IntuitionLabs

INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

Data Layer Architecture for AI Scientific Research

> Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-03-15 · data layer architecture · fair data principles · scientific data management · ai research infrastructure · data pipelines · machine learning workflows · metadata management

Original article: https://intuitionlabs.ai/articles/data-layer-architecture-ai-scientific-research



In this article

1. Executive Summary
2. Introduction and Background
3. The Need for a Specialized Data Layer
4. Core Components of the Data Layer
5. 1. Data Sources and Ingestion
6. 2. Storage and Data Management
7. 3. Data Processing and Pipelines
8. 4. Metadata, Catalogs, and Semantic Layer
9. 5. Access and Interfaces
10. 6. Governance, Quality, and Sustainability
11. Data Integration and Enrichment
12. Heterogeneous Data Fusion
13. Knowledge Graphs and Semantic Web
14. Semantic Layer for AI
15. Case Study: Materials Data Integration
16. Tools, Technologies, and Platforms
17. Distributed Storage and Compute
18. Data Processing Frameworks
19. AI and Analytics Tools
20. Hybrid and Emerging Approaches
21. Data Layer in Practice: Case Studies
22. Case Study 1: High-Energy Physics Data Pipelines
23. Case Study 2: Materials Discovery Infrastructure
24. Case Study 3: Healthcare and Life Sciences
25. Case Study 4: Machine-First FAIR in Scholarly Data
26. Data Quality, Governance, and FAIR Principles
27. FAIR Principles in the Data Layer
28. Data Quality Assurance
29. Data Principles Beyond FAIR
30. Sustainability
31. Analysis and Evidence-Based Discussion
32. Implications, Opportunities, and Future Directions
33. Conclusion
34. References

Executive Summary

The modern era of scientific research is characterized by an unprecedented deluge of data and the rise of powerful artificial intelligence (AI) methods to analyze that data. Building a robust **data layer** – the underlying infrastructure and processes that collect, curate, store, and serve data – is critical to enable AI-powered science. A well-architected data layer integrates diverse data sources (from sensors, simulations, experiments, and literature), ensures data quality and interoperability, and provides efficient pipelines for data preparation and access for AI models. This report provides an in-depth, evidence-based exploration of how to construct such a data layer specifically tailored for AI-driven scientific discovery.

Key insights include:

- **Data Deluge in Science:** Scientific domains are generating **massive, complex datasets** (e.g. petabytes per second at high-energy physics detectors (db-blog.web.cern.ch), billions of health records (www.nature.com)). This “big data” era has ushered in a new paradigm of “big data-driven science” or “AI-enabled discovery,” where machine learning models exploit the richness of data to accelerate insights (www.nature.com).
- **FAIR and Machine-Actionable Data:** To harness AI effectively, data must be **Findable, Accessible, Interoperable, and Reusable (FAIR)**, especially for machines (www.nature.com) (graphwise.medium.com). Numerous initiatives (e.g. FAIR4HEP, Materials Data Facility, Common Fund Data Ecosystem) emphasize FAIR data and metadata standards so that AI algorithms can discover, integrate, and learn from scientific data (www.nature.com). Machine-first data design is crucial; researchers have advocated prioritizing machine-readability to unlock AI-driven knowledge discovery (www.digital-science.com).
- **Architectural Components:** A high-quality data layer comprises multiple components: *(a)* **Data ingestion pipelines** that capture raw data from experiments, simulations, and external sources; *(b)* scalable and reliable **storage** (e.g. distributed file systems, object stores, data lakes) to hold raw and processed data; *(c)* **Data processing and curation tools** (e.g. ETL workflows, feature engineering) that clean, transform, and annotate data for AI; *(d)* **Metadata catalogues and knowledge graphs** to index and interlink data, enhancing discoverability and integration (www.nature.com); *(e)* **Access and query services** (APIs, portals) that allow scientists and AI models to retrieve data; and *(f)* **Governance frameworks** for data quality, provenance, privacy, and compliance (e.g. using FAIR principles and privacy-preserving methods) (www.nature.com). These components must work together through orchestrated pipelines to feed AI workflows.
- **Technological Tools and Platforms:** Building the data layer leverages a diverse toolchain. Cloud platforms (AWS, Google Cloud, etc.) offer scalable storage and ML services, but many scientific organizations also rely on High-Performance Computing (HPC) systems with specialized file systems (e.g. Lustre, GPFS) and GPU clusters. Container and orchestration technologies (e.g. Kubernetes, Kubeflow, Argo) enable portable data pipelines on HPC and cloud (link.springer.com). Data processing frameworks like Apache Spark, Dask, and Hadoop provide distributed computation for large datasets

(db-blog.web.cern.ch). Specialized databases (graph databases, NoSQL document stores) capture complex scientific data models. Emerging “data fabric” and “data mesh” architectures offer unified or decentralized approaches to manage data across silos, often linked by semantic technologies.

- **Case Studies and Examples:** Diverse scientific fields illustrate these principles. In **high-energy physics** (HEP), detectors generate ~1 PB/s of raw data, requiring hierarchical triggers and pipelines that convert ROOT-format sensor data into Apache Spark tables and Parquet files for ML models (db-blog.web.cern.ch). In **materials science**, open platforms and knowledge graphs (e.g. Open Quantum Materials Database, Materials Project) curate millions of compounds with rich metadata, powering ML models for property prediction (www.nature.com). In **biomedical research**, initiatives like the NIH’s Common Fund Data Ecosystem and Korea’s health claims database adopt common data models (OMOP-CDM) and distributed analytics to make billions of health records FAIR and available for AI-driven analysis (www.nature.com). Each case underscores tailored data layer design: from sensor ingestion and labeling in astronomy to interoperable ontologies in genomics.
- **Integration of AI and Data Systems:** Modern “AI factories” integrate data pipelines with model training. For example, the KubePipe project demonstrates running many ML workflows in Kubernetes clusters, abstracting parallelism and containerization so scientists can focus on models rather than infrastructure (link.springer.com). Scientific data layers often include **Feature Stores** (to cache intermediate features), **Model Zoos** (libraries of pretrained models), and connectors from data services to machine learning platforms.
- **Governance, Privacy, and Ethics:** Data layers must also address governance. FAIR principles guide metadata and sharing, while regulations (GDPR, HIPAA) dictate privacy controls especially for human-sensitive data (e.g. in healthcare or social science). Techniques like data de-identification, federated learning, and secure enclaves enable model training without exposing raw sensitive data. Ethical considerations—such as bias in training data and transparent AI practices—are integral to a science-focused data layer.
- **Future Trends:** The frontier includes **federated AI networks** for cross-organizational science without moving data, **ontologies and semantic web services** enabling richer data integration, and **explainable AI** requiring lineage tracking in data pipelines. “Machine-first” approaches suggest next-gen scholarly infrastructure from birth will embed AI-readiness: structured data, linked open vocabularies, and direct machine interfaces (www.digital-science.com). Large language models and generative AI promise to not only consume scientific data but to generate new hypotheses, further looping back into the data ecosystem.

In conclusion, building the data layer for AI-powered scientific research requires **holistic planning**: identifying data sources and requirements, deploying robust storage and pipelines, enforcing FAIR and governance standards, and continuously integrating new AI methods. The result is an infrastructure that magnifies the scientific return on data by enabling advanced analytics, improving reproducibility, and accelerating discovery. The following sections delve into each aspect in detail, supported by case studies, data analyses, and expert perspectives.

Introduction and Background

Scientific progress has always been driven by data – from early experiments documented in notebooks to the modern era of terabyte-scale observations. In recent decades, three major forces have reshaped scientific data ecosystems:

1. **Big Data Growth:** Advances in instrumentation, sensors, and computation have exploded data volume, variety, and velocity in nearly every field (www.nature.com). Scientific data rates vary by instrument and processing stage. SKAO's pipeline overview, accessed October 20, 2026, projects over 700 petabytes of archived data per year (SKAO); this retained-data volume should be distinguished from the raw detector throughput discussed below. For example, the Large Hadron Collider produces on the order of *1 petabyte per second* raw data in its detectors (db-blog.web.cern.ch). Traditional analytics cannot cope, requiring new infrastructure for ingesting, storing, and processing such streams.
2. **Data-Driven "Second Scientific Revolution":** The availability of massive datasets has shifted scientific methodology. Machine learning (ML) and AI techniques can uncover patterns and predictions beyond human-coded models. As a recent review notes, the 21st century ushered in an era of "big data-driven science," where data analytics and ML transform discovery in fields like materials science, climate modeling, and biology (www.nature.com). In materials science, for instance, large experimental and computational databases now allow ML models to predict material properties and reveal hidden correlations (www.nature.com). Generative AI further enables the design of entirely new molecules and materials without explicit theoretical models (www.nature.com). Across domains, from astrophysics to epidemiology, AI acts as a powerful tool to sift through data deluges for insights.
3. **Global and Collaborative Research Landscape:** Scientific research is increasingly collaborative and cross-disciplinary. Global data-sharing initiatives (e.g. CERN's Open Data, NASA's Earth Observations, bioinformatics genomics consortia) mean that data layers must support federated access and common standards. There is a growing emphasis on **open science** and **FAIR principles**, where data must be not only accessible, but also semantically interoperable and reusable (www.nature.com). Funding agencies (NIH, DOE, NSF, EU) now often mandate data management plans and FAIR stewardship. In parallel, major philanthropic and industrial investments (e.g. Chan Zuckerberg Initiative, Navigation Fund, AWS) highlight the strategic importance of data infrastructure for science (www.digital-science.com).

These trends create both opportunities and challenges for scientific data infrastructure. AI capabilities promise accelerated discovery, but only if the underlying data *layer* is well-designed. The "data layer" can be understood as the stack of systems that collect, index, store, and serve data to analytics and AI. It sits between raw data sources (experiments, sensors, simulations, publications) and the algorithms that analyze them. A robust data layer solves heterogeneity in formats, ensures metadata richness, and provides scalable pipelines. In practice, it encompasses **storage (file systems, databases)**, **compute resources**, **networking**, **catalogs and metadata registries**, **tools for data curation and cleaning**, and **APIs or interfaces** for data access. Conceptually, it is the domain "infrastructure" for science: much as lab equipment forms the physical layer for experiments, the data layer is the digital bedrock enabling AI-enhanced experiments and analyses.

This report examines *how* to construct such a data layer for AI-powered research. We will define its components, review design patterns and technologies, and discuss practical concerns (e.g. metadata, scalability, governance). Historical context illustrates how scientific discovery has transformed through data-driven approaches, and current practices illuminate state-of-the-art solutions and gaps. Case studies from particle physics, materials science, healthcare, and other fields exemplify real implementations. Finally, we consider future directions: emerging paradigms like data mesh, knowledge graphs, federated learning, and the sociotechnical shifts toward truly machine-centric data sharing (www.digital-science.com). The goal is to provide a comprehensive, evidence-based guide, backed by the latest research and expert insights, on building a data layer fit for 21st-century AI-driven science.

The Need for a Specialized Data Layer

Scientific data has unique characteristics that distinguish it from typical enterprise data. These characteristics drive the requirements for a specialized data layer:

- **Volume and Velocity:** Scientific instruments can produce insurmountable streams of data. For example, the Vera C. Rubin Observatory's LSST is expected to deliver roughly 20 terabytes of imaging data per night (issc.science.lsst.org); weather satellites generate terabytes per day; genomic sequencers produce gigabytes per experiment. HEP detectors see collision events every 25 ns, resulting in raw data flow on the order of *1 PB/s* (db-blog.web.cern.ch) (though vast triggers quickly downselect). Handling such magnitudes requires high-throughput pipelines and distributed storage architectures.
- **Variety:** Data spans many formats: from raw sensor reads and time series to complex structured outputs (e.g. images, spectra, molecular structures). A single project may aggregate heterogeneous sources – e.g. climate research integrates satellite images, in situ sensor logs, simulation model outputs, and unstructured field notes. The data layer must accommodate structured tables, scientific image formats (FITS, NetCDF, HDF5), graph/sequence data, etc., and often *semantic integration* is needed to unify them.
- **Integrity and Provenance:** Scientific inference demands traceability. Every datum (e.g. measurement, simulation output) must carry provenance metadata specifying how it was produced, processed, and by whom. Data transformations (filtering, calibration, feature derivation) must themselves be documented. A robust data layer includes metadata catalogs or knowledge graphs that record lineage, enabling reproducibility and validation (www.nature.com). For AI applications, provenance is also crucial to understand biases and uncertainties in models.
- **Longevity:** Scientific data can be relevant for decades or longer (e.g. archived genome sequences, astronomical catalogs). The data layer must ensure durable storage and format sustainability. It must also handle continuous data accumulation – e.g. yearly solar observations, or successive wet-lab experiments – while preserving backward compatibility. Metadata standards (Dublin Core, domain ontologies) help ensure reusability.

- **Accessibility and Collaboration:** Data layers often span institutions and countries. Data may be subject to access controls (e.g. human-subject health data), or licensing (e.g. some genomics). The infrastructure should support controlled sharing (via portals, APIs, or federated query) so that authorized researchers can access pooled data, possibly without copying (e.g. through federated learning). Initiatives like the European Open Science Cloud (EOSC) exemplify efforts to integrate national research data across borders.
- **Computational Context:** Unlike static business data, scientific data is routinely consumed by large-scale simulations and AI workflows. The layer should integrate closely with compute resources: e.g. data locality to HPC clusters or near GPU farms, in situ processing for streaming data, and support for parallel I/O. High-performance filesystems (Lustre, GPFS) or parallel object stores are common in supercomputers. Efficient data ingestion pipelines (using high-speed networking, RDMA, streaming abstractions) are vital.

These requirements translate into concrete architectural needs. In particular, supporting AI adds demands beyond traditional data warehousing:

- **AI-Ready Data Preparation:** AI models often require labeled or curated datasets. The data layer must include tools for data cleaning, normalization, augmentation, and labeling. This includes not just raw data cleaning but generating training/testing splits, synthetic data generation, and feature stores. The data layer often stages data into formats suited for ML frameworks (NumPy arrays, TFRecords, etc.).
- **Scalability and Flexibility:** As AI models grow (e.g. physics-guided neural networks, huge language models trained on scholarly texts), the data layer must scale. Cloud infrastructures offer scalable elastic resources, but many scientific users continue to use on-prem HPC clusters. A flexible data layer might span both, using hybrid models (some data on institutional clusters, with overflow to cloud buckets when needed).
- **Metadata and Semantic Layer:** For AI research, simple keyword search is insufficient. Advanced search (e.g. query by features, similarity search) and semantic reasoning can accelerate data discovery. Knowledge graphs can unify disparate datasets via rich metadata (see Section on Semantic Integration). For example, linking publications, datasets, and models via an ontology can reveal new correlations for AI to exploit.
- **Quality Control and Validation:** In enterprise settings, “garbage in, garbage out” also holds: poor data quality degrades AI performance. The data layer must include validation checks (schema conformance, anomaly detection) and versioning of datasets. For example, in medical imaging, a data layer might flag inconsistent scan parameters or missing patient info before model training.

In summary, a data layer for AI-powered science must not simply store data; it must actively *prepare and shepherd* data for intelligent analysis. It is an ecosystem combining distributed systems, data engineering pipelines, and domain-specific curation. The remainder of this report details these components and processes, illustrating best practices and lessons learned through scientific data-intensive projects.

Core Components of the Data Layer

A practical data layer can be visualized in *layers and modules*, each responsible for a part of the data lifecycle. These typically include: **(1) Data Sources and Ingestion**, **(2) Storage Systems**, **(3) Processing and Transformation**, **(4) Metadata and Catalog**, **(5) API and Access Services**, and **(6) Governance and Operations**. Below we describe each component and its role.

1. Data Sources and Ingestion

Data Sources

Scientific data originates from a wide variety of sources. Examples include:

- **Experimental Instruments**: e.g. particle detectors, telescopes, laboratory sensors (mass spectrometers, genomic sequencers). These often have specialized formats (RAW, FITS, ROOT, etc.).
- **Simulations and Models**: e.g. climate models, molecular dynamics, computational physics codes. Simulation outputs can be structured time-series, synthetic images, or binary dumps.
- **Observational Networks and IoT**: e.g. sensor networks monitoring environmental conditions (weather stations, ocean buoys), Internet-of-Things labs (distributed biomedical monitors).
- **Data Repositories and Archives**: e.g. public datasets like GenBank, CERN Open Data, Environmental Data Corp. These may be large archives or continuously updated streams.
- **Literature and Unstructured Data**: e.g. scholarly articles, lab notebooks, images, which can be mined (using NLP or computer vision) to extract structured data.

Data Ingestion

Ingestion is the pipeline that reliably captures data from sources into the data layer. Key practices include:

- **CDC (Change Data Capture) and Streaming**: For live/incremental data (e.g. real-time sensors, lab equipment outputs), streaming platforms (Apache Kafka, MQTT, Flink) can ingest data at high throughput. For instance, an astronomical observatory might stream telescope images and catalog triggers as events for immediate processing.
- **Bulk Transfer and ETL**: For large static uploads (e.g. nightly raw dumps, satellite passes), batch transfers via high-speed protocols (Globus, GridFTP, Aspera) are used. Data is often transferred to intermediate storage (staging area) before long-term cataloging. ETL (Extract-Transform-Load) tools then convert raw files into standardized forms (e.g. root->Parquet, raw images->tiles).
- **Automated Pipelines**: Tools like Apache NiFi or Tekton (Kubernetes-based) orchestrate ingestion workflows, handling retries and monitoring. Some science domains build custom pipelines; for example, the LSST alert system was designed to distribute about 10 million alerts per night (issc.science.lsst.org).
- **Data Validation at Ingress**: Early validation (schema checks, quality filters) helps catch errors sooner. In healthcare, for example, incoming claims data may be validated for coding consistency before conversion to the OMOP model (www.nature.com). In genomics, sequence reads may be quality-

trimmed on ingest.

Table 1 outlines common data sources and ingestion techniques by domain:

Domain	Data Source	Ingestion Method	Examples
High-Energy Physics	Detector raw data (ROOT format)	Real-time triggering + batch archival	CERN LHC experiments (1 PB/s raw triggered to ~GB/s) (db-blog.web.cern.ch)
Astronomy	Telescope images, event alerts	Scheduled transfers, streaming for alerts	LSST alert streams (LSST:UK)
Earth Science	Satellite imagery, sensor networks	High-bandwidth links, Planetary Data Systems	NOAA GOES satellite data
Genetics & Biology	Sequencer reads, protein structures	Cloud upload, S3 buckets	1000 Genomes, PDB
Healthcare	Clinical records, insurance claims	Secure ETL pipelines, HL7/FHIR interfaces	HIRA (Korea) claims (www.nature.com)
Experimental Materials	Lab instruments, robotics logs	Local collection + laboratory LIMS	DCIM (Digital Curation) frameworks
Publications	Research articles, patents, images	Bulk ingestion (FTP, API), web scraping	Semantic Scholar ingestion of PDFs (www.digital-science.com)

Table 1: Examples of Scientific Data Sources and Ingestion Methods. (LIMS = Laboratory Information Management Systems)

2. Storage and Data Management

Once data is ingested, it must be stored in a way that is scalable, durable, and accessible. Storage architecture typically comprises tiers or “data lake” zones:

- **Raw Data Storage (Bronze Layer):** A low-level, immutable store for raw ingested data. Often implemented with distributed file systems (Hadoop HDFS, Amazon S3, Ceph). It preserves original files/filesets unaltered for reproducibility. For example, the CERN EOS system archives raw CMS detector files (db-blog.web.cern.ch).
- **Processed Data Storage (Silver/Gold Layers):** Data transformed into analysis-ready form (e.g. cleaned tables, feature matrices). Stored in columnar databases or Parquet tables for fast analytics. This may include merging multiple raw sources. In the CERN ML pipeline, ROOT files were converted to Spark DataFrames and then to Parquet tables as intermediate output (db-blog.web.cern.ch).
- **Metadata Catalog/Index:** Not a “store” per se, but integral: metadata (see Section 4) about files, experiments, schemas is indexed in a searchable database.

Scientific data layers also often include:

- **Hierarchical Storage Management (HSM):** Automated migration between hot (SSD, NVRAM) and cold (tape archives) storage. Large projects keep petabytes cold (tape) and bring into HPC scratch only subsets needed for processing.
- **Object Storage Layers:** Many modern systems favor object stores (S3-compatible) for raw and processed data due to scalability. HPC centers increasingly provide object APIs on traditional storage. The Materials Data Facility (MDF) collects TB-scale materials datasets and indexes contents for queries (www.nature.com).
- **Database Systems:** For tabular or semi-structured data, relational (PostgreSQL/MySQL) or NoSQL databases (MongoDB, Cassandra) may be used. Graph databases (Neo4j, AWS Neptune) store interconnected metadata (see Section 4).

Scalability and Performance: Scientific computations are often I/O-bound. High-throughput parallel file systems (Lustre, GPFS) are standard in HPC. Cloud object stores scale linearly but require tuning (e.g., partitioning keys to avoid hot spots). Databases need careful schema design (normalized vs. wide tables) and sharding for large data. Many projects use a hybrid: HPC scratch for high-speed needs and long-term object store for archival.

Data Lake vs. Warehouse: Traditional data warehouses (highly structured) have limitations in science, because schema evolve and new data types appear. Data “lake” approaches, where schema-on-read is allowed, are common. This is supported by big data frameworks (Spark, Dask) reading semi-structured or raw files and applying transformations dynamically. However, some curated “data warehouses” of cleaned data (e.g. the OMOP healthcare CDM) are still used for standardized analytics (www.nature.com).

Table 2 presents a conceptual comparison of common data architectures:

Architecture	Description	Strengths	Limitations
Data Lake	Centralized storage of raw data (often in object store or distributed FS). Schema applied on-read. Scales to large volumes.	Flexible schema; stores all data formats; cost-effective for massive storage	Data can become a “swamp” without governance; hard to query without metadata; risk of duplication.
Data Warehouse	Structured, curated data store (relational/OLAP). Enforces schema-on-write.	Fast queries on known schema; good for BI dashboards; ACID transactions.	Rigidity: difficult to adapt new data types; not ideal for unstructured/binary data; costly scaling.
Data Lakehouse	Hybrid approach (e.g. Delta Lake, Apache Iceberg) combining lake storage with ACID transactions and schema enforcement.	Balances flexibility and reliability; allows versioning of datasets.	Emerging tech; depends on vendor/stability; performance tunings needed.
Data Fabric	Overlay layer providing unified data access across disparate storage. Often includes virtualization and metadata services.	Hides location; provides unified API; supports governance and lineage across silos.	Can be complex; requires robust catalog; overhead of virtualization.
Data Mesh	Decentralized architecture where domains own their data “products” with standardized APIs. Governance is federated.	Scales organizationally; reduces central bottleneck; domain-driven.	Hard to coordinate; risk of inconsistency; requires cultural shift and catalog standards.

Table 2: Comparison of Data Architecture Patterns.

In practice, scientific projects often blend these. For example, one might use a central data lake for raw and intermediate data, implement a data fabric with metadata catalog (to make it findable), and foster a data mesh by allowing individual research groups to manage their own “domain data products” within that fabric.

3. Data Processing and Pipelines

Once stored, data must be processed for usefulness. The data layer implements pipelines to transform *raw data into AI-ready datasets*. Key pipeline stages include:

- **Cleaning and Standardization:** Removing noise or errors, standardizing units/terminology. E.g., aligning gene names to standard nomenclature, harmonizing time zones in sensor logs, filtering out instrument artifacts in images.
- **Feature Extraction/Engineering:** Converting raw signals into feature vectors or inputs. For image data, this might include object detection or segmentation; for time series, extracting statistical metrics; for text, generating embeddings. These steps often happen on the compute cluster using tools like Spark or Dask.
- **Data Labeling/Annotation:** Especially in supervised learning, data must be annotated. The data layer may include services for manual or automated labeling. Some workflows incorporate domain experts (e.g., radiologists labeling scans). When manual labeling is infeasible, techniques like weak supervision or synthetic augmentation are used.
- **Data Integration and Fusion:** Combining datasets from multiple sources, e.g. linking patient health records with genomic data. This involves join operations, record linkage (matching IDs), and possibly “semantic” integration via ontologies to align different vocabularies. The output is unified tables or graph structures.
- **Aggregation and Summarization:** Large-scale pooling (e.g. averaging sensor readings, computing histograms). For some analyses (like training deep nets), raw data might be tens of terabytes; aggregation reduces to manageable size.
- **Pipelines Orchestration:** Scientific pipelines are often complex DAGs of tasks (data ingestion → cleaning → transform → analysis). Tools like Apache Airflow, Argo Workflows, or Nextflow can orchestrate these steps. For instance, Argo on Kubernetes is used to chain containerized steps, enabling reproducibility and retry logic (link.springer.com).

Research on ML pipelines emphasizes their importance. A CERN blog notes that ML pipeline lies at the heart of success for HEP projects, integrating diverse components across the chain (db-blog.web.cern.ch). Another large-scale project, the Materials Data Facility, provides user-friendly tools to automate indexing and transformation of materials data, enabling pipelines that ingest TB of X-ray data or millions of files into queryable form (www.nature.com).

Standardization of pipeline steps is developing. For example, ISO/IEC 23053 (AI systems ML frameworks) outlines generic steps (data acquisition, preprocessing, modeling, deployment) which many data layer pipelines adhere to (link.springer.com). Publications like DataBench classify pipelines into phases: Data acquisition/storage, preparation/curation, analytics, and action/interaction (link.springer.com). These frameworks guide scientists in structuring their data flows.

Containerization and Parallelism

Parallel processing is essential. Data pipelines often use distributed computing (Hadoop/Spark, Dask) on clusters. An emerging best practice is containerization: packaging each stage as a Docker or Singularity container ensures reproducibility. Tools like KubePipe abstract this complexity, enabling scientists to run ML workflows over Kubernetes without deep expertise in parallel computing (link.springer.com). KubePipe launches concurrent ML tasks in containers, optimizing resource use on clusters.

Scientific pipelines also leverage specialized frameworks:

- **Apache Spark**: for large-scale batch processing (often used in CERN and genomics).
- **Dask**: for flexible parallel Python, integrated into HPC.
- **MPI-based tools**: for highly parallel simulations.
- **GPU acceleration**: for deep learning model training and even data pre-processing (e.g. GPU-accelerated Augmentation).
- **Serverless and Functions-as-a-Service**: in cloud setups can handle sporadic small tasks (e.g. on-demand queries of catalog).

Data Validation and Monitoring

Continuous data **validation** is part of processing. This may involve automated anomaly detection (e.g. outlier flux values flagged in telescope data) or schema validation (ensuring tables have expected columns, ranges). Scientific pipelines often include automated validation checks with alerts on deviation.

Equally important is **observability**: logging and metrics for pipelines. Tracking data throughput, errors, and resource usage helps troubleshoot. For example, the CERN ML pipeline logs data sizes and job latencies at each stage to improve efficiency. Tools like Prometheus and Grafana are used to monitor the data layer stack's health.

4. Metadata, Catalogs, and Semantic Layer

Arguably the cornerstone of a useful data layer is **metadata**: data about data. Metadata makes raw datasets **findable and interpretable**, linking them to context (authors, experiments, relationships). For AI, rich metadata allows algorithms to discover appropriate training data and understand feature semantics.

Metadata Catalogues

Most data platforms include a centralized or federated **data catalog**. This catalog indexes all datasets with descriptive metadata: title, creators, data type, keywords, date, format, location, licenses, and more. Advanced catalogs also track lineage (parent/child dataset relationships) and quality metrics.

For example, the Material Data Facility automatically harvests metadata from published datasets and provides an index. The NIH Common Fund Data Ecosystem (CFDE) offers a portal enabling search across multiple biomedical datasets via metadata tags (www.nature.com). CERN's Data Lake catalogs store have been integrated with metadata from HEP experiments, facilitating discovery of the right data for a given analysis.

The **Schemas and Standards** used for metadata vary by field. Generic standards like Dublin Core, DataCite, or schema.org provide basic fields. Domain ontologies (Gene Ontology, MeSH for biomedical, ObsCore for astronomy) add semantic richness. The FAIR principles emphasize metadata as keys to findability and interoperability (www.nature.com). In practice, implementing rich metadata often requires community consensus. Projects like FAIR4HEP and ESCAPE focus on developing community metadata standards in physics and astronomy (www.nature.com).

Knowledge Graphs and Semantic Integration

Beyond catalogs, some advanced data layers incorporate **knowledge graphs**: graph-structured databases linking datasets, publications, concepts, and data items through semantic relationships. Knowledge graphs unify heterogeneous data by focusing on relationships (e.g. "Experiment E *produced* Dataset D", "Material M *has property* X").

The "Open Research Knowledge Graph" (ORKG) concept envisions a semantic network of research contributions. In materials science, researchers have built knowledge graphs that encode measurements from literature and repository data. Deagen *et al.* (2022) created a knowledge graph for polymer nanocomposites, enabling interactive visualizations and queries (www.nature.com). Their approach combined SPARQL (semantic web query language) for data retrieval with Vega-Lite for visualization, demonstrating how linked data can empower AI and human users alike.

Knowledge graphs support **query-based analytics**: AI algorithms (graph neural networks, SPARQL-based reasoning) can traverse the graph to find relevant connections. They also underpin data integration: e.g. mapping different vocabularies (one graph node could be linked to equivalent terms in another ontology). Scientific communities increasingly adopt semantic web technologies (RDF, OWL) to achieve interoperability (www.nature.com).

Data Documentation

Metadata also includes documentation: codebooks, data dictionaries, schema definitions, and even provenance logs. For AI applications, knowing how data was collected is essential. Tools like W3C PROV or specific data management plan (DMP) repositories can document procedures. Embedding structured README files (e.g. in Data Packages) or using the Frictionless Data toolkit helps standardize these details.

The data layer should enforce **machine-actionable metadata**. This means not only having human-readable documentation, but structured fields that AI can parse: for example, JSON-LD or RDFa embedded in data portals, MLflow tracking for experiments, or ontologies accessible via API. The "Machine-first FAIR" principle argues for structuring data around machine consumption (www.digital-science.com), as exemplified by large language model (LLM) datasets that are published with full metadata and schemas.

5. Access and Interfaces

Once data and metadata are in place, the data layer must provide reliable interfaces for users and AI systems to retrieve it:

- **APIs and Services:** RESTful APIs, GraphQL endpoints, or gRPC services allow programmatic access to data. For example, CERN experiments expose APIs to query archived events, while climate archives use OPeNDAP to serve large gridded datasets. A common approach is to build microservices: a search API queries the metadata catalog, a data API streams raw or processed data, etc.
- **Portals and GUIs:** Web-based portals enable scientists to browse and download data. These often include visualization or summary tools. E.g., the NIH CFDE portal lets researchers filter biomedical datasets by tissue type or assay technology. The Materials Project API allows web queries of material properties. Semantic Scholar provides a GUI for scholarly literature search but is underpinned by a massive data layer of parsed papers (www.digital-science.com).
- **Data Lake Query Engines:** Tools like Hive, Presto, or BigQuery allow SQL-like queries directly on data lakes. This is useful for analysts doing ad-hoc exploration. Researchers in genomics use BigQuery to rapidly query large sequence variant tables.
- **Notebook Interfaces:** Integration with Jupyter or RStudio hubs is increasingly common. Data layers may mount datasets directly into cloud-based notebooks (e.g. via S3 filesystem integration) so scientists can code in Python/Julia/etc. The Data and Learning Hub for Science (DLHub) and funcX, mentioned by the NSF “Garden” project, facilitate remote access to data and compute for science .
- **Authentication/Authorization:** Science data often requires authentication (institutional logins, API keys) and fine-grained access control (e.g. access to controlled-use genomic data via dbGaP). The access layer must integrate identity providers and honor consent and usage restrictions, sometimes through Data Use Ontology tags.

It is worth noting that the interface is the AI’s point of contact with data as well. Platform-as-a-Service (PaaS) offerings for ML (e.g. AWS SageMaker, Google AI Platform) often ingest data via these APIs or direct S3 mounts. Containerization (above) ensures that pipelines can connect to the same data URLs as user applications.

Example: Federated Data Access

In some fields, data cannot be centralized due to privacy or logistics. Federated query systems allow analysis across distributed data. In the HIRA implementation, HIRA reviews submitted analysis queries and runs the code within its own environment (study). This is a controlled code-to-data workflow; a multi-holder federation would be a separate architectural design. Similarly, astronomy often federates access (e.g., querying data from several observatory archives via Virtual Observatory standards). Designing the data layer to support *federated queries* (SPARQL endpoints, federated SQL engines) is an advanced but increasingly important capability.

6. Governance, Quality, and Sustainability

The data layer is not a one-time deployment but a living system requiring governance and operational discipline:

- **Governance Policies:** These fix data standards, metadata requirements, and responsibilities. For example, a governance board might mandate that all datasets include certain metadata fields (keywords, license) before ingestion. Collaborative platforms (like EOSC or RDA) often provide policy frameworks.
- **Data Quality Assurance:** Periodic audits, schema validations, and user feedback loops help maintain high data quality. Scientific consortia sometimes convene workshops to discuss data best practices (e.g., the FAIR for AI workshop at Argonne (www.nature.com); FAIRplus initiative guides on metadata standards). Automated tools (e.g. https://howtofair.dk/) guide on FAIR compliance.
- **Versioning and Provenance:** Datasets evolve. The data layer should employ version control (DVC, Quilt, Delta tables) so analyses can pin to a specific data snapshot. Provenance metadata (timestamps, checksums) ensures reproducibility. For very large data, lightweight versioning (augmenting metadata rather than duplicating files) and immutable object numbering are used.
- **Preservation and Archiving:** Long-term archiving policies (e.g. migrate to new file formats, maintain tape archives) extend the life of scientific data. Some communities designate trusted repositories (e.g. CERN Open Data Portal, NIST Materials Data Facility) for preservation. The data layer must plan for periodic reviews or infrastructure refresh to avoid obsolescence.
- **Security and Privacy:** This covers data at rest and in transit. Encryption, firewalls, and compliance (e.g. HIPAA for patient data) are data-layer concerns. Use of secure computation (homomorphic encryption, Yao's protocol) is emerging for privacy. The data layer often liaises with institutional security teams.
- **Monitoring and SLOs:** Service level objectives (uptime, latency) are set for key data services. Continuous integration / continuous deployment (CI/CD) pipelines are used for data platform software to ensure reliability. Incident response plans (e.g. for data loss) are part of operational governance.

In aggregate, governance ensures that the data layer remains a trustworthy partner in research. Without it, data silos, orphan datasets, or inconsistent formats can undermine AI initiatives. The FAIR initiatives emphasize that cultural incentives (citations of datasets, funding mandates) must align with building good data layers for science.

Data Integration and Enrichment

One of the central goals of a scientific data layer is **integration**: making diverse datasets work together. This involves addressing heterogeneity, aligning semantics, and extracting latent information.

Heterogeneous Data Fusion

Scientific data is heterogeneous at multiple levels (schema, modality, scale). Integration strategies include:

- **Schema Mapping:** Aligning different database schemas through transformation rules. For example, two climate datasets may use different variable names (temp vs temperature), units (Kelvin vs Celsius), or dimensions. ETL scripts or tools (Talend, Pentaho) can remap these into a common schema.
- **Coordinate Standardization:** For spatiotemporal data, ensuring consistent reference frames (time zones, coordinate reference systems). This is critical in geosciences.
- **Record Linkage:** For tabular data lacking a common ID, probabilistic matching (e.g. linking patient records across hospitals, or matching chemical compounds by structure) is used. Duke is a general-purpose deduplication and record-linkage engine written in Java (developer documentation).
- **Entity Resolution via Ontologies:** When merging data semantically, ontologies play a key role. A materials dataset might use different terms for the same material property; linking these via an ontology (e.g. UMLS in biomedical, EMO in materials) creates a unified view. This can be done via knowledge-graph approaches (mapping nodes by "owl:sameAs" links).
- **Harmonization:** Sometimes raw data is aggregated to a common unit or grain. For example, aggregating daily weather station observations to monthly averages to compare with climate model outputs. The data layer should support such generic “scale bridging” operations.

Knowledge Graphs and Semantic Web

As introduced, **Knowledge Graphs (KGs)** are powerful for integration. In the context of AI data layers, KGs serve multiple purposes:

- **Unified Schema-less Representations:** Unlike relational tables, graph structures can flexibly represent entities and relationships without a fixed schema, accommodating new data without restructuring.
- **Inferencing:** With ontologies and reasoning engines, knowledge graphs can infer new connections (e.g. if dataset A is linked to author X and dataset B to the same author, a KG can discover they may be related). AI systems can exploit these inferences.
- **Query Flexibility:** SPARQL and Cypher queries over KGs allow complex queries (nearly natural language-like) over data that would require complex joins in SQL.

Example: The ORKG (Open Research Knowledge Graph) project is building a graph-based scholarly knowledge database, tagging contributions (problems, methods, results) from papers in structured form. AI tools can query ORKG to find all materials discovered with a given technique, etc. In environmental science, an ontology-driven KG might combine sensor readings, species databases, and policy documents, facilitating cross-domain AI analyses.

Implementing KGs typically involves:

- RDF triple stores (Apache Jena, Virtuoso) or property graphs.
- Ontologies (OWL) defining classes and relations.
- ETL processes to load triples from data (e.g. converting CSV with RDFa).
- APIs for SPARQL or GraphQL access.

One challenge is keeping KGs up-to-date. Incremental updating (via streaming triples) is feasible, but full rebuilds may be needed when schemas change. Proper modelling upfront mitigates churn.

Semantic Layer for AI

From an AI perspective, semantic integration improves model inputs. If a machine learning pipeline is informed by a knowledge graph, it can embed semantic context (e.g. encode hierarchical relations or constraints into feature engineering). Recent research explores **KG embeddings**: mapping graph nodes to numeric space so ML models can “understand” the KG structure (www.nature.com).

Moreover, natural language processing (NLP) can enrich structured data: e.g. training LLMs on research papers can yield knowledge about scientific entities; this knowledge can in turn label or annotate raw data. For example, an AI pipeline might use an LLM to tag images (via captions) which are then linked in the KG.

Case Study: Materials Data Integration

In materials science, data is extremely heterogeneous (from atomic simulations, experiments, to manufacturing logs). The Materials Data Facility (MDF) and related projects have tackled integration by:

- **Standard Metadata Schemas**: Defining what metadata fields materials data must have (composition, processing conditions, measured properties).
- **Automated Indexing**: Tools that crawl repositories for materials files and extract metadata, populating catalogs.
- **Combine Datasets**: Using shared vocabularies (e.g. the Materials Ontology) to align terms.
- **Knowledge Graphs**: The Citrine platform and the Materials Project store combine computational and experimental data with graph-like query capability for materials discovery.

Deagen et al. specifically demonstrate using a graph to unify data from experiments on polymer composites, enabling “FAIR” interactive graphics linked to the underlying KG (www.nature.com). This is an exemplar of enriching data with semantic structure to support both scientists and AI.

Tools, Technologies, and Platforms

Building a data layer requires selecting appropriate tools for each component. Here we survey key technologies that support the architecture.

Distributed Storage and Compute

- **Cloud Platforms:** AWS, Google Cloud, Azure all provide managed storage (S3, GCS, Blob), databases, and AI services. AWS offers specialized high-throughput storage (FSx for Lustre) and ML (SageMaker) that integrate with data lakes. Separately, NSF describes its National Artificial Intelligence Research Resource (NAIRR) as a vision for shared national research infrastructure supporting responsible AI discovery and innovation (NSF). However, cost and data egress must be considered.

- **HPC Clusters:** Many research institutions rely on supercomputers with parallel file systems. These handle simulation and data crunching tasks. Tools like Lustre provide >100 GB/s bandwidth. HPC centers increasingly add GPU nodes for AI. Summit is a historical example: it was decommissioned on November 15, 2024 (OLCF user guide).
- **Hybrid and Edge:** The data layer may extend to edge devices (e.g. local data caches at sensor networks) or use hybrid cloud-bursting for spiky workloads. The key is connectivity (low-latency links, federated identity).

Data Processing Frameworks

- **Big Data Engines:** Apache Spark (and Spark MLlib) is widely used in scientific pipelines for scalable ETL and ML. Hadoop/YARN clusters also exist but Spark (on Kubernetes or EMR) has largely supplanted raw Hadoop MapReduce. Dask is increasingly popular for Pythonic parallel workflows, and RAPIDS offers GPU-accelerated DataFrames.
- **Database and Query Engines:**
- *SQL Engines*: PostgreSQL or MySQL for smaller and relational tasks; new distributed SQL DBs (Amazon Redshift, Google BigQuery) for petabyte-scale SQL analytics.
- *NoSQL*: MongoDB or ScyllaDB for loosely-structured data; Elasticsearch for text search indexing in data catalogs.
- *Time-Series DBs*: InfluxDB or OpenTSDB for high-resolution time-stamped experimental data.
- *Graph DBs*: Neo4j, JanusGraph, AWS Neptune for knowledge graphs.
- **Workflow Orchestrators:** Mentioned above (Argo, Airflow, Nextflow). For example, Nextflow is popular in bioinformatics for chaining genomics tools; Argo is used extensively on Kubernetes for generic pipelines (link.springer.com).
- **Container Platforms:** Docker, Singularity for encapsulating pipeline tasks. Kubernetes clusters (on-prem or EKS/GKE) run containerized workloads. Tools like Kubeflow coordinate ML workflows end-to-end (including data copying, training, serving). KubePipe (2025) as an open-source tool explicitly targets parallelizing ML jobs on Kubernetes (link.springer.com).
- **Data Catalog Metadata Tools:** Apache Atlas and Amundsen (LinkedIn) provide automated data discovery and lineage in data lakes. EOSC uses EUDAT CDA. Many groups build domain-specific catalogs (e.g. semantic scholar for papers, which is essentially a data layer).
- **Machine Learning Ops (MLOps):** MLflow, MLMD (Google), DVC provide experiment tracking, model registry, and metric logs — all part of data governance. They connect to the data layer by storing references to data versions.
- **Semantic Tools:** RDF/SPARQL engines (Blazegraph, GraphDB) and ontology editors (Protégé) for knowledge graphs. The Python *rdflib* library or *graph-tool* library aid scientists in smaller tasks.

Below we list some representative technologies by category:

Component	Example Technologies
Storage	Hadoop HDFS, Amazon S3, Ceph, Lustre, GPFS, Object storage
Databases	PostgreSQL, Cassandra, MongoDB, Neo4j, InfluxDB
Compute	Spark, Dask, Hadoop, MPI, Kubernetes, CUDA/GPU clusters
Orchestration/Pipelines	Apache Airflow, Argo Workflows, Nextflow, Kubeflow, Tekton
Containerization	Docker, Singularity, Podman
Metadata Catalog	Apache Atlas, CKAN, Dataverse, Custom RESTful catalogs
Search/API	Elasticsearch, Solr, GraphQL, REST frameworks (FastAPI, Django)
Semantic	RDF triple stores (Virtuoso), OWL ontologies, SPARQL endpoints
Monitoring	Prometheus, Grafana, ELK stack

Table 3: Key technologies for data layer components.

AI and Analytics Tools

Since the layer is designed for AI, certain tools deserve emphasis:

- **Feature Stores:** Emerging pattern in AI systems. (Feast, Tecton) These store pre-computed features indexed by entity/time for reuse in training and serving.
- **ML Frameworks:** TensorFlow, PyTorch, scikit-learn are not data layer per se, but the data layer must support their inputs (TFRecords, numpy arrays). The data pipelines often write outputs in formats directly readable by these frameworks.
- **Large-Scale ML Tools:** For massive models, distributed training frameworks such as Horovod, AWS Sagemaker distributed, or custom MPI on GPUs are relevant. The data layer should supply data fast enough to feed multi-node training.
- **Pre-trained Models and Model Zoos:** The data layer can include a repository of models (Sci ML Commons, model garden). The GARDEN initiative (NSF-funded) is building repos where models link to FAIR datasets and papers , running on platforms like datahub (DLHub). Model metadata is part of the data ecosystem.
- **Autotuning and AutoML:** Tools like Optuna, Metaflow for hyperparameter tuning, which may run simulations of pipelines with varying data subsets.

Hybrid and Emerging Approaches

- **Serverless/Function-as-a-Service:** Used for event-driven small tasks (e.g. transform a new data file as it arrives). For example, AWS Lambda triggered on object upload to run a quick ETL step.

- **Blockchain/Provenance Chains:** Some have explored blockchain for immutable data provenance (though overhead and scalability are concerns). The materials infrastructure mentioned “blockchain-based mechanisms for secure data sharing, provenance, and traceability” (www.nature.com), though such use is still experimental.
- **GPU and Accelerator Storage:** As AI drives GPU-heavy workloads, specialized storage close to accelerators (GPUDirect storage, NVMe) can reduce bottlenecks. Distributed NVMe over Fabrics is an emerging HPC trend.

In designing the data layer, the ecosystem of tools must be chosen to fit the organization’s scale, domain needs, and expertise. Often, open-source platforms are preferred for scientific openness, whereas some enterprises leverage commercial cloud services for scale.

Data Layer in Practice: Case Studies

To ground these principles, we examine real-world examples where data layers were built for AI or data-intensive research.

Case Study 1: High-Energy Physics Data Pipelines

Context: Modern particle physics experiments (e.g. ATLAS/CMS at CERN) produce staggering data volumes. As described by Canali *et al.*, the LHC’s detectors collect ~1 PB/s, with collisions every 25 ns, but only ~1–10 GB/s can be stored after triggering (db-blog.web.cern.ch). AI/ML is increasingly used to improve event selection and analysis.

Data Layer Components:

- **Ingestion and Triggers:** In real-time, hardware triggers filter events; the surviving data are written in ROOT format to CERN’s EOS storage.
- **Storage:** CERN uses EOS (a custom object storage) plus tape archives. Raw ROOT files are ingested into a raw data lake.
- **Processing Pipeline:** Canali’s team built an ML pipeline using Apache Spark to process simulated events stored in ROOT files, rather than live detector acquisition data. The steps were:

1. **Data Ingestion:** Read ROOT files from EOS into Spark DataFrames, then write as Parquet tables (db-blog.web.cern.ch).
2. **Feature Engineering:** Filter events and compute features for neural network input.
3. **Model Training:** Use BigDL/Analytics Zoo on Spark to train classifiers.
4. **Feedback:** The classifier is intended for real-time triggers to improve purity.

This pipeline leveraged Spark’s distributed processing.

- **Tools and Tech:**
- **Apache Spark** for large-scale ETL and integration (instead of legacy HEP pipelines).

- **HDFS/EOS** for storage; Spark reads directly from EOS via xrootd or HDFS interface.
- **BigDL and Analytics Zoo** deep learning libraries on Spark for model computation.
- **Notebook Interfaces:** The team provided Jupyter notebooks on the cluster for users to interact with pipeline.
- **Importance of Pipelines:** The blog emphasizes that data pipelines were “paramount” for ML success (db-blog.web.cern.ch). Before using Spark, HEP relied on custom C++ workflows. By adopting open-source big data tools, they achieved easier sharing and maintenance.
- **Scalability:** The solution was tested on hundreds of cores, processing millions of events. The pipeline enables repeating classification experiments more efficiently. It exemplifies integration of scientific storage (ROOT/EOS) with enterprise-style data platforms (Spark).

Discussion: This case highlights key aspects of a scientific data layer:

- Handling extremely high throughput and converting to AI format.
- Using general-purpose tools (Spark) rather than only domain-specific software.
- Importance of data curation (only events of interest).
- Challenges: integrating legacy formats (ROOT) with big data tools.

Case Study 2: Materials Discovery Infrastructure

Context: The Materials Science community has embraced data-centric research, with large databases of computed and experimental properties. The 2026 Communications Materials review by Salas *et al.* outlines AI-driven infrastructure for materials discovery (www.nature.com).

Data Layer Components:

- **Open-source Platforms:** The review emphasizes *open-source tools* that unify data acquisition, modeling, and deployment (www.nature.com). Examples include:
- **Materials Project, OQMD, NOMAD:** large repositories of material properties with APIs.
- **Materials Cloud:** a platform that provides integrated data, computational notebooks, and visualizations.
- **Knowledge Graphs:** Efforts like Citrine, the “Materials Data Facility” (MDF), and property graph databases encode materials, processes, and properties. They use ontologies (FabricML, MSDL).
- **Self-driving Labs:** Advanced data ingestion from automated experiments feeding ML models. E.g., robots perform high-throughput synthesis and characterization, with data streamed to ML pipelines that decide next experiments (www.nature.com).
- **Secure Data Sharing:** The review mentions **blockchain** mechanisms for secure provenance and sharing in supply chains (www.nature.com), indicating interest in immutable audit trails for data.
- **AI Cloud Integration:** The shift to cloud+edge is noted for scalability (e.g. AWS SageMaker for model training). Big companies (AWS, Google) contribute open data: e.g. AWS published large chemical dynamics datasets that integrate with cloud ML tools.

- **Sustainability and Ethics:** The data layer also tracks environmental impact. The review discusses “lifecycle-aware design” and ethical AI, implying metadata for energy use and social factors.

Case Example: The MDF has collected 80+ TB of materials data from nearly 1000 datasets (www.nature.com). Datasets with millions of files were indexed using Elasticsearch and made queryable (supports parameter-based and text search) (www.nature.com). The data layer here included:

- A central index linking to distributed data sources.
- APIs for data discovery (via Globus and MDF portal).
- Automated curation: volunteers and AI tools tag materials compositions.

Additionally, Salas *et al.* discuss generative models in materials: e.g. GANs used to propose new compounds (www.nature.com). These models require high-quality training data and benefit from standardized data layers (consistent representation of material structures and labels).

Discussion: The materials case underscores:

- **Interoperability:** The importance of linking disparate tools (simulation codes, experiments) via common frameworks.
- **Reproducibility:** Open datasets and shared models (often with DOI) support reproducibility.
- **Industrial Integration:** Linkage to manufacturing/IoT suggests a data layer spanning research and production (Industry 4.0 / 5.0 context) (www.nature.com).
- **Quantitative Impact:** Precise metrics (data volume collected, user queries) are used to track infrastructure success.

Case Study 3: Healthcare and Life Sciences

Context: Biomedical research exemplifies sensitive, high-volume data requiring strict governance. The J-S Kim *et al.* study (Scientific Data 2023) presents Korea’s nationwide health insurance claims database transformed for research (www.nature.com).

Data Layer Components:

- **Common Data Model:** All data (~10 billion claims, 56 million patients) were converted into the **OMOP CDM** (version 5.3.1), a schema used by the OHDSI community (www.nature.com). This standardization is a classic data layer strategy: diverse institutional records become uniform tables.
- **Infrastructure:**
- A distributed research environment (privacy-by-design) was built. The claims data remained within HIRA (the Health Insurance Review and Assessment Service, an independent claims-review and quality-assessment agency; HIRA history) and not all moved centrally. Instead, analytics code ran within the HIRA secure environment, and only aggregate results were shared (www.nature.com).
- Data were physically stored in relational databases conforming to the CDM schema. The entire table set (conditions, drugs, etc.) contained hundreds of millions of rows.

- Metadata about the tables (number of records, variable definitions) was documented and made *FAIR*.
- **Scale and Quality:** The paper reports conversion of all tables with very high fidelity (e.g. 99% of condition codes mapped to standard terms) (www.nature.com). The entire population-level data for 11 years was available in research-ready form.
- **Open Data and APIs**: While raw data cannot be made public, HIRA published the *metadata* and access procedures, aiding transparency. They also demonstrated replication of scientific studies (Type 2 diabetes incidence, COVID-19 models) using the infrastructure (www.nature.com).

Discussion: This highlights:

- **Governance Model:** National oversight enabled building a data layer at country scale. It overcame issues of non-interop and reproducibility by enforcing a standard model (www.nature.com).
- **Distributed Analytics:** Embodies the federated approach: “bringing the code to the data” due to privacy. The data layer thus included analysis nodes next to the databases.
- **Large Cohort Analytics:** Examples of output (10 million+ records, patient-level rates) show how the data layer feeds population AI. Machine learning models on claims can now be trained on the standardized OMOP data.

This case underscores that sometimes the data layer is as much about governance and modeling decisions (OMOP CDM) as about technology. For AI, having a consistent schema across a massive dataset is invaluable.

Case Study 4: Machine-First FAIR in Scholarly Data

Context: The academic publishing ecosystem itself can be viewed as a data layer for research knowledge. Mark Hahnel of Digital Science has articulated the need for data (papers, datasets) to be FAIR not just for humans but for machines (www.digital-science.com).

Data Layer Components:

- **Knowledge Graph of Literature:** Projects like Semantic Scholar ingest millions of papers and extract metadata and citation graphs. These form a kind of data layer for science: an index of what is known. AI services query this graph.
- **Trained Foundation Models:** Allen Institute’s OLMo and others have published entire training datasets and code in machine-readable form (www.digital-science.com). These open science efforts use the data layer of training corpora (including metadata about sources).
- **Financial Backing for Data:** Venture funding in AI for science has targeted data infrastructure: e.g., Chan-Zuckerberg’s multi-billion investments in computational biology clusters and virtual platforms (www.digital-science.com) reflect strategic building of data/compute layers.
- **Machine-Actionable Metadata:** In his November 2025 “Machine-first FAIR” post, Mark Hahnel argues that institutional data often remains optimized for human consumption rather than computational processing (Digital Science). The push is to transform research outputs into structured, interlinked data.

Examples include using markdown/Jats XML for articles, embedding semantic annotations, and tagging datasets with rich schema.

Implication: While not a single project, this example shows a philosophical shift: building data layers at the level of *research knowledge itself*. It has led to tools like Kaggle for research, NL to SPARQL converters, and ontologies for research concepts (e.g. ORCID for people, RRIDs for resources).

Data Quality, Governance, and FAIR Principles

No data layer can succeed without strong data governance and quality control. The FAIR principles (Findable, Accessible, Interoperable, Reusable) provide a guiding framework (www.nature.com). Below we discuss their role and practical steps.

FAIR Principles in the Data Layer

- **Findable:** All datasets must be indexed with unique identifiers (DOIs, ARKs) and metadata. Catalog search and global indexes (e.g. re3data.org, DataCite) help scientists and AI discover relevant data. Persistent identifiers for datasets (and even for individual data items) are recommended.
- **Accessible:** Data should be retrievable via open protocols (HTTP, Globus). Even if access constraints apply, the metadata should state how to request. Data lakes with open APIs (or common authentication) satisfy this. In practice, many infrastructures use OAuth/OpenID or institutional logins (e.g. eduGAIN).
- **Interoperable:** This means using standard formats and vocabularies. The data layer enforces certified formats (NetCDF, CSV, JSON-LD) and ontologies. Deploying semantic standards (RDF, JSON-LD, domain ontologies) is a key interoperability tactic. The FAIR for AI community is actively defining what interoperability means for AI models and datasets (www.nature.com).
- **Reusable:** Data should have clear licenses and detailed provenance. The layer attaches licenses (CC-BY, CC0, etc.) to data. It also records context: experimental conditions, instrumentation, calibration. Without such context, data is not reusable. Machine-learning specific additions include documenting preprocessing steps and model performance bounds.

Checklist for FAIR Data Layer:

- Assign persistent IDs to datasets and code.
- Maintain a metadata registry catalog.
- Use open, standard data formats.
- Capture complete provenance of data transformations.
- Publish metadata and documentation (including DMPs).
- Adopt community ontologies (e.g. for chemistry: ChEBI; for health: SNOMED).
- Provide clear license and usage info.

The long list of initiatives in the FAIR for AI white paper shows the community's commitment to these ideals (e.g. FAIR4HEP, MDF, HPC-FAIR) (www.nature.com). Implementers should review these projects' guidelines.

Data Quality Assurance

Measures to ensure data is correct and fit-for-purpose include:

- **Automated Validation:** At ingestion or transformation, enforce schema checks (type, range), missingness thresholds, or cross-field consistency. For example, a climate dataset could fail QA if temperature values exceed physical limits.
- **Manual Curation:** Domain experts review subsets of data. Scientific data often requires human verification (e.g. labeling galaxies correctly in surveys). The data layer supports such feedback loops (e.g. annotation tools).
- **Provenance Tracking:** Little, if anything, is thrown away immediately. Early pipeline steps create intermediate artifacts with version tags. For example, after cleaning a dataset, the original is retained with a pointer in metadata. Making data and AI models FAIR helps clarify their content and context and supports transparent provenance and reproducibility (FAIR for AI).
- **Quality Metrics Dashboard:** Reporting metrics (coverage, completeness, error rates) can be automated. Some data platforms publish these to guide users and funders on data health.

Data Principles Beyond FAIR

- **CARE Principles:** Particularly for indigenous data or social data, the "CARE" (Collective benefit, Authority to control, Responsibility, Ethics) principles complement FAIR. For example, a genomics data lake may restrict certain tribal genomes to the controlling group, even if it is scientifically valuable.
- **Ethical AI Guidelines:** The data layer may incorporate fairness audits (e.g. statistical parity, bias checks on labels). One might tag datasets with known biases or gaps.
- **Security:** Classification levels (secret/restricted/public) might be applied. Encryption at rest and in transit, strict IAM policies, and audit logs are part of the layer.

Sustainability

Data infrastructure is expensive. Sustainability strategies include:

- **Funding Models:** Charging researchers for storage (or providing quotas) to allocate costs.
- **Community Support:** Open-source projects rely on volunteer maintainers and grant funding. Long-lived projects (e.g. Materials Project) often become community standards.
- **Standards and Interoperability:** Adhering to open standards avoids vendor lock-in; easier migration if technology changes.

Analysis and Evidence-Based Discussion

We now analyze the concepts above using data and expert input from the literature and case studies.

- **Productivity Gains from Pipelines:** Our CERN case shows adoption of an Apache Spark pipeline greatly simplified development compared to previous bespoke tools. The blog states that open-source "comes with several advantages, including reduced cost of development and the possibility of sharing solutions and expertise" (db-blog.web.cern.ch). In this light, the data layer's pipeline tools are not just technical – they foster collaboration among scientists.
- **Scale Metrics:** The Healthcare study provides concrete figures: 10.1 billion claims successfully converted into OMOP-CDM tables (www.nature.com). Such scale (billions of records) is rarely demonstrated in literature, showcasing what an enterprise-level scientific data layer can handle. Another numeric highlight: the Materials Data Facility reports 80+ TB across ~1000 datasets (www.nature.com). These illustrate that large data collection and integration is operationally feasible.
- **Return on Investment:** While hard to quantify in dollars, some insights emerge. For example, the Digital Science blog points to multi-billion-dollar investments by CZI, Navigation Fund, etc., into data/AI infrastructures (www.digital-science.com). This signals a perceived high value; academic studies show that accessible, well-managed data significantly accelerate research outcomes (e.g. the HIRA database supplying statistics used by the OECD for Korea (www.nature.com)). For AI research infrastructure, FAIR principles are also being adapted to models and datasets (FAIR for AI).
- **User Engagement and Community:** The FAIR for AI perspective highlights the community-driven nature of data layers; e.g. the RDA, CODATA, PUNCH4NFDI consortiums (www.nature.com). Data layers succeed when communities converge on practices. For instance, the credit-score-like recognition systems are emerging to reward good data publication (publication count for data sets, citations via DOIs).
- **Challenges and Gaps:** Despite progress, gaps are evident. The AI pipeline for authenticated scholarly data is still immature; many scientific outputs are not yet machine-actionable (www.digital-science.com). Tools for integrated provenance in multi-model pipelines are lacking. Real-time "data meshes" across federations are still largely aspirational. We see such gaps also in our literature review: many references (like Sci Data FAIR for AI) identify future research needs (e.g. self-driving labs not widespread yet (www.nature.com), privacy-preserving AI in science).

Implications, Opportunities, and Future Directions

A sophisticated data layer fundamentally changes scientific research. We conclude the report by exploring implications and future trends.

- **Democratization of Science:** Well-designed data layers lower barriers to entry. For example, the NSF's National AI Research Resource (NAIRR) aims to share data and compute nationwide (time.com). Cloud data platforms mean a small lab can leverage petabyte-scale public datasets with a few clicks. This broadens participation, but also requires training (AI literacy) to avoid misuse.
- **Convergence of HPC and AI:** HPC centers now often sponsor AI research, e.g. ECP (Exascale Computing Project) fosters merging HPC simulations with ML. TechRadar (2025) notes "convergence" of AI and HPC infrastructure (www.techradar.com). Data layers that seamlessly shift between simulation data and neural nets will define next-gen supercomputing.
- **On-the-Fly AI and Edge:** Future data layers will embed AI at data generation points. In-situ analytics (AI models running on sensors or at data acquisition) can filter or pre-process data, reducing storage needs. Self-driving laboratories (robotic experimentation coupled with immediate AI analysis) will produce data that flow directly into adaptive data pipelines (www.nature.com).
- **Responsible AI and FAIRness:** As the recent FAIR for AI workshop report (Nature Sci Data, 2023) suggests, defining FAIR for AI models and datasets is active work (www.nature.com). Future data layers will likely have **FAIR AI Model catalogs** (linking models to their training data provenance). Ethical oversight will become embedded: e.g. metadata fields recording whether data was bias-checked or consent-obtained.
- **Path to Fully Machine-Actionable Data:** The ultimate goal is that the majority of scientific data is natively machine-readable and integrable. Achieving "Machine-first FAIR" may involve rethinking publication itself (publishing code and data as first-class objects) (www.digital-science.com). Tools like semantic authoring aids, automated metadata generation (via AI), and data wrangling interfaces will help reach this.
- **Quantum and Beyond:** Looking ahead, quantum computing may handle certain data tasks; data layers may incorporate quantum-safe encryption and interfaces to quantum data stores. While nascent, data architectures should be flexible enough to integrate new paradigms (e.g. graph-specific hardware for knowledge graphs).
- **Global Collaboration:** Initiatives like the Einstein Telescope (gravitational waves) or SKA (radio astronomy) are planning intercontinental data fabric and federated clouds. Climate change research is pushing for unified Earth data exchange. The FAIR for AI paper's listings (PUNCH4NFDI, ESCAPE) show a movement toward shared European science clouds. The US is expanding NAIRR. The data layer is thus inherently international, requiring alignment on policies and tech.

In summary, **the data layer is the catalyst that transforms raw scientific data into collective knowledge via AI**. Its evolution will shape how science is done. A meticulously designed data layer amplifies AI's impact, but requires sustained multidisciplinary effort from data scientists, domain experts, and institutions.

Conclusion

The opportunities of AI in science are boundless, but realization depends critically on the underpinning data layer. This report has outlined a comprehensive vision for constructing such a layer: from ingesting heterogeneous data streams, through scalable storage and processing, to semantic integration and FAIR governance.

We have drawn on evidence from cutting-edge projects and literature to identify best practices. The themes are clear: design for scale, prioritize interoperability, and maintain rigorous provenance. Our case studies illustrate that even in the most data-intensive domains (particle physics, national health, materials discovery), these principles are not only theoretical but have been implemented with success.

Going forward, the data layer must continue adapting: embracing emerging AI paradigms, federated models, and open frameworks. Crucially, the community must converge on standards and incentives that make data sharing as routine as publishing papers. The FAIR for AI movement and initiatives like NIH's CFDE or NSF's Model Gardens are steps in this direction, weaving AI considerations into the fabric of data stewardship (www.nature.com) .

In closing, building a robust data layer is **an investment in the future of science**. It is the infrastructure that enables simulations, experiments, and observations to be transformed into insights by AI. Such an investment pays dividends: accelerating discoveries, reducing duplicated effort, and unlocking new interdisciplinary research. As Mark Hahnel observes, we must prioritize machines as first-class collaborators by giving them well-structured data (www.digital-science.com). The scientific enterprise that learns to do this effectively will lead the way in the AI-powered research revolution.

References

All factual claims and data in this report are referenced from peer-reviewed articles, technical reports, and authoritative sources. Inline citations (formatted as [URL]) are provided throughout. Key references include:

- E. Huerta *et al.*, *FAIR for AI: An interdisciplinary and international community building perspective*, *Scientific Data* **10** (2023) (www.nature.com).
- M. Salas *et al.*, *AI-powered open-source infrastructure for accelerating materials discovery and advanced manufacturing*, *Commun. Materials* **7** (2026) (www.nature.com).
- D. Suárez *et al.*, *KubePipe: a container-based high-level parallelization tool for scalable machine learning pipelines*, *J Supercomputing* **81**, (2025) (link.springer.com).
- L. Canali *et al.*, *Machine Learning Pipelines for High Energy Physics Using Apache Spark*, CERN DB Blog (2019) (db-blog.web.cern.ch).
- J.-W. Kim *et al.*, *Scalable Infrastructure Supporting Reproducible Nationwide Healthcare Data Analysis toward FAIR Stewardship*, *Scientific Data* **10**, Article 674 (2023) (www.nature.com).
- M. E. Deagen *et al.*, *FAIR and Interactive Data Graphics from a Scientific Knowledge Graph*, *Scientific Data* **9**, Article 239 (2022) (www.nature.com).

- M. Hahnel, *Machine-first FAIR: Realigning academic data for the AI research revolution*, Digital Science Blog (2025) (www.digital-science.com).
- Springer book chapter (B. Berre *et al.*), *Big Data and AI Pipeline Framework: Technology Analysis…* (2022) (link.springer.com).
- Plus numerous project websites and white papers (FAIR4HEP, Materials Data Facility, OMOP-CDM) as cited above.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://db-blog.web.cern.ch/blog/luca-canali/machine-learning-pipelines-high-energy-physics-using-apache-spark-bigdl
2. https://www.nature.com/articles/s41597-023-02580-7
3. https://www.nature.com/articles/s43246-026-01105-0
4. https://www.nature.com/articles/s41597-023-02298-6
5. https://graphwise.medium.com/how-knowledge-graphs-power-data-mesh-and-data-fabric-98ade5db93bf
6. https://www.digital-science.com/blog/2025/11/machine-first-fair-academic-data-for-the-ai-research-revolution
7. https://www.nature.com/articles/s41597-022-01352-z
8. https://link.springer.com/article/10.1007/s11227-025-06956-x
9. https://www.skao.int/nl/node/24
10. https://issc.science.lsst.org/pages/DataScienceOverview.html
11. https://www.lsst.ac.uk/technology/lasair
12. https://link.springer.com/chapter/10.1007/978-3-030-78307-5_4
13. https://howtofair.dk
14. https://github.com/larsga/Duke
15. https://www.nsf.gov/geo/updates/national-artificial-intelligence-research-resource-nairr
16. https://docs.olcf.ornl.gov/systems/summit_user_guide.html
17. https://db-blog.web.cern.ch/machine-learning-pipelines-high-energy-physics-using-apache-spark-bigdl
18. https://www.hira.or.kr/dummy.do?pgmid=HIRAJ010000003000
19. https://www.digital-science.com/blog/machine-first-fair-academic-data-for-the-ai-research-revolution
20. https://time.com/6589134/nairr-ai-resource-access
21. https://www.techradar.com/pro/hpc-and-ai-converging-infrastructures

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore AI enablement, pharma and biotech software development, data engineering and BI, and Veeva services. Contact IntuitionLabs to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: https://intuitionlabs.ai

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.