# Towards Serverless Data Exchange Within Federations

Boris Sedlak[1(✉)], Victor Casamayor Pujol[1], Praveen Kumar Donta[1], Sebastian Werner[2], Karl Wolf[2], Matteo Falconi[3], Frank Pallas[2], Schahram Dustdar[1], Stefan Tai[2], and Pierluigi Plebani[3]

[1] Distributed Systems Group, TU Wien, 1040 Vienna, Austria
{b.sedlak,v.casamayor,pdonta,dustdar}@dsg.tuwien.ac.at
[2] Information Systems Engineering, Technische Universität Berlin, Berlin, Germany
{sw,kw,fp,st}@ise.tu-berlin.de
[3] Politecnico di Milano, Milan, Italy
{matteo.falconi,pierluigi.plebani}@polimi.it

**Abstract.** In this paper, we propose a novel approach for sharing privacy-sensitive data across federations of independent organizations, taking particular regard to flexibility and efficiency. Our approach benefits from data meshes and serverless computing – such as flexible ad-hoc composability or minimal operational overheads – to streamline data sharing phases, and to effectively and flexibly address the specific requirements of highly variable data sharing constellations.

Based on a realistic scenario of data sharing for medical studies in a federation of hospitals, we propose a five-phase data product lifecycle and identify the challenges that each phase poses. On this basis, we delineate how our approach of *serverless data exchange* addresses the identified challenges. In particular, we argue that serverless data exchange facilitates low-friction data sharing processes through easily usable, customizable, and composable functions. In addition, the serverless paradigm provides high scalability while avoiding baseline costs in non-usage times. Altogether, we thus argue that the *serverless data exchange* paradigm perfectly fits federated data sharing platforms.

**Keywords:** Data Exchange · Data Mesh · Serverless Computation

## 1 Introduction

Data are one of the most valuable assets in many organizations, equally driving business processes and machine learning algorithms. To further exploit its potential value, data can be combined and extended with assets that are shared

Funded by the European Union (TEADAL, 101070186). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.


M. Aiello et al. (Eds.): SummerSOC 2023, CCIS 1847, pp. 144–153, 2023.
https://doi.org/10.1007/978-3-031-45728-9_9

between organizations, rather than sinking into the oblivion of data silos. This is motivated by data as a commodity, generating revenue, or from a research perspective, sharing data within a federation on a give-and-take basis. However, data often contains confidential business insights or personal information; thus the effort to share the data in a secure, trusted, performant, and efficient way – avoiding, for instance, accidental data leaks – becomes crucial. At the same time, the way in which data is needed depends on the *data consumer*. This usually leads to the creation of several copies of the initial dataset, each tailored to a specific consumer. All of these aspects contribute to additional *friction in data management* that in many cases hampers, if not blocks, data sharing [2,7]. We argue that the solution must be aware of these frictions and address most of them through careful distribution of responsibilities among actors [4] and applications. Moreover, we argue that a *federated data exchange platform* can leverage federated resources not only to reduce friction but also to improve performance, energy consumption, and transparency.

This paper introduces a novel data exchange architecture that combines serverless computing advantages with principles of the data mesh [6]. Additionally, we propose a novel data sharing lifecycle and address the critical responsibilities and challenges within these novel federated data exchange platforms. Lastly, we pinpoint the prospect of leveraging serverless data exchange to minimize friction and unlock optimization potential within the federated context.

In the remainder of this paper, we present a motivating scenario from the medical sector in Sect. 2, a data sharing lifecycle in Sect. 3, including responsibilities and challenges of data exchange in federations, and in Sect. 4 introduce our proposed serverless data exchange architecture. Finally, in Sect. 5, we summarize this paper and identify potential future work.

## 2 Motivating Scenario

The analysis of large and diverse patient datasets is essential for the successful implementation of medical trials; however, this only becomes possible through the aggregation of various sources. In this context, the current challenge is to simplify the data exchange among hospitals, which requires a lot of effort in selecting and preparing the data in compliance with internal regulations and general norms (e.g., GDPR), common data formats (e.g., OMOP), as well as agreements on semantics (e.g., SNOMED).

Based on what is already happening in this community, federations recognizing the importance of data sharing to the advancement of medical studies are under establishment (e.g., Elixir[1]). This association, led by domain experts, would share and enforce an agreement on data discovery, metadata standards, and access functions, and partially automate the enactment of data access policies. From a researcher's perspective, this simplifies the search for data that meet the medical study's requirements and allows quick assembly of large pools of diverse patient data. Nevertheless, additional aspects contribute to the friction

[1] https://elixir-europe.org.

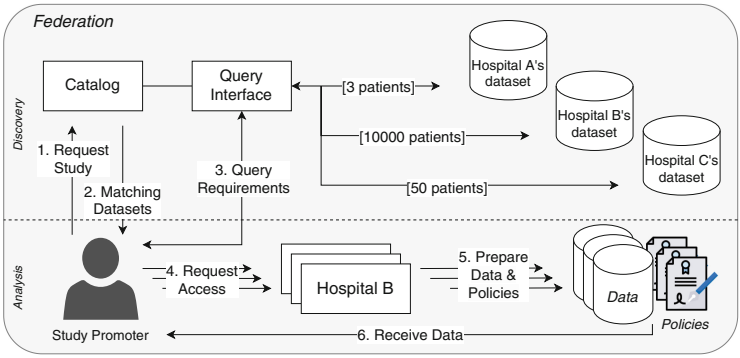


**Fig. 1.** Study Promoter Workflow

when sharing data and the scenario reported in Fig. 1 helps to describe them. In particular, we visualize the two phases required to enable a joint study in this type of federation: *discovery*, i.e., the search for relevant patient datasets, and the *analysis* of actual data.

To find relevant data, (1) the federation's data catalog (as established in data mesh [6]) provides the ability to search through datasets by filtering based on metadata (e.g., data types, usage consent). The study promoter can use the catalog (2) to find datasets that match their requirements, e.g., patients with a certain pathology or within an age range. The catalog can further (3) provide the number of accessible patients in desired datasets, for example, by querying how many patients consented to the study's purpose. This provides the promoter with a means of selecting the most favorable data providers to enter into an agreement with. After deciding which dataset to use, the analysis phase begins.

Before the data can be accessed directly, (4–5) an agreement must be reached between the study promoter and the data provider(s) to determine the rules for using and accessing the data. As soon as the promoter reaches an agreement with the organizations that offer the fitting datasets, the promoter can request the actual data. First, the provider has to establish *access policies* to ensure that only data that is relevant and contained within the agreement is exposed. Then, *transformations* imposed by the agreement must be performed to ensure interoperability between formats (e.g., unstructured historical patient data, MRI imagery), fulfillment of legal obligations, and compliance with federation guidelines. Both the location of transformations and computations (e.g., analysis steps) must be chosen to ensure compliance with regard to privacy, performance, and sustainable use of the federation's resources. Aggregated and transformed data from different sources in the federation can, in turn, themselves become data sources for other members of the federation if properly accompanied by a set of metadata and policies, and if permitted by the agreement with the original data providers. This allows federation members to reuse and enhance data products without wasting resources by re-performing expensive transformations.

## 3 Federated Data Product Lifecycle

In data mesh [6], a data product is defined as the smallest unit of architecture. Depending on the data product's domain, a specific team is in charge of managing its lifecycle. Revising this definition in a federated setting, we propose a *federated data product* as the shareable and comparable unit. It is built, according to the service orientation principles, by the data provider and, through the associated API that mediates the access, the data consumer (i.e., the study promoter in Fig. 1) can obtain the data and can combine it with other accessed data products. On this basis, it is also fundamental to define the lifecycle of the federated data products, to offer a systematic and holistic approach to address organizational and technical hurdles (i.e., friction) in exchanging data across organizations by identifying responsibilities, objectives, and design requirements in each phase of a federated data product, akin to the data mesh lifecycle [3].

The lifecycle that we envision is divided into five phases. In the following, we describe each phase and extract *responsibilities* and *challenges* (see Table 1) that a federated sharing platform must address to enable data sharing. Here, we assume the data provider has already joined the federation, including the necessary processes for interacting with other users.

**Data Onboarding:** Within the first phase of a federated data product lifecycle, data collected by the data provider is *prepared for storage and sharing*. This includes the *data classification*, the setup of necessary *ingestion* – either a one-off transfer or a streaming setup – including necessary transformations, and the *assignment of storage policies*. Once the domain experts have assembled the data, they need to *specify the data product's metadata* in accordance with the federation's metadata model. To ensure that operations on the data comply with internal rules, respective *policies are attached* (e.g., security, confidentiality or access policies or policies that require more complex data transformations to be performed) and provided alongside the metadata. The federated data product is considered onboarded once it is properly described, typically using a domain-specific language [6], and an initial version of it is placed in storage in line with its attached storage policies, e.g., within the EU, using a minimum redundancy, or a given level of encryption.

**Publishing:** Once the federated data product is onboarded, it can be made available to the federation by *publishing it to a shared data catalog*. This catalog of federated data products must allow consumers to discover data that match their requirements through its metadata. To avoid inconsistencies, the catalog must reflect the latest status of federated data products, e.g., their availability and assigned policies. Additionally, the metadata (e.g., the number of available records) may vary in-between potential consumers based on their identity and access context; these constraints must be reflected through consumer-aware policies. This entails that the metadata provided during the onboarding phase might be enriched further. As part of this phase, domain experts can specify the

*capabilities necessary to consume the federated data product*, e.g., the required resources or required product policies enforcement tools.

**Table 1.** Summary of the challenges for a federated data exchange platform

| ID | Challenge | Description | Lifecycle phase |
|---|---|---|---|
| C1 | *Shared metadata model* | A domain-specific metadata model to aid the discovery and matching of federated data products | Onboarding |
| C2 | *Policy language* | Usage of a sophisticated policy language to enable platform-supported lawful and trustworthy data exchange | |
| C3 | *Data control plane* | A control plane enabling domain experts to specify and update policies as data changes | |
| C4 | *Stretched data lake* | A policy-based data placement approach utilizing storage and streaming across federated resources | |
| C5 | *Federated data catalog* | Ensure that all members of the federation can discover all federated data products | Publishing |
| C6 | *Consistent metadata* | Keeping browsable metadata (e.g., policies, number of records) in sync with the federated data product | |
| C7 | *Matchmaking* | Support the aligning and matching of consumer requirements to product metadata | |
| C8 | *Context-aware discovery* | Support interactive negotiation queries for consumer-specific metadata, based on product policies, consumer's access purpose, and context | |
| C9 | *Consumer transformations* | Support required consumer transformations, e.g., ensuring format capabilities, storage policy needs | |
| C10 | *Shared agreements* | Ensure that agreements are available in a standardized and immutable format | Sharing |
| C11 | *Enforceable agreements* | Support codifying agreement policies in an unequivocal, automatically enforceable way | |
| C12 | *Trust mechanisms* | Ensure or prove bilateral compliance with accepted agreements (e.g., monetary incentives [8] or trustworthy transformations [5]) | |
| C13 | *Data lineage* | Enforcing and capturing agreed-upon consumption contexts, purposes, and transformations | Consumption |
| C14 | *On-demand transformations* | Support smart and on-demand transformations to comply with policies, i.e., allocation of computations within the federation | |
| C15 | *Federated access control* | Support fitting access control mechanisms, compatible with policies and execution environments | |
| C16 | *Enforceable deletions* | Support the deletion of all copies of a federated data product, possibly including derivatives | Discontinue |
| C17 | *Observable lifecycle actions* | Support the audit of all data consumption actions to find and discontinue a federated data product | |
| C18 | *Maintain knowledge* | Preserve functions and system optimization for future improvement | |

**Sharing:** Once the federated data product is published, interested members of the federation can request the data. This is the first occasion where data providers and consumers need to interact. Consumption of federated data products can be bound to various terms and conditions and implies that *both parties come to an agreement* on how the data can be consumed. Agreements *restrict the consumption in various dimensions*, e.g., by posing an end date, stating the purpose of the data consumption, or including transformations that the data must

undergo. These transformations can range from projection and selection mechanisms to advanced analysis and are *defined by domain experts*. Thus, agreements are a central part of the sharing processes and govern the rights, responsibilities, and obligations (e.g., technical or legal) of both parties. The agreement is typically formalized in a contract that both *parties sign*, a process that the federated sharing platform must support. From this agreement, the platform can derive the policies that must be enforced, e.g., setting up an access control and/or transformation mechanism or a shared identity provider.

**Consumption:** Ultimately, the dataset is consumed according to the conditions that were formalized; *compulsory operations (obligations)* included in the agreement must be performed by the federated sharing platform. To support audit mechanisms, *all interactions with the dataset must be documented*, which also improves data lineage, i.e., provide information on how the original data had been altered. At the same time, access logs must comply with privacy guidelines themselves. Moreover, the federated sharing platform can ease the consumption of federated data products, e.g., by *providing means to filter the data*, move it to a different location or *perform a purpose-based transformation* to ensure compliant consumption [9]. We assume that the *data consumer can initiate the consumption* as needed after a sharing agreement is reached. The consumption can be continuous, intermittent, or a one-time event. Thus, the federated sharing platform must support on-demand, continuous, and bulk transformations.

**Discontinue:** Once the federated data product is no longer needed, or the data provider decides to no longer provide it, it can be discontinued. This phase is the last in the lifecycle and is the counterpart to the onboarding phase. Here, *all active sharing agreements are terminated* and the data product is removed from the catalog. This process may require *prior notification to the data consumers*, e.g., to allow them to adjust their applications or to ensure that they can remove all copies of the federated data product. Here, the *federated control plane* should provide the functionality to ensure that the federated data product is removed from all controlled locations where it was stored, e.g., by allowing an audit of data consumption logs or by providing a means to remove all copies of the federated data product across the shared environment. However, additional nontechnical means such as legal agreements should be in place to ensure compliance.

## 4 Serverless Data Exchange

This section presents a novel architecture to exchange data products in a federation between organizations while addressing the challenges in Table 1. We leverage properties of serverless computing to enable trustworthy data sharing with minimal operational overhead [10], establishing the concept of serverless data as the capacity to manage the data lifecycle. Figure 2 depicts the serverless architecture we propose as a backbone for the federated sharing platform, following the presented lifecycle phases of a data product. This outlines an initial

proposal; however, we are aware that implementing this architecture presents further technical challenges not covered here.

**Data Onboarding:** Whenever a data provider (e.g., a hospital) offers data products to other members of the federation, the architecture for serverless data exchange provides capabilities to integrate data as a single logical product, regardless of physical (distributed) storage. Policies, supplied by domain experts (**C3**), can be attached to the federated data product before it is exposed through the catalog. Consider that most of the policies at the onboarding phase concern the storage, e.g., where within the federation the product may be stored. Given that policies also depend on the data consumer, the architecture provides

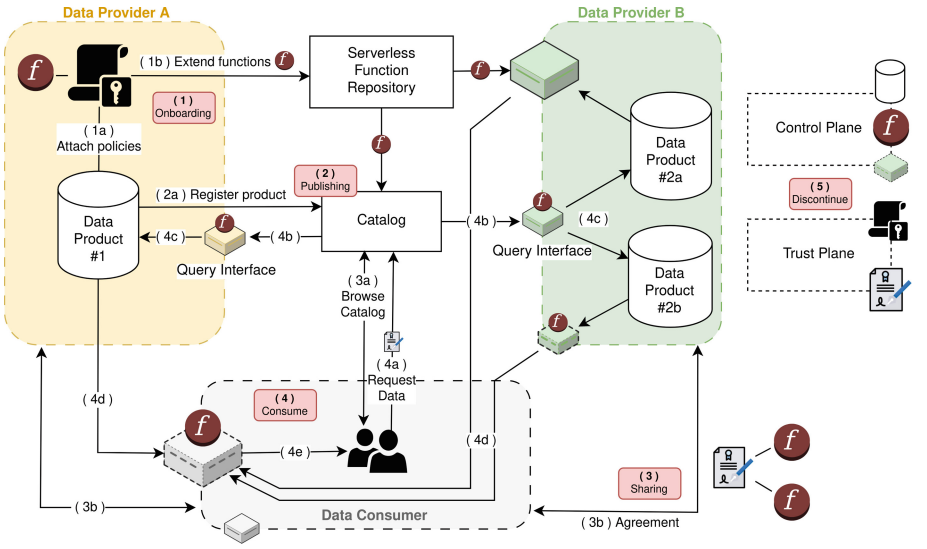


**Fig. 2.** Serverless data exchange within a federation. *Data Product #1* is onboarded using storage provisioned by the Control plane. Domain Experts supply metadata and policies (1a) which are attached to the federated data product; this might include serverless functions new to the federation that extend the existing function repository (1b). *Data Product #1* is registered by its provider through the federation-wide data catalog (2a), including mandatory policies and functions. The consumer, who is also part of the federation, uses the catalog to browse registered data products (3a) that can be matched to his/her requirements. The consumer then establishes an agreement with both providers (3b) on how data are delivered, i.e., formats, how and where data are transformed according to policies, retention period at the consumer, etc. The consumer requests the data through the catalog by providing the agreement (4a), which is received by the providers' Query Interface (QI) (4b). The QI then instructs individual data products to provide the data to the consumer (4c). Processing of data according to serverless functions can occur at various locations, e.g., at premises provided by the providers or the consumer, or at ad-hoc nodes on any site provided by the Control plane (4d). Finally, the consumer is served the data (4e) as specified in the agreement.

capacities to attach further policies at the following stages. In general, policies might contain references to functions (**C2**) that already exist in the federation; nevertheless, ad-hoc functions can be supplied by federation members.

**Publishing:** The federated data product is published by its provider using a federation-wide catalog (**C5**), which logically acts as a unified entity for the entire federation. However, the catalog's implementation can be distributed in the federation (e.g., a distributed database (**C6**)); custom instances can be ready on-demand thanks to the serverless functions available at the federation. The catalog includes references to federated data products and their metadata (**C1**), including all policies attached up to that moment. The purpose of the catalog is to make federated data products discoverable in the federation, allowing each member to search the catalog for products that meet their requirements (**C7-8**).

**Sharing:** When federated data products match the requirements of a consumer (e.g., study promoter), during the sharing phase, all concerned parties need to agree on how data will be provided. Policies specific to the agreement can be supplied, which can either originate from the federation's function repository (**C11**), be incorporated by agreement members, or be provided by third-party entities. All in all, the architecture provides an *extendable and composable framework* that may include any type of function within the agreement. Agreement members (or rather their domain experts) can customize functions by supplying individual implementations or creating multiple versions of functions. These functions can be composed to generate serverless data processing pipelines [10] (**C9**) that transparently manage the transfer of federated data products and conversion from the provider to the consumer. Agreements themselves are stored by all concerned parties (**C10**) and serve as proof of trust between them (**C12**); in this regard, the architecture envisions a Trust plane that ensures proper agreement compliance.

**Consumption:** The separation of invocation from execution given by the serverless paradigm enables the data provider and consumer to have an execution tailored to their needs. The execution of functions can be optimized by the architecture's Control plane using the computing continuum [1] of the federation (**C4**), e.g., to minimize resource usage or energy consumption. The proposed approach for serverless data facilitates on-demand access to federated data products; hence, functions are created, provisioned, and executed by triggering consumption events (**C14**). Afterward, they are evicted and the federation infrastructure is freed thanks to the scale-to-0 capability of serverless computing. The Trust plane has observability over data transformations, providing data lineage awareness (**C13**) and access control mechanisms (**C15**). Federation members can access published data products according to agreements, i.e., consumers provide a copy of an agreement and identity, which determines how data are prepared and served.

**Discontinue:** When the agreement finishes (e.g., maximum number of access or time exceeded) or one of the parties withdraws from the agreement, data are no more available. As a consequence, all resources utilized for consumption are released by the Control plane (**C16**); this includes all processing facilities to run serverless functions and consolidated storage for optimizing data consumption. The Control plane will leverage data lineage capabilities from the Trust plane to find all elements that must be discontinued (**C17**). Interestingly, all of the serverless functions in use, as well as Control plane optimization decisions, are kept within the federation for future use and continuous improvement (**C18**).

By seamlessly integrating serverless capabilities with federated data products, we aim to alleviate the provisioning burden of the data provider and eliminate obstacles that impede the exchange of data products, such as the discussed challenges. This relies heavily on components such as the Control plane and Trust plane, which provide resources and establish trust between the parties.

## 5 Conclusion and Future Work

We proposed a federated data platform that combines serverless computing and data mesh to reduce friction in data exchange across organizational boundaries, such as sharing medical research data. We delineated a five-step data product lifecycle, identified the associated technical challenges, and sketched an overall architecture to address them. Our argument is that through *serverless data exchange*, domain experts can handle complex data behavior, even for ad-hoc and non-continuous data sharing scenarios. Serverless principles support this by providing scalability, flexible placement, and composability of functions.

Having established the conceptual foundations, future work comprises the prototypical implementation and use case-driven evaluation of the platform and its components. This includes aspects such as the allocation of serverless functionality along the compute continuum, consideration of trust-related issues, and questions of overall platform management and control across the federation.

Despite the conceptual nature of considerations presented, we see strong points for *serverless data exchange* to gain significant momentum. Our five-phase data product lifecycle, our identified technical challenges, and our serverless data exchange architecture shall guide and drive respective future activities.

## References

1. Dustdar, S., Pujol, V.C., Donta, P.K.: On distributed computing continuum systems. IEEE Trans. Knowl. Data Eng. **35**(4), 4092–4105 (2023). https://doi.org/10.1109/TKDE.2022.3142856
2. Edwards, P., et al.: Science friction: data, metadata, and collaboration. Soc. Stud. Sci. **41**, 667–90 (2011). https://doi.org/10.2307/41301955
3. Eichler, R., et al.: From data asset to data product - the role of the data provider in the enterprise data marketplace. In: Service-Oriented Computing, pp. 119–138 (2022). https://doi.org/10.1007/978-3-031-18304-1_7

4. Eschenfelder, K.R., Shankar, K.: Of seamlessness and frictions: transborder data flows of European and US social science data. In: Sundqvist, A., Berget, G., Nolin, J., Skjerdingstad, K.I. (eds.) iConference 2020. LNCS, vol. 12051, pp. 695–702. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43687-2_59
5. Heiss, J., Busse, A., Tai, S.: Trustworthy pre-processing of sensor data in data on-chaining workflows for blockchain-based IoT applications. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. (eds.) ICSOC 2021. LNCS, vol. 13121, pp. 133–149. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_9
6. Machado, I.A., Costa, C., Santos, M.Y.: Data mesh: concepts and principles of a paradigm shift in data architectures. Procedia Comput. Sci. **196**, 263–271 (2022)
7. Sedlak, B., Casamayor Pujol, V., Donta, P.K., Dustdar, S.: Controlling data gravity and data friction: from metrics to multidimensional elasticity strategies. In: IEEE SSE 2023. IEEE, Chicago (2023). (Accepted)
8. Sober, M., et al.: A blockchain-based IoT data marketplace. Cluster Comput., 1–23 (2022). https://doi.org/10.1007/s10586-022-03745-6
9. Ulbricht, M.R., Pallas, F.: CoMaFeDS: consent management for federated data sources. In: 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), pp. 106–111 (2016). https://doi.org/10.1109/IC2EW.2016.30
10. Werner, S., Tai, S.: Application-platform co-design for serverless data processing. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. (eds.) ICSOC 2021. LNCS, vol. 13121, pp. 627–640. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-91431-8_39