Digital Professionalism in Health and Care: Developing the Workforce, Building the Future
P. Scott et al. (Eds.)

doi:10.3233/SHTI220921

Strategies and Recommendation for Data Loading of FHIR-Based Data Marts with Focus on GDPR Compliance

Michael ANYWAR[a,1] , Björn SCHREIWEIS[a] and Hannes ULRICH[a]
[a] *Institute for Medical Informatics and Statistics, Kiel University and University Hospital Center Schleswig-Holstein, Campus Kiel, Germany*

Abstract. Interoperability and portability of healthcare data to enable research in the healthcare sector is an important factor towards precision medicine and a learning health system. With many safety-nets put in place like the European General Data Protection Regulation, and local standards like the broad consent set up by the German Medical Informatics Initiative, management and compliance to these standards across all systems and clinical data repositories becomes a daunting task. An appropriate process needs to be established especially when patient data is transferred to and from different systems and standards. On extraction and transforming, an appropriate method of loading the modified data to a destination where it can be read and accessed needs to be established besides functional compliance by the repository systems. This paper makes recommendations in relation to data load strategies while working with FHIR server-based data marts.

Keywords. Healthcare data Interoperability, ETL Process, FHIR, openEHR, Health Information Exchange, Electronic Health Records, GDPR

1. Introduction

As digitization advances and new technologies are introduced into the healthcare system, new data sources and the need for data integration are emerging [1]. ETL (extract, transform, load) processes are fundamental steps for medical data integration within the Medical Data Integration Center (MeDIC) at the University Hospital Schleswig-Holstein (UKSH) [2, 3]. Within the UKSH MeDIC, healthcare data is extracted from source systems and subsequently loaded into an openEHR repository which acts as a central clinical data repository (CDR). From the CDR, HL7 FHIR-based data marts are then loaded with data for reuse and sharing purposes to external partners through external systems. The incremental data load strategy is used to continuously add, update or delete the existing data, but without affecting the data marts' structures. ETL processes are utilized and optimized to facilitate the transformation and loading of clinical data from the heterogenous source systems into different standardized clinical data repository.

The use of patient-related data for research projects is subject to strict legal principles, such as the European General Data Protection Regulation (GDPR) Article 7(3) [4]. Thus, a withdrawal of consent creates the need for complete deletion of patient data, even within the technical and organizational version management. This urges an

[1] Corresponding Author: Michael Anywar, University Medical Center Schleswig-Holstein, Hörn Campus, Kaistraße 101, 24114 Kiel, Germany. Email: Michael.anywar@uksh.de

imminent need to identify and use data loading methods that will ensure compliance with consent revocation regulations and the GDPR in general, regardless of changes in patient consents.

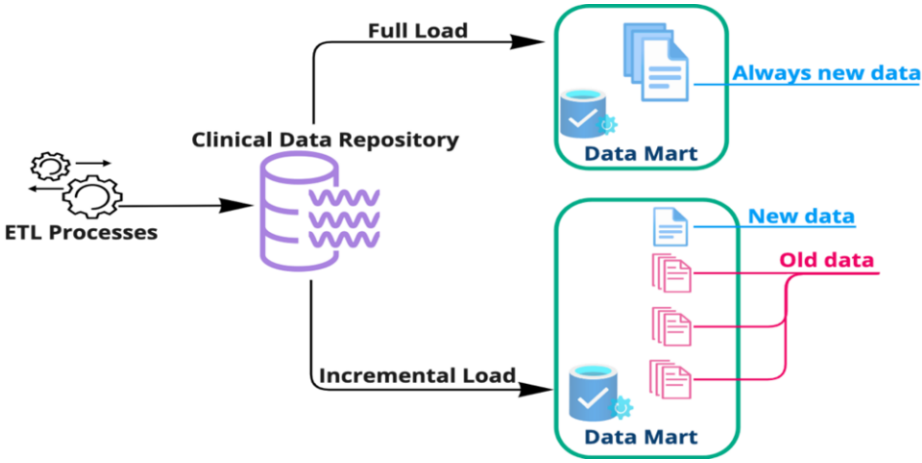

Figure 1. Representation of a full data load and an incremental data load

1.1. Requirements for patient privacy policy compliant data loading method

The demand for cross-border access to and sharing of health data across the European Union (EU) has grown strongly in order to harness the potential of health data for research, innovation and better healthcare. This has been further reinforced by the EU through its proposal for a domain-specific European Health Data Space [5]. However, for such an ecosystem to be successful, the systems used must be compliant with the GDPR.

In particular, Article 7, Chapter 3 of the General Data Protection Regulation states that the data subject has the right to withdraw consent at any time. Moreover, withdrawal must be as simple as giving consent. In terms of research data platforms, regardless of processing or data packetizing, the information has to be deleted and cleared from all systems once the patient withdraws the consent.

To realize GDPR compliance, systems will also need to have inbuilt mechanisms or ways of enabling compliance and trace patients' information. This can be in the form of not maintaining data history once consent is withdrawn, or a general procedure of purging patient data out of data marts. As FHIR-based data marts are core components for research data use and sharing in the German Medical Informatics Initiative(Semler et al., 2018) [6], these systems are required to have a mechanism of maintaining compliance of the GDPR especially when consents are withdrawn.

1.2. Objective

This study focuses on comparing two different data loading methods in respect to their data processing, their impact on downstream systems and GDPR compliance [7]; incremental data load and full data load. The latter strategy is used to initially populate information into an empty data mart.

2. Methods

To understand how themes and common issues related to the topic of data loading are handled, we performed semi-systematic literature review approach to develop knowledge on the topic. We used broad search terms, so as to widen the search results of learning mechanisms that developed overtime and do exist in relation to the topic.

A semi-systematic search strategy [8] was applied on academic databases, mainly; Google Scholar, PubMed, IEEE.org, and ScienceDirect. Searches for literature included combinations of key terms; "initial data load", "incremental data load" and "full data load" and "initial data load". This search queries resulted into 450 papers, whose abstracts were skimmed through and on reduction, 8 papers were identified to be useful for this research. The search was further extended to include gray literature from software providers like Oracle, SAP etc. The industry players were identified based on their strong expertise in data warehousing and in the field of big data handling.

3. Results

The literature search included 8 articles, meeting the requirements described above. The findings have been grouped per data load strategy and repository capabilities to be GDPR complaint.

3.1. European General Data Protection Regulation compliance

Among the principles, that both the Council for International Organizations of Medical Sciences (CIOMS) [9] and the United Nations Convention on the Rights of Persons with Disabilities (CRPD) article 25 [10] have in common with the EU GDPR, is the focus and their advocation for free and informed consents which have to be adhered to both in clinical care and research. The effect on research data marts is enormous, especially if there are various systems interconnected with vast amounts of clinical data held in the data marts. Consent results in either deleting all patient data from the research data mart or incrementally deleting the patient's data from data marts. Unfortunately, this does not correspond to the standardized behavior of FHIR server implementations in terms of the default FHIR delete operation. "DELETE [base]/[type]/[id]" rather refers to subsequent non-version specific reads of a resource which returns a 410 HTTP status code and that the resource is no longer found through search interactions [11]. This means that the resource is not destroyed, hence can be found through FHIR version-specific reads. This DELETE interaction significantly deviates from the objective of GDPR compliance as the delete interaction does not remove a resource's version history from the FHIR server. From a version history perspective, deleting a resource is equivalent to creating a special entry in the version history that has no content and is marked as deleted. Also, there is no support for deleting previous versions, so deleted resources may still be accessible.

3.2. Full Load and Incremental Data Load

The full load strategy is straightforward to implement as the data mart just must be truncated and the whole data mart or table reloaded again. The reload is applied often on clean and fresh structure. This eliminates the overhead tasks of key management and concern of inconsistent and unprocessed data. Regardless of the situation, all data are updated every time the full load strategy is used to load a table or data mart. Thus, the implementation of a full data load strategy is well manageable in terms of complexity. However, this simplicity has its disadvantages: the update of a single data point within a repository containing a high amount of longitudinal data is less efficient in terms of computing resources utilization, hence it is unsustainable and presents challenges during risk management [12]. Another scenario is the nightly upload to, for example, a FHIR server for data extraction. The full data load strategy becomes untenable if the processing time overgrows the update periods, i.e., takes more than one day.

From a technical perspective many system architects use administrative fields like “date_time_updated”, “date_time_inserted” to keep the integrity of tables and the data mart. When using a full data load strategy, these fields lose their significance as they refer always to the timestamp of the last full load. But there are use cases where this information is of interest for subsequent research. For example, it can be used to identify temporal relationships in processes.

Unlike a full load, an incremental data load involves loading only new or updated data after a previous synchronization or effecting the deletion of specific resource, without affecting the structure of the data mart entirely with clear performance benefits [13]. With increasing data size, and high data rate, full loading becomes inadequate and inefficient, hence incrementally loading the changes being a more practical strategy [12].

4. Discussion and Conclusion

When working with ETL processes, the notion of choosing a full data loading method sounds feasible and tempting due to its simplicity, but as the volume of patient data gradually increases, a choice must be made on which options to choose and how it will be implemented regardless of the complexity. However, though performing a full data load offers the opportunity of GDPR compliance for FHIR-based data marts, it becomes unsustainable in the long run with bulging huge amounts of data in the source systems. This concern can however, be resolved by an incremental strategy which is less susceptible to unexpected failures: e.g., in case of a network failure during synchronization, the process can be continued more effortlessly. This offers better risk management and sustainability. Executing the incremental data load guarantees a consistent system performance, especially if the strategy is executed in a uniform time-interval with small chunks of data. Contrary to full data load, it maintains the correct "lastUpdates" timestamps and retains historical data in the data mart as modification is only performed on new data and previous data are preserved. But the main drawback of this strategy is the complexity in its design and implementation. Special attention is hence needed in order to have features like “date_time_updated” or “date_time_inserted” correct and be used correctly for other purposes.

A major limitation of this study is the current focus on only two data loading strategies. Therefore, a continuing study should broaden the scope within the literature review and explore additional strategies according to the previously established criteria.

Thus, from our analysis, the combination of both approaches presents better possibilities for data quality and conformance to GDPR. Within the data integration context and the application of a FHIR data mart, our analysis reveals that FHIR operations per se are not GDPR compliant. This is a drawback for systems using the standard FHIR DELETE operation when removing patient data. A strong recommendation in terms of GDPR compliance, is to include permanent deletion in the FHIR standard. This allows patient consent to be respected while still promoting research through the re-use of clinical data.

Acknowledgement

This work was funded by the Federal Ministry of Education and Research (grant number 01ZZ1802T).

References

[1] Avazpour I, Grundy J, Zhu L. Engineering complex data integration, harmonization and visualization systems. J Ind Inf Integr. 2019 Dec 1;16:100103.

[2] Kock-Schoppenhauer AK, Schreiweis B, Ulrich H, Reimer N, Wiedekopf J, Kinast B, et al. Medical Data Engineering – Theory and Practice. Commun Comput Inf Sci. 2021;1481. Spri:269–84.

[3] Haarbrandt B, Schreiweis B, Rey S, Sax U, Scheithauer S, Rienhoff O, et al. HiGHmed – An Open Platform Approach to Enhance Care and Research across Institutional Boundaries. Methods Inf Med. 2018 Jul 17;57(S 01):e66–81.

[4] Regulation (EU) 2016/679 of the European Parliament and the Council of 27 April 2016. 2016.

[5] A European Strategy for data | Shaping Europe's digital future [Internet]. [cited 2022 Jun 20]. Available from: https://digital-strategy.ec.europa.eu/en/policies/strategy-data

[6] Semler SC, Wissing F, Heyder R. German Medical Informatics Initiative. Methods Inf Med. 2018 Jul 1;57(S 01):e50–6.

[7] Santos RJ, Bernardino J. Real-time data warehouse loading methodology. ACM Int Conf Proceeding Ser. 2008;299:49–58.

[8] Snyder H. Literature review as a research methodology: An overview and guidelines. J Bus Res. 2019 Nov 1;104:333–9.

[9] Vijayalakshmi M, Minu RI. Incremental Load Processing on ETL System through Cloud. 2022 Int Conf Adv Technol ICONAT 2022. 2022;1–4.

[10] Article 25 – Health | United Nations Enable [Internet]. [cited 2022 Jun 12]. Available from: https://www.un.org/development/desa/disabilities/convention-on-the-rights-of-persons-with-disabilities/article-25-health.html

[11] Benson T, Grieve G. Principles of health interoperability: SNOMED CT, HL7 and FHIR. Springer; 2016.

[12] Biswas N, Sarkar A, Mondal KC. Efficient incremental loading in ETL processing for real-time data integration. Innov Syst Softw Eng. 2020;16(1):53–61.

[13] Jörg T, Deßloch S. Towards generating ETL processes for incremental loading. ACM Int Conf Proceeding Ser. 2008;299:101–10.