A Data Lake Use Case for scientific research data management

Mark Greiner

Max-Planck Institute for Chemical Energy Conversion

2022-01-31

Presentation Outline

Problem Space

- Description of domain
- Problem statement

- Data Governance Architectures
- Chosen solution (why)
- Lessons learned (pitfalls)
- Future directions

Solution Space

1. Description of the Domain

Problem space

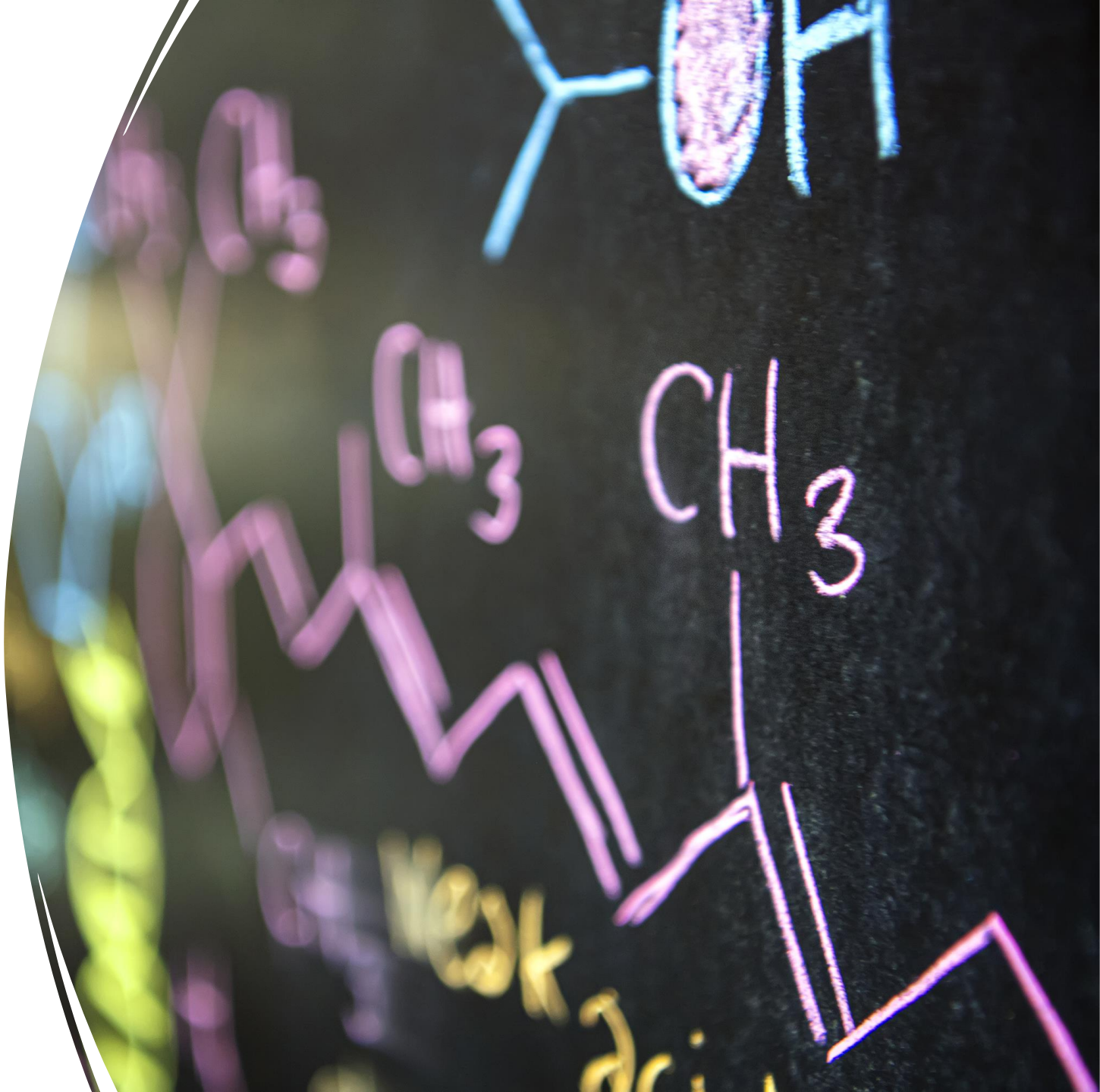
About MPI-CEC

Research Discipline

- Catalysis for chemical energy
- Water splitting, bio-catalysts, chemical production

Magnitude

- 218 Researchers
- 22 Research Groups
- 3 Departments



Diverse requirements

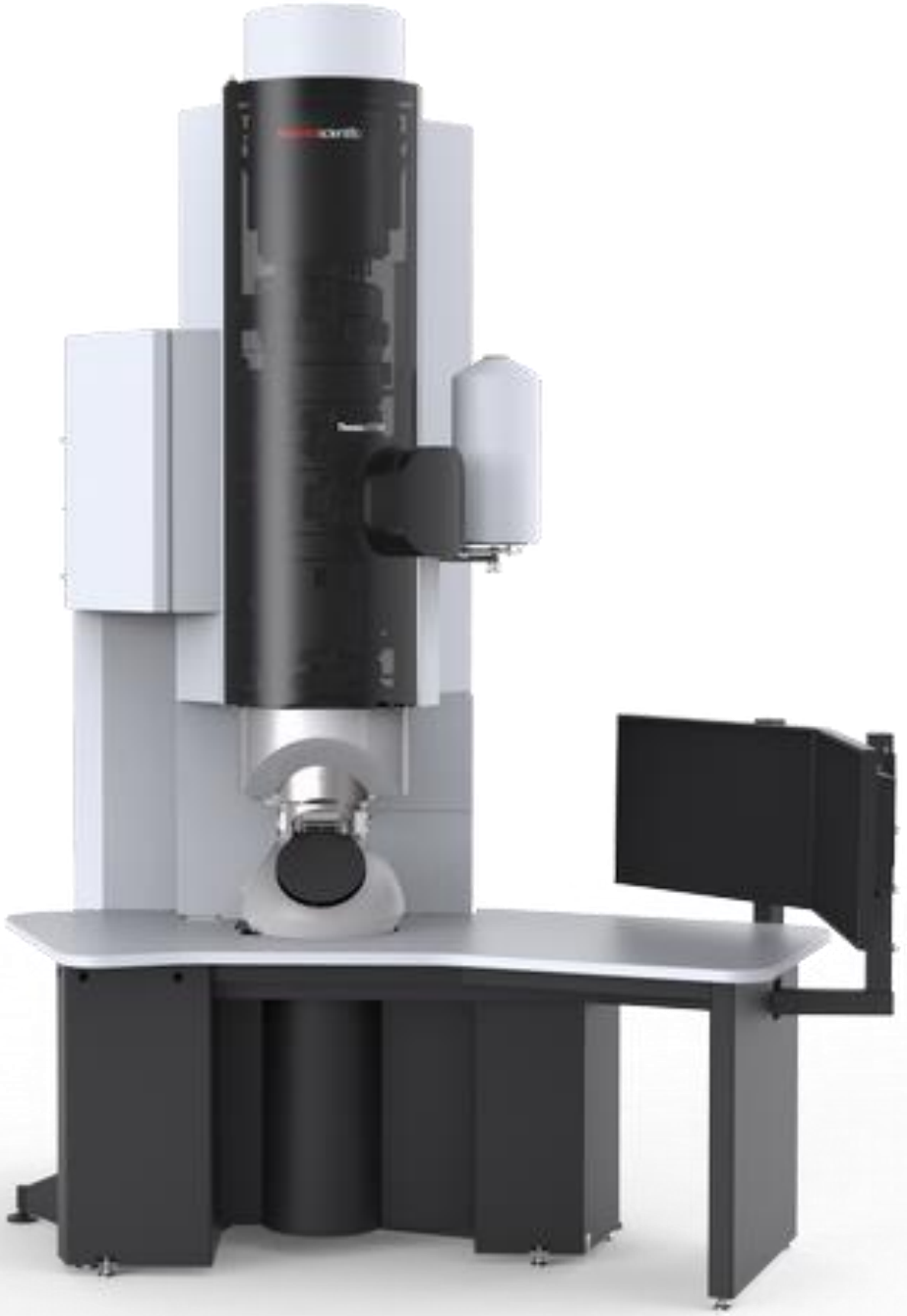
Synthesis Lab

- Focus on:
  - Creating new chemicals
- Workflow:
  - Plan, Synthesize, Characterize, Analyze results, Iterate, Test in some application
- Challenges:
  - Multi-disciplinary
  - Harmonize ELN with diverse data sources

Diverse requirements

Specialized Characterization

- Focus on:
  - in-depth analysis
- Workflow:
  - Job request, Plan with user, Sample management, Scheduling, Data analysis, Data to user
- Challenges:
  - Data analysis
  - Harmonize with user ELN



Diverse requirements

Self-service facilities

- Focus on:
  - routine measurements
- Can be performed with minimal training
- Workflow:
  - Schedule, Measure, Retrieve data, Analyze data,
- Challenge:
  - Harmonize users' ELN with instrument
  - Associate data with sample

Diverse requirements

Testing facilities

- Focus on:
  - Behavior in applications
  - Testing parameters
- Workflow:
  - Plan, Schedule, Measure, Retrieve data, Analyze data
- Challenges:
  - Analysis
  - Linking data-sample-conditions

Diverse requirements

Large facilities

- Focus on:
  - Characterizing
- Workflow:
  - Plan, Schedule, Measure, Retrieve data, Analyze data
- Major challenges:
  - Data sizes
  - Integrating with home ELN

Roles and skills

Student (Master/PhD)

Skills

- Conducts experiments
- Documents results

IT interactions

- Measure things
- Interact analysis software

Post-Doc

Skills

- Designing experiments
- Analysis workflows
- Documents results

IT Interactions

- Interact measurement software
- Interact analysis software
- Supervise students
- Review results

Principal investigator

Skills

- Provides research questions
- Supervises research of Students and Post-Docs
- Administrative Tasks

IT Interactions

- Interacts with management software

Roles and skills

Technical Staff

Skills

- Maintenance Laboratory equipment
- Keep services running
- Administer stock
- Supervise experiments

IT Interaction

- Interact with monitoring software

Group Leaders

Skills

- Project management

IT Interaction

- Interact with management software

Directors

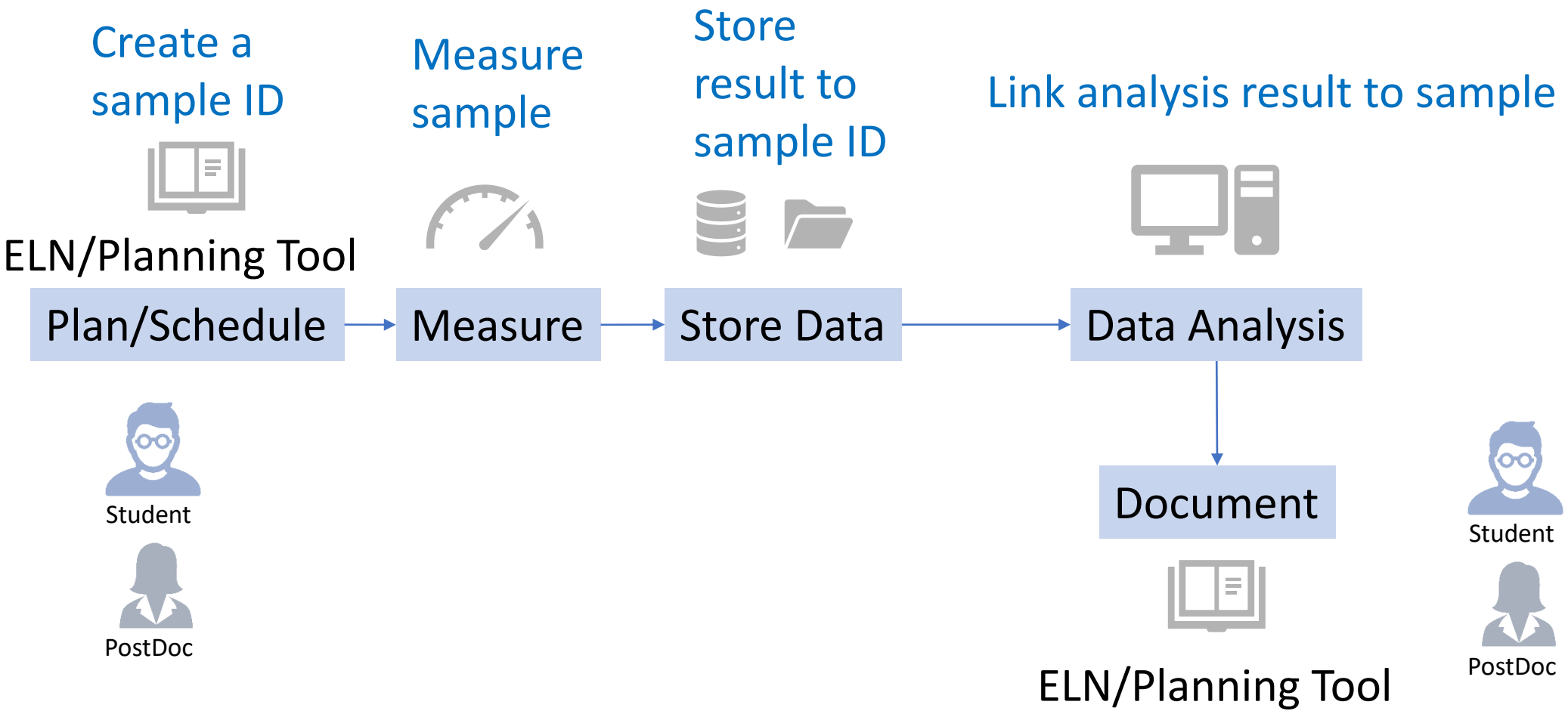
Skills

- Management

IT Interaction

- View dashboards and reports

Use Case: Self-service testing facitlity



Modified from 2021-05-31_NFDI4Cat Consortium meeting - ELN task force.pptx

2. Problem Statements

Problem Space

Problem Statement

- Data assets are not organized.
- It is distributed across many locations, with no contextual metadata.
- Thus, searching and organizing tools cannot be used to utilize the data.
- Knowledge cannot be automatically extracted from it.

Problem Statements

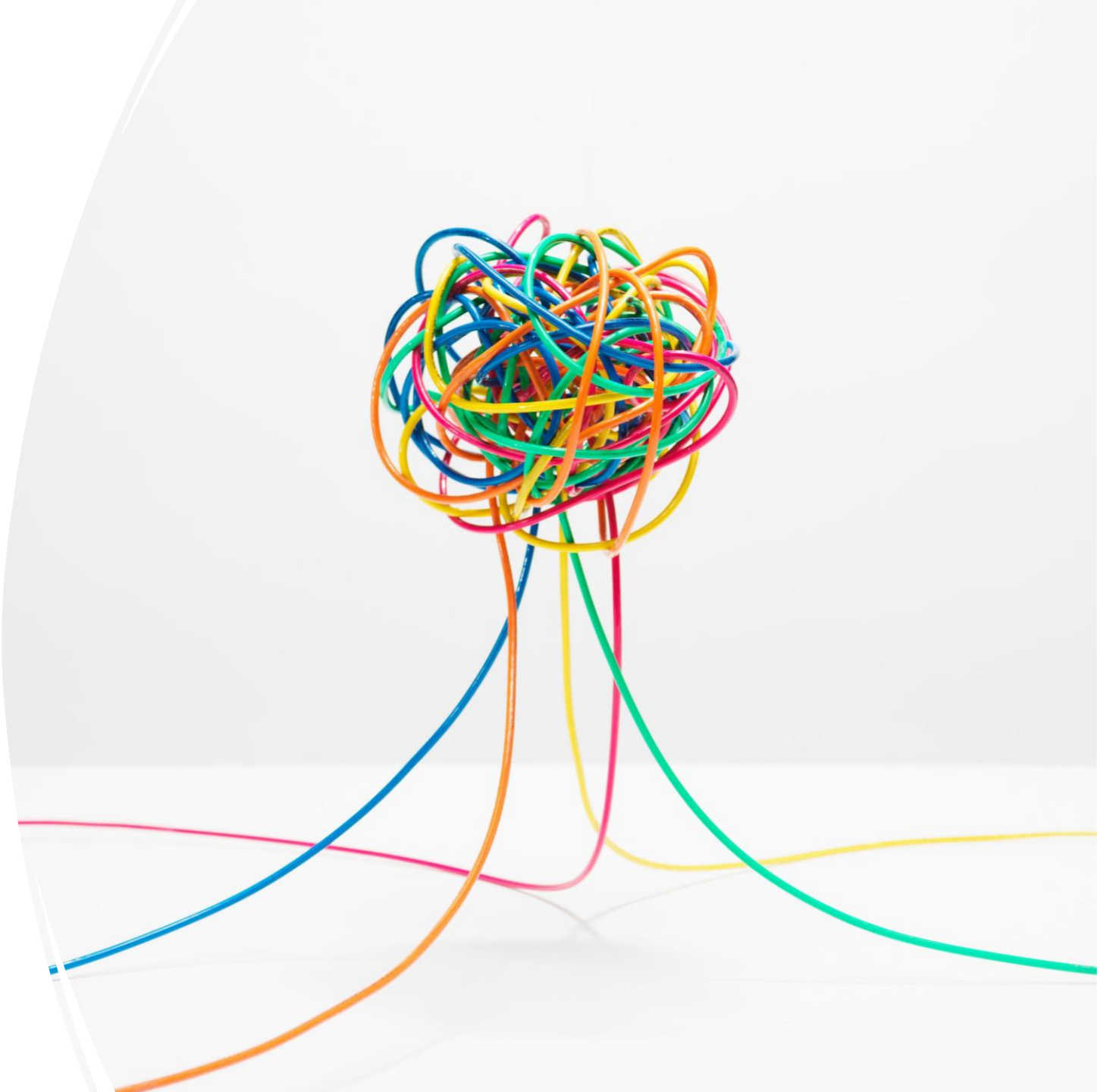
- Researchers spend too much time on repetitive manual work, related to organizing, searching and processing data.
- Takes away from value-added work, increases errors, leads to re-work.

Problem Statements

- Users do not have easy access to all their data assets.

Problem Statement

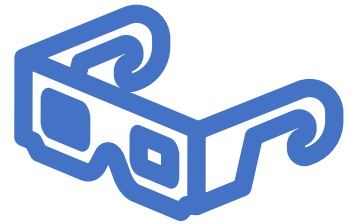
- It is difficult, sometimes impossible, to trace back the origin of a research result.
- Leads to excessive time spent searching when report revisions are needed.
- Decreases knowledge retention.



Problem Statement

- The structure of the organization's data assets are not suitable for large-scale analysis algorithms.
- Unable to utilize modern algorithms for meta-analysis.

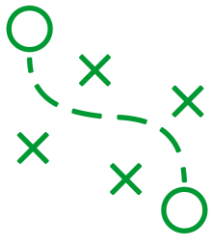
3. Data Governance Architectures



Solution Space

What is a data strategy?

- A data strategy can be considered as an approach that enables to derive new knowledge from data
- A comprehensive data strategy is the basis for the successful implementation of data related projects



A Data Strategy describes the …

- … organizational structure for the successful use of data with relevant processes for dealing with data
- … required skills & roles
- … technology and tools

https://www.gartner.com/en/documents/3975595/creating-a-data-strategy

Vision

Users **maintain their existing workflows**, while their **generated data is automatically digitized and categorized** for them, and is, subsequently, **available and easy to find** at a future time.

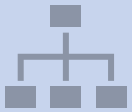
Goals of a Data Strategy

Remove Information Silos

Keep knowledge of research

Make reasearch more accessible

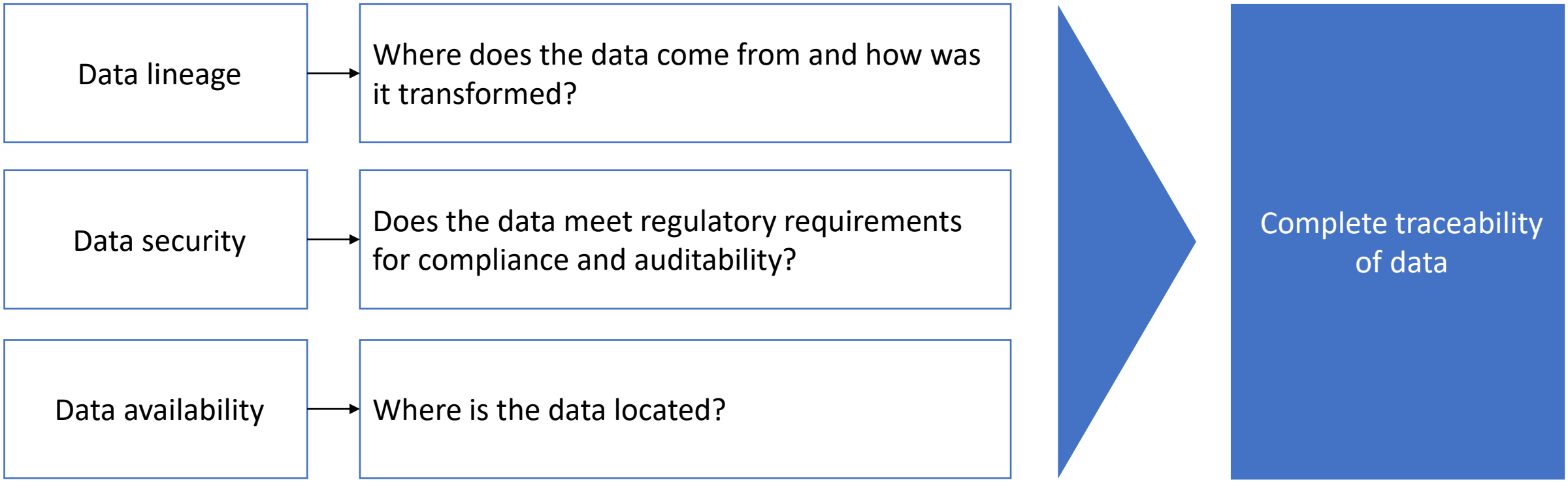
Enable meta research



Improve sustainability

https://datagovernance.com/goals-and-principles-for-data-governance/

Data Governance Concepts



https://datagovernance.com/goals-and-principles-for-data-governance/

Roles and skills in a business context

Data Engineer

Transform and harmonise data

Data Architect

Provide data processing concepts

Data Scientist

Analyse and model data

Data Artist

Visualize data

Data Custodian

Data storage and security

Data Steward

Steering and household data

Data Security Admin.

Data security concepts

Domain Expert

Domain knowledge

Data Evangelist

Explores data potential

Data Governance Roles

Student (Master/PhD)

DG Roles

- Data engineer
- Data scientist

Post-Doc

DG Roles

- Data scientist
- Data architect
- Data engineer
- Domain expert

Principal investigator

DG Roles

- Data steward
- Data evangelist
- Domain expert

Data Governance Roles

Technical Staff

DG Roles

- Data steward

Group Leaders

DG Roles

- Data steward
- Data evangelist
- Domain expert

Metadata-Management is the key to a successful data governance strategy

Data		Metadata		Taxonomy		Data Governance
Files		Technical	• Databasename • Tablename • Datatype	Directory Layout		Auditable & Compliance
		Functional	• Denomination • Classification • Tags			Security Standards
		Operational	• Who? (Access) • What & When? (Logs) • Where? (IP)			Availability
		Data Lineage	• Data Origin and Transformation			

Hortonworks (2016): **BUILDING GOVERNANCE INTO BIG DATA**

What is metadata?

- Data about data 
- Metadata can describe a single piece of data, a dataset or collection.
- Standard types of metadata:
  - *Descriptive*: information about **who** created a resource, **what** it is **about** and **what** it **includes** (e.g. title, author, subjects, keywords etc.)
  - *Structural*: information about the **way** data elements are **organized**, their **relationship** and the **structure** they exist in (e.g. ER-model)
  - Administrative: information about the origin of resources, their type and access rights
    (e.g. file type, date of creation etc.)

Table with **4** books, created by Joe Dow

Descriptive METADATA

ID	Title	Author	Year	Cover	Edition	Price
1	Emma	Austen	1815	Paperback	20th	€ 4,30
2	Dracula	Stoker	1897	Hardback	15th	€ 10,00
3	Ivanhoe	Scott	1820	Hardback	8th	€ 20,00
4	Kidnapped	Stevenson	1886	Paperback	11th	€ 3,50

Origin of resources: Book store
Access rights: read only - everyone; write and read – Joe Dow
Created on 5. January 2019

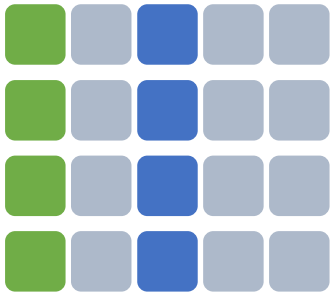
Administrative METADATA

Unsorted table;
related to sales data;
key is 'ID'

Structural METADATA

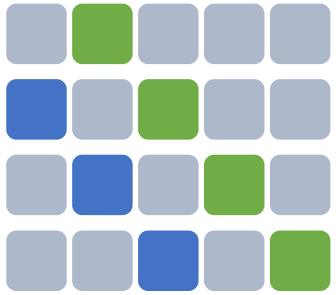
What is data structure?

Data structure is the particular way of ORGANIZING & STORING digital information.



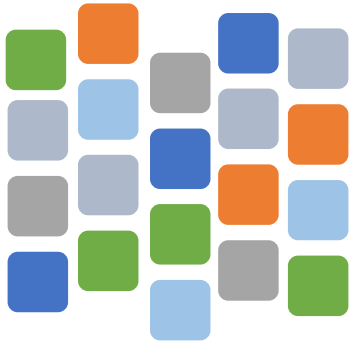
Structured data:

Information with a specific and high degree of organization (tabular form)



Semi-structured data:

Information with some degree of organization



Unstructured data:

Information with no pre-defined organizational structure

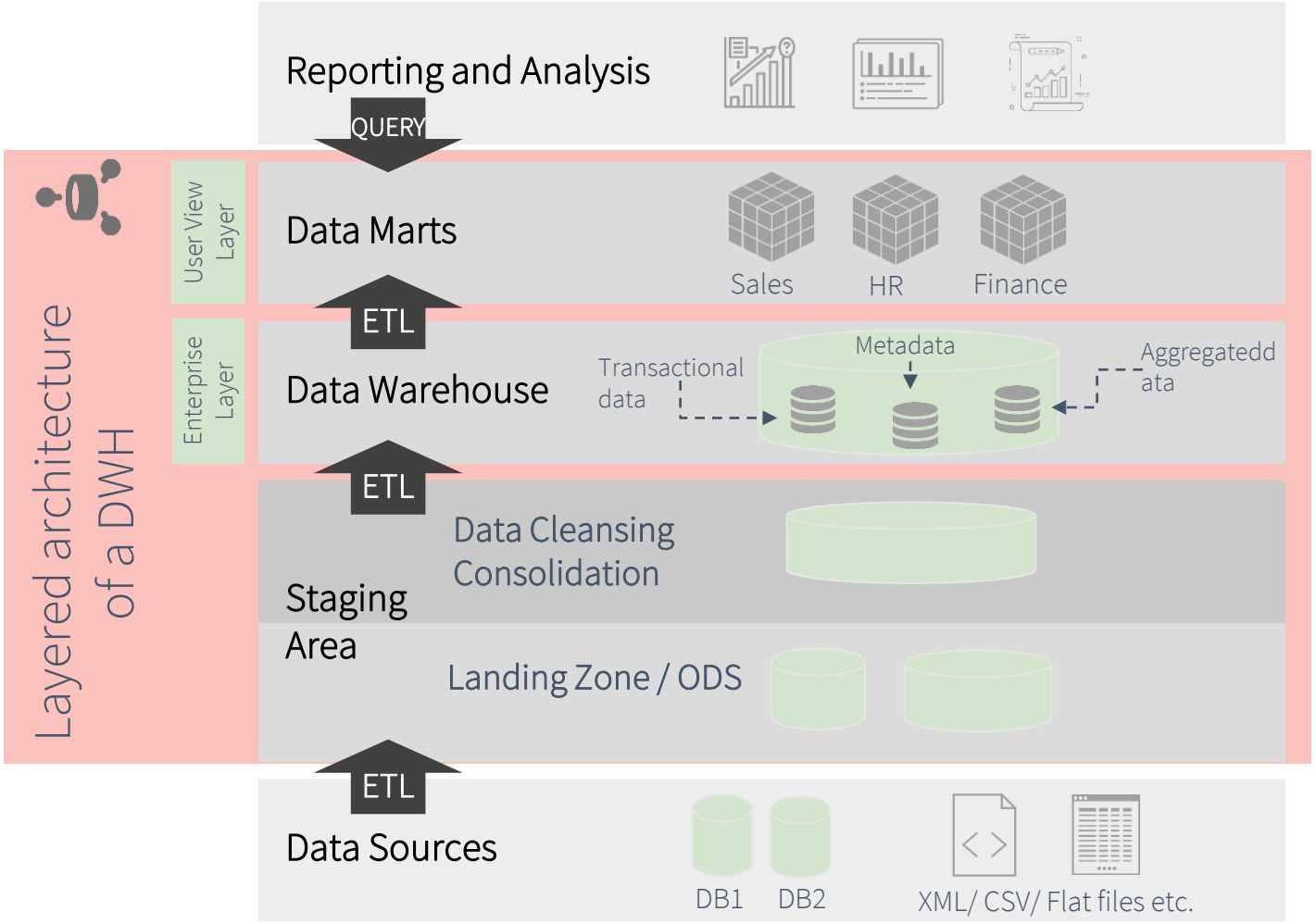
How do we store data?

	Databases		File System	Object Storage
Definition	**Collection of stored data** organized in a specific way and determined by the data model underlying the database (e.g. ER-diagram) 		• Store data in a space with a pre-defined scheme (e.g. a file hierarchy) 	• Store data as objects in a space with no pre-defined scheme • Every file has a unique identifier, so it can be found (e.g. URL) 
	Relational DB	**Non-relational DB**		
	• Based on relational data model (data is organized in tables) • Structured Query Language (SQL) for querying the database	• Based on any data model other than the relational model • Examples include key-value stores, document stores and graph databases		
Example	• MS SQL Server, MySQL, Oracle	• MongoDB, Cassandra, Neo4J	• NTFS (Windows), Hadoop Distributed File System, ext3	• AWS S3 buckets • Azure Blob Storage

Why is it important to use the right data storage technology?

	Databases		(Distributed) File System	Object Storage
Use Case	Relational DB	Non-relational DB	Store data in various forms (structured, unstructured etc.) Large volumes of data	
	• Store medium-sized data (difficult to scale) • Strict data consistency has to be ensured • Complex queries for analysis of data	• Data model together with rows		
Example	• Storing data on customers and their bank accounts (e.g. balance), where strict consistency is absolutely required	• Storing and retrieving chatbot conversations from a website, where low latency for chatbot to respond is ensured	• Storing social media content, e.g. pictures, alongside user-data and sensor data (e.g. GPS)	

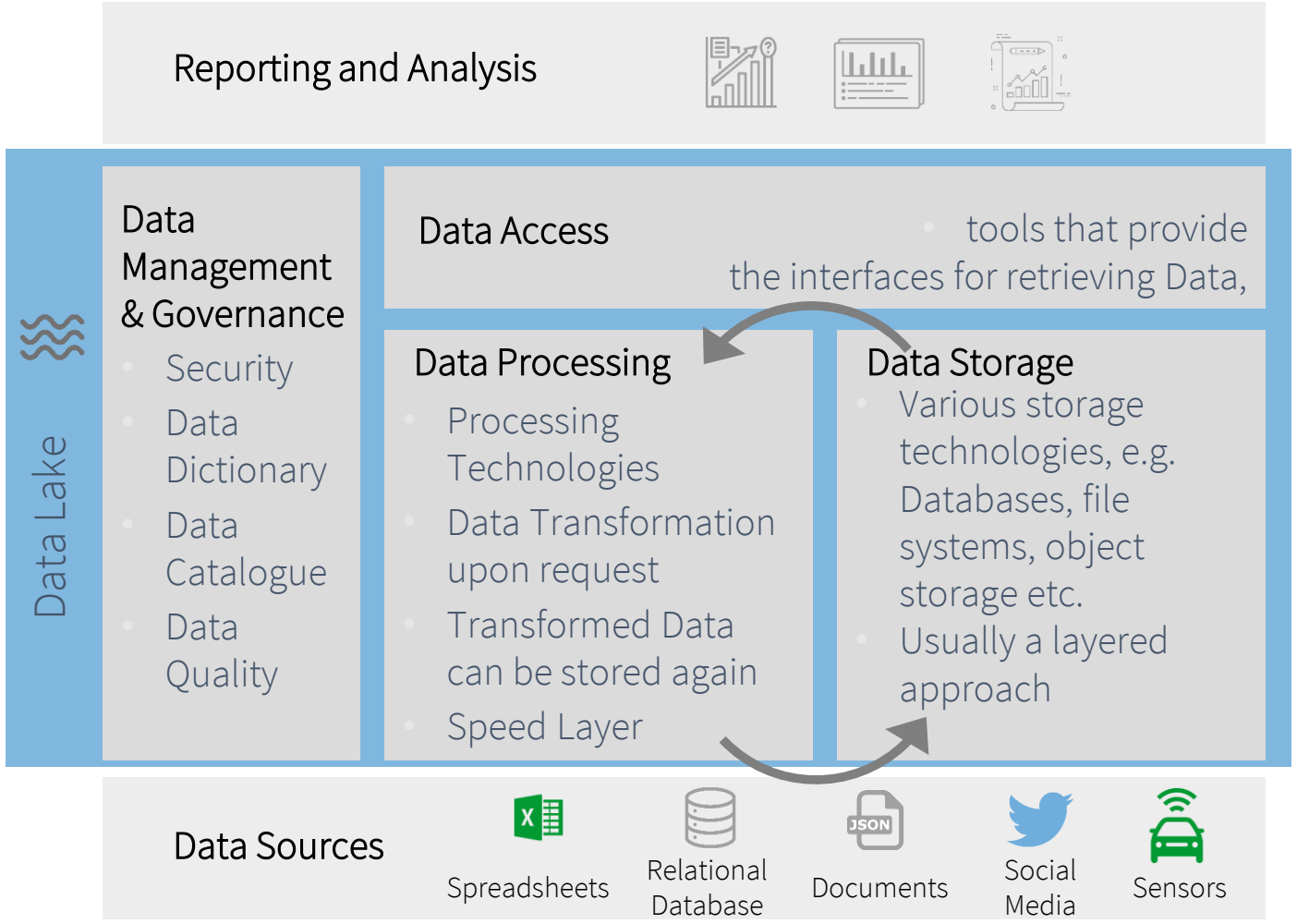
What is a Data Warehouse?



"A Data Warehouse is a subject oriented, integrated, non-volatile and time-variant collection of data in support of management decisions"*

*Source: Immon, WH. Building the Data Warehouse

What is a Data Lake?



A Data Lake is a modular system of data storage and processing technologies. Like a DWH it is a logical concept rather than a tangible entity.

Benefits of a Data Lake

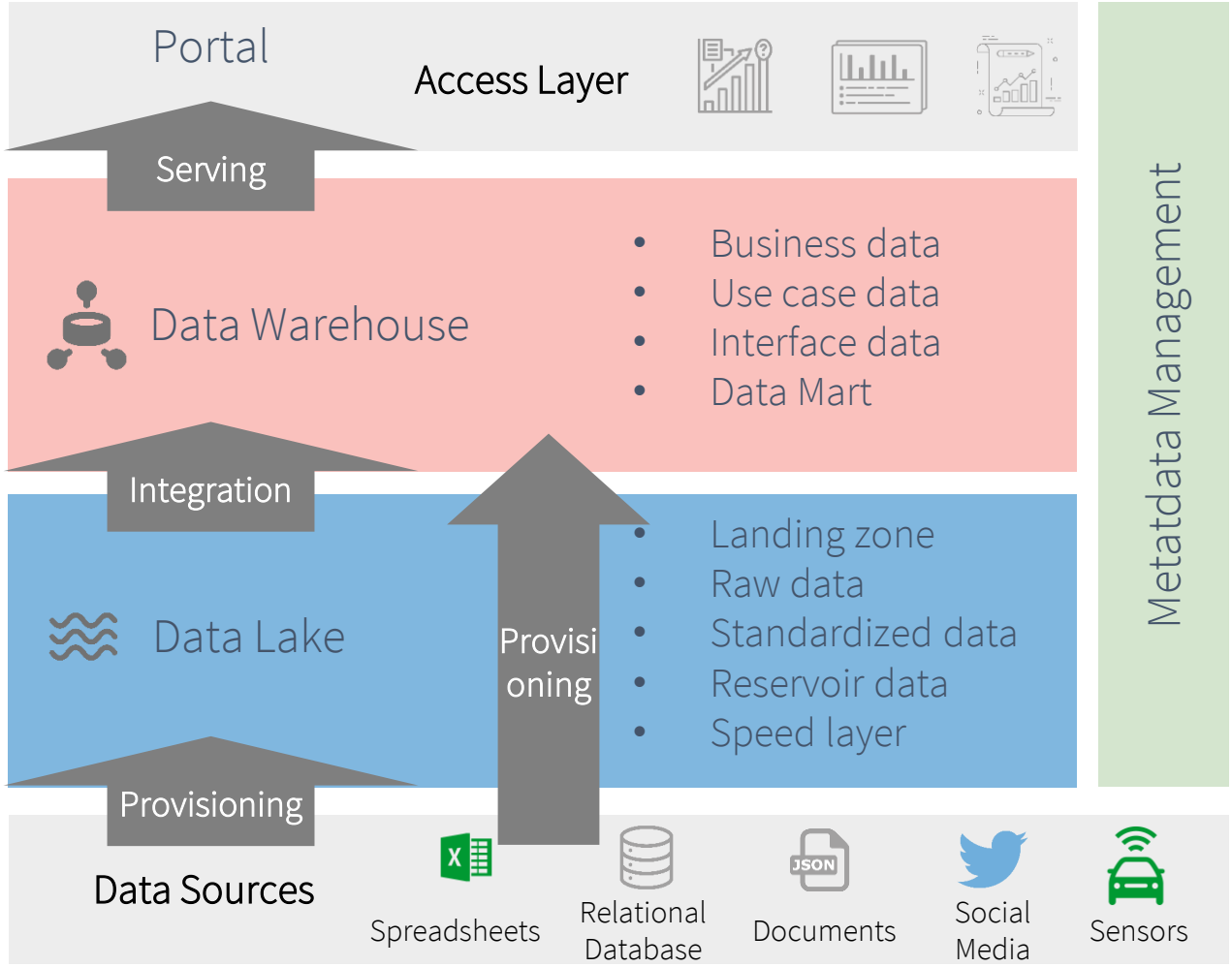
- stores all kinds of data, e.g.:
  - Structured Tables
  - Text Documents
  - Pictures
- Central entry point for data access

What are the differences between a DWH and a Data Lake?

Data Lakes are not necessarily always the better architecture, each architecture has its pros and cons

	Data Warehouse	Data Lake
Data Sources 	• Structured data • Traditionally medium sized amounts of data (although high scalability available by now) −	• All kinds of data, both structured and unstructured • Large volumes of data +
Use Cases	• Reports (dashboards, visualizations etc.) • Data Analysis on structured data	• Data Analysis on large data and unstructured Data, such as text or image
Agility & Effort	• Data must often be transformed structured, and cleaned before storing • Protection of doing the same thing • Large initial effort − +	• Storage of raw data (danger of data swamp) and preparation steps often multiple times • Transformation for analysis usually expensive • Easier for prototypes − +
Set-up	• Mature architecture + • Expertise is more available • Easier maintenance and config.	• Due to modularity more complex configuration • Up-to-date experts required −

How can both architectures be combined?

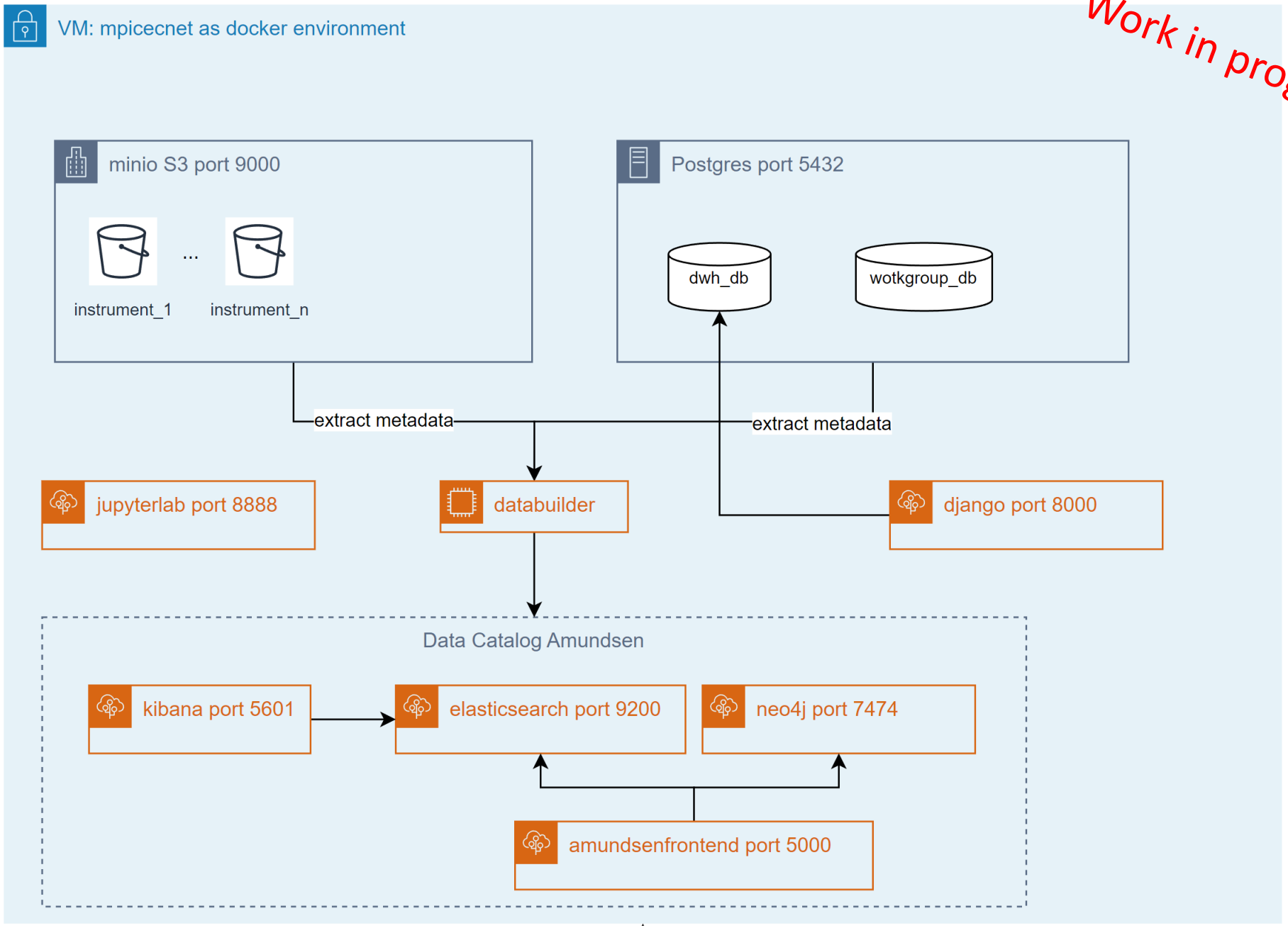


Advantage

- One platform for different requirements
- Best of both architectures

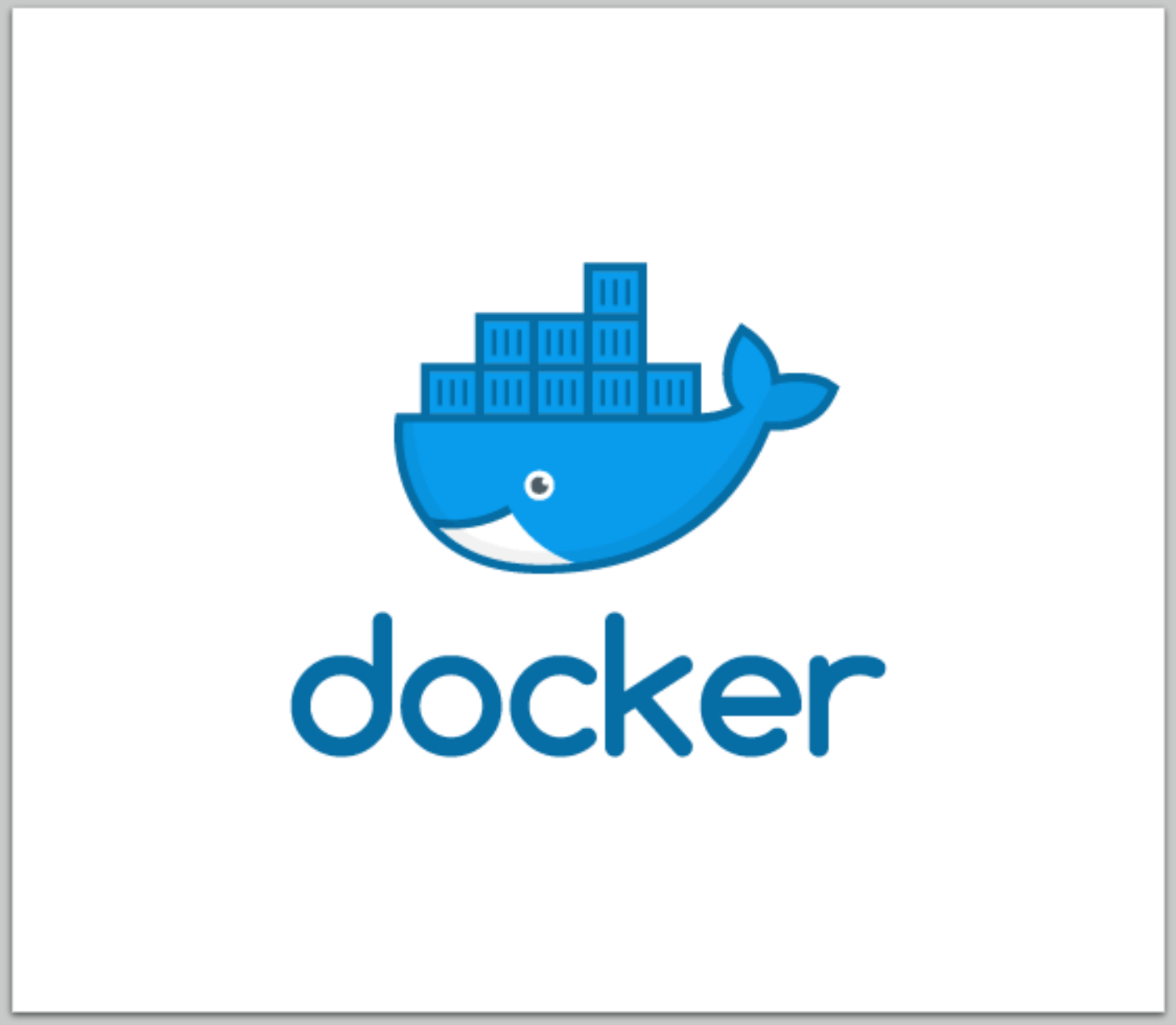
4. Chosen Solutions

Solution Space


VM: mpicecnet as docker environment
Work in progress
minio S3 port 9000
instrument_1
...
instrument_n
Postgres port 5432
dwh_db
wotkgroup_db
extract metadata
extract metadata
jupyterlab port 8888
databuilder
django port 8000
Data Catalog Amundsen
kibana port 5601
elasticsearch port 9200
neo4j port 7474
amundsenfrontend port 5000

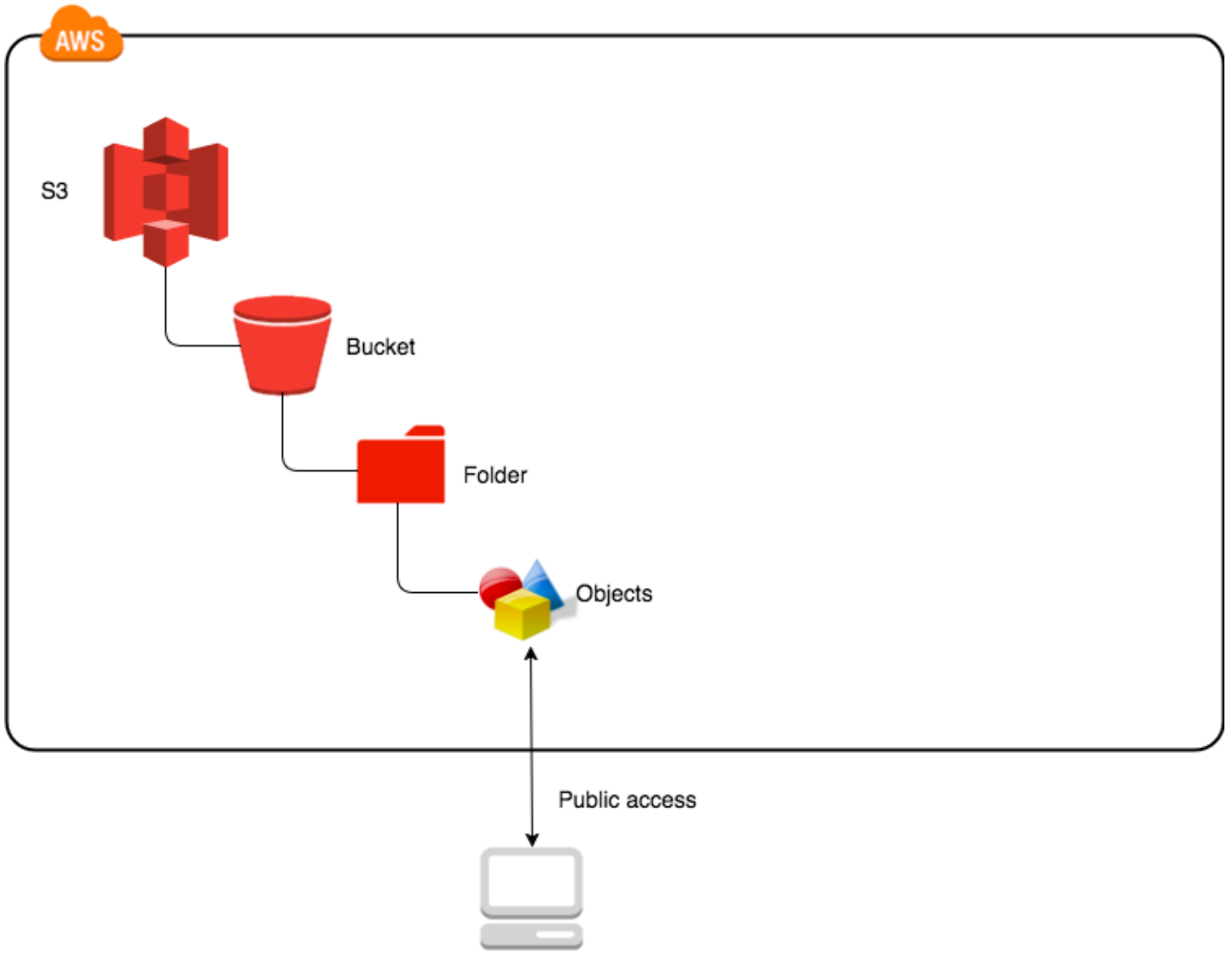
Docker

- Containers for running microservices
- Isolated environments
- Needs less resources than a VM



Minio

- Object storage
- Same as Amazon S3
- Can store any kind of data



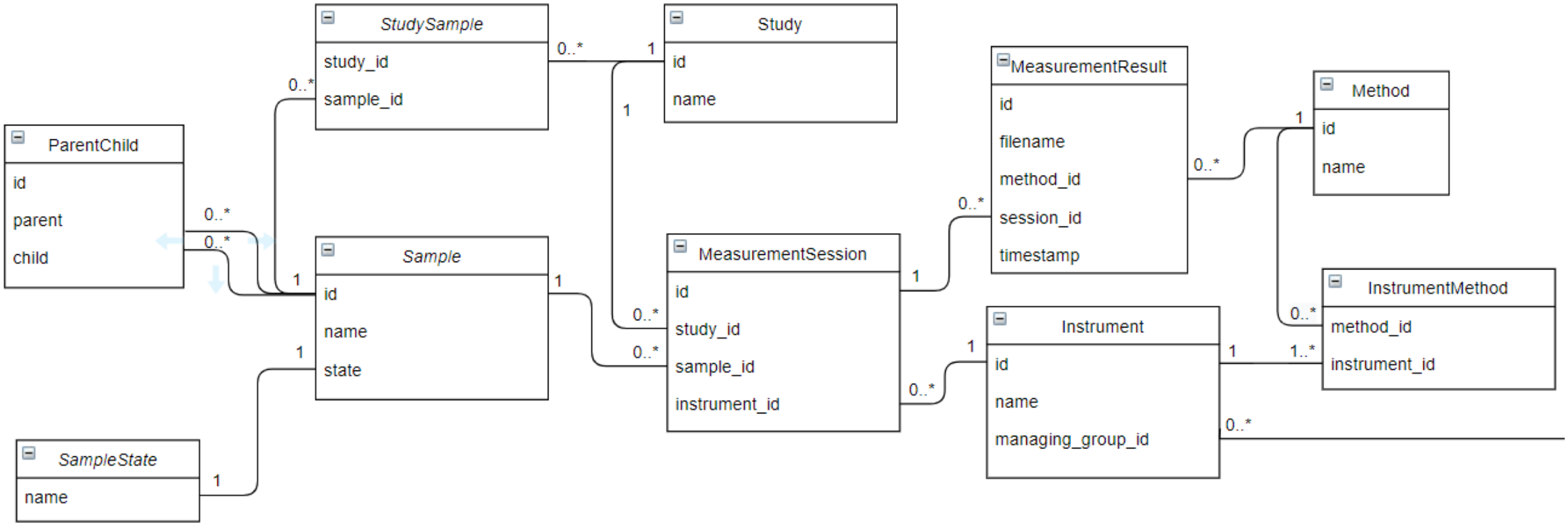
MINIO

- Buckets hold objects
  - Define access rights
- Objects have immutable metadata
- Client can access data over API

PostgreSQL

- Relational database
- Used for storing structured data
- Transactional data and data warehouse
- Strict and robust data models

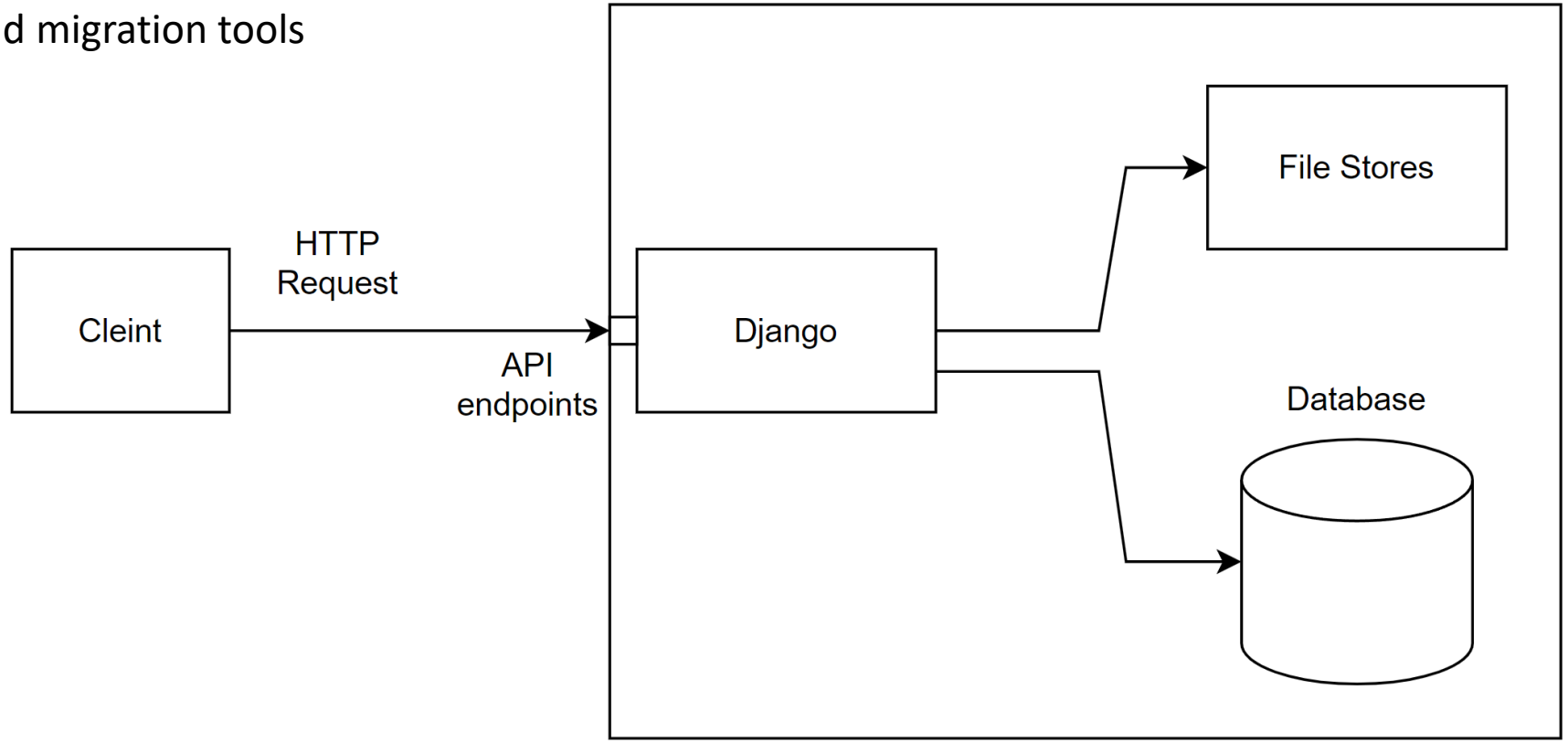
- Used for storing:
  - Transactional Data
  - Data Warehouse tables



Django

- Back-end Web application framework written in python
- Robust and scalable
- Used for providing REST API for accessing Relational Database

- API endpoints provide client with ability to CRUD database entries
- Will be used to connect front-end apps
- Access to File stores
- Django provides Authentication and Authorization to access resources
- Django provides convenient database schema history and migration tools



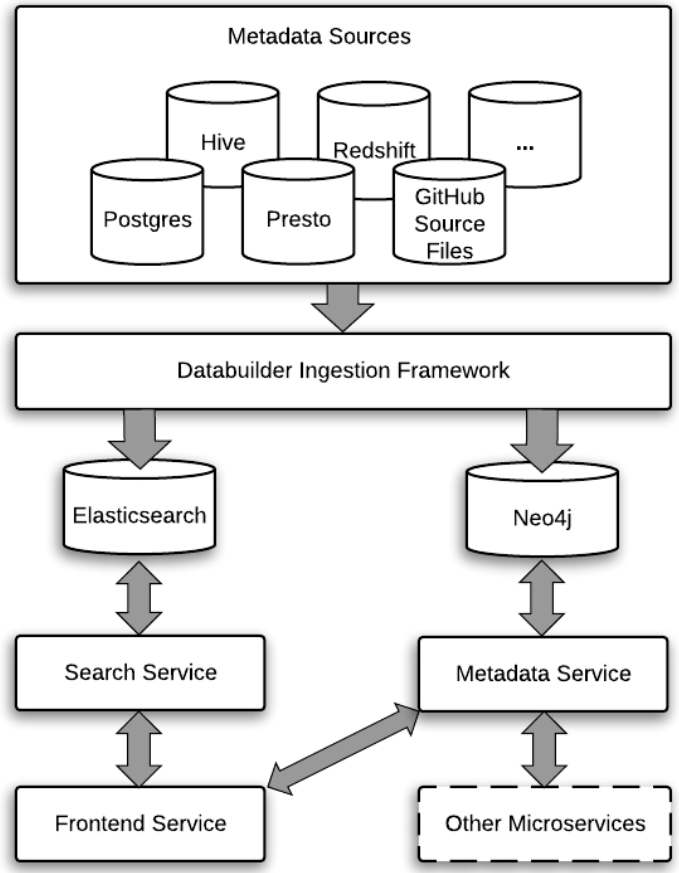
Amundsen

- Data catalog
- PageRank-inspired search algorithm
- Provides REST API for search engine

How does the data catalog work?

- Amundsen collects Metadata via the data builder ingestion framework
- Metadata and lineage data are stored in Neo4j and elasticsearch
- Metadata is made accessible via search interface
- Metadata can be made searchable for all users, whereas the content remains visible only with sufficient access permissions

Amundsen data catalog architecture



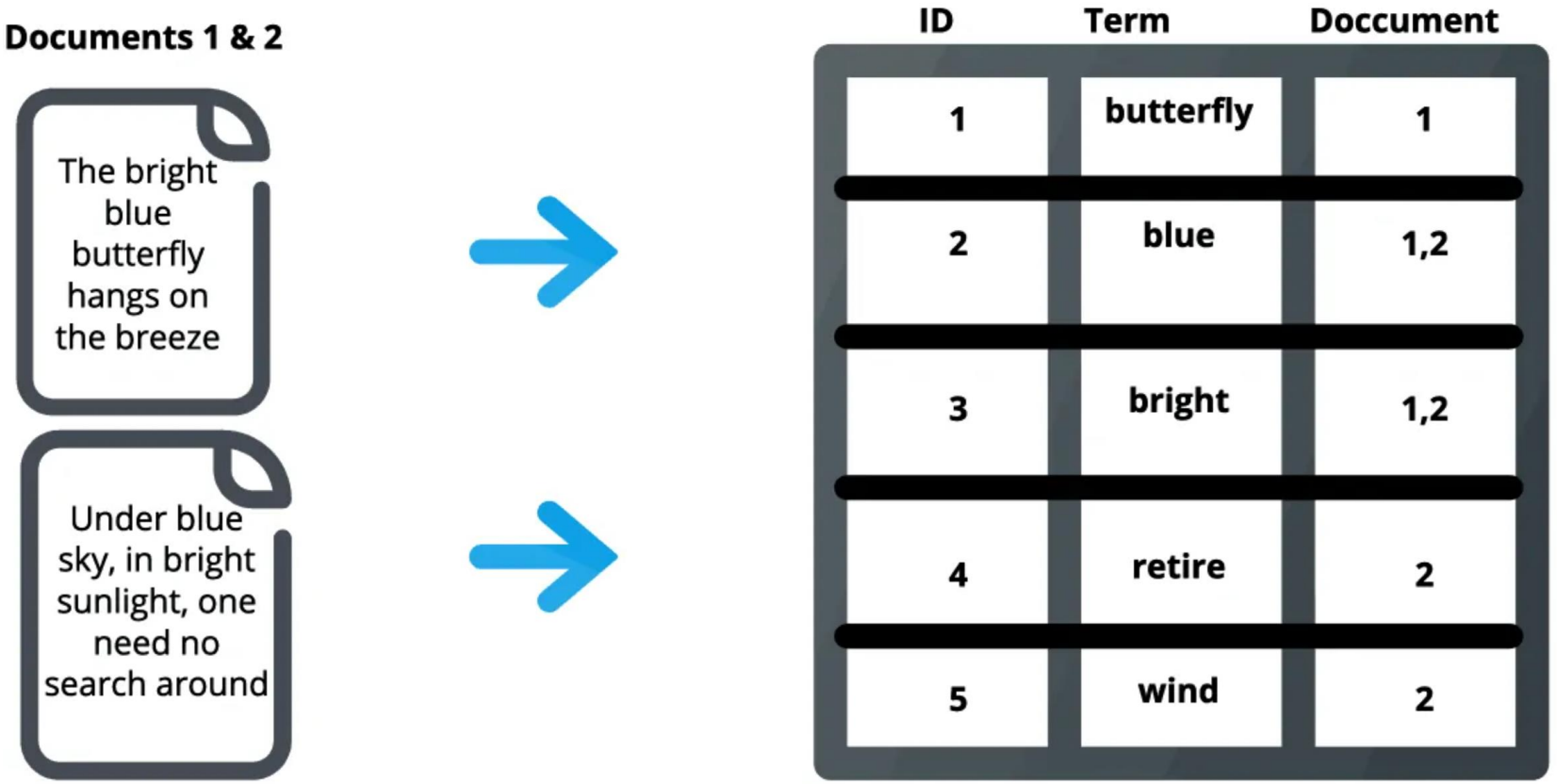
https://www.amundsen.io/amundsen/architecture/

Elasticsearch

- Search engine
- Service behind Amundsen's search library
- NoSQL data store
- Stores data as documents (like JSON)

Inverted Index

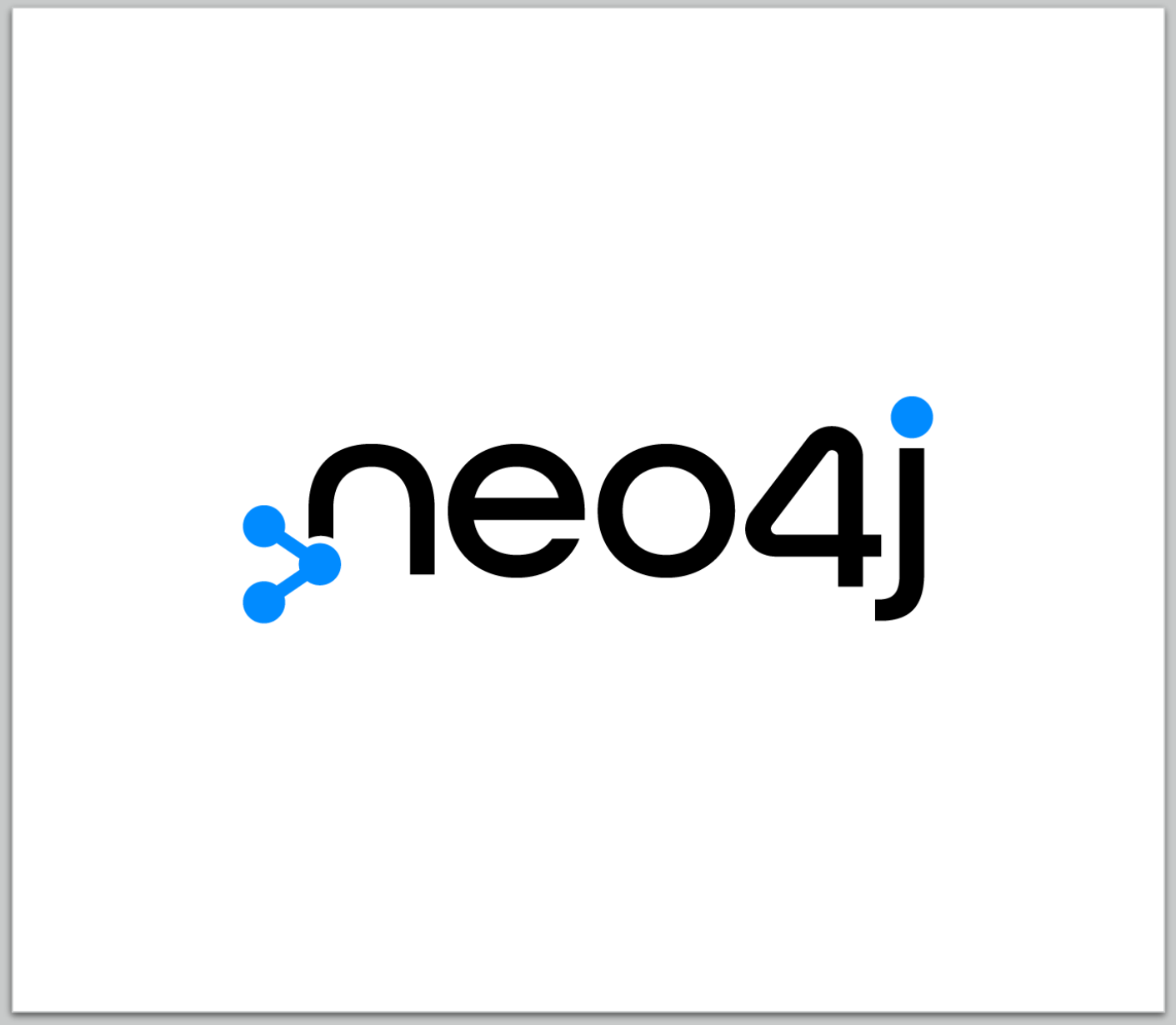
elasticsearch



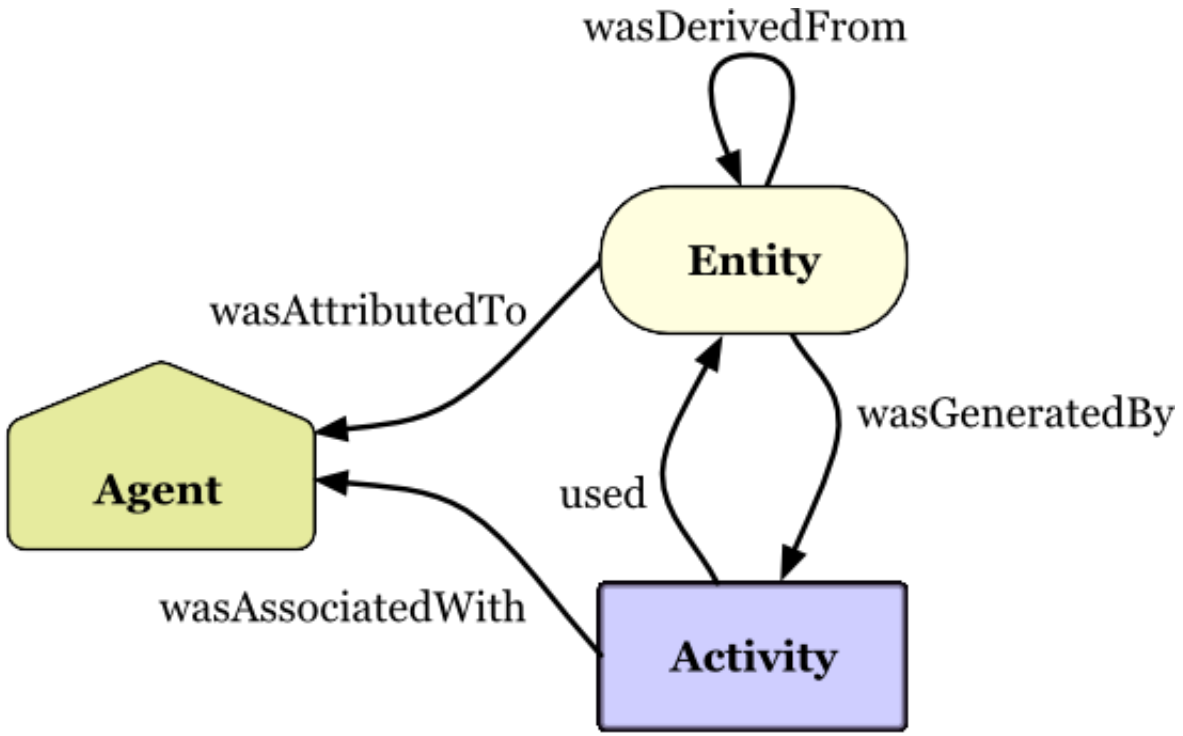
https://www.knowi.com/blog/what-is-elastic-search/

Neo4j

- Graph database behind Amunden's metadata service
- Stores 'Triples' (subject, predicate, object)
- Great for traversing relationships



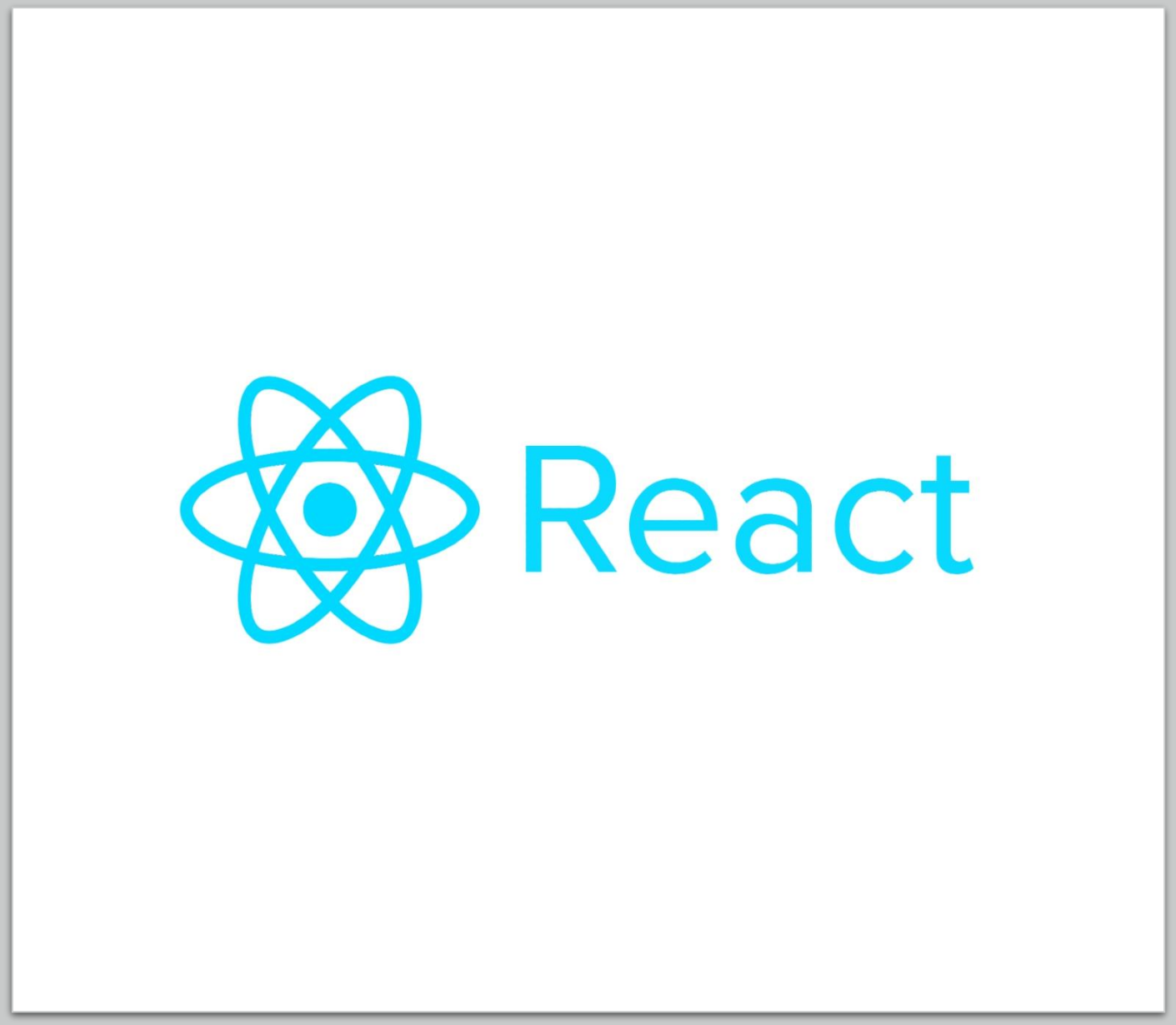
Data Provenance



https://www.w3.org/TR/prov-primer/

ReactJS

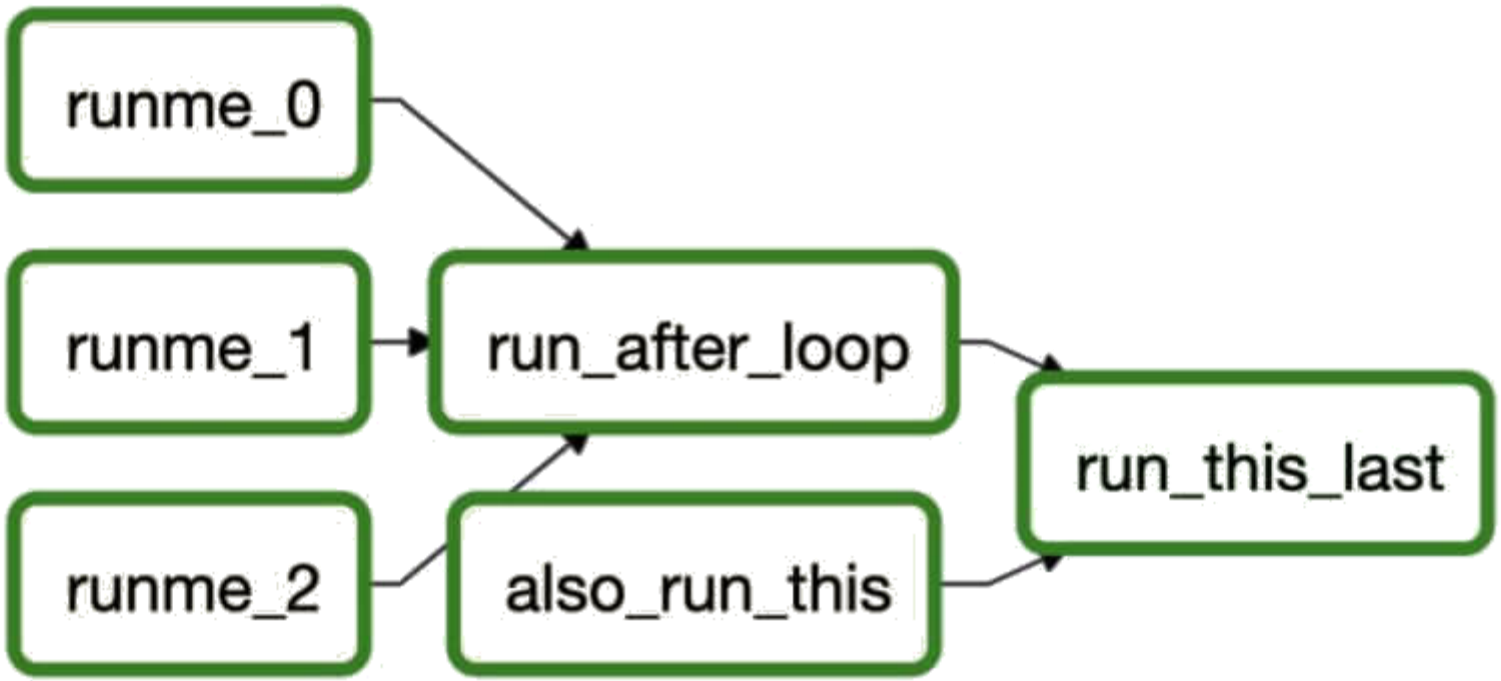
- JavaScript framework ideal for building single-page apps
- Will be used for building front-end apps
- For user-interaction with Data Lake



Apache Airflow

- Orchestration tool
- Build for running complex cron jobs
- Will be used for ETL processes

Orchestration



https://www.qubole.com/tech-blog/apache-airflow-tutorial-dags-tasks-operators-sensors-hooks-xcom/

5. Lessons Learned and Pitfalls

Solution Space

Provide Value as Soon as Possible

- Data Governance is a new concept to the organization
- Stakeholders expect immediate benefit
- Initial effort needed for back-end development does not immediately show value
- Set goal of a minimum viable product and achieve it in short term

Technological Debt

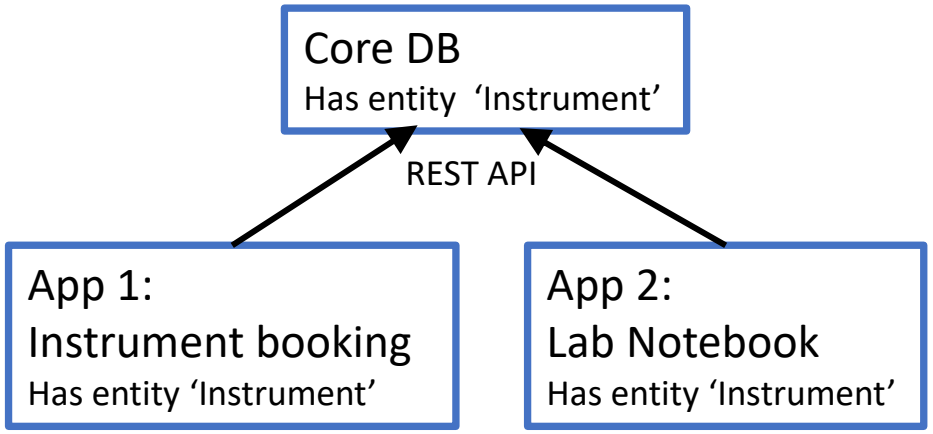
- Immediate needs at the cost of future needs
- Can result in substantial refactoring (technological debt)
  - E.g. CI-CD now or later?
- But are we really going to need it?
- Must be judicious about deciding when to incorporate components into the architecture

Preparing for the Long-term

- Product built by a team
- Teams change over time
- Keep good documentation
  - most maintainable when it is close to the code
- Automate documentation (e.g. Sphynx, swagger)
- Be diligent with writing unit tests and integration tests

‘Single Source of Truth’

- Find central domain concepts
- Put in core layer of the DWH



6. Future Directions

Solution Space

Planned Features

- Git Repo for analysis code
- Jupyter Hub
- Logging analysis pipelines to get data lineage
- Curated data sets
  - E.g. Database of spectroscopy data
- Publishing pipelines

7. Summary

Problem Space

- Diverse requirements
- Non-conventional use case
- Overlapping roles

Solution Space

- Common Data governance Philosophies
- Common Data Governance Architectures
- Chosen solutions
  - Docker, Object storage (S3), RDBMS (PostgreSQL), Data catalogue (Amundsen), Orchestration (Airflow)
- MVP target
  - Get unstructured data into Data Lake with some additional metadata
  - Provide data access point