrestricted. However, several laws in the physical world ensure rights to people regarding their privacy and information usage. In the last years, legislators passed many laws, regulations, and acts to replicate these rights to the digital world. By doing so, new constraints, rights, and duties appear on every component of the data usage and collection workflow. In this paper, we introduce some of these laws, describe some of the rights that highly impact the current design of DBMSs, discuss the challenges raised by these regulations, as well as related works and research opportunities.*

1. Introduction

Lately, data privacy regulations rule over sensitive personal information in many countries. European Union has the General Data Protection Regulation (GDPR). Canada approved in the late 1990s the Personal Information Protection and Electronic Documents Act (PIPEDA). Several USA states have similar regulations, like the California Consumer Privacy Act (CCPA). In Brazil, the *Lei Geral de Proteção de Dados (LGPD)* comes into effect year 2020. In general, these regulations protect individuals' data stored in organizations, giving the individual control on how their data is shared and processed. Many applications collect personal data, such as mobile applications, e-commerce, social networks, and any transactional system where users are involved. For instance, the coronavirus pandemic led several countries, cities, and health organizations to develop mobile applications that collect personal contacts based on geolocation data. Although this initiative is of great importance for controlling the spread of the virus in a community, it is clear that outside of this purpose, this type of data might be very sensitive for an individual.

Database systems (DBMS) are the primary tools organizations use to store and manage their data, including sensitive personal information. Mining data and sharing information between partners are both heavily based on data directly provided by DBMS. There is no doubt that personal data is stored, processed, and shared within organizations, among other transactions' data. Therefore, DBMS have to provide capabilities that allow organizations to comply with the regulations. That involves at least five concepts: Identifying personal data; Managing metadata about processing and sharing personal data; Giving the user the correct tools for declaring personal data visibility and usage; Providing efficient auditing interfaces; Sanitizing personal data before publishing or sharing among partners.

In this paper, we investigate the impact of managing sensitive personal data on DBMS. We aim to address the main features that have to be reviewed at the core level of

these systems to allow data controllers and processors to be compliant with privacy regulations. We enumerate law requirements that have to be met by DBMS when they store personal information. We also identify several open issues for research opportunities.

1.1. Policy Requirements

From the concepts stated above, the regulations, acts, and laws have many points of intersection with regards to users' rights and data controllers' and processors' duties. To discuss these rights and obligations, we will take GDPR [General Data Protection Regulation 2016] as an example. GDPR is composed of 99 articles, ranging from data privacy, protection, cryptography as well as users' own right to access collected personal data, know what it will be used for as well as request removal of said data. As GDPR is an extensive regulation, we highlight five rights to focus our discussion and vision:

Right of access - Article 15 states that the data subject can obtain confirmation from the controller as to whether their data is being processed and access to: data itself, purposes, categories of personal data, recipients of this data, the period of storage, usage in automated decision-making.

Right to be informed - Article 12 states that the data subject must be informed in a concise, transparent and easily accessible form if their data is obtained or not, if data will be used in automated individual decision-making, if data about this subject is breached, as well as their capabilities of data portability and objection.

Right to be forgotten - Article 17 states that the data subject has the right to obtain the erasure of personal data without undue delay. This covers not only active requests but also covers collected data that is no longer necessary, unlawfully processed. The controller must also inform other controllers of the erasure request, so that appropriate measures are taken.

Consent - Article 6 states that one of the conditions to make data processing lawful is that the data subject consent to the processing directed to one or more specific purposes. Consent is defined by article 4 as a freely given, informed, and unambiguous agreement, therefore, consented data collection may not be misused under a different purpose.

Singling out - Article 4 defines personal data as "any information relating to an identified or identifiable natural person; an identifiable natural person is one who can be identified, directly or indirectly". Following Recital 26, singling out is a way of identifying a natural person in a database. Therefore it is expected that the database does not store any data that would allow singling out a person. That is particularly true for any user who declares himself as "opted out" whenever he declines to allow the data holder to use his data differently from the original purpose.

2. Impacts on DB Systems

In this section we discuss six components necessary to achieve compliance, what are the responsibilities of the DBMS and storage engine, present research opportunities in each component, highlighting existing approaches.

Metadata explosion - This requirement relates to all rights described before given each right must have its own data to be guaranteed and refers to the amount of metadata needed

to comply with the regulation, that it will grow considerably, even exponentially, as new data is stored. The storage engine must be aware of these metadata, if it will be stored together with data or accessible through a different way, and the DBMS must guarantee that this overhead does not severely impact performance. GDPRBench [Shastri et al. 2020] evaluates the impact of adding metadata as attributes of the tuple, and finds that even minor modifications to comply with GDPR cause an increase of 3.5 times more data stored and this number rises to 5.95 times when secondary indexes are added to this metadata while decreasing performance considerably. SchengenDB [Kraska et al. 2019] discusses the addition of surrogate keys to link all data related to a user, and instead of using foreign keys, the relationship is also treated as an intermediary table, composed of these surrogate keys. Moreover, purpose filters are added through bit vectors, which validate for which purposes the data is allowed to be retrieved, although these can be compressed if there is a sense of hierarchy between purposes. The opportunities are in compact representation of this metadata, allowing for fast retrieval and modification.

Delete guarantees - This requirement relates to the right to be forgotten and brings the problem of ensuring that the requested deletes happen without undue delay. This means that deletes must be followed through, instead of them being a promise, as well as if such promise is made, that it is tracked, to be fulfilled in a given time. The storage engine must provide either a confirmation mechanism or have in its design the deletion guarantees, and the DBMS must guarantee that the deletes happen in able time, through management mechanisms handling the delete confirmation and/or enforcement. One approach that deals with this issue is Lethe [Sarkar et al. 2020], which works by triggering compactions in LSM-tree based storage engines in a periodic manner, ensuring that data does not remain indefinitely. Therefore, opportunities remain in approaches that also treat deletes as a first-class citizen, as well as approaches similar to garbage collection, in an active/passive hybrid approach, to guarantee deletes on time as well as be opportunistic with relation to the resources.

Efficient auditing - This requirement relates to the ability of auditing and ensuring the rights are being respected. It brings the problem of making sure that all data accesses are properly recorded and it is possible to retrieve all accesses to a given user/register. The storage engine must be able to record each access in able time, storing not only the access, but what was accessed, who accessed it and how it was accessed, and the DBMS must guarantee that this log is accessible in an efficient manner, through specialized and indexed logs. This subject has been widely explored in academia, and one example is Instant Recovery [Graefe et al. 2016], which proposes a structure to facilitate random access to the log, without hindering log write performance. A research opportunity presents in using such ideas in a more modern logging scheme, such as [Haubenschild et al. 2020].

Purpose-based access - This requirement relates to right of access, right to be informed, and consent, bringing the problem of only allowing data associated with a given purpose to be queried within this purpose. The storage engine must associate each data item with its purpose metadata, as well as return the data accordingly. The DBMS must guarantee that queries have well-defined purposes and that queries do not access unwanted data through access filters. Existing works deal with that by associating purposes to tuples as bit-vectors [Kraska et al. 2019] and creating purpose filters to avoid unwanted data leaking. In an application sense, it also describes a "sandbox" mechanism in which the

DBMS would only be accessible by applications through specific VMs, each with their own purpose. Meanwhile, there are several advances in filters. HOPE [Zhang et al. 2020] proposes a key compression mechanism for in-memory search trees, then, it can be used to query filters and indexes in an efficient and space-saving manner. Therefore, opportunities are many in how to store efficiently the purpose data, as well as how to query this data in a fast way, without violating purpose definitions.

Opted data - This requirement relates to right of access, singling out, and consent, bringing the problem of storing data, but filtering its access to only some types of queries. The storage engine must filter out all results that are opted out, however, in aggregation queries, these results must be computed, and the DBMS must guarantee that query results do not allow users to be singled out, as well as disallow single target purposed queries in opted-out data. The privacy field of study deals with this type of problem constantly, two recent examples of work are PrivSQL [Kotsogiannis et al. 2019], a differentially private SQL query engine and SAP HANA [Kessler et al. 2019], a widely known commercial DBMS. These works offer the power of differential privacy [Dwork 2006] to answer numerical queries closer enough to the real results to be useful but slightly different, so that person re-identification is unlikely. Although several approaches exist for publishing and sharing information in a differentially private way, they cover only aggregate queries, such as COUNT(*), and noise that has to be added in publishing settings is usually too high. Moreover, mechanism integration to existing DBMS is still a vast subject of research.

User related data retrieval - This requirement relates to right of access, consent and brings the problem of timely retrieving all the data related to a given user, even if distributed across relations and derived information. The storage engine must account for this data, and facilitate retrieval of all user-related data. Moreover, the DBMS must guarantee that every user has their own data tagged and available through metadata tags and index structures. This is also a vastly explored theme in academia, however, points to be taken into consideration are space allocation, as well as timely retrieval and deletion. Compliance by Construction [Schwarzkopf et al. 2019] suggests that all user data is stored in user shards. The shards are inaccessible for query, instead, materialized views based on the query and the purposes associated would be produced to provide the data while hiding the true data and allowing the user to request, remove or revoke access. Therefore, opportunities are in succinct data structures, that allow fast access to several instances of user data across tables and files.

As Shah et al. [Shah et al. 2019] mentions, GDPR compliance can be seen as a 2-dimensional spectrum, ranging from real-time to eventual compliance. This means that either the system may be GDPR-compliant at all times, or guarantee said compliance given some time. The system can also be evaluated from a full vs partial compliance stance, in terms that either it complies with all GDPR policies or some of them. We agree with this spectrum, arguing that this flexible compliance should be dynamic and configurable. DBMSs must have a well-defined set of rules regarding compliance, which component is accountable for each rule, and how it is configured. They must also make it clear to applications what should be considered to achieve full compliance, as well as to avoid interference from applications into the DBMS guarantees.

Data replication is an issue since when data is copied, it produces more metadata as well as needs more resources to track the copies. There are position papers on both

sides, that data should never be copied and that data must always be provided as a copy. We believe that copies should be discouraged, as tracking data may slow down the system as well as leak information, however, derived results that do not break single individual privacy can be copied because, even if a user requests removal of the data, their data is not possible to single out.

Another issue that appears is that many of the modifications may conflict so much with fundamental DB design that they can be either impractical or place trust in the DB administrator (DBA), a human component subject to malice and failure. SchengenDB provides examples of log maintenance since erased data will still be present in log records, as well as in the suggested "sandbox" approach, instead of stopping intercommunication altogether, the system can be permissive and warn that access across different purposes happens, then, it would notify the DBA, placing the trust that appropriate action is taken.

2.1. Takeaways

From the discussion on the regulations, the challenges raised, and research opportunities, we present some takeaways that may direct these future research opportunities, as well as provide inspiration on how to tackle the challenges.

Takeaway 1: Instead of one big solution, many small solutions. Many challenges raised are orthogonal to each other. While data replication is a risk in a data protection driven system, it can bring the desired benefit of allowing different processes or components to work together on the same data, speeding up performance. Being mindful of what to replicate and how, and keeping track of each by an efficient index structure can go a long way in achieving performance and data protection.

Takeaway 2: To develop prototypes and be formal. We have pointed out several issues one can dump in to investigate and carry out experiments in an existing DBMS or new ones. However, the related literature often presents formal specifications and evidence of the correctness of proposed algorithms and mechanisms. Usually, they have to guarantee compliance with regulations. Therefore it is expected that any new approach proves itself to meet the requirements.

Takeaway 3: Shared responsibility. The DBMS must guarantee data protection, however, that responsibility can be shared with applications. SchengenDB and Compliance by Construction go beyond the realm of the DBMS, and, while sharing the responsibility may increase the difficulty of giving the needed guarantees, it can be achieved by interfaces, to ensure that applications using the DBMS do so by complying with the rules and conventions associated with the data protection regulations.

3. Conclusion

Considering all the arguments and opportunities presented, we can build the following propositions: The DBMS workload profile will change drastically, once read-only transactions become write-intensive. The DBMS, as well as the storage manager, must become aware of the queries and the data, to provide required data as well as to make sure that no unauthorized data leaks through a query. These new challenges oppose the usual beliefs of DBMS design and, to achieve performance and compliance, it is necessary to rethink data structures as well as database architectures. A redesign of logging and index structures is mandatory to achieve performance in this compliance scenario. Alternatively, the DBMS may implement a flexible approach for compliance, as discussed, by aggregating log records or have some sort of tolerance interval on the compliance.

References

Dwork, C. (2006). Differential privacy. In *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer.

General Data Protection Regulation (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. *Official Journal of the European Union*, 59:1–88.

Graefe, G., Guy, W., and Sauer, C. (2016). *Instant Recovery with Write-Ahead Logging: Page Repair, System Restart, Media Restore, and System Failover, Second Edition*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Haubenschild, M., Sauer, C., Neumann, T., and Leis, V. (2020). Rethinking logging, checkpoints, and recovery for high-performance storage engines. In *SIGMOD Conference*, pages 877–892. ACM.

Kessler, S., Hoff, J., and Freytag, J. (2019). SAP HANA goes private - from privacy research to privacy aware enterprise analytics. *Proc. VLDB Endow.*, 12(12):1998–2009.

Kotsogiannis, I., Tao, Y., Machanavajjhala, A., Miklau, G., and Hay, M. (2019). Architecting a differentially private SQL engine. In *CIDR*. www.cidrdb.org.

Kraska, T., Stonebraker, M., Brodie, M. L., Servan-Schreiber, S., and Weitzner, D. J. (2019). SchengenDB: A data protection database proposal. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare - VLDB 2019 Workshops, Poly and DMAH, Los Angeles, CA, USA, August 30, 2019*, volume 11721 of *Lecture Notes in Computer Science*, pages 24–38. Springer.

Sarkar, S., Papon, T. I., Staratzis, D., and Athanassoulis, M. (2020). Lethe: A tunable delete-aware LSM engine. In *SIGMOD Conference*, pages 893–908. ACM.

Schwarzkopf, M., Kohler, E., Kaashoek, M. F., and Morris, R. T. (2019). Position: GDPR compliance by construction. In *Poly/DMAH@VLDB*, volume 11721 of *Lecture Notes in Computer Science*, pages 39–53. Springer.

Shah, A., Banakar, V., Shastri, S., Wasserman, M., and Chidambaram, V. (2019). Analyzing the impact of GDPR on storage systems. In *HotStorage*. USENIX Association.

Shastri, S., Banakar, V., Wasserman, M., Kumar, A., and Chidambaram, V. (2020). Understanding and benchmarking the impact of GDPR on database systems. *Proc. VLDB Endow.*, 13(7):1064–1077.

Zhang, H., Liu, X., Andersen, D. G., Kaminsky, M., Keeton, K., and Pavlo, A. (2020). Order-preserving key compression for in-memory search trees. In *SIGMOD Conference*, pages 1601–1615. ACM.