DataFabric-DermAlert: A Tool for Federated Health Data Integration and Fairness-Aware Dataset Construction

Abdul Hannan
Universidade de Brasília
Brasília, Brazil
hannanhoney5000@gmail.com

Heitor Marques S. Barbosa
Universidade de Brasília
Brasília, Brazil
heitormsb@gmail.com

Carla Silva Rocha Aguiar
Universidade de Brasília
Brasília, Brazil
caguiar@unb.br

Isaque Alves
Universidade de São Paulo
São Paulo, Brazil
isaquealvesdl@gmail.com

Abstract

Clinical and dermatology-related data are frequently distributed across heterogeneous systems, described using local vocabularies, and subject to governance restrictions that hinder full centralization. These challenges slow down dataset construction and complicate the reproducibility of subgroup-aware evaluations. This paper introduces DataFabric-DermAlert, an open source containerized tool designed to support governed health data integration through federated query definition, semantic equivalence management, Bronze and Silver dataset layers, and controlled sharing of curated datasets. The tool offers a Next.js interface for registering data sources, defining cross-source relationships, curating data dictionaries and equivalence groups, materializing versioned analytical datasets, and exporting selected datasets via Delta Sharing. DataFabric-DermAlert addresses key data engineering challenges, including the construction of auditable, multi-source, versioned datasets and the inspection of subgroup-related evidence under a reproducible protocol. The paper details the architecture, tool workflow, implementation stack, and three engineering evaluations covering controlled federated execution, semantic retrieval and normalization, and reproducible construction, validation, versioning, and sharing of dermatology metadata products derived from HAM10000, HIBA Skin Lesions, and PAD-UFES-20.

Demo video: https://figshare.com/s/d7a7e4d111f96353c3ef.

Keywords

Data Fabric, Health Data Integration, Data Virtualization, Semantic Interoperability, Dermatology AI, Dataset Construction, Software Tool

1 Introduction

Artificial intelligence applications in dermatology require reliable and well-described data. In practice, relevant information is distributed across databases, imaging repositories, questionnaires, pathology reports, and local metadata files. The same patient attribute or clinical concept may appear under different column names, local codes, value formats, or institutional conventions. As a result, researchers often need manual scripts and ad hoc spreadsheets before they can define a cohort or evaluate a model.

This issue is particularly significant in dermatology, as public skin image datasets have been criticized for limited transparency and inadequate representation of skin types and demographic information [1, 20]. Studies on diverse dermatology image collections have demonstrated performance disparities across skin tones and disease groups [5]. These observations underscore the need for enhanced data engineering infrastructure. Nevertheless, integrating datasets alone does not ensure the development of fair clinical models. While integration facilitates the inspection of diversity, missingness, provenance, and subgroup metrics, fairness ultimately relies on clinical validation, statistical analysis, and the specific deployment context.

DataFabric-DermAlert was developed to address this specific, well-defined requirement. It is a software tool designed to integrate distributed health data and generate auditable analytical datasets. The tool integrates federated query definition, semantic mapping, versioned Bronze and Silver layers, and controlled dataset sharing. It is intended for data engineers, clinical researchers, and AI researchers who require the ability to register heterogeneous sources, align local variables, execute federated queries, freeze dataset versions, and document the evidence supporting downstream experiments.

The primary contributions of this work are: (i) a web-based workflow for defining federated relationships across autonomous health data sources; (ii) a semantic workbench supporting data dictionaries, equivalence groups, value mappings, conflict resolution, and auditable normalization; (iii) Bronze and Silver dataset layers that differentiate raw or virtualized access from curated analytical datasets; and (iv) a sharing layer enabling controlled export of versioned datasets. The evaluation presented constitutes an engineering validation of the tool, rather than a clinical validation of a skin cancer detection system.

2 Architecture

DataFabric-DermAlert is organized as a data fabric tool rather than a monolithic clinical database. Figure 1 shows the main architectural elements. Distributed source systems remain under their original administrative control, while the tool stores the metadata required to expose selected tables, define relationships, apply semantic rules, and build reproducible datasets. This separation is important in health contexts because integration must preserve provenance, access boundaries, and the ability to audit how a dataset was produced.

The architecture has five responsibilities. First, the interface and API layer expose tool operations through a Next.js frontend and backend services for source registration, semantic catalog operations, dataset creation, and sharing configuration. Second, the federation layer uses source metadata and connector definitions to query distributed relational sources without requiring full centralization. Third, the semantic layer stores dictionary terms, source-column mappings, value mappings, confidence metadata, approval state, and conflict information. Fourth, the materialization layer separates Bronze datasets, which preserve source-level structure or virtualized access, from Silver datasets, which apply curated transformations and equivalence groups. Fifth, the sharing and audit layer records manifests, versions, and authorized recipients so that exported datasets can be traced back to the source versions and semantic catalog state used to produce them.

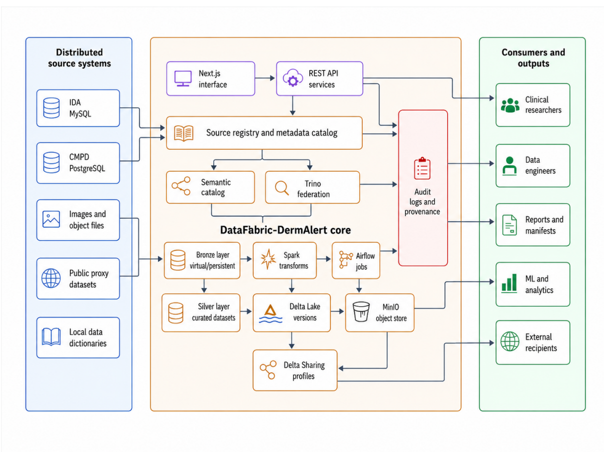

Figure 1: Architecture of DataFabric-DermAlert. Source systems are registered through connectors and metadata; Trino supports federated access; the semantic catalog governs equivalence mappings; Bronze and Silver layers separate source-level access from curated analytical datasets; Delta Lake, MinIO, and Delta Sharing support versioned storage and controlled export.

The architecture does not ensure clinical fairness on its own. It provides the infrastructure needed to construct auditable multi-source datasets, document semantic choices, and inspect subgroup-related evidence before downstream AI experiments are interpreted.

3 Tool Workflow

DataFabric-DermAlert follows a sequence that reflects the way the tool is used in the demonstration: federated query definition, semantic alignment, Bronze dataset management, Silver dataset construction, and Delta Sharing. This order keeps the workflow close to the implementation, rather than presenting the platform only as an abstract architecture.

3.1 Federated Query Definition

The workflow begins by registering source schemas and defining cross-source relationships. The demonstration uses two deterministic synthetic sources. IDA (MySQL) exposes the `ficha_dermato` table in the `ida` schema, containing 1,507 rows and ten columns, five of which are exposed. CMPD (PostgreSQL) exposes the `perfil_saude` table in the `public` schema, containing 1,507 rows and sixteen columns, seven of which are exposed. The synthetic keys `cpf_titular` and `numero_cpf` are unique and non-null in their respective sources, producing a one-to-one relationship. An inner join on these keys produced 1,507 Bronze rows. Figure 2 shows the registered relationship.

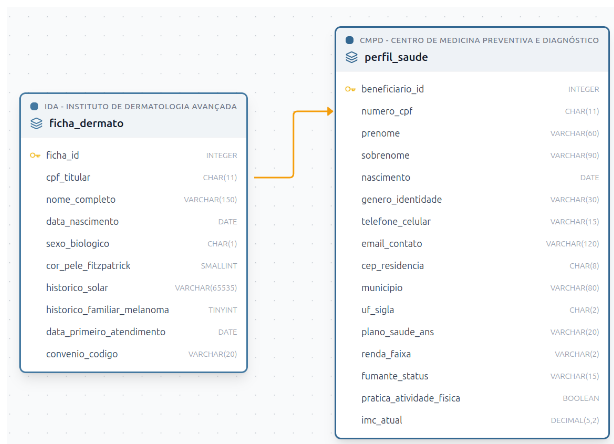

Figure 2: Federated query definition linking the two synthetic source schemas.

Trino is used as the interactive SQL engine for federated access. Its connector-based distributed SQL model, inherited from Presto, supports queries across heterogeneous data sources [16, 17]. When supported by the source connector, predicates are pushed down to reduce the amount of data transferred. This design does not replace institutional governance approval, but it avoids requiring complete data centralization for multi-source analysis.

3.2 Semantic Equivalence and Data Dictionary

The semantic layer stores dictionary terms, source columns, value mappings, equivalence groups, confidence scores, provenance, and mapping state. In the demonstration, `sexo_biologico_padronizado` associates `sexo_biologico` from `ficha_dermato` with `genero_identidade` from `perfil_saude`. Each source has one column mapping and two value mappings: `feminino`→`F` and `masculino`→`M`. Missingness was 88 (5.84%) and 65 (4.31%), respectively. Competing mappings are reviewed before Silver construction, and only the confirmed representation is applied. The four mappings in this demonstration were unambiguous, resulting in zero unresolved cases. Figure 3 shows the corresponding column-group view, while the complete mapping workflow is demonstrated in the accompanying video.

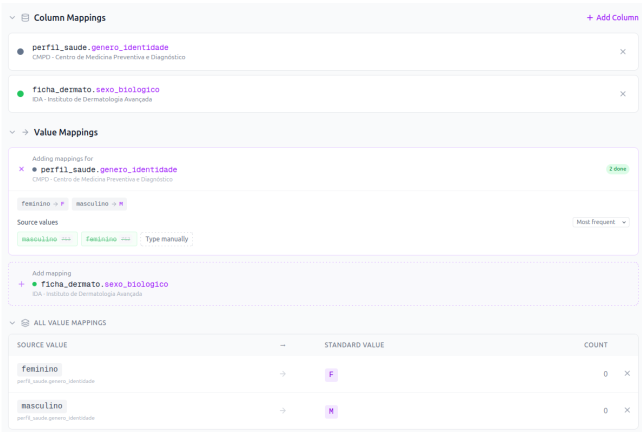

Figure 3: Semantic equivalence workbench. Local fields and values are mapped to standard concepts and values before being used in curated datasets.

3.3 Bronze Layer

The Bronze layer represents source-level datasets. A Bronze dataset may be persistent when the federated result is materialized in storage, or virtualized when the query remains on demand. Figure 4 shows a persistent Bronze dataset built from the two registered sources. The screen exposes source tables, cross-connection relationships, version history, execution status, and the number of rows produced by each materialization. In the verified demonstration, materialization of the federated one-to-one join produced 1,507 Bronze rows.

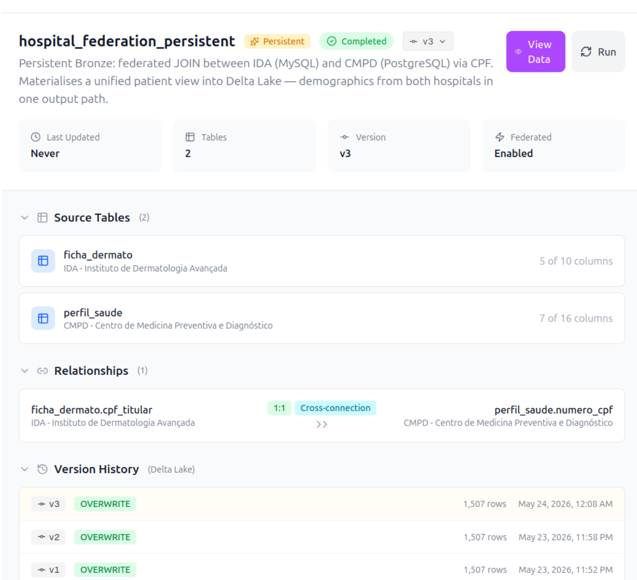

Figure 4: Bronze layer dataset view. The layer keeps source-level structure, federation status, relationships, and Delta version history.

This layer is intentionally close to the original sources. Its purpose is to make source access reproducible without prematurely applying all semantic and analytical transformations. This separation helps distinguish the question "what did the source expose?" from the later question "what analytical dataset was built from it?"

3.4 Silver Layer

The Silver layer constructs curated analytical datasets from Bronze inputs. It records the selected Bronze version, transformation rules, equivalence groups, filters, and export configuration. The interface supports common transformations such as template-based date handling, accent removal, uppercasing, and value normalization. It also supports the selection of semantic column groups, such as patient CPF and biological sex. Figure 5 shows the Silver dataset view with transformations and equivalence groups.

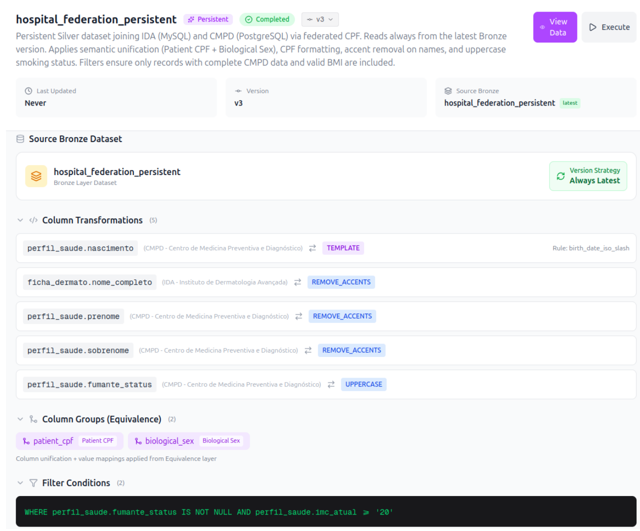

Figure 5: Silver layer dataset view. Curated datasets combine selected Bronze data, transformations, equivalence groups, and reproducible version selection.

The Silver build normalized `feminino` and `masculino` to `F` and `M`, removed accents from `cor_pele_fitzpatrick`, and uppercased `fumante_status`. It produced 1,507 rows while retaining the synthetic identifiers and provenance fields. The important design choice is to treat a Silver dataset as a reproducible artifact. The dataset builds records source versions, catalog versions, filters, selected fields, join keys, and transformation rules. This is necessary because small changes in mappings or filters can change subgroup counts and downstream metrics.

3.5 Delta Sharing

The final workflow step is controlled dataset sharing. DataFabric-DermAlert uses Delta Sharing to register curated datasets and authorize recipients. Figure 6 shows the sharing view with a clinical analytics share and authorized recipients. In a real deployment, this layer would need to be connected to institutional access policies, credential management, audit logging, and data-use agreements.

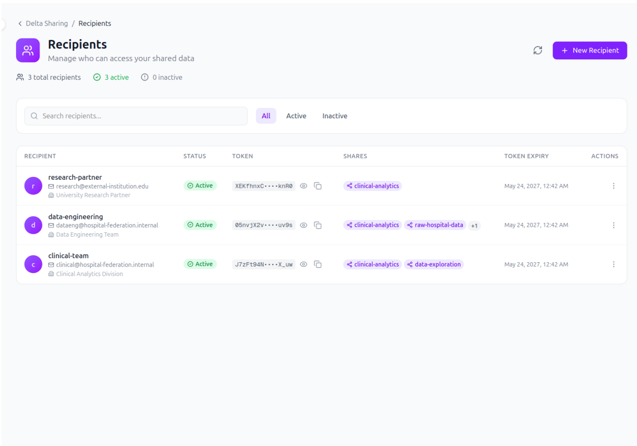

Figure 6: Delta Sharing screen. Curated datasets can be grouped into shares and exposed only to authorized recipients.

4 Implementation

The current implementation combines a web interface, API services, distributed query execution, batch processing, versioned storage, and object storage. Table 1 summarizes the main components. The web interface was implemented with Next.js. The backend exposes endpoints for source registration, semantic catalog operations, dataset creation, and sharing configuration. Trino handles federated SQL access; Spark handles batch transformations and materialization; Delta Lake stores versioned tables; MinIO provides S3-compatible object storage; and Airflow orchestrates repeatable jobs. The semantic catalog stores dictionary terms, mappings, equivalence groups, confidence metadata, and audit information.

Table 1: Main Components of DataFabric-DermAlert

Component	Role	Tool Functionality
Next.js	Web interface	Provides screens for federation, semantic mapping, Bronze/Silver datasets, and sharing.
REST API	Application service	Registers sources, mappings, dataset jobs, reports, and shares.
Trino	Federated SQL layer	Queries registered sources without copying all raw data.
Spark	Batch processing	Applies transformations and builds curated datasets.
Delta Lake	Versioned storage	Freezes analytical datasets and stores version history.
MinIO	Object storage	Stores images, manifests, and Delta tables in an S3-compatible service.
Airflow	Orchestration	Executes ingestion, validation, and export workflows.
Semantic catalog	Mapping layer	Stores concepts, equivalence groups, confidence scores, provenance, and normalization rules.
Delta Sharing	Controlled export	Exposes selected datasets to authorized recipients.

Delta Lake was selected because transactional table management over object storage supports schema enforcement and versioned analytical tables [2]. Spark was selected for large-scale batch processing and dataset materialization [21]. The architecture follows the data-fabric idea of metadata-driven access and governance across distributed systems rather than mandatory centralization [8, 10]. Health terminology standards such as SNOMED CT, UMLS, and FHIR guide semantic direction, although the current prototype focuses on tool-level mapping workflows rather than full-service terminology integration [4, 6, 15].

5 Experimental Evaluation

The experiments evaluate three engineering capabilities: controlled federated execution, semantic retrieval and normalization, and reproducible dermatology dataset construction. Table 2 summarizes the evidence and scope of each experiment.

Table 2: Evaluation Scope and Interpretation

Experiment	What was evaluated	Supported interpretation	Main limitation
Federated query performance	Trino-based workloads with $N = \{1, 2, 4, 8\}$ remote connections and 30 measured runs per scenario.	The prototype remained reliable in a controlled small-scale deployment.	Does not prove broad scalability or superiority over centralized alternatives.
Semantic layer	Baseline keyword interface compared with semantic mappings for five retrieval and normalization tasks.	Semantic mappings reduced manual alignment effort and improved retrieval quality in the tested tasks.	Internal evaluation; a larger independent user study is required.
Dermatology dataset construction	Three public metadata collections, a fixed Bronze version, three Silver products, repeated builds, and version-pinned sharing.	The tool constructed, reproduced, inspected, and delivered the tested datasets.	Controlled metadata experiment rather than a clinical deployment.

5.1 Experiment 1: Federated Query Performance

The first experiment evaluated the virtualization layer under a controlled deployment. We registered four topologies with $N = \{1, 2, 4, 8\}$ remote connections and evaluated three workloads: *Bronze pushdown*, which keeps filtering close to the source; *Bronze federated*, which issues a federated read across registered sources; and *Silver persistent*, which materializes the curated result. Each scenario was executed 35 times. The first five runs were discarded as warm-up executions, leaving 30 measured runs per scenario and 360 measured executions overall.

For each run, we collected latency, rows processed, rows output, size in bytes, number of files, CPU and memory usage, and failures. Only one measured failure occurred, in the Bronze federated workload with $N = 4$, corresponding to 3.33% for that scenario. Figure 7 shows the performance summary. Bronze federated had the most stable workload, with mean latencies of 4.53 s at $N = 1$, 4.69 s at $N = 2$, 9.45 s at $N = 4$, and 7.25 s at $N = 8$. The $N = 4$ result should be interpreted with caution because it included the only measured failure and the highest variability. Bronze pushdown and Silver persistent became more expensive as the number of remote connections increased.

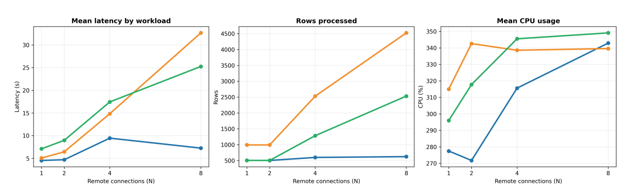

Figure 7: Federated query performance with 30 measured runs per scenario after warm-up removal. The result supports small-scale reliability of the prototype.

5.2 Experiment 2: Semantic Retrieval and Normalization

The second experiment compared a baseline keyword interface with the semantic interface. Both accessed the same federated dataset. The semantic condition enabled concept hierarchies, data-dictionary terms, value mappings, and transitive-equivalence groups. The tasks covered five concepts: sex/gender, diagnosis, lesion type, anatomical site, and age group.

The semantic condition improved the measured retrieval and normalization metrics in all five concepts. Average F1 increased from 0.874 in the baseline condition to 0.968 with semantic mappings. Average normalization accuracy increased from 0.748 to 0.965. Manual configuration effort also decreased: the average number of selected source columns dropped from 5.2 to 1.6, and average configuration time decreased from 259.8 s to 93.8 s. Query execution time increased slightly, from 0.74 s to 0.84 s. Figure 8 shows the summary.

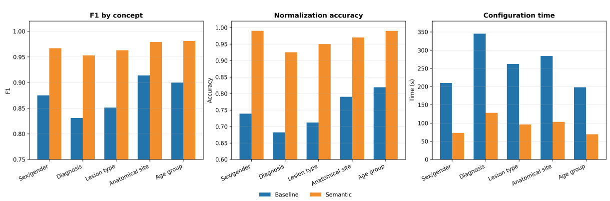

Figure 8: Semantic-layer evaluation. The semantic condition improved F1, normalization accuracy, and configuration effort in the tested tasks.

This experiment should be read as a tool evaluation rather than a definitive user study. The tasks were representative of the integration workflow, but the evaluation was conducted by researchers involved in the project.

5.3 Experiment 3: Governed Dermatology Data Products

We evaluated a frozen metadata snapshot from HAM10000, HIBA Skin Lesions, and PAD-UFES-20 [13, 14, 18]. The collections were registered as three isolated relational sources containing 11,720, 1,616, and 2,298 rows, respectively. Because no cross-collection patient key exists, the sources were combined by union with source-scoped identifiers rather than by an unsupported patient join. Bronze version 0 contained 15,634 unique image identifiers, 11,975 source-scoped lesions, and 1,996 source-scoped patients where identifiers were available. It retained the original source, lesion, patient, diagnosis, modality, confirmation method, phototype, attribution, and license fields. A versioned catalog harmonized diagnosis, recorded sex, modality, phototype, and anatomical site while preserving the original values and provenance.

Before backend execution, we froze three parallel Silver definitions (Table 3) and a separate local implementation of their expected image-identifier sets. G2 and G3 applied their filters after semantic harmonization. G4 evaluated an experiment-defined exact metadata-label policy; it is not a legal determination. Each product was built three times locally and materialized as three Delta versions in the backend.

Table 3: Governed Silver Products from One Bronze Snapshot

Product	Predeclared Silver rules	Images
G2	Dermoscopic; MEL, BCC, or SCC; histopathology confirmed	2,689
G3	Clinical; MEL, BCC, or SCC; histopathology confirmed; recorded phototype and patient ID	1,299
G4	Exact source license label `CC-BY`	3,914

All three outputs had 100% compliance with their frozen rules. G2 contained 1,768 lesions and 361 identified patients; patient identifiers covered 19.8% of its images. G3 contained 1,042 lesions and 807 patients, and G4 contained 3,137 lesions and 1,996 patients. Across the nine backend versions, row counts and condition-specific semantic hashes were stable, and every backend image-identifier set exactly matched the separate local oracle. Materialization took 9.08 ± 5.80 s and versioned reads took 2.28 ± 0.83 s (mean ± sample standard deviation over nine runs); these timings are descriptive because the products differ in size and the first run included a cold start. An authorized Delta Sharing recipient retrieved version-0 Parquet bytes after the producers advanced to version 2; an invalid token and a valid but unassigned recipient were rejected. This validates the exercised sharing path, not every external client.

We also audited unit integrity using deterministic 70/15/15 SHA-256 partitions. Under image-level partitioning, repeated lesion identity affected 33.4% of G2 images and 19.7% of G3 images; repeated patient identity affected 31.1% and 39.0% of the images with corresponding identifiers. Lesion grouping removed lesion leakage but not patient leakage, whereas grouping by patient when available and otherwise by lesion removed both measured modes (Figure 9). These are split-leakage proxies, not model-performance results.

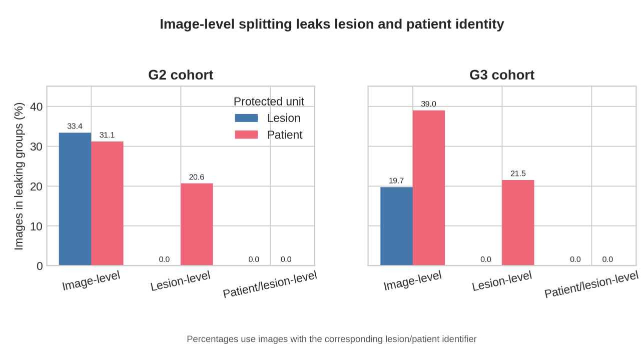

Figure 9: Identity leakage under deterministic split strategies. Percentages use images with the corresponding lesion or patient identifier.

Finally, the G3 patient counts for recorded Fitzpatrick phototypes I-VI were 77, 536, 171, 19, 4, and 0. Only I–III met the predeclared $n \geq 30$ descriptive gate; source-specific evidence was still narrower (Figure 10). This threshold is neither a power analysis nor a fairness criterion, and recorded Fitzpatrick phototype is not a direct skin-color measurement. Thus, the experiment demonstrates a fairness-aware data-engineering audit construction, versioning, provenance, identity, and evidence sufficiency not diagnostic effectiveness or a fair clinical model.

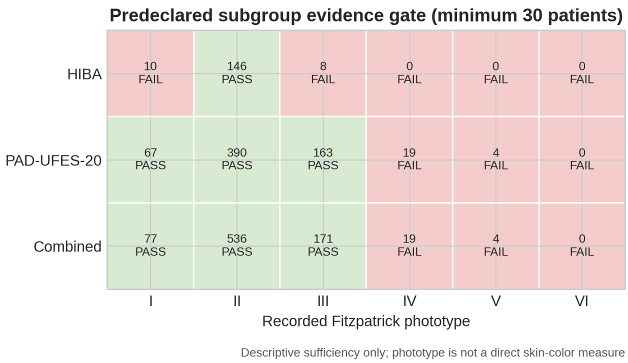

Figure 10: Patient-level evidence-sufficiency gate by recorded Fitzpatrick phototype. PASS indicates only that the predeclared descriptive threshold of 30 source-scoped patients was reached.

6 Related Tools and Approaches

DataFabric-DermAlert combines capabilities that are usually separated across data-management and clinical-research platforms. i2b2 provides an enterprise clinical-research warehouse, patient-set querying, and project data marts [12]. The OMOP Common Data Model maps observational databases to a shared schema and vocabulary [19]. DataSHIELD sends approved analyses to data-holding nodes and returns non-disclosive summaries, while Vantage6 executes containerized algorithms across distributed nodes without pooling their records [7, 11].

DataFabric-DermAlert addresses a different part of the workflow. It registers heterogeneous sources, records local semantic decisions, preserves source-level Bronze versions, materializes transformation-bearing Silver datasets, and pins shared releases. It may supply curated datasets to common-data-model or federated-analysis environments. The contribution evaluated here is the integration of source relationships, versioned equivalence groups, Bronze snapshots, Silver transformation contracts, and recipient-scoped shares in one inspectable tool.

7 Limitations

The current prototype has four main limitations. First, it has not yet been validated in a real multi-hospital SUS deployment. The demonstration uses controlled sources and synthetic institutional schemas. Second, Experiment 3 evaluates dermatology metadata engineering rather than image classification or prospective clinical use, and HAM10000 does not provide the patient and phototype fields used in the patient-level analyses. Third, the semantic evaluation was conducted internally and should be repeated with independent users and stronger experimental controls. Fourth, the tool can expose subgroup composition and selected fairness-related metrics, but it cannot certify fairness. Clinical fairness requires dermatology-specific datasets, clinical review, statistical analysis, and prospective validation.

8 Conclusion

DataFabric-DermAlert is a software tool for federated health data integration and fairness-aware dataset construction. The tool supports a concrete workflow: users define federated relationships, curate semantic equivalence groups, manage Bronze and Silver datasets, and share curated outputs through controlled recipients. The evaluation shows that the prototype can execute controlled small-scale federated workloads, support semantic retrieval and normalization in the tested tasks, and reproducibly construct, validate, version, audit, and share multi-source dermatology metadata products. The evidence supports the tool as a data-engineering artifact. It does not support broad clinical claims about fairness in dermatology or skin cancer detection.

Future work will focus on stronger independent usability evaluation, larger workload baselines, terminology-service integration, dermatology-specific datasets, institutional deployment profiles, and privacy-preserving extensions to federated learning [3, 9].

Artifact Availability

The complete DataFabric-DermAlert artifact package is permanently archived at https://doi.org/10.6084/m9.figshare.33042992.v3. The package includes the backend and frontend source code, Docker Compose deployment files, connector examples, semantic dictionary templates, dataset-building scripts, experiment-replication material, and demonstration data.

Frozen source-code releases are also available at https://github.com/DermAlert/datafabric-backend/releases/tag/sbes2026-artifact-v1.0.0 and https://github.com/DermAlert/datafabric-frontend/releases/tag/sbes2026-artifact-v1.0.0, respectively.

The demonstration video is available at https://figshare.com/s/d7a7e4d111f96353c3ef.

Acknowledgements

This study was financed in part by the project "Educação em saúde e triagem precoce de cancer de pele: Data fabric e machine learning em saúde, no 10/2023- PROGRAMA FAPDF LEARNING, FAP-DF".

References

[1] N. Alipour, T. Burke, and J. Courtney. 2024. Skin Type Diversity in Skin Lesion Datasets: A Review. *Current Dermatology Reports* 13, 3 (2024), 198–210.
[2] Michael Armbrust, Tathagata Das, Andres Torres, Burak Yavuz, Shixiong Zhu, Reynold Xin, Ali Ghodsi, Ion Stoica, and Matei Zaharia. 2020. Delta Lake: High-Performance ACID Table Storage over Cloud Object Stores. *Proceedings of the VLDB Endowment* 13, 12 (2020), 3411–3424.
[3] Jeffrey Casaletto, Anne Bernier, Robert McDougall, and Melissa S. Cline. 2023. Federated Analysis for Privacy-Preserving Data Sharing: A Technical and Legal Primer. *Annual Review of Genomics and Human Genetics* 24 (2023), 347–368.
[4] Arindam Chatterjee, Nibedita Pahari, and Andreas Prinz. 2022. HL7 FHIR with SNOMED-CT to Achieve Semantic and Structural Interoperability in Personal Health Data: A Proof-of-Concept Study. *Sensors* 22, 10 (2022), 3756.
[5] Roxana Daneshjou, Kailas Vodrahalli, Roberto A. Novoa, Michelle Jenkins, Weixin Liang, Veronica Rotemberg, Justin Ko, Susan M. Swetter, Elizabeth E. Bailey, Olivier Gevaert, et al. 2022. Disparities in Dermatology AI Performance on a Diverse, Curated Clinical Image Set. *Science Advances* 8, 32 (2022), eabq6147.
[6] Bruno H. de Mello, Sandro J. Rigo, Cristiano A. da Costa, Rodrigo da Rosa Righi, Bruna Donida, Marta R. Bez, and Luciano C. Schunke. 2022. Semantic Interoperability in Health Records Standards: A Systematic Literature Review. *Health and*

Technology 12, 2 (2022), 255–272.

[7] Amadou Gaye, Yannick Marcon, Julia Isaeva, Philippe LaFlamme, Andrew Turner, Eleanor M. Jones, Joel Minion, Andrew W. Boyd, Christopher J. Newby, Marja-Liisa Nuotio, et al. 2014. DataSHIELD: Taking the Analysis to the Data, Not the Data to the Analysis. *International Journal of Epidemiology* 43, 6 (2014), 1929–1944. doi:10.1093/ije/dyu188

[8] A. Gieß and A. Hutterer. 2025. The Future of Data Management: A Delimitation of Data Platforms, Data Spaces, Data Meshes, and Data Fabrics. *Information Systems and e-Business Management* 23, 4 (2025), 971–997.

[9] Hua Guan, Pew-Thian Yap, Andrea Bozoki, and Mingxia Liu. 2024. Federated Learning for Medical Image Analysis: A Survey. *Pattern Recognition* 151 (2024), 110424.

[10] A. Macías, D. Muñoz, E. Navarro, and P. González. 2023. Digital Twins-Based Data Fabric Architecture to Enhance Data Management in Intelligent Healthcare Ecosystems. In *Proceedings of the International Conference on Ubiquitous Computing and Ambient Intelligence (Lecture Notes in Networks and Systems, Vol. 594)*. Springer, 38–49.

[11] Arturo Moncada-Torres, Frank Martin, Melle Sieswerda, Johan van Soest, and Gijs Geleijnse. 2020. VANTAGE6: An Open Source Privacy-Preserving Federated Learning Infrastructure for Secure Insight Exchange. In *AMIA Annual Symposium Proceedings*, Vol. 2020. 870–877.

[12] Shawn N. Murphy, Griffin Weber, Michael Mendis, Vivian Gainer, Henry C. Chueh, Susanne Churchill, and Isaac Kohane. 2010. Serving the Enterprise and Beyond with Informatics for Integrating Biology and the Bedside (i2b2). *Journal of the American Medical Informatics Association* 17, 2 (2010), 124–130. doi:10.1136/jamia.2009.000893

[13] Andre G. C. Pacheco, Gustavo R. Lima, Amanda S. Salomão, Breno Krohling, Igor P. Biral, Gabriel G. de Angelo, Fábio C. R. Alves Jr., José G. M. Esgario, Alana C. Simora, Pedro B. C. Castro, et al. 2020. PAD-UFES-20: A Skin Lesion Dataset Composed of Patient Data and Clinical Images Collected from Smartphones. *Data in Brief* 32 (2020), 106221. doi:10.1016/j.dib.2020.106221

[14] María Agustina Ricci Lara, María Victoria Rodríguez Kowalczuk, Maite Lisa Eliceche, María Guillermina Ferraresso, Daniel Roberto Luna, Sonia Elizabeth Benitez, Luis Daniel Mazzuoccolo, et al. 2023. A Dataset of Skin Lesion Images Collected in Argentina for the Evaluation of AI Tools in This Population. *Scientific Data* 10 (2023), 712. doi:10.1038/s41597-023-02630-0

[15] Ramesh Saripalle, Mehdi Sookhak, and Majid Haghparast. 2020. An Interoperable UMLS Terminology Service Using FHIR. *Future Internet* 12, 11 (2020), 199.

[16] Raghav Sethi, Martin Traverso, Dain Sundstrom, David Phillips, Wenlei Xie, Yutian Sun, Nezih Yegitbasi, Haozhun Jin, Eric Hwang, Ninad Shingte, and Christopher Berner. 2019. Presto: SQL on Everything. In *Proceedings of the IEEE 35th International Conference on Data Engineering*. 1802–1813.

[17] Yutian Sun, Timothy Meehan, Rebecca Schlussel, Wenlei Xie, Marina Basmanova, Orri Erling, Alexandre Rosa, Stephen Fan, Rongrong Zhong, Aravind Thirupathi, et al. 2023. Presto: A Decade of SQL Analytics at Meta. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 189:1–189:25.

[18] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions. *Scientific Data* 5, 1 (2018), 180161.

[19] Erica A. Voss, Rupa Makadia, Amy Matcho, Qianli Ma, Chris Knoll, Martijn Schuemie, Frank J. DeFalco, Ajit Londhe, Vivienne Zhu, and Patrick B. Ryan. 2015. Feasibility and Utility of Applications of the Common Data Model to Multiple, Disparate Observational Health Databases. *Journal of the American Medical Informatics Association* 22, 3 (2015), 553–564. doi:10.1093/jamia/ocu023

[20] David Wen, Saad M. Khan, Andrew J. Xu, Habiba Ibrahim, Luke Smith, Jorge Caballero, L. Zepeda, C. de Blas Perez, Alastair K. Denniston, Xiaoxuan Liu, and Rubeta N. Matin. 2022. Characteristics of Publicly Available Skin Cancer Image Datasets: A Systematic Review. *The Lancet Digital Health* 4, 1 (2022), e64–e74.

[21] Matei Zaharia, Reynold S. Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J. Franklin, Ali Ghodsi, Joseph Gonzalez, Scott Shenker, and Ion Stoica. 2016. Apache Spark: A Unified Engine for Big Data Processing. *Commun. ACM* 59, 11 (2016), 56–65.