ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

# Latency-Optimized Interoperability Design

**Ghatoth mishra**
Independent Researcher
ghatothmishra@gmail.com

## Abstract

The use of Health Level 7's Fast Healthcare Interoperability Resources (FHIR) for facilitating real-time data exchange among healthcare systems in life-critical scenarios enables patients to receive timely and appropriate treatment whenever and wherever they need it. FHIR provides models, rules, and interfaces for data exchange between computerized applications operating in the healthcare domain. Being a web-based standard, it can be easily deployed using an event-based architecture. Furthermore, it allows for online communications and moves away from standard file batch processes. The eventing mechanism facilitates the push of data from an event publisher to interested subscribers and notifies them of updates for resources of interest.

The increasing diversity and complexity of healthcare information systems, along with the demand for timely integration of data across these systems, point toward the need for a real-time event-based integration approach using FHIR REST services. Open-source frameworks have been developed for achieving the required integration. However, architectural decisions in designing such frameworks have a significant impact on their ability to support data exchange with low latency. The selection of an appropriate integration architecture is also crucial for meeting requirements such as interaction with external enterprises, protocol translation, security, and compliance with cloud deployment models. Therefore, several architecture patterns commonly considered and deployed for event-based integration using FHIR are discussed. Each architecture pattern is detailed, along with its strengths and weaknesses for supporting real-time data exchange.

**Keywords:** HL7 Fast Healthcare Interoperability Resources (FHIR), Real-Time Healthcare Data Exchange, Event-Driven Architecture, FHIR RESTful Services, Healthcare Systems Integration, Low-Latency Clinical Data Processing, Interoperability Standards in Healthcare, Event-Based Integration Patterns, Cloud-Enabled Healthcare Platforms, Secure Health Information Exchange, Protocol Translation Frameworks, Healthcare API Design, Life-Critical Clinical Systems, Publish–Subscribe Mechanisms, Open-Source Healthcare Integration Frameworks, Scalable Health IT Architectures.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

## 1. Introduction

Real-time data exchange between healthcare stakeholders enables timely and precise decision-making, reducing the chances of medical errors and unnecessary examinations. Data flow across hospitals, physicians' offices, and other stakeholders is often facilitated by Enterprise Service Buses, SATs, or various proprietary solutions. However, clinical environments continually change: patients may experience sudden symptoms and distress, and clinical workflows may differ from the normal flowchart. These facts, in conjunction with the ever-increasing amount of information generated and stored about patients, require a stepping-stone between normal/complementary data flow between applications and real-time data refresh based on events/message exchange. Patients' states should be monitored and consequently reflected in the monitoring application by alerts and actions linked to patients at risk. Moreover, applications that rely on clinical decision support need data related to clinical indications and affected sites to be kept up to date in order to make a precise evaluation of possible interventions. Various techniques can help near real-time data refresh according to environment conditions such as service availability and delivery time requirements.

The Fast Healthcare Interoperability Resources (FHIR) is a modern family of specifications created by HL7 International that facilitates company interoperability through the definition of simple data models, a RESTful interface, declarative complementarities, and event-based notifications among others. Both the theory and implementation support the creation of FHIR-based interoperability solutions at a high level of fidelity that can serve as a foundation for further investigations. Three patterns of architecture design for REST and event-oriented approaches are defined, enabling near real-time data exchange. Strong solutions built on these architecture patterns address live behavior in an acceptable scenario foundation.

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

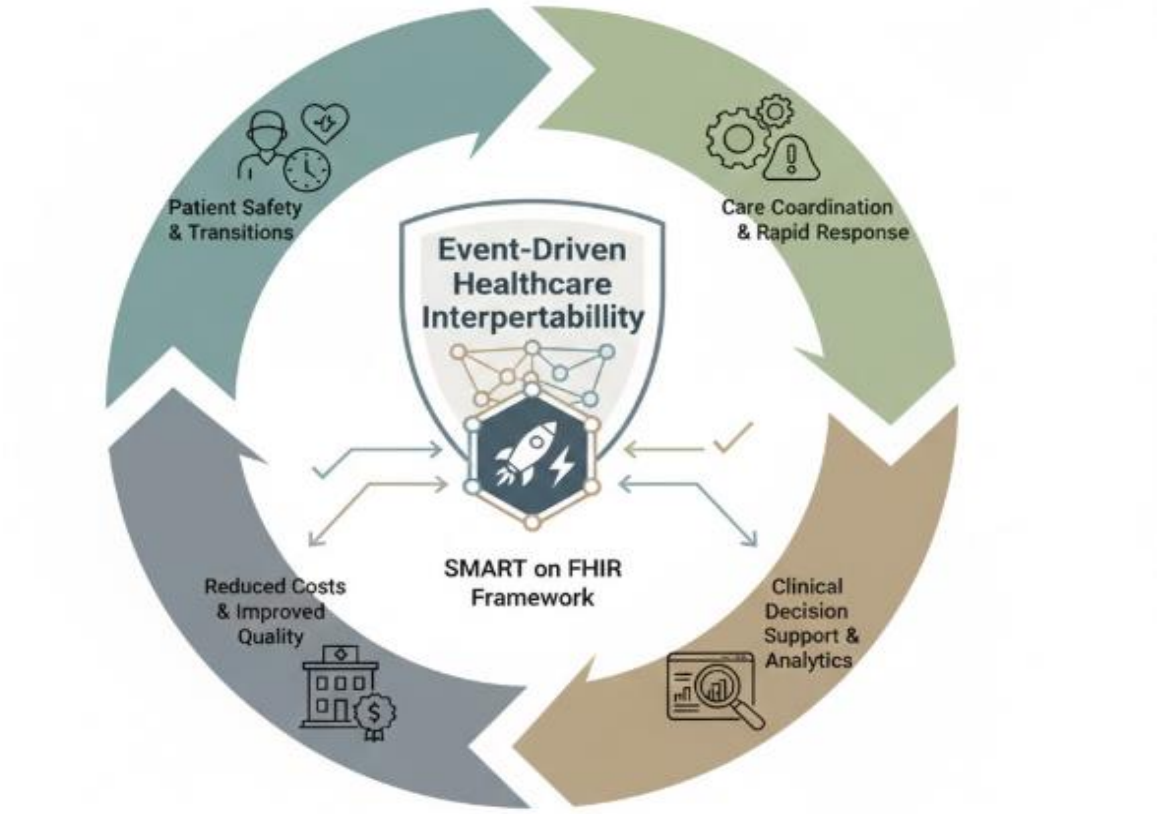



**Fig 1: Real-Time Interoperability via Event-Driven Architecture: Scaling SMART on FHIR for Value-Based Healthcare Ecosystems**

### 1.1. Overview of the Study and Its Objectives

Interoperable real-time exchange of healthcare data is crucial to fully materializing the value-based healthcare model. For patients, timely access to the latest clinical data is pivotal for safe care, particularly during critical care transitions. For healthcare organizations, effective internal and external care coordination, coupled with rapid response to issued alerts, reduces care costs while improving care quality and patient safety. For clinical decision support solution developers, real-time updates are crucial for delivering relevant logic. Developers of other systems operating at a higher level of abstraction also have an interest in accessing and

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

exploiting real-time data. Related benefits are materialized in terms of quality and safety increase and cost reduction. Hence, guidance for the design and implementation of real-time data exchange

It is evident that a shared event-driven architecture for real-time exchange of healthcare data across and within organizations and services brings immense benefits. Initial interest in real-time healthcare interoperability is mainly driven by developers of clinical decision support solutions. Nevertheless, the urgency for event-driven exchange is becoming increasingly tangible also among other stakeholders. The SMART on FHIR framework already enables seamless real-time integration for solutions hosted in an application portal supporting the corresponding FHIR App Launch mechanism. Within a few years, SMART on FHIR will cover all aspects of clinical decision support integration. Interest in the real-time update of the underlying healthcare data, however, extends well beyond the clinical decision support domain. For patients, timely access to the latest clinical data is pivotal for safe care, particularly during critical care transitions. For development organizations, effective internal and external care coordination, coupled with rapid response to issued alerts, reduces care costs while improving care quality and patient safety.

**Equation 1) Throughput (events/sec or transactions/sec)**

**Step-by-step derivation**

1. Let
    - $N_{\text{success}}$ = number of successfully completed transactions (or successfully published/delivered events)
    - $T$ = measurement time window (seconds)
2. "Per time window" means "divide by the window length".
3. Therefore, throughput is:

$$\boxed{\text{Throughput } (X) = \frac{N_{\text{success}}}{T}}$$

4. Units check: if $N_{\text{success}}$ is "events" and $T$ is "seconds", then $X$ is "events/second".

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

## 2. Background and Motivations

Traditional approaches to healthcare data exchange have generally centered on complete and often complicated datasets. Clinical requirements for timely access to healthcare information call for real-time interoperability, enabling the round-the-clock flow of data between organizations, applications, and medical devices. Such capabilities foster a variety of new clinical applications, redefined service models, and integrated delivery networks finally focused on improved patient outcomes.

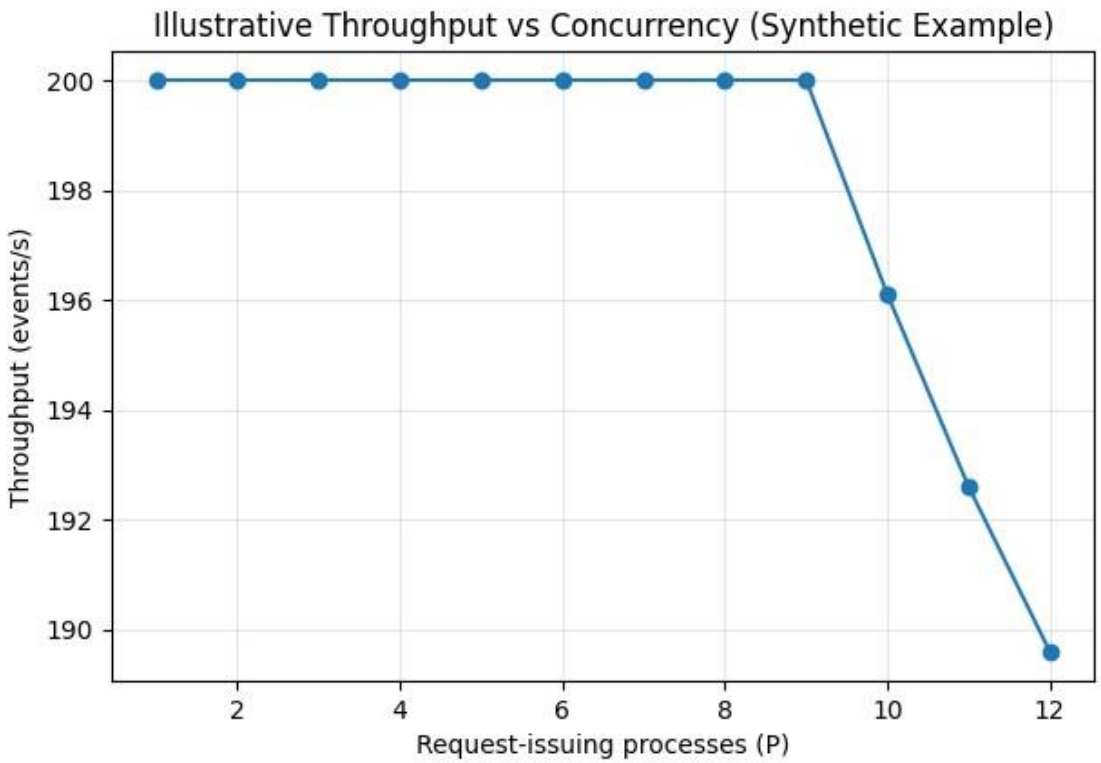
Motivations for real-time interoperability are many, clearly identified in numerous regulations and mandates. Hospitals, practices, and other healthcare organizations are under continual pressure to accomplish better patient outcomes and to enhance patient safety. Shortcomings in data sharing are well understood to have prevented useful capabilities from becoming more widely available. Additionally, market and clinical forces, policy initiatives, and the needs of an increasingly activated patient population are driving the need for real-time clinical data exchange capability. Delivering quality health care and improving patient safety are major concerns for regulators, and similar considerations influence data exchange mandates issued by health insurance payers. Resources are inherent in FAST, Advancing Interoperability by Streamlining Health Information Exchange, the 21st Century Cures Act, and HHS Interoperability and the Blocking Rule.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07




## 2.1. Drivers of Change in Healthcare Interoperability

Recent trends in information and communication technologies have enabled vast amounts of electronic patient data to be stored in institutions, but these data are largely inaccessible for clinical applications, such as clinical decision support, or for use by patients and clinicians. Commercial products offer specialized solutions to some of these problems by allowing apps to consume certain types of data, but such environments do not scale well to third-party development across institutions and geography. Previous arguments for open standards in healthcare exchange have focused on the patient's role as information provider, but clinical needs also demand greater provider capabilities and wider, granular access to patient data.

Incentives for connecting disparate information systems in healthcare have grown, and external pressures are mounting to overcome the technological resistance outlined. With limited

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

development resources, hospitals face a high burden of creating ad hoc interfaces to serve every vendor-specific need. National and international certification entities are linking rewards and outcomes to system connectivity; without it, hospital acquired conditions are being penalized. Failing to achieve semantically consistent, pan-institutional data can produce unwelcome legal exposure. Integrating information from other providers enables patient safety checks and reduces clinical duplication and cost, so stakeholders in an increasingly competitive environment recognize the value of providing timely—ideally real-time—access to patient data.

**Equation 2) Drop ratio (loss / non-delivery fraction)**

**Step-by-step derivation**

4. Let
    - $N_{\text{attempt}}$ = attempted/published events (or attempted transactions)
    - $N_{\text{success}}$ = successfully delivered/processed
5. Dropped events:

$$N_{\text{drop}} = N_{\text{attempt}} - N_{\text{success}}$$

3. Drop ratio is the fraction dropped out of attempted:

$$\boxed{\text{Drop ratio } (D) = \frac{N_{\text{drop}}}{N_{\text{attempt}}} = 1 - \frac{N_{\text{success}}}{N_{\text{attempt}}}}$$

## 3. FHIR Fundamentals for Real-Time Interoperability

Real-time interoperability relies on four aspects of FHIR: modelling, supported interfaces, the Resource-based Dynamic Eventing Model, and sub-resources/contained-resources. Resource modelling is rooted in abstraction, the need to cover nearly all social and clinical aspects of the provisioning of care, and a focus on CRUD operations. Exploration of modelling choices like profiling and defining specialization constraints is applied to a real-time use case of MedicationAdministration for resource-driven notifying.

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

A large part of the interaction patterns in clinical environments can be derived from REST, and therefore focus is on CRUD and CRUD-like operations. Naturally, Activity and Created have a meaning but are less important. The complete set of operations, such as Search with its built-in paging mechanism, conditional operations on Create, Update and Delete, and the use of Transaction can be found in the specification. Only the Server-Sent Updates approach, a special case of the Publish/Subscribe pattern, is elaborated. It is directed primarily at real-time situations with requirements for high throughput and timeliness for data generation and potential resource replication rather than full support for CRUD legality.

Finally, research work around the Resource-based Dynamic Eventing Model is summarized, demonstrating the dual use of Resource-based Dynamic Eventing as a form of Server-Sent Updates and as the Notification part of the Publish/Subscribe pattern. It also shows that Notification can enable simple-but-robust solutions when a full Publish/Subscribe solution cannot be achieved. The Guarantee on Notification Delivery axis enables specification of what is guaranteed concerning delivery of notifications within Notification workflows.

| Processes (P) | Attempted events (N_attempt) | Delivered events (N_success) | Dropped events (N_drop) |
|---|---|---|---|
| 1 | 12000 | 12000 | 0 |
| 2 | 12000 | 12000 | 0 |
| 3 | 12000 | 12000 | 0 |
| 4 | 12000 | 12000 | 0 |
| 5 | 12000 | 12000 | 0 |
| 6 | 12000 | 12000 | 0 |

### 3.1. Resource Modeling and Profiling

Resource modeling for real-time exchange adheres to FHIR foundations, with a broad range of domains and coverage areas. Its implementation balances specification and flexibility. The profile-based extension mechanism and modelling approach by pairs of resources refine a complex model by separating the data exchange from its origin in a relevant context. Event-based and notification delivery mechanisms for FHIR resources complement their generic modeling approach and aspects that are typical for real-time use cases. Resource binding to

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

interaction patterns creates avenues for specialization and enforces the use of given constraints without the need for specifying a profile on every instance.

Event-based use cases rely on subscriptions and differ from patterns that follow a service-oriented approach. Asynchronous message-oriented middleware such as event brokers decouple message producers from consumers in latency-sensitive applications because their use radically reduces the need for an explicit request. Other patterns for real-time update include the support for server-sent updates. Long-polling and WebSocket patterns provide the necessary fidelity without being monitored by many subscribers.

A pattern based on a dedicated event gateway, also called publish-subscribe architecture, further contributes to reducing latency by distributing messages to the interested consumers. The alternative of an event mediation layer allows access to an event broker from any other protocol and creates a security boundary to shield the backend. While the first pattern is typical for hospitals in major cities, the second is common to smaller hospitals, monitoring devices and services hosted in the cloud.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

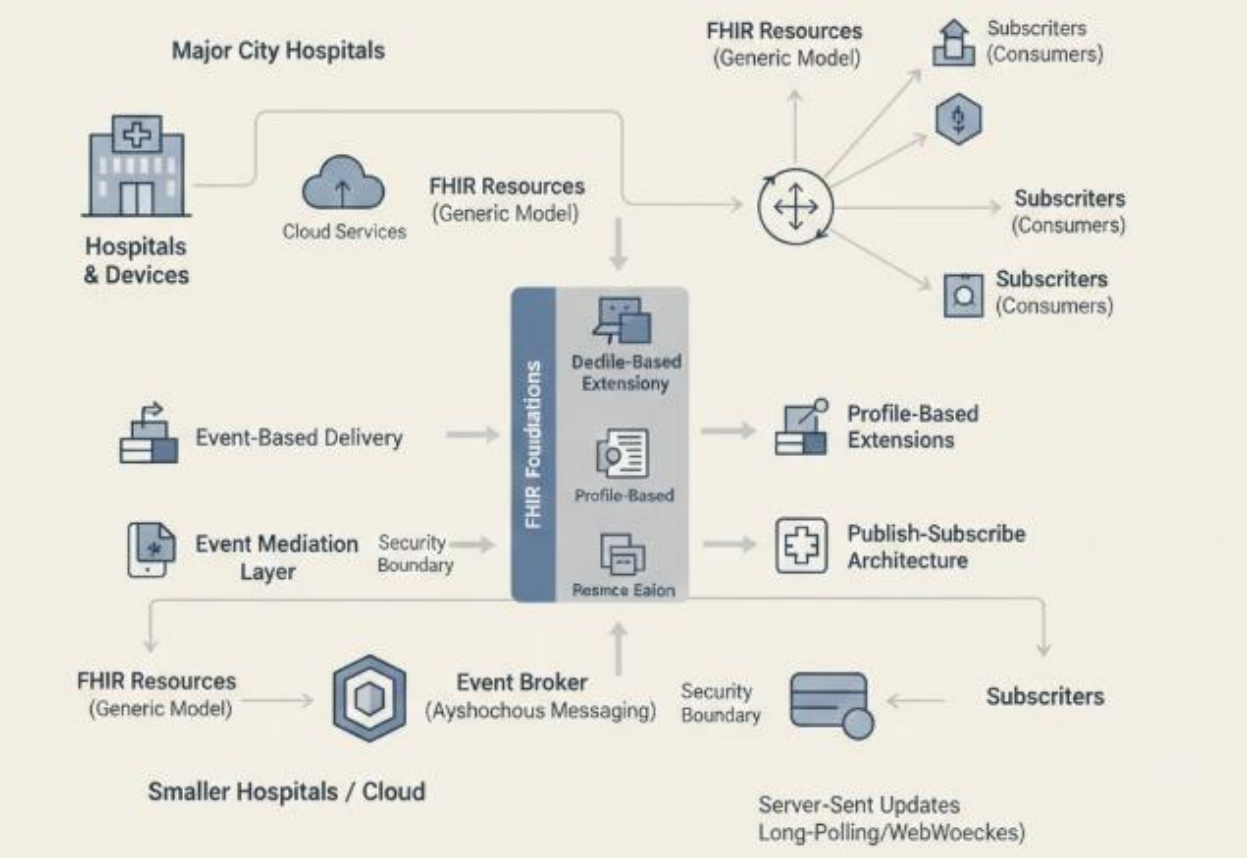



**Fig 2: Real-Time Health Data Interoperability: Optimizing FHIR Resource Binding and Event-Driven Architectures for Scalable Clinical Exchange**

### 3.2. RESTful Interfaces and Operations

Representational state transfer (REST) is an architectural style for application design in which resources are identified by URIs and manipulated through defined representations. The four core operations match the Create, Read, Update, and Delete (CRUD) operations of a database and this close mapping permits a straightforward and durable implementation of a RESTful API over a resource-oriented data model. REST has become the dominant architectural

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

style for the Web and is commonly placed in contrast with SOAP, which implements a remote procedure call (RPC) pattern that imposes a strong coupling between client and server.

While CRUD is sufficient for many applications, data access patterns also include searching or listing – retrieving a list of resources that match certain criteria – and conditional DELETE operations, in which a resource is deleted only if its current state meets a specified condition. Conditional operations allow client applications to perform statement-based operation sequencing without introducing update locks, so long as data conflicts are not severe. To promote data validity and reduce effort versus response time requirements when operating on large collections, CRUD-type operations support paging, while short read-caching enables rapid access to data that is relatively static.

### 3.3. Subscriptions and Real-Time Evening

Subscriptions form a powerful mechanism suitable for different distribution and scale requirements in real-time healthcare data exchange. The standard FHIR subscription support enables clients to register interest in particular resource updates on a FHIR server, and upon occurrence of a matching event, an automated notification is sent to a callback address.

Various implementations exist across different FHIR servers with respect to delivery guarantees, notification content, or notification workflows and support. Nevertheless, they provide a fundamental capability that is particularly suited for clinical settings. Making notification callbacks support webhooks is paramount for reducing the operational burden of delivering notifications, and an event broker g-level architecture can complete Server-Sent Events support. Other steps can be taken to better reconcile subscription support of FHIR RESTful APIs with the needs of real-time operation. Those indicatives can be augmented to publish-subscribe engines, allowing loose coupling, horizontal scalability, and filtering semantic. Integrating partitioning and replication introduced non-trivial characteristics, reducing temporal predictability while increasing reliability. Appropriate means can then be employed to minimize latencies.

By integrating a gateway that translates messages between FHIR clients and a publish-subscribe engine, consistent and efficient translation is supported, automating the required adaptation. The resulting architecture fits high-availability and high-throughput scenarios. The inclusion of a mediation layer capable of transforming notification messages among formats


makes the approach even more versatile, supporting heterogeneous eventing systems, enhancing security, and simplifying deployment.

**Equation 3) Throughput gain (relative improvement)**

**Step-by-step derivation**

6. Pick a baseline configuration (e.g., $P = 1$ process) with throughput $X_{\text{base}}$.
7. New configuration has throughput $X_{\text{new}}$.
8. Absolute gain: $X_{\text{new}} - X_{\text{base}}$
9. Relative gain (dimensionless):

$$\boxed{G = \frac{X_{\text{new}} - X_{\text{base}}}{X_{\text{base}}}}$$

5. Percent gain:

$$\boxed{G\% = 100 \cdot G}$$

## 4. Architecture Patterns for Real-Time Healthcare Data Exchange

Two types of architecture patterns are defined in relation to real-time healthcare data exchange, considering the characteristics of healthcare data, particularly its episodic nature. The applicability of the patterns to the clinical process is analyzed. Event-driven architectures together with FHIR-supported publish-subscribe mechanisms appear to be well suited for many use cases, but they are not without limitations. Alternative patterns incorporating gateways or mediation layers fulfill a different set of requirements and support other use cases.

The healthcare domain is becoming increasingly event-driven; newly generated data should be processed and made available to interested parties with little or no latency, according to plays or business process design, or even outside established plays. Event-driven architectures (EDA) use a production/consumption model for distributed systems where the events that occur in the system are captured and stored by one or more event brokers. Additional subscribers can

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

then consume copies of these events at their own pace. The event store can retain the events for a predefined amount of time or export them to other long-term data systems for future analysis. In the healthcare context, the significant amount of data generated, the often-episodic nature of medical events, and the high costs of maintaining those infrastructures make this digitally transient and accessible data an attractive target for commercial event brokers.

Real-time applications in this domain often follow the FHIR standard, and their requirements can be served directly through the pattern of server-sent events (SSE) for one-way, simplex communications from server to client, as well as publish-subscribe models through protocols like WebSockets—full-duplex TCP channels between browsers and web servers that operate through a single socket—using libraries such as Socket.io for operation in other environments. The combination of SSE and publish-subscribe models allows the evolution of information from clinical services to clients without having to code the collection of a resource or the subscription and reception of notifications at the level of the clinical logic; it is typically the final layer of clinical decision support that collects data and will send requests.

### 4.1. Event-Driven Architectures

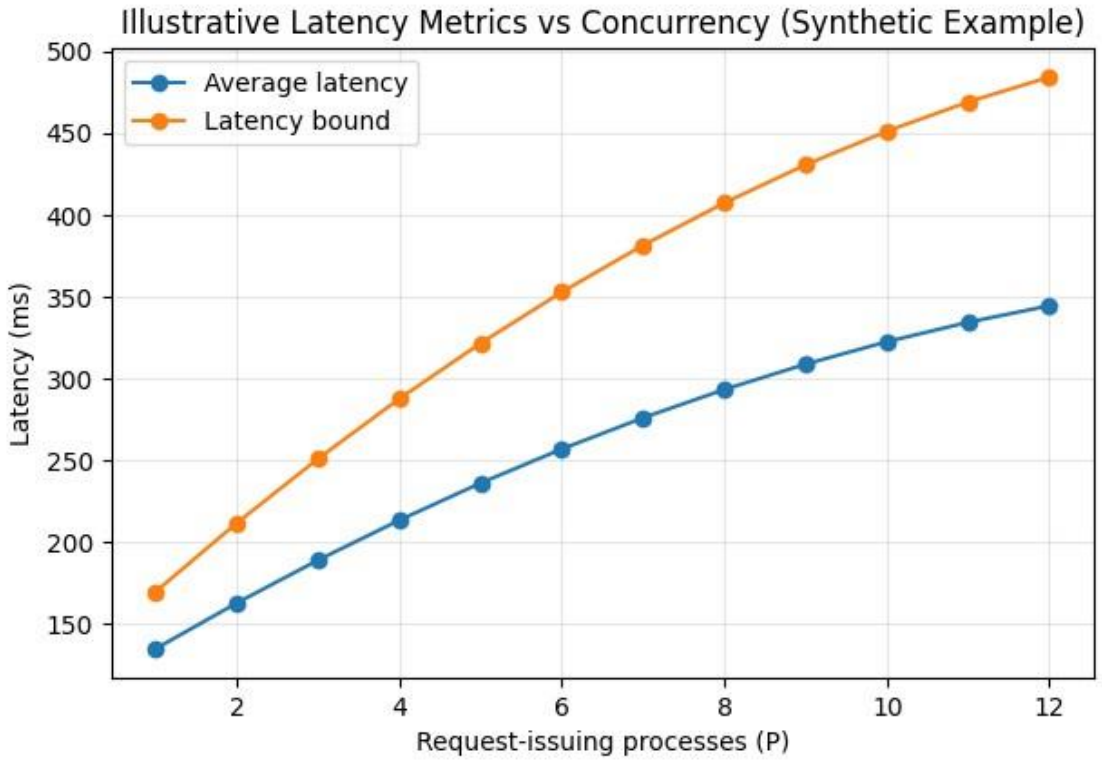
The first architecture pattern describes event-driven structures, where newly created or updated FHIR resources generate notifications through subscriptions. These notifications can trigger more complex processes within the monitoring organization. Event-driven architectures and services can be further categorized based on the technology chosen to implement them: event brokers that route notifications from publishers to subscribers, or a stream processing platform that monitors the resource streams of multiple organizations. A key aspect to consider is that many monitors only need to be informed about a subset of resources in one of the organizations. This can be achieved in different ways, depending on the needs of each use case.

Monitoring organizations subscribe for updates only about the specific resources relevant to the clinical use case. These subscriptions can also constrain the settings of the received resources, filtering out the ones not needed by the backend service or application. For example, the backend of a data warehousing solution for a regional health authority may want to receive complete medication administrations when a patient in the region has been administered a high-risk drug. In this case, it could create a subscription that listens to all medication administration resources of the patients residing in the region and that contain such drugs in their list of high-risk medications, but without the other recent drug administrations. Response times for

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

notifications depend mainly on how quickly the event broker or stream processor can evaluate these constraints.

In addition to supporting the main clinical decision support workflow, subscriptions can also make the addition of new, concurrent decision-checking workflows straightforward. If the corresponding item is published as part of the standard terminology services for the clinical collaboration environment, organizations can simply create the required subscriptions and the monitoring solutions will be ready to operate. For example, a monitoring organization could create a subscription that listens to all unplanned readmissions of patients older than 70 with a soft renal insufficiency, enabling the implementation of a decision workflow that executes a specific intervention after the second readmission.




### 4.2. Server-Sent Updates and Publish-Subscribe Mechanisms

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

Server-sent updates and publish-subscribe mechanisms support efficient notification distribution, allowing clients to listen asynchronously for resource updates. Clients initiate the subscription and receive real-time updates without having to continuously poll the server at regular intervals. Availability and scalability are critical for notification services that can experience bursts of message traffic at short intervals. Latency and delivery guarantees—such as at-least-once and exactly-once—are also important factors that can influence the choice.

Event-driven architectures are a class of EDA offering an elastic, fail-safe, and eventually consistent alternative to request-response REST interactions. Rather than extending the REST architecture style to support publish-subscribe interaction patterns that are not a natural fit for the style, an event broker is used to decouple the producers of events from consumers. Clients that are only interested in specific data update events subscribe to the publish-subscribe channels (or topics) of their interest and receive notification updates either in a restful manner (i.e., using HTTP long polling) or over dedicated connections using WebSocket or other streaming protocols. Subscribe channels serve as appropriate layers of functional decomposition to naturally subdivide and aggregate events occurring in the distributed system.

### 4.3. Gateway and Mediation Layer Designs

Models for real-time healthcare data exchange also include architectures in which an additional gateway functions at the border between a healthcare organization and Partner A, B, or C. These architectures introduce a demarcation boundary which is often motivated by non-functional requirements—such as concerns about security, management, Quality of Service, or access control—rather than by purely functional requirements. The gateway receives updates from the organization's systems, possibly translates them into another protocol (such as AMQP from REST), and makes them available to interested parties.

Gateways also simplify management (and maintenance) of multiple connections to multiple partners. By acting as a single point of access for data producers (e.g., an EHR system) and data consumers (e.g., decision-support applications), the number of connections can be reduced even further. A single data producer can connect to the gateway while it is subscribed to multiple data consumers and vice versa. The gateway does not have to involve the classic middleware function of decoupling systems operating on different protocols; these transformations only need to occur when data producers and consumers use different protocols. In addition to protocol translation, security between the governance-domain boundary can also be enforced at the gateway layer.

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

An important practical design decision concerns the distribution of the gateway service over multiple nodes. Distributing the gateway layer also provides an avenue for scaling out systems that require low latency or high throughput in real-time data exchange. Data updates can be selectively received and processed only in the data-sourcing gateway nodes, thereby decreasing the processing burden in all the gateway nodes, and ultimately lowering end-to-end latency. Distribution enables addressing the gateway layer where it is reasonably close to the data consumers that require low-latency service.

**Equation 4) Latency and latency decomposition (end-to-end)**

**Step-by-step derivation (per event $i$)**

10. Let
    - $t_{\text{trigger}}^{(i)}$ = time event $i$ is triggered
    - $t_{\text{processed}}^{(i)}$ = time subscriber finishes processing event $i$
11. End-to-end latency:

$$\boxed{L_i = t_{\text{processed}}^{(i)} - t_{\text{trigger}}^{(i)}}$$

4. Decompose into the components named in the article:
    - $t_{\text{broker}}^{(i)}$ = time spent in broker
    - $t_{\text{deliver}}^{(i)}$ = broker-to-subscriber delivery time
    - $t_{\text{proc}}^{(i)}$ = subscriber processing time
5. Then:

$$\boxed{L_i = t_{\text{broker}}^{(i)} + t_{\text{deliver}}^{(i)} + t_{\text{proc}}^{(i)}}$$

6. Average latency over $n$ events:

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

$$\bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$$

6. Latency bound (as "maximum duration" per the paper):

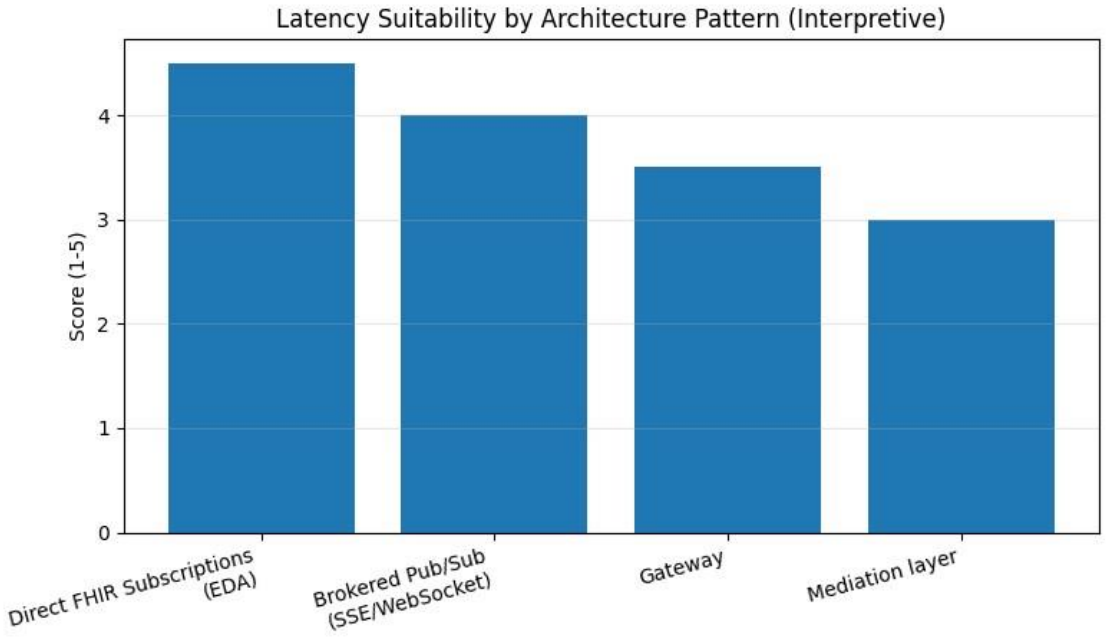
$$L_{\max} = \max_{i\in\{1,\ldots,n\}} L_i$$

## 5. Interoperability Frameworks and Standards Alignment

Event-driven communication patterns are an acknowledged need for real-time healthcare data exchange, but clinical decision-support frameworks and terminology services are equally important contributors to a well-functioning system. In addition, new clinical services often require data from multiple systems, thus crossing the boundaries of individual FHIR services. An interoperability framework describes the orchestrating components and principles necessary to achieve interoperability among independent systems. Given these drivers, an interoperability framework should align with FHIR to provisions for—1) the Smart Health IT (SMART) on FHIR architecture and the associated CDS-hooks eventing decision-support integration model; 2) the provision of real-time call services and the corresponding requirements for terminology mapping and services; and 3) the promotion of data integration across multiple independent systems exchanging data using FHIR.

The SMART on FHIR architecture extends FHIR for application integration by defining a call-and-response method for app execution requests, thereby integrating web-based apps in an organization's electronic health records system. The incorporation of decision-support services, particularly for diagnostic, therapy, and assessment suggestions, enhances the service provision of a FHIR-based architecture. For such decision-support services to be effective, they require access to coder services (that provide readable data for clinical coding), terminology services (to ensure the correct use of codes in message-generated term elements), and often data from multiple organizations, thereby invoking the composite-synchronization requirement.

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07




### 5.1. SMART on FHIR and Clinical Decision Support Integration

Applications and services interacting with the healthcare domain often require supporting clinical decision workflows, whose execution relies on a healthcare system. These interactions are enabled by a smart on FHIR approach, through which an application expresses its initialization and subsequent operations with the server via OAuth 2.0. When properly orchestrated, these workflows can also serve as premises for FHIR interoperability and be defined according to business service models, even if not formally supported in existing interoperability pyramids.

A clinical decision support (CDS) application operates as a service provider during the decision workflow, offering assessment capabilities that depend on being given as input the patient data required to perform the analysis. Events related to resource creation and updates can be exploited to kick off the decision-making process, boasting a time window that shortens response times and optimizes resource consumption. Once a patient is identified, the CDS service subscribes to the corresponding patient resource at the hosting FHIR server, requesting a


notification every time a resource in the patient scope changes. By registering for a notification every time a CRUD operation occurs for the subscribed patient, the service massively reduces unnecessary cycles. Upon reception of the notification, the patient resource data can now be fetched and readily input to the decision-analysis engine.

When the CDS service pushes a notification back to the invoking service, it bundles both the decision and the corresponding event. The invoking service processes the information as needed, and should the integration process be interrupted or the transported message lost, the notification is still automatically guaranteed. The approach is similar to that offered by CDS hooks; the notable difference is that, in this design, the service does not need to invoke the target FHIR service with an empty base URI.

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

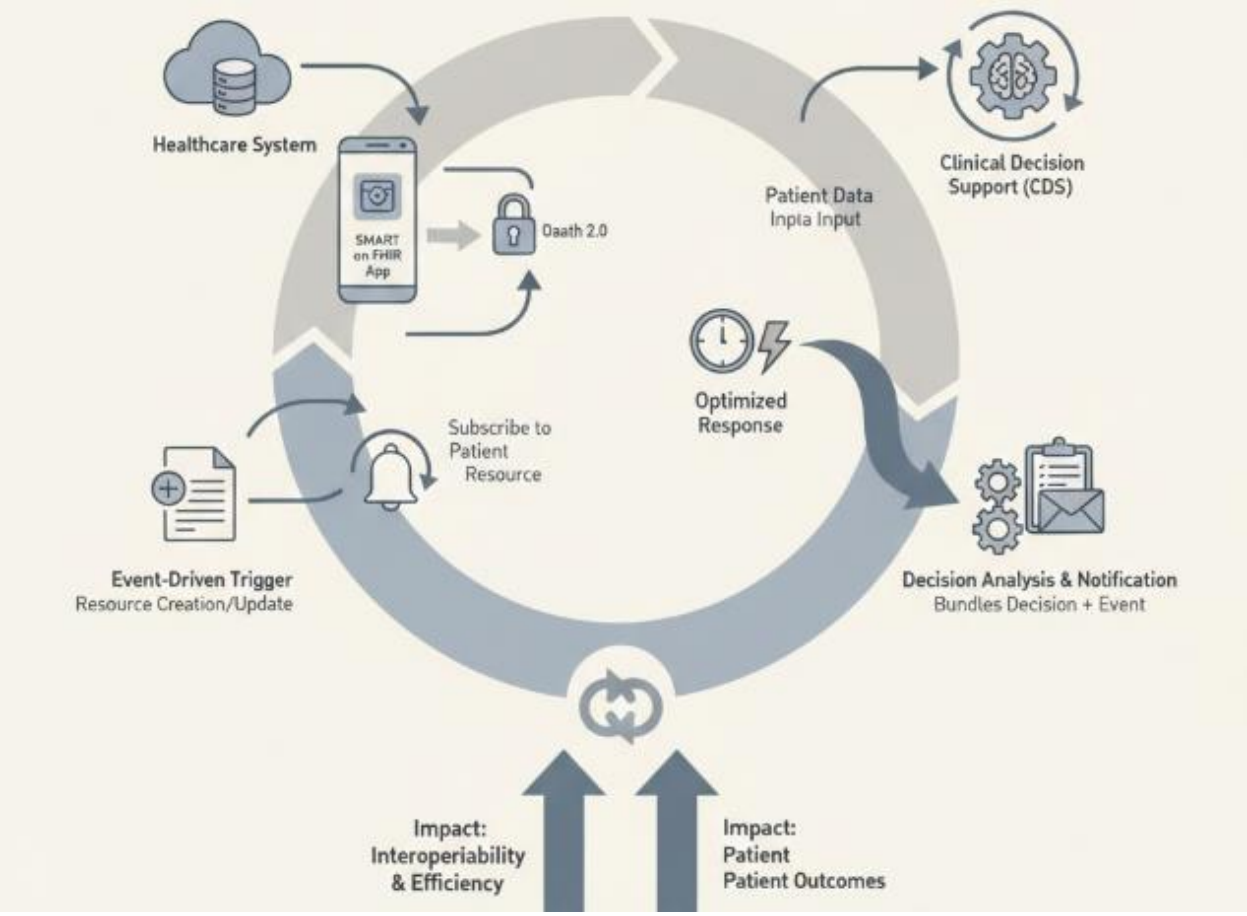



**Fig 3: Event-Driven Interoperability: Optimizing Clinical Decision Support through SMART on FHIR Subscription Architectures**

### 5.2. Terminology and Coding Systems Alignment

Terminology services are essential for real-time healthcare data interoperability. They provide the means to translate resource codes, ensuring semantic coherence across any message. Various levels of granularity and coverage are required in a real-time environment because resources may contain many coded fields with different domains. These needs can be fulfilled by

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

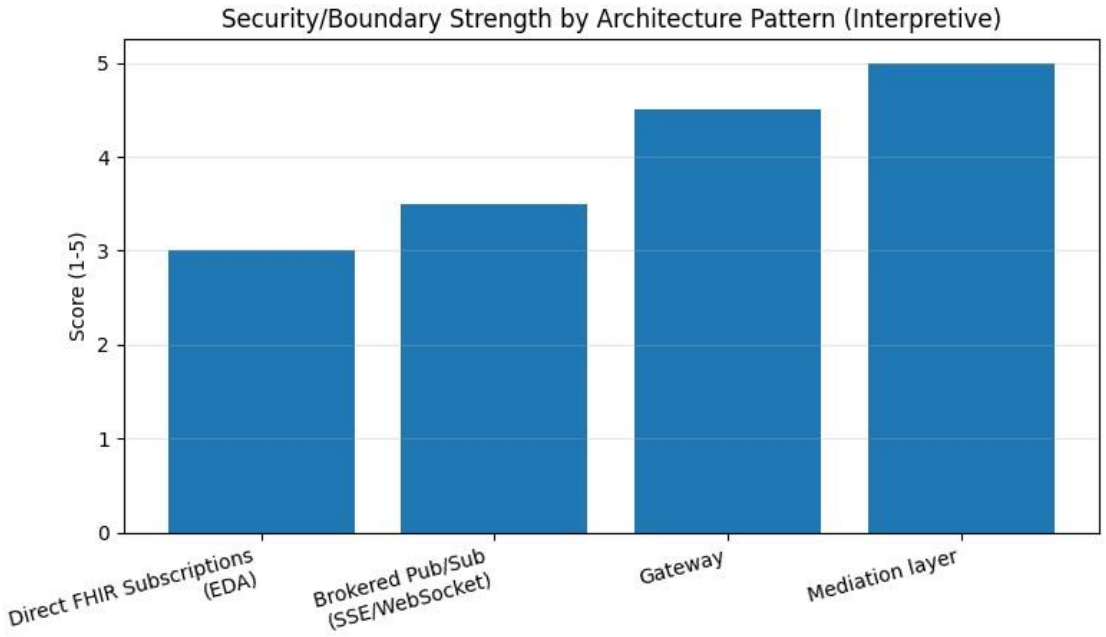
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

applying coding systems with a hierarchical structure, where the codes with general meaning in higher nodes are mapped to codes with specific meaning in lower nodes.

Terminology services are generally based on coding systems with hierarchical structures and the mapping of equivalent codes among different systems. In fact, the concept of terminology service itself has become broader and now covers the mapping of codes belonging to different coding systems, as well as the definition of valid value sets for described resources. Additionally, Semantic Mapping, Data Integration, and Information Fusion have become major research areas by their own and have received a lot of attention in the last two decades. When it comes to Terminology Services for Healthcare Data Interoperability, a set of hierarchical coding systems with appropriate coverage is usually provided for the most important domains of a given application. Furthermore, a proper mapping between those code systems is defined in order to guarantee the possibility of translating values. These translated values allow terms belonging to several domains to be unified.

Apart from coding systems, terminology services may also define the valid values for the coded fields of a resource. A FHIR profile defines all the constraints on a resource, including coding system and value set. Terminology services can also provide the means of defining the valid value set. Therefore, it is common that the set of coding systems used by the real-time architecture fulfills the requirement for terminology services. However, when coding systems do not cover a particular domain, Terminology Services may resort to mapping by mapping the codes belonging to the missing domain with the available codes of other services.

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07




### 5.3. Data Integration and Interoperability Myramids

Real-time data exchange can provide a seamless integration experience for all the consumers of data. However, achieving the lowest latency across the spectrum of data consumers can be difficult, especially when specific consumers require combinations of data elements from multiple sources for their application to be executed. The canonical FHIR-based solution must be tested against such cases of real-time data integration to understand the challenges and the mitigation strategies.

The process of integrating data from multiple sources to create a more informative data structure is often referred to as data fusion or data integration. An early conceptual model for data integration defines a three-layer pyramid of data integration. The first layer consists of data sources that can produce base-level data primitives that are required for the integration. The next layer consists of data sources that perform data integration services on the previous layer's data. The top layer contains special-purpose data analysis systems that support specific applications, which can include areas such as knowledge discovery.

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

The general area of data integration services research is to study the integration service layer, which is often hard to implement and can incur a fair amount of latency. In the case of real-time systems, however, such services must adhere to the lowest possible level of latency for the highest-up layer that requires them. Unfortunately, such high-up layers also tend to implement a composite service, which is a service composed of multiple real-time services working at various latencies. For such composite services, an architectural pattern to perform data integration services without introducing extra latency should be used.

## 6. Performance Optimization Techniques

Interoperability frameworks define the interaction pattern and behavior of the integrated system, specify the quality of service (QoS) requirements, and enable monitoring and evolution of the composition. Quality metrics such as throughput and latency can be calculated and analyzed. Performance optimization techniques for improving throughput and lowering response times have been proposed. The impact of the optimization techniques is appraised by testing a SMART on FHIR service using the benchmarking protocol defined in Section 6.1. The QoS of the SMART on FHIR services appraised using the HI-Event framework for clinical alerts and notifications is also presented. Furthermore, optimization techniques that are orthogonal to throughput and latency are discussed.

Throughput refers to the number of transactions successfully completed in a given time window. For event-driven applications, it can be defined as the number of events successfully published within a specified period. Latency refers to the time elapsed between the instant an event was triggered and the instant it is processed by the subscriber. The latency bound of a subscriber can be defined as the maximum duration to deliver the event notification, including the time spent in the broker, the broker-to-subscriber delivery phase, and the subscriber processing time. Quality of service metrics for real-time systems consisting of a sender, broker, and receiver can be extracted from the behavior of the broker and the receiver. The proposed benchmarking protocol for these systems is composed of three main activities: 1) measuring the throughput of a health-information brokerage system, 2) applying a latency-calculating methodology, and 3) measuring the receiver processing duration.

**Equation 5) Benchmarking protocol as measurable equations**

12. measure throughput, 2) apply latency-calculating methodology, 3) measure receiver processing duration.


That corresponds to collecting:

- $X = N_{\text{success}}/T$
- $\{L_i\}$, then $\overline{L}$, $L_{\max}$ (or bounds)
- processing duration $t_{\text{proc}}^{(i)}$ directly (timer at receiver)

So "receiver processing duration" is simply the $t_{\text{proc}}^{(i)}$ term in the decomposition:

$$\boxed{t_{\text{proc}}^{(i)} = t_{\text{processed}}^{(i)} - t_{\text{received}}^{(i)}}$$

### 6.1. Throughput, Latency, and Quality of Service Metrics

Throughput, latency, latency bounds, quality of service metrics, and benchmarking protocols are defined. Quantitative evaluation of the decision support service is realized through the throughput measure, with average values indicating the task-granularity scale for the number of open workflows. Throughput gain and drop ratio allow the tuning of the number of requests issuing processes to optimize for throughput.

There is a large emphasis placed on minimizing latency values. The average latency is heavily influenced by the execution time of the called backend service. Considering minimum latency values is related to testers' positioning in the clinical decision support service network and to the chosen strategy to perform the requested HTTP requests. Nevertheless, under optimal conditions, latency values are sufficiently small to allow real-time integration with any clinical service. Latency bounds, therefore, represent more relevant performance metrics.

### 6.2. Caching, Batching, and Delta Synchronization

Caching improves performance in many places, including buffering results of expensive computations. In FHIR frameworks, the main caching opportunity is on the server side (view caching) in response to read requests. The FHIR specification mentions using view caching in FHIR servers to optimize responses when clients request the same resource repeatedly. The specification also provides a guiding principle regarding the frequency of refreshing view caches: "the more frequently the resource changes, the shorter the lifetime of the cached resource

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

should be." If resources do not change frequently, it can be efficient to cache the retrieved resource with a long expiration period (e.g., one day), as most reads will be satisfied by the cache. However, update requests ("write" operations in REST terminology) must ensure that the cache is invalidated appropriately.

Batching reduces network overhead when multiple items are exchanged. When many resources are created or updated at once, batching allows them to be sent in a single request instead of a separate request for each resource. Batching is first supported in the operation out capability and is commonly used in the import process when a client pushes resources to a server. Delta synchronization allows the client to synchronize only the resources that have changed since the last synchronization, rather than sending the entire dataset. Delta synchronization is critical in low-bandwidth scenarios because sending the whole dataset requires very large payloads relative to the bandwidth between the parties. The major requirement of delta synchronization is maintaining a log (e.g., a stable data structure) that allows querying which resource has changed since a specified point in time. If the application domain can tolerate a certain level of inconsistency (e.g., a decision-support application can use an inconsistent copy of the dataset), it is possible to avoid the substantial overhead of maintaining a stable log.

### 6.3. Efficient Resource Retrieval and Paging Strategies

Efficient retrieval and paging strategies reduce response times to data-access requests. Adopting a appropriately-scaled HTTP infrastructure and tuning the back-end data store can maintain low latency even at high loads. However, when fetching a single resource involves only a small subpart of the total request-response volume, it can be beneficial to introduce methods that also relieve the HTTP infrastructure of some of the work. Resource read requests and pagination access patterns are two scenarios where latency optimizations translate into real-world benefits.

Efficient access patterns are crucial for paginated requests. A Delta-Page approach adapts Paging to Delta- Syncing, sending a series of small requests that collectively cover the page of interest, each fetching a small set of entries. The advanced access method can trigger a synchronous batch retrieval of several pages with the back-end data store, effectively turning it into a BatchDelta-Sync operation.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

Despite the widespread use of paging combined with low-priority caching on the server side, this access pattern is not natively optimized by HTTP. For classical paging, a dedicated method (GET /{resource type}/{id}/_paginated) minimizes the time taken to serve the “page N” requests, while keeping the page size constant. What the dedicated method does is prefixing the */_paginated* mapping and providing additional parameters available only for that specific method.

## 7. Conclusion

The proposed study contributes to a better understanding of frameworks supporting real-time healthcare data exchange. Specifically, it synthesizes the main clinical drivers and resulting data-provisioning requirements; describes Fast Healthcare Interoperability Resources, focusing on real-time aspects; identifies architecture patterns supporting real-time exchange; aligns proposals with the larger FHIR ecosystem; and identifies techniques to optimize real-time performance.

The increasing need for specific data provisioning, especially for actionable decision-support systems, makes the ability to respond to multiple interdependent demands in real time paramount. This need cannot be deferred, as it ultimately concerns patient welfare and safety. Service-level agreements in corporate environments shape response-time expectations, especially when mere negligence can lead to the loss of life. Furthermore, existing systems are expected to evolve, with these new features being integrated seamlessly following the same interoperability principles. The health sector is thus called to find a solution to this multidirectional query in a real-time context. In the broader context of real-time medicine, FHIR offers many features that support the implementation of event-based dynamic data exchange. Common clinical architectures, clinical decision-support systems, terminology alignment, and the use of FHIR-provided back-end services are therefore discussed in detail, alongside how associated frameworks respond to real-time performance.

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
ISSN: 3067-1140
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

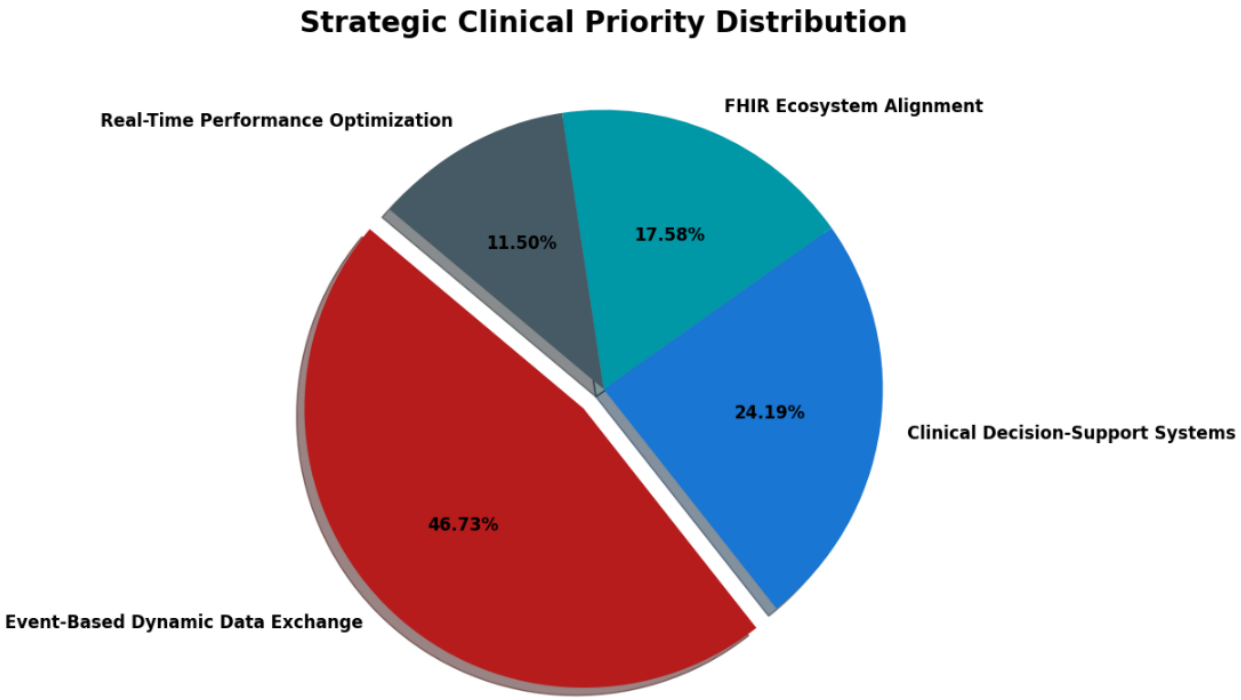



**Fig 4: Strategic Clinical Priority Distribution**

### 7.1. Final Thoughts and Future Directions

When evaluating the performances in real-time healthcare data interchange, it is important to consider the presence of two key components: the role of event-based systems and the architecture pattern employed. Concerning the first factor, real-time communication requires that event signaling occurs the soonest possible, attaining a history-less latency, which is reflected in the formal definitions of the events. In order to reach such bounds, one possibility is the moment-by-moment establishment of event miss such that produced events are signaled as soon as possible. Regarding the architecture pattern, it is often important to present the data in a localized manner, especially when one is subscribing to the events. In such cases, an events broker, when properly configured and deployed, can distribute events to interested listeners in an efficient way.


Nevertheless, given the multi-organizational nature of real-time healthcare data exchange and the practical interest of hiding, adapting, and transforming communication protocols, a mediation layer is often required. The study has focused on such a layer. Real-time performance – the time needed to deliver the first and all notifications – has received special attention, and modeling and caching solutions to mitigate the time-to-first notification – an important Quality of Experience (QoE) metric – have been proposed and evaluated experimentally. The analysis has suggested that, although the media can be made dynamically aware of the notification subscriptions, the actual constraints of the media and the data sources should still be taken into account in order to efficiently support communication.

**References**

[1] Mandel, J. C., Kreda, D. A., Mandl, K. D., Kohane, I. S., & 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), 899–908.
[2] Benson, T., & Grieve, G. (2021). Principles of Health Interoperability: SNOMED CT, HL7 and FHIR (3rd ed.). Springer.
[3] HL7 International. (2024). HL7 FHIR Release 5 Specification. Health Level Seven International.
[4] Decker, M., Lhotska, L., Kubicek, J., & Suchanek, M. (2021). Event-driven architectures for healthcare interoperability. Computer Methods and Programs in Biomedicine, 200, 105930.
[5] Unertl, K. M., Jaffa, H., Field, J. R., Price, L., Peterson, K. J., & Richardson, J. E. (2020). SMART on FHIR implementation guide: Enhancing clinical decision support. Applied Clinical Informatics, 11(5), 746–754.
[6] Kamel Boulos, M. N., Brewer, A. C., Karimkhani, C., Buller, D. B., & Dellavalle, R. P. (2019). Mobile medical and health apps: State of the art, concerns, regulatory control. Journal of Medical Internet Research, 21(3), e13369.
[7] Groppe, J., & Groppe, S. (2020). Real-time data integration in healthcare using publish-subscribe systems. Future Generation Computer Systems, 108, 1042–1055.
[8] Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record analysis. IEEE Journal of Biomedical and Health Informatics, 22(5), 1589–1604.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

[9] Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., & Wang, Y. (2017). Artificial intelligence in healthcare: Past, present and future. Stroke and Vascular Neurology, 2(4), 230–243.
[10] Abouelmehdi, K., Beni-Hessane, A., & Khaloufi, H. (2018). Big healthcare data: Preserving security and privacy. Journal of Big Data, 5(1), 1–18.
[11] Martínez-García, A., Escudero, J. M., & García-Saiz, D. (2022). Cloud-based interoperability frameworks for healthcare IoT. IEEE Access, 10, 34421–34434.
[12] Rumbold, J. M. M., & Pierscionek, B. K. (2017). The effect of the general data protection regulation on medical research. Journal of Medical Internet Research, 19(2), e47.
[13] O'Connor, M. J., & Das, A. K. (2014). A methodology for modeling clinical knowledge in SNOMED CT. Journal of Biomedical Informatics, 50, 178–187.
[14] Payne, T. H., Corley, S., Cullen, T. A., Gandhi, T. K., Harrington, L., Kuperman, G. J., Mattison, J. E., McCallie, D., McDonald, C. J., Murphy, S., et al. (2018). Report of the AMIA EHR 2020 task force. Journal of the American Medical Informatics Association, 22(5), 110–112.
[15] Yu, F., Bilberg, A., & Rasmussen, J. (2021). Microservices architecture for healthcare systems. Procedia Computer Science, 180, 388–397.
[16] Haux, R. (2010). Medical informatics: Past, present, future. International Journal of Medical Informatics, 79(9), 599–610.
[17] Behl, A., & Behl, K. (2017). Cyberwar: The next threat to national security and what to do about it. Oxford University Press.
[18] El-Sappagh, S., Ali, F., Hendawi, A., & Kwak, K. S. (2020). A mobile health monitoring-and-treatment system based on integration of the Internet of Things and cloud computing in medical applications. IEEE Access, 8, 19877–19898.
[19] Sittig, D. F., & Singh, H. (2016). A socio-technical approach to preventing, mitigating, and recovering from EHR-related safety hazards. Journal of the American Medical Informatics Association, 23(4), 641–647.
[20] Bender, D., & Sartipi, K. (2013). HL7 FHIR: An agile and RESTful approach to healthcare information exchange. Proceedings of the IEEE 26th International Symposium on Computer-Based Medical Systems.
[21] Zhang, Y., Qiu, M., Tsai, C. W., Hassan, M. M., & Alamri, A. (2017). Health-CPS: Healthcare cyber-physical system assisted by cloud and big data. IEEE Systems Journal, 11(1), 88–95.
[22] Kuo, A. M. H. (2011). Opportunities and challenges of cloud computing to improve health care services. Journal of Medical Internet Research, 13(3), e67.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

[23] Lehne, M., Luijten, S., Vellinga, J., & Chablis, G. (2019). The use of FHIR in digital health. Studies in Health Technology and Informatics, 267, 52–58.
[24] Alshahrani, A., & Sheikh, A. (2020). Interoperability challenges in healthcare information systems. Journal of Healthcare Engineering, 2020, 1–13.
[25] Belle, A., Thiagarajan, R., Soroushmehr, S. M. R., Navidi, F., Beard, D. A., & Najarian, K. (2015). Big data analytics in healthcare. BioMed Research International, 2015, 1–16.
[26] Perera, C., Liu, C. H., & Jayawardena, S. (2015). The emerging Internet of Things marketplace. IEEE Cloud Computing, 2(4), 50–58.
[27] Chen, M., Ma, Y., Li, Y., Wu, D., Zhang, Y., & Youn, C. H. (2017). Wearable 2.0: Enabling human-cloud integration in next generation healthcare systems. IEEE Communications Magazine, 55(1), 54–61.
[28] Gagnon, M. P., Ghandour, E. K., Talla, P. K., Simonyan, D., Godin, G., Labrecque, M., & Ouimet, M. (2014). Electronic health record acceptance by physicians. Journal of Medical Systems, 38(2), 1–10.
[29] Kouroubali, A., & Katehakis, D. G. (2019). The new European interoperability framework as a facilitator of digital transformation for citizen-centric public services. Journal of Biomedical Informatics, 94, 103166.
[30] Hashmi, M. F., Katiyar, S., Keskar, A. G., Bokde, N. D., & Geem, Z. W. (2020). Efficient healthcare data management using machine learning and cloud computing. Sensors, 20(15), 4238.
[31] Zhang, P., White, J., Schmidt, D. C., Lenz, G., & Rosenbloom, S. T. (2018). FHIRChain: Applying blockchain to securely and scalably share clinical data. Computational and Structural Biotechnology Journal, 16, 267–278.
[32] Roehrs, A., da Costa, C. A., & Righi, R. R. (2017). OmniPHR: A distributed architecture model to integrate personal health records. Journal of Biomedical Informatics, 71, 70–81.
[33] Hoy, M. B. (2016). Personal health records. Medical Reference Services Quarterly, 35(3), 213–224.
[34] Abidi, S. S. R., & Abidi, S. R. (2007). Ontology-based modeling of clinical pathways. Journal of Biomedical Informatics, 40(6), 722–738.
[35] Chen, Y., Argentinis, J. D., & Weber, G. (2016). IBM Watson: How cognitive computing can be applied to big data challenges. IBM Research White Paper.
[36] Raghupathi, W., & Raghupathi, V. (2014). Big data analytics in healthcare. Health Information Science and Systems, 2(1), 1–10.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

[37] Meystre, S. M., Lovis, C., Bürkle, T., Tognola, G., Budrionis, A., & Lehmann, C. U. (2017). Clinical data reuse or secondary use. Yearbook of Medical Informatics, 26(1), 54–62.
[38] Koufi, V., & Malamateniou, F. (2015). Privacy-preserving solutions in eHealth. Studies in Health Technology and Informatics, 213, 245–248.
[39] Schreiber, G., Akkermans, H., Anjewierden, A., et al. (2000). Knowledge Engineering and Management. MIT Press.
[40] Behl, A., & Behl, K. (2019). Cybersecurity and cyberwar. Oxford University Press.
[41] Lian, J., Zhou, M., & Wang, F. Y. (2017). Cloud computing for electronic health records. IEEE Intelligent Systems, 32(3), 10–14.
[42] Jensen, P. B., Jensen, L. J., & Brunak, S. (2012). Mining electronic health records. Nature Reviews Genetics, 13(6), 395–405.
[43] Goldstein, M. M., & Thorpe, J. H. (2010). The first anniversary of the HITECH Act. Health Affairs, 29(6), 1042–1047.
[44] Wilkowska, W., & Ziefle, M. (2012). Privacy and data security in e-health. Health Policy and Technology, 1(3), 119–131.
[45] Kocabas, O., Soyata, T., & Aktas, M. K. (2018). Emerging security mechanisms for medical cyber-physical systems. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 15(3), 866–880.
[46] Al-Janabi, S., Al-Shourbaji, I., Shojafar, M., & Shamshirband, S. (2017). Survey of main challenges in smart healthcare. IEEE Access, 5, 6765–6776.
[47] Bui, N., & Zorzi, M. (2011). Health care applications. IEEE Communications Magazine, 49(1), 110–117.
[48] Ristevski, B., & Chen, M. (2018). Big data analytics in medicine and healthcare. Journal of Integrative Bioinformatics, 15(3).
[49] Khan, F. A., Sadiq, S., & Qureshi, M. A. (2020). Interoperable IoT-based healthcare system. IEEE Access, 8, 228482–228497.
[50] Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics. MIS Quarterly, 36(4), 1165–1188.
[51] Devarakonda, M., Kella, K., & Sarma, A. (2018). Healthcare interoperability using FHIR and RESTful web services. International Journal of Engineering & Technology, 7(3), 159–164.
[52] Khezr, S., Moniruzzaman, M., Yassine, A., & Benlamri, R. (2019). Blockchain technology in healthcare. IEEE Access, 7, 159–175.
[53] Zhou, L., & Parmanto, B. (2019). Reaching people with disabilities through accessible health information technology. Journal of Medical Internet Research, 21(2), e12222.

ISSN: 3067-1140
AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING
VOLUME: 03 ISSUE: 03
RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

[54] Boehm, B. (2006). A view of 20th and 21st century software engineering. Proceedings of ICSE.
[55] Laplante, P. A., & Neill, C. J. (2004). The demise of the waterfall model. ACM Queue, 1(10), 10–12.
[56] Fielding, R. T. (2000). Architectural styles and the design of network-based software architectures. Doctoral dissertation, University of California.
[57] Pahl, C. (2015). Containerization and microservices. IEEE Cloud Computing, 2(3), 24–31.
[58] Villamizar, M., Garcés, O., Castro, H., et al. (2016). Evaluating microservices architecture. Journal of Internet Services and Applications, 7(1), 1–20.
[59] Dragoni, N., et al. (2017). Microservices. Springer.
[60] Newman, S. (2015). Building microservices. O'Reilly.
[61] Erl, T. (2017). Service-oriented architecture. Prentice Hall.
[62] Tanenbaum, A. S., & Van Steen, M. (2017). Distributed systems. Pearson.
[63] Bass, L., Clements, P., & Kazman, R. (2013). Software architecture in practice. Addison-Wesley.
[64] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design patterns. Addison-Wesley.
[65] Fowler, M. (2018). Patterns of enterprise application architecture. Addison-Wesley.
[66] Dean, J., & Ghemawat, S. (2008). MapReduce. Communications of the ACM, 51(1), 107–113.
[67] Zaharia, M., et al. (2016). Apache Spark. Communications of the ACM, 59(11), 56–65.
[68] Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka. Proceedings of NetDB.
[69] Akidau, T., et al. (2015). The dataflow model. Proceedings of VLDB.
[70] Stonebraker, M., et al. (2005). The 8 requirements of real-time stream processing. ACM SIGMOD Record, 34(4), 42–47.
[71] Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things. Future Generation Computer Systems, 29(7), 1645–1660.
[72] Buyya, R., Broberg, J., & Goscinski, A. (2010). Cloud computing. Wiley.
[73] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing. NIST.
[74] ISO/IEC 27001. (2022). Information security management systems.
[75] ISO/IEC 27799. (2016). Health informatics security management.
[76] World Health Organization. (2021). Global strategy on digital health 2020–2025.
[77] OECD. (2023). Health data governance.
[78] European Commission. (2022). European Health Data Space proposal.

ISSN: 3067-1140

AMERICAN ONLINE JOURNAL OF SCIENCE AND ENGINEERING

VOLUME: 03 ISSUE: 03

RECEIVED: JULY 23 REVISED: AUGUST 10 ACCEPTED: AUGUST 25 PUBLISHED: SEPTEMBER 07

[79] National Academy of Medicine. (2021). Digital health interoperability.
[80] HIMSS. (2024). Interoperability in healthcare report.
[81] Topol, E. (2019). Deep medicine. Basic Books.
[82] Shortliffe, E. H., & Cimino, J. J. (2021). Biomedical informatics. Springer.
[83] McGinnis, J. M., et al. (2014). Best care at lower cost. National Academies Press.
[84] Stead, W. W., & Lin, H. S. (2009). Computational technology for effective health care. National Academies Press.
[85] Bates, D. W., Cohen, M., Leape, L. L., et al. (2001). Reducing adverse drug events. Journal of the American Medical Informatics Association, 8(4), 299–308.
[86] Berwick, D. M., Nolan, T. W., & Whittington, J. (2008). The triple aim. Health Affairs, 27(3), 759–769.
[87] Mandl, K. D., & Kohane, I. S. (2015). No small change for the health information economy. New England Journal of Medicine, 372(14), 1278–1281.
[88] Greenhalgh, T., et al. (2017). Beyond adoption: A new framework. Journal of Medical Internet Research, 19(11), e367.
[89] Sinsky, C., et al. (2016). Allocation of physician time. Annals of Internal Medicine, 165(11), 753–760.
[90] Adler-Milstein, J., & Jha, A. K. (2017). HITECH Act. Health Affairs, 36(5), 789–795.