H2020-SFS-2018-2020

DECIDE

Data-driven control and prioritisation of non-EU-regulated contagious animal diseases

Deliverable 1.3

Testing and evaluation of alternative approaches for data access

WP1 – Data identification, characterisation and acquisition

Authors Saba Noor (UGent), Céline Faverjon (EpiMundi), Miel Hostens (UGent)
Lead participant UGent
Delivery date 03 July 2025
Dissemination level Public
Type Report

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 101000494.

Revision History

Author Name (Partner short name)	Description	Date
Miel Hostens (UGent), Saba Noor (UGent)	Draft deliverable	18.06.2025
Céline Faverjon (EpiMundi)	Revision	24.06.2025
Saba Noor (UGent)	Updated version	26.06.2025
Gerdien van Schaik (UU)	Revision	27.06.2025
Miel Hostens (UGent), Saba Noor (UGent)	Final draft	02.07.2025
Johannes Ripperger (accelCH)	Final checks and formatting	03.07.2025

Content

Abbreviations

Abbreviation	Description
BCV	Bovine coronavirus
BRSV	Bovine respiratory syncytial virus
EU	European Union
FAIR	Findable, Accessible, Interoperable, and Reusable
GDPR	General Data Protection Regulation
H2020	Horizon 2020
HS	Histophilus somni
IB	Infectious bronchitis
LHO	Livestock Health Ontology
M	Milestones
O	Objectives
ODKFADM	Ontology-driven knowledge-based framework of Farm Animal Data Management
OWL	Ontology web language
PLF	Precision livestock farming
RDF	Resource Description Framework
SET	Surveillance Evaluation Tool
SI	Swine influenza
SPARQL	SPARQL Protocol and RDF Query Language (recursive acronym)
T	Task
TCO	Total cost of ownership
WOAH	World Organization for Animal Health
WP	Work Package

Partner short names

Short name	Organisation
UU	Universiteit Utrecht
UGent	Universiteit Gent
SVA	Swedish Veterinary Agency
accelCH	accelopment Schweiz AG
EpiMundi	EpiMundi (formerly AUSVET)

Executive Summary

Deliverable D1.3 supports DECIDE's goal of improving data-driven livestock health decisions by comparing three data access models, Direct, Centralized, and Decentralized (Solid Pods). It defines best practices for managing distributed veterinary data based on performance, privacy, scalability, and FAIR-compliance.

Objectives of the deliverable

- Assess and compare three data access approaches: Direct, Centralized federated access, and Decentralized federated access (using Solid Pods), for veterinary surveillance within the DECIDE project.
- Define FAIR- and GDPR-aligned criteria for structured evaluation.
- Identify best practices for scalable, secure, and semantically consistent data sharing.
- Develop practical recommendations to support compliant and sustainable data reuse in livestock health.
- Inform governance, ethics, and technical progress in WP2, WP3, and WP5.

Current activities:

- Assessed the three data access models using 11 criteria grounded in FAIR and GDPR principles (Tasks 1.1–1.3).
- Tested all approaches using the Cattle Barometer case study with data from GD, DGZ and PathoSense, ARSIA, and the Irish labs, benchmarked performance and integration trade-offs.
- Delivered key resources: species-specific ontologies (D1.2), Ontologies training via ECPLF 2024 workshop Program | Agenda, annotated Jupyter notebooks, and the Cattle Barometer tutorial.
- Developed tools and infrastructure include the centralized pipeline GitHub Cattle Barometer, tutorials DECIDE Cattle Barometer, federated solid pod deployment, and federated SPARQL query API, and draft guidelines supporting FAIR and privacy-conscious data sharing in veterinary surveillance.
- These insights support DECIDE's broader mission by guiding governance, ethics, and sustainability discussions across WP2, WP3, and WP5.

Outcome

- Centralized federated access offered strong automation and performance but required standardized formats and limited data ownership.
- Decentralized federated access provided better privacy and semantic alignment but introduced latency as data and pod count increased.
- Direct sharing was the simplest and useful in the early phase of the Cattle Barometer, but lacks automation and scalability for continuous, real time multi-labs data
- M12: Federated data access was implemented through both centralized and decentralized setups using cattle diagnostic data. Decentralized access used Solid Pods and federated SPARQL, while centralized integration enabled benchmarking via a harmonized and integrated automated pipeline.

Next steps

- Focus shifts to federated learning prototypes that use ontology-driven knowledge discovery to predict future trends.
  M13: Federated learning is in progress, building on RDF-formatted cattle diagnostic data from the Cattle Barometer. It benchmarks prediction on centrally aggregated data versus a privacy-preserving federated learning setup using local model updates. Completion is expected by the end of 2025, ahead of the next General Assembly.

1 Introduction

The DECIDE project supports data-driven decision-making in livestock health management by enhancing the accessibility, sharing, and interpretation of surveillance data. Effective disease control depends not only on the availability of diagnostic data but also on the mechanisms through which that data is exchanged among veterinary laboratories, stakeholders, and digital tools (Van Schaik et al., 2023).

This deliverable, part of WP 1 task 1.3, evaluates and compares three procedures for accessing distributed livestock health data: Direct data sharing, Centralized federated data access, and Decentralized federated data access using a decentralized technology. It is important to note that this work focuses on federated access, not federated learning, which involves distributed model training and is planned for future implementation. The comparison focuses on the underlying architecture and data flow, as well as the tools and technologies used for implementation. It also includes a discussion of strengths and limitations, particularly in terms of scalability, privacy, ease of use, and implementation.

The Cattle Barometer case study, a surveillance tool designed to visualize pathogen-specific test results from cattle across European laboratories, serves as the real-world example for this analysis. The Cattle Barometer (https://decide-project-eu.github.io/case-studies-website/case-studies/cattle-barometer.html) evolved across all three access models. Initially implemented using manual data sharing (Direct), it later transitioned to a centralized, automated ingestion system and is currently being explored using decentralized, federated querying. Each stage of evolution highlights the opportunities and limitations associated with different data access strategies in real-world research and analytics settings. The following sections describe the decentralized architecture, experimental evaluation, and a comparative discussion of these three access procedures within the DECIDE framework. All the code documenting the development, testing and evaluation of alternative approaches for data access is made available here: https://github.com/decide-project-eu/data-access-guidelines-best-practices-wp1-decide.

2 Methodology

Figure 1 presents the classification of data access procedures considered in DECIDE WP1. Each approach is examined based on twelve evaluation criteria: data access, data privacy, data integration, data consistency, data scalability, data interoperability, data reusability, data findability, data availability, ease of use and implementation, total cost of ownership (TCO) and performance.

These evaluation criteria were developed based on insights from the DECIDE project, particularly Tasks 1.1 to 1.3, and are informed by the FAIR (Findable, Accessible, Interoperable, Reusable) principles (Wilkinson et al., 2016). While FAIRness is technically a property of data, in this deliverable we apply FAIR as an evaluation lens to understand how different data access approaches support or limit the production, maintenance, or reuse of FAIR-aligned data. In other words, we do not label an approach itself as FAIR or non-FAIR but rather examine whether it enables or undermines FAIR data practices across the data sharing lifecycle.

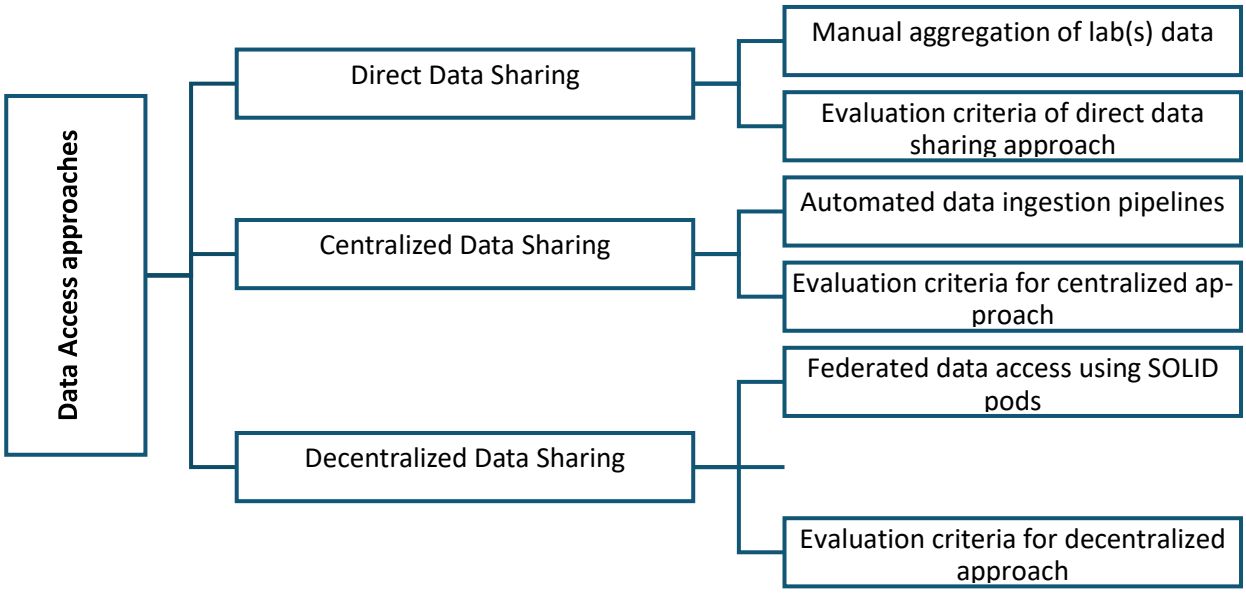

Figure 1. Classification of data access approaches: direct, centralized, and decentralized.

This approach is further supported by findings from Meyer et al. (2021), who identified major gaps in FAIR compliance across veterinary epidemiology datasets, especially in interoperability, metadata quality, and reusability. Similarly, Delavenne et al. (2025) highlight real-world challenges in livestock health data sharing, including inconsistent formats, limited metadata use, and semantic alignment barriers, and concerns around data privacy and compliance with regulations such as the GDPR. Together, these sources reinforce the need for a structured and context-aware evaluation of data access models in this domain.

Data access includes the processes used by the data analyst to reach the data from the data provider, including data sharing agreements and data sharing tools. Data privacy addresses the degree of control and protection data providers retain over sensitive information. Data integration investigates how the data can be integrated with others, depending on the approach used to access it. Data consistency assesses how the three approaches contribute to having data consistent, accurate, and valid over time. Data scalability evaluates the system's ability to accommodate additional data, users, or institutions without major redesign. Interoperability in a data access approach means making sure that data can be easily shared and understood between different systems and people. This involves using common formats, consistent structures, and agreed terms (like shared vocabularies or coding schemes). It helps reduce confusion, avoids errors during

integration, and minimizes the need for manual adjustments, especially in projects involving multiple partners or data sources. Reusability in a data access approach means making it easy for others to use the same shared data again for new projects or analysis, without needing to reprocess it or ask for clarification. A reusable approach includes enough context (like clear structure, documentation, and meaning), so users can understand and work with the data over time, across teams, or in different tools, without starting from scratch. Data findability means, does the approach help users quickly locate the correct version of a dataset when needed? A good data sharing method includes indexing, clear naming, and storage in a common place. This avoids confusion like "Which version is latest?" or "Where is the file saved?" It makes coordination smoother, especially in large teams or multi-country collaborations. Data availability in the context of data access approaches refers to how reliably and consistently the shared data can be accessed when needed. This includes whether the data is up-to-date, online, and reachable without repeated manual requests or delays.

Ease of use and implementation reflects how technically demanding each approach is for both data providers and analysts, including setup, maintenance, and day-to-day usability. TCO criteria refers to the overall effort, time, and money needed to set up and maintain a data-sharing approach. While "ease of use and implementation" focuses on how simple a system feels for users, TCO provides a broader view of what it truly costs to operate that system over time. TCO is shaped by three key factors. First, personnel cost, the amount of human effort required for setup, data cleaning, coordination, and ongoing support. Second, infrastructure cost, the expenses related to cloud storage, servers, and any specialized tools such as Solid Pods or automated data pipelines. Third, maintenance cost, the continuous effort needed to keep the system running smoothly, including tasks like updating queries, fixing issues, and managing access. Together, these elements help assess the long-term sustainability and practicality of each data access model. The performance criteria investigate how each approach performs in accomplishing various tasks on the data. The first eleven criteria are assessed based on the output of the work done for the rest of the DECIDE project. The last one, i.e., performance, is investigated in more detail in this report using experimental settings. The results are then discussed in light with DECIDE's broader goals and used as a basis to develop guidelines and recommendations using the knowledge acquired during the project for supporting broad, scalable, and privacy-conscious data sharing and use in animal health.

To ensure consistency, all models were applied to the same Cattle Barometer use case. Pathogen-specific test data were initially received from multiple laboratories via direct data sharing (e.g. emailed spreadsheets or shared folders). These raw files were then cleaned and harmonized into a structured schema, which served as the baseline dataset across all three access models:

- **Direct data sharing** involved manual exchange of these structured files, which were then uploaded directly into Tableau for visualization.
- **Centralized federated access** used the same standardized dataset, processed through an automated pipeline hosted on a shared cloud server.
- **Decentralized federated access** distributed the same schema-aligned data across Solid Pods, enabling query-based access using federated tools like the Communica engine.

Each model was then evaluated using the twelve criteria previously defined in the methodology section, drawing on practical insights from the Cattle Barometer use case. These assessments helped identify strengths, limitations, and real-world applicability across different data access strategies. Performance testing, focused on query response time, was applied only to the centralized and decentralized models, as direct sharing involves manual processes not suitable for automated benchmarking.

3 Data access approaches

3.1 Direct data sharing approach

3.1.1 Overview

Direct data sharing refers to manual or semi-manual exchange of data between data providers and data analysts, usually in the form of spreadsheets or CSV files sent by email or uploaded to shared drives. This if often the approach used by default to share data in research projects and was the initial foundation of the Cattle Barometer use case (Bokma et al., 2023). In the European Veterinary Barometer for Bovine Respiratory Diseases (BRD), diagnostic test results from labs in Belgium, France, the Netherlands, and Ireland were anonymized and manually aggregated into a combined CSV structure for visualization. The initial tutorial developed for the first version of the tool can be accessed at https://decide-project-eu.github.io/case-studies-website/tutorials/cattle-barometer.html. As shown in Figure 2, the workflow involved raw data cleaning using tools like Excel, R, or Python, followed by storage in static formats without structured access control. Aggregated insights were then visualized using platforms such as Tableau.

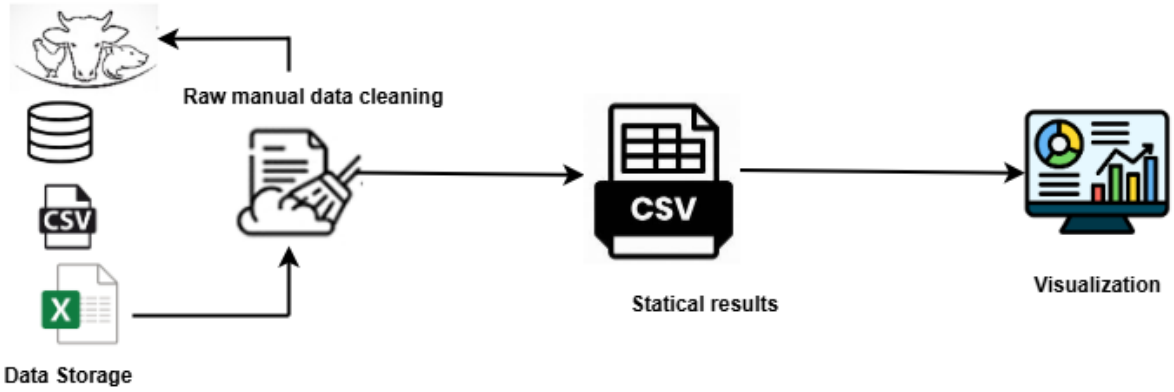

Figure 2. Direct data access workflow, adapted from (Noor et al., 2024).

3.1.2 Evaluation criteria of direct data sharing approach

Data access: One of the main advantages of direct data sharing is its simplicity in terms of data access. It is easy, quick, and cheap to set up as it requires no complex infrastructure or integration platforms to start sharing and accessing data. Data providers can participate by simply exporting data in spreadsheet formats and sharing it via common channels such as email or cloud folders. From a data sharing agreement perspective, this approach is also quite straight forward as it involves only bilateral agreements usually easier to set up than agreements between multiple stakeholders. The simplicity of the approach comes however at the cost of time. Indeed, the absence of automatic steps to get access to the data can make this task rapidly time-consuming for both the data providers and the data analysts.

Data integration: In this setup, the burden of data cleaning and integration usually relies mostly or only on the data analyst. It has advantages for the data provider who does not need to invest much before making their data accessible to the data analyst (e.g., no specific data format is needed, no specific infrastructure either). However, it comes with challenges for the data analyst because data cleaning and integration of heterogeneous data can be very complex and quickly become time-consuming.

Data privacy: Data analysts can end up being the sole guardians of data privacy if they are also in charge of data anonymization. When done manually on an ad hoc basis, it can be associated with errors or inconsistencies and therefore with risks for data privacy.

Data consistency: Despite its simplicity, direct data sharing is labor-intensive and prone to human error due to the manual nature of data cleaning and aggregation, which may cause issues in terms of data consistency. Moreover, direct data sharing lacks real-time access, as data is shared in static files and must be updated manually. This is causing risks in terms of data consistency, as each new data upload may not be consistent with the previous one.

Data scalability: Direct data sharing lacks scalability due to its reliance on manual processes. As the number of data providers increases, so does the variety of data formats, schemas, and quality levels, each requiring individualized cleaning and harmonization. This not only increases the workload for analysts but also introduces bottlenecks, especially when updates are frequent or time sensitive. For example, in the Cattle Barometer use case, integrating CSV files from multiple labs required repeating anonymization and transformation steps manually for each new file submission. Moreover, the absence of automated data pipelines means there is no efficient way to monitor new incoming data, validate files, or update visualizations in real-time. Instead, each step—requesting data, downloading, cleaning, merging, and uploading- is repeated manually, making it infeasible for continuous data streams or for scaling to dozens of labs or species. This results in delays, inconsistent update cycles, and significant coordination overhead, which makes the model unsuitable for long-term or large-scale surveillance systems.

Data findability: In DECIDE, data files such as "DECIDE_MTA_UGENT_14nov2022.xlsx", and "DECIDE_xxLab_pig_diagnostic_data_for_ontologies_12102023.xlsx" were exchanged directly via email or shared drives. Although these files sometimes included structured schemas or explanatory notes, that contextual information was often lost during the file-sharing process. Key details such as schema definitions, descriptions, or version history were usually stored separately, for example, in emails or known only to individuals, and were not embedded within the files themselves. As a result, team members often struggled to identify the most recent version, leading to repeated requests and delays. The absence of a central location for storing or searching shared files made it difficult to trace what had already been exchanged. This lack of structure significantly reduced the findability of shared files across institutions.

Data availability: In the Cattle Barometer use case, data availability was often delayed due to the manual nature of stakeholder coordination. Laboratories needed to be contacted individually for updated files, and response times varied, sometimes significantly, due to staff workload or availability. This reliance on human action introduced bottlenecks and made timely data integration unpredictable, reducing the reliability of the overall system for near-real-time surveillance.

Data interoperability: The interoperability of directly shared data is constrained by inconsistent formatting and the lack of enforced standards. Even when files are created with some structure, variations in naming conventions, column formats, or coding practices across contributors introduce semantic mismatches. Since there is no system to align schemas or ensure shared vocabularies, integrating data from multiple sources typically requires manual mapping and cleaning. This limits machine-readability and slows down scalable, automated integration of shared files and resources across systems.

Data reusability: Reusability is low because, even when data files are shared with documentation, metadata, and a clear structure, they are still hard to reuse later. This is because important information is often kept in separate places, like emails or notes, and not saved together with the file. Once the file is saved, copied, or changed, that extra information can be lost. There is usually no way to track updates or know where the data came from. This makes it difficult for others to understand or reuse the data later without asking the original sender again.

Ease of use and implementation: For data providers, direct sharing is simple and requires no technical setup. However, handling and sending files manually again and again can become tiring over time. For analysts, the diversity of data formats used means extra work is needed to clean and prepare the data each time it is received.

TCO for direct data sharing: In direct data sharing setup, most of the cost comes from personnel time. Since data is shared manually, people must spend time cleaning, merging, and interpreting files every time something changes. Infrastructure needs are low, because basic tools like email or cloud folders (e.g. Dropbox) are enough. However, the lack of automation means staff must repeat the same work for every new dataset. Maintenance is also high, since there's no central system to keep things organized or updated automatically.

3.2 Federated data access: Centralized approach

3.2.1 Overview

In the centralized data access model, implemented in the later stages of the Cattle Barometer use case, the DECIDE project transitioned from manual data exchange to a unified, automated infrastructure. Diagnostic results from multiple laboratories were collected through automated ingestion pipelines, which cleaned and harmonized the incoming data to reduce human error and improve scalability. As illustrated in Figure 3, these datasets were funnelled into a central computing environment for integration and analysis.

The raw laboratory data, typically in CSV, PDF, or Excel format, was cleaned and standardized using custom R scripts developed for each lab. These scripts performed filtering, renaming, mapping, and normalization of variables to conform to a unified schema. Once processed, all five datasets were merged on the centralized server into a single flat-file CSV, structured with 10 standardized fields to support downstream analysis. This centralized format enabled fast data processing and was directly compatible with visualization platforms like Tableau. Access the source code at https://github.com/decide-project-eu/cattle-use-case-barometer.

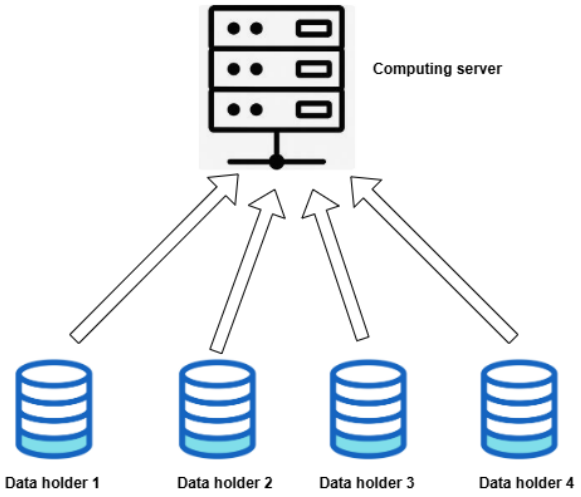

Figure 3. Centralized data access workflow.

3.2.2 Evaluation criteria for the centralized approach

Data access: Centralized data access implies that data providers hand over their data to a central system. This can ultimately be more convenient for the data providers and analysts as automated pipelines may save them a large amount of time. However, it requires the implementation of a central system, which may take time from a technical perspective. Furthermore, such centralized data access raises concerns regarding privacy, security, and user autonomy (Ragab et al., 2024). This often leads to challenges in setting up appropriate legal frameworks to make such infrastructures work in practice.

In the Cattle Barometer use case, centralized data access was enabled through bilateral agreements between DECIDE joint data controller and participating laboratories. These contracts outlined the terms of data contribution, processing, and usage, and were further supported by the overarching DECIDE consortium agreement.

Data privacy: While centralized data access streamlines processing, it raises privacy concerns regarding data sovereignty, unauthorized access, or data misuse. For many providers, giving up control over sensitive diagnostic data may be a barrier, especially in multi-institutional or cross-border contexts. Robust data governance, clear access control policies, and legal agreements are necessary to mitigate privacy risks and increase trust in centralized sharing environments. Aggregating data at the regional level can help reduce privacy sensitivity, but this is not always sufficient, especially in cases where datasets are small or sparsely distributed, increasing the potential for re-identification even within aggregated information.

In the Cattle Barometer use case, the diagnostic data was aggregated at the regional level, which significantly reduced privacy sensitivity and made centralized processing more acceptable to data providers. However, as discussed in Chapter 4, privacy was further addressed through technical measures such as SHA-256 anonymization of sensitive identifiers. While effective for preventing re-identification, this approach also introduced limitations, especially around data reusability and interoperability. Thus, highlighting the need for careful trade-offs between privacy and long-term utility in centralized systems.

Data integration: Centralized data access allows to automate all the steps related to data cleaning and integration, which significantly contribute to reducing human error. This also reduces the workload for the data analysts as manual tasks of data management are reduced. Moreover, it facilitates de facto data reuse because all the data is in a centralized place. However, this requires that incoming data follow a predefined structure, something often lacking in real-world veterinary datasets, as shown in Task 1.1 and highlighted by Delavenne et al. (2025), who note the widespread absence of metadata, standard vocabularies, and consistent data formats across livestock sources. This approach also introduces infrastructure dependency, as the central platform becomes a single point of failure and requires continuous costly maintenance and high availability.

Data consistency: The centralized model offers strong consistency benefits. All data passes through uniform scripts and is mapped to a standardized schema before storage, ensuring harmonized structure and terminology across labs. The automated ingestion process eliminates manual merging, reducing errors and ensuring stable, version-controlled datasets suitable for longitudinal analysis.

Data scalability: Centralized systems offer moderate scalability. New labs can be added to the pipeline, but only if they adhere to the predefined data format or if additional scripts are developed for their specific structure. This format dependency poses a bottleneck: while adding more data is technically feasible, variability in incoming file structures can limit the ability to scale efficiently. The system also relies on sufficient server resources to handle increasing data volume and update frequency.

Data findability: Data stored centrally can be catalogued and indexed, enabling easy search and retrieval. This makes it easier for teams to locate specific datasets, monitor updates, and avoid duplicative efforts. Dashboard integrations further enhance visibility and access for stakeholders. However, in the current DECIDE implementation, the centralized endpoint is not publicly exposed and sits behind firewalls, which limits external discoverability. Full findability benefits would require a public or authenticated metadata catalog and accessible endpoints.

Data interoperability: Interoperability is moderately supported. If the central system incorporates schema alignment or maps variables to shared ontologies during ingestion, semantic interoperability, the meaning

and interpretation of the data is conserved. However, this depends on explicit design choices. Without semantic enrichment, the system remains syntactically interoperable (through consistent structure) but not semantically aligned across domains or species.

Data availability: Unlike in manual sharing models, centralized systems offer higher availability, as data is stored in a shared location accessible to authorized users. The automated pipeline detects and integrates new files if they follow the agreed structure. However, availability is still dependent on platform uptime and accurate data submission by stakeholders. If files are not uploaded or deviate from the format, availability of processed data may still be delayed.

Data reusability: Cleaned and standardized datasets are highly reusable, especially when combined into a single harmonized output. Analysts can easily run additional queries or use the data in other tools without repeating the cleaning process. Reusability is maximized when the system includes metadata, data dictionaries, and persistent file storage.

Ease of use and implementation: Once the infrastructure is in place, the system is relatively easy to use for both data providers and analysts, assuming data is submitted in the expected format. Providers simply upload files to the central platform, and analysts receive clean, ready-to-visualize datasets via automated processing. However, setting up the system requires technical expertise in scripting, data transformation, and server management. Additionally, changes in data structure require manual intervention to update or extend the scripts, adding to long-term maintenance needs.

TCO for Centralized Data Sharing: In a centralized setup, the initial setup requires time and technical expertise to define data formats, develop processing scripts, configure cloud servers, and test dashboards. But once the system is running, personnel effort drops thanks to automation. Infrastructure costs (cloud storage, processing tools) remain ongoing, and some maintenance is needed to monitor performance and keep the pipeline updated. This model is more efficient over time, especially when data arrives in consistent formats.

3.3 Federated data access (Using Solid Pods)

3.3.1 Overview

The decentralized model involved full implementation on Solid Pods, with federated SPARQL queries executed using Communica and deployed via Azure functions, as shown in Figure 4. In contrast to the centralized architecture, the decentralized data access model using solid pods emphasizes data ownership, security, and real-time federation (Dedecker et al., 2022). Instead of aggregating all diagnostic data into a single repository, this model enables each laboratory to retain control over its own data. Each lab's data is stored on Solid Pods, personal online data stores developed using W3C web standards for identity, access control, and interoperability (Sambra et al., 2016). Solid Pods offer a standards-based mechanism for data publishing and sharing, empowering data providers to manage their information while enabling structured collaboration.

At the core of this architecture is the Solid Server, deployed on Azure at https://solidserver.bovi-analytics.com. Federated SPARQL queries are executed through a dedicated API endpoint: https://decide-federated-functions.azurewebsites.net/api/FedQuery, which is built using the Comunica SDK and deployed as a cloud-based Azure Function. This engine dynamically queries remote RDF (Resource Description Framework) datasets hosted on the Solid Pods at runtime, without pre-aggregating or relocating the data.

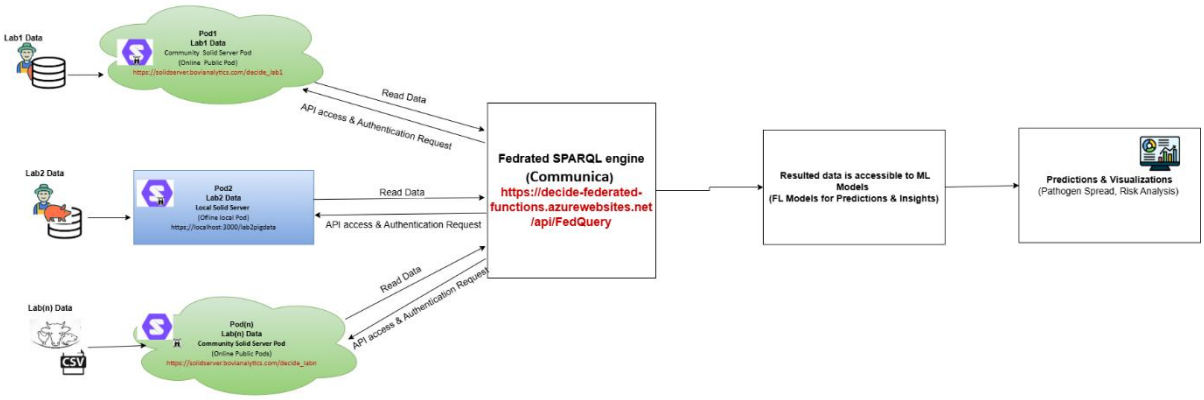

Figure 4. Decentralized data access workflow.

Each pod exposes livestock diagnostic data in RDF format, a graph-based structure for encoding knowledge using subject, predicate, and object triples. These are semantically aligned with the Livestock Health Ontology (LHO) to ensure a shared vocabulary across species and labs. When a query is submitted through the API, the engine traverses each pod, extracts relevant RDF triples based on the defined filters, and assembles the results into a structured response. These results can then be visualized in platforms like Tableau. This federated system was developed as part of the DECIDE project's goal to explore decentralized data exchange via federated access.

3.3.2 Evaluation criteria – Decentralized data access (Federated via Solid Pods)

Data access: Decentralized data access via Solid Pods offers real-time querying without transferring ownership of data. Each laboratory maintains its own data on a personal online data store (Solid pod), and authorized consumers can access it using a federated SPARQL query engine. This model enables live access to distributed data. Initial setup can be complex, but once in place, queries can be executed dynamically via the API without requiring file transfers or synchronization. Moreover, data sharing agreements are typically lightweight and provider specific. Because data remains in the control of each laboratory, there is no need to transfer ownership or manage multi-party legal agreements. Instead, each provider can define simple access conditions, such as who may query their data and under what constraints, through pod-level configuration or informal agreements. This reduces legal and administrative overhead while preserving full data sovereignty.

Data privacy: Data privacy is a core strength of this model. Because each data provider retains full control of their Solid pod, they determine who can access what information, and under what conditions. Access control is enforced through W3C-compliant policies, minimizing governance risks. No data is centrally stored or duplicated, and sensitive information remains under the direct control of its owner.

Data integration: Integration is achieved at query time through semantic alignment. All RDF datasets are structured using a shared ontology LHO, which enables consistent interpretation of data across different labs and species. However, unlike centralized models where integration is done upfront, in decentralized systems, integration is virtual and performed dynamically. This means integration is flexible and non-invasive, but it also requires well aligned schemas and careful ontology design to avoid mismatches or ambiguity during queries.

Data scalability: This architecture is highly scalable. New labs or species can be added simply by linking new pod Uniform Resource Locators (URLs), without changing the core system. For example, a new lab's data can be made accessible by adding its Solid Pod URL such as https://solidserver.bovi-analytics.com/decide_lab1/Vertical/RDFoutputCattleSampleLab1.ttl. The federated query engine can dynamically discover and query multiple pods in parallel, regardless of how many are connected.

Data findability: In decentralized systems using Solid Pods, data findability depends on how dataset URLs and metadata are structured, published, and made accessible. Solid Pods do not offer centralized indexing by default, so discoverability relies on explicit sharing, well-defined metadata, and standardized structuring. To improve discoverability, metadata from the schema.org vocabulary is embedded directly within each turtle file. Elements such as schema:Dataset, schema:name, schema:description, schema:creator, and schema:dateModified are now applied across all datasets. This enables indexing by linked data tools and semantic search engines, making datasets more easily findable even without a centralized catalog. When combined with public or controlled-access sharing policies, this setup supports FAIR-aligned findability in a decentralized, federated environment. This can be found here: solidserver.bovi-analytics.com/decide_lab1/Vertical/RDFoutputCattleSampleLab1.ttl

Data availability: In the decentralized setup, data is stored on several separate servers (called Solid Pods), each managed by a different lab or organization. If one of these servers is temporarily offline or has an issue, the system can still run the query and get results from the other servers that are working. This means the system doesn't completely fail — you just get partial results instead of everything. It makes the system more reliable overall, but if you want full results, all pods need to be available.

Data interoperability: In this architecture, each dataset is assigned a globally unique Uniform Resource Identifier (URI) and described using RDF aligned with shared ontologies like LHO. For instance, a specific cattle sample might be identified as *<http://www.purl.org/decide/LiveStockHealthOnto/LHO#Lab1CattleSample_0>.* Such URIs enable consistent interpretation and integration across distributed datasets, supporting semantic interoperability.

Data reusability: The decentralized model strongly supports data reusability. Since each pod exposes machine-readable, semantically structured data using global identifiers, the same data can be queried for multiple purposes without duplication. Reuse is further enhanced by ontology-based annotation and metadata, which provide context and meaning to each data point.

Ease of Use and Implementation: The decentralized model using Solid Pods is powerful, but it can be challenging to use without technical help. Once the system is running, analysts can access data from different sources in real time, without needing to collect files manually. However, using it requires writing SPARQL queries, a special type of computer language that is used to query the federated data.

TCO for Decentralized (Solid Pods) Federation: Setting up a decentralized system involves significant effort at the start. Skilled technical staff or data analysts are needed to configure Solid Pods, deploy the SPARQL query engine, and ensure that all data follows shared ontologies. However, once the system is up and running, the amount of manual work decreases thanks to automation. Infrastructure costs remain higher than in other models because each lab or partner requires its own hosted pod and server setup. While automated pipelines help reduce repetitive tasks, ongoing maintenance is still complex. You need to make sure pods stay online, monitor query performance, and keep ontologies in sync. This approach offers strong privacy and control, but it also requires more coordination, technical oversight, and long-term investment to keep everything working smoothly.

4 Experimental setup and evaluation of performance

Two key experiments were conducted to evaluate the performance data access models as shown in Table 1. Direct data sharing was not included in these experiments, as it involves manual data retrieval and local processing by analysts, making it unsuitable for benchmarking under automated or federated conditions. Instead, performance testing focused on the Centralized and Decentralized (federated) access approaches. Experiment 1 focused exclusively on the Cattle Barometer data, comparing the performance of centralized and decentralized data access approaches. It benchmarked data retrieval time, architecture, and integration complexity for retrieving cattle diagnostic results. Experiment 2 expanded the evaluation to test the scalability of decentralized data federation using a broader range of data, including cattle, pig, and poultry diagnostics. This experiment examined both vertical (species-specific) and horizontal (mixed-species) federated queries to understand the effects of data distribution and query complexity on performance.

4.1 Experiment 1: Centralized vs decentralized query performance (Cattle Barometer)

To benchmark data access architectures, an experiment was designed to compare centralized and decentralized querying approaches using diagnostic data from five European cattle laboratories. Table 1 shows centralized vs. decentralized Cattle Barometer data metrics.

In the **centralized setup**, each lab's raw Excel files were initially processed using dedicated R scripts for cleaning, filtering, SHA-256 anonymization, and mapping to a standardized cattle https://decide-project-eu.github.io/case-studies-website/tutorials/cattle-barometer.html. This process was automated through a GitHub-managed Quarto pipeline that detects incoming files, transforms them, and exports the cleaned data directly to the Barometer dashboard. The final dataset, consisting of 24,448 records across 10 standardized columns (~27.35 MB), was exported as a flat .csv file and integrated with Tableau https://github.com/decide-project-eu/case-studies-website. This centralized benchmark now serves as a reference point for evaluating federated query performance.

In the **decentralized setup**, the same cattle diagnostic data converted to RDF was uploaded to five Solid Pods hosted at https://solidserver.bovi-analytics.com/, with controlled read access managed via ACLs. Each lab maintained its RDF data independently, uploaded via the PENNY tool https://github.com/djsf-kobayashi/penny), and structured using the Livestock Health Ontology (LHO) to ensure semantic consistency. The RDF triple counts per lab were as Lab 1: 227,310 triples, Lab 2: 148,574 triples, Lab 3: 267,168 triples, Lab 4: 5,980 triples, Lab 5: 91,024 triples, and Total RDF triples across all five cattle pods: 740,056. A federated SPARQL query was executed via the Communica engine deployed on Azure (https://decide-federated-functions.azurewebsites.net/api/FedQuery) targeting Mycoplasma bovis-positive swabs. No pre-aggregation was done, and all data remained in their original pods. The full query execution time was 228.79 seconds. For demonstration purposes, the complete Jupyter Notebook implementation has been uploaded to the GitHub repository and is available here: https://github.com/decide-project-eu/data-access-guidelines-best-practices-wp1-decide.

Table 1. Centralized vs. decentralized (cattle only).

Metric	Centralized (CSV Pipeline)	Decentralized (Federated RDF Query)
Data Source	5 CSV files	5 Solid Pods
Total Data Size	~27.35 MB (24,448 rows)	~27.35 MB (740,056 RDF triples)
Query Type	Flat-file filter (R)	SPARQL query
Execution Environment	Local Quarto + GitHub	Azure Function + Communica
Total Query Time	≈ 14.21 seconds	≈ 228.79 seconds

Figure 5 clearly shows that centralized querying significantly outperforms decentralized querying in terms of data processing time. However, this comes at the expense of flexibility and real-time access. Decentralized querying introduces performance overhead but enables federated access to live, distributed data sources which is an essential capability in collaborative, multi-institutional settings.

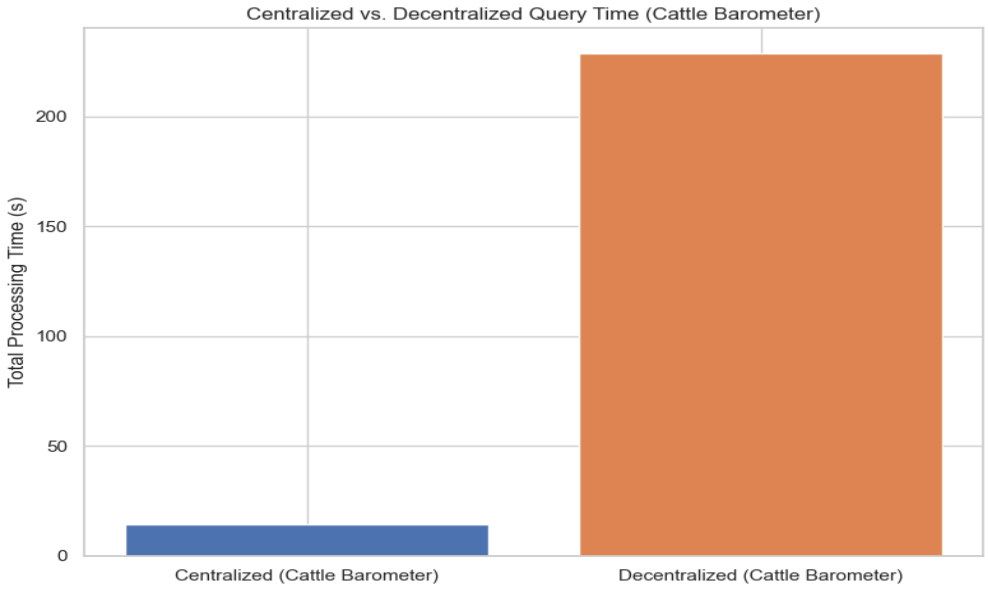

Figure 5. Centralized vs. decentralized query time (Cattle Barometer).

4.2 Experiment 2: Federation scalability (vertical vs. horizontal)

In this experiment, we expanded the number of pods and included data from additional species (cattle, pigs, and poultry) to test the scalability of the decentralized federation approach. The aim was to observe how performance varies as the data volume and pod count grow, reflecting real-world scenarios of distributed veterinary diagnostics.

To further explore the scalability of federated access, two types of federation were tested: vertical (species-specific) and horizontal (mixed species). For example, Pods 1 to 5 represented cattle data from five separate labs, Pods 6 to 11 held pig data from six different labs, and Pod 12 contained poultry data. Although each group was species-specific, the datasets across labs were still heterogeneous, with differences in format, structure, and terminology. These variations were semantically harmonized using the LHO to enable

consistent federated querying across vertically aligned pods. In contrast, the horizontal setup included multiple species per pod, combining cattle, pig, and poultry data, resulting in even higher heterogeneity, both in terms of content and the complexity of the ontology structure. Experiments were designed to assess query performance across these federated configurations using both simple and complex query patterns, as shown in Table 2.

In a vertical federation setup, we ran simple federated SPARQL queries over homogeneous species-specific Solid Pods using a batching strategy, with increasing LIMIT values (5,000, 10,000, 15,000) to test scalability which controlled the maximum number of results returned per query execution. This simulated progressively larger query workloads without changing the query structure. The LIMIT parameter was applied to the result set rather than to individual pods, meaning that the federated engine had to traverse all pods until the combined results reached the specified threshold. The query structure was intentionally lightweight, targeting attributes like Sample, Pathogen, and SampleType, to isolate scalability from query complexity. Despite schema alignment, the number of RDF triples varied across pods, reflecting real-world heterogeneity even within the same species.

In contrast, the horizontal federation setup queried across mixed-species pods (Pods 1–6). Here, each pod contained data from cattle, pig, and poultry, increasing heterogeneity both in content and ontology structure. Query times for simple queries rose from 113.71 to 172.7 seconds for LIMITs of 5,000 and 10,000, respectively, but failed at 15,000 (HTTP 500). Complex queries in this setup also struggled, with a LIMIT of 500 taking over 205 seconds, and 1,000 failing due to timeouts, indicating that horizontal federation is more sensitive to scale and complexity.

In a vertical federation setup, we ran simple federated SPARQL queries over homogeneous species-specific Solid Pods using a batching strategy. Because each batch (e.g., cattle, pig, or poultry) contains data from a single species and shares a consistent schema and ontology structure across its pods. For example, cattle data was queried from Pods 1–5, pig data from Pods 6–11, and poultry data from Pod 12. Each group was queried independently with increasing LIMIT values (5000, 10000, 15000 triples/rows of data) to test scalability. Although the queries targeted homogeneous data (i.e., data from the same species and schema), the number of RDF triples and dataset sizes varied across the pods. This uneven distribution reflects real-world data heterogeneity within species-level labs. The query structure was intentionally lightweight, retrieving only key attributes such as Sample, Pathogen, and SampleType to isolate the impact of scalability rather than query complexity. In the vertical complex case, even when the query fails for cattle at LIMIT 1000 (HTTP 500), results are still successfully returned for pig and poultry pods. This indicates partial execution is possible, failure in one batch doesn't halt others, highlighting resilience of species-wise federation under complex joins. The horizontal federation setup queried across mixed-species pods (Pods 1–6). Simple queries with LIMITs of 5,000 and 10,000 resulted in query times of 113.71 and 172.7 seconds. However, the system completely failed at the 15,000 triple limit with an HTTP 500 error. For complex queries, a LIMIT of 500 took 205.27 seconds to complete, while a LIMIT of 1000 also failed due to timeout constraints.

Table 2. Vertical (species-specific) and horizontal (mixed species) experiments results.

Experiment	Species	Row	Time	Status
Vertical experiment				
Vertical Simple (Limit 5000)	Cattle (1-5)	5000	40.58	Success
	Pig (6-11)	4248	33.52	Success
	Poultry (12)	1889	2.74	Success
Vertical Simple (Limit 10000)	Cattle (1-5)	10000	73.93	Success
	Pig (6-11)	4248	32.14	Success
	Poultry (12)	1889	4.14	Success
Vertical Simple (Limit 15000)	Cattle (1-5)	15000	127.3	Success
	Pig (6-11)	4248	36.25	Success
	Poultry (12)	1889	5.15	Success
Vertical Complex (Limit 500)	Cattle (1-5)	500	230.3	Success
	Pig (6-11)	59	4.36	Success
	Poultry (12)	500	2.25	Success
Vertical Complex (Limit 1000)	Cattle (1-5)	0	-------	HTTP 500
	Pig (6-11)	59	8.04	Success
	Poultry (12)	750	54.33	Success
Horizontal experiment				
Horizontal Simple (Limit 5000)	Mixed (1-6)	5000	113.71	Success
Horizontal Simple (Limit 10000)	Mixed (1-6)	10000	172.7	Success
Horizontal Simple (Limit 15000)	Mixed (1-6)	0	---------	HTTP 500
Horizontal Complex (Limit 500)	Mixed (1-6)	500	205.27	Success
Horizontal Complex (Limit 1000)	Mixed (1-6)	0	---------	HTTP 500

Figure 6 demonstrates that vertical federation offers more efficient and predictable performance as SPARQL LIMIT values increase, compared to horizontal federation. In the vertical setup, cattle queries exhibit a steady linear rise in query time, which correlates with the larger volume of RDF triples present across the cattle-specific pods. In contrast, pig and poultry pods return consistently low query times due to their smaller, fixed triple counts, which quickly saturate the results even at higher limits.

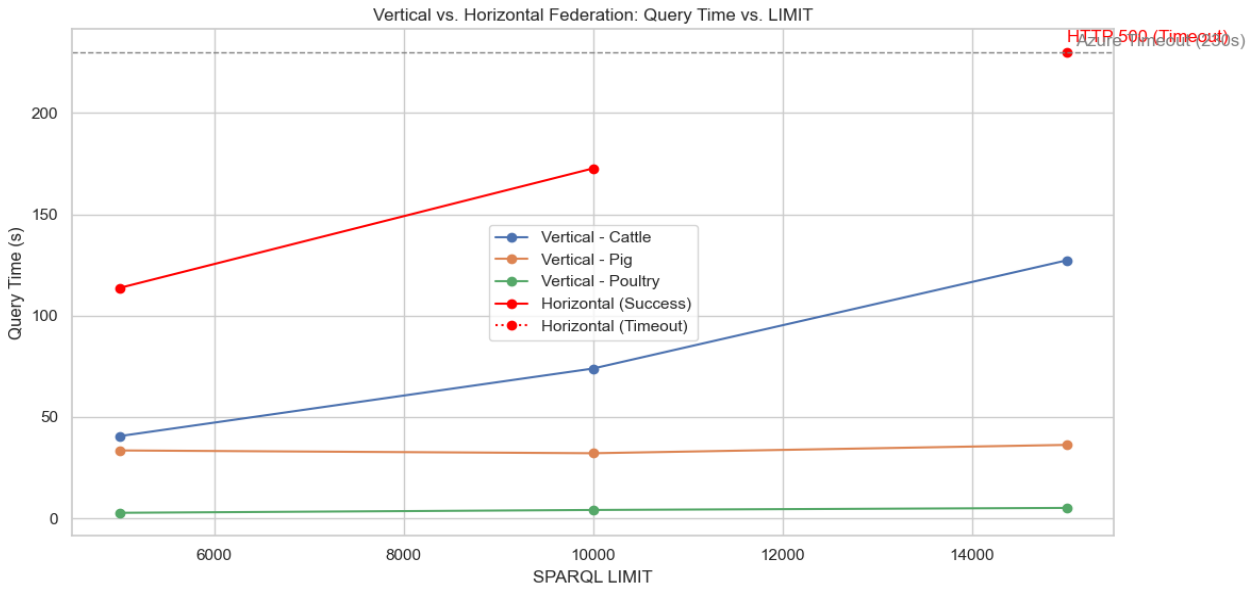

Figure 6. Combined plot: vertical vs. horizontal query time comparison.

On the other hand, horizontal federation, while initially fast at lower LIMITs, fails to scale under higher demand. At LIMIT 15000, the system encounters an Azure timeout (HTTP 500). This is likely due to the increased computational complexity of mixed-species UNION queries across multiple large, uniformly sized pods (~170K triples each). These queries require more triple pattern matching, broader joins, and deeper traversal across heterogeneous species data. Additionally, the Solid server used in this setup was deployed on a lightweight backend with limited memory and computing resources, which was sufficient for testing but not optimized for large-scale federated query execution. As a result, server resource limitations contributed to the timeout observed in the horizontal federation scenario.

4.3 Error analysis and optimisation strategies

Federated querying introduces inherent performance variability, particularly under heterogeneous workloads and distributed environments. One major bottleneck encountered during the experiments was the HTTP 500 error, which typically arose when query complexity exceeded the execution time limits of the Azure Function environment (~230 seconds).

In addition to timeout errors, inefficient query structure and the lack of pagination contributed to latency. Federated SPARQL queries are inherently more complex to optimize due to their runtime dependence on network conditions and remote pod availability.

To mitigate these challenges, three solutions are recommended: increasing the function timeout limit, using SPARQL OFFSET/LIMIT pagination strategies to divide large queries, and simplifying query structure by reducing joins and targeting only essential patterns. Additionally, deploying the federated query engine on larger Azure computational backends (e.g., higher-tier App services or dedicated function plans) could provide more consistent performance for long-running queries. These improvements will help enhance system stability and responsiveness in large-scale, federated deployments (Peng et al., 2016).

5 Discussion

Table 3 shows a comparative summary of three data access approaches. For example, direct data sharing is straightforward to use, offering high accessibility and ease of implementation for small-scale or short-term collaborations. However, it ranks low across most other criteria due to its manual nature. Data must be requested, cleaned, and integrated manually, leading to poor consistency, limited reusability, and virtually no scalability. There is minimal data privacy or control once files are sent. Although FAIRness is a characteristic of the data itself, its preservation depends heavily on the method of sharing. In direct sharing, even data that initially follows FAIR principles can lose these qualities during informal exchanges. Metadata, schema definitions, and version histories are rarely transferred along with the file, especially when files are shared via email or private folders without centralized tracking. Over time, this degrades findability, reusability, and interoperability, key goals of FAIR data management. The issue is not that the data is inherently non-FAIR, but that direct sharing lacks the mechanisms to preserve or promote FAIRness over time. TCO for direct sharing is moderate. It uses simple tools and low-cost infrastructure, but requires a lot of manual effort to clean, merge, and coordinate data each time, making it time-consuming and hard to scale for larger collaborations. This limitation is particularly visible in multi-stakeholder settings like DECIDE, where data exchange depends on coordination between different labs, file versions, and analysts.

Table 3. Comparative summary of three data access approaches.

Evaluation Criteria		**Direct data sharing**	**Centralized federated data sharing**	**Decentralized (Solid Pods) federated access**
Data Access		High	Moderate	High
Data Privacy		Low	Moderate	High
Data Integration		Low	Moderate	High
Data Consistency		Low	High	High
Data Scalability		Low	Moderate	High
Data Findability		Low	Moderate	High
Data Availability		Low	High	Moderate
Data Interoperability		Low	Moderate	High
Data Reusability		Low	Moderate	High
Ease of Use and Implementation		High	Moderate	Moderate
TCO	Personnel	High	Moderate	Moderate
	Infrastructure	Low	Moderate to High	Moderate to High
	Maintenance	High	Moderate	Moderate to High
Performance		Not Applicable	High	Moderate to Low

Centralized Data Sharing offers balanced strengths, performing well across consistency, scalability, and availability. Standardized scripts and automated pipelines help ensure clean, harmonized data, while privacy is

moderately protected through techniques like anonymization of sensitive data. This method prevents reverse engineering and ensures permanent anonymization, but it also means that original identifiers cannot be recovered later, even if re-linking is needed under secure conditions. As a result, the data may become less meaningful or harder to reuse, especially by other users or systems that rely on linking data to original sources. Permanent anonymization can also reduce interoperability if identifiers cannot be aligned or mapped across datasets or domains. Importantly, centralized systems excel in performance, with fast query execution and minimal overhead, as shown in the Cattle Barometer first experiment. TCO for centralized access is moderate to high. Setting it up takes time and technical skills, but once in place, automated pipelines reduce daily effort. Costs mainly come from infrastructure and occasional updates, but overall, the system is efficient, especially when data follows expected formats. While the technical setups can be daunting for non-experts, this complexity is hidden from end-users. Tools like Tableau (https://decide-project-eu.github.io/case-studies-website/case-studies/cattle-barometer.html) or R -Shiny (https://connect.posit.vetinst.no/laksetap/), can be used to present the processed outputs in a user-friendly way, making it easy to explore the data without needing to understand the technical backend.

Decentralized data access using Solid Pods delivers the strongest strengths in privacy, data control, interoperability, availability, and semantic reusability. Data remains under full control of providers, and real-time querying is made possible through a federated SPARQL engine. Integration is achieved dynamically using shared ontologies, and new data sources can be added flexibly. However, performance is a limitation. Federated queries over graph-based data introduce significant latency, especially for complex or multi-pod queries, as seen in 2nd experimental benchmarks. Ease of use and implementation in this model differ by role; technically, it requires familiarity with RDF, SPARQL, and access control policies, making setup more demanding. Yet, like the centralized approach, the final outputs can be consumed by end-users through user friendly interface, offering an intuitive interface that separates the complexity of the backend from the accessibility of the front end. TCO in decentralized systems is high because setting up and managing many separate data servers takes time, money, and technical skills. Even though automation helps reduce daily work, each server still needs its own hosting and regular updates, which adds to long-term costs, especially when more labs are involved. Despite higher costs, decentralized models support long-term scalability and modularity, especially in privacy-sensitive or multi-institutional environments. They reduce the need for data transfer agreements by allowing institutions to retain control of their data. Still, their cost-benefit balance should be considered carefully when evaluating readiness for production-scale deployment. Yet, like the centralized approach, the final outputs can be consumed by end-users through user-friendly interfaces, offering an intuitive experience that separates the complexity of the backend from the accessibility of the front end.

6 Conclusion

Direct data sharing remains useful for short-term or low-volume exchanges, particularly in early project stages where simplicity is key and technical overhead must be minimized. Thus, making it suitable for structured, multi-institutional collaborations like DECIDE. Analytical tools (like SPARQL engines or Tableau) rely on well-defined schemas or ontologies. Inconsistencies introduced through direct sharing make automated tools less effective or require additional pre-processing. From a TCO perspective, this model incurs low infrastructure costs but high personnel costs due to repeated manual effort in cleaning, merging, and maintaining data. Over time, this makes it unsustainable for larger or long-term deployments. Therefore, beyond research settings, other methods for data sharing are a preferred option.

Centralized data sharing is best suited for coordinated workflows requiring standardized formats, automation, and high-performance data pipelines, making it ideal for surveillance dashboards and routine processing. The TCO for centralized access is moderate: initial setup costs are higher due to technical and infrastructure requirements, but automated pipelines reduce recurring personnel effort. This model is effective when data providers are willing to accept centralized control and consistently provide data in agreed-upon formats. Decentralized access offers the greatest privacy, flexibility, and semantic interoperability. It is especially appropriate for independent data providers who prioritize data ownership and selective access. From a TCO perspective, decentralized systems have the highest setup and maintenance costs. While automation can reduce manual effort, infrastructure demands, such as persistent pod hosting and federated query tuning, remain significant. This model works well for distributed networks such as independent labs or stakeholders. However, this approach is only effective when users are not only trained on the use of ontologies but also have a clear understanding of their own data quality, structure, and sharing responsibilities. In practice, this is a major limitation in the field of veterinary surveillance, as highlighted by Meyer et al. (2021) and Delavenne et al. (2025), where basic data documentation and metadata are often lacking, and stakeholders struggle to describe the content or limitations of their own datasets. As such, the decentralized approach currently functions more as a proof of concept, as demonstrated through the Cattle Barometer, rather than a widely implementable solution. The same holds for centralized access models, which require substantial effort to standardize formats, metadata, and ingestion pipelines. At this stage, most tools in DECIDE are not yet equipped for full implementation of either the centralized or decentralized access approaches without further investment in metadata, infrastructure, shared schema alignment, and user training.

The choice between models should reflect both technical requirements and user roles. Analysts and centralized agencies may favor performance and automation, while individual labs or farmers may value sovereignty and selective sharing. However, these users may not have the capacity or willingness to engage with complex, advanced technical tools such as ontologies or federated query systems. Each model represents a trade-off between control, complexity, and scalability.

Future improvements should build on the training and resources already delivered in DECIDE, such as the user workshop on semantic data access ECPLF 2024 workshop agenda, the annotated jupyter notebooks for using ontologies and querying SPARQL, and the step-by-step tutorial for the Cattle Barometer. These efforts should be extended by continuing to ease technical adoption, promote shared standards, and support user-friendly tools that lower the barrier to semantic technologies for non-technical users. Federated learning will be addressed in a separate phase of the project (Milestone M13), building on the federated access infrastructure evaluated here. Future improvements should move beyond technical training and focus on user-friendly tools. For example, the Rosanne Semantic Excel Add-in (Wigham et al., 2015) helps map unstructured data to an ontology directly within Excel, making it easier for lab providers to contribute structured data. On the front end, users could ask questions in natural language and receive both visualizations and plain-language answers, making the system accessible without needing to understand SPARQL or ontology concepts.

All three approaches have been thoroughly documented and assessed within the DECIDE project to inform future guidelines. This work is led by UGent in close collaboration with EpiMundi. Findings from this deliverable will be shared with WP2, WP3, and WP5 to inform upcoming discussions on governance, sustainability, and ethical data access practices. This deliverable (D1.3) completes Milestone M12 (federated data access implemented at least once) by evaluating both centralized and decentralized access approaches. These results will serve as a foundation for Milestone M13, which focuses on implementing and testing federated learning approaches in future project phases.

7 References

Bokma, J., Santman-Berends, I., Vidal, G., Hostens, M., Ribbens, S., Evrard, J., Theuns, S., van Schaik, G., & Pardon, B. (2023). *European veterinary barometer for Bovine Respiratory Diseases: A tool showing diagnostic test results and geolocation of respiratory tract samples from cattle*. 179–181.

Dedecker, R., Slabbinck, W., Wright, J., Hochstenbach, P., Colpaert, P., & Verborgh, R. (2022). *What's in a Pod? A knowledge graph interpretation for the Solid ecosystem*. *3279*, 81–96.

Delavenne, C., van Schaik, G., Frössling, J., Cameron, A., & Faverjon, C. (2025). Reusability challenges of livestock production data to improve animal health. *Scientific Data*, *12*(1), 458.

Meyer, A., Faverjon, C., Hostens, M., Stegeman, A., & Cameron, A. (2021). Systematic review of the status of veterinary epidemiological research in two species regarding the FAIR guiding principles. *BMC Veterinary Research*, *17*(1), 270.

Noor, S., Bokma, J., Pardon, B., van Schaik, G., & Hostens, M. (2024). Agri Semantics: Developments to improve data interoperability to support farm information management and decision support systems in agriculture. In *Smart farms: Improving data-driven decision making in agriculture* (pp. 75–96). Burleigh Dodds Science Publishing Limited. http://dx.doi.org/10.19103/AS.2023.0132.05

Peng, P., Zou, L., Özsu, M. T., Chen, L., & Zhao, D. (2016). Processing SPARQL queries over distributed RDF graphs. *The VLDB Journal*, *25*, 243–268.

Ragab, M., Savateev, Y., Oliver, H., Tiropanis, T., Poulovassilis, A., Chapman, A., & Roussos, G. (2024). ESPRESSO: A Framework to Empower Search on the Decentralized Web. *Data Science and Engineering*, *9*(4), 431–448.

Sambra, A. V., Mansour, E., Hawke, S., Zereba, M., Greco, N., Ghanem, A., Zagidulin, D., Aboulnaga, A., & Berners-Lee, T. (2016). Solid: A platform for decentralized social applications based on linked data. *MIT CSAIL & Qatar Computing Research Institute, Tech. Rep.*, *2016*.

Van Schaik, G., Hostens, M., Faverjon, C., Jensen, D. B., Kristensen, A. R., Ezanno, P., Frössling, J., Dórea, F., Jensen, B.-B., & Carmo, L. P. (2023). The DECIDE project: From surveillance data to decision-support for farmers and veterinarians. *Open Research Europe*, *3*, 82.

Wigham, M., Rijgersberg, H., Timmer, M., & Top, J. L. (2015). *Rosanne: Islands of structure in unstructured data: Final report 2015: Rosanne valorisation project* (Issue 1590). Wageningen UR-Food & Biobased Research.