Privacy-preserving Data Sharing Collaborations: Architectural Solutions and Trade-off Analysis.

Michiel Willocx, Vincent Reniers, Dimitri Van Landuyt,
Bert Lagaisse, Wouter Joosen, Vincent Naessens
DistriNet - KU Leuven
Gent and Leuven, Belgium
firstname.lastname@kuleuven.be

***Abstract*—Businesses and governments possess vast data with potential for analytical insights in areas like business intelligence (consumer behavior, business solvability) and governmental insights on population (crime, fraud). However, two primary challenges hinder the adoption of data-driven analytics: the lack of in-house expertise and the absence of sufficient data, which often requires collaboration with third parties. Such partnerships, especially involving Machine Learning (ML), raise concerns due to the sensitive nature of the data. This paper outlines two realistic use cases and proposes two privacy-preserving data sharing architectures tailored for business-to-business and government-to-business contexts. The first architecture uses de-identification techniques before and during data transmission, while the second assumes an already existing baseline ML model to test and refine predictions without sharing data. We present an in-depth analysis and evaluation of these architectures focusing on their complexity, trust requirements, and data-sharing efficacy.**

Keywords-privacy enhancing technologies, data collaborations, anonymity, utility

I. Introduction

Nowadays, companies and organizations have widely adopted the practice of collecting massive amounts of data during daily business activities. Businesses increasingly recognize the value of this data, or as commonly said “data is the new gold”. Companies apply data for targeted marketing campaigns, to optimize the manufacturing process and inner workings, or even to increase customer satisfaction. Not only businesses, but also governments are increasingly interested in looking into ways to further collect or apply existing data. Governments are already sitting on vast amounts of data that can be put to work, for example to detect social fraud. For example, the case in which people benefit from social housing while in fact already owning (foreign) housing property [1]. Similarly, governmental data can be used to further improve crime fighting effectiveness (e.g., identifying problematic areas) [2] and for predictive analyses [3]. In order for governments and organizations to be able to perform these analyses, they first require (i) sufficient data, and secondly (ii) the important know-how to process this data. However, many business and governmental bodies are often lacking in both areas. While many organizations already possess a significant amount of data, more external data is often required in order to build qualitative prediction models. This data is to be acquired from third parties. Moreover, building these models (ML or otherwise) is often no easy feat, requiring expertise and experience to establish such models, and to subsequently evaluate and validate their accuracy. Many businesses therefore rely on third parties (i.e., data analytic parties) specialized in performing data analytics and building ML models for this.

In practice, organizations engage or desire to participate in a data sharing ecosystem that is highly beneficial to all parties. Such a data ecosystem involves close cooperation between different data owners on one hand, and between data owners and the ML party on the other hand. The benefits of sharing data are typically win-win situations, although establishing these data sharing collaborations comes with key problems related to both privacy and trust. The privacy problems are related to the fact that the data shared between parties often contains sensitive and/or personal information. The GDPR regulation [4] states that the type of personal data can only be shared when sufficiently anonymized. This means that records — or in some cases even attributes — in the shared dataset may no longer be linkable to an individual. In most cases, these types of issues can be tackled with state-of-the-art privacy models such as k-anonymity [5] and l-diversity [6]. Furthermore, the information may also be sensitive to the company, involving details on their inner workings (e.g., customer base), results (e.g., sales), or on collaboration with other companies, and sharing these data may impact their competitive advantage or reputation. Companies are very reluctant to share this type of information with their direct competitors, requiring significant trust, even if the result of the analysis of the data over all parties involved could be beneficial for all parties. Similarly, governments also posses data which may be highly beneficial when analyzed further by third-party analytical parties, yet this may not negatively affect the trust citizens have in their government and the safekeeping of their data.

In cases where a data provider is sharing data with a data analytics party, it sometimes suffices that both parties sign a non disclosure agreement. Even in this case, reducing the amount of required trust in such collaborations is desirable. New data collaboration strategies are required to solve these situations, where the end goal is the sharing of data for the purposes of gaining analytical insights via ML. In particular, such governmental-business or B2B data sharing require in-depth analysis of the requirements involved, as well as the technological solutions present to limit issues related to privacy and trust.

In this paper, we present two real-life use-cases from our collaboration with industry partners. In these cases, data

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-244-9

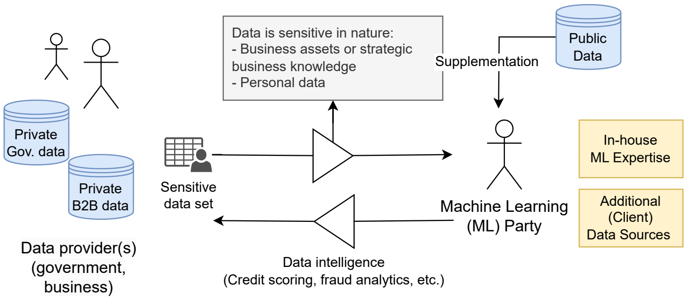
Data is sensitive in nature:
- Business assets or strategic business knowledge
- Personal data
Supplementation
Public Data
Private Gov. data
Private B2B data
Sensitive data set
In-house ML Expertise
Additional (Client) Data Sources
Machine Learning (ML) Party
Data provider(s) (government, business)
Data intelligence (Credit scoring, fraud analytics, etc.)

Figure 1. Use cases overview: Single or multiple data provider(s).

collaborations between different parties (data owner(s) and data analytics party) are required to solve important societal and economical problems. The specific nuances of each use case are laid out in detail. Next, two reference architectures are presented that allow to perform this type of data collaborations. The first strategy focuses on purely statistical methods, while the second one combines cryptographic constructs and statistics. The main advantages and disadvantages of each approach are evaluated and discussed in-depth.

The remainder of this paper is structured as follows. Section II introduces our motivating industry use cases and their requirements. Subsequently, Section III proposes architectural solutions, and the respective technological solutions that can be applied for each architecture. Section IV provides an in-depth comparison of both architectures via a trade-off analysis on aspects such as approach, complexity and implementation effort, and trust assumptions. Finally, Section V positions our work in the state-of-the-art, and Section VI concludes.

II. Motivating industry use cases

The research presented in this paper is driven by the use cases of two industry partners involving B2B and G2B data sharing for the purpose of ML analytics. In each use case, data is provided to the *machine learning (ML) party* by either one or *multiple data providers* (e.g., companies, organizations or governmental institutions), as shown in Figure 1. The ML party processes this data to generate insights via ML model creation and training, which they subsequently offer as products to the *ML party's customers*. The incentive for the *data provider(s)* is either of monetary value, or to acquire improved insights using their own data. Typically, the data provider lacks either (i) know-how to generate such analytical insights in-house or (ii) lack additional data gathered from a multitude of sources to do so. For example, it's possible that the ML party may incorporate additional collected data from private or public sources to enrich the data provider's data, and which enable it to in fact generate these insights. In both of our cases, the data provider is in fact also the end customer, which is the direct beneficiary of the obtained insights.

A. Use cases

We start off first by explaining the common denominator between all use cases, regarding their willingness to share data, which has to be abetted with privacy-preserving data sharing techniques. Subsequently, we detail our uses cases.

1) Problem statement: Willingness to share privacy-sensitive data: The common denominator between both use cases is that the data providers wish to share data, but are inhibited by the sensitive nature of the data. The data to be shared may involve information about other companies, internal company information that when leaked may for example impact the company's competitive standing or reputation. The data may also come from governmental sources, or involve information on data subjects that should not be disclosed. Therefore, typically a relationship of trust has to be established between the data provider and the machine learning party before data is shared, for example via contractual agreements. In this research, we want to avoid or minimize the degree of trust required before sharing data, by either applying privacy tactics and abstracting the data to a degree, or keeping the data on-site and steering the ML party's generated insights. Such tactics can enable B2B data sharing without a significant vote of trust in the ML party, by managing what is shared, and what can be learnt from it. Finally, the customers of the end ML party's trained model, should never be able to infer on which data was used to train the data set. In the next subsection, we more precisely formulate the nuances of each use case, after which we will translate these use cases into concrete requirements for both data provider and the ML party.

This paper identifies two theoretical dimensions that are possible in a data sharing ecosystem, namely regarding (i) sensitive nature of data (e.g., personal or non-personal data), and secondly (ii) number of data providers. The first dimension involves the nature of the sensitive data that is being shared, which may either be personal or non-personal data, and which influences the solution architecture on the potential requirement for GDPR-compliancy. The second dimension involves the number of data providers which are involved for one case, which may either be one entity providing all data required, or many entities providing pieces of the puzzle. The latter dimension may also impact potential solutions, for example when there is only one data provider the source is already self-evident. In the case of multiple data providers, possible solutions may rely on techniques such as e.g., multi-party computation to enable data intelligence gathering. These dimensions theoretically cover numerous use cases involved with data sharing, which amount to several possible combinations, for example non-GPDR data and many-to-ML data providers. Our two industry use cases that motivate this work represent different characteristics in these both dimensions. These use cases may practically apply to a wide range of B2B or G2B data sharing scenarios. We will detail the nuances of each of these two specific use cases, their dimensions, and analyze the requirements of each stakeholder, more specifically the requirements of the ML party or data provider.

2) Industry use case A: Single governmental data provider and ML party (G2ML): In our first use case A, which motivates this work, the data provider is a single entity, namely the government, that contains all required data on which the ML party can generate insights. Governments can have vast pools of even public data that can be accessed, such as information on the population, maps (building zones, agriculture), environment (e.g., pollution). There may also be more restrictive information involved, for example stored in police or judicial databases. Based on such vast pools of both public and private data the government can generate a multitude of insights, to improve their enactment on for example environmental polluters, or to identify problematic areas in society. Governments typically lack in-house expertise to generate such insights and therefore ideally rely on external an ML party. In this case, when private data is involved, the government has to either establish a trust relationship with the ML party, although this can be tricky. Therefore, we aim to provide technological solutions such as privacy-preserving techniques and architectures to limit such trust requirements in the ML party. Certain other requirements may also apply, such as requiring the ML party not to disclose any of the shared data, or even when privacy techniques are applied, to not disclose the learned ML insights.

3) Industry use case B: Joining business data to generate sector insights (B2ML): In our use case B, multiple industry companies are data providers to a single machine learning party. The data shared in the many-to-one relationship can be of non-personal sensitive data, or data which has to be GDPR compliant. For example, supermarkets may store a lot of data on the shopping behavior of customers, such as products frequently bought together, or even have the potential to store very specific information on a per-customer basis. These companies could contract a ML party to process their data and generate sector-wide insights regarding purchasing behavior, which could for example optimize advertising campaigns. In certain cases, companies are however not inclined to share the full details of the data set, or wish to omit certain person-specific aspects, and this then requires technological solutions to enable the B2ML data sharing process. Another requirement is that these companies may not wish to share data among each other, as this may impact their individual competitive standing. In addition, the ML party must be trusted to not disclose individual datasets to other competitors, or we can rely on privacy-preserving tactics to facilitate this requirement. In the next section, we generalize the requirements for each stakeholder, for example regarding the data provider and ML party. These requirements will guide our architectural solutions for privacy-preserving B2B or G2B data sharing.

B. Requirements analysis

We enumerate the requirements for each party of primarily the data provider, and the machine learning party that processes this data. These requirements relate either to privacy aspects, functionality, or non-functional requirements.

1) Data provider requirements (R_{DP}): The data providers are companies or governmental institutions, which feature a certain degree of willingness to share data, or are highly reluctant to do so unless certain privacy guarantees are met. This property of willingess to share data will impose more subtle or stringent privacy tactics. For example, the data that is shared may not be leaked to any other party than the ML, or even more stringent, the ML party itself may not be able to deduct which subjects were in the shared data set, a property referred to as unidentifiability or unlinkability. These privacy tactics ideally won't impede certain functional requirements the data provider desires in return for providing the data. For example, the data shared must yield the data provider itself with additional insights learned by and from the ML party.

2) ML party requirements (R_{ML}): In terms of functional requirements, similarly the ML party will want to acquire as much data as possible, or sufficient data that enables them to generate ML insights. These insights can be of use to their customers, which in our use cases is the data provider itself. In terms of privacy requirements, which may be imposed by the data provider, the shared insights cannot be used to deduce which data subject was involved in the training set (membership inference). In addition, the ML party wants to establish a certain degree of trust in their process, which can come from the privacy tactics that are applied before sharing the data with the ML party. Such trust in algorithms, rather than the parties themselves, will promote future data sharing collaborations. Finally, regarding the insights generated, the process of the ML model may be subject to intellectural property rights (IPR), and therefore details regarding the ML model are ideally not disclosed to any of the other stakeholders, as this may compromise their core business. Optionally, the insights when disclosed are also of a sufficient quality when shared (i.e., good accuracy), to promote the reputation of the ML party.

3) ML party customer requirements (R_C): The customer of the ML party is typically the data provider itself, but also other organizations or institutions that can benefit from these insights, or generate additional insights using additional combinations of their data. The requirements for this customer is that the insights are sufficiently accurate, and potentially that these insights do not come with stringent privacy measures (for example imposed by legal laws). The latter could be the case when certain data subjects can be identified from the analysis, such linkability/identifiability is ideally not possible and a previously listed requirement of the data provider, imposed on the ML party.

4) Data subject requirements (R_{DS}): In our use cases, the data may or may not be subject to the GDPR legal framework as in most cases the data is related to organizations or institutions, and not individuals. An edge-case in this regard is the situation where a company is a one-man company, in which case data related to that organization is considered personal data [7]. In cases where personal data in involved, sufficient data anonymization should be applied in order avoid re-identification.

III. Architectural designs for privacy-preserving data sharing

We present two architectural solutions to meet with the general use case requirements outlined in the previous section. The driving factor to choose between both architectures is

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-244-9

mainly the degree of willingness to share data by the data provider, and the individual architectural properties. Section III-A presents an architecture in which privacy-preserving tactics can be applied at the data provider-side before it reaches the ML party. Alternatively when dealing with many data providers, these privacy-preserving tactics can be applied once more and intermittently by a mediator. Section III-B details an architecture in which no large data sets are shared between data provider and the ML party. Instead, the ML party tests its predictions at the data provider, which only provides data in the form of minimal feedback to correct the ML model. We will detail each architectural design in-detail, followed by a discussion of their properties and respective trade-offs.

A. Architecture 1 – Sharing privacy-enhanced data

The first architecture to accommodate data sharing between data provider(s) and the ML party relies on statistical anonymization strategies. In this approach, the data is sent from the data provider(s) to the ML party. However, before transferring the data, it is anonymized by the data provider.

1) Core approach: Figure 2.A presents the situation in which one data provider shares data with the ML party. A client-side module (which can be provided by the ML party) locally performs de-identification operations on the data. These operations can include data scrubbing (deleting vulnerable records/attributes), pseudonymization [8] (replacing identifying attributes with a pseudonym), generalization (applying privacy metrics such as k-anonymity [5]) and noise addition. The required operations strongly depend on both the nature of data and the intended use case. Hence, for each case specific settings are required in the client-side privacy module. These settings require considerable insights in the data and the attacker model, and are therefore ideally created in collaboration between the data provider and a privacy expert party (e.g., which may be coincidentally be the ML party).

The major limitation of this approach is that it is only applicable with suitable datasets. Data collected by companies is often sparse or incomplete. In order to enable meaningful data de-identification it is desirable to first complete and enrich the data using external data sources. This data enrichment step in Figure 2.A(1.b) is a second task of the client-side module. Examples of data enrichment are replacing a social security number of individuals by a range of quasi-identifying attributes (e.g., date of birth, residence location, gender) or replacing the VAT-number of a company by relevant information (e.g., location, amount of employees, profit margins). The required data can be collected from either public and private sources, or provided to the client-side module by the ML party.

The first approach presented here allows the data owner to remove personal identifying information (and hence to share personal data in a privacy-friendly and GDPR compliant manner). It can also ensure that certain sensitive aspects of the data are also generalized away, although the input and output of sensitive data should be carefully curated by the data owner to ensure confidentiality. The main downside of this configuration is that the machine learning party can directly link the data to the data provider as it is originating from this source. Our next alternative on this architecture in Figure 2.B introduces a mediator to also further generalize between multiple data providers, but also to hide which data originates from which party.

2) Advanced approach with mediator: There are two major disadvantages of the core approach, the first is that data originating from the data provider is directly linkable to the data source origin. In addition, when multiple data providers are present, further data generalization and privacy-preserving tactics can be applied on the combination of these data sets.

These issues can be solved by introducing mediator in the system, such as a TTP (trusted third party). Figure 2.B presents the approach that includes a mediator, and as example we will assume the use of a TTP. In this setup, all data providers send the data to the TTP. This data can be provided to the TTP in a (partly) privacy-enhanced (e.g., de-identified) state. The TTP collects the data from the different sources, after which the records from the different data providers are merged and mixed. Next, the TTP performs a de-identification step on the data to ensure that the confidentiality of the data and/or the privacy of the data subjects are preserved.

The addition of a TTP allows a decoupling of the data from the data owner. In addition, it can provide an additional de-identification step, and when this is performed correctly and sufficiently, the ML party should be unable to link a record back to the correct data owner. However, this approach is based on the important assumption that the TTP functions correctly, behaves honestly, and does not collude with the ML party. Collusions with the ML party could allow the ML party to link records to one specific business, as would be the case for the core approach without mediator. The trust required by the data provider shifts from the ML party to the TTP. As the data still leaves the premise of the data provider, he might still be reluctant to allow this. A high degree of auditability could offer a solution in this regard. In this setup, the TTP could also be responsible for the data enrichment step (as described in the core approach). This is often even more desirable for the ML party, as the data used for enrichment (which can be intellectual property of the ML party) no longer needs be shared with the data provider.

In the architecture we just presented, proposed a method to share data after applying privacy-tactics at a client-side privacy module, and possibly additionally again at a privacy module located in an intermediate mediator. In these approaches, data is effectively still charged, and even after the deliberate steps and transformations to preserve privacy, this may still be undesirable by parties that are highly reluctant to share data. In our next architecture, we propose a method to improve the ML party's models without sharing data.

B. Architecture 2 – ML feedback-loop validation interface

The second architecture relies on cryptography in combination with statistics. The main advantage of this approach is that the data provider is no longer required to share any data with the ML party. This approach assumes that the ML party

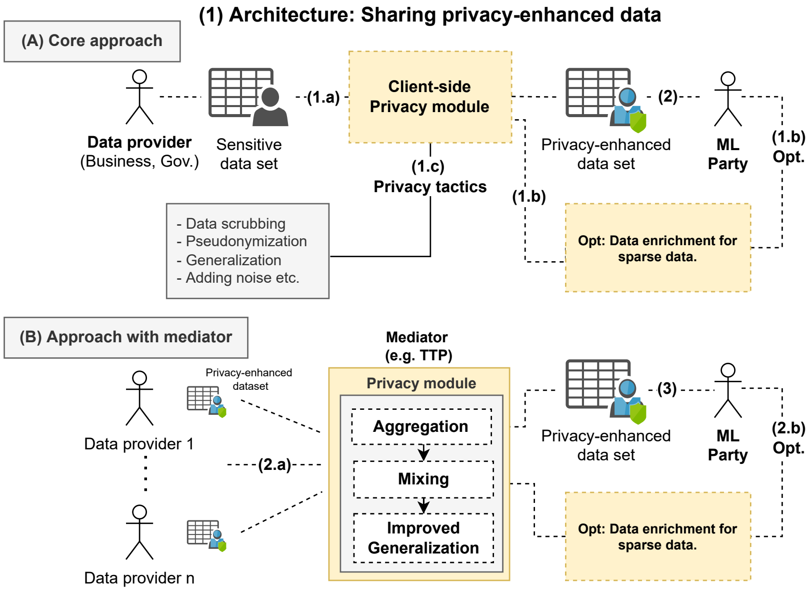

Figure 2. Architecture 1: Privacy-enhancing tactics on (non- or personal) sensitive data.

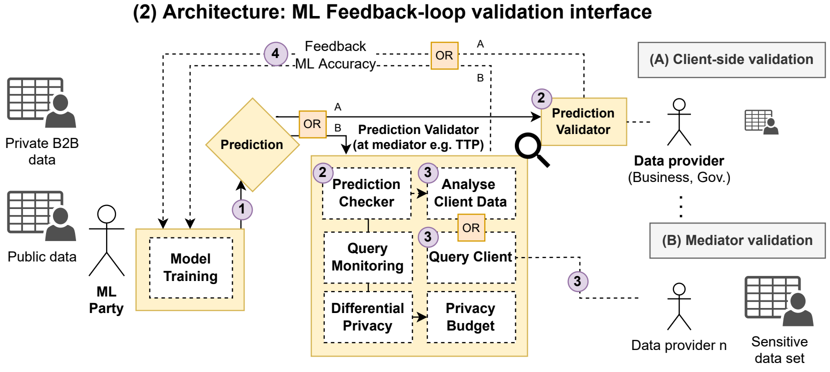

Figure 3. Architecture 2: Feedback-loop for validating or improving ML accuracy without sharing data.

is able to create a preliminary ML model. This model can be created based on a small amount of data retrieved from a data provider (under NDA or by applying Architecture 1) or by using publicly available data.

1) Core approach with the prediction validator module at the data provider: In this approach, the ML party provides a software module to the data provider(s), named the *prediction validator*. This module runs on the premise of the data provider and has access to the sensitive data of the data provider. The data itself never leaves the premise of the data provider. Next, the ML party makes predictions using its preliminary basic ML model, and sends these predictions to the prediction validator module at the data provider. Here, the validator compares the ML party's predictions to the sensitive data records. Based on this analysis, the validator is able to generate the feedback the ML party requires to enhance the ML model.

The kind of feedback and the level of granularity of the feedback depend on the type and sensitivity of the data involved. It is of major importance that the feedback does not leak the sensitive data of the data provider. Therefore it is not possible for the validator to provide the ML party with feedback about individual predictions. However, it is for example possible to give a general accuracy score about groups of predictions. For example, the ML party can group its predictions in confidence intervals, which allows the mediator module to give feedback about a group of predictions. Furthermore, this approach could also support typical machine learning metrics such as recall, precision and F1-score [9].

It is important to be aware not to disclose any sensitive information of the actual data set. By executing consecutive queries, the ML party could attempt to learn sensitive information. For example, if in two consecutive queries, a set of predictions is validated, but one query leaves out the prediction of one record, the ML party could easily learn the details of that one record. Query monitoring [10] prevents this type of unwanted behaviour. Moreover, by applying differential privacy [11], [12] to the output of the validator (adding noise), an additional layer of protection against data leakage is

Courtesy of IARIA Board and IARIA Press. Original source: ThinkMind Digital Library https://www.thinkmind.org

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-244-9

introduced, protecting against information leakage even if the ML party has (partial) knowledge about the sensitive dataset.

The presented approach requires the ML party to share its predictions with the data provider in order to retrieve feedback on these predictions. In an early phase of the training, these predictions may still be very immature and may lead to reputation damage for the ML party as the predictions may be potentially very inaccurate. Furthermore, these intermittent predictions may reveal part of the inner workings of the ML party's (protected) model, which may be subject to IP rights. This may undermine the ML its competitiveness if the data provider is able to steal (part of) the ML party's model. As a solution, we suggest an alternative approach with a validator module located at a mediator to avoid the need for the ML party to directly share its predictions with the data provider.

2) Approach with the prediction validator module in a trusted mediator: One of the main disadvantages of the core approach is that the ML party leaks its (preliminary) predictions to the data provider. This is especially important as both the ML parties in our industry cases listed this as a major concern. In order to avoid data leakage from the ML party to the data provider, the extended version of this approach moves the prediction validator module to a trusted mediator. Hence, the predictions of the ML party are no longer sent to the data provider. The mediator's main task is to compare the predictions of the ML party to the (sensitive) data of the data provider. In order to prevent the mediator from gaining access to the sensitive parts of the data, the ML party and the data provider can apply symmetric encryption to certain attributes before transmission. To achieve this, the data provider and the ML party must first exchange a shared secret key, of which the trusted third party has no knowledge. Comparisons can be made on the encrypted data without the need of additional technologies However, homomorphic encryption – a technique enabling (basic) operations on encrypted data [13] – can also support more fine-grained analyses of the predictions. By applying this approach, the trust required in the mediator is significantly reduced compared to Architecture 1. The only assumption that needs to be made is that the mediator operates honestly. Honest means that it does not tamper with the analysis results for the ML party, and that it does not share the input retrieved from one data provider with another, or data from a data provider to the ML party or vice versa.

IV. Architectural trade-off analysis

In this section we provide a comparison via a trade-off analysis on the properties and merits of both architectures in terms of approach, complexity (e.g., implementation effort), but also requirements on trust between all stakeholders. Specifically, we also detail which technologies, and for example privacy techniques, can be concretely applied towards the implementation of these proposed architectures.

A. Architectural comparison

Table I lists an extensive comparison between architectures 1 and 2, both when or when not using a mediator. This comparison is conducted in terms of the factual shared data, the required quality of shared data, the implementation effort for each architecture, and overall complexity. In addition, we detail the mediator's task in each architecture, as well as the overall followed approach, and eventual consequences for data linkability. The main goal of the proposed architectures includes that individual subjects cannot be identified from the data set. We discuss each aspect from Table I in turn.

1) General approach: In architecture 1, we privacy-enhance and actually share data with the ML party. In the second architecture, we do not directly share data, but instead provide feedback information (i.e., validation) on the ML party its trained model, namely on its prediction accuracy.

2) Optional mediator's involvement and task: In the first architecture, optionally, a mediator is involved to further generalize from multiple data providers', and consequentially hide the data source from the ML party. In the second architecture, the mediator is involved to either hide the data provider which validates the forwarded predictions by the mediator. Alternatively, the data provider can also entrust the data set to the mediator, which then assumes the responsibilities to provide feedback on ML predictions. The mediator has to however ensure that subsequent feedback responses do not reveal anything about the original data set, via techniques such as differential privacy and query monitoring.

3) Data linkability: In terms of data linkabilty, depending on the choice for the first architecture and the involvement of a mediator, either the ML party can directly attribute a certain data set to a certain provider, or the mediator is able to do this and hides such information from the ML party. In the second architecture, predictions cannot be linked to a concrete data set, only the feedback to a certain mediator or data provider.

4) Shared data and quality: The main distinction between both architectures is the willingness to share data, and architecture 1 is ideally suited for sensitive data which can be privacy-enhanced and still shared, whereas in architecture 2 only feedback is given on the ML model its accuracy. In the first case, we therefore need sufficient quantity and diversity of data as to enable the application of such privacy tactics.

5) Implementation effort and complexity: In terms of implementation, architecture 1 can make use of readily-available software libraries featuring privacy tactics, such as the ARX library [14]. The only difficult aspect is that the chosen tactics have to be carefully considered regarding their suitability on the involved data set, and their respective impact on privacy threats and remaining data utility. In contrast, the second architecture is more visionary, and requires careful consideration on how to provide feedback or validation of the ML's predictions, of which optionally this feedback can steer the training in a positive manner. In addition, this feedback should not reveal anything about the data provider's data set, which may require differential privacy and query monitoring, which consider previously released queries' and their respective feedback.

TABLE I. CHARACTERISTICS AND PROPERTIES OF THE ARCHITECTURES WITH- OR WITHOUT MEDIATION.

	Architecture 1		**Architecture 2**	
Shared data	Data sets which are privacy-enhanced (e.g., de-identification, generalization of attributes).		No datasets shared, only minimal feedback regarding the accuracy of the ML model.	
Data quality required	Needs sufficient data (e.g., minimum number of rows) to privacy-enhance data (e.g., generalization).		Doesn't need directly shared data. Feedback is given to validate the ML model, and potentially improve it.	
Implementation effort	Standard libraries available to apply the privacy tactics, such as the ARX library [14].		No ready-made available libraries/frameworks for such an approach.	
Complexity	Hand tailored selection of privacy tactics per use case.		Complex process to determine how to structure valuable and privacy-preserving feedback.	
Mediator's task	Generalization, de-identification, and other tactics of multiple already privacy-enhanced data sets.		Responding with feedback, querying or optionally collecting data from/to data providers.	
	No mediator (1.A)	**With mediator (1.B)**	**No mediator (2.A)**	**With mediator (2.B)**
Approach	Data is privacy-enhanced at the data provider and then sent to ML.	After privacy-enhancing at the data provider, additional generalization at mediator.	Data is not directly shared, but the prediction accuracy of the ML model is validated at the client side.	Data not shared directly, ML model validated at the mediator (entrusted the data or has to query client).
Data linkability	Data originates directly from the data provider, and is linkable to the source.	Removing direct link to origin of data per DP.	Predictions cannot be linked to a certain concrete data set.	Client can maintain its own data set, or share it with mediator.

B. Trust model analysis

Table II elicits the trust assumptions, which are generally minor, for all involved stakeholders, namely data provider, ML party, and optionally a mediator.

Data provider: Regarding the data provider, we assume that this provider acts honestly and shares a correct (privacy-enhanced) data, or in the case of the second architecture provides honest feedback on the basis of this data. In theory, this should be in the interest of the data provider himself, as he will typically rely on the insights gathered by the ML party, which is a win-win for both actors. In turn, the data provider could possibly attempt to reverse engineer the ML model based on the insights, although this could prove technically challenging, and is therefore a weak assumption. These insights are more valuable in the second architecture, which are presented intermediately, and the process of model learning could be more evident.

ML party: The trust assumptions placed in the ML party are more of a minor nature, as in the first architecture the data that arrives is already privacy-enhanced. Yet, potentially we could expect the ML party to not further disclose this privacy-enhanced data set. We expect the ML party to share honest insights gathered, but this could be facilitated or verified by the data provider on the basis of real world scenarios, or applicability in its own business processes. In the second architecture, no concrete data is shared, but insights which out of self-interest are ideally honest.

Mediator: As a mediator, many of the trust assumptions of the data provider and ML party are partially inherited. For example, we assume it forwards the original privacy-enhanced data set in architecture 1, or the corresponding feedback on the ML party its predictions when querying the client. Alternatively, when the mediator is entrusted the data set in architecture 2, we also assume it does not disclose this data (which may be a stringent requirement, although when sensitive we also do not expect the ML party to do this). In addition, in this case we assume a correct handling of the predictions. Furthermore, we also expect the mediator to hide the data source, more specifically the data providers involved.

1) Meta-data encryption: In our architecture, and when it is opted for a mediator, and specifically in the case of a trusted third party, we assume that this TTP is honest-but-curious. In Architecture 1, the data which arrives at the mediator is already privacy-enhanced, and is ideally further aggregated. The trust at this stage, is therefore mainly in the correct application of this method and the forwarding. In Architecture 2, the operations applied by the mediator are more complex, and he can have insight into the predictions passed by the ML party, as well as verifying these predictions by a query to the data provider (or when trusted against the data set provided to the mediator). In order to hide the insights that the mediator can gain into the process, both ML party and data provider can agree on a shared key to encrypt meta-data before sending it over the mediator. This will enable part of the task of responding to the prediction query by the data provider, or in reading part of the feedback by the ML party.

2) Trusted execution environments: Such encryption can not always be applied however, as the mediator may have to be actively involved in assessing whether the prediction is correct, and involved in the feedback process. Therefore such key values may have to be in readable format. In order to prevent the trusted mediator, which is assumed to be honest-but-curious, from gaining such insights, and to actually also relax this trust assumption we can integrate trusted executions

TABLE II. TRUST ASSUMPTIONS OF THE ARCHITECTURES WITH- OR WITHOUT MEDIATION.

	Architecture 1		**Architecture 2**	
Trust in	**No mediator (1.A)**	**With mediator (1.B)**	**No mediator (2.A)**	**With mediator (2.B)**
Data provider	Shares correct data set (win-win for insights), least possible noise. No reverse engineering of ML insights.		Provides only honest feedback. Keeps intermediate insights confidential.	Provides honest feedback and keeps intermediate insights confidential, or trusts true data set to mediator.
ML party	No data disclosure. Shares truthful insights.		Provides truthful insights out of self-interest.	
	With mediator (1.B)		**With mediator (2.B)**	
Trust required in mediator	Shares correct privacy-enhanced data, and only to ML party. Hides source of the data.		Mediator passes correct insights and feedback, or optionally keeps data confidential and provides feedback.	

environments. Trusted execution environments provide a secure tamper-resistant execution environment, isolated from – in this case – the rest of the mediator's own platform [15]. For example, Intel SGX [16] could be used to execute certain of the mediator's its functionalities. Subsequently, the data provider its data can be sent to this module encrypted, and will only be readable to the trusted execution environment.

C. Architectural selection process

The selection of one of the presented architectures for a specific use case is influenced by multiple factors, namely the quality and the nature of the data, the willingness of the data provider to share the data with the ML party and the willingness of the ML party to leak information about the ML model with the data provider. A first distinction is made between whether the data concerns sensitive company data, as this is a driving factor for the willingness of the data provider to share the data with the ML party. If the data is not sensitive, and does not contain personal data, no privacy enhancing tactics are required. If the dataset contains personal data that is not sensitive to the company, Architecture 1.A is advised when the data is suitable for anonymization (by default or after enrichment). Alternatively, when data anonymization techniques are not feasible, a variant of Architecture 2 is required. In the scenario where the data provider is reluctant to share the data with the ML party, Architecture 1.B can be applied if multiple data providers are available and the link between the data providers and their respective data can be severed by mixing (and anonymizing) data from multiple data providers. When this is not possible, a variant of Architecture 2 is advised depending on whether or not the ML party requires model protection.

V. RELATED WORK

Our work is situated within the domains of classical privacy-preserving techniques, data anonymization for ML purposes, and collaboration strategies in this context such as federated learning.

Data anonymization strategies: Many well-known data anonymization strategies are described and evaluated in literature. Privacy metrics such as k-anonymity [5] and its derivatives such as l-diversity [6] and t-closeness [17] are extensively studied, in the context of privacy [18], [19] as well as the theoretical [20] and the practical utility [21], [22]. Moreover, they are readily available in tools such as ARX [14]. Many real-life use cases have benefited from these types of metrics. For example, Jakob et. al. [23] described an anonymization pipeline to aid the gathering of medical data for research during COVID.

Anonymized data applied in ML: The applicability of anonymized data in machine learning applications specifically has also already been discussed by several papers. For example, Slijepcevic et. al. [24] and Carvalho et. al. [25] investigate the effect of applying metrics such as k-anonymity on the classification performance. Both works argue that it is hard to exactly predict the impact of privacy preserving operations on the accuracy of ML models, but find that the effects are manageable if the anonymization operations are not too harsh. The aforementioned data anonymization strategies are applied as part of the solution in Architecture 1, but require additional components in order to fulfill the trust requirements related to the sensitive nature of the data.

Protecting machine learning models: In the context of this paper, two important attack vectors on machine learning models should be considered. First of all, many papers [26], [27] have demonstrated that machine learning algorithms are often prone to leak data used in the training set. Two popular types of attacks are membership inference [28], [29] and attribute inference [30], [31]. Defenses against these types of attacks are proposed [32], [33] and should be considered in both architectures presented in this paper.

A second threat to ML models that is relevant in the context of this work is model stealing [34], [35]. As the machine learning model is intellectual property (and the core business incentive) of the ML party, the model should be protected against such theft. Several defenses have been proposed [36], [37], and should be implemented by the ML party.

Privacy-preserving querying: Within the context of privacy preserving data sharing, the concept of differential privacy [11], [12] is currently often presented as a one-size-fits-all solution. In contrary to the aforementioned data anonymization techniques, differential privacy is not a property of a dataset but of a function. Therefore, differential privacy is not directly applicable for the data sharing part of our industry use cases, as they require the actual records and not aggregates over multiple records. Differential privacy is however applicable in Architecture 2 in the mediator module, as it can prevent data leakage in the feedback to the ML party.

Alternative privacy preserving data collaboration strategies: In the realm of machine learning, federated learning strategies [38] are being proposed, allowing companies to collaborate towards a common machine learning model without the need to contribute their own data in one shared data pool. For example, Dayan et. al.[39] demonstrate the advantages of federated learning to create data collaborations in the context of a large COVID-19 clinical study across multiple countries and health institutions. Tools and frameworks such as Flower [40] and Sherpa.ai [41] support developers in implementing such strategies. However, it should also be noted that successful attacks have been performed on federated learning models before [42]. The business driver in our industry cases is the ML party, whose incentive is the financial benefit from commercializing the created machine learning models. Applying federated learning in our industry use cases would cut the ML party (and therefore the technology enabler) from the equation. Additionally, a federated learning approach would also rely on the data providers to set up such collaborations (and processing infrastructure) among themselves. Such solutions are therefore undesirable and unfeasible in our industry use cases. Another stream in privacy preserving data collaborations is found in the realm of cryptography, where techniques such as fully homomorphic encryption (FHE) [13] and secure multi party computation (SMPC) [43] have gained traction. FHE allows computations to be executed on encrypted data. In this work, FHE can be applied as one of the building blocks in Architecture 2 in order to support more complex model validation in the mediator module and to enhance the feedback towards the ML party. Note that the set of available operations on encrypted data is still rather limited. SMPC is a cryptographic protocol that allows multiple parties to contribute to a common computation without the need to show their data to the other parties. However, MPC is very resource intensive, and therefore do not scale well in in larger and more complex applications. The latter, in combination with the required domain knowledge to build such systems makes MPC unfeasible in our industry cases.

VI. Conclusions

The research which we presented in this paper is motivated by two use cases from industry partners, respectively in the context of B2B and G2B data sharing for ML purposes. The problem which we identified is that there are two major hindrances towards data-driven intelligence gathering, namely a lack of in-house ML knowledge, and insufficient data to enrich existing data sets and enabling the extraction meaningful insights. As a solution, a third-party ML expert with the necessary expertise is often brought in, which can gather additional data from public or private sources when required. However, the involvement of a third party ML party introduces its own challenges, namely that business or governmental entities now must trust these external parties with their data.

In this paper, we provide technological solutions in the form of privacy-preserving data sharing architectures to alleviate or reduce the stringent requirement of trust in a third party. We outline two architectures, depending on the degree of willingness by the data provider to share data, which is typically dictated by the sensitivity of the data involved. The first architecture involves readily-available privacy-enhancing techniques (e.g., generalization, de-identification) at the data provider-side before sharing datasets to the ML party. Optionally, a mediator can be involved such as a trusted third party to hide the source of the data set, as well as to further aggregate, mix, and generalize data sets when they originate from multiple data providers.

A second architecture is designed for situations where a data provider is highly reluctant to share data (e.g., in the case of highly sensitive data). In this case, an interface allows the ML party to present predictions from an established baseline ML model to the data provider. These predictions can be validated or used to provide feedback for improving the analytical model and deriving insights. It is crucial that even in such a case, the feedback presented does not leak any information on the original data set, which can be facilitated by means such as differential privacy. Similarly, a mediator can be involved to assume such responsibilities for a multitude of data providers.

We presented a trade-off analysis on both architectures in terms of their approach, the type of required data, and shared data, as well as the required complexity and implementation effort. The first architecture is highly feasible, although requires specific tailoring of the required privacy tactics on a per-use case basis as it is highly dependent on the quality and type of data provided. The second architecture is more visionary in its nature, with many future technological challenges that can enable validation of ML models, as well the ability to provide useful feedback to steer and improve an ML process without information leakage. This architecture provides a way to meet many of the legal requirements such as GDPR and other current and future responsibilities related to data ownership.

References

[1] The Brussels Times, *Flanders cracks down on social housing fraud*, https://www.brusselstimes.com/news/belgium-all-news/161102/flanders-cracks-down-on-social-housing-fraud, [Online; accessed 18-Aug-2022], 2021.

[2] Towards Data Science, *Smart Policing for Safer Cities: A Data-Driven Approach*, https://towardsdatascience.com/smart-policing-for-safer-cities-a-data-driven-approach-ed84e801526f, [Online; accessed 20-Oct-2022], 2020.

[3] Towards Data Science, *Predictive analytics in government decisions*, https://towardsdatascience.com/predictive-analytics-in-government-decisions-8128ba019a77, [Online; accessed 20-Oct-2022], 2019.

[4] EUR-Lex, *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016*, https://eur-lex.europa.eu/eli/reg/2016/679/oj, [Online; accessed 20-Oct-2022], 2016.

[5] L. Sweeney, “K-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[6] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “L-diversity: Privacy beyond k-anonymity,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, 3–es, 2007.

[7] European Commission, *Do the data protection rules apply to data about a company?* https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/application-regulation/do-data-protection-rules-apply-data-about-company_en, [Online; accessed 20-Oct-2022], 2017.

[8] H. Ko, “Pseudonymization of healthcare data in south korea,” *Nature Medicine*, vol. 28, no. 1, pp. 15–16, 2022.

[9] R. Yacouby and D. Axman, “Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models,” in *Proceedings of the first workshop on evaluation and comparison of NLP systems*, 2020, pp. 79–91.

[10] A. Kumar, J. Ligatti, and Y.-C. Tu, “Query monitoring and analysis for database privacy-a security automata model approach,” in *International Conference on Web Information Systems Engineering*, Springer, 2015, pp. 458–472.

[11] C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014, ISSN: 1551-305X. DOI: 10.1561/0400000042.

[12] D. Desfontaines and B. Pejó, “Sok: Differential privacies,” *Proceedings on privacy enhancing technologies*, vol. 2020, no. 2, pp. 288–313, 2020.

[13] P. Martins, L. Sousa, and A. Mariano, “A survey on fully homomorphic encryption: An engineering perspective,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–33, 2017.

[14] F. Prasser, J. Eicher, H. Spengler, R. Bild, and K. A. Kuhn, “Flexible data anonymization using arx—current status and challenges ahead,” *Software: Practice and Experience*, vol. 50, no. 7, pp. 1277–1304, 2020.

[15] M. Sabt, M. Achemlal, and A. Bouabdallah, “Trusted execution environment: What it is, and what it is not,” in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, 2015, pp. 57–64. DOI: 10.1109/Trustcom.2015.357.

[16] V. Costan and S. Devadas, “Intel sgx explained,” *Cryptology ePrint Archive*, 2016.

[17] N. Li, T. Li, and S. Venkatasubramanian, “T-closeness: Privacy beyond k-anonymity and l-diversity,” in *2007 IEEE 23rd international conference on data engineering*, IEEE, 2006, pp. 106–115.

[18] A. Zigomitros, F. Casino, A. Solanas, and C. Patsakis, “A survey on privacy properties for data publishing of relational data,” *IEEE Access*, vol. 8, pp. 51 071–51 099, 2020.

[19] G. D’Acquisto *et al.*, “Privacy by design in big data: An overview of privacy enhancing technologies in the era of big data analytics,” 2015.

[20] M. M. Almasi, T. R. Siddiqui, N. Mohammed, and H. Hemmati, “The risk-utility tradeoff for data privacy models,” in *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, IEEE, 2016, pp. 1–5.

[21] K. De Boeck, J. Verdonck, M. Willocx, J. Lapon, and V. Naessens, “Dataset anonymization with purpose: A resource allocation use case,” in *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, IEEE, 2021, pp. 202–210.

[22] R. Hoogervorst, Y. Zhang, G. Tillem, Z. Erkin, and S. Verwer, “Solving bin-packing problems under privacy preservation: Possibilities and trade-offs,” *Information Sciences*, vol. 500, pp. 203–216, 2019.

[23] C. E. Jakob, F. Kohlmayer, T. Meurers, J. J. Vehreschild, and F. Prasser, “Design and evaluation of a data anonymization pipeline to promote open science on covid-19,” *Scientific data*, vol. 7, no. 1, pp. 1–10, 2020.

[24] D. Slijepčević *et al.*, “K-anonymity in practice: How generalisation and suppression affect machine learning classifiers,” *Computers & Security*, vol. 111, p. 102 488, 2021.

[25] T. Carvalho and N. Moniz, “The compromise of data privacy in predictive performance,” in *International Symposium on Intelligent Data Analysis*, Springer, 2021, pp. 426–438.

[26] B. Liu *et al.*, “When machine learning meets privacy: A survey and outlook,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–36, 2021.

[27] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, “Privacy risk in machine learning: Analyzing the connection to overfitting,” in *2018 IEEE 31st computer security foundations symposium (CSF)*, IEEE, 2018, pp. 268–282.

[28] L. Song and P. Mittal, “Systematic evaluation of privacy risks of machine learning models,” in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2615–2632.

[29] H. Hu *et al.*, “Membership inference attacks on machine learning: A survey,” *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[30] B. Z. H. Zhao *et al.*, “On the (in) feasibility of attribute inference attacks on machine learning models,” in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2021, pp. 232–251.

[31] J. Jia, B. Wang, L. Zhang, and N. Z. Gong, “Attriinfer: Inferring user attributes in online social networks using markov random fields,” in *Proceedings of the 26th International Conference on World Wide Web*, 2017, pp. 1561–1569.

[32] J. Jia and N. Z. Gong, “{Attriguard}: A practical defense against attribute inference attacks via adversarial machine learning,” in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 513–529.

[33] J. Chen, W. H. Wang, and X. Shi, “Differential privacy protection against membership inference attack on machine learning for genomic data,” in *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, World Scientific, 2020, pp. 26–37.

[34] B. Wang and N. Z. Gong, “Stealing hyperparameters in machine learning,” in *2018 IEEE symposium on security and privacy (SP)*, IEEE, 2018, pp. 36–52.

[35] H. Yu *et al.*, “Cloudleak: Large-scale deep learning models stealing through adversarial examples.,” in *NDSS*, 2020.

[36] T. Lee, B. Edwards, I. Molloy, and D. Su, “Defending against neural network model stealing attacks using deceptive perturbations,” in *2019 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2019, pp. 43–49.

[37] M. Juuti, S. Szyller, S. Marchal, and N. Asokan, “Prada: Protecting against dnn model stealing attacks,” in *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2019, pp. 512–527.

[38] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, “Federated learning: Challenges, methods, and future directions,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[39] I. Dayan *et al.*, “Federated learning for predicting clinical outcomes in patients with covid-19,” *Nature medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[40] D. J. Beutel *et al.*, “Flower: A friendly federated learning research framework,” *arXiv preprint arXiv:2007.14390*, 2020.

[41] N. Rodríguez-Barroso *et al.*, “Federated learning and differential privacy: Software tools analysis, the sherpa. ai fl framework and methodological guidelines for preserving data privacy,” *Information Fusion*, vol. 64, pp. 270–292, 2020.

[42] Z. Sun, P. Kairouz, A. T. Suresh, and H. B. McMahan, “Can you really backdoor federated learning?” *arXiv preprint arXiv:1911.07963*, 2019.

[43] Y. Lindell, “Secure multiparty computation,” *Communications of the ACM*, vol. 64, no. 1, pp. 86–96, 2020.