

Comparison of OpenEHR and HL7 FHIR Standards

Jacek Kryszyn, Waldemar T. Smolik, Damian Wanta, Mateusz Midura and Przemysław Wróblewski

Abstract—Health informatics is characterized by the need to securely store, process and transmit large amounts of sensitive medical data while ensuring interoperability with other systems. Among many standards used in such systems there are two which have gained interest in recent years and cover most of those needs: openEHR and HL7 FHIR. In this paper, both standards are discussed and compared with each other. The architecture of both systems, the similarities and differences, methods of data modeling and ensuring interoperability were presented.

Keywords—medical computer systems; healthcare information technology; openEHR; HL7 FHIR

I. INTRODUCTION

MANY standards are used in health informatics, whether for demographics or medical data. Some of them define general concepts of healthcare systems and some focus on a specific aspect. Examples of former are ISO 18308:2011 [1] which describes the requirements of the electronic healthcare record (EHR) architecture, ISO 13940:2015 [2] which covers the concept definitions needed whenever structured information in healthcare is specified as requirement, or the HL7 RIM (Reference Information Model) standard [3], which describes the reference information model. An outstanding example of latter is DICOM [4] which defines the medical imaging data format and the communication of medical imaging and database systems. It is mainly focused on image data and achieves its goals, among others, due to being object-oriented, open, and extensible. Other specialized standards are LOINC [5], SNOMED CT [6] and ICD-10 [7] which systematize medical terminology. Many healthcare computing standards are incomplete in the sense that they only describe one aspect, such as the format of the data being transferred, without defining the protocol for their transmission (or vice versa). This is the case with most standards issued by HL7. An example is HL7 Messaging, both v2.x [8] and v3 [9]. In this standard, the structure of information object is taken from other standards: HL7 RIM and HL7 CDA (Clinical Document Architecture) [10]. Moreover, HL7 v3 does not impose data encoding, but proposes default XML encoding. ISO/HL7 10781:2015 [11] describes a functional model of medical systems. CEN/ISO EN 13606 [12] describes the communication of systems in the context of preserving the original clinical significance

This work was supported by the grant No. 504/04763/1034/43.052202 "Porównanie wydajności standardów HL7 FHIR oraz openEHR" financed from internal science fund for employees of Warsaw University of Technology supporting scientific activity in the discipline of Biomedical Engineering.

J. Kryszyn, W. T. Smolik, D. Wanta, M. Midura and P. Wróblewski are with Faculty of Electronics and Information Technology, Warsaw University of Technology, Warsaw, Poland (e-mail: jacek.kryszyn@pw.edu.pl).

as well as the confidentiality of particularly sensitive data. Most standards used in healthcare informatics, such as the American CCR (Continuity of Care Record), HL7 CDA and ISO 13606, adopt a document-centric approach (e.g. report, patient records, discharge, etc.). The complex structure of clinical documents with medical data is modeled and the format in which the data is exchanged (e.g. XML) is described. These standards do not describe the communication protocol, nor do they focus on the transfer of atomic units of health data (such as heart rate, blood pressure, glucose, etc.). Although the HL7 CDA is designed to build complex clinical documents, it can be used to transmit individual data elements, but by design this standard is very complex and by itself, without specifying a communication protocol, does not ensure interoperability (ability to work with other products) of information systems.

The openEHR [13] and the HL7 FHIR [14] standard standard that are the subject of this work are relatively new and their history begins in 2003 and 2012, respectively. Their acronyms should be read like the English words "open air" and "fire". These standards arouse great interest and are popular, which results from the greater completeness of their specifications. HL7 FHIR is focused on interoperability and data transmission, although it provides its own data model. openEHR on the other hand is about data modeling and persistence, although from some time it also provides specification of data transfer using REST API. Both standards use existing and commonly used IT standards. The application layer protocol is based on the HTTP protocol layer. Data encoding in XML and JSON can be used interchangeably. Stateless software architecture is assumed, which is very popular in Internet business solutions nowadays. Currently, a large number of open source implementations are available that enable local and remote testing. There is also a large community of developers who support each other.

The FHIR and openEHR standards are proposed by different organizations and were intended to solve different issues regarding health care informatics. They are distinguished by fundamental differences. Despite this, they both can be used similarly due to simplified features of data modeling (HL7 FHIR) and data transfer (openEHR). This is a source of confusion for many who think that they are competitive and intend to do exactly the same. The aim of this work is to compare both standards. The comparison will consider aspects such as the overall concept and system compatibility, the data model and the application level communication protocol. The flexibility will be discussed, including, but not limited to, the type of database systems supported, query syntax and complexity, and supported data encoding formats. The



extensibility of the standard, including the addition of new data elements and terminology, will be assessed. Existing libraries and open source software will be presented.

II. GENERAL CHARACTERISTICS OF THE STANDARDS

A. HL7 FHIR

The information provided in this chapter applies to version 4.3.0 of the standard (completed on May 25th, 2022). The HL7 FHIR standard (Fast Healthcare Interoperability Resources) defines the data format and protocol for the exchange of medical information between different computer systems, regardless of how it is stored in these systems. The main purpose of the standard is to ensure interoperability between systems. It doesn't impose how the data should be stored in those systems. It is up to vendors of the software to decide how the data should be persisted. They have to provide means to translate their proprietary data to a format proposed by HL7 FHIR in order to exchange data with other systems. HL7 FHIR format is meant to be understood as many formats that are approved by the organization developing the standard, i.e. HL7 (Health Level Seven). FHIR is based on Internet standards such as HTTP, XML, JSON and OAuth, which undoubtedly facilitates software implementation. FHIR assumes that communication via HTTP will follow the REST (Representational State Transfer) architecture style, in which the communication protocol is stateless [15]. In such architecture, the receiver does not save the session state. The sender sends complete data each time, which must be intelligible in isolation from previous or subsequent messages. Stateless architecture facilitates scalability and software development, however, it can affect system performance due to complete session state transfer every time. In the FHIR standard, information objects, called resources, are transferred, which are based on other HL7 standards, such as the HL7 CDA, which allows for easier integration with existing systems using the data structure defined in CDA (Clinical Document Architecture). Resources present small portion of data, some simple entities which may be further extended and binded in more complicated objects. They are not only medical objects, but also demographic and administrative. A resource is described by an information model that defines data elements, constraints, and relationships between objects such as patient, procedure, meeting, observation, etc. There are currently about 150 defined resources. Resources have six elements: a resource type, an identifier, metadata, human readable XHTML summary, an extensible block for future definition and optional set of defined data elements - a different set for each resource type. All resources may have a URL that identifies it (for example, url/patient/1234 where 1234 is the id of the resource). Resource data items contain relationships to other items and resources. The Application Programming Interface (API) describes the FHIR protocol as a set of operations on resources known as "Interactions". Interactions are performed directly on the server resources using HTTP requests and responses. The basic interactions are summarized in Table I. The more advanced interactions are Batch/Transaction (perform Create, Read, Update, Delete - CRUD operations on a set of resources in a single interaction),

VRead (read the state of a specific version of the resource), Patch (update an existing resource by making a set of changes to it), and Capabilities (get a capability statement for the system). Moreover, there is also Operation interaction which allows to perform more complicated actions using an RPC-like paradigm. Interactions are categorized into Instance, Type, and System types. The API also describes endpoints for validation and documentation.

TABLE I
FHIR INTERACTION

Name	Interaction
Create	create a new resource with assigned server id = POST url/{resourceType}
Read	reading the current resource status = GET url/{resourceType}/{id}
Update	update an existing resource by its id (or create one if new) = PUT url/{resourceType}/{id}
Delete	delete resource = DELETE url/{resourceType}/{id}
Search	search for resources by filtering criteria = GET url/{resourceType}?search parameters
History	get the change history for a specific resource = GET url/{resourceType}/{id}/history

Depending on the server, different sets of interactions may be available, as well as different sets of resources to handle. Servers must support a capability statement resource that describes which interactions and resources are available. The specification of the standard provides mechanisms of extending the possibilities of resources through Extensions and Profiles. These allow to further specialize resources in order to correspond to local requirements and specifics. Resources can be transferred interchangeably in XML or JSON format. The standard includes access authorization and data encryption.

B. openEHR

The information provided in this chapter applies to specifications which were current at the time of writing (November, 2022). The openEHR standard, developed by the openEHR Foundation, describes the management, storage, retrieval and exchange of data in EHR systems [16] [17]. Actually, currently it is described more as a technology consisting of many open specifications, clinical models and software rather than one standard, hence there is no specific version by which the whole can be identified. There are four "programs" (committees) which manage all those components. openEHR uses a multi-level data model in which information is separated from knowledge divided into content element definitions (called archetypes) and content-specific data set definitions (called templates). Originally it was described as two-level modeling in which archetypes and templates were on the same level. The first level contains a reference model with a small set of classes that describe how data is represented in a patient record. It contains data types, data structures, identifiers and design patterns. It is a logical model, published as a set of specifications and UML diagrams. It covers basic engineering requirements. The

TABLE II
 OPENEHR API FUNCTIONS

Command	Meaning
POST /ehr	Create an EHR
PUT /ehr/{ehr_id}	Create an EHR with the specified ID
GET /ehr/{ehr_id}	Read an EHR with the specified ID
POST /definition/template/adl1.4	Register the template
GET /definition/template/adl1.4	Read the list of templates
GET /definition/template/adl1.4/{id}	Read a single template with the specified ID
POST /ehr/{ehr_id}/composition	Save the composition in the specified EHR
GET /ehr/{ehr_id}/composition/{id}	Read a particular version of the composition
DELETE /ehr/{ehr_id}/composition/{id}	Logical removal of the composition
Execute Query	<pre> SELECT c/name/value FROM COMPOSITION c WHERE c/uid/value = \$uid GET {baseUrl}/query/myQuery? uid=2340-sd424fdv87::url::1 </pre>

general structure of the patient's health record is described by the classes called entries, divided into care and administrator entries: observation, evaluation, instruction, action, administrator entry. There are other entries derived from the above-mentioned. Knowledge level, defined by the ADL (Archetype Definition Language), provides formal definitions of clinical concepts in the form of archetypes and templates. Archetypes contain metadata, a set of names, rules, and constraints that describe how to use reference model blocks to create tree-like data structures. Archetypes are the maximum shareable datasets that contain all the data items required by all use cases. Archetypes specify which data points are mandatory and which are optional, their type, and validation criteria. Archetypes are designed by field professionals, typically in international collaborative environments (including discussion/review platforms). Templates are used to combine several archetypes into a larger structure intended for a specific use case, for example as a basis for hospital discharge. A template may also restrict, hide, or set default values on the archetypes and the reference model on which it is based. Templates do not add new clinical concepts; they use and restrict concepts defined by existing archetypes. openEHR defines application layer communication using the HTTP protocol. A stateless REST communication architecture is assumed, however other solutions such as SOAP or Google Protocol Buffers may be used. openEHR uses the concept of a resource that can be identified, addressed, hosted or managed via a URI/URL. Resources must be encoded in XML or JSON. The API is divided into patient record (EHR) functions, containing queries directed to the server (Query) and introducing new definitions and models on the server (Definitions). Sample API commands are listed in Table II.

III. COMPARATIVE ANALYSIS

A. General concept and compatibility of systems

Both standards were designed with interoperability being one of the important goals, but they try to solve this problem differently. Main task of HL7 FHIR is to allow data exchange between systems. It doesn't require systems to persist data in a certain way. Software might have its own proprietary data model, it just have to be able to translate its own data to HL7 FHIR model and vice versa. This allows to communicate existing systems from various vendors. openEHR, on the other hand, was intended to provide solution for data persistence and data modeling. It delivers a reference model and archetypes created by professionals which are reusable. Every openEHR system is able to understand and process data from another openEHR system; it should be possible to exchange an openEHR repository from one vendor to another without any modifications. In this regard, true and full interoperability would be achieved if all EHR systems would use openEHR. In such a case there would be no need for data translation between systems. There are, however, some concerns about the actual way that openEHR data should be persisted [18]. Both HL7 FHIR and openEHR, to a certain extent, enable the implementation of standard-compatible systems supporting the electronic patient record. Since HL7 FHIR defines data model,

it might be used by vendors creating new solutions as their internal data model. This is not endorsed by HL7 itself, but it is possible to find many discussions in which HL7 FHIR users show that using NoSQL databases it is possible to store FHIR documents as is. This would simplify communication with external software. On the other hand, openEHR quite recently added specification of REST API allowing CRUD operations in a similar manner to HL7 FHIR. It seems that openEHR better describes the problem of conversion to other standards. openEHR is strictly hierarchical and implements a multi-level modeling approach (reference model and archetypal model) for its data structures and concepts. FHIR adopts a flatter representation of the concepts, although it refers to the hierarchical model contained in the HL7 CDA and allows the creation of Bundles corresponding to the Compositions in openEHR. The encoding and communication method is similar, although openEHR includes extensions. openEHR, unlike FHIR, uses inheritance to implement data specialization and complex data compositing. There are works covering translation from openEHR to HL7 FHIR data model and vice versa [19], [20], [21]. It seems that it should be always possible to translate to openEHR since its architecture allows creation of very complicated data structures. For example, Clinical Knowledge Manager (CKM, a system for collaborative development, management and publishing of archetypes) provides a template called ePrescription [22] which was created to fit HL7 FHIR Medication Order. On the other hand, complexity of openEHR makes it sometimes impossible to translate

openEHR templates to structures described by much simpler HL7 FHIR data model.

B. Standard flexibility and extensibility

Regardless of the standard, it is possible to support any database system, both relational and noSQL. Both specifications support various types of simple and complex data. OpenEHR AQL (Archetype Query Language) queries are very similar to SQL syntax, and they allow very advanced processing of data stored in EHR repository. It is possible to perform AQL queries using REST API by sending the query as a parameter in the requests body. An example from openEHR REST API documentation for GET /query/aql:

```
SELECT e/ehr_id/value, c/context/start_time/value as
startTime, obs/data[at0001]/events[at0006]/data[at0003]/
items[at0004]/value/magnitude AS systolic, c/uid/value AS
cid, c/name FROM EHR e CONTAINS COMPOSITION
c[openEHR-EHR-COMPOSITION.encounter.v1]
CONTAINS OBSERVATION obs[openEHR-EHR-
OBSERVATION.blood_pressure.v1] WHERE obs/
data[at0001]/events[at0006]/data[at0003]/items[at0004]/
value/magnitude = 50.
```

With the FHIR standard, queries are also possible by specifying search parameters as an URL parametr. This might be unreadable and hard to maintain in case of more complicated queries, so this feature of HL7 FHIR seems to be less powerful. Both FHIR and openEHR use XML and JSON to encode data, and openEHR also allows to use its own language (ADL/AOM). The JSON encoding looks similar for both standards, but with openEHR encoding with XML leads to very large text files. This is because in openEHR the attributes of the data items are coded as consecutive data items. In FHIR, values are carried in attributes and data items do not contain values. The openEHR archetypes and FHIR resources are built from various components that can be used in other compositions and bundles. The archetype and resource repositories are systematically expanded (about 600 archetypes for openEHR and about 150 resources for FHIR). In openEHR, archetype compositions, extensions, and specializations are based on direct reference, nesting, and pattern mechanisms. In FHIR, complex structures are built by reference to resources, contained resources, and bundles. These mechanisms allow for the development of the standard, and on the other hand reduce the costs of software development. The internal term system is not exhaustive for both standards. Both standards support a variety of external terminology systems to encode element names and sometimes data values. Development tools are available to add terminology when building new archetypes and resources, such as the Archetype Editor or clinFHIR.

C. Existing tools and software

It is impossible to describe in such a short text all the existing tools and software, even the most important ones. Only a few representative examples will be given.

In the case of HL7 FHIR, there is a list of open source implementations which covers a vast number of software written in different programming languages [23]. The list contains

libraries, SDKs, clients and servers. It is worth mentioning HAPI FHIR and Firely .Net SDK libraries. The HAPI FHIR library is an open source Java implementation [24]. This library defines classes for each FHIR resource. It provides exemplary implementations of both a HL7 FHIR server and a client. The library also provides a tool to validate modeled FHIR resources to ensure that resources are compliant with the specification. Firely .Net SDK is a library for the C# language [25]. It includes, but is not limited to: classes describing data models, helper classes for working with metadata, XML and JSON parser, FHIR client example, data validation tool based on profiles, and query validation tool. There is also an open source C# FHIR server built by Firely and maintained by Incendi [26]. FHIR servers, including open source ones, are provided by large companies such as IBM and Microsoft. However, there are some caveats regarding those projects. IBM doesn't support its open source solution anymore. This server is currently known as LinuxForHealth FHIR Server [27]. On the other hand, IBM's Watson Health (their closed FHIR server) is now handled by Merative [28]. In case of Microsoft's open source software, it is intended to run specifically on the Microsoft Azure cloud [29].

The official website of openEHR also lists tools and libraries, both open source and commercial. Interestingly, there is no mention of a Java implementation of openEHR created by a team from Sweden and donated to openEHR Foundation [30] which at the time became an open source reference library [31]. It is divided into several software components. openehr-rm-core implements all the specifications of the reference information and query model. rm-builder is used to construct objects using archetypes. The openehr-aom and openehr-ap components provide archetype representation, object creation, and validation support. adl-parser implements ADL and converts archetypes in text format to the AOM (Archetype Object Model) form. Other Java implementations, currently listed on the official openEHR website, are archie [32] and EHRbase [33]. There are also implementations for other languages: openEHR.NET for C# [34], pyEHR for Python [35], skoba for Ruby [36], but their latest releases count at least several years, so they seem to be unsupported from a very long time. OpenEHR software developers can use open source servers such as EHRServer, EHRbase or OpenEyes [37]. When testing the software, one can use a publicly available data set for testing the correctness and performance of ORBDA servers [38]. Recently Microsoft announced that it will support openEHR and became an industry partner of openEHR Foundation [39].

IV. CONCLUSIONS

The aim of this study was to present and compare two standards for data exchange and persistence in healthcare systems: HL7 FHIR and openEHR. Both standards were created to solve two different problems: the exchange of medical data between systems of different manufacturers (HL7 FHIR) and to create a vendor-neutral data model that could be understood by any system based on this standard (openEHR). Each of them achieves its goal, and each of them contains features that overlap with those of the other standard. HL7 FHIR defines the

data model that must be used to transfer information between systems. It is not as extensive as the openEHR model, but it was created according to the Pareto principle (80/20): the simplicity of the model allows it to describe most, but not all cases encountered in medicine. For manufacturers working on new medical systems, this model may be tempting to use as an internal data model, which would avoid the problem of translation between the data representation within the system and the data representation in FHIR messages. openEHR, on the other hand, has had a REST API specification for several years that can be used to perform CRUD operations on a data repository. In this way, it is possible to communicate the openEHR repository with external systems, which coincides with the functions of the HL7 FHIR. In theory, both systems are complementary and one can imagine that the HL7 FHIR would be used to transfer data stored in openEHR, but it would not always be a trivial task to translate one into the other due to openEHR's much broader data modeling capabilities. It would be possible to use the resource called Questionnaire in HL7, which is a quite generic structure, but it would involve some work. On the other hand, the overlap of some features of the standards, even to a small extent, is a source of confusion for many users. Due to the existence of a certain defined data model and REST API in both standards, these users treat both standards as competitors and substitutes. There are even scientific works which treat them like that [21]. On the internet one can find a lot of questions formulated "HL7 FHIR vs openEHR" etc. There is even a section on the openEHR discussion group dedicated to this topic, where one can find very extensive explanations about the relationship between the two standards [40]. Therefore, the question arises - is the existence of both standards justified and beneficial from the user's point of view? It seems that the best solution would be to merge them into one, using the best features of both. On the other hand, it would be interesting to compare open-source solutions for both standards, treating them as competing solutions for storing and transmitting medical data. Perhaps, analyzing the efficiency of such solutions, one could notice a certain advantage of one standard over the other. As shown in [37], the use of openEHR in a hospital IT system is associated with a noticeable reduction in efficiency and throughput compared to the solution based on a proprietary data model. There are no similar tests for other openEHR and HL7 FHIR implementations. Most of the scientific work focuses on the application of both standards and does not resolve these issues.

REFERENCES

- [1] "Iso - iso 18308:2011 - health informatics — requirements for an electronic health record architecture." [Online]. Available: <https://www.iso.org/standard/52823.html>
- [2] "Iso - iso 13940:2015 - health informatics — system of concepts to support continuity of care." [Online]. Available: <https://www.iso.org/standard/58102.html>
- [3] "Reference information model (rim) downloads — hl7 international." [Online]. Available: <https://www.hl7.org/implement/standards/rim.cfm>
- [4] "Dicom." [Online]. Available: <https://www.dicomstandard.org/>
- [5] "Home - loinc." [Online]. Available: <https://loinc.org/>
- [6] "Snomed - home — snomed international." [Online]. Available: <https://www.snomed.org/#>
- [7] "Icd-10 version:2019." [Online]. Available: <https://icd.who.int/browse10/2019/en>
- [8] "Hl7 standards product brief - hl7 version 2 product suite — hl7 international." [Online]. Available: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=185
- [9] "Hl7 standards - section 1e: Version 3 (v3) — hl7 international." [Online]. Available: https://www.hl7.org/implement/standards/product_section.cfm?section=14
- [10] "Hl7 standards product brief - cda® (hl7 clinical document architecture) — hl7 international." [Online]. Available: https://www.hl7.org/implement/standards/product_brief.cfm?product_id=496
- [11] "Iso - iso/hl7 10781:2015 - health informatics — hl7 electronic health records-system functional model, release 2 (ehr fm)." [Online]. Available: <https://www.iso.org/standard/57757.html>
- [12] "Iso - iso 13606-1:2019 - health informatics — electronic health record communication — part 1: Reference model." [Online]. Available: <https://www.iso.org/standard/67868.html>
- [13] "openehr home." [Online]. Available: <https://www.openehr.org/>
- [14] "Fhir v4.3.0." [Online]. Available: <https://hl7.org/fhir/>
- [15] E. Sundvall, M. Nyström, D. Karlsson, M. Eneling, R. Chen, and H. Öрман, "Applying representational state transfer (rest) architecture to archetype-based electronic health record systems," *BMC Medical Informatics and Decision Making*, vol. 13, pp. 1–25, 5 2013. [Online]. Available: <https://bmcmidinformatdecismak.biomedcentral.com/articles/10.1186/1472-6947-13-57>
- [16] L. Min, Q. Tian, X. Lu, and H. Duan, "Modeling ehr with the openehr approach: An exploratory study in china philip payne," *BMC Medical Informatics and Decision Making*, vol. 18, pp. 1–15, 8 2018. [Online]. Available: <https://bmcmidinformatdecismak.biomedcentral.com/articles/10.1186/s12911-018-0650-6>
- [17] C. Pahl, M. Zare, M. Nilashi, M. A. de Faria Borges, D. Weingaertner, V. Detschew, E. Supriyanto, and O. Ibrahim, "Role of openehr as an open source solution for the regional modelling of patient data in obstetrics," *Journal of Biomedical Informatics*, vol. 55, pp. 174–187, 6 2015.
- [18] L. Wang, L. Min, R. Wang, X. Lu, and H. Duan, "Archetype relational mapping - a practical openehr persistence solution," *BMC medical informatics and decision making*, vol. 15, 11 2015. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/26541142/>
- [19] E. Rinaldi and S. Thun, "From openehr to fhir and omop data model for microbiology findings," *Public Health and Informatics: Proceedings of MIE 2021*, pp. 402–406, 7 2021. [Online]. Available: <https://ebooks.iospress.nl/doi/10.3233/SHTI210189>
- [20] C. Bönisch, D. Keszytüs, and T. Keszytüs, "Harvesting metadata in clinical care: a crosswalk between fhir, omop, cdisc and openehr metadata," *Scientific Data* 2022 9:1, vol. 9, pp. 1–7, 10 2022. [Online]. Available: <https://www.nature.com/articles/s41597-022-01792-7>
- [21] G. Fette, M. Ertl, and S. Störk, "Translating openehr models to fhir," *Studies in health technology and informatics*, vol. 270, pp. 1415–1416, 6 2020. [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/32570686/>
- [22] "eprescription (fhir) - clinical knowledge manager." [Online]. Available: <https://ckm.openehr.org/ckm/templates/1013.26.80>
- [23] "Open source implementations - fhir - confluence." [Online]. Available: <https://confluence.hl7.org/display/FHIR/Open+Source+Implementations>
- [24] "hapi.fhir/hapi-fhir: Hapi fhir - java api for hl7 fhir clients and servers." [Online]. Available: <https://github.com/hapi.fhir/hapi-fhir>
- [25] "Firelyteam/firely-net-sdk: The official firely .net sdk for hl7 fhir." [Online]. Available: <https://github.com/FirelyTeam/firely-net-sdk>
- [26] "Firelyteam/spark: Firely and incendi's open source fhir server." [Online]. Available: <https://github.com/FirelyTeam/spark>
- [27] "Linuxforhealth/fhir: The linuxforhealth fhir server and related projects." [Online]. Available: <https://github.com/LinuxForHealth/FHIR>
- [28] "Healthcare data, technology and analytics — merative." [Online]. Available: <https://www.merative.com/?ref=ibm>
- [29] "microsoft/fhir-server: A service that implements the fhir standard." [Online]. Available: <https://github.com/microsoft/fhir-server>
- [30] "openehr/java-libs: Standard java libraries for java implementations of openehr." [Online]. Available: <https://github.com/openehr/java-libs>
- [31] R. Chen and G. Klein, "The openehr java reference implementation project," *Studies in Health Technology and Informatics*, vol. 129, pp. 58–62, 2007. [Online]. Available: http://svn.openehr.org/ref_impl_java
- [32] "openehr/archie: Openehr library implementing adl 2, aom 2, bmm, rm 1.0.4 and many tools." [Online]. Available: <https://github.com/openehr/archie>
- [33] "ehrbase/ehrbase: An open source openehr server." [Online]. Available: <https://github.com/ehrbase/ehrbase>

- [34] H. Demski, S. Garde, and C. Hildebrand, "Open data models for smart health interconnected applications: The example of openehr standards, technology, machine learning, and modeling," *BMC Medical Informatics and Decision Making*, vol. 16, pp. 1–9, 10 2016. [Online]. Available: <https://bmcmmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-016-0376-2>
- [35] "crs4/pyehr: Python modules for managing openehr projects." [Online]. Available: <https://github.com/crs4/pyEHR>
- [36] "skoba/openehr-ruby: Ruby implementation of the openehr specification." [Online]. Available: <https://github.com/skoba/openehr-ruby>
- [37] J. Kryszyn, K. Cywoniuk, W. T. Smolik, D. Wanta, P. Wróblewski, and M. Midura, "Performance of an openehr based hospital information system," *International Journal of Medical Informatics*, vol. 162, p. 104757, 6 2022.
- [38] D. Teodoro, E. Sundvall, M. J. Junior, P. Ruch, and S. M. Freire, "Orbda: An openehr benchmark dataset for performance assessment of electronic health record servers." *PloS one*, vol. 13, p. e0190028, 2018. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/29293556><http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5749730>
- [39] "Microsoft join openehr international as diamond industry partners." [Online]. Available: https://www.openehr.org/news_events/openehr_news/380
- [40] "Latest integration/hl7 fhir topics - openehr." [Online]. Available: <https://discourse.openehr.org/c/integration/fhir/101>