SEDAR: A Semantic Data Reservoir for Integrating Heterogeneous Datasets and Machine Learning

by Sayed Hoseini and Christoph Quix

SEDAR is a comprehensive semantic data lake that includes support for data ingestion, storage, processing, analytics and governance. The key element of SEDAR is semantic metadata management, suitable for many use cases, e.g. provenance, versioning, lineage, dataset similarity or profiling. The generic ingestion interface can deal with any external data source ranging from files to databases and streams change and incorporates data capture with data versioning and automatic metadata extraction. Machine learning (ML) is integrated into the data lake as its artefacts (e.g., ML pipelines, notebooks, models) are stored in the data lake to allow a coherent development of data preparation and ML pipelines. As all these artefacts are related, their relationships and versions are maintained in the extended metadata repository.

Data lakes have been addressed intensively in research and practice in recent years as they provide the required data for data science and ML projects. Data lakes are scalable schema-less repositories to ingest raw data in its original format from heterogeneous data sources; thus, only a minimal effort is required for ingesting data in a data lake, which makes it an efficient tool to collect, store, link and transform datasets. While first implementations for data lakes aimed at processing “big data” efficiently using distributed, scalable systems like Hadoop, the need for proper management of metadata and data quality in data lakes has also been recognised [1].

Data lakes have many applications, one example is management of heterogeneous data in Industry 4.0 scenarios [2]. For efficient data management in this context, the FAIR principles (findable, accessible, interoperable, reusable) are important requirements. SEDAR is a comprehensive data lake that addresses these challenges within one uniform system. The key components of the system are the generic ingestion interface, the metadata repository (or data catalogue) and integrated machine learning (MLOps).

Our implementation of the data catalogue satisfies several technical requirements for comprehensive metadata management: 1. indexing, 2. data linkage, 3. data polymorphism / multiple zones (e.g. different formats or degree of processing), 4. data versioning, 5. usage tracking, 6. granularity levels, 7. similarity relationships, 8. semantic enrichment and 9. data quality features. Furthermore, the data catalogue must ensure ease-of-use for non-technical users and provide means to manage access to sensitive information, i.e. privacy-preserving measures. These requirements can only be met partially by existing data lake systems [3].

Conceptually, the architecture of SEDAR is divided into four functional areas: (a) ingestion, to continuously extract data from heterogeneous sources (e.g. files, databases, streams), (b) storage, to efficiently store the data in big data systems, (c) transformation, to have a scalable environment to preprocess, integrate and transform data and (d) interaction, to provide tools to the user to interact with the data.
The ingestion layer aims at generality in order to support a wide range of data formats and source systems. All configuration settings for data sources can be changed at run time; new systems can be integrated using an extensible ingestion interface. Metadata management is an important issue in the whole data lake; it is initiated by an extensive metadata extraction. Furthermore, data ingestion is not a one-time process; data sources are updated continuously and these updates need to be reflected in the data lake. Thus, versioning of datasets is supported, including change data capture between two versions of a dataset. By doing so, the system can incrementally process the updates instead of loading the dataset as a whole. On the other hand, the data versioning mechanism enables access to previous states of a dataset. The information about ingested datasets, their versions, updates, etc. are extracted automatically by integrated the Delta Lake API [L1] and stored in the data catalogue.

The storage layer provides efficient storage systems, based on big data systems, such as Hadoop, or NoSQL systems, such as MongoDB. According to the basic idea of data lakes, data is stored in its original format. Thus, we chose an appropriate storage mechanism for different data formats (e.g. JSON is stored in MongoDB). The data catalogue keeps track of all datasets and their storage locations.

Apache Spark provides uniform access to the storage systems and is used as the main component in the transformation layer. With its data access layer based on SparkSQL and the DataFrameAPI, Spark provides a very helpful tool for implementing data preprocessing and integration pipelines. We integrated Amazon’s Deequ [L2] library built on top of Apache Spark for automating the verification of data quality at scale.

While the functions of the ingestion, storage and transformation layers are very common for data lakes, SEDAR provides a semantic metadata management and integrates MLOps (i.e. operational support for ML applications) in the interaction layer. Datasets can be annotated with and queried through semantic terms (e.g. using ontology-based data access). To keep track of the various inputs and outputs of an ML application (e.g. data versions, code and tuning parameters, reproducing results and production deployment), MLOps is proposed to streamline the ML lifecycle. SEDAR addresses the key challenges in ML applications: experimentation, reproducibility and model deployment.

Technically, SEDAR uses a modular and flexible architecture based on microservices that can be extended easily with additional functionality, making the system suitable as a workbench for further research in data lakes. SEDAR is a web application with a modern, responsive user interface and several data management technologies in the backend exposed by a REST API.

Links:
[L1] https://github.com/delta-io/delta
[L2] https://github.com/awslabs/deequ

References:


**Please contact:**

Sayed Hoseini, Hochschule Niederrhein University of Applied Sciences, Germany

[mailto:sayed.hoseini@hs-niederrhein.de](mailto:sayed.hoseini@hs-niederrhein.de)
Figures:

Figure 1 The SEDAR system architecture.