Data-CASE: Grounding Data Regulations for Compliant Data Processing Systems

Vishal Chakraborty
vi.c@uci.edu
University of California (UC) Irvine

Stacy Ann-Elvy
selvy@ucd.edu
UC Davis School of Law

Sharad Mehrotra
sharad@ics.uci.edu
UC Irvine

Faisal Nawab
nawabf@uci.edu
UC Irvine

Mohammad Sadoghi
msadoghi@ucdavis.edu
UC Davis

Shantanu Sharma
shantanu.sharma@njit.edu
New Jersey Institute of Technology

Nalini Venkatasubramanian
nalini@uci.edu
UC Irvine

Farhan Saeed
fsaeed1@uci.edu
UC Irvine

ABSTRACT

Data regulations, such as GDPR, are increasingly being adopted globally to protect against unsafe data management practices. Such regulations are, often ambiguous (with multiple valid interpretations) when it comes to defining the expected dynamic behavior of data processing systems. This paper argues that it is possible to represent regulations such as GDPR formally as invariants using a (small set of) data processing concepts that capture system behavior. When such concepts are *grounded*, i.e., they are provided with a single unambiguous interpretation, systems can achieve compliance by demonstrating that the system-actions they implement maintain the invariants (representing the regulations). To illustrate our vision, we propose Data-CASE, a simple yet powerful model that (a) captures key data processing concepts (b) a set of invariants that describe regulations in terms of these concepts. We further illustrate the concept of grounding using "deletion" as an example and highlight several ways in which end-users, companies, and software designers/engineers can use Data-CASE.

1 INTRODUCTION

The rise in organizations collecting and mishandling personal data has led to the emergence of data regulations across the world. Examples include the California Consumer Protection Act (CCPA) [15], the Virginia Data Protection Act (VDPA) [78], and Canada's Personal Information Protection and Electronic Documents Act (PIPEDA) [54]. Many countries are in the process of enacting their own laws. Of these, the most developed, scrutinized, and used is "The Regulation (EU) 2016/679" also known as the *General Data Protection Regulation (GDPR)* [50]. The Data Governance Act adopted by the EU in 2022 complements the GDPR [49].

Building systems that allow compliant data governance has been identified as a key challenge in the recent Seattle Report on Database Research[1]. While GDPR has resulted in tangible improvements in how organizations handle data [10, 28], it has nonetheless led to negative economic impacts, one of the causes for which is the uncertainty companies face in ensuring compliance [14, 16] and the risk of penalty if found non-compliant [24, 25, 59, 70]. A key reason for such uncertainty is the ambiguity in the legal language used in data regulations when it comes to how systems should process data, i.e., which actions should the system take and when to comply with data regulations. Many of the concepts listed are open to several valid interpretations.

Consider a company MetaSpace that stores personal data of individuals, including their location for smart space applications, using PostgreSQL (PSQL). A user wishes to exercise their right (GDPR Article 17) to have their data deleted in a reasonable time. Persistently deleting data, e.g., using VACCUM-DELETE in PSQL can be extremely expensive. On the other hand, adopting logical deletes as in Cassandra, a NoSQL distributed database [42], — inserts a *tombstone* when data is deleted— can be efficient. Prior work [62] has shown that using delete markers like tombstones in LSM trees may lead to data being, illegally, physically retained for a long duration. The impact of the ambiguity is further highlighted when we consider distributed systems that may replicate /cache data across different nodes using various complex protocols [48]. If erasure means removing the data not just from the primary location, but removing it completely (from all locations in disk and memory), a technique will have to be built to track the copies and delete all of them. Thus, due to the lack of system specifications, ambiguities arise which can expose the company to legal action.

Ambiguity in interpreting GDPR has been a cause of concern for the research community as evidenced by the work of the Article 29 Data Protection Working Party (AWP29) [12] and EDPB [13], which issues clarifications and recommendations on how organizations may attain compliance. These reports, however, clarify only limited aspects of the GDPR regulation. Furthermore, prior work [19] has shown that, at times, AWP29's clarifications and recommendations have been unsound [19, 53] and do not meet the desired compliance.

To bridge the gap between ambiguous legal specifications and grounded (system-level) technical specifications that can serve as a blueprint for compliance in systems, we need a model with a set of concepts that can be used to translate the data governing requirements in regulations into a set of well-defined specifications of dynamic system behavior. Such a model must: (a) consist of a set of (data processing) concepts to fully describe data regulations; (b) allow different valid interpretations of the regulations; (c) allow for unambiguous interpretations of the concepts to be mapped to system-actions using which they can be implemented in a system.

Using such a model, organizations working along with regulatory bodies can agree upon possible desirable interpretations of concepts/properties and attain demonstrable compliance.

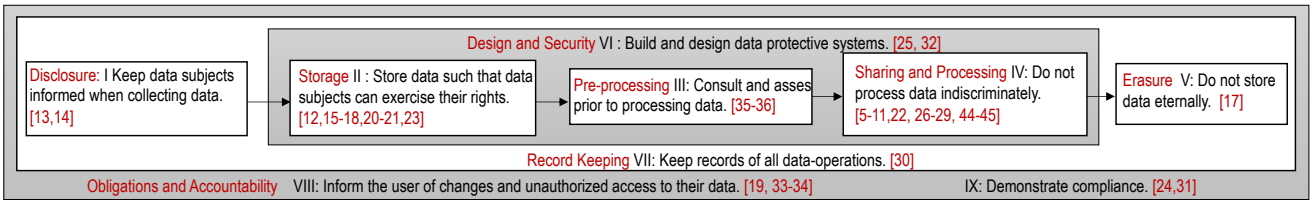


Figure 1: The GDPR requirements for data governance stated as informal invariants and the grouped articles.

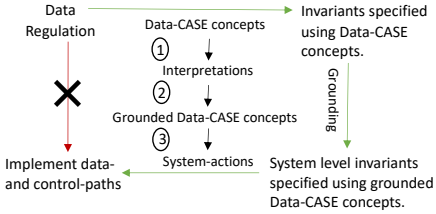


Figure 2: Schematic representation of Data-CASE

Towards the goal of developing such a model, we propose Data-CASE which stands for *Data Collection, Access, Sharing, and Erasure* model. Data-CASE consists of a small set of key system concepts (e.g., erasure, encryption, policies), referred to as *Data-CASE concepts*. Concepts in Data-CASE are chosen such that the specifications in regulations that relate to the requirements of systems can be expressed formally using these concepts as invariants in a logic framework. Each Data-CASE concept may have several possible interpretations, e.g., deletion/erasure may have different interpretations as discussed earlier. Data-CASE allows for such interpretations to be formally defined (①, Figure 2). Furthermore, the system developers/deployers can choose the specific interpretation of the concepts they wish to support in their system (②, Fig. 2). Such an interpretation is then finally mapped to system-level actions — *system-actions* — (③, Fig. 2) to achieve the specified concept interpretation. We refer to this process of choosing a specific interpretation for a given concept and formally specifying the chosen interpretation as the process of *grounding* and refer to the formally defined concepts as the *grounded* interpretations. Note that the system-action to which the grounded concept is mapped is system dependent. In cases where a system-action is not supported to exactly implement the chosen interpretation, the system might need to be retrofitted or changed to support the necessary system-actions. Examples of system-actions include DELETE and VACUUM in PSQL; deleteOne and remove in MongoDB. System-actions may also include user defined functions. Note that Data-CASE is neither a system nor system-dependent. It is a formal framework that can be used for reasoning about compliance in any system by creating mappings between grounded concepts and system-actions.

Organization. In Section (§) 2, we propose a basic model for Data-CASE focusing on only key concepts we think such a model must contain. In §3, we show how concepts defined in §2 can be grounded to remove ambiguity by demonstrating, as an example, how valid interpretations of data erasure can be formally stated using Data-CASE. In §4 we show some use cases of Data-CASE. §5 discusses related work. §6 highlights the challenges towards the goal of a fully specified model that can abstract compliance for data processing system development and deployment.

2 DATA-CASE MODEL

Data regulations specify the kind of data that falls under their domain and legislate how such data is handled as it flows through a data processing system and the responsibilities of entities processing such data. So, Data-CASE groups the requirements of a given data regulation under the following eight categories (the first five corresponding to the data life cycle and the remaining three to system properties) —(1) Disclosure, (2) Storage, (3) Pre-processing, (4) Sharing and Processing, and (6) Erasure, (7) record keeping, and (8) obligations and accountability. We illustrate this schematically in Figure 1 where we also group the articles of GDPR (only those that legislate data processing and impact system design [68]) under these categories.

To capture the data governing principles in a data regulation, the first step is to logically differentiate personal data from other kinds of data. In addition, it is essential to capture the provenance between various kinds of data in a system. Since data regulations grant rights to owners of personal data, there is a need to support user policies, track their evolution over time, and validate them. Finally, when data is accessed/used, it is done so for specific purposes by specific entities which are restricted by policies set by the owner of such data or the regulation. Below, we develop a simple set of concepts that capture the above.

2.1 Data Processing Concepts in Data-CASE

As data flows through the data-life cycle, it is collected from the data-subject by the controller who might share it with processors. Auditors verify and certify compliance. In Data-CASE, these roles are referred to as entities. We denote them with e. As a running example, consider Netflix collects the credit card information of its subscribers and stores it on the AWS cloud.

The concept of **data unit** refers to the finest granularity at which we refer to data in Data-CASE. The granularity depends on the system, application, as well as data regulation. E.g., in a website collecting info about click-stream, a specific user's data might be a data unit. In a sensor such as a camera, the unit might correspond to all data (irrespective of who is in it) of a camera within a certain interval. For a service provider like Netflix, credit card information can be considered a data unit.

We denote a data unit as a tuple $X = (S, O, V, P)$ where S is the data-subject—the entity whom the data identifies; O is the origin—where the data was collected from; V is a set $\{(v_1, t_1), (v_2, t_2), \ldots\}$ of values where v_i is the value at time t_i, and P is the set of associated policies. A collection of data units is denoted as $\vec{X}$.

Data-CASE classifies data units into three categories — (1) **base data**, which is directly or indirectly collected, (2) **derived data**, which is obtained from base data, and (3) **metadata**, which includes data-subject, policies, etc. A *derived* data unit has the same four aspects as base data except that data-subject (and origin) are possibly varying sets of the data-subjects (and origins) of the base data from which it was derived. The aspects of the derived data unit are aggregated over the aspects of the base data.

A task or service, for which collected data is used, identifies its **purpose** of data processing. Collected base data can have more than one purpose. E.g., Netflix collects credit card information

for billing; saves view-history for targeted advertisements, etc. The flow of data units through various stages of processing in a system is controlled using policies. A **policy** on a data unit X is a tuple $\langle p, e, t_b, t_f \rangle$ is a constraint specifying that an entity e can access the data unit for purpose p from time t_b to t_f.

Example: The policy $\pi_1 = \langle \texttt{billing}, Netflix, 010123, 01012024 \rangle$ for a data unit $X = credit\_card$ of user 1234 states that Netflix has access to X for the purpose of billing from 01/01/23 through 01/01/24. The policy $\pi_2 = \langle \texttt{retention}, AWS, 010123, 010124 \rangle$ on X states that AWS can retain this data from 01/01/23 through 01/01/24. For a data unit X, we write $V(t)$ to denote its value at time t and $P(t) := \{(p, e, t_b, t_f) \in P \mid t_b \leq t \leq t_f\}$ to denote the set of policies on X at time t. The **state** of a data unit X at a given time are the values of its aspect at that time and are denoted as $X(t) = (S(t), O(t), V(t), P(t))$. The state of a database is the collection of the states of all data units in the database.

Example: The state of X in the previous example at time $t = 00{:}10$ on 02/26/23 is $X(t) = (1234, 0, credit\_card\_info, \{\pi_1, \pi_2, \ldots\})$.

We refer to any operation that changes the state of data units as an **action**. Actions include the creation and deletion of data units, changes to the value of a data unit, and reads and writes on any aspect of a data unit. An action can influence one or more data units. For an action τ on data unit X, we denote the changed state of X with $\tau(X)$. Actions on data units in a database D give rise to a series of states $\mathcal{D}_1, \mathcal{D}_2, \ldots$. Actions can produce derived data. A derived data unit $Y = (S_Y, O_Y, V_Y, P_Y)$ produced by action τ from a collection of base data units $\vec{X}$ has S_Y and O_Y as the union of all the data-subjects and origins of data units $\vec{X}$, respectively. The set of policies P_Y is generally a restriction of the policies of the data units in $\vec{X}$.

Data regulations often require monitoring how data is processed or changes over time. Each action on a data unit is denoted as an **action-history tuple**. A collection of action-history tuples is called an **action-history.** For a data unit X, and a database D,

- an action-history tuple is given by $(X, p, e, \tau(X), t)$ denoting that entity e performed action τ on X for purpose p at time t.
- action-history of X denoted, $\mathcal{H}(X)$, is the set of all actions on X, i.e., $\mathcal{H}(X) = \{(X, p, e, \tau_i(X), t)_i\}_{i=1}^{n}$.

Example: The action-history tuple (1234, comp, $Netflix$, CtrC1234, 010223) records that on 01/02/23, Netflix made a contract to collect data of user 1234. Such a contract gets the consent of the user to set policies π_1 and π_2 in previous examples. Similarly, the tuple $(X, \texttt{billing}, Netflix, \texttt{read}(credit\_card), 0010 - 022623)$ records that Netflix accessed the credit card information of 1234 for billing at 00:10 on 02/26/23.

Data regulations specify what constitutes lawful data processing. Data-CASE abstracts lawful data processing as **policy-consistent** data processing. For a data unit $X = (S, O, V, P)$, action τ on X for purpose p at time t, we say that the action-history tuple $(X, p, e, \tau(X), t)$ on data unit X is *policy-consistent* if there exists a policy $\langle p, e, t_b, t_f \rangle$ in $P(t)$ in the state $(S, O, V(t), P(t))$ of data unit X or the action in the tuple is required by a data regulation. We say that actions on X are policy-consistent if every action-history tuple in $\mathcal{H}(X)$ is policy-consistent.

2.2 Formal Invariants For Compliance

Having described the set of concepts above (data unit, policy-consistent action, etc.), we can now specify data regulations formally in the form of invariances. We provide two examples.

GDPR Article (denoted $\mathcal{G}$) 6 defines when processing personal data is lawful. Data-CASE abstracts this notion using the concept of *policy-consistent data processing*. It can be stated as: For all data units X, and for all actions τ on X, it holds that τ is policy-consistent. The legally permissible grounds and data-subject's consent for processing data can be encoded as specific purposes through policies in Data-CASE.

Consider $\mathcal{G}17$. It requires that personal data be not retained longer than necessary for the purpose they were collected and they be deleted without undue delay. Formally, this can be specified as follows. For all data units $X = (S, O, V, P)$, there exists a policy $\pi \in P$ such that $\pi = \langle \texttt{compliance-erase}, e, t_b, t_f \rangle$ and the last access tuple on X is $(q, \texttt{compliance-erase}, e, \texttt{erase}(X), t)$ s.t. $t \leq t_f$. The above statement states that every data unit X has a policy associated with it, which states that the data unit has to be erased due to compliance requirements at a specified time. Moreover, the last action on the data unit X is `erase(X)` at a time earlier than the time within which the policy requires the data unit to be deleted.

Observe that the above invariants capture $\mathcal{G}6$ and $\mathcal{G}17$ formally. However, the Data-CASE concepts *erasure* and *policies* are still open to multiple valid interpretations. In the next section, we discuss the process of grounding such concepts so that they can be mapped to specific implementations in systems.

3 GROUNDING CONCEPTS

The process of grounding consists of mapping a concept to a unique interpretation and formalizing it in Data-CASE. We illustrate the process through *erasure*. Recent work [60–62] has shown that many complexities arise when interpreting and implementing erasure and can have a significant impact on system performance. Besides, erasure is a requirement of most regulations, and has received considerable attention recently [75, 76].

3.1 Data erasure

In a system, erasure can be interpreted in various ways. We consider four interpretations - inaccessibility, deletion, strong deletion, and permanent deletion.

- We say that data is *reversibly inaccessible* in a system when it cannot be read by any data-subjects in the system but remains accessible to the controller or processor. Often, in such cases, it can be accessed by the data-subject after a specific action.
- We say that data has been *deleted* when the data and all its copies have been physically erased.
- We say that data has been *strongly deleted* when it has been deleted and all dependent data, where the data-subject is identifiable, has been deleted.
- We say that data has been *permanently deleted* when it has been strongly deleted, and some advanced physical drive sanitation technique has been used.

Observe that these interpretations can be ordered based on their restrictiveness. For example, strongly delete implies delete. This gives rise to the notion of strictness of interpretation of compliance. Figure 3 depicts the temporal relationship between the different interpretations. While some notions of erasure are arguably better than others when privacy and security of personal data are considered, these deletion methods have different overheads and may or may not be considered practical or feasible. To ground these interpretations, we identify three properties.

- *Erasure-inconsistent read:* (Illegal Reads-IR) We say that there is an erasure-inconsistent read on data unit X if there exists the tuple $(X, q, p, e, \texttt{read}(X), t_j) \in \mathcal{H}(X)$ and in $X(t_j)$ we have

Table 1: Interpretations of erasure and their characteristics.
✓ indicates feasibility and × indicates not.

Erasure	IR	II	Inv	PSQL System-Action(s)
reversibly accessible	×	✓	✓	Add new attribute
delete	×	✓	×	DELETE+VACUUM
strong delete	×	×	×	DELETE+VACUUM FULL
permanently delete	×	×	×	Not supported

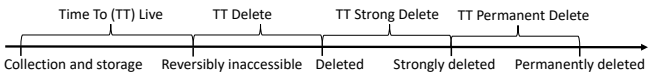


Figure 3: Data Erasure Timeline

$P(t_j) = \emptyset$, i.e., X was read although there were no policies authorizing it.

• *Erasure-inconsistent inference*:(Illegal Inference-II) We say that there is an erasure-inconsistent inference on data unit X if there exists the tuple $(X, q, p, e, \texttt{erase}(X), t_j) \in \mathcal{H}(X)$ and $X = f(Y)$ where Y is other data units and f is some dependency that can be used to reconstruct X from Y. i.e., although X has been erased, it can still be inferred using dependent data, provenance data, or from other data units in the database.

• *Transformation invertibility*:(Invertibility-Inv.) A data unit X is usually transformed to some $f(X)$ (e.g., an encryption function, a function that rewrites X with 0 bits, etc.) to prevent illegal reads and illegal inferences. The transformation (function) may be invertible, i.e., recoverable, or non-invertible.

We can formally ground the four different deletion notions. Table 1 characterizes the different notions of erasure in terms of the actions introduced earlier. Observe that although *strong delete* and *permanent delete* have the same properties, the latter entails the additional step of advanced data sanitization like [21].

3.2 Other Concepts

Other Data-CASE concepts such as purpose, histories, and policies also need to be grounded when reasoning with systems being compliant with a given data regulation. Grounding concepts require a careful analysis of actions different systems use for these concepts, as well as, interactions between the actions. For example, to ground histories, one has to consider various logs a system maintains, their granularity, and uses — logs may be temporary or kept for a long duration to not only recover data but also to support the rights of data-subjects. Furthermore, logs directly impact requirements like demonstrating compliance, system recovery, and data erasure. Similarly, purposes need to be grounded to specific actions. A purpose typically calls for a set of authorized actions. E.g. the purpose of billing only allows the credit card information to be read and processed with the bank and not share it with a third party. This, in turn, directly impacts policy-consistent data processing. Finally, each grounded concept needs to be mapped to system-actions.

Thus, grounding all concepts which enable study interactions between them in a given system is a complex task and is fundamental to defining what compliance means for that system.

4 USING DATA-CASE

We illustrate how we envision Data-CASE to be used using some experiments. We classify the uses based on the end-user. All our implementations were run on Oracle VM VirtualBox with a 6-cores (12 threads) AMD Ryzen 5 5600x 3.7GHz processor, 16GB RAM (DDR4-3200), and 50GB of disk space.

4.1 Service Providers & App. Developers

Service providers and application developers often have their own system requirements which influence which database engine they use. Data-CASE offers a principled approach for them to identify desirable database properties and the corresponding data governance principles. This helps in choosing an appropriate data service provider which meets the desired requirements.

Case Study 1: Continuing our example, suppose MetaSpace, a service provider, wants to offer strong semantics of erasure to its customer to satisfy the requirements of $\mathcal{G}17$. They want to analyze which interpretations of erase can be supported by their database, PSQL, and their associated costs. To that end, they ground erase in Data-CASE (see Section 3) and identify system-actions [56] offered by PSQL [55] that can implement the groundings. We map the grounded erasure interpretations to system-actions supported by PSQL in Table 1. To assess how the implementation of each grounding impact system performance, they are benchmarked using the customer workload (20% deletes on data, rest are reads.) in GDPRbench [68]. The results are summarized in Figure 4(a). VACUUM+DELETE surprisingly takes less time than only DELETE for the GDPRBench workload. VACUUM reclaims storage occupied deleted tuples that are not physically removed when only DELETE is used. The extra time taken by VACUUM in the deletion operations (20%) is offset by the improved performance of the other operations (80%) in the workload. Note that the expected performance is observed for a workload composed only of deletions.

4.2 Database providers

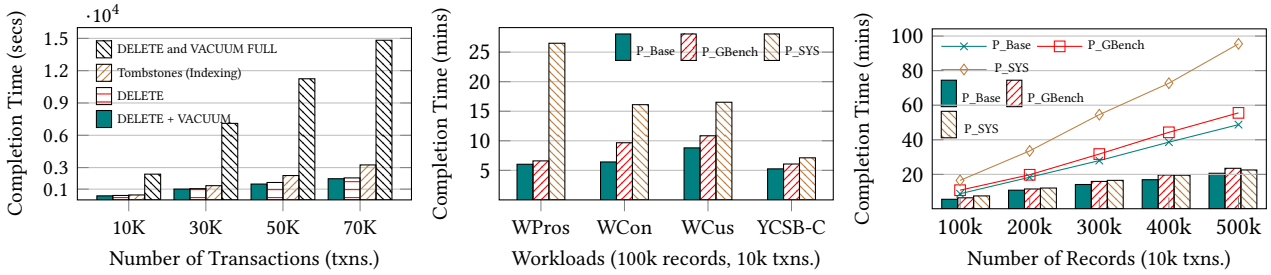
Database providers like Oracle are often faced with the challenge of how to design databases that are compliant with data regulations or how to retrofit existing deployments to make them compliant. In Data-CASE, database providers can express concepts and actions supported in their existing deployments in terms of fundamental system properties or the effects the actions have on personal data. This fixes the interpretations of the concepts defined in Data-CASE. Now, a set of invariants is obtained that express the requirements for data governance. The system can then be retrofitted to meet those requirements it initially didn't. When building new systems, Data-CASE lets the designer consider a wide range of interpretations, analyze their possible overheads, impacts on data- and control- paths, and expenses.

Case Study 2: Consider RelDB, which offers its services to various service providers. System designers at RelDB want to determine how to make their system, which runs on PSQL, GDPR compliant efficiently — minimize costs and impact on system performance and at the same time offer meaningful interpretations useful for its clients. We show Data-CASE supports this.

Three interpretations of GDPR-compliance are implemented using associated groundings of concepts and by extending PSQL to support these groundings using system-actions. These are:

1) P_Base: The system implements role-based access control using roles, role attributes, and role memberships. It implements histories using native `csv` logging and setting up security policy to record query responses at row-level and the data is encrypted using AES-256 [2]. It implements deletes (see Table 1 for grounding) to erase data using DELETE + VACUUM. The first interpretation of compliance is the least restrictive, and thus, is expected to have the least impact on system performance.

2) P_GBench: The system stores policies and other metadata in a table separate from the one containing personal data. Thus, all



(a) Interpretations of Data Erasure in PSQL on WCus **(b) Completion time for workloads** **(c) WCus (lines) & YCSB-C(bars) workloads**

Figure 4: (a) Impact of different interpretations of data erasure on PSQL. (b) Completion time. (c) Scalability.

queries must perform joins to implement appropriate policies. Histories are implemented by logging all queries and responses (no csv logs). Data is encrypted using LUKS(SHA 256) [45, 67]. Erasure is implemented using DELETE in PSQL.

3) P_SYS: The system implements fine-grained access control (FGAC) [11]. Since PSQL does not support FGAC, it is retrofitted with a middleware that comprises Sieve [51] and associated metadata which implements FGAC by exploiting a variety of its features such as UDFs, index usage hints, etc. to scale to a large number of policies. Data units and logs are encrypted using AES-128 [2] and erasure is implemented using DELETE + VACUUM FULL as well as deleting logs of the data units being deleted.

Evidently, the three systems have different interpretations of GDPR-compliance which changes the code and design. To validate and measure the expected varying amount of impact of the interpretations on the performance of these three systems, they are evaluated using the GDPRBench [68] and the industry-standard Yahoo Cloud Servicing Benchmark (YCSB) [20] (Workload-C). GDPRBench has three workloads namely Controller[WCon] (25% create, 25% deletes, and 50% updates to metadata), Processor[WPro] (80% reads of data using keys, and 20% reads of data using metadata), and Customer[WCus] (20% each of reads, updates, and deletes of data, and reads and updates of metadata). We enriched the data records in GDPRBench with the Mall dataset from [51] comprising simulated data generated from personal devices in a shopping complex. Each record consists of a personal data-id and the recorded date and time generated using the SmartBench simulator [35].

Metrics. We analyzed *completion time*, i.e., the total time taken to complete all the queries for each workload. To evaluate the "Metadata explosion" [69] associated with each grounding/ implementation, we define *space factor* as the ratio of the total size of the database to the total size of personal data in it.

Summary of experiments and evaluation. Figure 4(b) shows the overhead of implementations P_Base, P_GBench, and P_SYS across the four workloads (each with 100k records and 10k transactions). In each case, the overhead of P_SYS is higher compared to P_GBench which is higher than P_Base. This is expected since the implementations use increasingly restrictive notions of compliance. The overheads in P_GBench are small in comparison to those in P_Base in WPro which consists of read queries. This small overhead is due to a slight increase in the information being logged. In contrast, since P_SYS requires fine-grained policies to be checked, it incurs significant overhead. The difference between the completion times for P_Base and P_GBench in WCon is larger compared to that in WCus and WPro due to a larger share of create, delete, and updates WCon. Such operations require more metadata access and logging. Likewise, the slowdown due to policies is more profound in WPro in P_SYS since it contains a larger number of read queries (100% compared to others that are 50% percent). For WCon, P_SYS still has a higher completion time compared to P_Bench and P_Base even though the workload is comprised of create, delete, and update and no reads which invoke expensive policy checks. However, all policies are logged at the time of all the operations to implement demonstrable accountability for logging requirements.

The YCSB Workload-C takes the least time to complete in each implementation since it does not require any associated metadata operations thereby highlighting the impact of changes required for compliance is small on non-GDPR operations.

Scalability was explored by increasing the volume of data but keeping the number of total transactions the same. Fig. 4(c) plots the completion time for the three systems running the WCus workload with an increasing number of data records and 10k transactions. As observed in earlier experiments, P_Base takes the least time to complete the workload and P_SYS takes the longest. As expected, P_SYS is impacted the most by the increase in the size of data whereas the effect in P_Base is the least. Groundings and their corresponding implementations impact database size. The size of PID remains the same (7 MB) but that of the metadata changes across interpretations (see Table 2). P_GBench and P_SYS use indices that occupy additional space. Recall that P_SYS uses Sieve [51] which uses additional metadata.

Table 2: Storage space overhead corresponding to Figure 4(b). The total size of P_GBench and P_SYS include indices.

System	Personal data size (MB)	Metadata size (MB)	Total DB size (MB)	Space factor
P_Base	7	14	21	3×
P_GBench	7	10	26	3.7×
P_SYS	7	111	120	17.1×

In summary, for entities like RelDB, our model provides a systematic approach to fix an interpretation of a data regulation and identify system-actions required to implement the interpretation and make design choices such as adding components, plugins, etc. to support chosen interpretations and study their characteristics. This paves the way to achieving demonstrable compliance.

4.3 Multinational organizations

These often need to comply with conflicting and varying principles of data governance. GDPR itself allows EU member states to define their own data processing principles. Moreover, countries around the world have different data regulations. Entities are not ready to deal with the resulting complexities [32]. Data-CASE supports varying interpretations of data regulations and makes the process of mapping data regulation requirements to precise

system-actions transparent and unambiguous. Thus, it can help make decisions such as data geo-location, processors to use, and the consequences on services and features offered to its clients.

4.4 Other Uses

Privacy Impact Assessment (PIA): GDPR ($\mathcal{G}35$) imposes the burden of a PIA on controllers prior to starting data processing. Such pre-deployment assessments of new, potentially high risk to the privacy and security of personal data, are often required by data regulations. Data-CASE supports impact assessments by providing system designers with system-actions (to implement specific groundings of the concepts) corresponding to each step in the data processing pipeline and their properties and interactions with each other. Once the risks have been identified and assessed, Data-CASE supports in implementing specific system-actions to mitigate those risks.

Regulatory Agencies: All data regulations establish regulatory agencies (e.g. see $\mathcal{G}$ 31) which certify that a data processing system is, indeed, compliant with that data regulation. For example, GDPR is enforced by Individual data protection authorities (DPAs) from the 27 EU member states. These agencies often have conflicting, non-transparent certifying processes and have repeatedly expressed frustrations with data regulations [29]. Data-CASE provides such agencies to identify groundings of concepts that are required, at minimum, to be in compliance with a data regulation. Conversely, agencies may require entities to use frameworks like Data-CASE to demonstrate the groundings of data processing adopted in a system and the system-actions that implement them.

5 RELATED WORK

Conflicting priorities of data regulations and prevalent database systems [70, 71], motivated preliminary studies in [68] which showed that GDPR-compliance severely impacts the performance of databases. Domain-specific work like [39, 77] explores the consequence of GDPR in named data networking and Healthcare Systems, respectively. The consequences on policy and privacy management have been investigated in [9, 34, 79].

Retrofitting databases to make them compliant has been explored in [3, 23, 44, 68] and new, compliant-by-construction, systems have been proposed in [41, 46, 64]. Frameworks to implement GDPR compliance have been explored in several prior works, especially in the context of data retention/erasure [60–62, 66] and policies. The work in [47] explores privacy policies in large-scale cloud systems, [27] explores policy compliance in web frameworks, [73] explores compliance in operating systems, [44] builds a visual tool for managing data flow in systems, while [43] explores auditing and retention policies in databases. A middleware layer to implement consent management [22] and access control [51, 52] in databases have also been explored.

Unlike such frameworks, our goal in this paper is to develop a simple model for data and data processing that can be used to define GDPR/data regulation requirements formally such that designers of systems such as the above can precisely define their interpretation of regulations and establish compliance by illustrating that their software techniques indeed meet the formal requirements. In this sense, our vision is more related to prior approaches such as [40, 57, 58] that have explored formal logic-based GDPR specifications that support verification of compliance through model checking, such as in [7, 17]. But these logic-based specifications at the level of broad data processing concepts remain vague from a system-compliance perspective. Unlike Data-CASE, these frameworks do not support the specification of how the concepts are interpreted or implemented in the system being verified. Such work complements Data-CASE with formal specifications potentially serving as *invariants* in Data-CASE.

A rich line of work exists on modeling, specifying, implementing policies, and auditing [4–6, 17, 26, 36, 65]. These can be a part of Data-CASE and the middlewares to audit such policies can support system-actions to maintain related invariants in Data-CASE. Privacy frameworks such as contextual integrity [8, 72] and origin privacy are related but orthogonal to our contributions. Our goal is not to create a new way of specifying what privacy should mean. Instead, given what privacy should mean, and data processing concepts grounded based on it, our framework's goal is to provide ways to reason about whether a given system is compliant. Some compliance guidelines, specific to data regulations, are available from Governmental organizations, white papers, and blog posts [31, 33, 37, 74, 80] and offer some insights.

6 CHALLENGES AHEAD

GDPR violations (tracked by [30]) related to non-compliant systems continue to rise exponentially [18, 30, 38]; the latest being that by Meta in May 2023 [63] which incurred the largest fine till date. Our paper makes a case that designing, deploying, and reasoning about the compliance of systems to data regulations like GDPR and others requires a formal framework to express data processing concepts. The paper proposes such a formal framework entitled Data CASE and shows how data governing principles of data regulations can be formalized in the form of grounding concepts defined into concrete system-actions and by defining invariants using these concepts. A full realization of our vision of a formal framework to support the development and deployment of compliant systems opens up several challenges:

- **Completeness and correctness:** Demonstrating the correctness of a formal framework like Data-CASE and to what degree it can capture the system requirements of a data regulation remain unexplored. This is an important step towards compliance.
- **Grounding concepts:** We focused on erasure and its possible interpretations in systems. Other concepts which capture the requirements of data regulations need to be defined. Once defined, special attention needs to be given to the interactions between and compatibility of different possible interpretations of the concepts.
- **From a formal framework of compliance to a system to support compliance:** Data-CASE supports compliance-related decision-making. Automating this process will enable us to build a comprehensive tool that can be retrofitted on any non-compliant system to make it compliant with a given data regulation. Traditionally, retrofitting for compliance requires multiple tools and often complicates data flow [23].

In summary, we believe Data-CASE opens up a new direction of exciting research possibilities.

ACKNOWLEDGMENTS

The first author was supported in part by the Irvine Initiative in AI, Law, and Society and the HPI Research Center in Machine Learning and Data Science at UC Irvine. This work was supported by NSF Grants No. 2032525, 1545071, 1527536, 1952247, 2008993, 2133391, and 2245372. The authors thank the reviewers for their feedback.

REFERENCES

[1] Daniel Abadi, Anastasia Ailamaki, David Andersen, Peter Bailis, Magdalena Balazinska, Philip A. Bernstein, Peter Boncz, Surajit Chaudhuri, Alvin Cheung, Anhai Doan, Luna Dong, Michael J. Franklin, Juliana Freire, Alon Halevy, Joseph M. Hellerstein, Stratos Idreos, Donald Kossmann, Tim Kraska, Sailesh Krishnamurthy, Volker Markl, Sergey Melnik, Tova Milo, C. Mohan, Thomas Neumann, Beng Chin Ooi, Fatma Ozcan, Jignesh Patel, Andrew Pavlo, Raluca Popa, Raghu Ramakrishnan, Christopher Re, Michael Stonebraker, and Dan Suciu. The seattle report on database research. *Commun. ACM*, 65(8):72–79, jul 2022. doi:10.1145/3524284.

[2] ADVANCED ENCRYPTION STANDARD (AES) . https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf, 2001.

[3] Archita Agarwal, Marilyn George, Aaron Jeyaraj, and Malte Schwarzkopf. Retrofitting gdpr compliance onto legacy databases. *Proceedings of the VLDB Endowment*, 15(4):958–970, 2021.

[4] Mohammad Javad Amiri, Tristan Allard, Divyakant Agrawal, and Amr El Abbadi. Prever: Towards private regulated verified data. In *EDBT 2022-International Conference on Extending Database Technology*, 2022.

[5] Emma Arfelt, David Basin, and Søren Debois. Monitoring the gdpr. In *Computer Security – ESORICS 2019: 24th European Symposium on Research in Computer Security, Luxembourg, September 23–27, 2019, Proceedings, Part I*, page 681–699, Berlin, Heidelberg, 2019. Springer-Verlag. doi:10.1007/978-3-030-29959-0_33.

[6] Ahmed A. Ataullah, Ashraf Aboulnaga, and Frank Wm. Tompa. Records retention in relational database systems. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, page 873–882, New York, NY, USA, 2008. Association for Computing Machinery. doi:10.1145/1458082.1458197.

[7] Masoud Barati, Omer Rana, Ioan Petri, and George Theodorakopoulos. Gdpr compliance verification in internet of things. *IEEE Access*, 8:119697–119709, 2020. doi:10.1109/ACCESS.2020.3005509.

[8] A. Barth, A. Datta, J.C. Mitchell, and H. Nissenbaum. Privacy and contextual integrity: framework and applications. In *2006 IEEE Symposium on Security and Privacy (S&P'06)*, pages 15 pp.–198, 2006. doi:10.1109/SP.2006.32.

[9] David Basin, Søren Debois, and Thomas Hildebrandt. On purpose and by necessity: compliance under the gdpr. In *International Conference on Financial Cryptography and Data Security*, pages 20–37. Springer, 2018.

[10] BBC. Whatsapp privacy policy tweaked in europe after record fine. https://www.bbc.com/news/technology-59348921, last accessed on 2022-01-01.

[11] Elisa Bertino, Gabriel Ghinita, and Ashish Kamra. Access control for databases: Concepts and systems. *Foundations and Trends® in Databases*, 3(1–2):1–148, 2011. URL: http://dx.doi.org/10.1561/1900000014, doi:10.1561/1900000014.

[12] European Data Protection Board. Endorsed working party 29 guidelines. https://edpb.europa.eu/our-work-tools/general-guidance/endorsed-wp29-guidelines_en, last accessed on 2022-04-01.

[13] European Data Protection Board. Guidelines, recommendations, best practices. https://edpb.europa.eu/our-work-tools/general-guidance/guidelines-recommendations-best-practices_en, last accessed on 2022-04-01.

[14] Ryan Browne. Fines for breaches of eu privacy law spike sevenfold to $1.2 billion, as big tech bears the brunt. https://www.cnbc.com/2022/01/18/fines-for-breaches-of-eu-gdpr-privacy-law-spike-sevenfold.html, last accessed on 2022-01-01.

[15] CCPA. Title 1.81.5. california consumer privacy act of 2018 [1798.100 - 1798.199.100]. https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5, last accessed on 2022-02-01.

[16] Eline Chivot and Daniel Castro. What the evidence shows about the impact of the gdpr after one year. https://datainnovation.org/2019/06/what-the-evidence-shows-about-the-impact-of-the-gdpr-after-one-year, last accessed on 2022-01-01.

[17] Omar Chowdhury, Limin Jia, Deepak Garg, and Anupam Datta. Temporal mode-checking for runtime monitoring of privacy policies. In Armin Biere and Roderick Bloem, editors, *Computer Aided Verification*, pages 131–149, Cham, 2014. Springer International Publishing.

[18] CNIL. The CNIL's restricted committee imposes a financial penalty of 50 Million euros against Google LLC, 2019. https://www.cnil.fr/en/cnils-restricted-committee-imposes-financial-penalty-50-million-euros-against-google-llc, last accessed on 2021-10-12.

[19] Aloni Cohen and Kobbi Nissim. Towards formalizing the gdpr's notion of singling out. *Proceedings of the National Academy of Sciences*, 117(15):8344–8352, 2020.

[20] Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, and Russell Sears. Benchmarking cloud serving systems with ycsb. In *SoCC '10*, 2010.

[21] Defense Counterintelligence and Security Agency. National industrial security program, 2019. https://www.dcsa.mil/ma/ctp/io/fcb/nisp/, last accessed on 2021-08-23.

[22] Said Daoudagh, Eda Marchetti, Vincenzo Savarino, Roberto Di Bernardo, and Marco Alessi. How to improve the gdpr compliance through consent management and access control. In *ICISSP*, pages 534–541, 2021.

[23] Maryam Davari and Elisa Bertino. Access control model extensions to support data privacy protection based on gdpr. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4017–4024. IEEE, 2019.

[24] Jessica Davies. Gdpr mayhem: Programmatic ad buying plummets in europe, 2018. https://digiday.com/media/gdpr-mayhem-programmatic-ad-buying-plummets-europe/, last accessed on 2021-10-12.

[25] Jessica Davies. After gdpr, the new york times cut off ad exchanges in europe — and kept growing ad revenue, 2019. https://digiday.com/media/gumgumtest-new-york-times-gdpr-cut-off-ad-exchanges-europe-ad-revenue/, last accessed on 2021-10-12.

[26] Henry DeYoung, Deepak Garg, Limin Jia, Dilsun Kaynar, and Anupam Datta. Experiences in the logical specification of the hipaa and glba privacy laws. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '10, page 73–82, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1866919.1866930.

[27] Mafalda Ferreira, Tiago Brito, José Fragoso Santos, and Nuno Santos. Rulekeeper: Gdpr-aware personal data compliance for web frameworks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1014–1031. IEEE Computer Society, 2022.

[28] Chirs Fox. Facebook and instagram disable features in europe. https://www.bbc.com/news/technology-55350795, last accessed on 2022-01-01.

[29] Europe's sweeping privacy rule was supposed to change the internet, but so far it's mostly created frustration for users, companies, and regulators. https://www.cnbc.com/2019/05/04/gdpr-has-frustrated-users-and-regulators.html, last accessed on 2023-04-20.

[30] GDPR Enforcement Tracker - List of GDPR fines. https://www.enforcementtracker.com/, last accessed on 2022-02-01.

[31] GDPR. Complete guide to gdpr compliance, 2016. https://gdpr.eu/ last accessed on 10-12-2021.

[32] How gdpr is failing. https://www.wired.com/story/gdpr-2022/, last accessed on 2023-04-20.

[33] Google. Google cloud & the general data protection regulation (gdpr). https://cloud.google.com/privacy/gdpr, last accessed on 2022-01-01.

[34] Samuel Greengard. Weighing the impact of gdpr. *Communications of the ACM*, 61(11):16–18, 2018.

[35] Peeyush Gupta, Michael J Carey, Sharad Mehrotra, and oberto Yus. Smartbench: a benchmark for data management in smart spaces. *Proceedings of the VLDB Endowment*, 13(12):1807–1820, 2020.

[36] Ragib Hasan and Marianne Winslett. Trustworthy vacuuming and litigation holds in long-term high-integrity records retention. In *Proceedings of the 13th International Conference on Extending Database Technology*, EDBT '10, page 621–632, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1739041.1739115.

[37] Google Inc. Google cloud whitepaper. google cloud and the gdpr. *Technical Report*, 2018.

[38] UK Information Commissioner's Office. Mariott international inc. penalty notice (case ref: Com0804337), 2020. https://ico.org.uk/media/action-weve-taken/mpns/2618524/marriott-international-inc-mpn-20201030.pdf, last accessed on 2021-08-24.

[39] Florian Kammueller. Formal modeling and analysis of data protection for gdpr compliance of iot healthcare systems. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3319–3324. IEEE, 2018.

[40] Farzane Karami, David Basin, and Einar Broch Johnsen. Dpl: A language for gdpr enforcement. In *2022 IEEE 35th Computer Security Foundations Symposium (CSF)*, pages 112–129. IEEE, 2022.

[41] Tim Kraska, Michael Stonebraker, Michael Brodie, Sacha Servan-Schreiber, and Daniel Weitzner. Schengendb: A data protection database proposal. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, pages 24–38. Springer, 2019.

[42] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, 44(2):35–40, 2010.

[43] Wentian Lu and Gerome Miklau. Auditguard: A system for database auditing under retention restrictions. *Proc. VLDB Endow.*, 1(2):1484–1487, aug 2008. doi:10.14778/1454159.1454207.

[44] Connor Luckett, Andrew Crotty, Alex Galakatos, and Ugur Cetintemel. Odlaw: A tool for retroactive gdpr compliance. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2709–2712, 2021. doi:10.1109/ICDE51399.2021.00310.

[45] Cryptsetup and luks - open-source disk encryption. https://gitlab.com/cryptsetup/cryptsetup, last accessed on 2022-01-01.

[46] Soumyadeb Mitra, Marianne Winslett, Richard T. Snodgrass, Shashank Yaduvanshi, and Sumedh Ambokar. An architecture for regulatory compliant database management. *2009 IEEE 25th International Conference on Data Engineering*, pages 162–173, 2009.

[47] Jayashree Mohan, Melissa Wasserman, and Vijay Chidambaram. Analyzing gdpr compliance through the lens of privacy policy. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, pages 82–95. Springer, 2019.

[48] Faisal Nawab and Mohammad Sadoghi. Consensus in data management: From distributed commit to blockchain. *Foundations and Trends® in Databases*, 12(4):221–364, 2023. URL: http://dx.doi.org/10.1561/1900000075, doi:10.1561/1900000075.

[49] Council of the EU. Data grovernance act, press release 16 may 2022, 2022. https://www.consilium.europa.eu/en/press/press-releases/2022/05/16/le-conseil-approuve-l-acte-sur-la-gouvernance-des-donnees/.

[50] Publications Office. Regulation (eu) 2016/679, 2019. https://eur-lex.europa.eu/eli/reg/2016/679/oj, last accessed on 2021-10-11.

[51] Primal Pappachan, Roberto Yus, Sharad Mehrotra, and Johann-Christoph Freytag. Sieve: A middleware approach to scalable access control for database management systems. *Proc. VLDB Endow.*, 13(12):2424–2437, jul 2020. doi:10.14778/3407790.3407835.

[52] Primal Pappachan, Shufan Zhang, Xi He, and Sharad Mehrotra. Don't be a tattle-tale: Preventing leakages through data dependencies on access control protected data. *Proc. VLDB Endow.*, 15(11):2437–2449, 2022.

[53] ARTICLE 29 DATA PROTECTION WORKING PARTY, 2007. https://iapp.org/media/pdf/resource_center/wp136_concept-of-personal-data_06-2007.pdf.

[54] PIPEDA. Personal information protection and electronic documents act (s.c. 2000, c. 5). https://laws-lois.justice.gc.ca/ENG/ACTS/P-8.6/index.html, last accessed on 2022-02-01.

[55] Postgresql: The world's most advanced open source relational database. https://www.postgresql.org/, last accessed on 2022-05-05.

[56] Postgresql documentation. https://www.postgresql.org/docs/9.3/routine-vacuuming.html#VACUUM-FOR-SPACE-RECOVERY, last accessed on 2022-05-01.

[57] Livio Robaldo, Cesare Bartolini, and Gabriele Lenzini. The dapreco knowledge base: representing the gdpr in legalruleml. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 5688–5697, 2020.

[58] Livio Robaldo, Cesare Bartolini, Monica Palmirani, Arianna Rossi, Michele Martoni, and Gabriele Lenzini. Formalizing gdpr provisions in reified i/o logic: the dapreco knowledge base. *Journal of Logic, Language and Information*, 29(4):401–449, 2020.

[59] Jukka Ruohonen and Kalle Hjerppe. The gdpr enforcement fines at glance. *Information Systems*, page 101876, 2021.

[60] Subhadeep Sarkar and Manos Athanassoulis. Query language support for timely data deletion. In *EDBT*, pages 2–429, 2022.

[61] Subhadeep Sarkar, Jean-Pierre Banâtre, Louis Rilling, and Christine Morin. Towards Enforcement of the EU GDPR: Enabling Data Erasure. In *iThings 2018 - 11th IEEE International Conference of Internet of Things*, pages 1–8, Halifax, Canada, July 2018. URL: https://hal.inria.fr/hal-01824058.

[62] Subhadeep Sarkar, Tarikul Islam Papon, Dimitris Staratzis, and Manos Athanassoulis. Lethe: A tunable delete-aware lsm engine. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, page 893–908, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3318464.3389757.

[63] Adam Satariano. Meta fined $1.3 billion for violating e.u. data privacy rules, 2023. https://www.nytimes.com/2023/05/22/business/meta-facebook-eu-privacy-fine.html, last accessed on 2023-06-20.

[64] Malte Schwarzkopf, Eddie Kohler, M Frans Kaashoek, and Robert Morris. Position: Gdpr compliance by construction. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, pages 39–53. Springer, 2019.

[65] Nick Scope, Alexander Rasin, Ben Lenard, Karen Heart, and James Wagner. Harmonizing privacy regarding data retention and purging. In *Proceedings of the 34th International Conference on Scientific and Statistical Database Management*, SSDBM '22, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3538712.3538718.

[66] Nick Scope, Alexander Rasin, Ben Lenard, James Wagner, and Karen Heart. Purging compliance from database backups by encryption. *J. Data Intell.*, 3(1):149–168, 2022.

[67] Secure Hash Standard (SHS) . https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf, 2015.

[68] Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar, and Vijay Chidambaram. Understanding and benchmarking the impact of gdpr on database systems. *arXiv preprint arXiv:1910.00728*, 2019.

[69] Supreeth Shastri, Vinay Banakar, Melissa Wasserman, Arun Kumar, and Vijay Chidambaram. Understanding and benchmarking the impact of gdpr on database systems. *Proceedings of the VLDB Endowment*, 13(7), 2019.

[70] Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram. The seven sins of personal-data processing systems under {GDPR}. In *11th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 19)*, 2019.

[71] Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram. Gdpr anti-patterns. *Communications of the ACM*, 64(2):59–65, 2021.

[72] Yan Shvartzshnaider, Zvonimir Pavlinovic, Ananth Balashankar, Thomas Wies, Lakshminarayanan Subramanian, Helen Nissenbaum, and Prateek Mittal. Vaccine: Using contextual integrity for data leakage detection. In *The World Wide Web Conference*, WWW '19, page 1702–1712, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3308558.3313655.

[73] Alain Tchana, Raphael Colin, Vincent Berger, Benoit Combemale, Natacha Crooks, and Ludovic Pailler. rgpdos: Gdpr enforcement by the operating system. *arXiv preprint arXiv:2205.10929*, 2022.

[74] IT Governance Privacy Team. *Eu general data protection regulation (gdpr)–an implementation and compliance guide*. IT Governance Ltd, 2020.

[75] TikTok and US Congress Proceedings . https://www.nytimes.com/live/2023/03/23/technology/tiktok-hearing-congress, 2023.

[76] TikTok and User Data Deletion . https://techcrunch.com/2023/03/23/in-congressional-hearing-tiktok-commits-to-deleting-u-s-user-data-from-its-servers-this-year/, 2023.

[77] Casey Tran, Reza Tourani, Satyajayant Misra, Travis Machacek, and Gaurav Panwar. Analyzing gdpr compliance of named data networking. In *Proceedings of the 8th ACM Conference on Information-Centric Networking*, ICN '21, page 107–117, New York, NY, USA, 2021. Association for Computing Machinery.

[78] VDPA. Sb 1392 consumer data protection act (Virginia). https://lis.virginia.gov/cgi-bin/legp604.exe?211+sum+SB1392, last accessed on 2022-02-01.

[79] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[80] Chad Woolf. All aws services are gdpr ready, 2018. https://aws.amazon.com/blogs/security/all-aws-services-gdpr-ready/, last accessed on 2021-10-12.