International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Bridging the Gap in Integrated Care: A Semantic Medical Data Management Framework Leveraging Linked Data and FHIR Standards

Wasim Fathima Shah

Independent Researcher
Email: *wasimfathima786[at]gmail.com*

Abstract: ***In this paper, we present a novel telehealth data management framework specifically tailored to enhance integrated care services for patients facing chronic and multimorbid conditions. This innovative framework employs an OWL ontology, which is firmly rooted in HL7 FHIR resources, to facilitate the storage and representation of semantically enriched EHR data while adhering to Linked Data principles. We provide a comprehensive overview of the implementation of a persistent storage solution and communication web services, which enable efficient EHR data management and ensure the preservation of validity and integrity of patient data as self-describing ontology instances. One of the key features of this framework is its emphasis on flexibility and reusability. By considering ontology as the central point of change, the framework can easily adapt to different integrated care scenarios and requirements. The development of this telehealth data management framework is a timely response to the growing need for effective and integrated care services, especially for patients with complex and chronic health conditions. By leveraging cutting-edge technologies and data management strategies, this framework offers a robust, adaptable, and user-friendly solution for healthcare providers, ultimately contributing to improved patient outcomes and more efficient healthcare delivery.***

Keywords: Telehealth Data Management; Integrated Care Services; HL7 FHIR Resources; Semantically Enriched EHR Data; Linked Data Principles.

1. Introduction

Integrated care is a patient-centered care model that emphasizes coordination among various levels of care management, services, and collaboration among professionals, even within separate organizations [1]. This approach focuses on the continuum of healthcare delivery for patients and populations, particularly the prevention and management of chronic diseases and multiple morbidities [1]. Integrated care implementation can be context-specific, influenced by policies, legislations, and public-private partnerships, with the goal of providing high-quality, efficient care for better health outcomes while controlling costs.

Effective data management is essential in integrated care solutions, which involve multiple data sources, access points, and computational workflows. This necessitates cloud data storage, management, and security to support the storage and retrieval of diverse sensor data, biosensor data, user reports, metadata, and decision support system outputs [3-5]. Although current state-of-the-art standards and technologies can support integrated medical data management, modern systems only partially adhere to them [6, 7].

A contemporary framework for medical data management must address several challenges to be embraced by developers and knowledge management experts, contributing to the integrated care landscape. It should comply with current standards and best practices, be easy to maintain and update, and seamlessly manage various data types. Moreover, it must be flexible enough to cater to the needs of standard EHR systems, PHRs, Telehealth systems, and clinical trial support systems with minimal modification [8]. This framework should leverage cloud computing advantages, provide unrestricted storage with native integrity checking, and facilitate standardized data exchange with external applications. Additionally, it should serve as the foundation for knowledge discovery and learning health systems development, relying on Linked Data principles [9].

We propose a flexible, semantic-based framework for medical data management that offers Restful web services and the following benefits:

a) Easy manipulation of managed concepts by altering the semantic model for both new deployments and existing frameworks.
b) A single point of maintenance, ensuring maintainability and sustainability without the need for changes on multiple layers or re-compilation/re-deployment during modifications.
c) Adherence to modern health informatics standards and Linked Data principles, allowing for the use of stored information in ways not initially anticipated during system development.

This paper presents the rationale behind this work, defines design goals, and compares current state-of-the-art solutions with the proposed framework. It also discusses relevant modern technologies and outlines the framework's characteristics. Finally, the paper showcases the deployment results of the framework in two paradigm domains.

2. Literature Survey

In this part, we outline the goals that served as a guide for the creation of the framework. These goals embody the necessities for designing and executing a framework that facilitates data management for integrated care, founded on Linked Data, HL7 FHIR, and Ontologies. Besides the requirements stemming from best practices for integrated care data managementwe also introduce extra goals to achieve our aims towards crafting a framework surpassing state-of-the-art regarding flexibility, reusability, and semantic interoperability.

2.1 Background and Significance:

a) Officially delineate/depict the data model entities pertinent to telehealth and integrated care fields.

Usually, identifying the relevant entities of a domain constitutes the initial step for any system managing data. A data model represents the organization and relationships of data corresponding to a domain's relevant entities this field, we must consider that, beyond standard medical record data on focal diseases, multiple entities relate to portable/wearable device measurements and secondary calculated parameters, some resulting from telehealth device-recorded biosignal processing. An example includes incorporating arrhythmia characterization, derived from home-based ECG analysis, into the data model. Moreover, since it's nearly impossible to define a comprehensive, concrete data model in one attempt, we should enable future modifications and additions, minimizing risks to the existing model and the overlying framework layers.

b) Continuously store domain-related medical record and telehealth data entities.

Data storage should enable retrieval by applications other than the creators, necessary in integrated care where multiple users and entry points contribute to chronic and comorbid patient management. This process demands speed and consistency, particularly when linked to decision support procedures. The chosen persistent storage solution must not rely on a specific architecture (i.e., vendor/product) or file system, ensuring long-term system viability and data repurposing capability.

c) Offer a web service for robustly handling and exchanging domain data.

The framework needs to supply a web service facilitating data exchange and modification remotely through a standardized method. The web service's endpoints should adhere to familiar guidelines for potential client developers. The exchanged messages' format must eliminate the need for complex or entity-specific transformations before data storage. High web service simplicity is a complementary requirement for a robust data management and exchange framework.

d) Establish, formally represent, and enforce rules ensuring valid and meaningful stored information.

To guarantee the accuracy and reliability of persistently stored data, the framework should enable formal representation of integrity constraints (defined by domain experts) and a robust enforcement method. Domain, Entity, and Referential integrity [10] must be accessible as both definitions and functional constraints. Data quality is crucial in healthcare, especially when produced in telehealth and various unattended settings. Defining data integrity rules necessitates collaboration between knowledge management and domain experts (e.g., pulmonologist, cardiologist).

e) Comply with interoperability standards as much as possible.

Healthcare data management encompasses numerous proposed solutions and knowledge accrued over years of research. Standard usage is nearly essential in all medical systems. The HL7 standard suite is arguably the most recognized standardization effort in health informatics. HL7 FHIR2, a modern and leading open standard for interoperability, is ideal for inspiring and providing best practices concerning the domain.

f) Enable semantic interoperability of data using Linked Data principles.

In chronic care and comorbidity management project, complex information flows are automatically activated upon data reception, processing data and generating features. These combine as inputs for patient decision support and the coordinated healthcare professional team [11]. Thus, data must be integrated and actionable for regular system use. Additionally, data from coordinated care and telehealth applications need harmonization before usage for other purposes, such as research. To facilitate future functionalities, a system can inherently adhere to Linked Data principles, minimizing manual concept mapping and complex data transformations. These principles advocate using Uniform Resource Identifiers (URIs) for unique resource identification and defining exchanged data to include not only actual values but also relationships among them.

g) The web service facilitating communication should be auto-deployed, concept-agnostic, and promote framework flexibility and reusability.

To increase the framework's adaptability to changes, the web service must not necessitate source code alterations for common data model modifications. Specifically, adding, removing, or updating model concepts should reflect on web service endpoints without re-compilation. This must apply to both new and existing deployments. Since incorporating additional entities and semantics from post-deployment emerging requirements is common, shifting modification costs to the semantic data model layer reduces overall effort, resulting in a maintainable and sustainable framework. Designing the web service to be concept-agnostic for exchanged data allows redeployment in related domains, making it reusable in other applications.

h) Support the framework with free and open-source tools.

To offer a solution adoptable by the community without constraints, the framework must rely on freely available technologies. This protects the framework from vendor lock-in practices and establishes a reusable and sustainable system foundation. The design goals targeted a robust telehealth data management scaffold, maintaining flexibility for post-deployment modifications and the ability to be repurposed in different related domains. Flexibility and reusability support must involve limited effort and not require changes across all system layers. Lastly, the intended framework must inherently be interoperable through basic-level compliance with accepted standards (e.g., data types), not through transformation operations that might cause fidelity loss.

3. Problem Definition

In this section, we provide an overview of similar data collection and integration approaches applied in the context of recent or ongoing research projects, focusing on Integrated Care scenarios. The presented overview focuses on the respective projects' use cases, data modeling and management approach, application of widely accepted interoper ability standards, and use of semantic data annotation capabilities. Also, we present a review of the state-of-the-art technical solutions and frameworks that are targeted at flexible management of data. UniversAAL is an open source platform providing an interoperability framework upon which individual applications could communicate to exchange data in specific use cases [12], focused on Ambient Assisted Living (AAL) scenarios, including health data monitoring.3 UniversAAL was initially an EU-funded research project and its data model was based on the so-called Consolidated Reference Model providing the application specific semantics organized as an underlying conceptual model. In its current version, the UniversAAL project pro vides an open source platform based on a Service Oriented Architecture which engages ontologies in the form of RDF as the main semantic interoperability layer among the various services.4 In the context of the in CASA European project, a telemonitoring system has been developed focusing on elderly chronic patients [13] and monitoring both activity and physiological data, providing data analysis and alerts. The inCASA telemonitoring framework employs IEEE 11073 medical device standards over ZigBee5 for communication from the sensor to the local gateway and IHE PCD-01 HL7 profile to communicate data to the respective service through General Packet Radio Service (GPRS). Transmitted data include peripheral oxygen sa turation (SpO2), blood pressure, activity data, etc., and they are transmitted as HL7 (version 2) messages exchanged through a WSDL interface defined by IHE.6 The inCASA technical implementation en gaged Semantic Web technologies (e.g. SPARQL and SWRL) and rea soning upon semantically annotated data to produce alerts and notify the respective consumer applications. Respectively, the Dem@Care project developed a multi-sensor ac tivity telemonitoring solution to facilitate the timely diagnosis, assess ment, maintenance and promotion of self-independence of dementia patients7

The Dem@Care framework takes advantage of a Service Oriented Architecture approach, with client applications interacting with Dem@Care services through standard web service protocols (WSDL/SOAP) and employing a specific XSD schema to semantically identify communicated entities [14]. Additionally, Dem@Care utilizes Semantic Web technologies (e.g., ontologies) for semantically annotating data and leveraging automatic reasoning capabilities combined with rule-based SPARQL queries to identify distinct types of physical activities. C3 Cloud8 is a currently active EU-funded initiative, striving to create an ICT infrastructure that facilitates ongoing coordination of patient-centered care activities by a multidisciplinary care team and patients/informal caregivers. The project's central idea is the development of a personal care plan based on existing clinical guidelines. Information exchange is built on HL7 FHIR and REST web services to ensure interoperability with proprietary EHR systems implemented at pilot sites, following the HL7 Care Plan Domain Analysis Model (DAM). The onFHIR platform serves as the project's storage layer, offering secure FHIR-compatible storage on MongoDB, a popular NoSQL database.9, 10, 11 An "Integrated Terminology Server" is also provided as a core semantic infrastructure to enable semantic interoperability and furnish reasoning capabilities.12, 13 The OpSIT project, based in Germany, concentrated on the formal definition and synthesis of various healthcare processes using Business Process Modeling Notation (BPMN), employing telehealth through smart devices (e.g., a smartwatch) as its primary use case [15].

Service Oriented Architectures employing REST or SOAP paradigms were generally utilized to address syntactic interoperability. Concerning semantic interoperability, HL7 standards (now evolved to HL7 FHIR) and openEHR archetypes emerged as the leading message exchange paradigm. Semantic Web technologies have been extensively adopted as they offer both semantic interoperability and automatic reasoning capabilities. In terms of the data storage layer, relational databases and RDF triple stores have been used. However, most of the showcased approaches don't adhere to standards-based data management across all data processing levels (i.e., data exchange, storage, querying, reasoning). Conversely, for approaches exhibiting high compliance with standards (i.e., MobiGuide), semantics are focused in a single layer of their approach to support specific functionality, rather than being a fundamental aspect of the model, exchanged, and stored data. Additionally, the presented strategies and decisions made by the mentioned projects sufficiently cater to the specific program's data modeling and exchange requirements through defined communication APIs,15 but reusability in various deployments is not the focus for most of these efforts. In projects supporting extensibility via the addition of new concepts (i.e., MobiGuide), necessary modifications involve changes across multiple system layers. Regarding existing technical solutions aiming to provide flexible data management systems, OData16 is a method of standardizing how typical create, retrieve, update, delete (CRUD) operations are delivered via REST web services. OData, standardized by OASIS17, is considered the closest approximation to a de facto standard in prescribing a RESTful communication API. It outlines how such web services enable

CRUD and querying operations independently of the actual backend implementation.

Within the .NET ecosystem, data is typically stored in an SQL server backend, and an Entity Framework18 ORM model is constructed atop it to offer an object-oriented perspective of data. Subsequently, the EF model is utilized by the ASP.NET Web API services to expose CRUD operations on the client side. Similarly, in the Java ecosystem, tools like Jersey,19 Spring MVC20 are employed to supply the REST service layer, and an ORM such as Hibernate21 is utilized for providing an object-oriented data modeling, as data is usually stored in a relational DBMS. These implementation methods exhibit OData compatibility, are tested by a broad community, and deliver multilevel data validation (from REST input validation to SQL data integrity). However, they lack a single maintenance point and don't offer semantic interoperability support. If the data model undergoes a change, the entire software stack must be recompiled. Using the .NET stack as an illustration, adding a table to the database requires rebuilding the EF model and manually adding the respective Web API service endpoint. While mainstream scaffolding implementations support numerous formats, it is surprising that they have not yet embraced Linked Data principles and associated technologies such as RDF.

3.1. Terminologies and Standards

A semantic data model not only describes the meaning of its instances but also enables the interpretation of the meaning directly from the instances. This enhancement, along with the expressiveness provided by semantic tools, makes ontologies a modern way of describing domain data models. Several documented attempts have been made to use ontologies as data models for supporting EHR data, personalized care of chronic diseases, and Decision Support Systems. The Web Ontology Language (OWL) is a family of knowledge representation languages used for authoring ontologies. Ontologies serve as a formal method for describing taxonomies and classification networks, essentially defining knowledge structures for various domains. OWL DL, a sublanguage of OWL, supports users seeking maximum expressiveness without sacrificing computational completeness and decidability of reasoning systems.

One of the most recognized standards in health informatics is HL7, an organization dedicated to providing a comprehensive framework and related standards for exchanging, integrating, sharing, and retrieving electronic health information to support clinical practice and the management, delivery, and evaluation of health services. HL7 FHIR (Fast Healthcare Interoperability Resources) is a new HL7 standard for exchanging electronic health records. Building on previous HL7 data format standards, FHIR incorporates more modern technological concepts and approaches, aiming to be more developer-friendly. FHIR solutions consist of modular components called "Resources" that can be easily assembled to address real-world clinical and administrative problems. It is essential to note that FHIR is an open standard and, during the framework implementation phase, was published as a Draft Standard for Trial Use (DSTU).

Health ontologies and terminologies are widely used in healthcare information systems. One of the most user-friendly and freely accessible repositories of health-related ontologies is the National Center for Biomedical Ontology's BioPortal. BioPortal includes search and representation mechanisms for various health ontologies and terminologies, such as SNOMED Clinical Terms, International Classification of Diseases (ICD), Logical Observation Identifier Names and Codes (LOINC), and the World Health Organization's Anatomical Therapeutic Chemical (ATC) classification system, among others. BioPortal provides Persistent Uniform Resource Locators (PURLs) for all concepts defined in the ontologies it hosts. PURLs are web addresses or Uniform Resource Locators (URLs) that serve as permanent identifiers in the face of a dynamic and changing web infrastructure.

3.2 Persistent Storage and Data Integrity

Regardless of their origin, scope, or size, all relevant data can be classified into three main categories: Structured Information, Binary Large Objects (BLOBs), and BLOB-related metadata. Storing BLOBs is relatively simple, with various solutions available, from storing flat files on a server's file system accessed through a basic custom-built web service to more advanced approaches using commercial file storage cloud solutions like Amazon S3, Microsoft Azure BLOB Storage, or Google Cloud Storage. However, the main concern is choosing a storage engine for handling structured information data. The primary options for a storage engine are:
- Relational SQL databases
- NoSQL databases

Relational databases are the most common solution for data storage in enterprise systems but have some drawbacks. Maintaining a changing data model in a relational database is challenging, and they lack built-in support for semantic web technologies, requiring implementation outside the storage layer. NoSQL databases, a term covering various database technology families, differ from traditional relational databases in their data storage and retrieval mechanisms. These families include key-value stores, key-document stores, column-family stores, and graph stores. NoSQL databases are typically cluster-friendly, schema-less, and lack model definition and integrity checking mechanisms. They also implement custom, non-standardized ways to access data, resulting in vendor lock-in.

However, RDF Triple Stores are an exception. These graph databases store triples of information, with a triple consisting of subject-predicate-object. Triples can be imported and exported using queries, RDF, and other formats. Triple stores offer several benefits, such as a standardized, vendor-independent way of accessing data (SPARQL for querying and RDF serializations for data exchange), easy integration into a service-oriented architecture, and easier maintenance of a changing data model while natively supporting semantic interoperability. RDF triple stores comply with various W3C standards related to RDF, providing a highly flexible solution. Swapping one storage engine for another is simple and requires minimal changes to the existing codebase or architecture.

OpenLink Virtuoso Universal Server is open-source, offers commercial support if needed, and supports cluster setups. Studies have shown that Virtuoso, when used as a storage engine in an integrated Web Services approach based on RDF data, performs as quickly as traditional relational databases. Virtuoso also outperforms other SPARQL-enabled and semantic repositories in real-world or generated datasets. It is important to note that semantic repositories operate under the Open World Assumption, where the absence of a statement doesn't necessarily mean it's false. RDF Triple Stores don't provide a standard way to enforce constraints like traditional relational databases. This is partly due to their historical use as views of data stored in a relational database, making constraint enforcement the database engine's responsibility. The introduction of the SPARQL 1.1 UPDATE language has made it possible to use RDF Triple Stores as storage engines, leading to various solutions for constraint checking, making them more similar to relational databases. Examples include OWL Restrictions, Resource Shapes (ReSh), Shape Expressions (ShEx), Description Set Profiles (DSPs), Stardog ICV, Pellet ICV, SPARQL, and SPIN (SPARQL Inferencing Notation). SPIN is a W3C Member Submission, has an open-source Java API, and is supported by a commercial entity (TopQuadrant Inc.) that actively supports it. In SPIN, standard SPARQL is used to formulate constraints, and the SPIN API checks RDF data against these constraints.

4. Methodology

Methods

The proposed framework is comprised of three interconnected components: (a) the underlying data model, implemented as an OWL-DL ontology, (b) the integrity mechanisms and persistent storage solution, and (c) the dynamic RESTful web service interface that exposes data management functionalities. This section outlines the main techniques and software tools employed in designing and developing the scaffolding framework.

4.1 Data Model Development and Definition

The data model is built upon (a) semantic technologies, (b) HL7-FHIR, and (c) medical terminologies. The process of integrating these technologies into the data model is described. We followed a methodology [36] to define and implement the semantic data model, which involved several steps to construct an ontology:

a) Clearly identify and delineate the ontology's domain.
b) Determine the concepts and their relationships within this domain, and express this information in a representational artifact.
c) Ensure logical, philosophical, and scientific coherence, compatibility with relevant ontologies, and human intelligibility.
d) Formalize and implement the representational artifact in a computer language within a specific computing context.

To implement this methodology, we addressed each step as follows:

a) We decided that the ontology should represent entities required for storing medical record data and telehealth data related to COPD, CHF, diabetes, and depression. The chosen entities were represented as ontology classes.
b) The concepts were initially represented as a simple list of domain entities in a table. Then, the relationships between them were defined, along with necessary restrictions to ensure that the stored information is valid and meaningful.
c) Logical and scientific coherence was achieved through iterative review of the defined relationships by knowledge management experts and domain experts (pulmonologists and cardiologists).
d) The representation language used was OWL-DL.

4.1.1. The Semantic Model

The resulting ontology is detailed in [37]. To maintain the coherence of this paper, we will only discuss the main characteristics. In this ontology, domain entities are represented as OWL classes, while instances of these classes correspond to the actual data managed by the framework.

4.2. Persistent Storage and Integrity Mechanisms Development and Definition

OpenLink Virtuoso Universal Server was chosen as the persistent storage solution for our framework, while SPARQL GraphStore Protocol [39] was selected as the internal communication protocol between the data management web service and Virtuoso. A one graph per resource approach was used, and SPIN was employed to represent the required functional restrictions on the stored data.

Virtuoso was used not only for the persistent storage of exchanged data but also for the persistent storage of the ontology. Although Virtuoso was the choice for an RDF Triple Store, it would require minimal additional effort to switch to another one in the future if newer, better-performing triple store technologies emerge. This is due to specific design choices, such as not using vendor-specific SPARQL functions or non-prescribed SPARQL query forms, and ensuring communication is strictly over standard SPARQL HTTP protocols.

After reviewing recent [40,41] and older work [42], we chose to use SPIN for enforcing constraints when storing data in an RDF Triple Store, as it seemed to offer maturity, accessibility, integration with our development toolset, and a relatively gentle learning curve. SPIN rules were defined and used to validate and enforce value-level constraints.

Listing 237 above provides an example of the Device class definition. When a device is assigned to a patient, the patient's URI must already be present in the triple store, and no two devices can share the same identifier. These constraints are converted into SPARQL queries at runtime and executed against the triple store's SPARQL endpoint.

4.3. Web Service Interface Definition and Construction

The top layer of our framework's architecture involves communication with service consumers through an application programming interface (API). A RESTful API, adhering to best practices for REST, was implemented for this purpose. REST is popular, supports various security methods, and has open-source client implementations for almost all platforms.
Structured information is exchanged in the form of RDF graphs, serialized in Turtle, and stored in an RDF Triple Store. Turtle was chosen due to its compactness, readability, and ease of development and debugging for client applications.

Figure 3 provides an overview of this software architecture stack. The REST API comprises three main services: (a) Ontology/Schema service, (b) RDF Resources Service, and (c) Files service. These services and their endpoints follow general best practices for REST, taking into account REST APIs for semantic technologies and the HL7 FHIR API.

a) The Ontology/Schema service handles storage and retrieval of the framework's ontology. Endpoints allow administrators to store and update the data model (ontology), and application developers can access the data model for designing, implementing, and maintaining their applications. As the data model evolves, client applications can dynamically calculate and display information based on the stored model.

Table 1: CRUD Paradigm

Endpoint	HTTP Verb	Action	Result
/data/{Resource Type}	POST	CREATE	Save a new resource of the specified {Resource Type} (e.g., /data/Patient to create a new Patient resource)
/data/{Resource Type}/{UUID}	GET	READ	Fetch a specific resource (e.g., /data/BodyTemperature/f0ad1dcb-7273-4ba7-9b13-9438e7963717 to retrieve a specific BodyTemperature resource)
/data/{Resource Type}/{UUID}	PUT	UPDATE	Modify a specific resource (e.g., /data/VitalSignsSession/f555ec19-841d-490c-8649-bc2a22e103e0 to update a specific VitalSignsSession resource)
/data/{Resource Type}/{UUID}	DELETE	DELETE	Remove a specific resource (e.g., /data/Patient/159feaee-df29-4ba0-8abb-56123dad7dbb to delete a specific Patient resource)
/data/{Resource Type}	GET	READ	Obtain a collection of resources of the specified {Resource Type} (e.g., /data/Patient to retrieve a list of Patient resources)

b) The RDF Resources Service manages storage and retrieval of structured data. Clients can create, retrieve, update, delete (CRUD paradigm), and search for resources. The base documentation for this service is simple enough to fit in Table 1. This endpoint is designed based on two key decisions: (1) dynamic, as it accepts resources based on the ontology, and (2) generic, as it can store different types of information in a homogeneous manner. The API consumer can perform search operations using a GET parameter.

c) The Files service handles binary files, allowing clients to store, retrieve, and delete binary large objects (BLOBs) such as EDF files for sensor data or PDF files from external reports.

The communication API features gzip compression for all API endpoints, supports ETag for HTTP caching, and manages concurrency control for resource update operations. The CRUD Layer is responsible for actual CRUD operations (Fig. 1) exposed by the REST API. API endpoints rely on corresponding method calls of this module to perform their operations. The CRUD Layer uses the SPIN API to evaluate SPIN rules (that enforce integrity) and a SPARQL/Graph Store Protocol Client to communicate with the Triple Store and file store.

Fig. 1 Architecture stack illustrates the components and layers involved in a communication API software system
Fig. 1 Architecture stack illustrates the components and layers involved in a communication API software system

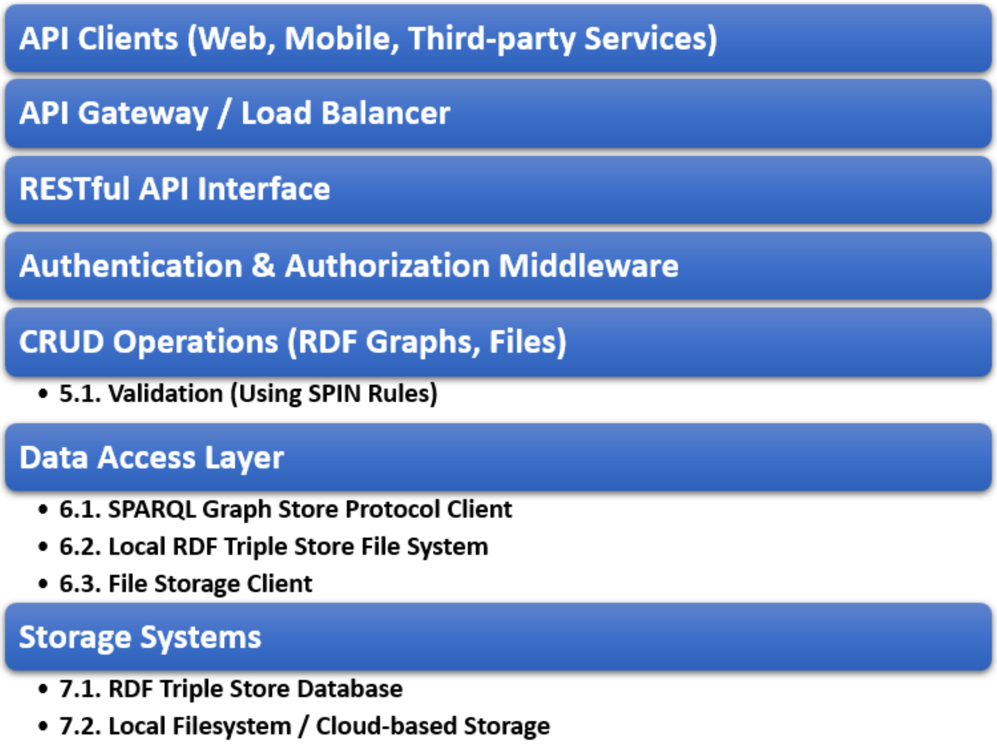

Figure 1: Architecture stack illustrates the components and layers involved in a communication API software system

5. Results & Discussion

5.1. Application Scenario

The deployment sequence of the framework proceeds as follows:

a) A domain expert and a knowledge management expert modify and enhance the ontology's final semantic level to incorporate domain-specific entities and restrictions.
b) The Communication API and Persistent Storage Server are deployed, configuring settings such as communication addresses. This step is independent of the previous step, as the API automatically adjusts the web service endpoints based on the terminal node definitions.

The ease of deployment using free and open-source software for ontology editing, communication API, and the persistent storage server encourages the adoption and utilization of our framework in scenarios not initially anticipated. It is important to note that the standardized communication between framework components (e.g., SPARQL 1.1 Graph Store HTTP Protocol) mitigates future vendor lock-in issues. The framework's sustainability is also supported by the loose coupling of its components (i.e., the communication API is agnostic of the ontology entities and the triple store vendor).

5.2 Adaptation and Reuse

The presented framework is reusable, as it can be redeployed for use in related domains. The framework can manage data in various scenarios, including personal health record systems, well-being management, and telemonitoring research applications in multiple health domains. In a new deployment, the "Domain specific data entities ontology" layer of the model needs to be modified to add or remove classes and, if necessary, properties to align with the concepts corresponding to the specific domain and usage scenario. Figure 5 shows a schematic example of this procedure.

The fifth layer of the model for this deployment, contained entities defined to describe the domain and satisfy end-user application requirements, as all patient-related data was managed by the framework. Entities and hierarchies of entities were defined as children of FHIR Resource classes. For instance, "Biosignal Recording Session" was defined as a child of the FHIR Resource Administrative class, with "Raw Data Session" and "Vital Signs Session." The resulting framework was deployed on Amazon Web Services cloud infrastructure and used to manage the data. Two pilot studies were conducted using the cloud deployment of our framework: one in Greece with 17 patients and one in the UK with 14 patients. Both studies lasted six months, monitoring each patient for nearly one month and also including data from a specialized wearable vest device and sophisticated feature extraction from biosignals like multi-lead ECG (Electrocardiography) and EIT (Electro Impedance Tomography). The feature extraction procedures resulted in a large volume of data managed by the API. The deployed framework operated reliably without downtime, and the pilot data analysis confirmed that there was no data loss or corruption during system use. Using AWS Elastic Beanstalk allowed us to effortlessly and automatically scale the number of deployed API instances when needed without downtime.

The framework was adapted with minor changes to the ontology to support the Guardian Angel tool of the INdependent LIving support Functions for the Elderly (INLIFE) EU project platform. Guardian Angel is a telemonitoring platform consisting of a mobile app for monitoring users during their daily activities and a web-based review and analysis app for meaningful data presentation by caregivers. The study supported sensors such as a wrist wearable device and Bluetooth portable devices for Peripheral oxygen saturation (SpO2) and Blood pressure measurement. The changes required for our framework to support Guardian Angel were limited to the fifth layer of the ontology ("Domain specific data entities ontology"), named "INLIFE Entities" for this deployment. These changes involved removing classes corresponding to specific entities (e.g., "Biosignal recording session") that were no longer applicable in INLIFE. Also, specific concepts of the INLIFE domain, such as "Daily Step Count" or "Rem Non-Rem Sleep Ratio," were defined as new classes. The overall effort needed for transitioning the model to "INLIFE Entities" took a few person-days. Once the ontology was defined, the framework was deployed on local servers to support the pilot study, which began in December 2016. The framework managed the synchronization of pilot data, monitored by the application in near real-time (at most 5 minutes after each measurement was taken), from 92 elderly patients with mild cognitive impairment along with COPD, diabetes, or mobility issues for a continuous period of six months. The framework operated problem-free and with zero downtime throughout both projects. It is worth noting that although detailed performance benchmarking was not conducted for the complete framework, there was no noticeable difference in response times compared to
The RDF store's role was to record every other aspect of the generated data (from the patient or Healthcare Professional side) alongside the metadata concerning the binary files. This led to a relatively small database size compared to the rest of the storage. Conversely, in the case of INLIFE, the relatively simple format of the recorded data allowed us to directly capture all the recorded information inside the RDF store. These two distinct use cases serve as another demonstration of the reusability advantage of our approach.

During the pilot studies, a few new requirements were introduced. Besides other parts of the project's software, these requirements also necessitated modifications to the data model and communication API to enable the management of new concepts. These modifications were applied to the framework using the Ontology/Schema service of the communication API to update the semantic model. Upon such a change, the RDF Resources Service is automatically updated, providing the required functionality in the already deployed framework.

An example of this procedure occurred during the WELCOME project's pilot. An additional requirement was introduced to manage data about Risk Assessments performed not only by a Practitioner but also assessments originating from the developed Clinical Decision Support System (CDSS). To support this, two modifications were made to the model using the Ontology/Schema service.

Although a detailed usability and performance evaluation of the framework was not conducted during the aforementioned studies, the results of the deployment in two pilot studies demonstrated the robustness, reusability, and flexibility of the presented Data Management Scaffolding Framework. The framework successfully supported data management operations without data loss or corruption, with zero downtime, and with minimal redeployment effort when transitioning from the WELCOME scenario to the INLIFE GA scenario.

6. Conclusion and Future Scope of Work

This paper introduces a versatile scaffolding framework for managing integrated care health data, specifically targeting tele-health streaming data. The proposed framework is a standards-based, semantically enhanced data processing system, which enables adaptation to new use cases or data models through a single point of change, without necessitating a software rebuild. This is especially valuable in Integrated Care situations that typically involve multiple healthcare pathways and need customization for local information workflows or data models.

By applying the single-point-of-change principle, only the underlying ontology semantic model needs modification, and all data processing framework components adjust accordingly without requiring a code rebuild or software engineer. The framework, inspired by HL7 FHIR, utilizes an expandable OWL/RDF ontology, supports OWL instances serialized in turtle format, and implements measures to ensure data integrity and validity. Storing semantically enriched EHR data as RDF graphs simplifies data reuse in retrospective studies compared to proprietary RDBMS representations.

The dynamic REST API relies solely on the ontology, enabling management of additional entities and properties without API code changes. Successfully deployed in two EU-funded projects with different use cases, the framework required only minor ontology changes for adaptation. Integrated Care scenarios often involve large, diverse data collections, making them suitable for the Big Data paradigm, characterized by volume, velocity, variety, and veracity.

The proposed framework addresses data variety and veracity challenges by adopting Linked Data as its main data paradigm, facilitating integration of various data schemes through widely accepted standards. Semantic annotation of data reduces ambiguity in data description across all processing steps, enhancing data veracity.

During development, the immaturity of HL7 FHIR led to resource changes, necessitating ontology adjustments. The framework's single point of change paradigm facilitated continuous FHIR integration. Validation rule enforcement was another consideration; choosing SPIN for constraint checking meant validation had to occur between the storage and data management web service layers, a standard approach in relational databases.

The framework achieves its objectives by demonstrating adaptability across domains, employing cost-effective, easily maintained open-source software, and ensuring scalability and sustainability. The use of HL7 FHIR, medical terminologies, and the linked data layer promote interoperability and versatile data use, with FHIR and RDF/OWL utilized in all data processing steps. This robust framework supports Integrated Care Services.

Future work should address limitations such as full interoperability through a FHIR server/client exchange mechanism, incorporating additional data exchange formats like JSON-LD, optimizing data stream handling, and facilitating easy and flexible data querying. Legal and ethical considerations prevent data access from the pilot applications, but Linked Data remains crucial for managing Integrated Care scenarios. Additionally, the framework should support specialized security practices, including defining and enforcing authorization access rules for specific data resources, to further mature towards a digital single market in health.

References

[1] D. Kodner, All together now: a conceptual exploration of integrated care, Healthc. Q. 13 (2009) 6–15, https://doi.org/10.12927/hcq.2009.21091.

[2] I. Chouvarda, N.Y. Philip, P. Natsiavas, V. Kilintzis, D. Sobnath, R. Kayyali, J. Henriques, R.P. Paiva, A. Raptopoulos, O. Chetelat, N. Maglaveras, WELCOME - Innovative integrated care platform using wearable sensing and smart cloud computing for COPD patients with Comorbidities, 2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014 (2014) 3180–3183, https://doi.org/10.1109/EMBC. 2014.6944298.

[3] C.W. Kelman, A.J. Bass, C.D.J. Holman, Research use of linked health data – A best practice protocol, Aust. N. Z. J. Public Health. 26 (2002) 251–255, https://doi.org/ 10.1111/j.1467-842X.2002.tb00682.x.

[4] M. Aranguren, J. Fernandez-Breis, M. Dumontier, Special issue on linked data for health care and the life sciences, Semant. Web. 5 (2014) 99–100, https://doi.org/ 10.3233/SW-130115.

[5] B. Tilahun, T. Kauppinen, Potential of linked open data in health information representation on the semantic web, in: SWAT4LS, 2012: pp. 3–6.

[6] H. Liyanage, S.-T. Liaw, C. Kuziemsky, A.L. Terry, S. Jones, J.K. Soler, S. de Lusignan, The Evidence-base for using ontologies and semantic integration methodologies to support integrated chronic disease management in primary and ambulatory care: realist review. Contribution of the IMIA primary health care informatics WG, Yearb. Med. Inform. 8 (2013) 147–154.

[7] S.T. Liaw, A. Rahimi, P. Ray, J. Taggart, S. Dennis, S. de Lusignan, B. Jalaludin, A.E.T. Yeo, A. Talaei-Khoei, Towards an ontology for data quality in integrated chronic disease management: a realist review of the literature, Int. J. Med. Inform. 82 (2013) 10–24, https://doi.org/10.1016/j.ijmedinf.2012.10.001.

[8] L. Lessard, W. Michalowski, M. Fung-Kee-Fung, L. Jones, A. Grudniewicz, Architectural frameworks: defining the structures for implementing learning health systems, Implement. Sci. 12 (2017) 78, https://doi.org/10.1186/s13012-017- 0607-7.

[9] T. Berners-Lee, Linked Data, 52 (2011) 284–292, < https://www.w3.org/ DesignIssues/LinkedData.html > (accessed February 18, 2015).

[10] H. Darwen, C.J. Date, The third manifesto, ACM SIGMOD Rec. 24 (1995) 39–49, https://doi.org/10.1145/202660.202667.

[11] I. Chouvarda, V. Kilintzis, N. Beredimas, P. Natsiavas, E. Perantoni, I. Vogiatzis, V. Vaimakakis, N. Maglaveras, Clinical flows and decision support systems for coordinated and integrated care in COPD, 3rd IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2016, 2016, https://doi.org/10.1109/BHI.2016.7455938.

[12] S. Hanke, C. Mayer, O. Hoeftberger, H. Boos, R. Wichert, M.-R. Tazari, P. Wolf, F. Furfari, universAAL - An open and consolidated AAL platform, in: Ambient Assited Living 4 Dtsch. AALKongress, 2011, doi:10.1007/978-3-642-18167-2_10.

[13] H. Gokalp, J. de Folter, V. Verma, J. Fursse, R. Jones, M. Clarke, Integrated telehealth and telecare for monitoring frail elderly with chronic disease, Telemed. eHealth. (2018), https://doi.org/10.1089/tmj.2017.0322 tmj.2017.0322.

[14] K. Avgerinakis, A. Briassouli, I. Kompatsiaris, Recognition of activities of daily living for smart home environments, in: Intell. Environ. (IE), 2013 9th Int. Conf., 2013, doi:10.1109/IE.2013.37.

[15] J. Schubert, S. Ghulam, L. Prieto-González, Integrated care concept using smart items and cloud infrastructure, Procedia Comput. Sci. 63 (2015) 439–444, https:// doi.org/10.1016/J.PROCS.2015.08.365.

[16] L. Prieto González, C. Jaedicke, J. Schubert, V. Stantchev, Fog computing architectures for healthcare, J. Inform., Commun. Ethics Soc. 14 (2016) 334–349, https://doi.org/10.1108/JICES-05-2016-0014.

[17] A. González-Ferrer, M. Peleg, B. Verhees, J.-M. Verlinden, C. Marcos, Data Integration for Clinical Decision Support Based on openEHR Archetypes and HL7 Virtual Medical Record, Springer, Berlin, Heidelberg, 2013, pp. 71–84, , https:// doi.org/10.1007/978-3-642-36438-9_5.

[18] C. Martínez-Costa, S. Schulz, Ontology content patterns as bridge for the semantic representation of clinical information, Appl. Clin. Inform. 5 (2014) 660–669, https://doi.org/10.4338/ACI-2014-04-RA-0031.

[19] T. Mallaug, K. Bratbergsengen, Long-term temporal data representation of personal health data, Adv. Databases Inf. Syst. 9th East Eur. Conf. ADBIS 2005, Tallinn, Est.

Sept. 12–15, Proc., Springer, Berlin, Heidelberg, 2005, pp. 379–391, , https://doi. org/10.1007/11547686_28.
[20] N. Lasierra, A. Alesanco, S. Guillén, J. García, A three stage ontology-driven solution to provide personalized care to chronic patients at home, J. Biomed. Inform. 46 (2013) 516–529, https://doi.org/10.1016/j.jbi.2013.03.006.
[21] D. Riaño, F. Real, J.A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico, C. Caltagirone, An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients, J. Biomed. Inform. 45 (2012) 429–446, https://doi.org/10.1016/j.jbi.2011.12.008.
[22] S.R. Abidi, S. Hussain, M. Shepherd, Ontology-based modeling of clinical practice guidelines: a clinical decision support system for breast cancer follow-up interventions at primary care settings, Stud. Health Technol. Inform. 129 (2007) 845–849.
[23] H. Knublauch, D. Oberler, P. Tetlow, E. Wallace, A semantic web primer for objectoriented software developers, W3C Ed. Draft (2006) < http://www.w3.org/ TR/ > .
[24] P.L. Whetzel, N.F. Noy, N.H. Shah, P.R. Alexander, C. Nyulas, T. Tudorache, M.A. Musen, BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications, Nucleic Acids Res. 39 (2011) W541–W545, https://doi.org/10.1093/ nar/gkr469.
[25] P.J. Sadalage, M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Pearson Education, 2013.
[26] F. Gandon, G. Schreiber, RDF 1.1 XML Syntax, W3C Recomm. 25 Febr. 2014 (2016) 1–35, < http://www.w3.org/TR/2014/REC-rdf-syntax-grammar-20140225/ > (accessed October 1, 2014).
[27] C.B. Aranda, O. Corby, S. Das, L. Feigenbaum, P. Gearon, B. Glimm, S. Harris, S. Hawke, I. Herman, N. Humfrey, N. Michaelis, C. Ogbuji, M. Perry, A. Passant, A. Polleres, E. Prud'hommeaux, A. Seaborne, G.T. Williams, SPARQL 1.1 Overview, W3C Recomm (2013), < http://www.w3.org/TR/sparql11-overview/ > (accessed October 1, 2014).
[28] V. Kilintzis, N. Beredimas, I. Chouvarda, Evaluation of the performance of opensource RDBMS and triplestores for storing medical data over a web service, in: 2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014, IEEE, 2014, pp. 4499–4502, doi:10.1109/EMBC.2014.6944623.
[29] M. Voigt, A. Mitschick, Yet another triple store benchmark? practical experiences with real-world data, Citeseer. (n.d.). [30] M. Morsey, J. Lehmann, S. Auer, A.C. Ngonga, Ngomo, usage-centric benchmarking of RDF triple stores, AAAI Conf. Artif. Intell. 26 (2012) 2134–2140.
[30] C. Bizer, T. Heath, T. Berners-Lee, Linked data-the story so far, Int. J. Semant. Web Inf. Syst. 5 (2009) 1–22.
[31] R. Reiter, On closed world data bases, Readings Artif. Intell. Elsevier, 1981, pp. 119–140, , https://doi.org/10.1016/B978-0-934613-03-3.50014-3.
[32] T. Bosch, E. Acar, A. Nolle, K. Eckert, The role of reasoning for RDF validation, Proc. 11th Int. Conf. Semant. Syst. (2015) 33–40.
[33] G. Schenner, S. Bischof, A. Polleres, S. Steyskal, Integrating distributed configurations with RDFS and SPARQL, CEUR Workshop Proc. (2014) 9–15.
[34] W3C, SPIN - Overview and Motivation, (2011), < https://www.w3.org/ Submission/spin-overview/ > .
[35] A. Spear, Ontology for the twenty first century: An introduction with recommendations, Inst. Form. Ontol. Med. Inf. Sci. Saarbrücken, Ger, 2006.
[36] V. Kilintzis, N. Beredimas, P. Natsiavas, I. Chouvarda, N. Maglaveras, A fully functional HL7 FHIR based ontology for telehealth data, 17th Int. HL7 Interoperability Conf. IHIC 2017, 2017, pp. 19–26.
[37] N. Beredimas, V. Kilintzis, I. Chouvarda, N. Maglaveras, A reusable ontology for primitive and complex HL7 FHIR data types, Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS (2015) 2547–2550, https://doi.org/10.1109/EMBC.2015. 7318911.
[38] C. Ogbuji, SPARQL 1.1 Graph Store HTTP Protocol, W3C (2013), < http://www. w3.org/TR/sparql11-http-rdf-update/ > (accessed October 1, 2014).
[39] T. Bosch, K. Eckert, Towards description set profiles for RDF using SPARQL as intermediate language, Int. Conf. Dublin Core Metadata Appl. (2014) 129–137.
[40] T. Bosch, K. Eckert, Requirements on RDF constraint formulation and validation, Int. Conf. Dublin Core Metadata Appl. (2014) 95–108.
[41] C. Fürber, M. Hepp, Using {SPARQL} and {SPIN} for data quality management on the semantic web, in: 13th Int. Conf. Bus. Inf. Syst., 2010, pp. 35–46.
[42] C. Pautasso, RESTful web services: principles, patterns, emerging technologies, Web Serv. Found. Springer New York, New York, NY, 2014, pp. 31–51, , https://doi.org/ 10.1007/978-1-4614-7518-7_2.
[43] M. Masse, REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces, 2011.
[44] T.I. Hunter, CONSUMER-CENTRIC API DESIGN., BLURB, 2015.
[45] B. Mulloy, Web API design, 2013.
[46] D.J. Abadi, P.A. Boncz, S. Harizopoulos, Column-oriented database systems, Proc. VLDB Endow 2 (2009) 1664–1665, https://doi.org/10.14778/1687553.1687625.
[47] D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, S. Zdonik, Monitoring streams: a new class of data management applications, in: Proc. 28th Int. Conf. Very Large Data Bases, 2002, pp. 215–

Author Profile

Wasim Fathima Shah is a Healthcare IT Scrum Master with a deep expertise in Healthcare Standards Implementation. Hailing from the United States of America, Shah brings a wealth of knowledge and experience to the field of healthcare technology. With a Master's degree in Engineering Management, Shah has acquired a strong foundation in both technical and managerial aspects of healthcare IT. Shah's passion lies in leveraging technology to drive improvements in healthcare delivery and patient outcomes. As a Scrum Master, Shah excels in facilitating agile development processes, ensuring efficient collaboration among cross-functional teams, and implementing best practices to deliver high-quality healthcare IT solutions. Shah's specialization in Healthcare Standards Implementation is a testament to their commitment to ensuring the interoperability and compatibility of healthcare systems. With an in-depth understanding of industry standards and regulations, Shah possesses the ability to guide organizations in adopting and implementing these standards effectively.