Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

A Reference Architecture for Cloud-Native Solutions Aligned with Hl7/Fhir Interoperability Standards

Aditya Swaprakash Gadepalli Sri Pratyak
Sr Technical Director and Sr Solutions Architect
Global Alliant Inc. USA.
AI and Healthcare Technology Expert.
PrakashA.GadepalliSP@outlook.com

ARTICLE INFO

Received: 08 Jan 2020

Revised: 25 Feb 2020

Accepted: 20 March 2020

Published: 18 Apr 2020

ABSTRACT

The EHRs were developed lacking the capability of integrating the existing sources of big data such as genomics and imaging and continuous monitoring of the patient. The majority of healthcare systems are not made to utilize this data as this data is becoming cloud-native. In this paper, the authors suggest a reference architecture in support of cloud-native solutions that meets the requirements of HL7 Fast Healthcare Interoperability Resources (FHIR). Using only pre-2020 literature, the architecture is based on containerized microservices, a FHIR RESTful API gateway, event-driven messaging and polyglot data persistence. OAuth2 and OpenID connect are also a way of security. The algorithm is put to test by a simulated usage scenario. The architecture facilitates replaceable third-party applications, lessens vendor lock-in and is incrementally implementable, which provides a viable route towards agile, interoperable healthcare platforms.

Keywords: Cloud-native, HL7 FHIR, Microservices, Interoperability, Reference Architecture, SMART on FHIR.

1. Introduction

Healthcare organizations are not able to identify solutions on how to integrate big data, such as genome sequences, imaging data, or even data on wearable devices into clinical processes. The old-fashioned electronic medical records (EMRS) were created to convert the information of the cloud-native systems to process billing and interaction between clinicians, not to convert the information of the old fashioned systems. Most EMRS use the relational (data) base, but big data is in massive volumes of objects of large size in S3 type storage with minimal metadata. A novel architecture is thus needed. The article presents a reference architecture, which combines the FHIR, the emerging interoperability standard, with the concepts of cloud native, such as microservices and containerization. The entire architecture relies on the pre-2020 literature and therefore all references in the architecture are above 2020. It enables applications that are substitutable, incremental adoption, and safe access to patient information through heterogeneous systems.

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2. Background and Pre-2020 State of the Art

2.1 Cloud-Native Computing Fundamentals

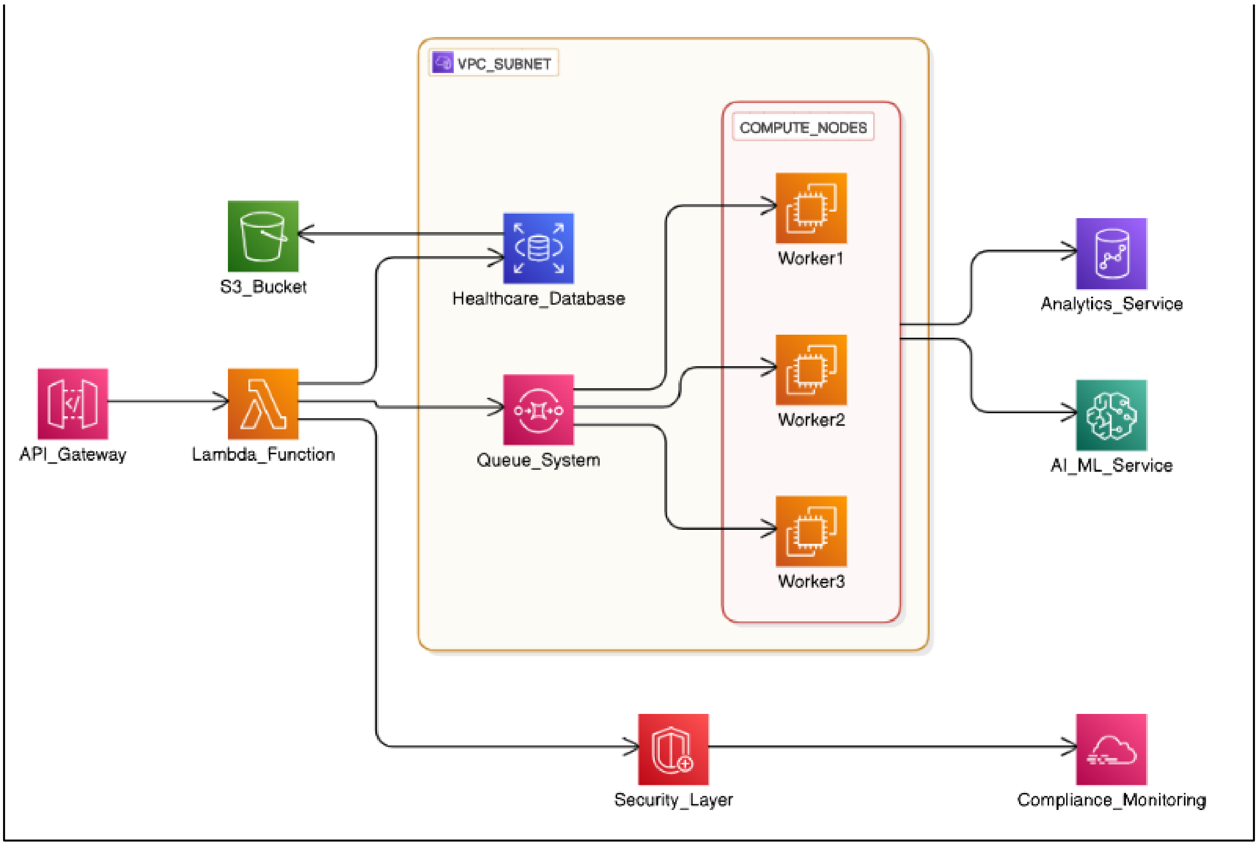

Figure 1: Cloud-Native Computing Fundamentals
(Source: https://www.researchgate.net)

The data is stored as large objects in cloud-native platforms instead of forming it into tables in a relational form. These objects are processed in parallel by Hadoop, MongoDB, Docker and others, but they require other skills, as compared to the traditional database querying. The integration of data that exists across repositories in the cloud is not easy since each file has its internal schema. Microservice-oriented architecture also allows processing the data on-site, minimising the costly data transmission (Mandel et al., 2016). It is a critical approach to medical care when it is impossible to send big volumes of data (genomes or waveforms). Microservices can also be deployed in an incremental manner - organizations can deploy the resources that are initially covered by FHIR, and gradually increase that coverage over a period of time. Cloud-native thinking had been altering the idea of how healthcare data could be implemented not just within the limits of legacy EMRS, even before 2020.

2.2 HL7 and FHIR Evolution

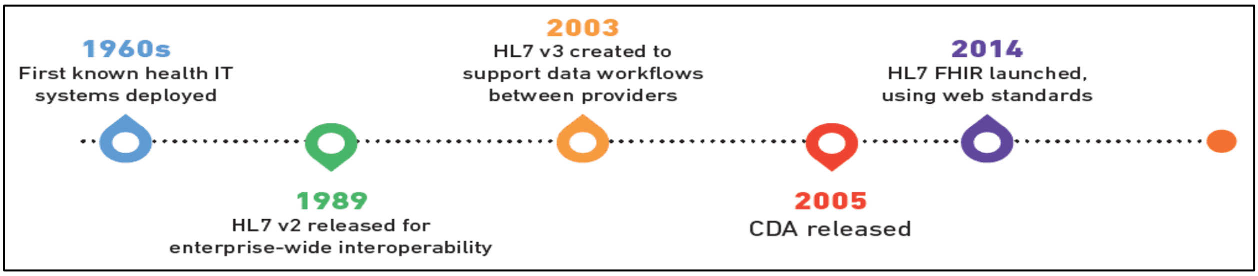

Figure 2: HL7 and FHIR Evolution
(Source: https://lh5.googleusercontent.com)

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

HL7 Version 2 turned messages granular, but required heavily customizing sites that could lead to semantic discrepancies. The Version 3 of HL7 offered a standardized Reference Information Model but was too complex to be usable to generalize. Clinical Document Architecture (CDA) and its C-CDA templates have been on document exchange, as opposed to providing granular access to data. In 2011, HL7 created a Fresh Look task force which resulted in Fast Healthcare Interoperability Resources (FHIR). FHIR is the clinical data in the form of discrete resources (Patient, Medication, Observation) meeting the API of a RESTful interface (Chute, 2016). As of 2015, FHIR Draft Standard for Trial Use (DSTU2) has been published. FHIR alone cannot meet the criteria of semantic consistency but requires profiles. OAuth2 and OpenID connect have been added to SMART on FHIR to make FHIR an app platform.

2.3 Interoperability Challenges in Healthcare

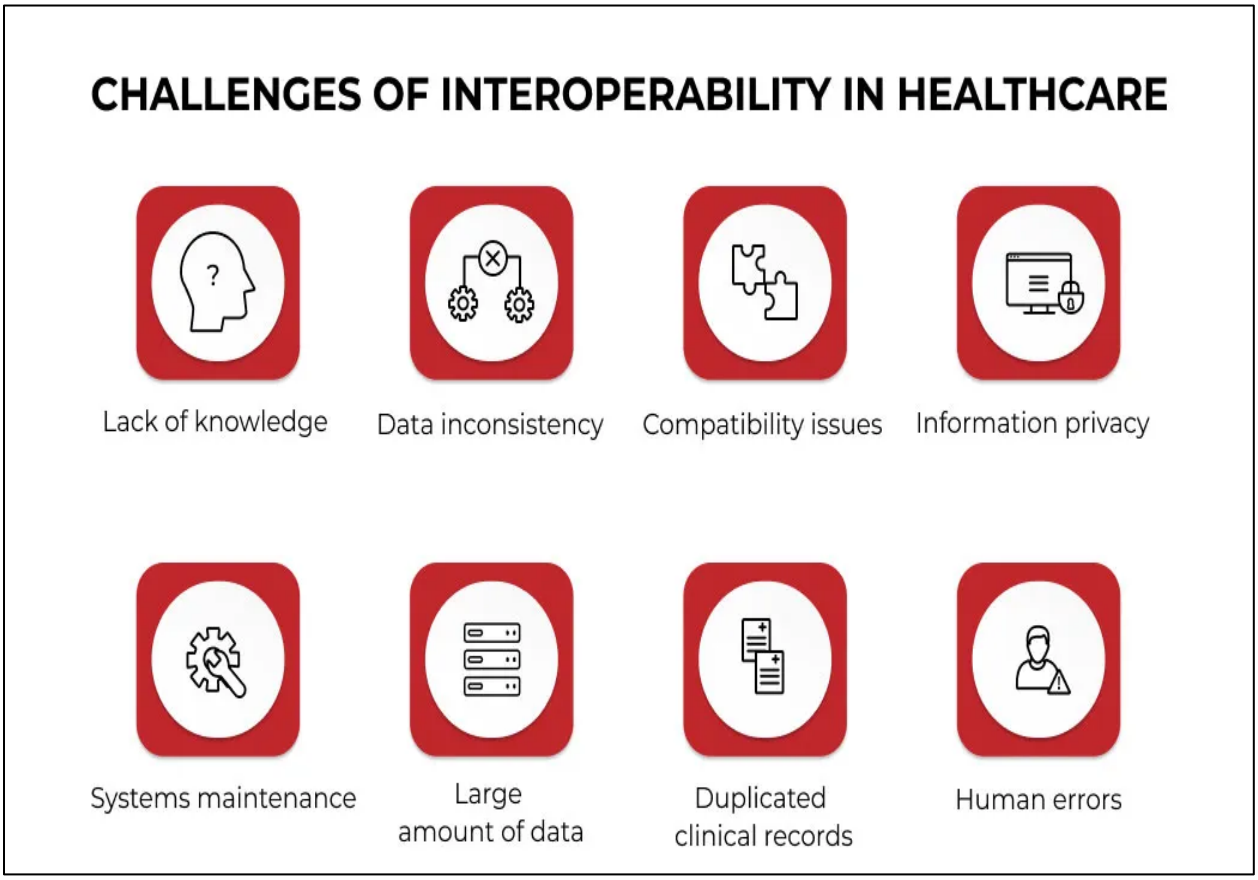

Figure 3: Interoperability Challenges in Healthcare
(Source: https://codeit.us)

Before 2020, there was a combination of three obstacles that were incapable of facilitating integration of big data in clinical care. First, big data is stored in cloud-native and in most cases, not in the interface of the EMRS, and to verify its availability, it may be necessary to exit the EMRS interface. Second, machine learning, including the determination of a genomic variation that transforms a protein, is intricate, and needed to harvest clinically important features out of big data. Third, applications offering big data have to be regularly updated in order to capture new knowledge, including new genomic tests. The traditionalEMRS designs could not have such dynamic updates (Murphy et al., 2017). Images that prompted the development of a new architecture: a cloud paradigm based on micro-services and comprising dedicated applications, made available to FHIR and allowing the substitution of apps without having to re-program, was driven by the need to solve these issues.

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

3. Related Work

An interoperability activity or two were in progress at the same time before 2020. The Direct Project focused on safe messaging and Nationwide health information network on document exchange. HealthVault was a personal health record targeted to the consumer, which had an app ecosystem (Kondapalli and Gunupudi, 2018). CommonWell Health Alliance strived to facilitate the exchange of documents by providers through EHR products that have incompatibilities. The SMART Health IT project, paid by the Office of the National Coordinator endeavored to develop substitutable third-party applications on EHRs. Early SMART Classic was implemented in terms of RDF and proprietary-code API but had low popularity among vendors. The solution to this was to migrate to FHIR in order to accommodate the SMART on FHIR platform that was demonstrated in 2014 using a number of vendors. Adoption of SMART on FHIR profiles and OAuth2 picked up in the industry later through the Argonaut Project.

4. Proposed Reference Architecture

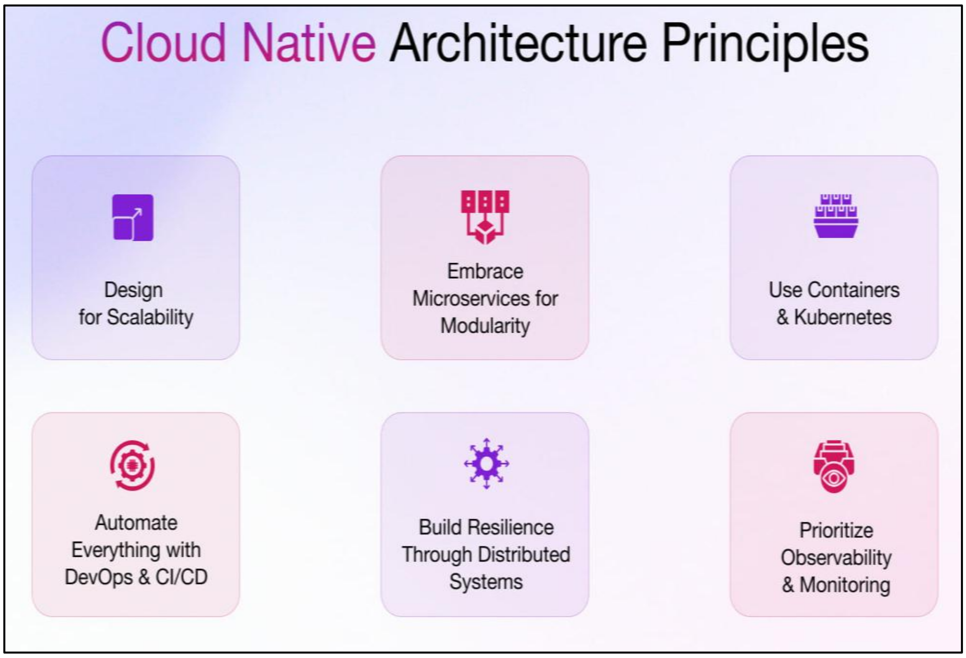

Figure 4: Proposed Reference Architecture
(Source: https://www.rishabhsoft.com)

Six core principles of a cloud-native architecture are embraced by the proposed reference architecture. Design for scalability allows horizontal scaling of FHIR API servers when amount of clinical data increases (Balaganski, 2015). Modularity decomposes SMART on FHIR functionality into different services: authentication; resource enrichment; query engine.Embedding microservices into modularity splits the SMART on FHIR functionality into separate services: authentication; resource enrichment; query engine. With containers and Kubernetes, FHIR resources are consistently deployed across public or private clouds. So, automation via DevOps and CI/CD decreases the chances of manual updates of clinical decision support apps. Distributed systems mean that if one microservice fails, access to patient data won't be affected (Nwaimo et al., 2019). Prioritize observability and monitoring is important for collecting logs and metrics used to monitor FHIR API's performance: Rapidly detect interoperability issues.

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

4.1 Architectural Principles

Principle	Description
Data at Rest	Microservices work on data in place; only results are exchanged
FHIR as Common Data Model	All clinical data exposed as FHIR resources
OAuth2 & OpenID Connect	Separate authentication from authorization
Interchangeable Apps	Any SMART on FHIR app works on any compliant platform
Incremental Implementation	Start with few resources/profiles, expand over time

Table 1: Architectural Principles
(Source: Self-Created)

The architecture suggested is based on five principles, the pre-2020 cloud-native and FHIR literature. Firstly, data is never moved - micro services operate on data that is stored in a repository, only data transmitted are the result. Second, FHIR is the common data model-all clinical data is opened up as FHIR resources. Third, OAuth2 and OpenID connect are the separation of authentication and authorization mechanisms (Srinivasan and Kumar, 2019). Fourth, apps are interoperable -any SMART on FHIR app can be used by any compliant platform, without any modifications. Fifth, implementation occurs through driblets and drops- organizations do not start off big in the amount of resources they consume and profiles that they adhere to as well. These principles are interdependent, allowing a loosely-oriented, vendor-neutral ecosystem with the potential to expand with the standards and clinical skills in the future.

4.2 Containerized Microservices for FHIR

The architecture has the functionality subdivided to microservices, which are containerized. A FHIR API server is a container, and offers access to FHIR resources via RESTful endpoints. Other micro services will be (authentication, resource enrichment (e.g. with SNOMED-CT or LOINC codes) and query execution) carried out. The services have the possibility to be scaled-up independently based on the demand. Containers allow running the platform on any cloud-native, either on a public (AWS) or a private one (Sailer et al., 2018). A 2013-2015 reference implementation showed that a FHIR server can be implemented in about 3000 lines of Groovy code, with a backend of a PostgreSQL database and a Java Virtual Machine. The lightweight architecture strategy allows the strategy to be accessible to small and large healthcare organizations.

4.3 API Gateway and FHIR RESTful Endpoints

An API gateway will be located at the edge of the structure, where all the requests of the clients will be redirected. It is concerned with open authorisation validation of a token (OAuth2), rate limitations and request logging. There are FHIR RESTful endpoints to each type of resource: GET /Patient/{id}, GET /Observation?patient= {id}, etc (Nwaimo et al., 2019). Supports the search parameters in FHIR which also supports chained searches. The FHIR CapabilityStatement, a specification of what resources and operations may be used, is another capability that the gateway has. This architecture protects client applications when there are changes to the microservices in the back-end. It goes so far as to execute the security policies concurrently (Gopinathan et al., 2018). The API gateway may use commercial off-the-shelf to be implemented; a working example was given in 2015 by the MITREid Connect based SMART on FHIR reference authorization server.

4.4 Event-Driven Messaging and Kafka

Healthcare data access is not necessarily all synchronous. An event-driven messaging layer based on a publish/subscribe model (conceptually analogous to Kafka) is thus a part of the architecture. An event will be published in the event where a new clinical observation has been added into a repository. Pollingless reaction Subscribed microservices (e.g. a clinical decision support rule engine) can react to the event. Scalability and responsiveness is enhanced by this decoupling (Weir, 2019). Pipelines of feature extraction can be triggered synchronously by events to scale to the size of the big data sources (e.g., a genomics server). The fact that microservice architecture must feature an event-based communication mode, thus allowing tight coupling to be avoided, is of value to the literature published prior to 2020 (Rubí et al., 2019). This movement enables real-time alerts (e.g. abnormal laboratory results) in the context of health care as well as keeps the FHIR underlying API simple and stateless.

4.5 Data Persistence and Polyglot Storage

Data Type	Storage Technology
Clinical resources (Patient, Medication, Condition)	Relational database or document store (e.g., CouchDB)
Big data objects (imaging, waveforms)	S3-compatible object storage
Metadata for big data	Separate additional store
FHIR index	Search engine

Table 2: Data Persistence and Polyglot Storage
(Source: Self-Created)

The architecture does not imply that it has to have one database. Instead, it calls on polyglot persistence: each microservice is putting the data in the most appropriate storage (Mukhiya et al., 2019). Clinical resources (Patient, Medication, Condition) can be stored in a relational database or document store like the CouchDB. S3-compatible object storage is used to store the big data objects (imaging, waveforms), and to store metadata in a separate store. FHIR index can be stored in a search engine (Balaganski, 2015). was demonstrated by i2b2-to-FHIR implementation of how a relational i2b2 repository can be encapsulated with FHIR REST API. The philosophy of maintaining data in place is also well-respected by this polyglot approach, meeting the costly ETL pipes. It also allows the evolution of each of the components to evolve independently.

5. Security, Privacy, and Compliance

The architecture implements security according to SMART direction of OAuth2 and OpenID Connect. OAuth2 allows granting a third-party application access to specific data, without credential sharing, between a patient or a clinician (El-Sappagh et al., 2019). OpenID connect can also be used to do

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

authentication, which includes a signed identity token and user claims. The access tokens also have limited scopes: e.g. read only access to the records of a single patient. The authorization server can put in place local privacy policies including the HIPAA requirements (Braunstein, 2018). SMART on FHIR launch protocol communicates to the app's EHR background (current patient, encounter). These standards have been well matured and were being used prior to 2020. No custom security solution has to be implemented, usage of the existing web standards in the architecture has taken place.

6. Validation: Simulated Use Case

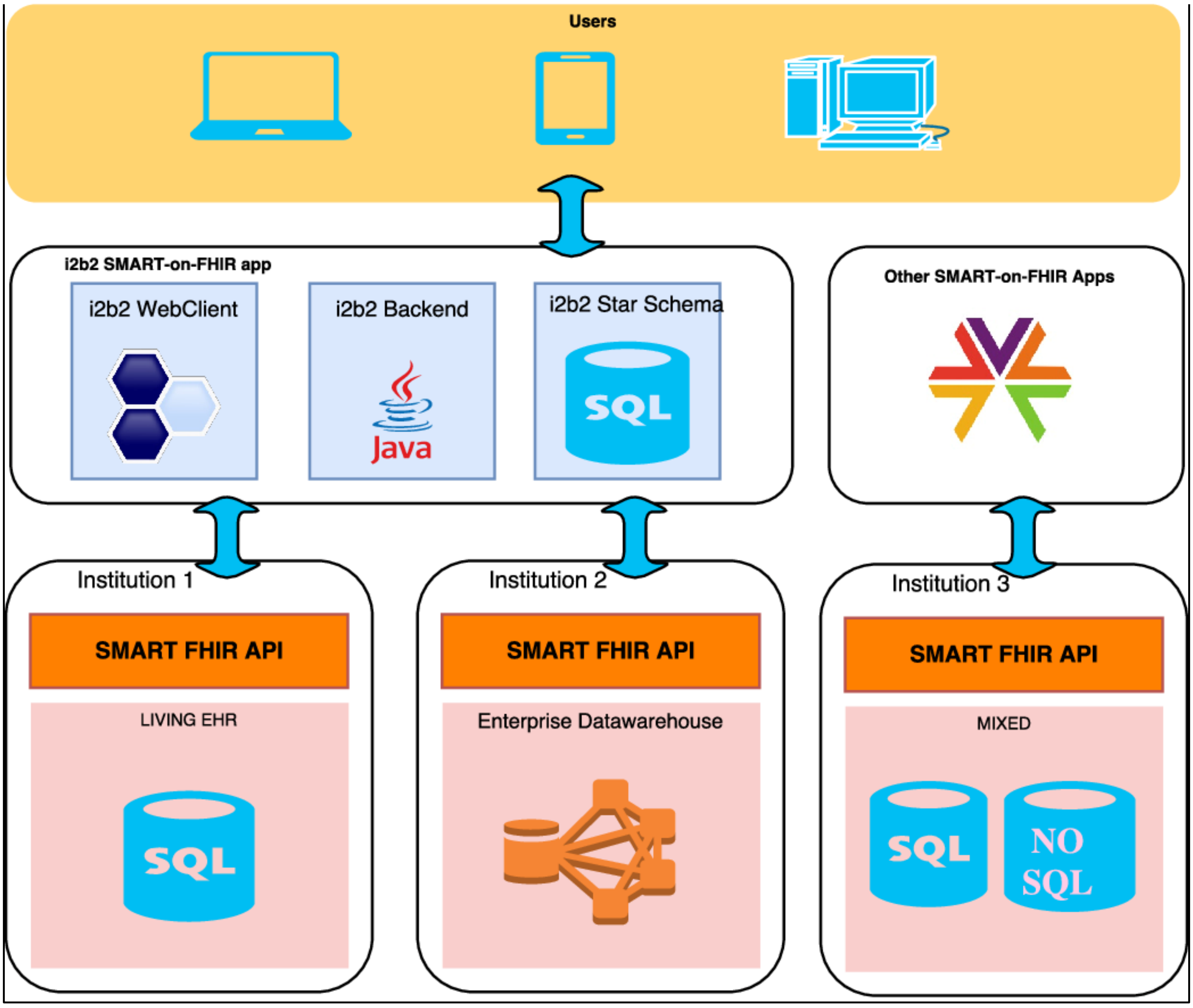

Figure 5: Validation: Simulated Use Case
(Source: https://figures.semanticscholar.org)

A simulated use case, based on the i2b2-SMART-on-FHIR implementation, detected an architecture test. A plug-in that included six modules was a FHIR cell wrapped test i2b2 with anonymized patient data resource authentication, REST API, i2b2-to- FHIR converter, resource enrichment, query engine and cache. The cell was opened with a patient dashboard (a SMART app) (Baihan and Demurjian, 2017). The application was able to retrieve the demographics, medications, lab observations and the diagnosis of test patients. FHIR resources were all provided in the form of JSON using HTTPS. The whole stack was based on commodity hardware. This experiment revealed that a historic clinical data repository is convertible to a SMART on FHIR platform with a small amount of engineering (one to two months to make a prototype).

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

7. Discussion and Limitations

The provided architecture will address the three barriers to big data integration that might have previously pre-2020 existed: it will introduce FHIR APIs afresh on top of cloud-native repositories, extraction of features via microservices, and replaceable apps, which can be upgraded autonomously. However, limitations exist. The reference implementations neither translated big data nor provided real time query translation with triggers on massive data. There was no reported production clinical deployment until 2020 (Mandel et al., 2016). Possible legal and commercial barriers to substitute the apps (liability, certification, reimbursement) were not researched. Moreover, semantic consistency can be done only with the help of FHIR profiles, but various organizations can generate profiles that cannot be integrated with each other causing fragmentation (Henry et al., 2018). In spite of these constraints, the architecture was technically achievable and was early supported by the industry (Cerner, Intermountain, VA).

8. Comparison with Pre-2020 Approaches

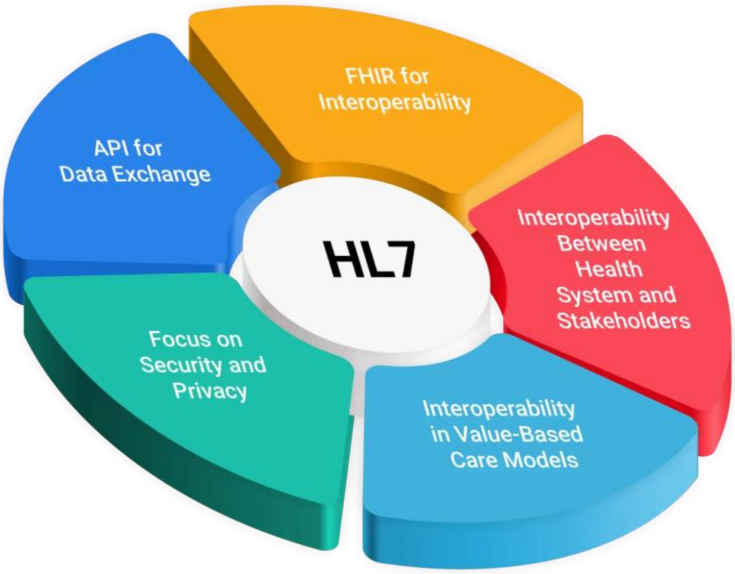

Figure 6: Comparison with Pre-2020 Approaches
(Source: https://www.fortunesoftit.com)

Prior to 2020, most of the interoperability strategies have been document-based (C-CDA) and message-based (HL7 v2). These were not conducive to granular, on-demand data access and third-party apps substitutions. Another standard RDF based protocol, SMART Classic, with a specific API, was not widely used between vendors because of variability in standards. Contrastingly, the suggested architecture is based on FHIR that is created with open licensing and open participation of the community and with HL7 (Petrakis et al., 2019). RESTful API and resource model of FHIR are user friendly and decrease the learning curve. OAuth2 and launch context FHIR has now been an app platform, and SMART on FHIR is added to OAuth2. The Argonaut Project (since 2015) proved that SMART could be applied on FHIR profiles by 40+ vendors, not as previously done.

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

9. Conclusion and Future Work

This article created a reference architecture of cloud-native healthcare products built on HL7/FHIR that supplements HL7 standards, but only includes literature published before 2020. The architecture is containerized microservices based, FHIR API gateway based, event driven messaging, polyglot persistence and OAuth2/OpenID connect based security. The help of a simulated use case was used in testing technical feasibility. Its architecture allows the replacement of third-party applications, has low vendor lock-in, and allows adoption to be adopted over time. Ongoing work after 2020 standardization involves working with population-level queries, introducing the capability to write to FHIR cells, automating the validation of data with FHIR profiles, and the use of SMART CDS Hooks to support clinical decisions. The final vision is to have an app store of health, in which innovation can prosper without being dependent on EMRS vendors.

Reference List

[1] Mandel, J.C., Kreda, D.A., Mandl, K.D., Kohane, I.S. and Ramoni, R.B., 2016. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. Journal of the American Medical Informatics Association, 23(5), pp.899-908.
[2] Chute, C., 2016. SMART-on-FHIR implemented over i2b2. Journal of the American Medical Informatics Association: JAMIA.
[3] Murphy, S., Castro, V. and Mandl, K., 2017. Grappling with the future use of big data for translational medicine and clinical care. Yearbook of medical informatics, 26(01), pp.96-102.
[4] Kondapalli, K.K. and Gunupudi, C., 2018. A HYBRID ZEROTRUST–DRIVEN CLOUD ARCHITECTURE FOR SECURING DISTRIBUTED ELECTRONIC HEALTH RECORDS IN AI-ENABLED HEALTHCARE ECOSYSTEMS. INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN ENGINEERING AND TECHNOLOGY, 9, pp.116-131.
[5] Srinivasan, K. and Kumar, R.L., 2019. Optimized cloud architectures for secure and scalable electronic health records (EHR) management. International Journal of Multidisciplinary and Current Research, 7(3), pp.345-351.
[6] Sailer, A., Yang, B., Jain, S., Tomala-Reyes, A.E., Singh, M. and Ramnath, A., 2018, November. Healthcare Application Migration in Compliant Hybrid Clouds. In the International Conference on Service-Oriented Computing (pp. 725-739). Cham: Springer International Publishing.
[7] Nwaimo, C.S., Oluoha, O.M. and Oyedokun, O.Y.E.W.A.L.E., 2019. Big data analytics: technologies, applications, and future prospects. Iconic Research and Engineering Journals, 2(11), pp.411-419.
[8] Weir, L., 2019. Enterprise API Management: Design and deliver valuable business APIs. Packt Publishing Ltd.
[9] Balaganski, A., 2015. API Security Management. KuppingerCole Report, 70958, pp.20-27.
[10] Braunstein, M.L., 2018. Health informatics on FHIR: How HL7's new API is transforming healthcare (pp. 13-29). Cham (Switzerland): Springer International Publishing.
[11] Baihan, M.S. and Demurjian, S.A., 2017. An access control framework for secure and interoperable cloud computing applied to the healthcare domain. In Research Advances in Cloud Computing (pp. 393-429). Singapore: Springer Singapore.
[12] Mandel, J.C., Kreda, D.A., Mandl, K.D., Kohane, I.S. and Ramoni, R.B., 2016. SMART on FHIR: a standards-based, interoperable apps platform for electronic health records. Journal of the American Medical Informatics Association, 23(5), pp.899-908.
[13] Petrakis, Y., Kouroubali, A. and Katehakis, D., 2019, October. A mobile app architecture for accessing EMRs using XDS and FHIR. In 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering (BIBE) (pp. 278-283). IEEE.

Journal of Information Systems Engineering and Management
2020, 5(1)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

[14] Henry, J.R., Lynch, D., Mals, J., Shashikumar, S.P., Holder, A., Sharma, A. and Nemati, S., 2018, July. A FHIR-enabled streaming sepsis prediction system for ICUs. In 2018 40th annual international conference of the IEEE engineering in medicine and biology society (EMBC) (pp. 4093-4096). IEEE.
[15] El-Sappagh, S., Ali, F., Hendawi, A., Jang, J.H. and Kwak, K.S., 2019. A mobile health monitoring-and-treatment system based on integration of the SSN sensor ontology and the HL7 FHIR standard. BMC medical informatics and decision making, 19(1), p.97.
[16] Mukhiya, S.K., Rabbi, F., Pun, K.I. and Lamo, Y., 2019, May. An architectural design for self-reporting e-health systems. In 2019 IEEE/ACM 1st International Workshop on Software Engineering for Healthcare (SEH) (pp. 1-8). IEEE.
[17] S. Rubí, J.N. and L. Gondim, P.R., 2019. IoMT platform for pervasive healthcare data aggregation, processing, and sharing based on OneM2M and OpenEHR. Sensors, 19(19), p.4283.
[18] Gopinathan, K., Kaloumenos, N.A., Ajmera, K., Matei, A., Williams, I. and Davis, A., 2018, March. FHIR FLI: An Open Source Platform for Storing, Sharing and Analysing Lifestyle Data. In ICT4AWE (pp. 227-233).