Lessons Learned: Surveying the Practicality of Differential Privacy in the Industry

Gonzalo M. Garrido
TUM, BMW Group
Germany
gonzalo.munilla-garrido@tum.de

Xiaoyuan Liu
UC Berkeley
USA
xiaoyuanliu@berkeley.edu

Florian Matthes
TUM
Germany
matthes@tum.de

Dawn Song
UC Berkeley
USA
dawnsong@berkeley.edu

ABSTRACT

Since its introduction in 2006, differential privacy has emerged as a predominant statistical tool for quantifying data privacy in academic works. Yet despite the plethora of research and open-source utilities that have accompanied its rise, with limited exceptions, differential privacy has failed to achieve widespread adoption in the enterprise domain. Our study aims to shed light on the fundamental causes underlying this academic-industrial utilization gap through detailed interviews of 24 privacy practitioners across 9 major companies. We analyze the results of our survey to provide key findings and suggestions for companies striving to improve privacy protection in their data workflows and highlight the necessary and missing requirements of existing differential privacy tools, with the goal of guiding researchers working towards the broader adoption of differential privacy. Our findings indicate that analysts suffer from lengthy bureaucratic processes for requesting access to sensitive data, yet once granted, only scarcely-enforced privacy policies stand between rogue practitioners and misuse of private information. We thus argue that differential privacy can significantly improve the processes of requesting and conducting data exploration across silos, and conclude that with a few of the improvements suggested herein, the practical use of differential privacy across the enterprise is within striking distance.

KEYWORDS

Privacy-enhancing technology, data sharing, data analytics, platform, SQL, machine learning, case study, interviews, survey

1 INTRODUCTION

Several factors have spurred the development of more advanced privacy-enhancing technologies (PETs) in the past years. On the one hand, from an adversarial perspective, (i) multiple white-hat *attacks* have shown that "traditional" anonymization techniques such as suppressing names are vulnerable to re-identification across industries [8, 29, 33, 67, 79, 94]. Additionally, between 2020 and 2021, (ii) the total cost of *data breaches* have increased by 10% on average [87]. Moreover, (iii) governments have promulgated *data protection laws* in the past years, such as the European General Data Protection Regulation (GDPR) [30] or the California Consumer Privacy Act [80]. In particular, the GDPR has issued fines as high as $887M [54] and $120M [20]. Furthermore, (iv) beyond the *ethical and moral obligations* of companies to protect people's privacy, providing the best privacy protection available could (v) *differentiate and appreciate their brands* [72], (vi) provide *fairer products and services* that avoid price discrimination [35], and (vii) *increase data collection* as PETs help to surmount regulatory barriers fairly [61]. Aiming to materialize these benefits while mitigating the privacy risks, researchers have turned to differential privacy (DP), which, since its inception in 2006 by Dwork et al. [28], has become the golden privacy standard in academia due to its unique privacy guarantees.

However, despite numerous open-source utilities, only a few tech companies [6, 7, 24] and the US Census Bureau [57] have adopted DP. Accordingly, our work addresses the research gap in bringing DP into organizations' workflows and reaching broader adoption. Dwork et al. [27] partly covered the gap by interviewing DP experts, while our study closes the remaining gap by bringing non-experts into the spotlight. Thus, we interviewed 24 practitioners (19 analysts and 5 data stewards) across 9 major companies that have not yet deployed DP. Overall, our main contributions are:

(i) **Survey Results.** We formulated 5 research questions and derived 24 interview questions thereof. The results of the interviews provide an overview of the current state of data access models (§ 5.1), privacy practices (§ 5.2), motivation behind privacy protection (§ 5.3), and analysis workflows (§ 5.4) in the industry.
(ii) **Key Findings**. From the survey results, we extract 11 key findings, suggest improvements, and answer the 5 research questions about the practicality of DP in the industry (§ 6).
(iii) **Functional Requirements**. Based on the key findings, we propose 10 key desiderata to guide organizations in building privacy-enhancing analytics systems that tackle the privacy-related pain points in their workflows (§ 7.1).
(iv) **Missing Building Blocks**. Given the identified key desiderata, we outline 7 gaps in state-of-the-art DP tooling (§ 7.2).

Privacy officers and *legal practitioners* will find (i) and (ii) helpful in understanding the landscape of privacy and analysis workflows in the industry. *Software engineers* and *developers* will also appreciate (iii) and (iv) as these contributions focus on tooling, and, additionally, will find our early-stage privacy-enhancing analytics system design presented in Appendix H helpful. Overall, notable findings reveal that cumbersome data request processes block analysts for significant periods for every new project. Additionally, we note that SQL was more important than machine learning, and data stewards are more concerned with security than privacy. We conclude that DP could shorten data access processes,

Proceedings on Privacy Enhancing Technologies 2023(2), 151–170

https://doi.org/10.56553/popets-2023-0045

enable data exploration across silos, and is applicable to specific use cases. Moreover, DP tool designers can learn from one another as no tool outperforms the rest in every aspect, and, most importantly, bridging the gap between theory and practice is primarily an engineering problem within striking distance.

2 DIFFERENTIAL PRIVACY

Unlike traditional privacy techniques, which are vulnerable to auxiliary information attacks [8, 29, 33, 67, 79, 94], differential privacy [28] mathematically formalizes a privacy guarantee agnostic to background information. A function guarantees differential privacy (e.g., an analytics query or a machine learning (ML) model) if it bounds the information gain that an attacker can expect from its outputs. Aligned with this adversarial perspective, for the context of this study, we define *privacy* as the prevention of an individual's re-identification [108].

In practice, the outputs of a differentially private function are similarly likely, regardless of an individual's contribution to the input data. This similarity is bounded by the parameter ε, which is inversely proportional to the strength of the privacy protection. A randomized function $\mathcal{M}(\cdot)$ satisfies differential privacy by adding calibrated random noise, typically to a deterministic function's output. Formally, differential privacy is defined as [26]:

Definition 1. *(ε-Differential Privacy). A randomized function $\mathcal{M}(\cdot)$ is ε-differentially private iff for any two datasets D and D' differing on at most one element, and any set of possible outputs $\mathcal{S} \subseteq Range(\mathcal{M})$:*

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^{\varepsilon} \times \Pr[\mathcal{M}(D') \in \mathcal{S}].$$

We introduce other concepts useful in the context of this paper:
Sensitivity. Beyond ε, the other parameter that affects the scale of the noise is the sensitivity of the deterministic function, which determines the maximum difference of the function's outputs over all possible neighboring datasets D and D'.
Central/Local Model. An application can add differentially private noise in the *central* model after aggregating data points from different clients or in the *local* model by adding noise to each data point individually. While the local model requires less trust assumptions with the aggregator, it is usually noisier than the central model.
Sequential Composition. Differential privacy algorithms follow *sequential composition* [26], i.e., if one executes a sequence of (possibly different) DP mechanisms n times over D with ε_i, the consumed *privacy budget* $\varepsilon = \sum \varepsilon_i$.
Privacy Budget Tracker. Because the added noise is centered around 0, an attacker could reverse engineer the n outputs by averaging out the noise. Thus, systems should implement privacy budget trackers to prevent this attack.
Floating-Point Vulnerability. Proofs of differential privacy mechanisms work on continuous distributions, which leads to privacy bugs in practice as the implementations rely on floating-point arithmetic [76]. There are a few solutions to this problem. In short, Mironov's Snapping mechanism [76] discards the least-significant bit in a post-processing step, Naoise et. al [51] combine four random samples, and Haney et. al [46] designed a variant of the Laplace mechanism that avoids a precision-based attack.

3 RELATED WORK

Some organizations have developed and deployed differential privacy tooling and have documented their purpose. Specifically, Apple [6, 7], Google [6], and Microsoft [24] employ algorithms based on the local model of differential privacy to collect information from users. The local model is not as predominant in the industry as the global model (our focus), which has seen more deployments in the past years: Google's Plume [4] enables simple statistics (count, mean, sum, variancer, and quantile) over large-scale datasets. Moreover, LinkedIn [63, 88, 89] proposed an API to analyse user data, and the U.S. Census Bureau in 2020 [57] released microdata; however, these two approaches only considered count queries. Additionally, there exist open-source differential privacy *libraries*, *frameworks*, and *systems* from Google [39–42, 107], Harvard [31, 47], IBM [52], Meta [74], OpenMined [83] (experimental product), Tumult Labs [99], and the University of Pennsylvania [78] and Texas [90]. Note that OpenDP encapsulates SmartNoise core [82]. Additionally, researchers have also developed open-source systems focused on *user interfaces* for differentially private analytics: Bittner et. al [12], DPcomp [49], DPP [56], Overlook [97], PSI (Ψ) [32], and ViP [77]. However, only a few libraries have been discussed in a utility benchmark [36]. Moreover, Johnson et al.'s work on differentially private SQL [60] at Uber [59] focused on a quantitative evaluation of the queries without discussing its practicality with practitioners. Unlike the previous literature above, we aim to qualitatively understand the practicality and adaptability of differential privacy in the central model to existing data analysis pipelines within an organization beyond count queries.

Among top searches of surveys related to differential privacy in digital libraries such as IEEE [53], ACM [3], ScienceDirect [92], or ArXiv [19], one may notably find surveys of applications or analysis models for differential privacy in the context of social networks [55], cyber physical systems such as IoT [48], statistical learning [93], location-based services [65], a user survey about privacy in data sharing [15], and lessons learned from employing differential privacy in the US Census [34]. Notably, Kifer et al. [64] distills a set of best practices and implementation details from their experience designing differential privacy systems at Meta, which we consider in our key system desiderata proposal (see section 7.1). However, our work instead explores systems from companies unfamiliar with differential privacy and focuses on answering whether differential privacy could help data analysts in the broader industry. Lastly, the closest work to ours is from Dwork et al. [27]. They interviewed differential privacy experts regarding their implementation specifications. We differentiate from Dwork et al. [27] in that the hereby interviewed practitioners and the organizations as a whole had no significant technical expertise on differential privacy, which are the vast majority in any industry, and, specifically, we sought to understand whether differential privacy could lift the privacy-related roadblocks in their data analysis workflow.

4 RESEARCH METHOD

While a few organizations have successfully deployed differential privacy for data analysis [6, 7, 24, 57, 63], the large majority have not. To understand whether differential privacy in the central model is practical in their analysis workflow, following a method inspired

by Dwork et al. [27], we performed an empirical study of a set of institutions that have not deployed differential privacy yet for their internal analysis workflows in production. Since the focus is learning whether institutions could benefit from differential privacy, the unit of analysis is the institutions themselves.

Our study captures the answers of 24 practitioners from 9 organizations (19 data analysts/engineers and 5 data stewards). These organizations belong to different industries and are of different sizes (see details in Table 2 of Appendix A). The jurisdictions under which the companies operate contextualize our key findings to the EU (5 companies) and the USA (4 companies). In some organizations, we interviewed multiple practitioners to produce a holistic picture of their data analysis ecosystem. Most interviewees held the title of *data analyst*, while a few were data engineers or team leaders. Irrespective of their title, all practitioners had at least two years and at most 10 years of experience in the field (around 5 years on average) and a comprehensive knowledge of their organization's tools and workflows for data analysis.

Interview Format and Research Questions. We interviewed each of the 19 data analysts for approximately one hour through a video conference, except for three via email correspondence, between November 2021 and August 2022. We produced the research questions (RQs) and the questionnaire prior to the interviews and based on the authors' knowledge of differential privacy and feedback from practitioners other than the ones interviewed. The research questions aimed to understand whether differential privacy could enhance their corresponding institutions' analysis workflow by identifying missing opportunities, assessing the impact of differential privacy in their workflow, and identifying roadblocks.

We carefully formulated the questions broadly to enable interviewees to express their views freely, recount their experiences fully, and reduce response bias and priming. Because the organizations have not deployed differential privacy, most interviewees were not familiar with differential privacy; only two had some non-technical familiarity. We tackled this challenge by explaining differential privacy at a high level before starting the questionnaire. We produced the questionnaire for data analysts in Appendix C, whose results are collected in section 5. Only 4 of the 24 questions contained the word "differential privacy", which the interviewees could nonetheless answer without a deeper technical understanding (see Appendix C).

Furthermore, we performed a deep dive in one corporation by interviewing 10 analysts. Additionally, to understand the process and motivation behind this corporation's privacy protection, we interviewed five *data stewards* via video conference or email correspondence with a second questionnaire (see Appendix B). Data stewards control access to and minimize the risk of data interactions, e.g., auditing analysts' purposes before granting them access. Altogether, we distill key findings and answer these 5 RQs:

RQ1: *What is the context of privacy protection in the targeted organization?* The data stewards provided a perspective of their data protection practices, shedding light on their motivation, concerns, and possible improvements of their methods in privacy protection.

RQ2: *Could differential privacy tackle the privacy-related pain points of an analysis workflow in an organization?* The answer draws a picture of the workflow and the improvements analysts would welcome. This holistic picture helps us identify opportunities for differential privacy in organizations' analytics workflows.

RQ3: *When does differential privacy impede an analysis?* Differential privacy is not a silver bullet; thus, we aim to explore the limitations of differential privacy in an organization. Moreover, as the mechanisms to make SQL-like queries fulfill differential privacy are well-understood [40, 59, 60], we investigate whether this type of query is common in analysts' workflows and bring significant benefits in exchange for moderate effort.

RQ4: *How would differential privacy affect the workflow of an analyst?* Analysts are not accustomed to the noisy outputs of differentially private mechanisms. With this RQ, we aim to understand the impact of noise in their analysis and explore their views on different uses of differential privacy.

RQ5: *Can differential privacy be applied to the frequent SQL-like queries analysts execute?* To exclude the impossibility of using differential privacy, we must assess whether analysts can use it in their queries.

5 RESULTS OF THE PRIVACY STUDY

To frame the research questions in the appropriate context, we first depict how the interviewed organizations usually access data and present the state-of-the-art anonymization techniques in the industry. Subsequently, to provide a perspective on the motivation behind privacy protection, we summarize the results of the interviews with the data stewards from the deep-dive organization (RQ1). Finally, we delve into the data analysts' answers to assess whether deploying differential privacy is useful and possible (RQ2-6).

5.1 Data Access Models

This study focuses on practitioners performing data analysis internally, i.e., without publicly releasing the results. The interviewed organizations used one of two models for accessing data internally: *segregated* and *federated* [13]. Fig. 1 provides an informal diagram for a quick intuition of the models. These models used distinct roles: *data owners* in charge of collecting data, *data engineers* building pipelines, *data stewards* assigned to overseeing the data access request processes, and *data analysts* fulfilling analytics use cases.

Analytics teams in the segregated model engaged directly with data owners, whose data are stored in different data centers and regions running different systems. The data owners would provide the data and also act as stewards. Without an established system for automated data exchange and preparation, the analytics teams had data engineers to prepare data for every use case. An improvement over the segregated model is its federation. After collection from multiple sources and pre-processing and anonymization, in the federated model, data from all domains (e.g., demographics, financial, health, etc.) are stored and easily accessible from a single application interface. The data engineers build such data pipeline and are not usually part of an analytics team. Data stewards guard multiple data sources, interact with analysts, and are detached from the data owner role, which is dedicated exclusively to data collection.

In both models, *data protection officers* from the legal department could interact in the dataset request process, namely when analysts requested data for the first time or data were highly sensitive.

While the initial monetary investment to build a federated system could be larger than for the segregated model, the federated

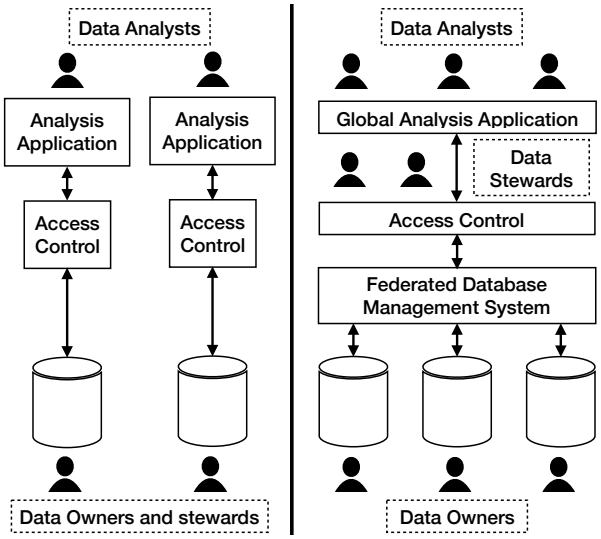

Figure 1: Informal diagram depicting a segregated (left) vs. a federated (right) models for accessing data.

model holds some advantages: it (i) curtails overhead by eliminating the repetition of some tasks in the dataset access request process (e.g., user identification or analysis' purpose specification) and (ii) reduces time-intensive and cumbersome dataset exploration across different systems. Moreover, it (iii) streamlines building data pipelines and defining access request processes by following the same standards across domains and sources. A federated model (iv) simplifies providing precise access control across sources and enforcing policies. Furthermore, it allows to (v) assign non-overlapping roles to practitioners, and (vi) establish re-usable channels between data and analysts. Finally, it could (vii) log the different analyses that other analytics teams have already performed such that others may use them (preventing work duplication). Nonetheless, while the federated model holds such advantages over the segregated model, we observed a similar analyst workflow (see Q7) and an adversarial position for the dataset request process.

5.2 Current Anonymization in the Industry

In this section, we discuss the status quo of the anonymization that companies used to remain compliant without differential privacy, providing a baseline in the context of this work.

Companies can use data collected with user consent exclusively for the agreed *primary* purpose. If companies choose to use data for purposes other than the one agreed (*secondary* purpose), data must be anonymized. All companies had not deployed differential privacy in production or other advanced privacy-enhancing technologies, and employed traditional means of anonymization: *suppressing* direct identifiers such as names, emails, or social security numbers, *truncation* of, e.g., GPS locations and traces, *generalization* (e.g., transforming 28 into [20, 30]), and *dropping* unnecessary attributes and outliers. We consider these techniques *syntactic* [21] because an algorithm transforms the data's syntax following a predetermined model (e.g., GPS locations must only have three decimals). Additionally, data were always encrypted at rest.

Beyond anonymization, to avoid merging multiple sources that could re-identify individuals, some companies did not allow analysts to access multiple datasets at once. In one company, depending on the purpose, stewards granted access solely to a subset of the dataset or a mock dataset for experimenting purposes. Furthermore, for critically sensitive datasets (e.g., illnesses), one company provided access only to an anointed small set of analysts, limited access times, applied anonymization, and restricted analyses to cloud environments. These environments produced logs for later auditing (if needed) and blocked analysts from downloading data. On the other hand, based on user consent for primary use, analysts from one company could access detailed client profiles (names, house prices, mortgages, income, among others). Despite having user consent, we recommend decoupling direct identifiers from the rest of the data (e.g., hashing the direct identifiers) to minimize the consequences of malicious analysts' actions, and encourage the integration of an automated process (or another practitioner) that can only access the analysis output and the direct identifiers to serve the customer (e.g., linked by a hash table only known to the additional process/practitioner).

Altogether, companies applied the principles of *factual anonymity* (i.e., the effort of re-identification is disproportionate to the upside potential of an attacker learning about the individual), *proportionality* (i.e., collection restricted to data necessary to fulfill the primary purpose) [13], *audit logging*, data sharing on a *need-to-know basis*, *data retention* and *purging* [13], *access controls*, and traditional *anonymization*. However, the companies could not measure the privacy achieved by their systems and could only rely on their experience of what is compliant with regulation [30, 80].

5.3 Motivating Privacy

RQ1: *What is the context of privacy protection in the targeted organization?*

(**Q1**) *What is the institution's motivation for privacy protection?* The five stewards agreed on two main motivations: (i) organizations have a legal and moral duty to abide by data-protection laws, (ii) privacy protection is an asset whose "*quality has to be equal to the premium product offered.*"

(**Q2**) *What are your privacy concerns when an analyst has full dataset access?* When proceeding with data protection risk assessments of dataset requests, stewards are predominantly concerned with misappropriation (i.e., unauthorized use of data) and data leakage. While stewards do not expect analysts to be malicious, they are apprehensive of a potential lack of privacy skills, privacy-oriented mindset, and dataset understanding or pure negligence. Specifically, stewards strive to prevent attacks such as unsolicited customer profiling, disclosing data to, or colluding with third parties to take advantage of the customer, combining datasets for re-identification, or using the data for purposes other than the one consented.

(**Q3**) *At what level of data granularity are you protecting and measuring privacy?* The granularity of privacy protection is at the attribute level, and stewards measure privacy based on the fulfillment of data protection regulation. For example, attackers could use the attribute *location* to re-identify individuals; thus, according to GDPR [30], the attribute must be obfuscated so that their home, work, and other points of interest cannot be linked to the individual. Furthermore, the corporation must guarantee the "*security, transparency,*

and legitimacy of the [data] processing." Overall, the anonymization approach strives to achieve the *factual anonymity* principle.

(**Q4**) *What could be improved in the dataset request process?* Data stewards suggested to (i) perform an audit to verify that the executed analysis aligns with the previous commitment, (ii) increase the quality of the datasets' metadata so that analysts can better define a purpose, (iii) increase the privacy training of analysts, (iv) produce privacy-enhanced dataset reports so that after the permission expires analysts can still retain some information, and (v) increment efforts in request process automation.

(**Q5**) *What are your typical questions for the current interview-based full dataset access authorization?* To help other practitioners in the development of their risk assessment process, we gathered the most frequently asked questions from data stewards to data analysts during the dataset request process (see Appendix E). Notably, without a clearly defined data usage purpose, the data stewards would not grant access to analysts.

(**Q6**) *Instead of the interview process, would you be capable to run a program provided by the analyst such that the analysis is carried out without the analyst ever "seeing" the dataset?* While most considered this an efficient, plausible, and necessary step in the future, the five data stewards did not yet have the required technical training, and their system did not enable the functionality. "*At the moment, it is not possible, but it will be a necessary step in the future, if not already today.*" One steward remarked the importance of this functionality, as in some cases, e.g., requesting data from a branch of the company in another country, is extremely challenging.

5.4 The Practicality of Differential Privacy

RQ2: *Could differential privacy tackle the privacy-related pain points of an analysis workflow in an organization?*

(**Q7**) *What is your workflow to analyze data?* Despite the use of either a segregated or a federated model, the workflow was similar across organizations and employed common practices and tools; the main differences were in *dataset exploration*.

(1) **Business Use Case Demand**. A business unit asked an analytics unit to conduct a study for supporting a business need, or analysts continuously studied data from a specific (customer's) domain.

(2) **Dataset Exploration**. Only the companies using the federated model for accessing data could explore datasets' metadata through a *data portal* without requesting access first (unless the dataset was tagged as critically sensitive), making the identification of the suitable dataset for the business need easier. Analysts would find datasets using keywords in a search bar, and datasets provided descriptions, depicted their schema, and had data owners' contact information (analysts sometimes interviewed them to further understand the suitability of the dataset).

In the deep-dive organization, analysts could additionally perform any SQL aggregation query on the anonymized dataset prior to access (e.g., counts, averages, etc.), which they used for data understanding and quality checking (e.g., number of nulls and duplicates or measuring skewness). However, for privacy reasons, analysts could only retrieve a few rows when executing `SELECT *` query types and aggregations could time out (preventing excessive execution costs). Analysts used this *preview* functionality frequently "[...] *to get a feeling for the data*" and found it useful for exploration "*The preview query is the best feature.*" Companies without a federated model could not explore datasets, required data engineers for each use case, and analysts relied either on leveraging their contact network or on an experienced team lead to find promising datasets within the company.

(3) **Dataset Access Request**. Once the analysts identified a promising dataset, they formally requested access, which involved filling standard forms about the details and purpose of the analysis so that data stewards could assess the privacy risks. Except for three small companies, the request entailed interviewing with stewards, where they asked questions such as the ones in Appendix E.

(4) **Visual Inspection and Preparation**. With full dataset access, analysts would sometimes visually inspect the data values, types and schema. Analysts deemed these checks necessary because of the flaws sometimes found in the pipelines and dataset descriptions of the federated data portal or the data provided by the data owners in the segregated model. Moreover, as datasets consisted of many tables, analysts often checked which joins were possible and which attributes were most suitable for primary and foreign keys. With this information, they performed retrieval SQL queries with GROUP BY, WHERE, and `JOIN` clauses to build a sub-dataset fine-tuned for their analysis. Many analysts also performed quality (double) checks and data wrangling using the Python's Pandas library [84] instead of SQL.

(5) **Data Analysis**. Once analysts had checked the quality and wrangled the data, they primarily performed their analysis or ML model training in Python Jupyter Notebooks [5], and if the analyst dealt with *big data*, they employed PySpark clusters [85].

(6) **Output Interpretation and Model Deployment**. If the use case required building a model for online prediction, the analysts would sometimes load the model into a more performant language like Scala before deployment. However, analysts frequently only needed to report statistics and visualizations, from which the business units drew actionable information.

Most of the platforms and workflows employed AWS analytics tools [9] namely S3 buckets (storage), Glue (data preparation), Athena (SQL querying), Sage Maker (data analyses), and analysts also used Python for visualization (one used R) and two of them complemented their results with Tableau [95]. Additionally, two analysts used Knime [66] for drag-and-drop analysis and visualization, and another two employed SAP data management software tailored to their department's needs.

The small interviewed companies had a few major differences, namely, they used a hybrid between the central (all datasets stored in a single data warehouse) and the segregated model. Because of their small customer pool (managed centrally), they collected data from their customers or purchased user-data products from other companies to analyse or train ML models with more data, which required interaction with a segregated set of external data owners. Furthermore, because of the small size of some companies, they had no need for formal dataset request processes as most employees were aware of the activities of the rest; their overhead was at the time of signing the initial contract with customers, which included data access policies and non-disclosure agreements. They also employed traditional anonymization techniques and only retrieved with SQL data stored in, e.g., Google Cloud [38], if strictly needed

(less data for building the model and testing, and more data for the final training or analysis).

(**Q8**) *Why do you need full dataset access?* The main reasons given for accessing all the records of a dataset instead of, e.g., through solely a query interface, were:

(i) *Obtaining a Holistic Understanding of Data.* All analysts worked uncomfortably if they could not make preliminary statistics or visualizations that encompassed all records "*I need to see the entire dataset to understand the data,*" "*I am not necessarily sure of what I need to look at until I look at it. It is an improvisation, you start with a broad question and then you delve into it.*"
(ii) *Less Effort.* A few analysts could fulfill their analysis with only SQL aggregation queries (e.g., counts and averages) and produced visualizations afterward; however, some found using other tools easier: "*Having access to the entire dataset allows me to use Pandas.*"
(iii) *Cleaning Data.* Given that there could be flaws in previous data preparation steps, analysts tended to (double) check all data for quality.
(iv) *Wrangling Data.* In the federated model, data engineers often built datasets without precisely knowing the purpose of a data analyst; thus, analysts sometimes took an engineering role, creating features for ML models or further tailor the dataset for their analysis by grouping or executing queries with `JOIN` clauses.
(v) *Debugging ML Models.* Analysts frequently needed to debug their ML models when testing and training, as there might be corrupted data points.
(vi) *Visually Inspecting Values.* Some use cases, such as root-cause analysis, required analysts to check specific IDs and attribute values, and at times analysts needed to check whether an output table is feasible or map (truncated) GPS traces to street names for the analysis to be interpreted.

Other analysts, however, did not always require access to all records because their ML model already converged, did not overfit, and provided enough accuracy: "*Since I am normally only doing exploratory work, I usually do not need access to the full dataset to prove that the given problem can be solved.*"

(**Q9**) *How often do you request full dataset access? How long does it usually take?* Among the large companies, the request frequency varied widely between 4 times a month to once every 6 months, with an average between once and twice a month. Likewise, regarding waiting times, the minimum hovered around one to three days, while the maximum was two months, with an average between one to two weeks. If another country hosted the data, the first request could take 9 months. Overall, analysts from the interviewed large organizations were blocked for at least one week for every new requested dataset, which they solicited on average once a month. Specifically, in the deep-dive organization, analysts requested 5073 datasets altogether in 2021 (around 14 requests per day, which increased to 18 as of 2022). Out of all the requests in 2021, stewards rejected around 5.6%, amounting to fruitless weeks of revisions[1]. Moreover, the number of requests was more than double the number of available datasets in the deep-dive organization in 2021 (a sign of significant duplication of work, accruing more costs). On the other hand, three of the small organizations did not have such a formal request process, making them agile.

(**Q10**) *What do you think about the process to request full dataset access in your organization?* While analysts at small and US-based organizations were satisfied with the request process, there was an overall consensus at the EU-based large organizations on the following statement: "*The process to get customer data is slow. It might take from three days to weeks, to months*" and for some, even "*Two to three days is too slow.*" In the worst-case scenario, an analyst could wait weeks for a rejection.

Some analysts thought the interviews with stewards were primarily for building trust, and once built "*I always receive access. I do not see the point of waiting and interviewing every time.*" Furthermore, frequently there were too many practitioners involved, leading to lengthy discussions about which dataset to use and often suffered a dilemma because responsibility entailed accountability in one organization "*If there is more than one data steward responsible, then it seems no one takes full responsibility for the acceptance or rejection of the request.*" On the other hand, there were bottlenecks in the vacation season when only one steward was responsible.

Analysts agreed that accessing data has become better since they moved from a segregated model to a federated model; however, the process was still cumbersome, so much so that some teams incurred into the malpractice of entrusting a single analyst to manage the process. One analyst summarized the inefficiency of the segregated model: "*There is a lot of bureaucracy and everyone is extremely reluctant to grant access to a full dataset. Even for internal problems and non-sensitive data. It is cumbersome to request full dataset access because there is no central point where the dataset access can be requested and no central entity which manages access control and usage control for all datasets. For every instance, the process is a bit different depending on the responsible department, underlying workflow and data pipeline.*" In the segregated setting, the process was lengthier, and an analyst could not explore what others had analyzed or requested, sometimes leading to redundant work.

(**Q11**) *What features do you think are missing in your organization's data analysis workflow?* The most notable proposed improvements were: (i) including rich information regarding dataset metadata (preferably with visualizations) and their access request process, (ii) improve real-time analytics performance, (iii) enabling full analytics in data portals such that an analyst does not need to transfer data to other tools, (iv) limiting access times to improve security, and, from a data engineering perspective, (v) automating sensitive data detection and (vi) improving quality and automated checks to minimize visual inspections.

RQ3: *When does differential privacy impede an analysis?*

(**Q12**) *In which analytics use cases have you been involved?* Most analysts worked on *descriptive* use cases. Some of these use cases focused on reporting conclusions from the past by performing root cause (error), cost down, and warranty costs analysis. Other analysts strove to increase the situational awareness of the company by analyzing location-based time series of users (identify

[1]The daily rejection rate went from 0.8 in 2021 to 0.9 in 2022, potentially indicating updated stricter policies.

points-of-interest or common traces), their behavior when using a product or a service (frequently used features, A/B tests, purchases or component performance), and demographics (user-base analysis or advertisement). Additionally, some analysts focused on alerting internal stakeholders of quality defects in real-time, and another analyst performed correlation analysis to better understand the interplay of different variables in products and services. Most of these use cases required performing aggregate statistics (including visualizations to report to management), namely for situational awareness, while others demanded visually inspecting exact values (namely for error detection or financial data), and one analyst used classification ML models for quality checks.

The minority of interviewed analysts were involved in *predictive* use cases: forecasting product lifetime, labelling spam and inappropriate images, user behavior, the company's profit and loss, claim costs, and predictive maintenance and creating automated underwriting models. While these use cases relied on basic statistics, some used vanilla ML such as linear regression (for underwriting models). Nonetheless, the interviewed analysts agreed that using ML was rare; thus, most analysts relied on aggregation and visualizations, as the business units demanded *quick* and easily *interpretable* results.

(**Q13**) *Is SQL-meaningful for your work? How many SQL-like queries do you make weekly?* Most of the interviewees employed SQL, chiefly during exploration, and they deemed SQL an important part of their workflow "*SQL is amazing, everyone who tells you SQL is going away is wrong*," because they could quickly look into rows and performed preliminary statistics, and, with `JOIN` clauses, prepare a dataset for their use case. The least adept analyst executed 5 weekly queries, while the most assiduous SQL user performed 250, being the average around 50 queries per week.

(**Q14**) *How often do you need machine learning to fulfill your analysis in contrast to using SQL?* Two interviewees always needed ML to fulfill their analysis, while another 4 used ML for some of their use cases. The analysts who were allowed to explore datasets used SQL for exploration, and three used SQL to generate statistics and completely fulfill their analysis (complemented with visualizations), while the rest preferred Python or other tools for analysis. Furthermore, analysts often visualized data to accompany their results with other tools (see Q7) and employed retrieval SQL queries for visual data inspections (e.g., for error analysis) or building tailored datasets for their analysis.

(**Q15**) *What are your most used machine learning models?* The 6 analysts employing ML most often resourced to decision trees and linear regression because they are easy to debug, interpret and visualize the results. These analysts also mentioned the use of random forests or XGboost (preferred), Bayesian approaches, support-vector machines, and, for time series, they used outlier detection techniques for error analysis and autoregressive integrated moving average for forecasting. Analysts avoided neural networks because they are hard to interpret; nonetheless, one practitioner indicated they were working on deploying neural networks in the future for underwriting models. In particular, one analyst employed PyCaret [86] for automated ML workflows, as in the corresponding department "*It is more important to be quick and give a good-enough overview than having well trained precise models*," "*Complex machine learning is often never required.*" Other analysts voiced that such is often the case.

(**Q16**) *If you were to use differential privacy to fulfill your analysis, when and how much accuracy would you be willing to forgo?* The willingness to forgo accuracy depended on the use case, with a spectrum ranging from the need for absolute accuracy for quality, error, or financial analyses, to indifference for accuracy in exploratory use cases (only enough accuracy to prove a solution works). For the rest of the use cases, while the interviewees would need to estimate the minimum accuracy formally, they informally reported on average that an accuracy of around 98% would be sufficient, and none reported below 95%. Some financial analyses could also allow errors in the magnitude of cents of a monetary unit, and one analyst reported the need for at least 99% accuracy for finding suitable primary keys for joins. Additionally, comments such as "*I am scared of introducing noise into the analysis. [...] From all the analyses I do every year, there will be some that will be wrong. [...] How well you are compensated depends on how well you do. [...] Because you are paid to have an opinion, you are not allowed to be wrong," suggest that organizations' incentive systems for data scientists, e.g., bonuses, should change to account for errors due to differential privacy.*

RQ4: *How would differential privacy affect the workflow of an analyst?*

(**Q17**) *How much would the noise affect your analysis?* Depending on how much the noise could affect an analysis, we observed three categories for use cases: (i) suffer adverse effects, (ii) reach a tradeoff, and (iii) robust to noise. The first one relates to analyses reporting error, quality, or financial results, where noise could have catastrophic consequences, e.g., a defective component is installed in a product, or yearly budgets are inflated. Moreover, analysts sometimes dealt with low data quality (notably from sensors) that noise could worsen, e.g., GPS locations may already have a 10m error, making a points-of-interest analysis noisy in itself. Adding noise to the aggregation might produce unusable results.

The second type concerns aggregation and visualization reports, where, given enough data, the noise would not affect the interviewees' analysis workflow (e.g., demographics or product usage studies); however, analysts would prefer working with error bounds to report confidently to management. The third type of noise relates to analysts testing solutions "*Since my work is exploratory and we mostly try to prove that the problem can potentially be solved, noise would not have any negative effects for my analysis.*"

(**Q18**) *Would you find it helpful to execute differentially private SQL queries to explore and fully analyse datasets without the standard permissions?* We theorized that given the plausible deniability guarantees of a differentially private analysis, which can be argued to comply with the identifiability notion in GDPR [30, 50], some uses cases that heavily rely on aggregation might abate or not need the standard dataset request processes. From this perspective, most analysts found differentially private SQL queries helpful, in summary, because "*If having differentially private SQL queries for data exploration implies reduced bureaucracy and easier access, then this would save a lot of time and discussions.*"

Notably, one interviewee saw the potential of differentially private queries for data exploration: companies could expose data

externally through an API, allowing others to understand their data products by conducting preliminary analyses. Another analyst favored integrating differential privacy into, e.g., AWS Athena [9]. On the other hand, few analysts did not see the value of differential privacy because their use cases required, e.g., visual inspections for error detection, or their organizations were already agile in accessing data. Lastly, two analysts voiced a general concern "*[Differential privacy] is a double edge sword. You could get quick [data] access, but then [results are] noisy,*" "*I think I would find it annoying, since it adds an additional step and obfuscates the results*," and it could lead to confusion as analysts usually work with accurate data.

(**Q19**) *Only based on the information extracted from a dataset exploration with differential privacy, could you write a script to fulfill your analysis goal?* A couple of interviewees shared their inability to program their script as they needed to see the data (e.g., error analysis), and the others shared their skepticism by highlighting the problem of low data quality. Even if an analyst developed an intuition for the data through differentially private aggregation queries, programming other statistics, visualizations, or ML models would likely require debugging, which may lead to visual inspections.

(**Q20**) *What are the minimum properties for you as an analyst such that you are confident to write an analysis script without full dataset access?* Assuming enough data quality and a use case that does not require visually inspecting data, the interviewees indicated that for tentatively writing code without dataset access, they needed: good metadata from the dataset, such as attribute descriptions, knowledge about the events that trigger data collection, primary keys, data types (IDs, dates, timestamps, floats, strings), dataset size (number of rows and columns), and attribute distributions to learn about sparsity in the form of histograms or box plots.

(**Q21**) *Would you find it helpful to use a dynamic dashboard that visualizes dataset information with differential privacy?* Since data platforms may not expose sensitive data on a dashboard for exploration, we conceptualized enabling this functionality with differential privacy. All but one interviewee considered such a dashboard helpful for finding a suitable dataset faster and with a better user experience than their available utilities (static and scant summaries or using SQL). Specifically, an interviewee commented that, in general, one should be able to visualize the data and get basic statistics before requesting access, and another analyst would have liked to preview similar information as the "`describe`" method of a Pandas dataframe [84] (count, mean, standard deviation, minimum, quartiles, maximum). Nonetheless, one analyst noted that a dashboard is a nice-to-have because it is only more convenient than SQL. Lastly, another interviewee underlined a problem that may arise when an analyst does not trust the data provided by the visualization, e.g., when the plot seems implausible. The interviewee suggested that a dashboard should enable the analyst to drill down or provide contact information from a data owner to verify correctness.

RQ5: *Can differential privacy enhance the privacy of the frequent SQL-like queries analysts execute?*

(**Q22**) *What are your top SQL-like queries before you have full dataset access?* If analysts could explore datasets, most would usually conduct a metadata analysis with SQL to assess data quality: finding the number of duplicates, outliers, nulls, and not-a-number values and measuring the skewness. Analysts would also explore the dataset for data understanding using `COUNT`, `DISTINCT`, `MAX`, `MIN`, `AVG`, and `VARIANCE` functions with `WHERE` and `GROUP BY` clauses. Analysts were typically interested in frequent values within a column (see details in Appendix D). Furthermore, the deep-dive organization allowed to use `SELECT * LIMIT(X)` for a few `X` rows so that analysts could have a "*feeling*" for the data. On the other hand, fewer analysts performed retrieval queries (limited in output rows) to verify whether an ID was present or two tables could be joined.

(**Q23**) *What are your top SQL-like queries after you have full dataset access?* Analysts who could not explore the dataset prior to having dataset access would execute queries such as those in Q22 first (see Appendix D for details). Afterward, if they did not already retrieve the necessary information from the exploration, they resorted to Python and other visualization tools to fulfill the use case. Some analysts performed additional retrieval SQL queries with `JOIN` and `SELECT *` clauses with different filters to visually inspect data points (e.g., IDs or potential errors), identify cut-offs (e.g., where an attribute data type changes), or fetch the specific data they needed.

(**Q24**) *What is the ratio between aggregation queries and queries to retrieve items?* While the interviewees would need to calculate the percentage formally, they reported informally, on average, that around 30% of their queries were for aggregation, being the lowest 0% and the highest 90%. Another three analysts used SQL for retrieval and Python for aggregation or vice versa.

6 DISCUSSION

In this section, we present selected key findings (KF) distilled from the data stewards' and analysts' answers to the 24 interview questions of section 5, most accompanied by succinct recommendations. Lastly, we answer the research questions proposed in section 4.

6.1 Key Findings

(KF1) *Data stewards seem to be more concerned about security than privacy.*[2] Data misappropriation and leakage retain the most attention (Q2), which is reflected in the established cumbersome dataset request processes that dictate access controls and accountability in the name of building trust with analysts. However, we highlight that privacy lacks such attention, even when some companies still allow their analysts to "*see*" or download sensitive data. Attacks on privacy (Q2) could enable malicious analysts to misuse data for spying on or leaking secret information of celebrities, acquaintances, "friends", or relatives [100], blackmailing, or discriminating individuals in social or commercial transactions online [35]. Despite the risks, companies predominantly use traditional and potentially vulnerable anonymization techniques (e.g., pseudo-anonymization or k-anonymity), as demonstrated by the research community [8, 29, 33, 67, 79, 94]. Thus, we suggest companies increase efforts to research and deploy more advanced PETs.
(KF2) *Running analysts' scripts without "seeing" the data is a distant reality for the interviewed companies.* We explored multiple ways for analysts to run scripts without direct dataset access. In

[2] In the context of this work, we refer to *security* as the measures for blocking *unauthorized* data access, while *privacy* focuses on limiting harm by *authorized* analysts [13].

Q6, stewards declared their technical inability to execute scripts that analysts could share and, thus, avoid granting them access (saving time). With the proper tooling, a non-technical steward could potentially run the script; however, current systems do not offer such abstracted functionality and this option would relay the responsibility to the stewards instead. An alternative to transferring the trust to stewards consists on executing analytics in trusted execution environments[3]. Additionally, analysts altogether gave six reasons why they needed full dataset access (Q8) and reported skepticism when asked about writing a script (beyond aggregation) based on a differentially private exploration (Q19 and Q20).

The main impediment reported was data quality, which often led to visual inspections of dataset values. As encouraged by point Q11-(v), we suggest companies prioritize increasing data quality as it will indirectly improve privacy and increase the technical training of data stewards and owners, enabling more data security options.

(KF3) *Given the analysis workflow, differential privacy could have a significant impact on dataset exploration* (see Q7). As long as exploration does not require visually inspecting a particular ID or an exact attribute value, differential privacy can provide noisy statistics for the analyst to familiarize with the data (e.g., number of rows, averages, quantiles, etc.), which is often enough to assess the dataset's suitability. Furthermore, while analysts were not allowed to explore critically sensitive datasets with SQL, employing differential privacy could arguably enable their exploration by adding an extra layer of protection. Additionally, platforms could provide privacy-enhanced dataset previews (e.g., only revealing a few rows or producing dummy or synthetic data with or without differential privacy). Overall, differential privacy could facilitate exploration that otherwise might not be possible or timely.

(KF4) *Analysts could employ differentially private mechanisms to fulfill certain use cases* (see Q7). If the analysis requires summary statistics and visualizations, a differentially private analysis could fulfill the privacy-utility tradeoff given enough data. Consequently, analysts could fulfill use cases without exact outputs, avoiding potential privacy leaks. Regarding ML, while its differentially private implementations are at an early stage, researchers and practitioners could explore systems to assess whether a model shows signs of converging with enough accuracy after training on a sample of the target dataset. Such a system could help analysts to determine the validity of the model or the dataset. Lastly, we suggest exploring whether differential privacy can enable more accurate analyses than the current organizations' anonymization processes.

(KF5) *After fulfilling the use cases, the interviewed companies do not have a human-supported privacy auditing step.* The last reported step of the workflow in Q7 was "*output interpretation and model deployment*". Aligned with a steward in Q4: "*perform an audit to verify* [*alignment with analysis commitment*]," we suggest privacy officers in companies add a randomly-sampled auditing step with a human in the loop after the conclusion of the use case. We also suggest audit logs, which one of the interviewed companies produced for every execution on sensitive data in secured machines, where analysts could not download data or install new software.

[3]Hardware and software designed to run applications securely against unsolicited retrieval of sensitive information or key material [81].

(KF6) *Given the six reasons analysts shared for fully accessing datasets, differentially private mechanisms could help in (i) "obtaining a holistic understanding of data" by providing dataset summary statistics* (see Q8). Additionally, we suggest substituting tedious SQL analyses with dashboards for visualization, so that tasks require (ii) "*less effort*". We also suggest engineers develop and integrate tools that enable analysts to (iii) "*clean*" and (iv) "*wrangle data*" without visually inspecting the values (i.e., no complete data access required). With such tools, filtering values, imputing, removing duplicates and outliers, fixing wrong formattings, handling missing data, or creating new attributes would also help with (v) "*debugging ML*." Moreover, aligning data engineers with analysts could improve data quality, e.g., by involving engineers in the conversations between stewards and analysts. Lastly, researchers could investigate how differentially private set union mechanisms [22, 45] could help analysts to (vi) "*visually inspect values*." Meanwhile, we suggest increased security measures for such cases.

(KF7) *Analysts are frequently blocked for significant periods every time they request access to datasets* (see Q9). There are a few consequences of such delays. Data stewards and privacy officers must also invest their time in reviewing the requests. From our conversations with the interviewees, we also learned that long waiting times could hamper analysts' bursts of creativity and productivity, which indirectly negatively affect the quality of work. Additionally, an interviewee recounted the malpractice of deferring all the dataset request process responsibility to a single analyst in the team (see Q10). Such practice overburdens an individual with the responsibilities of the entire team for, e.g., a data leakage, creating an unhealthy imbalance in accountability. This practice further increases the company's privacy risk by potentially having the other analysts handle data without privacy training. We suspect this malpractice is a sign of over-complicated dataset request processes and long waiting times; thus, we suggest privacy officers streamline their processes and prompt teams to refrain from overburdening a single analyst.

Differential privacy's stronger guarantee could reduce the complexity of the interactions between practitioners by offloading their data protection demands and, thus, reduce the costs accrued by these human-intensive processes. Lastly, given that there were multiple requests for the same datasets from different teams, we encourage companies to build interfaces depicting privacy-enhanced summaries of past fulfilled use cases per dataset. An example is the repository designed by Johnson et al. [58] in the health industry.

(KF8) *Differential privacy could arguably reduce the time to access data.* As differential privacy brings a higher and formal guarantee of privacy, it could relax the inquisitiveness of data stewards, eliminate (steps of) the request process, and enable exploration that was otherwise not possible. By enabling exploration, analysts reduce the likelihood of investing time in request processes that could even result in accessing a non-suitable dataset. With exploration and higher privacy guarantees, differential privacy could also speed up requesting data from other countries, which seemed the most significant bottleneck (see Q9). Additionally, differential privacy could potentially prolong access times (if these are limited) and shorten development cycles with an earlier data access by testing algorithms and applications with noisy data or outputs. Regarding

applications specifically, once finished, the customers can confirm whether the product works appropriately with real data.

We have also observed that, once analysts have access, much of the data protection and accountability lies on their shoulders, which differential privacy could lift to a degree by protecting beyond trust and policy. However, the analyst somewhat familiar with differential privacy pointed out that, unless data quality is improved (as also suggested in Q11), "*There is a still a ways to go to deploy differential privacy,*" because the need to debug by visually inspecting data will prevail. To increase privacy protection in those cases, we suggest using differential privacy with limited visual inspection.

(KF9) *Most analysts employed aggregations and visualizations to fulfill their use case in a timely manner, while machine learning was not as predominant* (see Q12). We found that analysts could employ differential privacy to explore datasets suitable for all the identified use cases. However, for the analysis itself, the interviewees voiced that the noise would invalidate the use cases related to quality, error, and (some) financial analyses because mistakes in safety decisions and financial planning are company critical. Nonetheless, for the use cases that required aggregation and visualization, with enough data, we suggest analysts fulfill these use cases with differentially private queries such as counts, averages, and percentiles, among others (e.g., user behavior, demographics, and some location-based analyses). However, the available tools for differentially private ML are not mature for widespread adoption. Thus, we encourage researchers and practitioners to improve and build systems around existing proposals in future work, e.g., location-date analysis [105], heavy hitter identification [71], mining frequent itemsets [114], deep and supervised learning, random forests, and linear regression, among others [1, 52, 70, 111, 113].

(KF10) *For the interviewed companies, SQL was more important than machine learning and was considered a meaningful tool frequently employed in their workflow* (see Q13 and Q14). Additionally, on average, 30% of the top SQL queries executed before and after full dataset access were for aggregation (see Q22, Q23, and Q24), which researchers have already adapted to fulfill differential privacy [40, 59, 60]. Therefore, there is still a gap between what researchers have enabled and what practitioners need for enhancing the privacy of their frequently used SQL queries—a gap we intend to partly cover in section 7 by proposing 10 key system desiderata that an integrable privacy-enhancing analysis system should fulfill. Beyond SQL, differential privacy and its available tools are also suitable even when analysts preferred using Python for aggregation and ML use cases that allowed for lower precision. In particular, we encourage using Python libraries such as IBM's diffprivlib [36, 52] that provide many off-the-shelf differentially private ML models that could provide enough precision for the intended purpose, such as for the linear regression model one company used for underwriting (see Q15). However, practitioners will require further engineering to limit Python to strictly privacy-enhancing libraries and amenable standard functionalities (e.g. by using policy enforcement paradigms such as Wang et al.'s Data Capsule [102]).

(KF11) *Analysts confirm that differential privacy would be helpful for dataset exploration, fulfill certain use cases, and for enabling privacy-enhancing dashboards for dataset visualization* (see Q18 and Q21). For aggregation-based use cases where noise has no detrimental effects, analysts informally reported, on average, a required accuracy of 98% (see Q16 and Q17). While such a figure might seem high, given the large amount of data handled, analysts could potentially find enough for aggregations that fulfill their privacy/utility tradeoff. For example, as of early 2022, the deep-dive organization had roughly 2260 datasets in its federated system amounting to 3.4PB (1.5TB per dataset on average) with an average daily query execution of over 900TB. However, size might not be enough for some use cases, as the analysis could be sensitive to outliers or corrupted data. Lastly, we observe that it is critical for analysts to know whether the accuracy is above their required level, which would consume privacy budget and be hard to estimate, e.g., when the analysis needs post-processing (clamping or truncation).

6.2 Answers to the Research Questions

RQ1: *What is the context of privacy protection in the targeted organization?* The deep-dive organization invests more resources to security than privacy-enhancing analysis—pattern also present in the other organizations. Moreover, stewards consider privacy an asset and strive to provide the best standard for their customers. However, organizations still employ traditional anonymization techniques. Furthermore, companies today are unable to tangibly measure the privacy of their process (see Q3), and while there are specific privacy and security measures such as access controls, they are hard to quantify formally. Lastly, we observed that the interviewed companies are far from having "*data at their fingertips*", one of the reasons being the onerous dataset request processes, which confuse access hardship with access protection.

RQ2: *Could differential privacy tackle the privacy-related pain points of an analysis workflow in an organization?* Yes, to a large extent—In essence, the main problems are (i) lengthy and cumbersome dataset request processes. Moreover, given that analysts can sometimes "*see*", download, and share the data once they are granted access, and even collude with other co-workers with access to linkable datasets, (ii) *only* policy protects data once stewards grant access. Based on our work, we argue that differential privacy can reduce time-to-data by enabling exploration of critically sensitive data or across third-party data sources, relax the current data access restrictions thanks to its formal privacy guarantee, is applicable to some aggregation-based use cases, and, for some use cases, engineers should consider building solutions that block analysts from "*seeing*" the data.

RQ3: *When does differential privacy impede an analysis?* The answer to this RQ heavily depends on the use case and whether the analysts are willing to forgo accuracy. On the one hand, noise addition-based differential privacy is useful in aggregations performed by the interviewees (e.g., querying demographics or frequently used product features). Moreover, on average, interviewees were comfortable with 98% accuracy. However, differential privacy is not a silver bullet, as some of the interviewees' use cases cannot rely on it (e.g., error analyses or critical financial estimations). Therefore, we suggest building systems that enable differential privacy while maintaining the flexibility of allowing non-differentially private queries when the use case strictly needs them.

RQ4: *How would differential privacy affect the workflow of an analyst?* If differential privacy enabled previously unavailable exploration and provided data for privacy-enhanced dashboards, analysts

would have a better user experience in their workflow and lower time spent on processes and exploration, but would also need to accustom to working with noisy data.

RQ5: *Can differential privacy be applied to the frequent SQL-like queries analysts execute?* Yes—While not as frequent as retrievals, around a third were aggregations amenable to differential privacy.

7 TOWARDS PRACTICAL DIFFERENTIAL PRIVACY

This section provides a set of critical system desiderata a differential privacy (DP) analytics system should satisfy for practical deployments. Subsequently, we identify requirements fulfilled by state-of-the-art tools (see Table 1) and highlight the gaps in practice.

7.1 Key System Desiderata

In secondary use cases, an alternative to *syntactic* anonymization (see section 5.2) for sharing data is an inherently private analysis, i.e., the analysis satisfies a *semantic* privacy definition such as DP [21], which uniquely provides a measure of privacy (ε). With DP, organizations do not necessarily need to use potentially vulnerable syntactic techniques (e.g., rounding or truncation) because the analysis itself already enhances individuals' privacy. Based on the (i) interviewees' description of their analytics workflows and systems, (ii) the authors' knowledge in the domain of privacy, and (iii) the feedback provided by additional privacy practitioners and researchers who work closely with/in our lab, we propose 10 key desiderata. The desiderata correspond to a system that enables differentially-private analyses in the central model and focuses on dataset exploration and fulfilling use cases requiring aggregations (see use cases in Q12). These use cases often rely on SQL-like queries such as counts, averages, etc. Additionally, we inspired some of the characteristics of the key desiderata related to (III) *Security* and (V) *Visualization* from Kifer. et. al [64] and Nanayakkara. et. al [77].

(I) *Differentially Private Analytics.* The system bestows DP to a learning function (e.g., a query or an ML algorithm) by adding calibrated noise to the deterministic outputs (or by other means). The system supports the (i) aggregation queries: `COUNT`, `MAX`, `MIN`, `AVG`, `VAR`, and `SUM`, (ii) provides a complementary ML feature, and stores executed queries for future retrieval. The queries (iii) allow for `WHERE`, `GROUP BY`, and `JOIN` clauses.

(II) *Usability.* The system provides logic to preserve (i) the semantic consistency of queries (e.g., variance > 0) and across overlapping domains (e.g., the sum of noisy element counts is not larger than the noisy total). Moreover, the system presents the option to (ii) estimate the sensitivity of a query without user input, and (iii) recommends or sets privacy parameters automatically depending on the dataset and query.

(III) *Security.* The system (i) provides a stochastic tester or other functions to automatically verify whether the algorithm fulfills DP, (ii) employs cryptographically secure pseudo-random number generation with careful seed management, (iii) generates noise impervious to floating-point vulnerabilities [46, 76]. Furthermore, the system (iv) blocks the user from "*seeing*" the data, i.e., while analysts can execute queries, they cannot download or visually inspect the dataset, (v) does not allow to execute arbitrary code, (vi) executes heuristic optimizers only at post-processing, and (vii) protects against timing attacks [64]. A libraries' and frameworks' scope limits to fulfilling (i), (ii), and (iii).

(IV) *Synthetic Data Generation.* When the goal is to develop an application or explore whether an ML model is suitable for a task, the system produces synthetic data. After testing, the analyst can proceed with the real data (without "*seeing*" it). Synthetic data generation could rely on simple techniques (e.g., sampling from a normal distribution with the same mean and standard deviation as the target attribute), ML [18, 96, 112], or combining DP with either. If the analyst is only interested in the data schema, the system produces dummy data, preserving only the schema and data types.

(V) *Visualization.* The system presents a dashboard depicting interactive plots (e.g., histograms) relying on DP queries for quick and intuitive (i) dataset exploration. Additionally, the dashboard visualizes an analysis' expected (ii) accuracy and (iii) disclosure risk, (iv) uncertainty (i.e., a measure of how the same mechanism can produce different outputs with the same input arguments), (v) statistical inference (i.e., privacy parameter estimation with confidence intervals), and (vi) budget splitting (i.e., help in splitting the privacy budget across queries) [77].

(VI) *Privacy Budget.* The system (i) tracks the budget spent (ε "odometer"), (ii) blocks further queries if analysts exhaust their budget, and (iii) accommodates the budget for growing datasets. (iv) It should enable data stewards to specify budgets for teams, individual analysts, and use cases depending on the data's sensitivity.

(VII) *Accuracy Adjustment.* The system allows the user to propose a desired accuracy level. Alternatively, after the query execution, the system provides either information about the noise scales (without additional budget) or a confidence interval (spending budget) [101].

(VIII) *Query Sensitivity.* The system enables a practitioner, e.g., a data analyst or steward, to input the attributes' bounds (maximum and minimum values) as function parameters or in the dataset schema so that the system calculates the query's sensitivity.

(IX) *Privacy-Sensitive Data Annotation.* The system enables data stewards to allowlist attributes based on teams, roles, and use cases. The system automatically obfuscates attributes outside the allowlist.

(X) *Authentication and Access Controls.* The system easily integrates with existing authentication and access control services and enables data stewards to define their access policies.

7.2 Gaps in Differential Privacy Practice

Despite available open-source tooling, one company found it hard to find external partners that could bring DP into practice in their internal analysis workflow. Furthermore, another company stated after exploring the use of DP that, while it seemed helpful, "*[Deploying differential privacy] was more expensive than doing nothing.*" Instead, the department decided to upload syntactically anonymized data to a highly secured system, with limitations on access time, downloads, number of analysts, and audit logs. We kindly argue that their over-statement was due to the intangible costs of the dataset request processes and the lack of integrability of current DP tooling, which makes deployment a complex endeavor.

Overall, our findings indicate a gap between the theory and practice of DP. Working towards bridging the gap, we qualitatively mapped in Table 1 our key system desiderata with DP tools to highlight areas of future work for the privacy community. We selected

Table 1: Mapping between open-source tools and user interfaces and the key system desiderata. Legend: ✓= functionality fully available; ✗= limited functionality or not available; N/A = not applicable; P. = Privacy; DP = Differential P.; TF = TensorFlow; I.i = Enables aggregation queries; I.ii = Enables machine learning; I.iii = Enables query clauses (e.g., `JOIN`); II.i = Query semantic consistency; II.ii = DP sensitivity calculation; II.iii = Privacy parameter search; III.i = DP correctness verification; III.ii = Cryptographically secure pseudo-random number generation; III.iii = Protection against floating-point vulnerability; III.iv = Block data visibility; III.v = Block arbitrary code; VI.i = Budget accountant; VI.ii = Query blocker.

Table 1A: Libraries, frameworks, and systems for differential privacy analytics.

Tool/Desiderata	(I) DP Analytics	(II) Usability	(III) Security	(IV) Synthetic Data	(V) Visuals	(VI) Privacy Budget	(VII) Accuracy Adjustment	(VIII) Query Sensitivity	(IX) Data Annotation	(X) Access Controls
Libraries†										
diffprivlib [52]	I.i, ii ✓	II.i ✓	III.ii, iii ✓	✗	N/A	VI.i ✓	✗	✓	N/A	N/A
Google DP [41]	I.i ✓	II.ii ✓	✓	✗	N/A	VI.i ✓	✗	✓	N/A	N/A
Opacus [74]	I.ii ✓	✗	III.ii ✓	✗	N/A	VI.i ✓	✗	✓	N/A	N/A
OpenDP [47]	I.i ✓	II.iii ✓	III.ii, iii ✓	✗	N/A	VI.i, ii ✓	✓	✓	N/A	N/A
TF Privacy [39]	I.ii ✓	✗	✗	✗	N/A	VI.i ✓	✗	✓	N/A	N/A
Frameworks†										
Chorus [60]	I.i, iii ✓	✗	✗	✗	N/A	VI.i ✓	✗	✓	✓	N/A
PipelineDP [83]	I.i ✓	✗	III.ii, iii ✓	✗	N/A	VI.i ✓	✗	✓	✗	N/A
P. on Beam [42]	I.i ✓	II.ii ✓	✓	✗	N/A	VI.i ✓	✗	✓	✗	N/A
Tumult Analy.[99]	I.i, iii ✓	✗	✓	✗	N/A	VI.i, ii ✓	✗	✓	N/A	N/A
ZetaSQL [40]	I.i, iii ✓	II.ii ✓	✓	✗	N/A	✗	✗	✓	✗	N/A
Systems										
Airavat [90]	I.i, ii ✓	✗	III.iv ✓	✗	✗	VI.i, ii ✓	✗	✓	✗	✓
DJoin [78]	I.i, iii ✓	✗	III.ii, iv, v ✓	✗	✗	VI.i, ii ✓	✓	✓	✗	✗

†Libraries' and frameworks' (III) Security scope is limited to three sub-desiderata (i), (ii), and (iii).

Table 1B: User interfaces for differential privacy analytics (cf. adapted [77]).

User Interface/Desiderata	(V.i) Dataset Exploration	(V.ii) Accuracy Visualization	(V.iii) Risk Visualization	(V.iv) Uncertainty Visualization	(V.v) Statistical Inference	(V.vi) Budget Splitting
Bittner et. al [12]	✗	✓	✗	✗	✗	✗
DPcomp [49]	✓	✓	✗	✗	✗	✗
DPP [56]	✗	✓	✓	✗	✗	✗
Overlook [97]	✓	✓	✗	✓	✗	✗
PSI (Ψ) [32]	✓	✗	✗	✗	✗	✓
ViP [77]	✓	✓	✓	✓	✓	✓

the tools from the related work in section 3 that offer open-source implementations for the central model of DP (see tool descriptions in Appendix F). We must highlight that some of these tools are *libraries* (provide specific functions) or *frameworks* (abstractions used to build specific applications) and, thus, lack functionalities that a *system* (end-to-end application) like Airavat [90] could provide, such as (III.iv) *Blocking the visibility of data* or (X) *Authentication and access controls*. Note that libraries and frameworks assume analysts have data access. Additionally, we regard *user interfaces* (systems focused on visualizations and providing analytics metadata) as a set of tools that should fulfill key desiderata specific to (V) *Visualization*. Accordingly, we assign each open-source software to its category in Tables 1A and B for an appropriate comparison.

We must highlight that the mapping of Table 1 provides high-level guidance, as there are (out-of-scope) nuances Table 1 does not capture. For example, user interfaces such as Bittner et. al [12] and DPP [56] in Table 1B provide exploratory results for using DP ML and for disclosure risk, respectively; however, they do not help understanding the dataset, which is a critical requirement for data analysts. Regarding the tools in Table 1A, diffprivlib [52] offers multiple ML models (PCA, Naive Bayes, liner and logistic regression, k-means) while others focus on deep learning (Opacus [74] and TensorFlow (TF) Privacy [39]) or MapReduce functionality (Airavat). Additionally, the frameworks are designed for large-scale datasets. We note that Google DP [41] provides the building blocks for ZetaSQL [40] and Privacy on Beam [42] (and PipelineDP [83]), which add more functionality for considering datasets with multiple individual's contributions. Lastly, most tools provide only an "odometer" for privacy budgeting, while a few block new queries if the budget is spent (e.g., OpenDP [47] and Tumult Analytics [99]), and Google DP offers functionality to distribute budget across different DP mechanisms [14, 23]. None, however, account for growing datasets, which is a challenge recently tackled in [68]. One may find more of these nuances in [36].

Based on the non-availability or limited implementations of some desiderata in Table 1, we conclude that *differential privacy tool designers can learn from one another, no tool outperforms the rest in every aspect*, and, most importantly, that *bridging the gap is primarily an engineering problem*. Subsequently, we identify the major gaps in differential privacy practice:

Gap 1: (II) *Usability*. While semantic consistency is sometimes desirable for analysts, it can also introduce more error/bias in some scenarios. Only diffprivlib implements mechanisms to fulfill DP and consistency for specific queries (e.g., variance > 0), whereas Google DP or Tumult Analytics only truncate values in post-processing. Furthermore, only Google DP can calculate the query sensitivity in a privacy-enhancing manner without any user input, which is necessary when an analyst lacks domain knowledge of the application (i.e., input bounds). Thus, none of the tools in Table 1 completely fulfill the usability desiderata. *Guidance*: [2, 41, 52, 88, 89, 103, 104]
Gap 2: (III) *Security*. The tools do not provide many security features individually. E.g., most lack stochastic testers to verify that an analysis fulfills DP, and none implement protections against time-attacks [64]. Wrt to secure random number generation: TF Privacy inherits TF's insecure RNG [43, 44] and Airavat employs the insecure utility `java.util.Random` [91] in contrast to DJoin, which relies on FairplayMP [10, 11]. Moreover, while TF Privacy developers are aware [73], we encourage them to include floating-point protections in their deep learning models or rely on discrete noise distributions [16, 37]. Moreover, most tools should tackle their precision-based attack vulnerability [46]. Lastly, we highlight some of the good practices Kifer et. al [64] proposed: open-sourcing systems (the community can check for vulnerabilities) and performing code audits and unit tests to ensure correctness in DP, privacy accounting, and noise sampling. *Guidance*: [4, 41, 64]
Gap 3: (IV) *Synthetic Data Generation* (SDG). Similarly to tools offering DP ML [39, 52, 74, 90], we suggest developers package and include DP SDG logic. *Guidance*: [17, 18, 96, 98, 106, 109, 110, 112].
Gap 4: (V) *Visualization*. While there is enough research on user interfaces, the most popular frameworks and libraries do not adopt them. We suggest packaging available DP user interfaces for patching analytics tools. *Guidance*: [77, 97].
Gap 5: (VI) *Privacy Budget*. A surprisingly high number of tools implement privacy "odometers" without a logic to block queries after exceeding the budget. *Guidance*: [42, 78, 90].
Gap 6: (VII) *Accuracy Adjustment*. While most user interfaces provide some form of accuracy calculation and visualization, many other tools overlook such feature. *Guidance*: [77, 101].
Gap 7: (IX & X) *Functionality for Data Stewards*. Only a few tools enable data stewards and owners to (IX) annotate sensitive data and (X) define and enforce access controls. Developers do not need to reinvent the wheel, as they adopt current best practices from popular cloud platforms [9, 38, 75]. *Guidance*: [56, 60, 90].

Given that most functional requirements are fulfilled in components across tools, we conclude that *engineering efforts are within striking distance*. To complement these building blocks, we offer an early stage, high-level system design blueprint in Appendix H. The blueprint aims to spark interest in practitioners to develop holistic analytics tooling that follows the identified key system desiderata.

8 FURTHER CHALLENGES

Beyond the engineering and organizational challenges discussed in the previous sections, there exist other critical technical challenges in DP. In combination: Managing privacy budgets on large-scale user data streams [68] with unknown domains and user contributions on multiple records [4] across different systems while adapting the noise level as the budget diminishes. Furthermore, fitting a mathematical model to such a system's semantics and verifying DP fulfillment with, e.g., unit tests, poses additional difficulties [64]. Additionally, DP might not be *fair* [62] in some use cases where a DP calculation determines a critical outcome, e.g., a user's financial support in an underwriting model (see challenges in Appendix G).

Our work highlights the challenges blocking the broader adoption of DP in organizations' workflows. Dwork et al. [27] partly studied these challenges by interviewing DP experts, while our study brings non-experts into the discussion. Dwork et. al distilled four main challenges from their interviews (section 3.6 [27]), which overlap with a few of our findings: (i) *Part of the challenges deploying DP were design based*. In section 7, we highlight that current DP tools still require engineering effort to be easily deployable in organizations. (ii) *DP deployment complexity is also institutionally based*. A common theme of the interviewed companies was their intricate networks of stakeholders and processes, which hamper goal alignment and technology deployments. (iii) *There was no consistency in DP approaches across institutions, indicating a need for shared learning*. One of our conclusions in section 7 signals that tool designers can learn from one another. (iv) *Transparency and testable privacy statements can benefit companies in the regulatory landscape*. Similarly, section 7 advocates for transparency in system designs and moving towards DP-centered systems and away from syntactic privacy definitions that only guarantee *factual anonymity*.
Future work. We suggest privacy practitioners fill the gaps highlighted in section 7 and tackle the challenges of Appendix G. Moreover, specifically for privacy researchers, we encourage (i) improving guidance on selecting ε [27, 56] and (ii) studying and communicating to non-experts how mechanism designs affect utility. For example, studying how output consistency can imbue bias [2, 103, 104] or floating-point protection may provide less utility. As new DP deployments increasingly resort to more complicated algorithms [69], we suggest (iii) studying the unpredictable artifacts these algorithms may introduce (e.g., in the 2020 US Census [34]). Lastly, we encourage improving current proposals of differentially private (iv) ML and (v) synthetic data generation.

9 CONCLUSION

We conclude that DP can improve the work of data scientists across industries by enabling sensitive data exploration across silos, potentially shortening data access times by relaxing the adversity of data request processes, and can fulfill some types of use cases. Furthermore, analysts meaningfully and frequently employed analyses amenable to DP and, on average, would feel comfortable with a 98% of accuracy. Therefore, we suggest companies focus on privacy-enhancing analysis to harvest these benefits, not mainly on security. Moreover, we regard enabling analysts to work without "*seeing*" data and providing analysis accuracy expectation as critical, multifaceted challenges for the research community to solve. We also highlight that current open-source tools do not facilitate easy deployments, a problem requiring engineering effort within striking distance. Consequently, we encourage the community of privacy practitioners to tackle this engineering problem and ease deployments by enabling interactive dashboards, accuracy expectation measurements, improving usability and security, and integration of data annotation and access control capabilities, for ultimately bridging the identified gap between theory and practice.

ACKNOWLEDGMENTS

We sincerely thank the BMW Group for generously funding this project.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 11 pages. https://doi.org/10.1145/2976749.2978318

[2] John Abowd, Robert Ashmead, and Ryan Cumings-Menon. 2021. An Uncertainty Principle is a Price of Privacy-Preserving Microdata. *NeurIPS* (2021), 13. https://arxiv.org/abs/2110.13239

[3] ACM. 1947. ACM Digital Library. https://dl.acm.org/. Online; accessed 20 May 2022.

[4] Kareem Amin, Jennifer Gillenwater, Matthew Joseph, Alex Kulesza, and Sergei Vassilvitskii. 2022. Plume: Differential Privacy at Scale. https://doi.org/10.48550/ARXIV.2201.11603

[5] Apache Spark. 2014. PySpark Documentation. https://spark.apache.org/docs/latest/api/python/. Online; accessed 6 Mar 2022.

[6] Apple and Google. 2021. Exposure Notification Privacy-preserving Analytics (ENPA). https://covid19-static.cdn-apple.com/applications/covid19/current/static/contact-tracing/pdf/ENPA_White_Paper.pdf. Online; accessed 15 August 2022.

[7] Apple Differential Privacy Team. 2017. Learning with privacy at scale. http://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html. Online; accessed 18 February 2022.

[8] Maryam Archie, Sophie Gershon, Abigail Katcoff, and Aileen Zeng. 2018. Who's Watching? De-anonymization of Netflix Reviews using Amazon Reviews.

[9] AWS. 2006. Analytics on AWS. https://aws.amazon.com/big-data/datalakes-and-analytics/. Online; accessed 7 Mar 2022.

[10] Assaf Ben-David, Noam Nisan, and Benny Pinkas. 2008. FairplayMP: A System for Secure Multi-Party Computation. In *Proceedings of the 15th ACM Conference on Computer and Communications Security* (Alexandria, Virginia, USA) *(CCS '08)*. Association for Computing Machinery, New York, NY, USA, 257âĂŞ266. https://doi.org/10.1145/1455770.1455804

[11] Ben-David, Assaf and Nisan, Noam and Pinkas, Benny. 2015. FairplayMP repository. https://github.com/FaiplayMP/FairplayMP/blob/master/runtime/src/utils/PRG.java#L11. Online; accessed 29 July 2021.

[12] Daniel M Bittner, Alejandro E Brito, Mohsen Ghassemi, Shantanu Rane, Anand D Sarwate, and Rebecca N Wright. 2021. Understanding Privacy-Utility Trade-offs in Differentially Private Online Active Learning. *Journal of Privacy and Confidentiality* 10, 2 (2021). https://doi.org/10.29012/jpc.720

[13] Courtney Bowman, Ari Gesher, John K. Grant, and Daniel Slate. 2015. *The Architecture of Privacy.* O'Reilly.

[14] Mark Bun and Thomas Steinke. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Theory of Cryptography*, Martin Hirt and Adam Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 635–658.

[15] Andre Calero Valdez and Martina Ziefle. 2019. The usersâĂŹ perspective on the privacy-utility trade-offs in health recommender systems. *International Journal of Human-Computer Studies* 121 (2019), 108–121. https://doi.org/10.1016/j.ijhcs.2018.04.003 Advances in Computer-Human Interaction for Recommender Systems.

[16] ClĂĺment L. Canonne, Gautam Kamath, and Thomas Steinke. 2021. The Discrete Gaussian for Differential Privacy. *arXiv:2004.00010 [cs, stat]* (Jan. 2021). http://arxiv.org/abs/2004.00010 arXiv: 2004.00010.

[17] Dingfan Chen, Tribhuvanesh Orekondy, and Mario Fritz. 2020. GS-WGAN: A Gradient-Sanitized Approach for Learning Differentially Private Generators. *Neural Information Processing Systems (NeurIPS)* (2020).

[18] Rui Chen, Qian Xiao, Yu Zhang, and Jianliang Xu. 2015. Differentially Private High-Dimensional Data Publication via Sampling-Based Inference. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Sydney, NSW, Australia) *(KDD '15)*. Association for Computing Machinery, New York, NY, USA, 10 pages. https://doi.org/10.1145/2783258.2783379

[19] Cornell University. 1991. ArXiv. https://arxiv.org/. Online; accessed 28 Jul 2022.

[20] Tech Crunch. 2020. France fines Google and Amazon. https://uk.news.yahoo.com/france-fines-google-120m-amazon-085553384.html. Online; accessed 8 April 2022.

[21] Sabrina De Capitani Di Vimercati, Sara Foresti, Giovanni Livraga, and Pierangela Samarati. 2012. Data privacy: Definitions and techniques. *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems* 20, 6 (2012), 793–817. https://doi.org/10.1142/S0218488512400247

[22] Damien Desfontaines, James Voss, Bryant Gipson, and Chinmoy Mandayam. 2020. Differentially private partition selection. https://doi.org/10.48550/ARXIV.2006.03684

[23] Differential Privacy Team, Google. 2022. Privacy Loss Distribution. https://eprint.iacr.org/2018/820.pdf. Online; accessed 23 August 2022.

[24] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3574-3583.

[25] Josep Domingo-Ferrer, David SÃąnchez, and Alberto Blanco-Justicia. 2021. The limits of differential privacy (and its misuse in data release and machine learning). *Commun. ACM* 64, 7 (2021), 33–35. https://doi.org/10.1145/3433638

[26] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. 2006. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology - EUROCRYPT 2006*, Serge Vaudenay (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 486–503.

[27] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. 2019. Differential Privacy in Practice: Expose your Epsilons! *Journal of Privacy and Confidentiality* 9, 2 (2019). https://doi.org/10.29012/jpc.689

[28] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*, Shai Halevi and Tal Rabin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 265–284. https://link.springer.com/chapter/10.1007/11681878-14 Online; accessed 30 December 2021.

[29] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. 2017. Exposed! A Survey of Attacks on Private Data. *Annual Review of Statistics and Its Application* 4, 1 (2017), 61–84. https://doi.org/10.1146/annurev-statistics-060116-054123 Publisher: Annual Reviews.

[30] European Parliament and Council of the European Union. 4 May 2016. REGULATION (EU) 2016/679 Directive 95/46/EC (General Data Protection Regulation): General Data Protection Regulation. , 88 pages. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679

[31] Marco Gaboardi, Michael Hay, and Salil Vadhan. 2020. A Programming Framework for OpenDPâĹŰ. (2020), 31.

[32] Marco Gaboardi, James Honaker, Gary King, Kobbi Nissim, Jonathan Ullman, Salil Vadhan, and Jack Murtagh. 2016. PSI (Ĩĺ): a Private data Sharing Interface. In *Theory and Practice of Differential Privacy*. New York, NY. https://arxiv.org/abs/1609.04340

[33] Xianyi Gao, Bernhard Firner, Shridatt Sugrim, Victor Kaiser-Pendergrast, Yulong Yang, and Janne Lindqvist. 2014. Elastic pathing: your speed is enough to track you. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*. ACM Press, Seattle, Washington, 975–986. https://doi.org/10.1145/2632048.2632077

[34] Simson L. Garfinkel, John M. Abowd, and Sarah Powazek. 2018. Issues Encountered Deploying Differential Privacy. In *Proceedings of the 2018 Workshop on Privacy in the Electronic Society* (Toronto, Canada) *(WPES'18)*. Association for Computing Machinery, New York, NY, USA, 133-137. https://doi.org/10.1145/3267323.3268949

[35] Rod Garratt and Maarten R.C. van Oordt. 2018. Privacy as a Public Good: A Case for Electronic Cash. *Journal of Political Economy* (2018). https://doi.org/10.1086/714133

[36] Gonzalo Munilla Garrido, Joseph Near, Aitsam Muhammad, Warren He, Roman Matzutt, and Florian Matthes. 2021. Do I Get the Privacy I Need? Benchmarking Utility in Differential Privacy Libraries. (2021). arXiv:2109.10789 http://arxiv.org/abs/2109.10789

[37] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2012. Universally Utility-maximizing Privacy Mechanisms. *SIAM J. Comput.* 41, 6 (2012), 1673–1693. https://doi.org/10.1137/09076828X

[38] Google. 2008. Website. https://cloud.google.com/. Online; accessed 8 Mar 2022.

[39] Google. 2019. TensorFlow Privacy repository. https://github.com/tensorflow/privacy. Online; accessed 8 May 2022.

[40] Google. 2020. ZetaSQL repository. https://github.com/google/zetasql. Online; accessed 22 August 2022.

[41] Google. 2021. Google DP repository. https://github.com/google/differential-privacy. Online; accessed 22 August 2022.

[42] Google. 2021. Privacy on Beam repository. https://github.com/google/differential-privacy/tree/main/privacy-on-beam. Online; accessed 13 June 2022.

[43] Google Brain. 2015. TensorFlow Random Number Generator. https://www.tensorflow.org/api-docs/python/tf/random/Generator. Online; accessed 23 August 2022.

[44] Google Brain. 2019. TensorFlow Privacy Gaussian Query. https://github.com/tensorflow/privacy/blob/e826ec717a8dd93542aa7038868c8a75213836a9/tensorflow-privacy/privacy/dp-query/gaussian-query.py#L20. Online; accessed 23 August 2022.

[45] Sivakanth Gopi, Pankaj Gulhane, Janardhan Kulkarni, Judy Hanwen Shen, Milad Shokouhi, and Sergey Yekhanin. 2020. Differentially Private Set Union. https:

//doi.org/10.48550/ARXIV.2002.09745
[46] Samuel Haney, Daniel Desfontaines, Luke Hartman, Ruchit Shrestha, and Michael Hay. 2022. Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers. https://tpdp.journalprivacyconfidentiality.org/2022/papers/HaneyDHSH22.pdf. *Theory and Practice of Differential Privacy, ICML 2022* (2022). Online; accessed 22 August 2022.
[47] Harvard. 2021. OpenDP repository. https://github.com/opendp/opendp. Online; accessed 22 August 2022.
[48] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. 2020. Differential Privacy Techniques for Cyber Physical Systems: A Survey. *IEEE Communications Surveys Tutorials* 22, 1 (2020), 746–789. https://doi.org/10.1109/COMST.2019.2944748
[49] Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, Dan Zhang, and George Bissias. 2016. Exploring Privacy-Accuracy Tradeoffs Using DPComp. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 2101âĂŞ2104. https://doi.org/10.1145/2882903.2899387
[50] J. Hoelzel. 2019. Differential Privacy and the GDPR. 5, 2 (2019), 184–196. https://doi.org/10.21552/edpl/2019/2/8
[51] Naoise Holohan and Stefano Braghin. 2021. Secure Random Sampling in Differential Privacy. In *Computer Security âĂŞ ESORICS 2021*, Elisa Bertino, Haya Shulman, and Michael Waidner (Eds.). Vol. 12973. Springer International Publishing, Cham, 523–542. https://doi.org/10.1007/978-3-030-88428-4-26 Series Title: Lecture Notes in Computer Science.
[52] IBM. 2020. diffprivlib repository. https://github.com/IBM/differential-privacy-library. Online; accessed 22 August 2022.
[53] IEEE. 1963. IEEE Xplore. https://ieeexplore.ieee.org/Xplore/home.jsp. Online; accessed 20 May 2022.
[54] Insider. 2021. Amazon has been fined a record $887 million. https://www.businessinsider.com/amazon-eu-fine-data-privacy-gdpr-luxembourg-european-union-2021-7. Online; accessed 8 April 2022.
[55] Honglu Jiang, Jian Pei, Dongxiao Yu, Jiguo Yu, Bei Gong, and Xiuzhen Cheng. 2021. Applications of Differential Privacy in Social Network Analysis: A Survey. *IEEE Transactions on Knowledge and Data Engineering* (2021), 1–1. https://doi.org/10.1109/TKDE.2021.3073062
[56] Mark F. St. John, Grit Denker, Peeter Laud, Karsten Martiny, Alisa Pankova, and Dusko Pavlovic. 2021. Decision Support for Sharing Data using Differential Privacy. In *2021 IEEE Symposium on Visualization for Cyber Security (VizSec)*. 26–35. https://doi.org/10.1109/VizSec53666.2021.00008
[57] John M. Abowd, Gary L. Benedetto, Simson L. Garfinkel, Scot A. Dahl, Aref N. Dajani, Matthew Graham, Michael B. Hawes, et al. 2021. The modernization of statistical disclosure limitation at the U.S. Census bureau. https://www.census.gov/library/working-papers/2020/adrm/modernization-statistical-disclosure-limitation.html. Online; accessed 18 February 2022.
[58] Alistair E W Johnson, David J Stone, Leo A Celi, and Tom J Pollard. 2017. The MIMIC Code Repository: enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association* 25, 1 (09 2017), 32–39. https://doi.org/10.1093/jamia/ocx084 arXiv:https://academic.oup.com/jamia/article-pdf/25/1/32/34149701/ocx084.pdf
[59] Noah Johnson, Joseph P. Near, and Dawn Song. 2018. Towards Practical Differential Privacy for SQL Queries. *Proc. VLDB Endow.* 11, 5 (January 2018), 526-539. https://doi.org/10.1145/3177732.3177733
[60] Joseph P. Near. 2020. Chorus repository. https://github.com/uvm-plaid/chorus. Online; accessed 22 August 2022.
[61] Nesrine Kaaniche and Maryline Laurent. 2016. Attribute-Based Signatures for Supporting Anonymous Certification. In *Computer Security – ESORICS 2016*, Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows (Eds.). Springer International Publishing, Cham, 279–300. https://www.semanticscholar.org/paper/Attribute-Based-Signatures-for-Supporting-Anonymous-Kaaniche-Laurent-Maknavicius/3b0624ff32b9258ca2351c894d320d83a546fcd6
[62] Christopher T. Kenny, Shiro Kuriwaki, Cory McCartan, Evan T. R. Rosenman, Tyler Simko, and Kosuke Imai. 2021. The use of differential privacy for census data and its impact on redistricting: The case of the 2020 U.S. Census. 7, 41 (2021), eabk3283. https://doi.org/10.1126/sciadv.abk3283
[63] Krishnaram Kenthapadi and Thanh T. L. Tran. 2018. PriPeARL: A Framework for Privacy-Preserving Analytics and Reporting at LinkedIn. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 2183-2191. https://doi.org/10.1145/3269206.3272031
[64] Daniel Kifer, Solomon Messing, Aaron Roth, Abhradeep Thakurta, and Danfeng Zhang. 2020. Guidelines for Implementing and Auditing Differentially Private Systems. https://doi.org/10.48550/ARXIV.2002.04049
[65] Jong Wook Kim, Kennedy Edemacu, Jong Seon Kim, Yon Dohn Chung, and Beakcheol Jang. 2021. A Survey Of differential privacy-based techniques and their applicability to location-Based services. 111 (2021), 102464. https://doi.org/10.1016/j.cose.2021.102464
[66] Knime. 2006. Website. https://www.knime.com/. Online; accessed 7 Mar 2022.
[67] Daniel Kondor, Behrooz Hashemian, Yves-Alexandre de Montjoye, and Carlo Ratti. 2020. Towards Matching User Mobility Traces in Large-Scale Datasets. *IEEE Transactions on Big Data* 6, 4 (Dec. 2020), 714–726. https://doi.org/10.1109/TBDATA.2018.2871693
[68] Mathias Lécuyer, Riley Spahn, Kiran Vodrahalli, Roxana Geambasu, and Daniel Hsu. 2019. Privacy Accounting and Quality Control in the Sage Differentially Private ML Platform *(SOSP '19)*. Association for Computing Machinery, New York, NY, USA, 181âĂŞ195. https://doi.org/10.1145/3341301.3359639
[69] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. 2014. A Data- and Workload-Aware Algorithm for Range Queries under Differential Privacy. *Proc. VLDB Endow.* 7, 5 (Jan. 2014), 341âĂŞ352. https://doi.org/10.14778/2732269.2732271
[70] Zekun Li and Shuyu Li. 2017. Random forest algorithm under differential privacy. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*. 1901–1905. https://doi.org/10.1109/ICCT.2017.8359960
[71] Min Lyu, Dong Su, and Ninghui Li. 2017. Understanding the Sparse Vector Technique for Differential Privacy. *Proc. VLDB Endow.* 10, 6 (feb 2017), 637–648. https://doi.org/10.14778/3055330.3055331
[72] McKinsey & Company. 2020. Four ways to accelerate the creation of data ecosystems. https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/four-ways-to-accelerate-the-creation-of-data-ecosystems. Online; accessed 18 April 2022.
[73] H. Brendan McMahan, Galen Andrew, Ulfar Erlingsson, Steve Chien, Ilya Mironov, Nicolas Papernot, and Peter Kairouz. 2018. A General Approach to Adding Differential Privacy to Iterative Training Procedures. https://doi.org/10.48550/ARXIV.1812.06210
[74] Meta. 2021. Opacus repository. https://github.com/pytorch/opacus Online; accessed 8 May 2022.
[75] Microsoft. 2010. Azure. https://azure.microsoft.com/en-us/. Online; accessed 28 July 2022.
[76] Ilya Mironov. 2012. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM conference on Computer and communications security - CCS '12*. ACM Press, Raleigh, North Carolina, USA, 650. https://doi.org/10.1145/2382196.2382264
[77] Priyanka Nanayakkara, Johes Bater, Xi He, Jessica Hullman, and Jennie Rogers. 2022. Visualizing Privacy-Utility Trade-Offs in Differentially Private Data Releases. 2022, 2 (2022), 601–618. https://doi.org/10.2478/popets-2022-0058
[78] Arjun Narayan and Andreas Haeberlen. 2012. DJoin: Differentially Private Join Queries over Distributed Databases. (2012), 14.
[79] Arvind Narayanan and Vitaly Shmatikov. 2008. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*. IEEE, Oakland, CA, USA, 111–125. https://doi.org/10.1109/SP.2008.33 ISSN: 1081-6011.
[80] Office of the Attorney General. 2018. California Consumer Privacy Act (CCPA) | State of California - Department of Justice - Office of the Attorney General. https://oag.ca.gov/privacy/ccpa.
[81] OMTP. 2009. Advanced Trusted Environment: OMTP TR1. , 204 pages. http://www.gsma.com/newsroom/wp-content/uploads/2012/03/omtpadvancedtrustedenvironmentomtptr1v11.pdf
[82] OpenDP. 2021. SmartNoise Core. https://github.com/opendp/smartnoise-core. Online; accessed 22 August 2022.
[83] OpenMined. 2022. PipelineDP repository. https://github.com/OpenMined/PipelineDP. Online; accessed 27 July 2022.
[84] Pandas developer community. 2008. Pandas Documentation. https://pandas.pydata.org/docs/. Online; accessed 6 Mar 2022.
[85] Project Jupyter. 2015. Jupyter Documentation. https://jupyter.org/. Online; accessed 6 Mar 2022.
[86] PyCaret community. 2020. PyCaret open-source repository. https://github.com/pycaret/pycaret. Online; accessed 22 August 2022.
[87] IBM report. 2022. Cost of a Data Breach Report. https://www.ibm.com/downloads/cas/OJDVQGRY. Online; accessed 11 April 2022.
[88] Ryan Rogers. 2020. A Differentially Private Data Analytics API at Scale. In *2020 USENIX Conference on Privacy Engineering Practice and Respect (PEPR 20)*. USENIX Association. https://www.usenix.org/conference/pepr20/presentation/rogers
[89] Ryan Rogers, Subbu Subramaniam, Sean Peng, David Durfee, Seunghyun Lee, Santosh Kumar Kancha, Shraddha Sahay, and Parvez Ahammad. 2021. LinkedIn's Audience Engagements API: A Privacy Preserving Data Analytics System at Scale. 11, 3 (2021). https://doi.org/10.29012/jpc.782
[90] Indrajit Roy, Srinath T V Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. 2010. Airavat: Security and Privacy for MapReduce. (2010), 16.

[91] Roy, Indrajit and Setty, Srinath T V and Kilzer, Ann and Shmatikov, Vitaly and Witchel, Emmett. 2013. Airavat repository. https://github.com/bochunz/Airavat/blob/master/AiravatJvmTest.java#L11. Online; accessed 29 July 2021.
[92] ScienceDirect. 1997. ScienceDirect Digital Library. https://www.sciencedirect.com/. Online; accessed 20 May 2022.
[93] Lingling Shen, Xiaotong Wu, Datong Wu, Xiaolong Xu, and Lianyong Qi. 2020. A Survey on Randomized Mechanisms for Statistical Learning under Local Differential Privacy. In *2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 1195–1202. https://doi.org/10.1109/HPCC-SmartCity-DSS50907.2020.00155
[94] Latanya Sweeney, Akua Abu, and Julia Winn. 2013. Identifying Participants in the Personal Genome Project by Name. *SSRN Electronic Journal* (2013). https://doi.org/10.2139/ssrn.2257732
[95] Tableau Software, Inc. 2003. Tableau website. https://www.tableau.com/. Online; accessed 6 Mar 2022.
[96] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. 2021. Benchmarking Differentially Private Synthetic Data Generation Algorithms. https://doi.org/10.48550/ARXIV.2112.09238
[97] Pratiksha Thaker, Mihai Budiu, Parikshit Gopalan, Udi Wieder, and Matei Zaharia. 2020. Overlook: Differentially Private Exploratory Visualization for Big Data. In *Workshop on Theory and Practice of Differential Privacy*. https://arxiv.org/abs/2006.12018
[98] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. 2019. Dp-cgan: Differentially private synthetic data and label generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.
[99] Tumult Labs. 2021. Tumult Analytics repository. https://gitlab.com/tumult-labs/analytics. Online; accessed 15 August 2022.
[100] United States Court of Appeals, Ninth Circuit. 2012. United States v. Huping Zhou. https://caselaw.findlaw.com/us-9th-circuit/1600563.html. Online; accessed 28 July 2022.
[101] Elisabet Lobo Vesga, Alejandro Russo, and Marco Gaboardi. 2020. A Programming Framework for Differential Privacy with Accuracy Concentration Bounds. *2020 IEEE Symposium on Security and Privacy (SP)* (2020), 411–428.
[102] Lun Wang, Joseph P. Near, Neel Somani, Peng Gao, Andrew Low, David Dao, and Dawn Song. 2019. Data Capsule: A New Paradigm for Automatic Compliance with Data Privacy Regulations. In *Heterogeneous Data Management, Polystores, and Analytics for Healthcare*, Vijay Gadepally, Timothy Mattson, Michael Stonebraker, Fusheng Wang, Gang Luo, Yanhui Laing, and Alevtina Dubovitskaya (Eds.). Springer International Publishing, Cham, 3–23.
[103] Tianhao Wang, Zitao Li, Ninghui Li, Milan Lopuhaä-Zwakenberg, and Boris Skoric. 2019. Consistent and Accurate Frequency Oracles under Local Differential Privacy. *CoRR* abs/1905.08320 (2019). arXiv:1905.08320 http://arxiv.org/abs/1905.08320
[104] Tianhao Wang, Milan Lopuhaa-Zwakenberg, Zitao Li, Boris Skoric, and Ninghui Li. 2020. Locally Differentially Private Frequency Estimation with Consistency. In *Proceedings 2020 Network and Distributed System Security Symposium* (San Diego, CA, 2020). Internet Society. https://doi.org/10.14722/ndss.2020.24157
[105] Yufeng Wang, Minjie Huang, Qun Jin, and Jianhua Ma. 2018. DP3: A Differential Privacy-Based Privacy-Preserving Indoor Localization Mechanism. *IEEE Communications Letters* 22, 12 (2018), 2547–2550. https://doi.org/10.1109/LCOMM.2018.2876449
[106] Matthew Wilchek and Yingjie Wang. 2021. Synthetic Differential Privacy Data Generation for Revealing Bias Modelling Risks. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. 1574–1580. https://doi.org/10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00211
[107] Royce J Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. 2019. Differentially Private SQL with Bounded User Contribution. https://doi.org/10.48550/ARXIV.1909.01917
[108] Felix T Wu. 2012. Defining Privacy and Utility in Data Sets. *84 University of Colorado Law Review 1117 (2013); 2012 TRPC* (2012), 1117–1177. https://doi.org/10.2139/ssrn.2031808
[109] Bangzhou Xin, Wei Yang, Yangyang Geng, Sheng Chen, Shaowei Wang, and Liusheng Huang. 2020. Private FL-GAN: Differential Privacy Synthetic Data Generation Based on Federated Learning. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2927–2931. https://doi.org/10.1109/ICASSP40776.2020.9054559
[110] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren. 2019. GANobfuscator: Mitigating Information Leakage Under GAN via Differential Privacy. *IEEE Transactions on Information Forensics and Security* 14, 9 (2019), 2358–2371. https://doi.org/10.1109/TIFS.2019.2897874
[111] Depeng Xu, Shuhan Yuan, and Xintao Wu. 2017. Differential Privacy Preserving Causal Graph Discovery. In *2017 IEEE Symposium on Privacy-Aware Computing (PAC)*. 60–71. https://doi.org/10.1109/PAC.2017.24
[112] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. *ACM Trans. Database Syst.* 42, 4, Article 25 (oct 2017), 41 pages. https://doi.org/10.1145/3134428
[113] Jun Zhang, Xiaokui Xiao, Y. Yang, Zhenjie Zhang, and Marianne Winslett. 2013. PrivGene: differentially private model fitting using genetic algorithms. In *SIGMOD '13*.
[114] Ding Zhe, Chunwang Wu, Zhao Jun, and Binyong Li. 2020. Frequent Itemsets Mining Algorithm based On Differential Privacy and FP-Tree. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 271–274. https://doi.org/10.1109/ICCWAMTIP51612.2020.9317373

A INTERVIEWED COMPANIES OVERVIEW

Table 2 presents a summary of the characteristics of the interviewed companies. The companies belong to a diverse set of industries, predominantly SW development, and 4 of the 9 companies are significantly large, with over 100, 000 employees. 5 companies are under the jurisdiction of the EU with regulations such as the General Data Protection Regulation [30], and 4 companies operate under US law, e.g., the California Consumer Privacy Act [80].

Table 2: Overview of the interviewed companies. Legend: SW dev. = Software development

Industry (focus)	Size (employees)	Team's Location
Team operates internationally		
Automotive (car manufacturer)	$> 100,000$	Germany
Insurance (health)	$> 100,000$	Germany
SW dev. (data processing)	$< 2,000$	Germany
SW dev. (subscription newsletters)	$< 2,000$	USA
Team operates nationally		
Consultancy (banking and big pharma)	$> 100,000$	Spain
Entertainment (finance)	$< 2,000$	USA
SW dev. (business operations)	$> 100,000$	Germany
SW dev. (data processing)	$< 2,000$	USA
SW dev. (smart sound system)	$< 2,000$	USA

B INTERVIEW QUESTIONNAIRE FOR DATA STEWARDS

RQ1: *What is the context of privacy protection in the targeted organization?*

Q1: *What is the institution's motivation for privacy protection?*

Q2: *What are your privacy concerns when an analyst has full dataset access?*

Q3: *At what level of data granularity are you protecting and measuring privacy?*

Q4: *What could be improved in the dataset request process?*

Q5: *What are your typical questions for the current interview-based full dataset access authorization?*

Q6: *Instead of the interview process, would you be capable to run a program provided by the analyst s.t. the analysis is carried out without the analyst ever "seeing" the dataset?*

C INTERVIEW QUESTIONNAIRE FOR DATA ANALYSTS

As we interviewed non-experts in differential privacy, we minimized the number of questions that contained the words or required knowledge of "*differential privacy*". We kept a few because we aimed to assess whether systems offering differential privacy functionality could be valuable to analysts. First, we briefly explained differential privacy in a simplified manner: "*Differential privacy is a technique that adds noise to analytics results so that one cannot reverse engineer the outputs to a specific person.*" Additionally, if we perceived the interviewees were disoriented with Q18 or Q19, we explained that the hypothetical system would be the same as the one they used every day, the only difference being that the results would slightly differ from the deterministic outputs. Picturing the system they used daily was very helpful for imagining one where the outputs are noisy. Furthermore, we carefully parsed their answers to assess whether they understood the concept or its integration into their system. If they did not, we kindly repeated the procedure above.

RQ2: *Could differential privacy tackle the privacy-related pain points of an analysis workflow in an organization?*
Q7: *What is your workflow to analyze data?*
Q8: *Why do you need full dataset access?*
Q9: *How often do you request full dataset access? How long does it usually take?*
Q10: *What do you think about the process to request full dataset access in your organization?*
Q11: *What features do you think are missing in your organization's data analysis workflow?*

RQ3: *When does differential privacy impede an analysis?*
Q12: *In which analytics use cases have you been involved?*
Q13: *Is SQL-meaningful for your work? How many SQL-like queries do you make weekly?*
Q14: *How often do you need machine learning to fulfill your analysis in contrast to using SQL?*
Q15: *What are your most used machine learning models?*
Q16: *If you were to use differential privacy to fulfill your analysis, when and how much accuracy would you be willing to forgo?*

RQ4: *How would differential privacy affect the workflow of an analyst?*
Q17: *How much would the noise affect your analysis?*
Q18: *Would you find it helpful to execute differentially private SQL queries to explore and fully analyse datasets without the standard permissions?*
Q19: *Only based on the information extracted from a dataset exploration with differential privacy, could you write a script to fulfill your analysis goal?*
Q20: *What are the minimum properties for you as an analyst such that you are confident to write an analysis script without full dataset access?*
Q21: *Would you find it helpful to use a dynamic dashboard that visualizes dataset information with differential privacy?*

RQ5: *Can differential privacy be applied to the frequent SQL-like queries analysts execute?*
Q22: *What are your top SQL-like queries before you have full dataset access?*
Q23: *What are your top SQL-like queries after you have full dataset access?*
Q24: *What is the ratio between aggregation queries and queries to retrieve items?*

D FREQUENT QUERIES

In Table 3, we include the most frequent SQL queries recorded during the interviews with the data analysts before and after accessing a dataset. Note that not all analysts were allowed to explore datasets and a few did not employ SQL for data preparation or analytics; instead, they resorted to Python scripts for statistical analysis, ML, and visualization or tools such as Tableau [95], Knime [66], or proprietary SAP data management software. For exploring the dataset prior to access, 14 analysts resourced to `SELECT *` to get a "*feeling*" for the data. Furthermore, `COUNT` and `DISTINCT`, and `WHERE` and `GROUP BY` were the most frequently used functions and clauses, respectively.

Table 3: Most frequent queries before (data exploration) and after (data preparation/analysis) data access. Legend: Freq. = Frequency (i.e., number of analysts who used such query).

Query	Freq. before access	Freq. after access
Function		
`COUNT`	7	7
`DISTINCT`	6	4
`MAX`	4	5
`MIN`	4	5
`AVG`	4	4
`VAR`	2	3
Statement		
`SELECT * LIMIT`	14	12
Clause		
`WHERE`	13	10
`GROUP BY`	12	9
`JOIN`	2	8

E FREQUENTLY ASKED QUESTIONS FROM DATA STEWARDS TO ANALYSTS

We compiled the most frequently asked questions data stewards make to data analysts during the data access request process.

- Could you describe in detail the analytics use case?
- Is the use case approved by the corresponding internal stakeholders?
- Why is the dataset needed?
- Is the dataset adequate regarding quality, volume, and use case?
- Could you reach the goal without dataset access?
- Is the entire dataset needed or only a set of attributes?
- Is the dataset already available, or must a data engineer create a new dataset?

- Is the dataset classified as very sensitive? If affirmative, additional access control measures and monitoring must be defined in detail.

F OPEN-SOURCE TOOLS DESCRIPTIONS

We provide a quick description of each of the selected open-source tools mapped to the key system desiderata in section 7 appearing in Table 1.

Libraries

diffprivlib: IBM researchers developed a general-purpose Python library to execute differentially private aggregation queries and machine learning in the context of data science (namely Notebooks) [52].

Google DP: Google researchers developed a library in multiple languages (C++, Go, and Java) that an expert may use to build new applications supporting differential privacy [41].

Opacus: Meta researchers developed a library dedicated to training machine learning models offered by PyTorch in a differentially private manner [74].

OpenDP: Harvard implemented a flexible architecture for differentially private analysis, consisting of a (pluggable) runtime in Rust wrapped around a Python API, in addition to a "validator" that calculates parameters such as the sensitivity of a query. [47].

TensorFlow Privacy: Google researchers developed a library that includes TensorFlow differentially-private optimizers for training machine learning models [39].

Frameworks

Chorus: Johnson et al.'s [59, 60] wrote a framework in Scala that works in cooperation with existing infrastructure (a SQL database) to explore the use of differentially private SQL queries at scale.

PipelineDP (experimental): OpenMined, in collaboration with Google, propose a framework to execute differentially private aggregations in large-scale datasets using batch processing systems (Apache Spark and Apache Beam) [83].

Privacy on Beam: Similarly to PipelineDP, Privacy on Beam [42] proposes a solution based on Apache Beam and Google DP [41] for executing differentially private analytics at scale.

Tumult Analytics: Tumult Labs provides a Python library built atop a framework similar to OpenDP for computing aggregate statistics over tabular data at scale [99].

ZetaSQL: Google researchers wrote a framework for SQL that defines a language, a parser, and an analyzer meant to work with an existing database engine [40].

Systems:

Airavat: Roy et al. [90] designed a MapReduce-base system written in Java for distributed computations on sensitive data that integrates differential privacy and access control with policies defined by data owners/stewards.

DJoin: Narayan et al. [78] built a system capable of processing a wide range of differentially private SQL queries across datasets from different organizations and leverages homomorphic primitives to hide inputs.

User Interfaces:

Bittner et. al [12]: With a focus on ML, Bittner et. al aim to help researchers decide which algorithm to use by offering an interface that quantifies the disclosure risk of different algorithms.

DPcomp: A web-based system enabling researchers to assess the utility of differentially private algorithms and understand their respective incurred error [49].

DPP: This user interface specifically helps data owners to set the noise level per the disclosure risk of an attribute. The underlying mechanism relies on a novel parameter selection procedure for differential privacy [56].

Overlook: Thaker et al. [97] designed a system for differentially private data exploration that supports counts with an interactive browser-interfacing dashboard (namely visualizing histograms).

PSI (Ψ): Harvard's Privacy Tools Project works on a data sharing interfaces for researchers to explore datasets with differential privacy [32].

ViP: Visualizing Privacy is an interface that provides information about the relationships between utility, ε, and disclosure risk (among others), allowing users to adjust the privacy parameters of their analysis based on visualizations of expected risk and accuracy [77].

G DIFFERENTIAL PRIVACY CHALLENGES

This section enumerates other critical challenges we encourage researchers and system designers to investigate.

(1) While DP is highly adaptable to use cases (e.g., using the local or central model) and algorithms (e.g., queries or ML), the adaptations are non-trivial and have often led to erroneous implementations [25]. Thus, practitioners should exercise extreme care to ensure the correctness of their DP implementation with the same sentiment as "*do not write your own crypto*."

(2) *Fairness* could be another obstacle to DP adoption, which Harvard researchers also highlighted when referring to the US Census of 2020 [62]. Specifically, one analyst underlined the topic of fairness when asked about how noise would affect their analysis (Q17). If analysts add differentially private noise during training underwriting linear regression models, users might be over- or under-funded. While the company would not incur a loss as the predictions would be "right" on average, the effect noise has on their users could impact their brand perception.

(3) Managing *user-level* privacy budgets in user data streams [68].

(4) Tracking the privacy budget across systems and adapting the noise level based on the remaining budget.

(5) There is a significant difference between the local and central model noise levels.

(6) Choosing ε and other privacy parameters [27].

(7) Building systems that fulfill DP for (i) large-scale datasets (ii) when users make contributions to multiple records (iii) with unknown domains [4].

(8) Verifying DP compliance of a complex system by proving and fitting a mathematical model to the system's semantics [64] and developing unit tests to ensure the system conforms to such model.

H SYSTEM DESIGN

Although there are many potential ways to construct privacy-enhancing analytics systems, to show the feasibility of covering all system desiderata presented in section 7, this section discusses one

design to guide practitioners in their development. The design is in an early stage, and, thus, we cannot discuss the components in detail. Instead, we sketch the system's primary components, aiming to spark interest in further system development and research in the community.

We consider two roles interacting with the system: (i) *data stewards* have the authority to access the original data and the legal background for data management. Stewards can authorize data access inside an organization and ensure compliance. (ii) *Data analysts* analyze data to fulfill use cases. Analysts often need to access data by employing SQL aggregation or retrieval and python scripts. In the current system design, we mainly consider SQL aggregation and retrieval. Lastly, we assume that analysts cannot share query results with unauthorized recipients. In such a setting, we present our high-level system design blueprint in Fig. 2.

The components are the following:

Database Schema: The system requires one dedicated component to manage the database schema and ensure its consistency at all places to make sure all components have a consistent view of the processed data format.

Policy Panel: Data stewards create and update the configuration stored in the policy panel to authorize data access from data analysts to satisfy desiderata **(X)**, annotate data sensitivity to satisfy desiderata **(VIII)** and **(IX)**, and ensure compliance. Other system components rely on the configuration in the policy panel to decide whether to proceed with particular requests or queries.

User Registration Service: The user registration service component maintains a user system to standardize the onboarding procedure of data analysts to satisfy desiderata **(X)**; thus, the system can distinguish between different data analysts with different data access requirements and permissions.

Statistic Dashboard: The statistic dashboard is a privacy-enhanced visualization for database statistics, which will help authorized data analysts explore datasets, thus satisfying desiderata **(V)**.

Query Gateway: The query gateway reads annotated data schema from the policy panel and uses it to analyze the query structure, parsing the query for later stages. The system can thus run a preliminary policy check on the incoming query and route it to the corresponding database proxy.

Original Database: The database service stores the original sensitive data securely to satisfy desiderata **(III)**, ideally with encrypted storage and restricted access for the data steward and other necessary system security components to fully comply with **(III)**.

Budget Manager: With the information about both the database schema and the sensitivity annotation from the policy panel, the budget manager models the differential privacy budget and keeps track of the budget consumption in various queries.

Differentially Private Database Proxy: Before executing the query from the data analyst on the tables in the original database, the proxy analyzes how to apply differential privacy by transforming the query and also calculates the budget consumption to satisfy desiderata **(I), (II)** and **(VI)**. Before returning the query result, it also outputs the query's accuracy estimation to satisfy **(VII)**.

Synthesized Database: The synthesized database maintains the dummy or differential-privately synthesized versions of tables in the original database to satisfy desiderata **(IV)**.

Synthetic Database Proxy: Upon receiving queries to the dummy or differential-privately synthesized data, the proxy checks whether the required version of the tables has already been generated in the synthesized database. If the required version is missing, the proxy orchestrates the generation procedure from the original database on-demand.

Lastly, we describe the communication between system components to explain the workflow to access different privacy-enhancing analytic functionalities.

(1) Data Analyst User Registration. Once the system is correctly set up, data analysts should begin to create their user accounts in the system with the user registration service.

(2) Submission of the Query Request. The request should include both the SQL query and a piece of metadata that specify the privacy details like accuracy requirements or whether to use the dummy or synthesized data. The query gateway checks the query request to see if it is compliant with the system policy and routes it to the corresponding database proxy.

(3) Exploring Data Statistics on the Dashboard. Data analysts can use the dashboard to explore dataset statistics that are periodically gathered from the original database.

(4) Data Steward Adjustment for Data Access Policy. In addition to allowing the data steward to configure the data access policy manually, ideally, the policy panel should also make suggestions on potentially useful policy changes. Such suggestions can be based on the data access application or frequently rejected requests to other system components during user registration.

(5) User Registration Following Policies in the Policy Panel. If the data steward decides to include specific steps during the registration procedure (e.g., signing acknowledgments, reading materials, finishing tutorials), the registration procedure would reflect such requirements.

(6) User Access to Query and Dashboard Service Determined by Policy. Considering both queries and the statistic dashboard exploration reveal information about the original data, the policy panel should control the access of data analysts to both services.

(7) Query Execution With One Proxy. Depending on the privacy-related metadata, one of the proxies executes the query with its transformation and returns the query result with privacy details like result accuracy or budget consumption. If the result consumes the privacy budget, the proxy also notifies the budget manager to track the change.

(8) On-Demand Data Synthesis. If the synthetic database proxy cannot find the required version of the dummy of synthetic tables from the synthesized database, it triggers the generation of that required version.

(9) Unified Schema Synchronization Between System Components. The database schema component enforces consistency by tracking the changes in the original database. It locks the whole system for schema changes until the updates are applied to all system components.

As of September 2022, our GitHub hosts an early-stage open-source effort to benchmark libraries and frameworks suitable for some of the system components in Fig. 2:

https://github.com/camelop/dp-lab

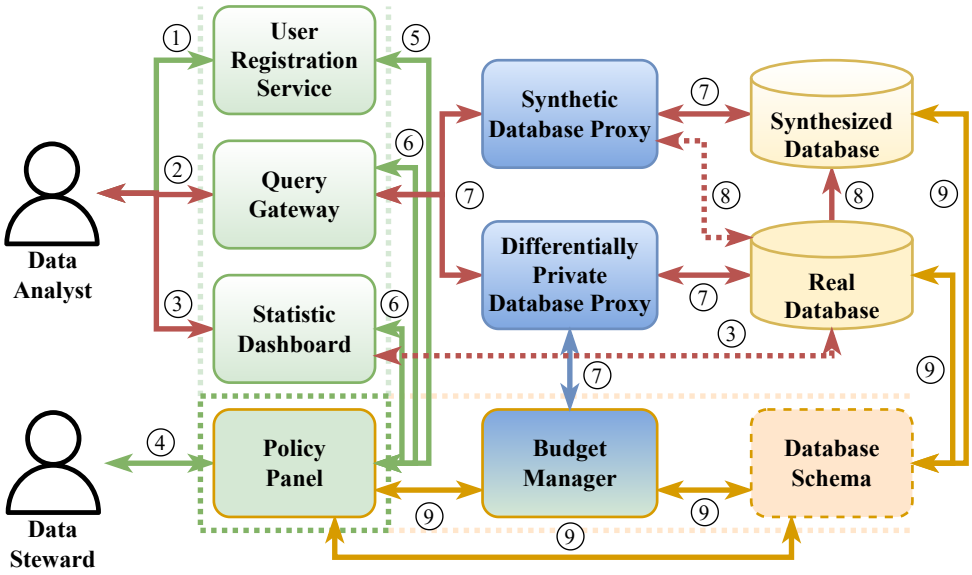

Figure 2: High-level system design blueprint of a privacy-enhancing analytics tool. The components and communication links are described in Appendix H. Solid lines represent communications between components triggered by all relevant query events, while dashed lines represent communications that happen periodically or only under certain circumstances. We specify those circumstances in the workflow description.