Conceptualizing open health data platforms for low- and middle income countries

Daniel Kapitan[a,b], Femke Heddema[a], Julie Fleischer[a], Chris Ihure[c], Steven Wanyee[d], Alessandro Pietrobon[e], Ryan Chrichton[f], John Grimes[g], Paula van Brakel[a], Mark van der Graaf[a], Nicole Spieker[a]

[a] *PharmAccess Foundation, Amsterdam, the Netherlands,*

[b] *Eindhoven University of Technology, Eindhoven, the Netherlands,*

[c] *PharmAccess Kenya, Nairobi, Kenya,*

[d] *IntelliSOFT, Nairobi, Kenya,*

[e] *ONA, Nairobi, Kenya,*

[f] *Jembi Health Systems, Durban, South Africa,*

[g] *Australian e-Health Research Centre, Brisbane, Australia,*

Abstract

TO DO: add abstract.

Keywords: Analytics-on-FHIR, SQL-on-FHIR, HIE, data platforms, LMICs, digital health

1. Introduction

1.1. The paradox of open for digital vs. data platforms in healthcare

It is a widely held belief that digital technologies have an important role to play in strengthening health systems in low- and middle income countries (LMICs), as exemplified by the WHO global strategy on digital health (**?**). The adoption rate of mobile phones in LMICs has been an important driver in implementing digital health solutions (**?**). Yet, there are many shortcomings and challenges, including the current fragmentation of digital platforms and the lack of clear-cut pathways of scaling up digital health programmes, such that they can support sustainable and equitable change of national health systems in LMICs (**????**).

A commonly used perspective to scrutinize digital health is to consider it as a digital platform (**?**). Digital platforms have disrupted many sectors but have

*Corresponding author
Email address: daniel@kapitan.net (Daniel Kapitan)

just started to make inroads into highly regulated industries such as healthcare (**?**). In this light, the challenges faced by LMICs in establishing national digital health platforms have a lot in common with those faced by high income countries. From a technological perspective, interoperability issues, weak integrations, siloed data repositories and overall lack of openness are often reported as key impediments (**??**). From a societal perspective, issues pertaining to the winner-takes-all nature of digital platforms are hotly debated as many jurisdictions make work to ensure these new digital health platforms indeed serve the common good of achieving universal health coverage (**?**).

Case studies on digital platforms in healthcare point to an emerging pattern where the focus shifts from the digital platform with its defining software and hardware components, to the data as the primary object of interest in and of itself (**??**). This observation ties into with the proposed research agenda by de Reuver et al. to consider data platforms as a phenomenon distinct from digital platforms (**?**). Generally, data platforms inherit the characteristics of digital platforms. From an economic perspective, for example, both types exhibit multi-sided markets. At the same time, data platforms differ as their main offerings revolve around data. From an ecosystem perspective, data platforms have more moderate network effects and are more susceptible to fragmentation and heterogeneity (**?**).

Particularly relevant in the context of open health data platforms (OHDPs), is the conceptualization of openness. The shift in perspective from digital platforms to data platforms coincides with the paradox of open (**?**). Originally, openness of digital platforms focused on open source, open standards and copyrights, which by has been superseded by "… conflicts about privacy, economic value extraction, the emergence of artificial intelligence, and the destabilizing effects of dominant platforms on (democratic) societies. Instead of access to information, the control of personal data has emerged in the age of platforms as the critical contention." (**?**). These conflicts are particularly salient in the healthcare domain, where people are generally willing to share their health data to receive the best care (primary use), while the attitude towards secondary use of health data varies greatly depending on the type and context (**?**). The shift in perspective from digital platforms supporting primary data sharing toward data platforms supporting secondary data sharing is one of the key issues surrounding the polemic of data spaces (**?**) and data solidarity (**????**). Openness is particularly relevant if we are to realize a solidarity-based approach to health data sharing that i) gives people a greater control over their data as active decision makers; ii) ensures that the value of data is harnessed for public good; and iii) moves society towards equity and justice by counteracting dynamics of data extraction (**?**).

1.2. From health information exchanges to health data platforms

This paper is motivated by the conflation of a number of developments relevant to the design and implementation of open, solidarity-based OHDPs in

LMICs. First, the OpenHIE framework (**?**) has been adopted by many sub-Saharan African countries (**?**) as the architectural blueprint for implementing nation-wide health information exchanges (HIE), including Nigeria (**?**), Kenya (**?**) and Tanzania (**?**). These countries have, as a matter of course, extended the framework to include "data & analytics services" as an additional domain. The rationale for this addition is to facilitate secondary reuse of health data for academic research, real-world evidence studies etc. which can be framed within the context of ongoing efforts towards Findable, Accessible, Interoperable and Reusable (FAIR) sharing of health data (**?**). In doing so, however, we have implicitly moved from conceptualizing digital health platforms for primary data sharing (the original OpenHIE specification) to health data platforms for secondary data sharing. This is problematic because the notion of openness, which is assumed to be essential in establishing solidarity-based approaches to data sharing, is inherently different for a data platform compared to a digital platform.

Conceptually, the OpenHIE framework constitutes a framework for an open digital platform. Openness for digital platforms refers to i) the use of open boundary resources, that is, specifications for the various healthcare specific workflows and information standards such as FHIR; and ii) the use of open source components that are available as digital public goods (**?**). If we are to use the OpenHIE framework as an open data platform, we need to extend the standards, technologies and architecture to include functionality for secondary data sharing and reuse. The lack of detailed specifications and consensus of this addition to OpenHIE currently stands in the way of development projects that aim to establish OHDPs in LMICs. The purpose of this paper, therefore, is to specificy how new standards and technologies can be integrated into the OpenHIE architecture such that an open OHDP can be realized that supports the following different types of data sharing.

1.2.1. 1. Sharing of data sets at the most granular (patient) level, persisted as longitudinal records

The Shared Health Record (SHR) in OpenHIE provides an operational, real-time transactional data source which is intended for primary data sharing. The specification explicitly states that the SHR is distinct from a datawarehouse. At the same time, the Health Management Information System (HMIS) component is specified to fulfil functional requirements that are typically provided by datawarehouses.

- **Q1:** How can proven solution designs and modern technologies from data warehousing and engineering be integrated in the OpenHIE specification and how does this relate to the curren specifications of the SHR and HMIS.

1.2.2. 2. Sharing of data products derived from the original data set

OpenHIE specifies that the Health Management Information System (HMIS) component should support a workflow to validate and save aggregate data based on the emerging IHE Aggregate Data Exchange (ADX) standard. This is just

one specific workflow within a larger data analytics value chain that covers i) collection, ii) standardization, iii) cleaning, iv) storage, v) analysis and vi) distribution (**?**). Sharing of benchmarking, and decision-support tools based on various statistical (machine learning) models that have been trained on data sets and/or data product, can be shared for standalone use.

- **Q2:** How can proven solutions designs from the data warehousing and engineering community be included in OpenHIE to support sharing of wider variety of data products.

1.2.3. 3. Sharing of secure computational environments to access and work with the data

Current solution design of centralized data storage is increasingly being challenged. Recent work points to federated learning (FL) (**?**) and privacy- enhancing technologies (PETs) (**??**) as a better long-term solution to secure and equitable sharing of data. Given the resource constrained environments of LMICs, is is particularly challenging to devise a solution that is economically feasible and … [TO DO: add feedback from Mark vdG]

Federated solutions:

- KETOS OMOP-FHIR (**?**)
- Personal Health Train on FHIR (**?**)
- OHDSI analytics (**?**)
- CODA project (**?**)
- GenoMed4All, in evalution phase for -omis (**?**)

Similar approaches taken in the context of data sharing for pandemic response, for example:

- VODAN Africa, based on electronic data capture with templates (**?**)
- Global Data Sharing Initiative (GDSI) (**?**)
- **Q3:** How can a solution be designed that is economically viable, can be deployed in a bottom-up fashion, support data governance principles …

2. Methods

In this paper, we present a design that extends the OpenHIE specification to include the three types of data sharing mentioned above. Using the full-STAC approach (**?**) we combine open standards, open technologies and open architectures into a coherent modular OHDP that can be configured and reused across a variety of use-cases. We employ a formative, naturalistic evaluation to assess the technical risk and efficacy of the design (**?**). Given that it is prohibitively expensive to evaluate the design a real-world setting, we aim to minimize technological risks and maximize the efficacy of the design by considering three examples of health data platforms in LMICs, namely:

1. the OpenHIM platform (https://jembi.gitbook.io/openhim-platform/)

2. the OnaData platform https://ona.io/home/products/ona-data/features/
3. the work conducted at PharmAccess Foundation as part of the MomCare programme (https://health-data-commons.pharmaccess.org).

These three real-world implementations are evaluated along the following dimensions:

- What is the level of openness of the implementation, specifically in terms of the three types of data sharing and platform-to-platform opennenss?
- How are the core maintenance functions of the OHDP implemented, where we particularly focus on functions related to metadata management, data lineage and access control mechanisms?
- Is the solution suitable for downward scalability, where we focus on the computer and storage requirements of the solution and assess the minimal requirements for on-site resources (computers, edge devices)

As part of our design research, we have taken a narrative approach in surveying existing scientific studies on health data platforms, focusing on the seminal reports and subsequently searching forward citations. In addition, we have searched the open source repositories (most notably GitHub) and the online communities (OpenHIE community, FHIR community) to search for relevant open standards, technologies and architectures. This paper should not be considered as a proper systematic review.

The main contributions of this paper are i) description of a framework for the components of the Data & Analysis Services that builds on current best practices from the data engineering community into the OpenHIE framework; ii) evaluation of different implementations and design options for type 1 and 2 data sharing within an extended OpenHIE architecture; and iii) provide open content for the MomCare implementation to facilitate adaptation and deployment. As such, it aims to inform future developments and implementation of open digital health platforms in LMICs.

3. Design

3.1. Open standards: using FHIR as the common data model

The recent convergence to FHIR as the de facto standard for information exchange has fuelled the development of OpenHIE. FHIR is currently used both for routine healthcare settings (**??**) and clinical research settings (**??**) and is increasingly being used in LMICs as well. The guidelines and standards of the African Union explicitly state FHIR is to be used as the messaging standard (**?**). The FHIR-native OpenSRP platform (**?**) has been deployed in 14 countries targeting various patient populations, amongst which a reference implementation of the WHO antenatal and neonatal care guidelines for midwives in Lombok, Indonesia (**??**). In India, FHIR is used as the underlying technology for the open Health Claims Exchange protocol specification, which has been adopted by the Indian government as the standard for e-claims handling (**?**). This range

Table 1: Comparison of OpenEHR, ISO 13606 and FHIR standards

Service	Health data platform service	Preferred standards
Modeling	Modeling and formalization of clinical-domain concepts	OpenEHR and ISO[a] 13606[b]
Persistence	Detailed and multipurpose data persistence	OpenEHR
Exchange	Complex and full-meaning data exchange	ISO 13606 and HL7[c] FHIR[d,e]
Exchange	Simple and agile point-to-point data exchange	HL7 FHIR
Querying	Data query according to complex semantic restrictions	OpenEHR
Implementation	Design of data entry components in EHR[f]	OpenEHR
Implementation	EHR repository for clinical decision support processes	OpenEHR
Implementation	EHR repository for populating RWD[g] repositories	OpenEHR
Implementation	Semantically interoperable platform for heterogeneous source EHRs	ISO 13606 and HL7 FHIR[b]
Implementation	Semantically interoperable exchange between EHR applications	HL7 FHIR
Implementation	Semantically interoperable exchange between EHR and RWD repositories	HL7 FHIR

of utilizations showcase the standards' widespread applicability. The proceedings of the OpenHIE conference 2023 attest to the fact that FHIR and open source technologies are embraced as critical enablers in implementing health information exchanges in LMICs (**?**).

Various studies have investigated the merits of FHIR and its performance vis-a-vis other healthcare standards. Comparisons between OpenEHR, ISO 13606, OMOP and FHIR have been made (**??????**). A study involving 10 experts comparing OpenEHR, ISO 13606 and FHIR concluded that i) these three standards are functionally and technically compatible, and therefore can be used side by side; and that ii) each of these standards have their strengths and limitations that correlate with their intended use as summarized in the Table 1.

For an infectious diseases dataset with a limited scope, OpenEHR, OMOP and FHIR have been compared and found all to be equally suitable (**?**). Comparing OMOP and FHIR, the latter has been found to support more granular mappings required for analytics and was therefore chosen as the standard for the CODA project (**?**).

Although FHIR was originally designed only for exchange between systems, we propose to use it as the common data model for the design presented here for the following reasons:

- Industry adoption has significantly increased, as exemplified by FHIR-based offering by major cloud providers such as Google, Azure and AWS. Also, Africa CDC has explicitly chosen FHIR as the preferred standard;
- The widespread availibility of the Bulk FHIR API (**??**) enables bulk, file-based batchwise processing for analytics using the lakehouse architecture as detailed in the next section;
- The concept of FHIR Profiles allow localisation to tailor the standard to a specific use case. A profile defines rules, extensions, and constraints for a resource. We posit that the possible penalty of this flexibility, namely

having to manage different FHIR versions and/or profiles, is less of an issue in the context of LMICs where first priority is to exchange datasets such as the International Patient Summary (IPS) that are less complex compared to the requirements for high income countries;
- Being based on webstandards, the FHIR standard lends itself best for further separation of concerns as envisioned by the composable data stack. This is an important enabler for the downward scalability of the solution;
- With its inherent, graph-like nature, FHIR can be readily incorporated into the principles of FAIR data sharing, where FHIR-based data repositories can be integrated in an overarching netwerk of FAIR data stations (**??**).

TO DO: re-use these elements from table 1 into this section:

- Given that the Bulk FHIR API has by now been incorporated in all major FHIR implementations (**??**), the FHIR standard can be readily used to bridge the gap between primary (transactional) and secondary (analytics) data sharing.
- **SQL-on-FHIR specification** (**?**): provides a standardised approach to make FHIR work well with familiar and efficient SQL engines that are most commonly used in analytical workflows. Builds on FHIRPath (**?**) expressions in a logical structure to specify things like column names and unnested items. Implementations of this approach are available or forthcoming, including open source implementations such as Pathling (**?**) and commercial offerings like Aidbox (**?**). Modern data platforms take this Within the HL7 ecosystem much progress has been made to support "analytics-on-FHIR" with standards including FHIRPath and the new SQL-on-FHIR specification (**?**). Given the use of FHIR as the common data model (the rationale of which will be described later), we formulate the second design question:

Possible risks pertaining to the use of FHIR as the common data model, most notably the possible incompatibilities and/or high costs of maintenance in supporting different versions, will be addressed in the Discussion.

3.2. Open architecture: extending OpenHIE with a composable data stack

3.2.1. Evolving the data lakehouse to a composable data stack

Data management and analytics platforms have undergone significant changes since the first generation of data warehouses were introduced. Recent studies have shown that the current practice has converged towards the lakehouse as one of the most commonly used solution designs (**???**). Lakehouses typically have a zonal architecture that follow the Extract-Load-Transform pattern (ELT) where data is ingested from the source systems in bulk (E), delivered to storage with aligned schemas (L) and transformed into a format ready for analysis (T) (**?**). The discerning characteristic of the lakehouse architecture is its foundation on low-cost and directly-accessible storage that also provides traditional database

management and performance features such as ACID transactions, data versioning, auditing, indexing, caching, and query optimization (**?**). Lakehouses thus combine the key benefits of data lakes and data warehouses: low-cost storage in an open format accessible by a variety of systems from the former, and powerful management and optimization features from the latter.

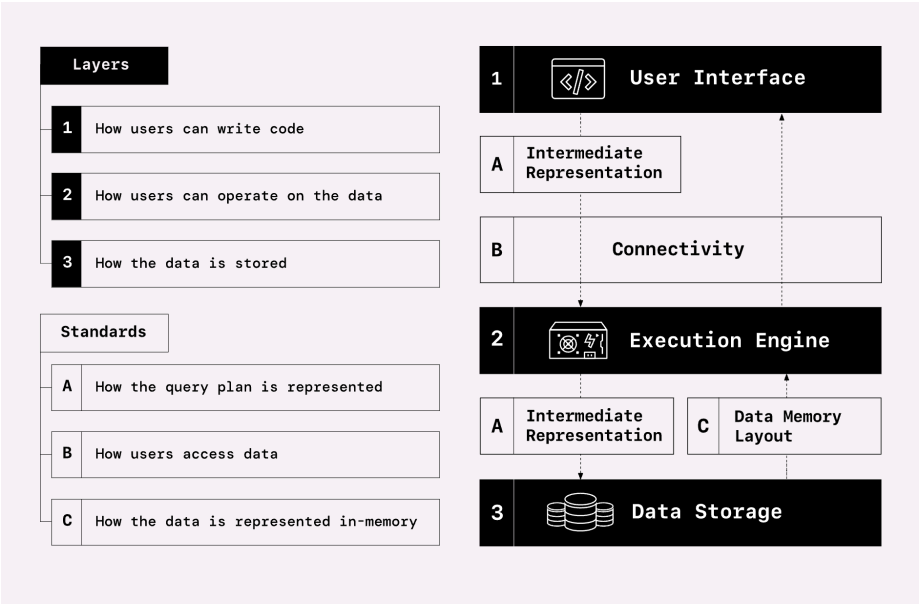
With respect to current implementations of lakehouse data platforms, we observe a proliferation of tools with as yet limited standards to improve technical interoperability. In the analysis of Pedreira et al. (**?**) the requirement for specialization in data management systems has evolved faster than our software development practices. This situation has created a siloed landscape composed of hundreds of products developed and maintained as monoliths, with limited reuse between systems. It has also affected the end users, who are often required to learn the idiosyncrasies of dozens of incompatible SQL and non-SQL API dialects, and settle for systems with incomplete functionality and inconsistent semantics. To remedy this, Pedreira et al. call to (re-)design and implement modern data platforms in terms of a 'composable data stack' as a means to decrease development and maintenance cost and pick-up the speed of innovation.

While the lakehouse architecture separates the concerns of compute and storage, the composable data stack takes the separation of concerns is taken one step further. A composable data system (Figure 2), not only separates the storage (layer 3) and execution (layer 2), but also separates the user interface (layer 1) from the execution engine by introducing standards including Substrait for Intermediate Representation (standard A, IR) and Apache Arrow for data connectivity and data memory layout (standards B and C, respectively). The first generation of open source components are already available. For example, the Ibis user interface is currently sufficiently mature to offer a standardized dataframe interface to 19 different execution engines (**?**).

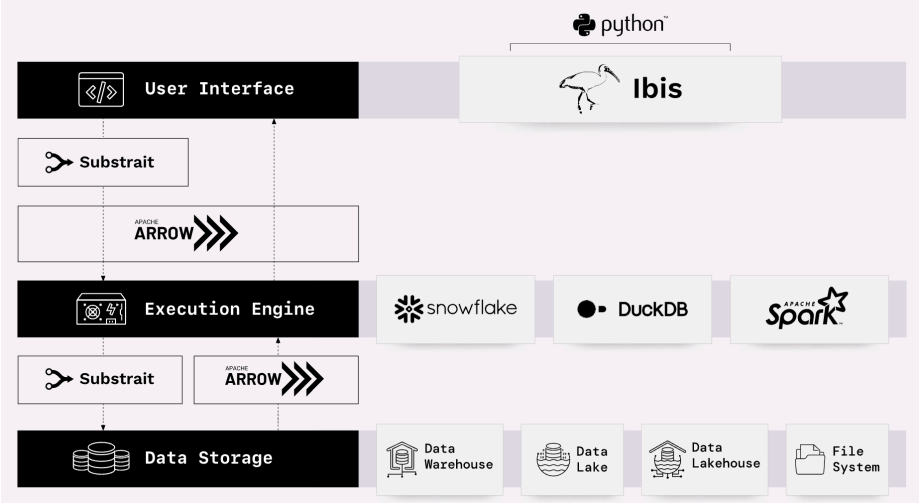
3.2.2. SQL-on-FHIR v2 as an intermediate representation for FHIR data in tabular format

The premise of separating the user interface from the execution engine is directly related to the key objective of the SQL-on-FHIR project (https://build.fhir.org/ig/FHIR/sql-on-fhir-v2/), namely to make large-scale analysis of FHIR data accessible to a larger audience, portable between systems and to make FHIR data work well with the best available analytic tools, regardless of the technology stack. However, to use FHIR effectively analysts require a thorough understanding of the specification as FHIR is represented as a graph of resources, with detailed semantics defined for references between resources, data types, terminology, extensions, and many other aspects of the specification. Most analytic and machine learning use cases require the preparation of FHIR data using transformations and tabular projections from its original form. The task of authoring these transformations and projections is not trivial and there is currently no standard mechanisms to support reuse.

The solution of the SQL-on-FHIR project is to provide a specification for



(a) a)



(a) b)

Figure 2: Schematic overview of the composable data stack showing a) the overall architecture and b) examples of implementations. Images taken from https://voltrondata.com/codex.

defining tabular, use case-specific views of FHIR data. The view definition and the execution of the view are separated, in such a way that the definition is portable across systems while the execution engine (called runners) are system-specific tools or libraries that apply view definitions to the underlying data layer, optionally making use of annotations to optimize performance.

3.2.3. Extending OpenHIE with a FHIR-based composable data stack

We propose to extend the OpenHIE architecture with a "Data and Analytics Services" domain with different service layers by synthesizing the current best practices of the a lakehouse architecture of (**???**) and the composable data stack (**?**) (Figure 3, Table 2). These 5 services are considered the core of the FHIR data platform, and although in practice often a downstream dashboarding or visualization component is used, that component is not the main focus of our analysis. Rather, we aim to elucidate and conceptualize the core "FHIR Data Lakehouse".

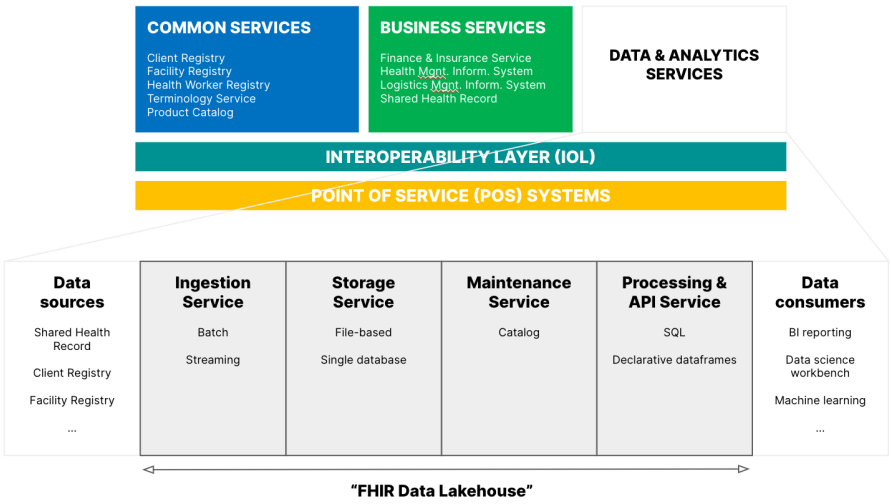


Figure 3: Proposed extension of the OpenHIE architecture that includes "Data and Analytics Services" as an additional service domain.

Table 2: Definition of Data and Analysis Services

Service	Functional requirements
Ingestion	• Bulk • Streaming
Storage	• File-based blob storage • Database optimized for online analytical processing (OLAP)

Service	Functional requirements
Maintenance	• SQL-on-FHIR View defintions • Catalog and other maintenance-related functions as defined by Hai et al.
Processing & API	• SQL-on-FHIR Runner • Execution engine on tabular data as defined in composable data stack • Capability to participate as a node in federated learning / MPC network • Read-only access to storage
Data Consumer	• SQL interactive development environment (IDE) • Interactive notebook computing environment (**?**) • BI reporting, dashboarding and data visualization

3.2.3.1. Ingestion.

Default workflow is extraction of data from SHR using Bulk FHIR API. Data contains metadata (incl. FHIR versions) and fully qualified semantics, for example, coding systems. Despite this, metadata extraction and metadata modeling is still required to meet the FAIR requirements. Issues that need to be solved by these services:

- To prepare for future updates of FHIR versions
- Implement late-binding principle of having increasingly more specific FHIR profiles as bulk FHIR data propagates through lakehouse

3.2.3.2. Storage.

- File-based:
  - from ndjson to parquet
  - possibly used delta lake for time versioning
  - separation of storage from compute not only for benefits of lower TCO, but also be ready for federated learning and MPC in future
- OLAP DBMS
  - Often columnar, like Clickhouse and BigQuery
  -

3.2.3.3. Query & Processing.

- fit in structure of OpenHIE specification
- check which workflows are related to analytics
- Hai calls this 'Maintenance'

3.2.3.4. Maintenance.

- SQL-on-FHIR Views provide new standard to support mADX aggregate reporting !! We need to stress this, because this is an existing OpenHIE workflow

- Maintenance-related functions remain the same
- NB: orchestration falls under data provenance

3.2.3.5. Data consumers.

- Many tools, often focus on creating information dashboards and visualizations
- Compatibility with processing & API: which query languages and interfaces are supported. Some dialect of SQL, some dialect of NoSQL, dataframe API, all of the above?

3.3. Open technologies: deploying Instant OpenHIE with digital public goods

Today, many components of the OpenHIE specification are now available as a digital public goods, as listed in Table 3. Typically, these open source components are intended to support deployments in small countries (population up to 10 million) or large NGOs out of the box, and should provide a stepping stone for customized deployments in medium-sized countries (population around 40 million).[1] To further ease the development, configuration and deployment of health information exchanges, the concept of 'Instant OpenHIE' has been championed to (i) allow implementers to engage with a preconfigured health information exchange solution and running tools (based on the architecture) and test their applicability and functionality with a real health context problem; and (ii) have a packaged reference version of the OpenHIE architecture that is comprised of a set of reference technologies and other appropriate tools that form the building blocks of the health information exchange that can be configured and extended to support particular use cases (**?**). Besides the core functional components of the OpenHIE architecture, the Instant OpenHIE toolkit allows packaging and integration of generic components such as Identity and Access Managment (IAM) and a reverse proxy gateway. In the following, we will evaluatie three of such configurations, with the aim to conceptualize and evaluate the proposed Data and Analytics Services domain of of the OpenHIE architecture.

Table 3: Overview of current open source implementations of components that fit in the OpenHIE specification. The category Analytics Services is not a part of the original OpenHIE and is discussed in this paper. Point-of-Service systems are excluded for brevity.List compiled using OpenHIE Reference Technologies, Global Goods for Digital Health, Digital Public Goods Alliance and search of open source code repositories. A systematic review of such digital public goods is beyond the scope of this document.

Category	Component	Digital Public Good
Interoperability layer (IOL)	IOL	OpenHIM

[1] Although the OpenHIE specification does not include details on dimensioning, these are typically the requirements that are used within the community. See OpenHIE Community Wiki.

Category	Component	Digital Public Good
		mHero
		OpenFN
Registry Services	Client Registry (CR)	SanteMPI
		JeMPI
		OpenCR
		OpenCRVS
	Facility Registry (FR)	Global Open Facility Registry (GOFR)
		GeoPrism Registry (GPR)
		Resource Map
		Healthsite.io
		GeoPrism Registry
		DHIS2
	Health Worker Registry (HWR)	iHRIS
	Terminology Service (TS)	OCL Terminology Service
	Product Catalog (PC)	PCMT
Business Domain Services	Shared Health Record (SHR)	HAPI FHIR
		Fhirbase
		FHIR Server for Azure
		I-TECH-UW SHR
	Health Management Information System (HMIS)[2]	DHIS2
	Finance and Insurance Service (FIS)	OpenIMIS
	Logistics Management Information System (LMIS)	OpenLMIS
		OpenBoxes

[2]Note that there is often a confusion on the acronym HMIS. Strictly speaking, the OpenHIE specification uses HMIS to refer to a Health Management Information System that is part of the Business Domain Services. Sometimes HMIS is used to refer to a Hospital Management Systems in the Point-of-Service domain, synonymous with an Electronic Medical Record (EMR) system.

Category	Component	Digital Public Good
Generic	Identity and Access Management (IAM)	Keycloak
	Gateway & proxy	FHIR Information Gateway
	Admin dashboard for SHR	OpenSRP FHIR Web
	Configuration and deployment	Instant OpenHIE

3.4. Open content

-

4. Evaluation

4.1. Jembi OpenHIM platform

To evaluate the extended OpenHIE architecture described above, we first consider the OpenHIM Platform. The Open Health Information Mediator (OpenHIM, http://openhim.org/)) component is the reference implementation of the Interoperability Layer (IOL) as defined in the OpenHIE specification. The most current version (8.4.2 at the time of writing) provides all the core functions including central point of access for the services of the HIE; routing functions; central logging for auditing and debugging purposes; and orchestration/mediation mechanisms to co-ordinate requests. By extension, the OpenHIM Platform (https://jembi.gitbook.io/openhim-platform) is a reference implementation of a set of Instant OpenHIE configurations, refered to as 'recipes' in the documentation. In the following we will evaluate the recipe for "a central data repository with a data warehouse" that provides "A FHIR-based Shared Health record linked to a Master Patient Index (MPI) for linking and mathing patient demographics and a default reporting pipeline to transform and visualise FHIR data" (https://jembi.gitbook.io/openhim-platform/recipes/central-data-repository-with-data-warehousing).

Figure 4 shows a schematic overview of two data stacks that are supported in the OpenHIM platform. The Shared Health Record (SHR, implemented with HAPI FHIR server) and the Client Registry (CR, implemented with JeMPI server) are the sources that store clinical FHIR data and patient demographic data, respectively. The default data stack is based on streaming ingestion using Kafka into a Clickhouse database. As part of the ingestion, incoming FHIR bundles that contain multiple FHIR resources are unbundled in separate topics using a generic Kafka utility component. Subsequently, each FHIR resource topic is flatted with Kafka mappers that use FHIRPath. Superset is used as the tool for consuming the data to create dashboard visualizations.

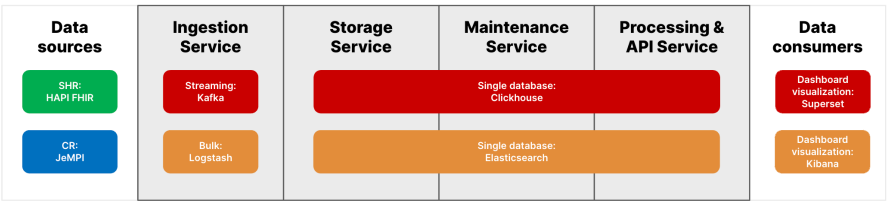


Figure 4: Overview of the default data stack of the OpenHIM Platform. The default stack (top, red) consists of Kafka, Clickhouse and Superset. An alternative solution based on the ELK stack is also supported (bottom, orange), consisting of Elasticsearch, Logstash and Kibana.

The OpenHIM platform also support data and analytics based on the ELK stack, where data is ingested in bulk using Logstash, stored in Elasticsearch and made available for consumption in Kibana. Also here, the incoming FHIR bundles are unbundled in Logstash into separate FHIR resources. However, given that Elasticsearch is a document-based search engine, the FHIR resources are stored as-is with no flattening. Exploring and analysing the data requires writing queries in Elasticsearch Query Language (ES|QL), either through the query interface of Elasticsearch or using Kibana.

Evaluating these two data stacks, we see the following:

- Pattern of flattening FHIR resources with FHIRPath expressions is very close to the idea of SQL-on-FHIR. Although it doesn't adhere to this new standard in the strict sense, the philosophy of generating tabular views is the same
- When using the ELK stack, flattening is done at the end. Implementations of FHIRPath support Elasticsearch as an execution engine, also here
- Main limitations: both Clickhouse en Elasticsearch don't follow decomposition of storage, compute and UI. Therefore, downward scaleability is limited.

4.2. ONA OpenSRP 2

Continuing our evaluation of the extended OpenHIE architecture, we can see a different flavor in the implementation driven by Ona. Ona is a social enterprise that has pioneered the adoption of FHIR data standard via the development of OpenSRP2, a FHIR-based data collection app built using Google's FHIR SDK and focused on enabling offline-first workflows for community-based care. OpenSRP 2 is a complete rewrite of the original OpenSRP application, a global public good maintained by Ona and deployed in XX countries worldwide.

OpenSRP2 applications are currently implemented in the field in three countries (Uganda, Liberia, and Madagascar) in collaboration with local Ministries of Health and with international donors such as UNICEF, supporting a variety of different workflows including antenatal care (ANC), postnatal Care (PNC),

immunization, and last-mile logistics. Besides the OpenSRP Android application and HAPI-FHIR backend, in each of its projects Ona also implements a companion set of tools that support analytics and various reporting needs.

4.2.1. Requirements for data sharing

Based on years of work in global health, Ona has learned that the data stack implemented to support a national-scale implementation of its FHIR-based application responds to the following requirements.

Table 4: Requirements and rationale for open health data platform developed and used by ONA, based on OpenSRP 2 (https://opensrp.io/).

Requirement	Rationale
Ingest data from multiple sources, both FHIR and non-FHIR based.	While most health record data can be collected and aggregated in FHIR, Ministries of Health rely on other data sources to govern their operations. For example, operationalizing an immunization campaign usually includes tracking against specific targets for locations to be visited on specific days and number of children to immunize per day. Such targets are often stored in spreadsheets or other applications where the data is not FHIR.
Ingest data in batches.	Most data ingestion can happen in batches, since Ona's applications are deployed in hard to reach areas where connectivity is an issue. Data ingestion closer to real-time can be relevant for disaster-response and other time-sensitive applications, but this is not a priority.
Support national-scale data volumes.	A data store that can grow from dozens to thousands of devices and where data can be aggregated up to the national level, matching the scale of implementation of data collection applications in the field.
Pre-compute complex business metrics.	Reporting on health systems requires pre-computing complex metrics and often performing cohort analyses to map trends in service provision. For example, understanding quality of care for children requires computing metrics such as the percentage of children fully immunized on schedule (i.e. children 6-59 months that have received the set of vaccines required by the Ministry of Health, and have received each of those vaccines within the expected age-window). For Business Intelligence applications, calculating such a vital metric cannot be performed at run time, to avoid long and expensive queries.

Requirement	Rationale
Outbound integrations.	While aggregated data and reports should be accessible by other applications such as BI platforms via pulls, there should be an easy integration framework to push data to other applications used by the Ministry of Health for other purposes, such as DHIS2 for health systems management or RapidPro for communications with program beneficiaries.
Open source and easily deployable in-country.	Given the extremely sensitive nature of health data, it is paramount for governments to have the flexibility to deploy the stack in various different environments, both on premise and in private clouds.

The architecture Learning from experience in the field and internal research and development, Ona has developed preferences for a specific data stack responding to the aforementioned requirements.

[[graphic]]

Core toolings in the stack include: Data ingestion with Airbyte. Ona uses Airbyte as the primary data ingestion tool, leveraging the wide array of connectors that come standard with the application as well as a dedicated suite of connectors developed internally by Ona, including HAPI FHIR, RapidPro, Ona Data, Kobo Toolbox and others.
Data storage with Clickhouse. While different health projects have varying requirements, Ona has found success in using Clickhouse as the main analytics data store in its most recent implementations. Clickhouse supports the scale required for analytics at a national level, as well as the speed that enables cross-application integrations and more real-time analytics. For example, in Madagascar Ona uses its reporting suite to identify facilities with stock in need of maintenance and can trigger the scheduling of a maintenance visit ad hoc.
Data transformation with dbt. Following global best practice, Ona leverages dbt to segregate the data warehouse in different levels (staging, marts, metrics), as well as pre-computing complex indicators for ease of reporting and for transmissions into other systems. For example, in Liberia Ona implements OpenSRP at community health worker level, but can aggregate immunization data at facility level in the data warehouse and then push quarterly summary metrics to DHIS2. No recommendation on reporting / BI tooling. Ona recognizes that business users have their own strong preferences for BI tooling, and some already have licenses for specific software, so the architecture is flexible to provide easy connections to different BI tools.

Evaluation

Evaluating the data stack, we see the following: Use of generic best-of-breed tooling. Ona focused on utilizing Open HIE tools that are widely adopted out-

side of the global health and development sectors. This approach aims to provide assurance on two main fronts, the ability to handle performance at scale and the long term dependability of the tools, rather than relying on smaller projects with uncertain long term funding or unproven implementations. Columnar data warehouse for analytics. The scale of Ona's project requires the implementation of a dedicated database for analytics. While original data can still be stored as parquet or other file system, being able to ingest it into a relational data store allows to create well defined indicators. Using clickhouse as a tool helps and combine the need accuracy with the speed of reporting as new data is ingested. Strong emphasis on SQL. While Ona has tested and experimented with FHIR-specific tooling, such as the definition of data projections using sql-on-fhir, Ona found that relying on sql for coding business logic remained the faster and most scalable approach.

In summary, for Ona building analytics with FHIR data looks similar to building analytics with any other type of data. While FHIR provides a clear and standard data model, managing information for most health systems requires custom integration of data between different sources, as well as computing indicators using business logic specific to the needs of the local users. Building upon well established best-of-breed tools allows Ona to implement FHIR applications at scale and provide trusted analytics on top.

4.3. PharmAccess demonstrator Momcare programme

MomCare was launched in Kenya (**??**) and Tanzania (**??**) in 2017 and 2019 respectively, with the objective to improve health outcomes for maternal and antenatal care. MomCare distinguishes two user groups: mothers are supported during their pregnancy through reminders and surveys, using SMS as the digital mode of engagement. Health workers are equipped with an Android-based application, in which visits, care activities and clinical observations are recorded. Reimbursements of the maternal clinic are based on the data captured with SMS and the app, thereby creating a conditional payment scheme, where providers are partially reimbursed up-front for a fixed bundle of activities, supplemented by bonus payments based on a predefined set of care activities.

In its original form, the MomCare programme used closed digital platforms. In Kenya, M-TIBA is the primary digital platform, on top of which a relatively lightweight custom app has been built as the engagement layer for the health workers (**?**). M-TIBA provides data access through its data warehouse platform for the MomCare programme, however, this is not a standardized, general purpose API. In the case of Tanzania, a stand-alone custom app is used which does not provide an interface of any kind for interacting with the platform (**?**). Given these constraints, the first iteration of the MomCare programme used a custom-built data warehouse environment as its main data platform, on which data extractions, transformations and analysis are performed to generate the operational reports. Feedback reports for the health workers, in the form of operational dashboards, are made accessible through the app. Similar reports are provided to the back-office for the periodic reimbursement to the clinics.

Clearly, a more open and scaleable platform was required if MomCare was to be implemented in more regions. This need led to a redesign of the underlying technical infrastructure of the MomCare project. The objectives of this work were in fact to demonstrate a solution design that could support the first three types of data sharing. First, to investigate the viability of using FHIR for bulk data sharing, MomCare Tanzania was used a testbed to assess the complexity and effort required to implement the facade pattern to integrate the legacy system into the FHIR data standard. Using the longitudinal dataset from approximately 28 thousand patient records, FHIR transformations script were developed and deployed using the mediator function of the IOL. The data was transformed into 10 FHIR v4 resources and the conceptual data model of the existing MomCare app could readily be transformed into the FHIR standard using SQL and validated with a Python library (**?**). The largest challenge during the transformation process pertained to the absence of unique business identifiers for patients and healthcare organizations. For patients, either the mobile phone number or the healthcare insurance number was taken, depending on availability. A combination of name, address and latitude/longitude coordinates were used to uniquely identify organizations and locations, as Tanzania does not have a system in place for this purpose.

The second objective was to reproduce existing analytic reports, using the bulk FHIR data format as input. Here, the focus was to standardize the logic required for producing metrics and reports. The transformed and validated data is uploaded into the FHIR server on a daily basis using an automated cloud function. Analysis of bulk data was done by directly reading the standard newline delimited JSON into the Python pandas data analysis library. Cross checking the output with queries on the original data confirmed that the whole data pipeline produced consistent results. For example, the report of the antenatal coverage metric (number of pregnancies with four or more visits) could be reproduced per patient journey and aggregated (per year, per organization etc.) as required for the MomCare reports.

TO DO: explain logic of patient-timeline table. Write standard transformation to go from FHIR resources to this standard table. On top of that the actual metrics and reporting. Explain serverless: we wanted to get rid of resource-heavy data visualization tools. This led to the idea of serverless: using duckdb-wasm and pipelines of cloud functions.

The third objective was to run a technical feasibility test for federated analytics. Using the MPC platform of Roseman Labs, we managed to do aggregations in the blind … TO DO: explain that we managed to reproduce the reports we generated in the clear, but then in the blind. Note, however, that in the remainder we will focus on first two types of data sharing.

Based on these experiments, we arrived at the following design for the data & analytics services

- Use ‘serverless’ file-based storage: bulk copy of data as-is in parquet

  - Tension: how to manage change data capture
  - Tension: how to manage access rights
- Use SQL-on-FHIR-v2 to create tabular views.
  - Example: patient timeline
  - TO DO: rewrite patient timeline queries with SQL-on-FHIR-v2 and run it with Pathling
- Use semantic modeling layer to define metrics
  - There are many options: dbt, cube.dev
  - Fulfills same function as ADX/mADX IHE profile in OpenHIE specification
  - Tension: going from patient-timeline to reported metrics still isn't standardized. This is where Ibis/Substrait comes in. Substrait as IR for cross-language serialization for relational algebra. Can be executed on different backends. Write once, run on different engines.
- Distribute and publish reports on resource-constrained devices
  - duckdb
  - sveltekit

TO DO: Add diagram

4.3.1. Level of opennness

TO DO: evaluate openness of OpenHIM platform

4.3.2. Core maintenance functions

TO DO: evaluate maintenance functions

4.3.3. Downward scaleability

TO DO: evaluate downward scalability

5. Discussion

5.1. Openness of data platforms

We specifically address the notion of openness of OHDPs in LMICs in terms of the design-related questions put forward by de Reuver at al.11:

- Object of openness: what data-related resources should data platforms make available when opening up (e.g. data, data products, datadriven insights, analytics modules)? Which user groups derive value from accessing data-related resources from data platforms (e.g. data providers, data users, intermediaries, developers)?
- Unit of analysis: what is platform-to-platform openness in the context of data platforms, given the expectation that different OHDPs will emerge at various aggregation levels? How do we distinguish meta-platforms, forking, and platform interoperability?
- Risk of openness: What are the novel (negative) implications of opening up data platforms? How can reflexivity in design help providers to resolve the negative implications of openness?

- Answers/insights to above:
  - Openness of standardized view on FHIR data and cross-language serialization of relational algebra makes it possible to fully standardize the workflow from start to finish
  - Platform-to-platform: MPC
  - Risk of openness: difficult to answer …

5.2. Comparison with HMIS component

- Workflow requirements: Report aggregate data (link): receiver is HMIS, mADX
- Functional requirements: https://guides.ohie.org/arch-spec/openhie-component-specifications-1/openhie-health-management-information-system-hmis

Requirements are similar, but implementation differs: Datamodel is non-FHIR, focused on DataValue, which conceptually equates to FHIR Measure

5.3. The need to a semantic layer?

- FHIR and FAIR
  - How does FHIR relate to approaches taken by the FAIR community, which tend to take more an approach of using knowledge graphs. For example, VODAN Africa (**??**).
  - FAIR principles vs FHIR graph: is FHIR a FAIR Data Object
- Since we use FHIR, we don't need a semantic layer because that is already provided
- We do need different semantic layer, namely with metrics. Explain different types of semantics.
  - The metrics layer same function as CQL. Discuss CQL vs generic metrics layer.

5.4. Attribute-based access control

- TO DO: if you have generated flattened SQL tables, how are you going to manage security?
- Cerbos, attribute based on lineage or anonymized tables
- Catalogs solve this: Tabular.io, Google BigLake. What is open source option?

5.5. Federated learning and multiparty computation

- data stations??!

6. Abbreviations

ACID	Atomicity, Consistency, Isolation, and Durability

CLI	Command-line Interface
CR	Client Registry
ELK	Elasticsearch, Logstach and Kibana stack
ELT	Extract, Load and Transform
FAIR	Findable, Accessible, Interoperable and Reusable
FHIR	Fast Healthcare Interoperability Resources
FL	Federated learning
OHDP	Health data platform, explicitly differentiated from health digital platform
HIE	Health Information Exchange
IR	Intermediate Representation
LMIC	Low- and middle income countries
MPC	Multiparty Computation
PET	Privacy-enhancing technologies
SHR	Shared Health Record

References