Exploring Architectural Solutions for Implementing the FAIR Principles in Big Data Environments

João P. C. Castro[1,2], **Cristina D. Aguiar**[1]

[1]Department of Computer Science – University of São Paulo – Brazil

[2]Information Technology Board – Federal University of Minas Gerais – Brazil

jpcarvalhocastro@ufmg.br, cdac@icmc.usp.br

Level: Doctorate (Computer Science and Computational Mathematics)
Admission: 08/2020 – **Qualification:** 08/2023 – **Defense:** 08/2025
Completed activities: Mandatory credits; bibliographic review; problem statement; proposal, implementation, and validation of architectural solutions; qualification writing
Future activities: Qualification defense; architectures extension; creation of pipelines, guidelines, algorithms, and artificial datasets; thesis writing and defense
Publications: ICEIS 2022 [Castro et al. 2022a], ADBIS 2022 [Castro et al. 2022b]

Abstract. *The concept of Open Science has emerged as a major enabler for scientific collaboration. To develop repositories adhering to this concept, the FAIR Principles have been proposed. However, fulfilling these principles can be challenging when dealing with a significant volume, variety, and velocity of data and metadata. A suitable solution is to develop a Software Reference Architecture (SRA) that considers the characteristics of big data environments and the FAIR Principles. Despite the importance of this solution for Open Science, existing literature lacks a big data SRA that achieves full FAIR compliance. In our research, we address this gap by proposing architectural solutions for the implementation of FAIR-compliant big data sharing repositories. We validate these solutions through case studies and performance evaluations. Future contributions include developing algorithms to instantiate the proposed architectures and creating FAIR-compliant artificial datasets to assist in further validations.*

Resumo. *O conceito de Ciência Aberta surgiu como um facilitador para a colaboração científica. Neste contexto, os Princípios FAIR foram propostos para desenvolver repositórios de dados. Porém, satisfazer esses princípios pode ser desafiador devido ao grande volume de dados e metadados científicos em diferentes formatos coletados e disponibilizados em alta velocidade. Uma possível solução é desenvolver uma Arquitetura de Referência de Software (SRA) que leve em consideração as características dos ambientes de big data e os Princípios FAIR. Apesar da importância dessa solução para a Ciência Aberta, a literatura existente carece de uma SRA para ambientes de big data que alcance plena conformidade com os Princípios FAIR. A pesquisa atual preenche esta lacuna ao propor duas arquiteturas FAIR para ambientes de big data, as validando com estudos de caso e avaliações de desempenho. Contribuições futuras incluem o desenvolvimento de algoritmos para instanciar as arquiteturas propostas e a criação de conjuntos de dados artificiais em conformidade com os Princípios FAIR para auxiliar em demais validações.*

1. Introduction

The rise of cloud computing and parallel and distributed data processing has led to an unprecedented collection, storage, and sharing of data. Open Science aims to harness this opportunity by making research data openly available, promoting collaboration and maximizing the impact of scientific activities [Medeiros et al. 2020]. To ensure standardized scientific data sharing, the FAIR Principles have been introduced [Wilkinson et al. 2016], emphasizing the importance of the Findability, Accessibility, Interoperability, and Reusability of digital datasets.

However, there is a gap between these principles and their implementation when dealing with a high volume, variety, and velocity of scientific data and metadata [Chen et al. 2014]. To bridge this gap, FAIR-compliant Software Reference Architectures (SRAs) become essential [Nakagawa et al. 2011]. These architectures serve as blueprints that guide data engineers in implementing big data sharing repositories that adhere to the FAIR Principles, utilizing components like data warehouses [Kimball and Ross 2011] and data lakes [Sawadogo and Darmont 2021] to efficiently store and retrieve data and metadata for analysis and decision-making.

Despite the significant importance behind adopting a big data SRA to support Open Science, existing solutions in the literature have certain limitations. We divide these solutions in three groups, as outlined in Section 2 and summarized as follows. The first group of studies propose SRAs for generic big data systems without considering the inherent characteristics of the FAIR Principles. The second group consists of implementations of the FAIR Principles in context-specific repositories, not providing a sufficiently generic architecture to qualify as an SRA. The third group encompasses big data SRAs developed to fulfill the FAIR Principles, but that are unable to achieve full compliance. These limitations motivate our research.

In our work, we propose two FAIR-compliant SRAs to handle big scientific data, namely BigFAIR [Castro et al. 2022a] and CloudFAIR [Castro et al. 2022b]. Both SRAs address the aforementioned limitations, with BigFAIR focusing on data ownership and flexibility by using two distinct infrastructures, and CloudFAIR focusing on performance and simplification for data providers by using a single cloud infrastructure. We also propose a generic metadata warehouse model employed in both architectures to ensure metadata persistence. A case study and a performance evaluation with real-world datasets are also conducted for validation. Moreover, we plan on achieving the following contributions during the remaining period of our doctorate research: (i) extension of the proposed SRAs by developing generic pipelines and guidelines to assist data engineers; (ii) development of algorithms to implement these pipelines in a parallel and distributed manner; and (iii) generation of FAIR-compliant standardized artificial datasets for further validations.

The remainder of this paper is organized as follows. Section 2 describes related work, Section 3 details the proposal, and Section 4 concludes the paper.

2. Related Work

We divide the related work in three groups. The first group consists of big data SRAs not designed for FAIR compliance. Rather, they prioritize real-time analytics. Examples include the traditional data warehousing, Kappa, Lambda, Liquid,

Table 1. Comparison of our work with the state-of-the-art.

Groups of Studies	Fits the concept of an SRA	Complies with FAIR	Retrieves source data by metadata	Big data analytics	Generic guidelines and pipelines
Group 1	✓	✗	✗	✓	✗
Group 2	✗	Not clear	✓	✗	✗
Group 3	✓	Partially	✓	✓	✗
Our proposal	✓	✓	✓	✓	✓

Solid, and Bolster architectures [Davoudian and Liu 2020]. There is also NeoMycelia [Ataei and Litchfield 2021], an SRA that employs microservices in its design. Metadata management is only encompassed in the traditional data warehousing, Bolster, and NeoMycelia SRAs. However, these SRAs lack features required by the FAIR Principles, such as retrieving data objects based on metadata and persistent metadata for non-existent data objects. Moreover, all metadata is handled a single component, which can diminish performance. Guidelines and generic data retrieval pipelines are also not proposed.

The second group includes FAIR implementations in context-specific repositories. For instance, there are data sharing repositories specific for COVID-19 patients [Borges et al. 2022] or for immunology data [Deng et al. 2022]. However, these solutions are domain-specific, lacking the generic nature of an SRA, thereby hindering reusability. Moreover, these studies do not explicitly clarify if all FAIR Principles are fulfilled and do not employ big data technologies, negatively impacting in the decision-making process. They also lack guidelines and generic data retrieval pipelines for data engineers.

The third group encompasses FAIR-compliant big data architectures that are generic enough to fit the concept of an SRA. To the best of our knowledge, only GADDS [Vazquez et al. 2022] can be included in this group. GADDS employs cloud storage and processing, keeping data in an object storage and metadata in a blockchain environment. However, GADDS is unable to achieve full FAIR compliance. Storing metadata in a blockchain environment restricts public access to the repository implemented by GADDS, compromising findability and accessibility. Additionally, this SRA lacks global unique identifiers, further impacting on its FAIR compliance.

Our work fills this gap in the literature. We propose BigFAIR and CloudFAIR, two novel FAIR-compliant SRAs capable of supporting big data analytics. We also propose a metadata warehouse generic model that is employed on these architectures to guarantee metadata persistence. We validate these solutions with real-world datasets, conducting a case study and a performance evaluation. We also plan to develop generic pipelines and guidelines to assist data engineers, algorithms to implement these pipelines in parallel and distributed environments, and artificial FAIR-compliant datasets to assist in further validations. Table 1 compares our proposal with the previously described groups of solutions.

3. Proposal

3.1. Objectives

In our doctorate research, we aim to develop different software reference architectures to assist data engineers in the process of implementing FAIR-compliant big data sharing repositories, each capable of satisfying distinct functional and performance requirements based on the context of the repository being implemented. We also seek to specialize human resources in the databases research field, promoting the writing and publication of scientific papers, and the participation in national and international scientific events.

Through these developments, our research also aims to achieve specific objectives. These include: (i) proposing FAIR-compliant SRAs, such as BigFAIR and CloudFAIR, which effectively handle big scientific data; (ii) validating these architectures through comprehensive case studies and performance evaluations; (iii) extending the proposed architectures by developing generic pipelines and guidelines to assist data engineers in their implementation; (iv) developing and validating algorithms to implement these pipelines in a parallel and distributed manner; and (v) generating FAIR-compliant standardized artificial datasets to facilitate the validation and comparison of different architectures.

3.2. Preliminary Results

3.2.1. The BigFAIR Architecture

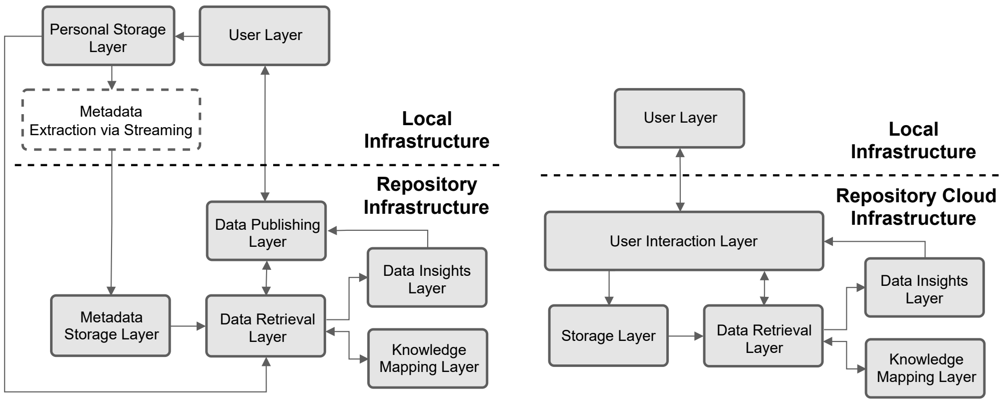
BigFAIR [Castro et al. 2022a] is a novel SRA designed to implement FAIR-compliant big data repositories. Figure 1a depicts its layers and components. The User Layer includes data providers and data consumers, while the Personal Storage Layer houses individual repositories owned by data providers. The Metadata Storage Layer stores extracted metadata first in a Metadata Lake and then in a Metadata Warehouse. The Data Retrieval Layer performs tasks such as retrieving, processing, and anonymizing research data using parallel and distributed frameworks. The Knowledge Mapping Layer employs ontologies and knowledge graphs to serve as a mapping between a generic data model known by data consumers and the data model implemented by the Metadata Storage Layer. The Data Insights Layer enables descriptive, predictive, and prescriptive analyses, while the Data Publishing Layer serves as the access point for data consumers.

The flexibility of BigFAIR allows data engineers to select the relevant components based on their specific environment and requirements. This architecture satisfies all FAIR principles, detailing which layer is responsible for each requirement. Its generic conceptual design allows for reusability in different contexts and integration with existing scientific data infrastructures. By separating data and metadata into distinct infrastructures, BigFAIR reduces development and maintenance costs, supports data ownership, and enables compliance with data protection regulations.

3.2.2. The CloudFAIR Architecture

CloudFAIR [Castro et al. 2022b] is an SRA that extends BigFAIR by managing data and metadata in the same cloud-based infrastructure, unburdening scientists in maintaining a local infrastructure and improving query performance. With encrypted storage for sensitive data, CloudFAIR ensures secure data management. Also, by being an extension of BigFAIR, CloudFAIR inherits is FAIR compliance and big data capabilities.

Figure 1b illustrates CloudFAIR. The User Interaction Layer acts as the interface for data consumers and providers, connecting them to the Repository Cloud Infrastructure. In the Storage Layer, data and metadata are loaded into a Data Lake, while the Metadata Warehouse and Metadata Governance Repository store relevant metadata and



(a) BigFAIR.

(b) CloudFAIR.

Figure 1. The proposed architectures.

provenance information. The Data Retrieval Layer utilizes the repository's big data infrastructure for decrypting, processing, and querying, while the Knowledge Mapping Layer facilitates data translation. Finally, the Data Insights Layer provides big data analytics.

3.2.3. Metadata Warehouse Generic Model

The effective implementation of a data warehouse relies on the modeling of numeric measures and dimensions [Kimball and Ross 2011]. Numeric measures represent the subjects of interest, while dimensions provide the context through a set of attributes. In a relational implementation, a star schema composed of fact and dimension tables is commonly employed. To achieve compliance with the FAIR Principles, we propose a generic model for the Metadata Warehouse, consisting of multiple dimension tables and one fact table [Castro et al. 2022a]. This model can be customized to include additional dimensions or attributes based on the specific scope of the repository being modeled.

The fact table captures the metadata extraction events for data cells at a given date and time, considering the data cell, repository, data provider, status, permissions, and license as dimensions. A data cell represents the intersection of an attribute and a tuple, such as a column value in a relational table or a field value in a document collection. The fact table contains surrogate keys for each dimension, enabling different perspectives for analysis, and it represents the size of the data cell, which can be measured in characters, bytes, or similar units. By adhering to the proposed generic model, repositories can associate data objects with comprehensive metadata, optimize analytical queries, and support the growth analysis of the repository over time. The flexibility and reusability of this model make it suitable for diverse repository contexts.

3.2.4. Case Study and Performance Evaluation

First, we describe a case study that deploys BigFAIR to the context of a real-world dataset of COVID-19 Brazilian patients [Castro et al. 2022a]. This dataset includes data from patients (e.g. birth year), exams (e.g. leukocytes), and outcomes (e.g. death). The case study includes the instantiation of the architecture and the execution of analytical queries on the dataset. BigFAIR is instantiated by storing the dataset in different Hadoop Distributed File System (HDFS) environments, extracting and loading metadata using Apache Kafka and Apache Spark, and implementing a Metadata Warehouse. The analytical queries involve analyzing data size over time and by data provider type, analyzing the relationship between patient sex and types of exams performed, and analyzing the data size of outcomes registered in emergency rooms by data provider and clinic. The results of the queries provide insights into the growth of the repository, the distribution of data providers, and the relationship between variables in the dataset. The case study demonstrates the effectiveness of BigFAIR in facilitating data sharing and analysis in a real-world context.

We also conduct a performance evaluation of the CloudFAIR and BigFAIR architectures [Castro et al. 2022b]. We conduct two types of experiments: (i) queries involving different storage components; and (ii) queries varying the number of data providers. No difference in performance is observed for queries involving only metadata. However, CloudFAIR outperforms BigFAIR by up to 75.96% when the queries encompass both data and metadata. This difference in performance is caused by the design choice of each architecture. CloudFAIR stores data and metadata in the same infrastructure as Apache Spark, improving performance. However, by doing so it renouces support to data ownership and flexibility, features that BigFAIR is able to achieve by employing separate infrastructures for data and metadata.

3.3. Methodology and Current Activities

The proposed methodology to achieve the objectives described in Section 3.1 includes a constant update of the literature review presented in Section 2, so that novel studies encompassing FAIR-compliant big data SRAs can be properly investigated. Besides, periodical meetings with other researchers in the same field are also included, so that discussions regarding the topics being researched and the results obtained can be performed.

Both the theoretical studies and the meetings held are intended to provide a basis for the work that is in development. This process encompasses proposing FAIR-compliant SRAs, such as BigFAIR and CloudFAIR, each with different emphasis (e.g. flexibility, performance, data ownership). Then, the next activity is the validation of these architectures through case studies and performance evaluations, employing real-world datasets and analyzing different types of requests. In the current moment of our doctorate research, these activities have all been concluded, as described in Section 3.2.

We are currently working on extending BigFAIR and CloudFAIR by proposing generic pipelines for data engineers. An example is a data retrieval pipeline that describes which components are accessed for processing a specific analytical query. We are also working on the proposal of guidelines, encompassing which technologies and procedures a data engineer needs for instantiating each layer in the architectures. Future activities encompass developing algorithms to implement the proposed pipelines in parallel and distributed environments, such as a data retrieval algorithm outlining the steps from obtaining connection parameters to executing the query. Another activity to be conducted is the development of FAIR-compliant standardized artificial datasets, facilitating performance comparisons between different architectures. These datasets should be comprised of scientific data and metadata, including a tool that allows data engineers to control dataset size for scalability experiments.

4. Conclusion

This doctorate research has the objective of developing different SRAs to assist data engineers in the process of implementing FAIR-compliant big data sharing repositories, each capable of satisfying distinct functional and performance requirements based on the context of the repository being implemented. To this end, we conducted a literature review that demonstrated a lack of architectural solutions capable of achieving full FAIR compliance in big data environments. To address this gap, we proposed BigFAIR and CloudFAIR, two novel FAIR-compliant big data SRAs. We validated these architectures through a case study and a performance evaluation. Future activities consist in proposing generic pipelines and guidelines for data engineers, algorithms to instantiate these pipelines, and FAIR-compliant artificial datasets for further validations.

References

Ataei, P. and Litchfield, A. (2021). NeoMycelia: A software reference architecture for big data systems. In *Proc. APSEC*, pages 452–462.

Borges, V. et al. (2022). A platform to generate FAIR data for COVID-19 clinical research in Brazil. In *Proc. ICEIS*, pages 218–225.

Castro, J. P. C. et al. (2022a). FAIR Principles and Big Data: A software reference architecture for Open Science. In *Proc. ICEIS*, pages 27–38.

Castro, J. P. C. et al. (2022b). Open Science in the cloud: The CloudFAIR architecture for FAIR-compliant repositories. In *Proc. ADBIS*, pages 56–66.

Chen, M., Mao, S., and Liu, Y. (2014). Big data: A survey. *Mob. Netw. Appl.*, 19(2):171–209.

Davoudian, A. and Liu, M. (2020). Big data systems: A software engineering perspective. *ACM Comput. Surv.*, 53(5):1–39.

Deng, N. et al. (2022). ImmuneData: an integrated data discovery system for immunology data repositories. *Database*, 2022.

Kimball, R. and Ross, M. (2011). *The data warehouse toolkit: the complete guide to dimensional modeling*. John Wiley & Sons.

Medeiros, C. B. et al. (2020). *IAP input into the UNESCO Open Science Recommendation*. Available at `https://www.interacademies.org/sites/default/files/2020-07/Open_Science_0.pdf`. Accessed in April 8, 2023.

Nakagawa, E. Y., Antonino, P. O., and Becker, M. (2011). Reference architecture and product line architecture: A subtle but critical difference. In *Proc. ECSA*, pages 207–211.

Sawadogo, P. and Darmont, J. (2021). On data lake architectures and metadata management. *J. Intell. Inf. Syst.*, 56(1):97–120.

Vazquez, P. et al. (2022). Globally accessible distributed data sharing (GADDS): A decentralized FAIR platform to facilitate data sharing in the life sciences. *Bioinformatics*, 38:3812–3817.

Wilkinson, M. D. et al. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data*, 3(1):1–9.