Interaction Patterns for Regulatory Compliance in Federated Learning

Mahdi Sellami[1], Tomas Bueno Momčilović[1], Peter Kuhn[1] and Dian Balta[1]

[1] *fortiss GmbH, Guerickestraße 25, Munich, Germany*

Abstract

Organizations in highly regulated domains often struggle to build well-performing machine learning (ML) models due to restrictions from data protection regulation. Federated learning (FL) has recently been introduced as a potential remedy, whereby organizations share local models while keeping data on premise. Still, regulatory compliance remains challenging in FL settings: training data needs to be shared to some extent, and models can be reverse engineered or misused towards violation of data privacy by each participating organization. Guided by design science methodology, we introduce four interaction patterns that allow for compliance-by-design and trust-context-sensitive analysis of an FL system by combining different approaches to privacy preservation. We match the patterns to privacy principles and exemplify how verifiable claims about compliance at design- and operation-time FL can be generated to make all participating organizations accountable.

Keywords

Federated Learning, Privacy, Compliance, Design Patterns

1. Introduction

Organizations in highly regulated domains, such as the government, health or banking, explore applications of machine learning (ML) with high intensity to gain efficiency, effectiveness, and a competitive edge (cf. e.g., [1-2]). Unfortunately, they often struggle with having a sufficient amount of data [3]. This insufficiency of data is often the leading cause of underperforming models of ML [4], thereby undermining the value proposition.

One remedy is to federate the learning process between different organizations and their data, in order to train a common (and higher quality) model that benefits the knowledge of all parties involved (cf. e.g., [4]). A prerequisite for such an approach is that the parties have to share some of the data or at least model training parameters. This is not an easy task, given strong regulatory constraints resulting from e.g., GDPR, HIPAA, and similar (cf. e.g., [5]). Implementing federated learning (FL) would mean that they have to be compliant, i.e. that 1) they have a corresponding, suitable design that complies with regulation; and 2) they operate according to it. While compliance needs to be upheld throughout the whole process of setting up the federation, model training and processing, an approach is required to design corresponding information systems (IS) that hold every participating organization accountable to preservation of privacy in every stage of ML.

In this paper, we address the following question: *How to design an architecture for accountable privacy-preserving data sharing in the entire ML process?* We propose **interaction patterns** for the architecture design of IS involved in inter-organizational ML with a focus on the level of trust between organizations. In terms of implications, the patterns lay the ground for (1) an academic discussion in mapping legal regulation requirements to technology, and (2) a practical guideline

CIISR 2023: 3rd International Workshop on Current Information Security and Compliance Issues in Information Systems Research, co-located with the 18th International Conference on Wirtschaftsinformatik (WI 2023), September 18, 2023, Paderborn, Germany

sellami@fortiss.org (M. Sellami); momcilovic@fortiss.org (T. Bueno Momčilović); kuhn@fortiss.org (P. Kuhn); balta@fortiss.org (D. Balta)

0000-0002-2817-2643 (M. Sellami); 0000-0003-4503-2244 (T. Bueno Momčilović); 0000-0001-6774-2904 (P. Kuhn); 0000-0001-8311-3227 (D. Balta)

for designing such systems. In order to evaluate the patterns, we present a prototypical implementation and discuss how one can make and prove claims of what was designed and promised.

2. Background

2.1. Federated Learning

Federated learning (FL) is a novel machine learning (ML) method that allows a set of distributed parties to jointly train a shared model while keeping their data on premise. The FL process involves an aggregator who provides specifications for applying an ML model, sends them to the parties to train the model on their own private data, and then combines the information from many different local models using an algorithm for aggregating training parameters [4]. The most common approach relies on a client-server or star network [6] comprising the Federated Averaging algorithm proposed by [4], where a central server orchestrates the contributions of clients (participants such as organizations or edge devices).

FL system designs vary in six main aspects [7]: 1. data partitioning (horizontal vs. vertical), 2. ML model, 3. scale of federation (cross-silo vs. cross-device), 4. communication architecture (centralized vs. decentralized), 5. privacy mechanism, and 6. motivation. To illustrate our results, we limit our scope to: 1. horizontal data partitioning; 2. neural network; 3. cross-silo scale; 4. centralized communication; 5. no privacy mechanism beyond the learning pipeline; and 6. a motivation for federation primarily driven by the need to comply with data protection regulation. Furthermore, we consider that the FL process contains four stages [8]: **setup** to specify the data, purpose of collaboration, and the ML pipeline; **preprocessing** to prepare the training data of each participant; **processing**, to optimize the model iteratively; and optional **postprocessing** for applying a privacy-enhancing technique, mitigate bias and similar.

With regard to compliance, **privacy mechanisms** are of particular importance. Even though each party's data remains on-premise during training, attackers may still extract sensitive information from the exchanged model updates. Several attacks including membership inference attacks [9] and model inversion attacks [10] can lead to data leakage. Thus, different privacy techniques (e.g., differential privacy [11]; cryptographic methods [12]) and trusted execution environments [13] can be independently combined with FL to provide stronger privacy guarantees. The **motivation** for FL is also an important compliance requirement in real-world scenarios, ensuring the involvement of participating parties. This motivation is based on regulations, incentives, or a combination of both. Applying FL within an organization (e.g., governments, companies, etc.) is generally driven by a need to comply with regulation [5].

2.2. Compliance based on accountability and verifiable claims

Interpreting and complying with legal requirements is a resource-intensive problem [14]. Accountability helps to address this challenge in FL setups. Implying different domains and stakeholders [15], accountability has been introduced in the 1960s as an important design principle for systems [16], with different operationalizations emerging throughout the decades since (e.g., "code is law"; [17-18]).

In this work, we consider accountability as a mechanism for enabling trust and legal compliance in FL systems. We adopt a definition that is applicable to distributed systems [19], i.e. "*the transparent assignment and ownership of responsibilities (…) enabling [the distribution of] business goals across multiple organizations*." We tackle accountability from an engineering perspective by introducing verifiable claims to FL systems, aiming to-wards trustworthy AI development [20] by emphasizing the following dimensions [8]:

- **Verifiable**: Every step of the learning process must be documented by specific claims. Each claim should be transparent and supported by evidence, that the corresponding step was conducted correctly with respect to a predefined specification.

- **Undeniable consent**: The executed learning process must be aligned with the expectations of all participants, who must give their explicit consent to the specification of the process. Furthermore, the execution of the process should be non-repudiable and provable.
- **Auditable**: Any deviation from the specification (e.g., system attacks) must be detectable and provable by any third party, based on the recorded claims and their corresponding evidence.
- **Tamper-evident**: All the interactions between the participants must have a corresponding record according to a predefined specification. Furthermore, any intended corruption of the shared knowledge of the participants should be detectable.

2.3. Data Protection and Privacy Principles

Privacy-related regulatory requirements in the EU center almost exclusively on the General Data Protection Regulation (GDPR; [21]), which came into force in 2016 and repealed an earlier framework from 1995. The draft EU Act on Artificial Intelligence [22] directly refers to the GDPR for all privacy-related requirements in AI systems development and deployment (Sec. 3 Sub. 2 Art. 10 Para. 5), and so do national laws (e.g., German Federal Data Protection Act; [23]) that harmonize (i.e., adapt) the stipulations to the national context.

GDPR is concerned with the protection of personal data, i.e., "any information relating to an identified or identifiable natural person" (Art. 4 Paragraph 1). It comprises six core **privacy principles** in Article 5. **Lawfulness, fairness, and transparency** refers to obtaining consent from the data subject and defining the legitimate reason for processing personal data. **Purpose limitation** refers to specifying explicit and unexceedable boundaries for processing data, whereas **data minimization** refers to explicit and unexceedable boundaries for collecting data. **Accuracy** refers to keeping data up-to-date and rectifying deviations, and **storage limitation** refers to specifying an explicit time limit when storing collected data. **Integrity and confidentiality** represent security "using appropriate technical or organizational measures." Finally, the point on **accountability** designates the data controller as responsible for ensuring compliance with the six principles.

These principles in GDPR correspond to a widely accepted set of best privacy practices. Legal scholars [24] provide seven overarching principles, of which six directly map to GDPR: respect for context with purpose limitation; consent, legitimacy and transparency with lawfulness, fairness and transparency; transparency with accuracy; proportionality with data minimization; and accountability with its GDPR counterpart. The unique principle that remains is **privacy by design –** a requirement to address data privacy concerns in the initial design stages and throughout the whole lifecycle of products, processes, and services, which is compatible with the notion of data protection by design in GDPR (Art. 25 paragraph 1, GDPR). Thus, a set of eight principles comprises the privacy requirements from the regulatory standpoint.

3. Research Approach: Pattern-based Design Research

We follow the pattern-based design research (PDR) method [25] to specify reusable patterns for privacy-preserving interactions between at least two parties who want to share knowledge, but not their private data. Patterns are "best practices that are bound to a specific context in which the provided solution has been proven to work" (p. 75, [25]); they represent empirically founded models that are defined in iterations between theory and practice.

Interaction patterns is a term we propose to describe an approach in information systems research for modelling a controlled sequence of information flow between parties with predefined roles (see, e.g., [26-27]). **Interaction** refers to an exchange of information between two or more parties for the purpose of achieving a common goal, whereas **interaction patterns** are design patterns which describe recurring interactions. These interactions correspond to the following simplified scenario of interest. The data processor wants to use the private data of the

data provider for some computation: e.g., data aggregation or pattern recognition. The data provider wants to make sure its private data is protected and correctly handled (e.g., that there is no data leakage).

The process of PDR comprises four stages [25]. First, input is collected from an existing scientific foundation or practical observations. Second, this input is used to generate pattern candidates, their description language, and/or the design theories to support the design activities. Third, pattern candidates are instantiated as tangible solutions to practical problems, and these instances may deviate from each other based on the context they are applied in. Finally, these deviations are evaluated and used as further input to refine existing candidates or define new patterns for the next cycle.

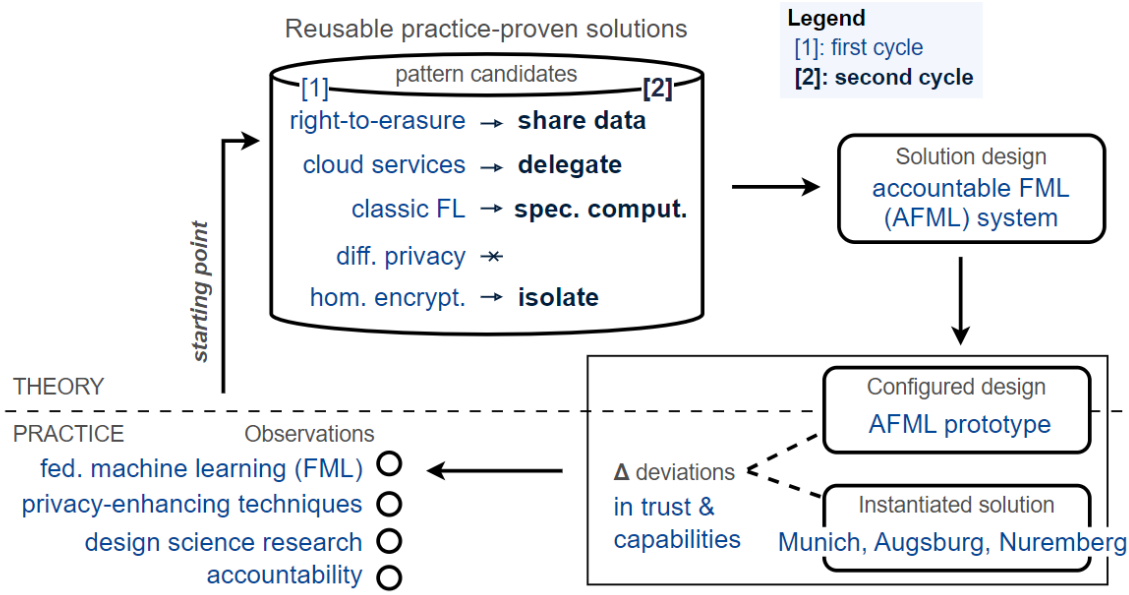

Figure 1. Annotated PDR diagram; adapted from p. 78, Buckl et al. (2013)

We completed an entire cycle of the PDR method, finishing with new pattern candidates (see: Figure 1). We compiled relevant 'observations' from a literature review: FL and classic ML methods [3]; privacy enhancing techniques for ML tasks [5]; design science research, a discipline for studying software architectures [25]; and accountability through verifiable claims [28]. With this input, we specified five initial pattern candidates, which reflect the general solutions that are commonly applied to privacy-oriented problems (e.g., [29]).

1. **Right to erasure**: One party shares its data with another party, under the expectation that the latter will delete the data once the purpose for which it was shared no longer exists (Article 17 paragraph 1, [21]).
2. **(Trusted) Cloud services**: A trusted entity hosts the data, performs the computation for all parties, and then deletes the data [30].
3. **Classic FL**: One party sends the model specifications to another party in the processing stage, and aggregates the results from training [4].
4. **Differential privacy:** Parties hide personal identifiable information with type-appropriate noise that does not affect ML model results [31].
5. **Homomorphic encryption:** Parties hide personal identifiable information by transforming it into ciphertext without affecting ML model results [32].

Using privacy by design (i.e., privacy in all stages; [24]) and accountable FL (i.e., federation that can be verified; IBM, [8]) as general guidelines, we created a blueprint of an "accountable federated machine learning" (AFML) system. With it, we developed a prototype for conducting experiments with three Bavarian municipalities – Munich, Augsburg, and Nuremberg.

Input from the experiments and stakeholders helped us restructure the pattern candidates into four patterns which we present below. First, we excluded differential privacy once we

determined that adding noise does not solve the trust problems between the parties themselves, nor excludes the need for data erasure. Second, we added the precondition of secure communication, as successful man-in-the-middle attacks invalidate any privacy claim. Third, we abstracted away from specific implementations. Cloud services are an instance of a pattern of delegating to a third party, and so is homomorphic encryption an instance of a pattern of isolating information without affecting the results. The classical FL has been extended to also include pre- and post-processing specification of computation. Right-to-erasure has become a data sharing pattern, since according to [21], data providers can share data with other parties (e.g., processors), but must delete and enforce deletion of all relevant instances upon a user's request. In ML scenarios, this can involve running a resource-intensive process to "unlearn" (i.e., retrain) any model to exclude the user's data [41].

We added four dimensions to distinguish patterns: the trust scenario, architectural model, claim, and exemplary application. **Trust scenarios** (similar to threat models; cf. e.g., [5]) describe the trust between the parties, and the challenge that the pattern is addressing. The **architectural model** is a diagram visualized in design science templates (4+1 view; UML), and contains the systems, the actors, and their interactions. **Claims** are short texts describing the promised way of private data handling, which can be linked with verifiable evidence. Finally, the **exemplary application** provides the instance of the pattern in industry or research.

4. Interaction Patterns

4.1. Patterns

In this section, we introduce four interaction patterns as solutions for private data handling issues: share data, specify computation, delegate and isolate. The visual structure in Figure 2 represents the minimal requirements for the interaction to be completed correctly, such as: necessary actors (i.e., the data provider, the data processor and in one case, a third party), technical components (i.e., modules for computation and a database of private data), and sequential actions for the exchange and processing of data. The names of the patterns reflect the overarching process taking place. Additionally, when implemented correctly, each pattern generates a privacy claim against which we verify that the private data indeed remains private, while maintaining the important assumption that parties have established secure communication.

The *share data* pattern relies on the provider trusting the processor to handle its data properly (i.e., behaves in a trustworthy manner). This pattern is widely observed in highly regulated areas like healthcare and government. In fact, since the introduction of the "right to erasure" or the "right to be forgotten" in the GDPR, this pattern has been adopted by a variety of developers whose applications depend on private data, or it is at least offered to the users as an option.

The *specify computation* pattern also relies on trust. Here, the processor trusts the provider (and its computation system) to execute the computation correctly: with the right data, in the prearranged manner, and without tampering of results. This is exemplified in a classical FL scenario. In the *delegate* pattern, by contrast, neither the provider nor the processor trust each other, but they both trust a third party. This party can be any entity which is deemed trustworthy enough to store and handle data securely, and execute computation steps. Put simply, the core parties shift their trust to an intermediary.

Finally, the *isolate* pattern requires the parties to trust the technology instead of one another. Although this can solve concerns related to trust, it is the most complex and resource-intensive approach. For example, homomorphic encryption [32] allows the provider to encrypt its data before sending it to the processor, who then performs the computation on ciphertext (i.e., the encryption space). The resulting output, when decrypted, equals the output of the computation in plaintext (i.e., original space). However, fully homomorphic encryption methods are still not practical due to storage, configuration, and efficiency issues [33].

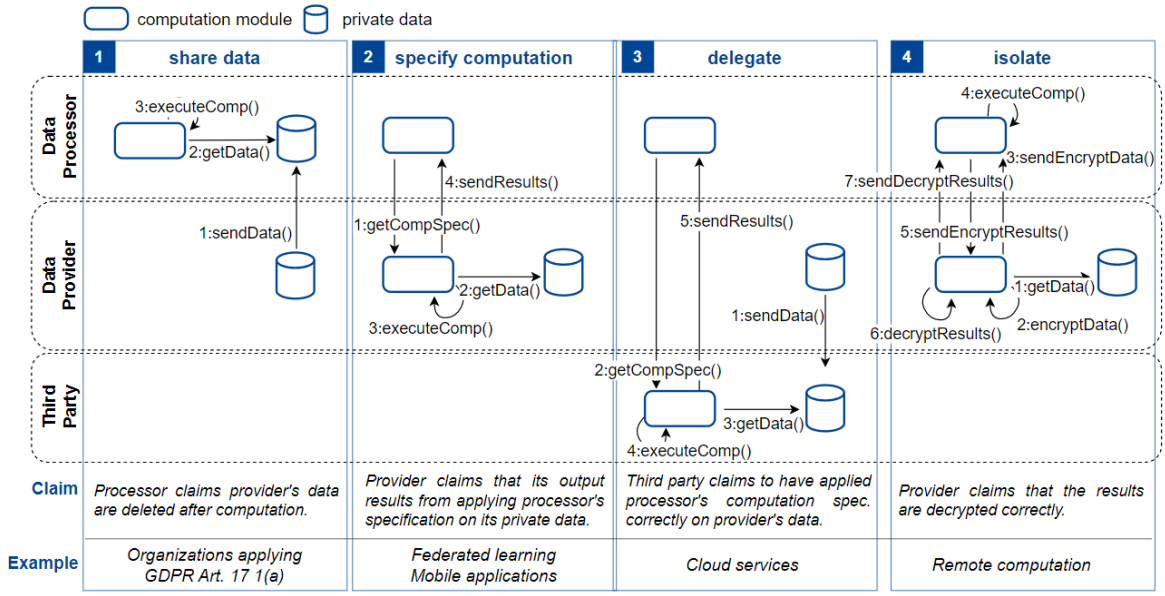

Figure 2. Interaction patterns with corresponding architecture, roles, claims and examples

4.2. Deciding Criteria and Preconditions

The additional outcome of refining pattern candidates has been a small framework for decision-making. Namely, deciding on applicable patterns depends on three criteria and five preconditions. Regarding criterions, first, the level of **trust** between parties is key. If the provider and processor trust that the other has capability and intent to handle information accordingly, then sharing the data or the model parameters is always a possibility. In other words, a simple verifiable promise that the data will be deleted or the computation will be executed suffices. Second, if either one or both parties are mistrustful, information sharing can still be intermediated by a **third party**. Finally, if no third party is trusted or available, more privacy-protective 'trustless' solutions are needed (cf. e.g., [29]). However, as Table 1 shows, introducing **complexity** (such as homomorphic encryption) is only justifiable in the strictest cases, because multiple options are available in less strict contexts [5].

Table 1
Trust scenarios (A-I) with available interaction patterns (1-4), sorted by complexity

Trust Scenarios	**Processor trusts party X with computation**		
Provider trusts party X with data	Provider trusted	Provider not trusted, 3rd party trusted	Neither party trusted
Processor trusted	A: 1,2,3,4	B: 1,3,4	C: 1,4
Processor not trusted, 3rd party trusted	D: 2,3,4	E: 3,4	F: 4
Neither party trusted	G: 2,4	H: 4	I: 4

Regarding preconditions, first, neither party has both the necessary components (i.e., sufficient data and computation specifications) nor the capability (i.e., data collection or processing capacity) to execute the process; thus, they have complementary roles [4]. Second, vulnerabilities are substantial enough (e.g., personally identifiable information cannot be anonymized without information loss) to prevent the parties from applying an easy solution. Third, the expected value is high enough to incentivize parties to collaborate (e.g., generated output is significantly more useful than the raw data itself; e.g., [1]). Fourth, expected costs of non-compliance are high enough to disincentivize it (e.g., punishment under GDPR that can reach up to 4% of revenue). Fifth and final, secure communication prevents man-in-the-middle attacks but is itself insufficient for verifying privacy is protected.

5. Exemplary Application of the Interaction Patterns

We evaluated the applicability of the interaction patterns during an FL project for German public services. In collaboration with Munich, Augsburg and Nuremberg, we designed and configured a system based on IBM FL [34] and used it to train a neural network for a multi-class text classification task. The dataset encompasses textual user feedback on their experiences and suggestions of using the German online public services as well as the usability rating in an ordinal range from 0 to 5. In our work, we used the raw text as input for the model and the categories as output (i.e., prediction). The municipalities are interested in collaborative training to automatically forward future feedback to the department corresponding to the category, such that, e.g., feedback about the user interface of the service is forwarded to the user experience (UX) team. Since the free texts may contain sensitive information, and it is not possible to detect all identifier entities (see, e.g.: [35]), we used the interaction patterns to provide data privacy guarantees.

The setup stage included three workshops with the cities to define the learning task, select the learning features of the data, and choose the architecture of the ML model. It is the only stage where an exchange of sensitive data is not needed. We used a character-level convolutional neural network as model architecture, inspired by the work of [36] and conducted two main training experiments. The first experiment was conducted on one municipality dataset to prove the usability of ML and the model architecture, with the accuracy of the fine-tuned model reaching 77,8%. The second experiment involved the FL system to train the model with all three datasets in a federated manner. The model's accuracy improved to 93,8%, confirming the hypothesis that sharing the data of the cities enhances the performance of the model.

Table 2
Application of the patterns 1 and 2 in the pre-processing and training stages

Dimensions		**Stages**	
		Preprocessing	Processing (training)
Pattern		1. Share data	2. Specify computation
Trust Scenario		[C:1,4] Municipalities do not have the ML proficiency to execute the preprocessing stage by themselves. By signing a data protection agreement and assuring them of access and usage (i.e., purpose) controls, we (data processors) acquired the trust.	[G:2,4] Municipalities involved the IT departments to integrate the system, ensuring the necessary infrastructure for local training is provided. We provided containerized applications as a form of specification. That has been enough to trust them to perform computation correctly.
Roles	Processor	fortiss: We provided expertise to preprocess the data as specified in the setup.	fortiss: We provided expertise to set up and orchestrate the FL system.
	Provider	Municipalities: They hold the training data and provide it as input for the learning task.	Municipalities: They provide the data and IT expertise to train the model locally, and send model weights to us.
	3rd Party	n/a: Trust is enabled by the agreement.	n/a: Trust is enabled by containerization.
Evidence for Claims (Accountability)		Documenting artifacts (e.g., intermediate results and metadata) and events (e.g., data deletion); in a Factsheet [40].	Documenting artifacts (e.g., model weights and evaluation metrics) and events (e.g., model convergence); in a Factsheet [40].

We applied the patterns in the preprocessing and training stages (cf. Table 2), as these stages involved sensitive data. Postprocessing was not included, as the project did not require any additional (e.g., robustness or fairness) checks. For preprocessing, we used the *share data* pattern because at the time, the municipalities lacked the necessary expertise or resources to preprocess their data. With the help of a contract and the verifiable claim that we will delete the data after the goal is completed, we received the raw data from the municipalities as an upload to a secure cloud. Preprocessing involved excluding the empty rows, cleaning the freetext from special characters, and fixing the misspelled words, with the help of the *pandas* Python package.

For the training, we applied the *specify computation* pattern. Taking the role of the aggregator, we specified computations by providing a containerized application using Docker for three municipalities to execute. The training of local models has been performed using the *keras* interface of the *tensorflow* Python package and aggregated into a common model according to steps in [34], using. Given the incomplete IT infrastructure at the time, we simulated the scenario of FL: we ran the application containers and provided the preprocessed training data as input to assure that the training is federated. In future training sessions, cities will have configured the IT infrastructure.

6. Discussion

6.1. Evaluation

Interaction patterns ideally help organizations pursue privacy-compliant ML. However, knowing which interaction to set up is not enough to satisfy the proposed principles (cf. Section 2.3). The table below evaluates whether a pattern satisfies a principle automatically or needs an additional mechanism to do so. Such mechanisms include internal policies within an organization with penalties for non-compliance; legally enforceable contracts for external entities; or other unspecified mechanisms. Since data providers are ultimately accountable, the onus is on them to set and enforce mechanisms.

In the **principle of lawful processing**, if consent for data collection from users is not obtained (Article 18 paragraph 1 point (a), [21]), or the reason for processing is not considered legitimate, patterns cannot help. In our case, legitimacy lies in "*the performance of a task carried out in the public interest*" (Article 18 paragraph 1 point I, [21]), which was already present before our involvement. The legality of the ML application is an implicit precondition for which provider is ultimately accountable.

Purpose and data limitation show that limits are easier to enforce when data remains on-site. *Specify computation* pattern requires a policy to ensure that the dataset is filtered beforehand, and that the computation parameters do not exceed the predefined purpose. When data is forwarded off-site, the provider can ensure that it has been filtered, but otherwise only enforce compliance with a contract. By contrast, the *isolate* pattern (when instantiated as homomorphic encryption) automatically ensures that an honest-but-curious, infected, or malicious processor cannot see the output, infer what the input consists of, or stretch the processing beyond the predefined purpose.

Table 3
Additional mechanisms each pattern needs for each principle in one stage

	Patterns			
Principles	**share data**	**specify computation**	**delegate**	**isolate**
Lawful processing	*not enough*	*not enough*	*not enough*	*not enough*
Purpose limitation	contract	policy	contract	*automatic*
Data minimization	policy	policy	policy	*automatic*
Accuracy	contract	policy	contract	policy
Storage limitation	contract	policy	contract	policy
Integrity & confidentiality	*not enough*	*not enough*	*not enough*	*automatic*
Accountability	contract	policy	contract	policy
Privacy by design	contract	*automatic*	contract	*automatic*

Accuracy and storage limitation deal with data maintenance, which are not directly addressed by interaction patterns. Instead, the data provider must have policies for repairing inconsistencies and deleting data after the allotted time. Integrity and confidentiality require more intensive data protection, which is only partly achieved by secure communication. Retaining the data on-site lowers the security risk from external attacks, but the provider can still be vulnerable to reconstruction attacks from an honest-but-curious or malicious processor – i.e., reconstructions of raw data points from model parameters or aggregate information [31]. As the processor is also at risk of being infected or having its results exploited, a risk assessment would be needed to determine vulnerability to such scenarios, and the selection of mechanisms that would guarantee privacy alongside the appropriate interaction pattern (e.g., differential privacy; [31]). If the collected data is immediately *isolated*, however, the only security measure that is needed is the protection of private keys.

Finally, privacy by design can be interpreted in two ways. Using a relaxed interpretation, a provider who demonstrates a conscious and institutionalized concern for protecting private data through pattern selection would be compliant, even when such protection cannot be easily operationalized in technical terms [37]. A stricter interpretation could only certify patterns in which the raw data does not leave the premises, or where more extensive protection exists. Thus, using only **share data** or **delegate** patterns without contracts or additional mechanisms would violate the "*proactive, not reactive, preventative not remedial*" foundation of the principle (p. 5, [38]). We assess our patterns in line with the stricter approach.

6.2. Connections with Related Work

Work connecting privacy, architecture design, and data processing provides two premises to support the use of interaction patterns. First premise is a need for operationalizing privacy-oriented legalese into technically legible steps, and documenting interactions. [39-40] uses privacy engineering methods, empirical validation and PDR to translate GDPR articles into a process model named Protection of Personal Data (ProPerData), sketching out an exemplary pattern candidate for documenting interactions. [29] conclude that despite attempts to satisfy GDPR requirements with privacy-enhancing techniques (e.g., homomorphic encryption or secure

multiparty computation), the gap in processor-provider interactions can only be solved by having rules for documenting them.

Second premise is a need for verifying claims that the proposed architecture is truly privacy-preserving. [5] modelled the threats to which default FL processing is vulnerable, concluding that FL alone cannot satisfy two (data minimization and anonymization) out of three (transparency and consent) core aspects of privacy. To verifiably guarantee protection, FL would need to be combined with additional techniques (e.g., secure enclaves, cryptography, differential privacy), based on trade-offs between accuracy, computational costs and competing objectives.

7. Conclusion

We propose patterns for designing information systems architectures around parties interacting in privacy-preserving contexts. Following a design science methodology, we describe four interaction patterns that involve data processors, providers and third parties, and provide claims as the basis for keeping parties accountable. With an exemplary application, we sketch how patterns can be applied, and discuss how they map to existing privacy principles. The intended benefit of our contribution is to structure the operationalization of privacy principles via a taxonomy of patterns, making preconditions and differences between core and non-core components explicit, and clarifying trade-offs between technical extensions and rule-building activities.

We identify the following limitations. First, preconditions that motivate parties to interact – vulnerability, complementarity, high expected value from collaboration, and high expected costs from non-compliance – might have degrees of interpretation. Our case involved parties with an incentive to improve efficiency, and freedom to establish trust via agreements; other cases may require stricter proofs of satisfied preconditions. Second, secure communication (the fifth precondition) cannot always be expected. Where data, computation specifications or cryptographic keys are vulnerable, patterns may require additional security components. Third, implicit assumptions may be present. We do not specify what policies or contracts should contain, and assume that organizational resistance is minimal. In reality, despite high expected value and low expected risk, data providers may still be reluctant to share data, opting instead for more intensive *isolate* methods or performing no learning at all.

Future work involves three areas. First, another iteration of PDR in different contexts will help operationalize decision-making and explore extensions of privacy-preserving design patterns. We expect that validation will introduce objectives, preconditions, and theories to expand our taxonomy. Second, further specification of tools for assessment and decision-making is needed; we are striving to build a corresponding web tool. Finally, different stages of ML and different assumptions may require different patterns. Combinations with the *specify computation* pattern in the training stage may reveal different ways of instantiating the federated learning concept, depending on other conditions beyond overall trust. We expect the patterns to be a useful conceptualization of privacy-preserving learning outside of our cross-silo scope, and invite research contributions to test the assumption.

Acknowledgements

This research was partially funded by the Bavarian Ministry of Ministry of Economic Affairs, Regional Development and Energy. We thank IBM Research Almaden, IBM Research Ireland and the public city administrations of Munich, Augsburg, and Nuremberg for their cooperation and advice. We thank our reviewers for their careful reading and constructive remarks.

References

[1] R. Miotto, F. Wang, S. Wang, X. Jiang, J. T. Dudley, Deep learning for healthcare: Review, opportunities and challenges, Briefings in Bioinformatics 19.6 (2017) 1236-1246.

[2] M. Flah, I. Nunez, B. W. Chaabene, M. L. Nehdi. Machine Learning Algorithms in Civil Structural Health Monitoring: A Systematic Review, Archives of Computational Methods in Engineering 28.4 (2021) 2621-2643.

[3] P. Courtiol, C. Maussion, M. Moarii, et al., Deep learning-based classification of mesothelioma improves prediction of patient outcome, Nature Medicine 25.10 (2019) 1519-1525.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A., Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS, 2017, pp. 1273-1282.

[5] K. Bonawitz, P. Kairouz, B. McMahan, D. Ramage, Federated Learning and Privacy: Building privacy-preserving systems for machine learning and data science on decentralized data, Queue 19.5 (2021) 87-114.

[6] T. Li, A. K. Sahu, A. Talwalkar, V. Smith, Federated Learning: Challenges, Methods, and Future Directions, IEEE Signal Processing Magazine 37.3 (2020) 50–60.

[7] Q. Li, Z. Wen, Z. Wu, S. Hu, N. Wang, Y. Li, X. Liu, B. He, A Survey on Federated Learning Systems: Vision, Hype and Reality for Data Privacy and Protection, IEEE Transactions on Knowledge and Data Engineering, published online, 2021. URL: https://ieeexplore.ieee.org/document/9599369

[8] N. Baracaldo, A. Anwar, M. Purcell, A. Rawat, M. Sinn, B. Altakrouri, D. Balta, M. Sellami, P. Kuhn, U. Schopp, M. Buchinger, Towards an Accountable and Reproducible Federated Learning: A FactSheets Approach, 2022. arXiv: 2202.12443. URL: https://arxiv.org/abs/2202.12443

[9] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership Inference Attacks Against Machine Learning Models, IEEE Symposium on Security and Privacy, 2017, pp. 3-18.

[10] M. Fredrikson, S. Jha, T. Ristenpart, Model inversion attacks that exploit confidence information and basic countermeasures, Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, 2015, pp. 1322– 1333.

[11] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. Quek, H. V. Poor, Federated learning with differential privacy: Algorithms and performance analysis, IEEE Transactions on Information Forensics and Security 15 (2020) 3454–3469.

[12] S. Truex, N. Baracaldo, A. Anwar, T. Steinke, H. Ludwig, R. Zhang, Y. Zhou, A Hybrid Approach to Privacy-Preserving Federated Learning, Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, AISec'19, 2019, pp. 1-11.

[13] T. S. Brisimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, W. Shi, Federated learning of predictive models from federated electronic health records, International Journal of Medical Informatics 112 (2018) 59–67.

[14] N. G. Packin, RegTech, compliance and technology judgment rule, Chi.-Kent L. Rev. 93.193 (2018).

[15] S. Kacianka, A. Pretschner, Designing accountable systems, FaccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 424–437, 2021.

[16] S. Eriksén, Designing for accountability, NordiCHI '02: Proceedings of the second Nordic conference on Human-computer interaction, pp. 177–186.

[17] L. Lessig, Code and Other Laws of Cyberspace, Basic Books Inc., New York, NY, 1999.

[18] L. Lessig, Code is law, Harvard Magazine, 2000. URL: https://www.harvardmagazine.com/2000/01/code-is-law-html

[19] M. Buchinger, P. Kuhn, D. Balta, Dimensions of Accountability in Interorganizational Business Processes, Proceedings of the 55th Hawaii International Conference on System Sciences, HICSS, 2022, Association of IEEE, pp. 429 – 438.

[20] M. Brundage, S. Avin, J. Wang, et al., Toward trustworthy AI development: mechanisms for supporting verifiable claims, 2020. arXiv:2004.07213. URL: https://arxiv.org/abs/2004.07213
[21] European Union, Regulation (EU) 2016/679 (General Data Protection Regulation), 2018. URL: https://gdpr-info.eu/
[22] European Union, Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act) and Amending Certain Union Legislative Acts, COM/2021/206 final, 2021.
[23] M. Eßer, P. Kramer, K. von Lewinski, DSGVO BDSG, DatenschutzGrundverordnung, Bundesdatenschutzgesetz und Nebengesetze, 6. Aufl., Köln, 2018.
[24] A. Toy, D. C. Hay, Privacy auditing standards, Auditing: A Journal of Practice & Theory 34.3 (2015) 181-199.
[25] S. Buckl, F. Matthes, A. W. Schneider, C. M. Schweda, Pattern-based design research – an iterative research method balancing rigor and relevance, in: J. vom Brocke, R. Hekkala, S. Ram, M. Rossi (Eds.), DESRIST 2013: Design Science at the Intersection of Physical and Virtual Design, volume 7939 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2013.
[26] K. Sedig, P. Parsons, Interaction Design for Complex Cognitive Activities with Visual Representations: A Pattern-Based Approach, AIS Transactions on Human-Computer Interaction, 5.2 (2013) 84-133.
[27] J. O. Horchers, A pattern approach to interaction design, AI and Society 15 4 (2001) 359-376.
[28] N. Truong, K. Sun, S. Wang, F. Guitton, Y. Guo, Privacy preservation in federated learning: An insightful survey from the GDPR perspective, Computers and Security 110 (2021) published online.
[29] T. K., Rodrigues, K. Suto, H. Nishiyama, J. Liu, N. Kato, Machine Learning Meets Computation and Communication Control in Evolving Edge and Cloud: Challenges and Future Perspective, IEEE Communications Surveys and Tutorials, 22.1 (2020) 38-67.
[30] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, Foundations and Trends in Theoretical Computer Science 9.3-4 (2014) 211–407.
[31] X. Yi, R. Paulet, E. Bertino, Homomorphic Encryption, Homomorphic Encryption and Applications (2014) 27–46
[32] C. Marcolla, V. Sucasas, M. Manzano, R. Bassoli, F. H. P. Fitzek, N. Aaraj, Survey on Fully Homomorphic Encryption, Theory, and Applications, Proceedings of the IEEE 110.10 (2022) pp. 1572-1609.
[33] H. Ludwig, N. Baracaldo, G. Thomas, et al., IBM Federated Learning: an Enterprise Framework White Paper, 2020. URL: https://arxiv.org/abs/2007.10987 (visited on November 15, 2022)
[34] W. B. Tesfay, J. Serna, K. Rannenberg, Privacybot: detecting privacy sensitive information in unstructured texts, Sixth International Conference on Social Networks Analysis, Management and Security, SNAMS, 2019.
[35] X. Zhang, J. Zhao, Y., LeCun, Character-level Convolutional Networks for Text Classification, Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15, 2015.
[36] B.-J., Koops, R. Leenes, Privacy regulation cannot be hardcoded. A critical comment on the 'privacy by design' provision in data-protection law, International Review of Law, Computers and Technology 28.2 (2014) 159-171.
[37] A. Romanou, The necessity of the implementation of Privacy by Design in sectors where data protection concerns arise, Computer Law and Security Review 34.1 (2018) 99-110.
[38] D. Huth, Patterns for GDPR Compliance, PoEM 2017: Doctoral Consortium and Industry Track Papers, 2017, pp. 34-40.
[39] D. Huth, Development of a reference process model for GDPR compliance management based on enterprise architecture, PhD thesis, Technical University of Munich, Munich, Germany, 2021. URL: https://mediatum.ub.tum.de/doc/1593644/1593644.pdf

[40] M. Arnold, R. K. E. Bellamy, M. Hind, et al., FactSheets: Increasing trust in AI services through supplier's declarations of conformity, IBM Journal of Research and Development 63.4/5 (2019) 1-13.
[41] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, N. Papernot, Machine unlearning, Proceedings of the IEEE Symposium on Security and Privacy, 2021, pp. 141–159.