Automating Data Governance with Generative AI

Linus W. Dietz[1], **Arif Wider**[2], **Simon Harrer**[3, 4]
[1]King's College London
[2]Hochschule für Technik und Wirtschaft Berlin
[3]innoQ Deutschland GmbH
[4]Entropy Data GmbH
linus.dietz@kcl.ac.uk, wider@htw-berlin.de, simon.harrer@entropy-data.com

Abstract

The exchange of data within and between organizations is governed by company policies and data protection laws. As policies and data flows change over time, maintaining compliance in data exchange poses a complex challenge. In federated data architectures, validating data access requests is both critical and labor-intensive. To formalize this task and enable automatic compliance checks, rule-based constraint languages can be used. However, access constraints often come from legal texts, and translating them into formal data contracts is tedious, repetitive, and prone to error. This can lead to inconsistencies and delays in staying compliant with evolving regulations. To address this, we developed Governance AI, a tool based on a large language model (LLM) that evaluates data access requests by considering relevant policies, the type of data, and the request's context. To test our approach at scale, we built an access request generator and a testing framework for computational data governance. In our evaluation of 110 access requests from two business domains, e-commerce and life insurance, we found that LLM-generated test cases were highly realistic and effective for comprehensive testing. Governance AI demonstrated a stricter approach than human experts, issuing a higher number of warnings and consistently flagging all critical cases where experts raised data sharing concerns. While the tool generated 3.6 times more warnings than human experts, further review showed that 80% of these were accurate. Our findings contribute to the automation of data governance by critically assessing the potential of generative AI in evaluating data access requests regarding legislation and internal policies.

Tool — datamesh-manager.com
Supplementary material —
github.com/LinusDietz/Automating-Data-Governance

Introduction

The growing reliance on digital technologies has led organizations to collect and store more sensitive personal data. While this data offers valuable insights and enables innovation, it also raises serious privacy concerns, prompting increased regulation (European Parliament and Council of the European Union 2016; European Union 2024; California State Legislature 2018). Organizations now face a key challenge: balancing data utility with strong privacy protection. Sharing data too freely risks privacy breaches and harsh regulatory penalties (Wolff and Atallah 2021); overly strict controls can limit useful analytics.

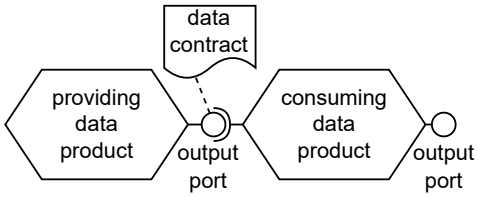


Figure 1: **Data products and their composition.** The sharing of data is defined through data contracts that specify the usage terms of the output ports.

To meet this challenge, organizations need sound data governance practices that support secure, controlled sharing. Manual access management is costly and prone to error. Instead, data owners need automation tools to help define precise access restrictions that let them share data safely without compromising privacy. A modular data architecture, such as data mesh, enables this kind of automated governance (Schultze and Wider 2021; Dehghani 2022; Wider, Jarmul, and Akhtar 2024).

Data mesh is a recent approach to enterprise data management. It emphasizes domain-based modularization, federated governance, and automation through self-service platforms (Dehghani 2020; Schultze and Wider 2021). The core unit in a data mesh is the data product. A data product is more than just a dataset treated with a product mindset. It is also a data service built for composability. This means each data product is designed to be easily combined with others, enabling flexible and scalable interoperation. Data products interact via output ports: defined interfaces that specify the schema of the shared data. Any conditions tied to an output port, such as service-level agreements (SLAs), guarantees, or constraints, are described in related data contracts. As shown in Figure 1, higher-order data products consume data from one or more sources by connecting to their output ports. These products typically transform the data internally and may expose their output ports to share new, use-case-specific data.

Automation and verification of access restrictions, as well as consumers' adherence to sharing constraints, can be achieved through clearly defined, verifiable data contracts. Using an open standard for structuring these contracts, such as the Open Data Contract Standard (Bitol 2025), supports this goal. Consumers, in turn, rely on the guarantees set out in the provider's contract, including data structure, quality, and SLAs (Wider, Jarmul, and Akhtar 2024).

Global privacy rules governing the handling of personally identifiable information (PII), whether set by law or internal policy, must also be followed–ideally through automated checks. One way to enable this is by expressing contract-level constraints and global governance rules in a machine-readable format, using a formal constraint language. However, these constraints often stem from legal or policy documents, and translating them into formal contracts is tedious, error-prone, and must be repeated whenever the originals change.

To address this, we propose a new approach to computational data governance that draws on the decentralized structure of data mesh and advances in generative AI. Rather than requiring translation into formal contracts, we used large language models (LLMs) to process legal documents, such as privacy policies, directly. Thereby, we addressed the core challenge in data governance: determining whether data flows comply with the constraints defined in data contracts and global policies, i.e., the critical decision of whether data access can be granted. While we hypothesize that such a system should not fully replace human experts in this decision-making process, our goal was to reduce their workload by automating access verification and to assist them with warnings and suggestions about potential policy violations to improve accuracy.

After reviewing related work, we described three methodological contributions (C1–C3) and evaluated them, which led to a fourth contribution (C4).

C1: We described the design of the Governance AI, an LLM-based tool that automatically checks access requests for policy violations in a data mesh architecture.

C2: We developed a testing framework for checking policy violations of data access requests at scale.

C3: We evaluated our Governance AI tool using a comprehensive set of $n = 110$ generated data access requests in two domains.

C4: Based on the results, we evaluated how the Governance AI tool compares to data governance experts in assessing access requests, as well as the accuracy of its warnings and suggestions.

Our findings have both theoretical and practical implications for the fast-growing field of computational data governance.

Related Work

Data mesh is a recent, industry-led approach to decentralized enterprise data management (Dehghani 2022; Schultze and Wider 2021; Perrin and Broda 2024). Because it originated in industry, academic research on the topic is still limited (Goedegebuure et al. 2024), with Machado, Costa, and Santos being the first to introduce the concept of data mesh to the academic community (Machado, Costa, and Santos 2022). A key reason for its adoption in industry is improved efficiency in data governance (Bode et al. 2024; Oppold, Fritz, and Woltmann 2025), making data mesh a suitable framework for the goals of our work.

Computational Data Governance

An important concept in data mesh is *computational data governance*, which refers to automating governance tasks using tools provided by a mesh-wide data infrastructure platform (Dehghani 2020). For example, by tagging sensitive data correctly, the platform can automatically enforce protections and ensure policy compliance (Wider, Harrer, and Dietz 2025). In this context, Joshi, Pratik, and Rao (2021) presented an industry case study on data governance in a data mesh setting, but with little focus on automation while Wider, Verma, and Akhtar (2023) examined different ways to automate governance tasks in data mesh platforms and proposed a method for checking data privacy constraints automatically (Wider, Jarmul, and Akhtar 2024). By adopting sharer-driven contracts, i.e., data product owners define constraints for data consumers, we aim to automate the aspect of data governance that ensures data flows comply with policies and regulations.

The work presented by Dolhopolov, Castelltort, and Laurent (2024) is similar to ours in that it also implements automated checks of data access attempts by enforcing governance policies. However, in their approach, these policies are formal, rule-based definitions that must be created from legal documents. This creates the need to repeatedly translate legal text into formal rules whenever the documents change. Borovits et al. (2023) focus on data privacy in the context of computational governance in their conceptual framework. While we build on several of their ideas, we extend their work by using generative AI and providing an implementation and evaluation of our approach.

An alternative approach to managing personal data in a privacy-preserving way is a concept like AuthApp (Both et al. 2024), which uses the principles of SOLID data pods to enable GDPR-compliant data sharing (Dedecker et al. 2022; Sambra et al. 2016). As such approaches shift control to individual users, they take a different direction from the enterprise-based model described in our work.

Generative AI for Policy Compliance Testing

Generative AI, specifically LLMs, appears well suited for analyzing legal texts and making decisions on a case-by-case basis. In a seminal study, (Bignotti and Camassa 2024) showed that GPT-4, when presented with historic Italian constitutional court rulings, was able to identify relevant articles and produce consistent rulings with few hallucinations. However, they also noted that the model showed a clear bias toward progressive interpretations of constitutional law. Li and Maiti (2025) applied an LLM for continuous compliance checks in the agricultural sector, but their work does not focus on privacy policies. In the field of privacy engineering, Amaral et al. (2022) investigated the use of LLMs for checking whether privacy policies meet

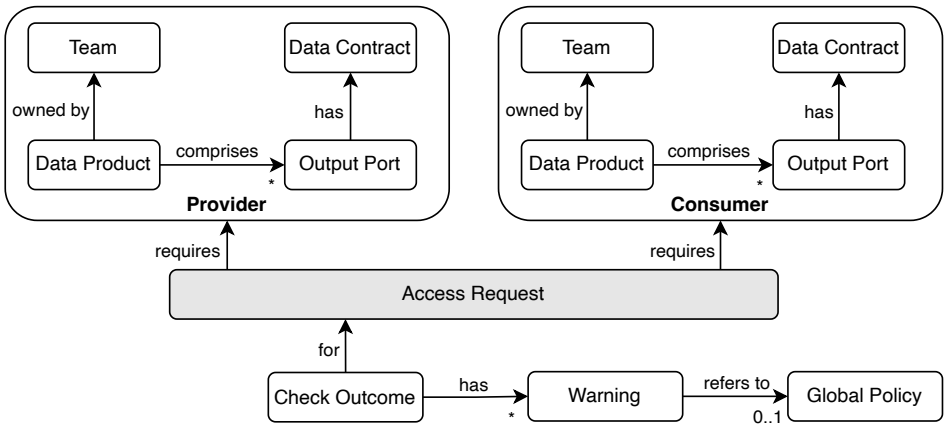


Figure 2: **Meta model of the elements involved in a data access request**. Each access request, if accepted, represents a data flow between a providing and a consuming data product. Governance AI examines the request for potential policy violations, producing a—potentially empty—list of warnings, each accompanied by a reference to the relevant section of the applicable global policy.

the standards set by the GDPR. Overall, legal LLMs are an emerging topic with both engineering and ethical challenges (Lai et al. 2024). In our work, we use LLMs to determine whether data access requests within a data landscape are complying with company policies and legislation.

To systematically evaluate our approach, we use LLM-generated data access request test suites inspired by the work of Herdel et al., which generated AI application scenarios to classify them according to the EU AI Act (Herdel et al. 2024; European Union 2024). Test-case generation using LLMs has been quite widespread in the previous years, such as in cybersecurity, for creating penetration tests for identifying security issues (Hilario et al. 2024) and in software testing (Shin 2024). To the best of our knowledge, our approach is the first to apply generative AI to support data governance in decentralized data architectures.

Governance AI for Analyzing Data Access Requests

This section describes the design of Governance AI (Wider, Harrer, and Dietz 2025), which is a feature of a deployed commercial data mesh management tool[1]. First, we present the relevant concepts of the tool needed for this work. Then, we explain in detail how Governance AI automatically evaluates access requests for policy violations.

Basic Concepts of Data Mesh Management

To clarify the concepts of decentralized data architectures, we use the following abstraction model. A *data mesh* is modeled as a metadata graph of different data products and data flows. The nodes represent *data products* with their output ports, and the edges represent *access dependencies* between data products. Each data product is owned by a team and provides datasets through separate output ports, which are distinguished by technology, version, environment, and data model. The guarantees, such as data model specifications, usage constraints, and limitations, are defined within a data contract for each output port. The commercial tool is an enterprise data marketplace that supports managing data mesh architectures with native support for data products and data contracts. It supports both the Open Data Contract Standard (Bitol 2025) and the Data Contract Specification (Christ and Harrer 2024) to represent data contracts.

An *access request* potentially adds an edge to the data mesh graph, also called a data map. It represents a request by a *consuming data product* to access an output port of a *providing data product*. Members of the team responsible for the providing data product can provide access to the consuming data product. *Global policies* are rules that restrict the structure of the data mesh graph. These rules are defined in plain text and are not embedded within the graph; however, they can apply to any element of the graph, including data products, teams, data contracts, and data access requests. An important data governance challenge is to ensure that the entire graph adheres to the global policies at all times. Figure 2 shows policy checks for open access requests, as this is the main focus of this paper.

Detecting Policy Violations of Access Requests Using LLM-powered Automated Policy Checks

To automate the checking of data access requests, we propose Governance AI as a feature of a data mesh management system. At the time of writing, Governance AI used GPT-4o hosted on Microsoft Azure in Sweden with the default content filter. The outcome of the automatic checks is a set of warnings describing potential policy violations. Each warning either references a specific global policy or serves as a general safeguard, such as ensuring compliance with legal requirements. Additionally, a warning may include a suggested action to resolve the violation. This information is presented to the owners of the providing data product, who decide whether to approve or reject the request.

[1]https://www.datamesh-manager.com

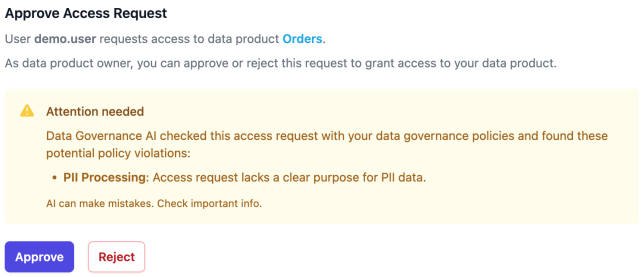


Figure 3: **Outcome of the Governance AI access request analysis.**

It is a design principle that a human has the final say. This is necessary for accountability, acceptance of AI solutions within organizations, and the potential flaws of the technology used. Figure 3 shows how the outcome of the Governance AI check is displayed to a data owner. The AI does not make decisions but generates warnings to help decision-makers make informed choices. Under the EU AI Act, this system is classified as *"limited risk,"* meaning users must be informed that the warnings are generated by an AI system.

The prompt engineering consists of both a system prompt and a user prompt. The final version was developed through a systematic, iterative process in which each numbered component of the prompt, shown in Figure 4, was modified and evaluated using a benchmark test suite, which we introduce later. We selected the configuration that maximized the F_1-score for correctly accepting recommendations and warnings, while ensuring that no access requests deemed inappropriate by domain experts were accepted.

The system prompt, shown in full in Figure 5, provides general instructions about the task, the AI's persona, and the steps to approach the task. The user prompt includes all necessary metadata and context, such as the access request, the providing and consuming data products, and the relevant global policies. It also contains specific instructions for detecting policy violations and converting them into warnings with suggested remediation actions.

Whenever a data access request is submitted, it is automatically checked by Governance AI. Additionally, Governance AI checks can also be called through a dedicated API, which we used for testing purposes.

A Data Mesh Testing Framework for Automatically Checking Policy Violations

This section describes the design and implementation of an automated framework for testing data governance.

Design Considerations

In general, the testing framework follows a traditional approach to test automation by executing access request test cases and producing test results. Each test case runs fully automatically, in isolation from others, and is designed to produce reproducible results. However, full reproducibility was difficult to achieve in practice due to the probabilistic

System prompt

1. **Task.** Describes the task of analyzing access requests.
2. **Persona.** The model should adopt the persona of a Data Governance Expert.
3. **Steps.** Six steps of how to analyze an access request.

User Prompt

1. **Access Request** that needs to be analyzed `(YAML)`
2. **Provider** side of the access request, including the providing data product, the relevant output port, the data contract, and the providing team `(YAML)`
3. **Consumer** side of the access request, including the consuming data product, all output ports, data contracts, and the consuming team `(YAML)`
4. **Global Policies** governing the data mesh (`text`).
5. **Detailed Instructions** about the task, the requirements, and additional constraints.
6. **Required Elements** of the output with an explanation. The structure of the required elements was enforced using the "Structured Outputs JSON mode."

Figure 4: **Structure of the access request analysis prompt.** The full prompt is provided in the supplementary material.

> Your task is to analyze data access requests whether the consumer should be granted access to the data offered by the provider.
>
> Respond as a data governance expert.
>
> Follow these steps:
> 1. Understand the purpose of the access request of the consumer.
> 2. Understand the data contract of the producer.
> 3. Check that the purpose of the access request is in line with the data contract of the producer.
> 4. Understand the global policies of the organization.
> 5. Validate if this access request violates any policies.
> 6. Formulate warnings if and only if there is an obvious policy violation.
>
> Make sure that the access request does not violate any policies or restrictions of the data contract of the producer.

Figure 5: **System Prompt of Governance AI.**

nature of LLMs. The elements of an access request test case and its corresponding test result are as follows:

- Data map
- Access request
- Expected outcome (No objections, warn)
- Actual outcome (No objections, list of warnings, error)

Each test case describes a complete data mesh, which includes a specific access request to be checked and an expected outcome to be asserted. The test execution flow consists of three steps: the *"Setup step,"* which resets the state to match the data mesh defined in the test case; the *"Test step,"* where Governance AI runs the policy check for the specified access request; and the *"Assert step,"* where the expected outcome is compared to the actual result.

Implementation

To enable fully automated test execution, we implemented the test framework in Python as a command-line application. The framework takes a JSON file as input, which contains all test cases. Each test case links to a folder containing all the YAML files describing the data products, data contracts, teams, and access requests of a particular data mesh, and it identifies a single access request via its ID to be checked within that data mesh. For each test case, the framework uses the tool's API to reset the system state to match the referenced data mesh, ensuring isolation between executions. It then calls the Governance AI API to analyze the access request and returns a list of warnings in JSON format. An empty list indicates the outcome is *"No objections,"* while a non-empty list corresponds to *"Warn."* If the API call fails, e.g., due to errors in referencing data products, the actual outcome is an *"Error"*. After all test cases have been executed, the framework produces a JSON file containing each test case along with its corresponding result.

While LLMs have shown relatively consistent output when handling complex cases (Bignotti and Camassa 2024), their probabilistic nature prevents test execution from being strictly idempotent. As a result, the test framework does not provide a summary score for the executed tests. Instead, we created a dashboard that allows quick visualization of each test case, highlighting whether the expected outcome matches the actual outcome.

Systematic Generation of Access Request Test Cases

To put the testing framework into action, we collaborated with domain experts to create two *data maps* in the insurance and e-commerce domains. These data maps enabled us to test realistic scenarios for evaluating our proposed system. The domains were chosen due to their increasing adoption of data mesh architectures and the high importance of data governance (Ramos et al. 2024). In the next step, we used these prototypical data maps to generate a comprehensive set of potential data access requests.

Creating and Validating Data Maps

The data maps were designed to represent a minimal yet complete data mesh architecture for a typical company in the insurance and e-commerce domains. Drawing on our experience, we first created data products along with their corresponding data contracts. To ensure realism, the resulting data maps were reviewed and validated by a domain expert.

To further enhance their authenticity, we incorporated real-world privacy policies from companies in the respective domains. According to the experts, publicly available privacy policies represent the most relevant basis for determining whether data access can be granted to another team within the organization. This is because, when a contract is established between a company and a customer, the privacy policy becomes part of the agreement. As the privacy policy defines the scope of data processing, such as requiring prior consent before contacting a user for marketing purposes, it imposes constraints on internal data sharing, particularly when PII is involved. Notably, both policies were available to the system in the companies' primary language (German), whereas all other inputs, including access requests, were in English. In the following, we briefly present the two data maps along with the experts' design rationales for the insurance and e-commerce domains.

Insurance domain. In the insurance domain, our initial data map proposal underwent substantial revisions by the domain expert, resulting in a final set of five providing data products: *"Actor," "Benefits," "Life Insurance Contracts," "Marketing Campaigns,"* and *"Underwriting Life Insurance."* The expert explained that, due to legal constraints, different business branches within the same insurance group must remain separate, making data sharing across company boundaries impermissible. Consequently, they modeled the life insurance domain as a core business branch subject to strict data governance limitations. Furthermore, based on their experience, the expert noted that each domain or team typically maintains only one outward-facing data product. Each data product exposes a single output port with a well-defined set of fields. To exemplify this, we listed the eight fields of the *Life Insurance Contracts* data product in Table 1. For replication purposes, all data products are available in the data map directory included in the supplementary material. The domain expert holds the position of "Head of Data & AI Governance" at a large insurance firm and has 15 years of industry experience.

E-commerce domain. The expert proposed three teams, each responsible for multiple data products. *Shop Operations* provided the data products *"Customers," "Order," "Order Lines,"* and *"Payments"; Logistics* managed *"Products and "Fulfillment;"* and *Marketing* was responsible for the (marketing) *"Campaigns"* data product. Similar to the insurance domain expert, the e-commerce expert confirmed that this foundational data map is both minimal yet representative of a typical e-commerce company. The e-commerce domain expert holds the position of "Head of Data Governance" at a large European e-commerce company and has nine years of industry experience.

Life Insurance Contracts (insurance domain)		
Field	**Type**	**Tags**
policyholder name	`string`	PII
policyholder base data	`string`	PII
prior medical record	`string`	PII
sum insured	`string`	
premiums	`string`	
agent	`string`	
agent commission	`int`	trade secret
beneficiaries	`string`	PII

Table 1: **Example of tagged fields in a data product output port.**

Designing an LLM Prompt for Access Request Generation

Building on previous work on generating use cases with LLMs (Herdel et al. 2024), we aimed to design a prompt capable of producing a comprehensive list of access requests. Following an iterative development process informed by OpenAI's prompt engineering guidelines (OpenAI 2025), we informally evaluated the generated responses and experimented with different prompt configurations.

Ultimately, the prompt structure outlined in Figure 6 yielded satisfactory results, which we then evaluated with data governance experts from both domains. Providing the LLM with extensive context directly from the data mesh in YAML format proved viable and enabled fully automated generation of test cases. Our test suite generation script iterates over all elements in the data map and dynamically inserts relevant information, such as the fields in the output ports of the providing data products, into the prompt. This design makes the prompt domain-agnostic, allowing reuse for generating access request test suites across different data maps. Enforcing a structured output format via a JSON schema was essential to ensure the completeness of information in all cases. The only element excluded from the final prompt was the entire data map, as including it led to misaligned access requests. To guarantee balanced coverage within the data maps, we executed the prompt independently for each providing data product.

Creating Access Request Test Cases

One relevant design consideration in the access request generation step was that the output of the LLM be available in a machine-readable format that can be automatically transformed into the YAML format required to describe access requests in the testing framework. Concretely, we used the LLM's response to create two artifacts: a newly introduced consuming data product representing a use case, and a corresponding access request toward the providing data product. In Figure 7, we exemplify six such consuming data products for the *"Customers"* and the *"Order"* data products. Note that the consuming data products could also be data applications; however, within the scope of this work, we do not differentiate between data products and data applications. The dashed lines represent the pending access requests.

The LLM's machine-readable output enables automatic conversion into the format required for testing the generated access requests. These can then be seamlessly added to the data map and executed within the testing framework.

System prompt

1. **Persona** We asked the model to adopt the persona of a Senior IT Governance Specialist in the respective domain

User Prompt

1. **Overview** of the task
2. **Providing data product** including the data contract of the output port `(YAML)`
3. **List of potential consuming teams.** This was to avoid generating access requests within the same team/domain. We provided the teams as `text` with their names and IDs.
4. **Detailed Instructions** about the task and the requriements in plain `text`.
5. **Privacy Policy** We specifically asked for a wide range of realistic access requests that can be accepted or should be rejected according to the privacy policy. We included the full privacy policy governing the data mesh into the prompt as unprocessed `text`.
6. **Required Elements** of the output with an explanation. The structure of the required elements was enforced using the "Structured Outputs JSON mode"
7. **Example** of a representative access request `(YAML)`

Figure 6: **Structure of the access request generation prompt.** Refer to Figure 2 for further details about elements in the data mesh. The full prompt is provided in the supplementary material.

Results

In total, we generated 10 use cases with respective access requests for each providing data product, resulting in 50 test cases in the insurance domain and 60 in the e-commerce domain. The number of test cases was limited by the available time of the domain experts to analyze them; nevertheless, with an overall total of 110 test cases, we ensured a comprehensive evaluation. In the following sections, we evaluate the test cases regarding their realism, Governance AI's ability to detect policy violations, and the usefulness of the warnings and suggestions.

Realism of Generated Access Requests

Since the access requests and the underlying use cases were LLM-generated, it was important to validate that the access requests were realistic and that the use cases were plausible within their respective domains. For this reason, as a first step, we asked domain experts to assess the access requests and judge their realism to eliminate any potentially hallucinated test cases. The experts classified each request as either

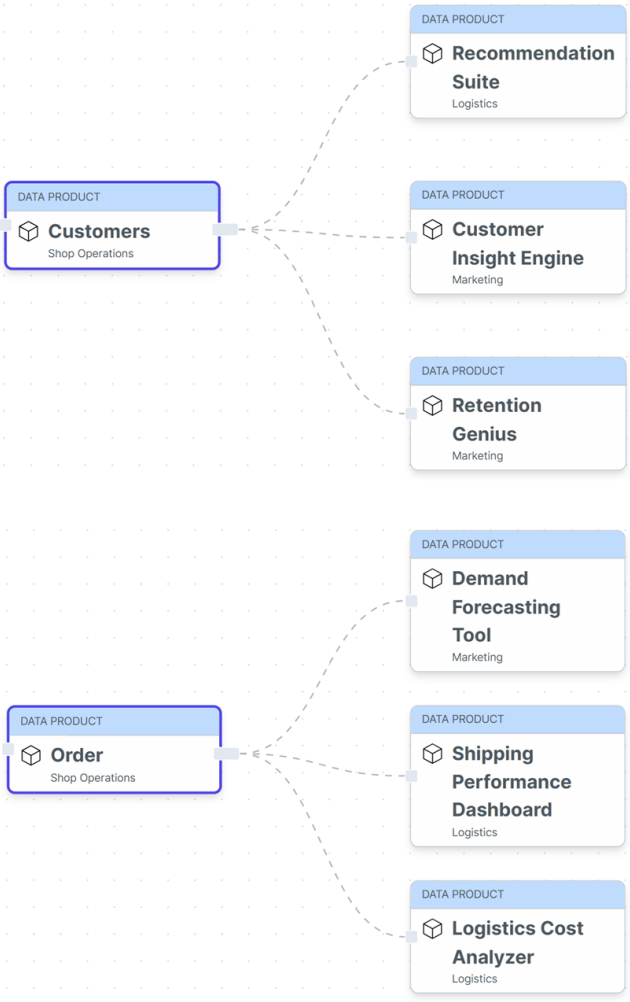


Figure 7: **Examples of access request scenarios generated for the e-commerce domain.** Overall, we generated 10 access requests for each providing data product.

already existing (i.e., they were aware of companies engaging in similar activities), *generally realistic*, or *unrealistic*, with the latter category discarded from subsequent steps.

In both domains, the vast majority of access requests were classified by the experts as realistic, with 85% already existing in the e-commerce domain (Figure 8). In the e-commerce domain, only one access request was discarded, while in the insurance domain, four were discarded. When analyzing the experts' reasons for excluding use cases, two were ethically dubious or violated laws, such as the EU AI Act (European Union 2024), one described nonsensical marketing efforts, and two were practically infeasible operational improvements. The expert in the insurance domain did not distinguish between existing and realistic use cases due to a misunderstanding of the difference. However, we ensured that the inclusion and exclusion criteria were correctly understood.

Takeaway: Almost all LLM-generated access requests are realistic, confirming the merit of LLM-based generation of the test cases.

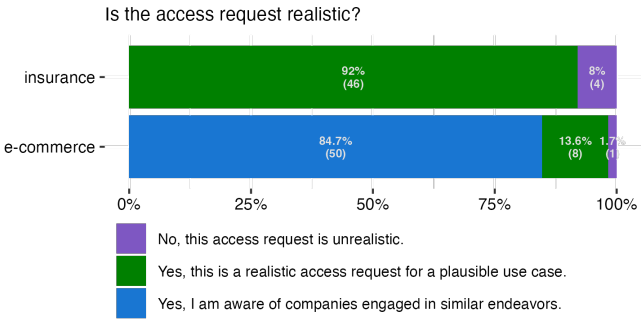


Figure 8: **Almost all LLM-generated access requests are realistic, with most already existing in the e-commerce domain.** The figure shows the distribution of the experts' assessment regarding the realism of access requests.

Detection of Policy Violations

Using the previously introduced testing framework, we systematically evaluated all realistic access requests regarding policy violations using Governance AI.

The output of the Governance AI was compared with the expert baseline, resulting in four possible outcomes. In the first two cases, the expert and the system agreed: either both had *"no objections"* to the access request or both identified issues that warranted a *"warning."* In cases of disagreement, the Governance AI either issued a warning while the expert had no objections, or the expert raised concerns while the AI did not. Figure 9 summarizes the four scenarios using confusion matrices, with e-commerce on the left and the insurance domain on the right. The numbers in the fields show the absolute and relative counts.

Importantly, there were no cases in the bottom-left "missed warnings" quadrants, where the expert had objections but Governance AI did not issue any.

In the **e-commerce domain**, the expert argued that all access requests are reasonable and would realistically be accepted. However, this assessment assumed that personal data used for analytics had been collected with user consent for marketing and third-party sharing. At the expert's company, data was collected centrally and only made available for analytics after filtering for appropriate consent. This is common in organizations with centralized data architectures. In contrast, a decentralized data mesh shares data in a more federated way, requiring each data owner to ensure privacy-preserving sharing. In such settings, global assumptions about user consent typically cannot be made. Governance AI was designed for this decentralized context and, accordingly, assessed the access requests without assuming prior consent. Despite this difference, Governance AI agreed with the expert in 74.1% of cases and issued warnings for the remaining 15 access requests.

In the **insurance domain**, the expert formulated warnings for 12 (26%) of the access requests and accepted the remaining 74%. Again, the Governance AI was stricter, producing warnings in 41 of 46 access requests, 29 of which the expert

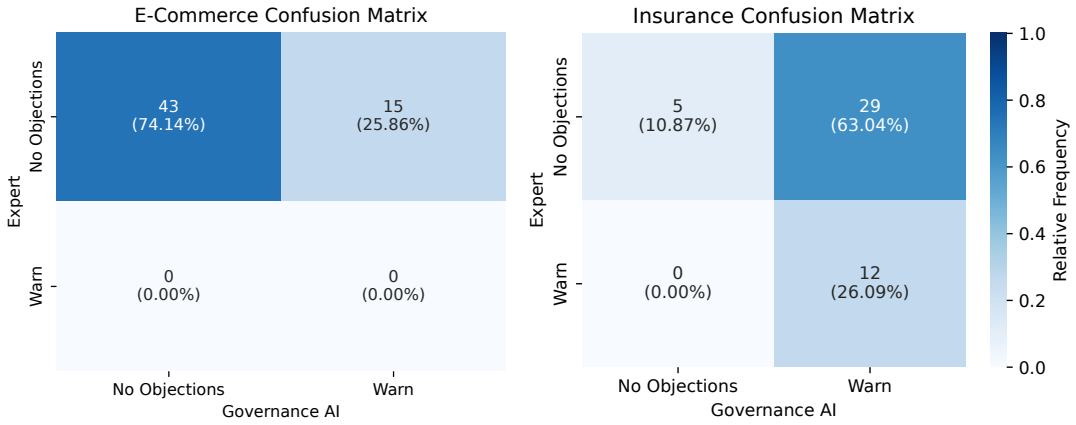


Figure 9: **The confusion matrices for the two domains show different patterns of agreement.** Crucially, the Governance AI never failed to issue a warning in cases where the expert deemed one necessary; however, it produced 3.6 times more warnings than the human experts.

found to be unproblematic.

Takeaway: Governance AI is generally stricter than the expert, leading to a higher number of warnings. Crucially, the Governance AI never fails to issue a warning in cases where the expert deems one necessary.

Usefulness of Governance AI Warnings and Suggestions

To better understand the differences in judgment between the experts and the Governance AI, we presented all access requests that received warnings to the experts again, this time including the warnings and the AI's suggested resolutions. Interestingly, in both domains, 80% of the AI-generated warnings were labeled as correct by the experts. However, only a minority, 25% in the insurance domain and 33% in the e-commerce domain, were considered entirely correct. The remaining warnings in this group were judged as generally valid but flawed in their reasoning. The experts classified the remaining 20% of warnings as incorrect. These were further divided into two categories: warnings based on generally valid concerns that did not apply in the specific context of the access request (8–13%), and warnings that addressed non-existent or entirely irrelevant issues. The distribution is shown in Figure 10(a).

The e-commerce domain expert remarked that *"[the AI] follows a GDPR-centric approach that always assumes the worst-case scenario. While this conservative stance is fundamentally not incorrect, the AI will consistently act as a perpetual naysayer."* We consider this a valuable property of Governance AI, as regulatory compliance is an explicit design goal and GDPR fines can be severe.

Governance AI also provided brief suggestions on how to improve access requests in response to each warning. The distribution of responses, shown in Figure 10(b), highlights key differences between the two domains. In the e-commerce domain, issues were relatively easy to resolve, for example, by limiting data access to specific fields or anonymizing data before processing. In contrast, the suggestions in the insurance domain were rated as less useful. While 53% were considered correct, the other 43% were deemed misleading, as they would not effectively address the underlying issues. The insurance domain expert commented: *"The warnings and suggestions are superficially acceptable but insufficient in detail and from a legal perspective."* A common pattern across both domains was that most suggestions proposed replacing PII with anonymized, non-PII data to resolve the warnings.

Takeaway: The warnings issued by the Governance AI are largely accurate; however, in a domain where privacy is highly relevant, it is hard to resolve policy violations in an easy way.

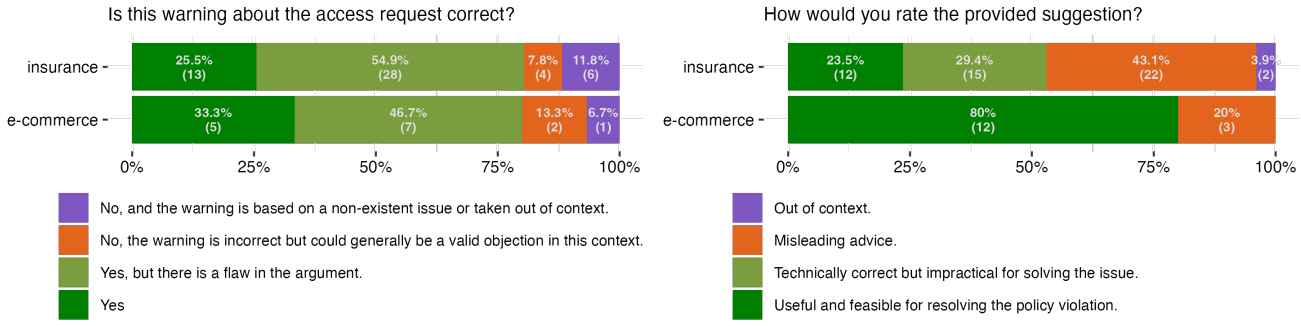
Discussion

We reflect on the implications of our findings and highlight key limitations and directions for future work.

Theoretical Implications

Since the majority of test cases were deemed realistic by domain experts, it is reasonable to assume that our generation process could yield an even higher number of realistic cases until reaching a point of saturation. To ensure thorough expert evaluation, we limited our analysis to 10 test cases per providing data product. This finding is consistent with previous research, which showed that 70% of LLM-generated use cases for facial recognition and analysis were already in existence, while 30% were classified as "upcoming" (Herdel et al. 2024).

Calibrating a warning system depends heavily on its real-world deployment, particularly on how users interact with it (Fu et al. 2020; Lee and See 2004). Excessive warnings may lead users to ignore them (Breznitz 1984), while insufficient sensitivity can result in missing critical issues, with potentially serious consequences (Hall 2000). In this study, false warnings had relatively minor consequences, as the final decision to approve a data-sharing request remained with the data owner, who could choose to disregard the warning.



(a) Expert assessment of the warnings

(b) Expert assessment of the suggestions

Figure 10: **The warnings issued by the Governance AI were largely accurate.** However, the more stringent data protection requirements in the insurance domain made it challenging to generate meaningful suggestions for resolving critical data access requests—whereas this was more feasible in the e-commerce domain.

Notably, in 26% of test cases in the e-commerce domain and 63% in the insurance domain, experts initially had no objections to access requests that the Governance AI flagged with warnings (top-right quadrant of the confusion matrices in Figure 9). However, after reviewing the warnings for these requests, both experts agreed in 80% of cases that the warnings were justified (Figure 10(a)).

On the other hand, failing to detect policy violations can lead to privacy breaches, with fines that vary depending on severity and jurisdiction (European Parliament and Council of the European Union 2016). Notably, our findings show that Governance AI never failed to issue a warning when the expert deemed one necessary (bottom-left quadrant of the confusion matrices in Figure 9). This indicates that the Governance AI was always at least as *"strict"* as the domain expert. Any attempts to calibrate the AI's warning threshold to increase the true *"no objections"* rate, 74% in the e-commerce domain but only 11% in the insurance domain, must ensure that this property is preserved to prevent over-reliance on the assistive system (Alberdi et al. 2009; Inagaki and Itoh 2013).

Practical Implications

This study presented an approach to automatically ensure policy compliance using generative AI by analyzing data access requests in a federated data mesh. We advanced the state of the art by proposing a system that automatically checks access requests. Governance AI produced more warnings than an expert would find, especially in the domain where protecting PII and other sensitive data is critical. The fact that Governance AI was at least as strict as human experts could be practically leveraged to introduce a higher level of automation, i.e., automatically approving access requests when Governance AI does not produce any warnings. While our results support this potential transfer of decision-making to the system, we caution against adopting this idea without careful consideration. A conclusive assessment would likely require further studies and additional safeguards (Sheridan 2002; Lee and See 2004).

Interestingly, Governance AI always provided a suggestion for every warning, even if there was nothing that could be done to rectify the access request. It is similar to a person who, when asked for directions, offers an answer, even if they don't know the way. This behavior is an inherent property of current LLMs, and while it is an area of ongoing research (Zhang et al. 2024), we could not address it by rewriting the Governance AI prompts. Instead, it may be more effective to introduce an additional LLM step to qualify the suggestion, potentially removing it if categorized as inappropriate.

The automation around Governance AI creates new possibilities. It could be used to assess the impact of proposed policy changes, providing a fast feedback loop to legal and governance experts before those changes take effect. This process could lead to better policies. Moreover, Governance AI enables continuous monitoring of evolving data meshes for emerging policy violations. This allows detection of otherwise unnoticed violations, such as when new data-sharing restrictions affect already approved data flows.

Finally, based on suggestions from both Governance AI and domain experts, it should become standard practice for data products handling PII to offer a secondary, less sensitive output port containing anonymized or aggregated data. This alternative output would allow consumers to access data that meets their needs without compromising privacy, especially for use cases that can be fulfilled with non-sensitive data.

Limitations & Future Work

We evaluated 46 and 58 realistic use cases across two domains where data mesh architectures are increasingly adopted. While these test cases cover many business processes within the two model companies, our findings are currently limited to these domains. We observed that data governance involves different utility-risk trade-offs across sectors, with e-commerce offering more flexibility than the more restricted life insurance industry. Therefore, it would be interesting to extend this study to further domains, such as business-to-business domains, and also domains that have been impacted by increasing privacy regulation, such as finance, and healthcare (Layton and Elaluf-Calderwood

2019). Due to limited access to detailed real-world company data architectures, we based our tests on the data products of two model company data meshes. This approach makes the results less realistic, yet more generalizable by avoiding company-specific data handling practices. In future work, we plan to develop data governance auditing methods building on the contributions of Nabar et al. (2008). For this purpose, it would be interesting to develop test coverage metrics for the data contracts and privacy policies. This would also give more informed indications about a "sufficient" number of test cases.

Analyzing the workflows of data product owners, requesters, or legal teams in formulating, assessing, and deciding on data access requests was beyond the scope of this paper. Our analysis focused on the realism of access requests rather than their formulation. In practice, different authors formulate access requests in different ways. Future research could quantify the benefits and drawbacks of AI-based computational governance, considering human factors, business needs, efficiency, and user acceptance (Allaham, Kieslich, and Diakopoulos 2025).

Additionally, further study is needed to explore how users with varying expertise levels interact with the Governance AI system. This is especially important given potential sociotechnical risks, such as user overreliance and long-term habituation, which may cause unintended consequences that require understanding over time.

Conclusions

In this paper, we proposed Governance AI, an LLM-based approach to automating data governance by evaluating data access requests against privacy policies and data contracts within a decentralized data mesh architecture. Our study showed that the system effectively identified policy violations, issuing more warnings than human experts while never missing critical cases. Although Governance AI applied stricter assessments, expert review confirmed that 80% of its warnings were valid. This demonstrated its potential to assist data governance experts in assessing many LLM-generated, yet realistic, data access requests.

Governance AI is a component of a data governance management tool capable of monitoring and managing an entire data mesh. The system was designed to consider all relevant policies without preprocessing and to track changes within the data ecosystem to prevent data breaches proactively. By leveraging the testing framework, operators of decentralized data architectures, such as multinational corporations with varying privacy policies across markets, could evaluate the effects of different policies through comprehensive test suites generated from the access request tool.

Additionally, Governance AI could act as an educational tool, guiding users in drafting data access requests that comply with regulations. It could suggest how to formulate access requests and use cases that meet regulatory requirements or recommend alternative data products with less sensitive information.

Despite our encouraging results, implementing computational data governance remains a complex challenge (Perrin and Broda 2024; Dolhopolov, Castelltort, and Laurent 2024). The ability to handle multilingual natural language showed the potential of generative AI in data governance. Our Governance AI tool and testing framework represent a step toward more effective implementations of computational governance in modern enterprise data architectures.

Ethical Considerations Statement

This work explores the use of Generative AI, specifically LLMs, to assist with automated data governance in enterprise environments. While our system, Governance AI (classified as "limited risk" under the EU AI Act (European Union 2024)), does not make final decisions, it provides policy violation warnings that could influence decisions regarding data access. To mitigate ethical risks, human oversight is embedded as a design principle, ensuring that accountability and final decision-making authority remain with human decision-makers.

The transfer of information for the purposes of checking access requests is governed by the individual contracts between users of Data Mesh Manager, as well as the terms of services the platform has with LLM providers.

Acknowledgements

The authors sincerely thank Martin Meermeyer and Matthias Wegmüller for their invaluable expert contributions to the evaluation of this work.

References

Alberdi, E.; Strigini, L.; Povyakalo, A. A.; and Ayton, P. 2009. Why Are People's Decisions Sometimes Worse with Computer Support? In Buth, B.; Rabe, G.; and Seyfarth, T., eds., *Computer Safety, Reliability, and Security*, 18–31. Berlin, Heidelberg: Springer.

Allaham, M.; Kieslich, K.; and Diakopoulos, N. 2025. Global Perspectives of AI Risks and Harms: Analyzing the Negative Impacts of AI Technologies as Prioritized by News Media. arXiv:2501.14040.

Amaral, O.; Abualhaija, S.; Torre, D.; Sabetzadeh, M.; and Briand, L. C. 2022. AI-Enabled Automation for Completeness Checking of Privacy Policies. *IEEE Transactions on Software Engineering*, 48(11): 4647–4674.

Bignotti, C.; and Camassa, C. 2024. Legal Minds, Algorithmic Decisions: How LLMs Apply Constitutional Principles in Complex Scenarios. *AAAI/ACM Conference on AI, Ethics, and Society*, 7(1): 120–130.

Bitol. 2025. *Open Data Contract Standard (ODCS)*. LF AI & Data Foundation. Version 3.0.1.

Bode, J.; Kühl, N.; Kreuzberger, D.; and Holtmann, C. 2024. Toward Avoiding the Data Mess: Industry Insights From Data Mesh Implementations. *IEEE Access*, 12: 95402–95416.

Borovits, N.; Kumara, I.; Tamburri, D. A.; and Van Den Heuvel, W.-J. 2023. Privacy Engineering in the Data Mesh: Towards a Decentralized Data Privacy Governance Framework. In *International Conference on Service-Oriented Computing*, 265–276. Springer.

Both, A.; Kastner, T.; Yeboah, D.; Braun, C.; Schraudner, D.; Schmid, S.; Käfer, T.; and Harth, A. 2024. AuthApp – Portable, Reusable Solid App for GDPR-Compliant Access Granting. In Stefanidis, K.; Systä, K.; Matera, M.; Heil, S.; Kondylakis, H.; and Quintarelli, E., eds., *Web Engineering*, 199–214. Cham: Springer. ISBN 978-3-031-62362-2.

Breznitz, S. 1984. *Cry wolf: The psychology of false alarms*. Psychology Press, 1 edition. ISBN 9780898592962.

California State Legislature. 2018. California Consumer Privacy Act of 2018.

Christ, J.; and Harrer, S. 2024. Data Contract Specification. https://datacontract.com. [Online; accessed 1 Aug 2025.

Dedecker, R.; Slabbinck, W.; Wright, J.; Hochstenbach, P.; Colpaert, P.; and Verborgh, R. 2022. What's in a Pod? – A knowledge graph interpretation for the Solid ecosystem. In Saleem, M.; and Ngonga Ngomo, A.-C., eds., *6th Workshop on Storing, Querying and Benchmarking Knowledge Graphs*, volume 3279 of *CEUR Workshop Proceedings*, 81–96.

Dehghani, Z. 2020. Data mesh principles and logical architecture. https://martinfowler.com/articles/data-mesh-principles.html. [Online; accessed 1 Aug 2025.

Dehghani, Z. 2022. *Data Mesh: Delivering Data-Driven Value at Scale*. O'Reilly Media, Inc. ISBN: 9781492092391.

Dolhopolov, A.; Castelltort, A.; and Laurent, A. 2024. Implementing Federated Governance in Data Mesh Architecture. *Future Internet*, 16(4).

European Parliament and Council of the European Union. 2016. The General Data Protection Regulation. Official Journal of the European Union (OJ).

European Union. 2024. Artificial Intelligence Act. Official Journal of the European Union (OJ).

Fu, E.; Johns, M.; Hyde, D. A. B.; Sibi, S.; Fischer, M.; and Sirkin, D. 2020. Is Too Much System Caution Counterproductive? Effects of Varying Sensitivity and Automation Levels in Vehicle Collision Avoidance Systems. In *CHI Conference on Human Factors in Computing Systems*, CHI'20, 1–13. New York, NY, USA: ACM.

Goedegebuure, A.; Kumara, I.; Driessen, S.; Van Den Heuvel, W.-J.; Monsieur, G.; Tamburri, D. A.; and Nucci, D. D. 2024. Data Mesh: A Systematic Gray Literature Review. *Computing Surveys*, 57(1): 1–36.

Hall, S. 2000. Psychological consequences for parents of false negative results on prenatal screening for Down's syndrome: retrospective interview study. *BMJ*, 320(7232): 407–412.

Herdel, V.; Šćepanović, S.; Bogucka, E.; and Quercia, D. 2024. ExploreGen: Large Language Models for Envisioning the Uses and Risks of AI Technologies. *AAAI/ACM Conference on AI, Ethics, and Society*, 7: 584–596.

Hilario, E.; Azam, S.; Sundaram, J.; Imran Mohammed, K.; and Shanmugam, B. 2024. Generative AI for pentesting: the good, the bad, the ugly. *International Journal of Information Security*, 23(3): 2075–2097.

Inagaki, T.; and Itoh, M. 2013. Human's Overtrust in and Overreliance on Advanced Driver Assistance Systems: A Theoretical Framework. *International Journal of Vehicular Technology*, 2013(1): 1–8.

Joshi, D.; Pratik, S.; and Rao, M. P. 2021. Data governance in data mesh infrastructures: the Saxo Bank case study. In *21st International Conference on Electronic Business (ICEB)*, 599–604.

Lai, J.; Gan, W.; Wu, J.; Qi, Z.; and Yu, P. S. 2024. Large language models in law: A survey. *AI Open*, 5: 181–196.

Layton, R.; and Elaluf-Calderwood, S. 2019. A Social Economic Analysis of the Impact of GDPR on Security and Privacy Practices. In *12th CMI Conference on Cybersecurity and Privacy*, 1–6. IEEE.

Lee, J. D.; and See, K. A. 2004. Trust in Automation: Designing for Appropriate Reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1): 50–80.

Li, J.; and Maiti, A. 2025. Applying Large Language Model Analysis and Backend Web Services in Regulatory Technologies for Continuous Compliance Checks. *Future Internet*, 17(3): 100.

Machado, I. A.; Costa, C.; and Santos, M. Y. 2022. Data Mesh: Concepts and Principles of a Paradigm Shift in Data Architectures. *Procedia Computer Science*, 196: 263–271.

Nabar, S. U.; Kenthapadi, K.; Mishra, N.; and Motwani, R. 2008. *A Survey of Query Auditing Techniques for Data Privacy*, 415–431. Boston, MA: Springer.

OpenAI. 2025. Prompt Engineering – Enhance results with prompt engineering strategies. https://platform.openai.com/docs/guides/prompt-engineering. [Online; accessed 1 Aug 2025.

Oppold, S.; Fritz, M.; and Woltmann, L. 2025. Data Contracts to Leverage (De-)centralized Data Management in Manufacturing Industries: An Experience Report. In *Datenbanksysteme für Business, Technologie und Web*, BTW'25, 731–743. GI.

Perrin, J.-G.; and Broda, E. 2024. *Implementing Data Mesh*. O'Reilly Media. ISBN 9781098156220.

Ramos, I.; Santos, M. Y.; Joshi, D.; and Pratik, S. 2024. Data Mesh Adoption: A Multi-case and Multi-method Readiness Approach. In Papadaki, M.; Themistocleous, M.; Al Marri, K.; and Al Zarouni, M., eds., *Information Systems*, 16–29. Cham: Springer.

Sambra, A. V.; Mansour, E.; Hawke, S.; Zereba, M.; Greco, N.; Ghanem, A.; Zagidulin, D.; Aboulnaga, A.; and Berners-Lee, T. 2016. Solid: a platform for decentralized social applications based on linked data. Technical report.

Schultze, M.; and Wider, A. 2021. *Data Mesh in Practice*. O'Reilly Media, Inc. ISBN: 9781098108496.

Sheridan, T. B. 2002. *Humans and Automation: System Design and Research Issues*. USA: Wiley. ISBN 0471234281.

Shin, W. 2024. Utilizing Generative AI for Test Case Generation: Comparative Analysis and Guidelines. *International journal of advanced smart convergence*, 13(4): 145–154.

Wider, A.; Harrer, S.; and Dietz, L. W. 2025. AI-Assisted Data Governance with Data Mesh Manager. In *International Conference on Web Services*, ICWS'25. IEEE.

Wider, A.; Jarmul, K.; and Akhtar, A. 2024. Towards Automating Federated Data Governance. In *International Conference on Web Services*, ICWS'24, 10–19. IEEE.

Wider, A.; Verma, S.; and Akhtar, A. 2023. Decentralized Data Governance as Part of a Data Mesh Platform: Concepts and Approaches. In *International Conference on Web Services*, ICWS'23, 746–754. IEEE.

Wolff, J.; and Atallah, N. 2021. Early GDPR Penalties: Analysis of Implementation and Fines Through May 2020. *Information Policy*, 11: 63–103.

Zhang, H.; Diao, S.; Lin, Y.; Fung, Y.; Lian, Q.; Wang, X.; Chen, Y.; Ji, H.; and Zhang, T. 2024. R-Tuning: Instructing Large Language Models to Say 'I Don't Know'. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 7113–7139. Mexico City, Mexico: ACL.