Received: 18 March 2022 | Revised: 4 July 2022 | Accepted: 11 July 2022

DOI: 10.1002/spe.3130

RESEARCH ARTICLE

Design and implementation of the new Italian healthcare digital interoperable registry for implantable medical devices

Duilio Luca Bacocco[1,2] | Eugenio Carrani[1] | Bruno Ciciani[2] | Pierangelo Di Sanzo[3] | Francesco Leotta[2] | Marina Torre[1]

[1] Istituto Superiore di Sanità, Rome, Italy

[2] DIAG, Sapienza University of Rome, Rome, Italy

[3] DISIM, University of L'Aquila, L'Aquila, Italy

Correspondence
Duilio Luca Bacocco, Istituto Superiore di Sanità, Rome, Italy.
Email: duilioluca.bacocco@iss.it

Abstract

Nowadays, the role played by registries in monitoring and improving healthcare, including the quality of medical devices, is widely recognized. A well-designed digital healthcare registry, in particular regarding data collection procedures and tools, can effectively support goals such as monitoring a (large) population subject to a specific condition, describing the natural history of diseases, supporting observational study methods, as well as evaluating the clinical effectiveness or cost effectiveness of healthcare products and services. This article describes the architecture of a new platform implementing a digital interoperable healthcare registry, the Italian Implantable Prostheses Registry (RIPI). One of the main goals of RIPI is to provide reliable and high-quality data for monitoring surgery outcomes, performing survival analysis, assessing the safety of devices and procedures, and supporting the traceability of patients. The article focuses on the key aspects and choices that guided the design and implementation processes of the new platform. Most of the design choices came from specific requirements to fulfill, in particular concerning data quality, access policy, interoperability,

Abbreviations: AmAGeT, "Administration Authorization and Territorial Management" (Amministrazione Autorizzazione e Gestione Territoriale); BRF, BRAVa Rules Format; csv, comma-separated values; DM Dictionary, Medical Devices Dictionary (Dizionario dei Dispositivi Medici); EHR, Electronic Health Records; FIFO, First In First Out; FOIA, Freedom of Information Act; GDPR, EU General Data Protection Regulation; GeDI, Management of Implantable Devices (Gestione dei Dispositivi Impiantabili); GTIN-EAN, Global Trade Item Number-European Article Number; ISS, Italian National Institute of Health (Istituto Superiore di Sanità); JSON, JavaScript Object Notation; MDS, Minimum Data Set; MeDIC, Medical Devices Querying System (Ortopedia Medical Device Interrogazione Completa); NJR, National Joint Registry (of UK); NSIS, Nuovo Sistema Informativo Sanitario; OrtMeDIC, Orthopaedic Medical Devices Querying System (Ortopedia Medical Device Interrogazione Completa); RaDaR, Hospitalizations Data Collection (Raccolta Dati Ricoveri); RBAC, Role Based Access Control; REST, Representational State Transfer; RIAP, Italian Arthroplasty Registry (Registro Italiano ArtroProtesi); RIAP-DM (Dictionary), RIAP Medical Devices Dictionary (Dizionario RIAP dei Dispositivi Medici); RIDEP, Italian Implantable Cardioverter-Defibrillator and Pacemaker Registry (Registro Italiano Defibrillatori e Pacemaker); RiDi, Devices Search (Ricerca Dispositivi); RIDIS, Italian Spinal Implants Registry (Registro Italiano Dispositivi Impiantabili per chirurgia Spinale); RIPI, Italian Implantable Prostheses Registry (Registro Italiano delle Protesi Impiantabili); RIVAC, Italian Heart Valves Registry (Registro Italiano Valvole Cardiache); RNPM, National Breast Implants Registry; SDO, Hospital Discharge Record (Scheda di Dimissione Ospedaliera); SOnAR, Automatic Online Synchronization of Hospitalizations (Sincronizzazione Online Automatica dei Ricoveri); UDI, Unique Device Identification; UDI-DI, Unique Device Identification-Device Identifier; UDI-PI, Unique Device Identification-Product Identifier; XML, eXtensible Markup Language; XSD, XML Schema Definition.

extensibility and modularity. Overall, the article discusses the main challenges and the adopted solutions, proposing a design perspective and describing an experience of interest for computer scientists, engineers and practitioners, in particular in the area of healthcare information systems.

KEYWORDS

data quality, data services, digital registry, interoperablity

1 INTRODUCTION

Over the last few years, studies highlighted the importance in healthcare of registries for monitoring the course of diseases.[1] Other studies outlined the critical role of registries in improving the outcome of surgeries.[2,3] In healthcare, a registry can be defined as "an organized system that uses observational study methods to collect data (clinical and other) to evaluate specified outcomes for a population defined by a particular disease, condition, or exposure, and that serves one or more stated scientific, clinical, or policy objectives."[4] Main purposes of a registry typically include describing the natural history of diseases, determining the clinical effectiveness or cost-effectiveness of healthcare products and services, measuring or monitoring safety and harm, and/or measuring quality of care.[5] A registry is also an ongoing quality assurance mechanism, designed to monitor outcomes within a specific region, country or multiple countries being surveyed.[2]

Among the different registry types, registries of medical devices are key tools in supporting the activity of vigilance on the devices available on the market, in evaluating their safety and in tracing patients in the case of implanted devices that needs to be recalled.[1-3,5]

To reach their goals, medical device registries might have to integrate data from various and heterogeneous sources. Moreover, they are required to provide high-quality data, because of their high impact on patients' health. Accordingly, the organization, availability, and timeliness of data in registries represent crucial aspects.[5] Also, registries typically involve many stakeholders, as they aim to enroll clinicians, patients, hospitals, governmental entities, manufacturers, and other kind of entities within the community being surveyed.[2] Thus, the community of participants may be large, spanning from regional or countrywide communities to international communities. Furthermore, registries are often required to support interoperability, in order to enable data exchange, in particular with registries of other countries, while ensuring strong data protection.[6] To support vigilance and surveillance activities on medical device, in 2017, the Italian Implantable Prostheses Registry (RIPI) was established at the Italian National Institute of Health (ISS) by a national governmental decree[7] and was then designed as an umbrella including several registries for specific devices.[8]

In this article, we present the design and the pilot implementation of a software platform for supporting data collection and management of RIPI, and we discuss the key aspects and choices that guided its development project. The new software platform, referred in the text as RIPI Platform, redefines and reimplements the functionalities introduced by the platform for the Italian Arthroplasty Registry (RIAP),[9] the first Italian medical device registry now included in RIPI, and extends them to a larger context targeting the support for a multi-registry environment.

Thus, the design of the RIPI platform aims at improving the whole data gathering and management process, and at overcoming a set of problems of the RIAP platform, like drawbacks related to a data scheme tailored for a single registry, low maintainability and extensibility, limited interoperability, and limitations associated with the adopted data formats and data exchange procedures. To efficiently manage multiple registries, the RIPI platform takes into account new capabilities and an extended set of functionalities, like automated data quality checks, extensibility through new modules, optimization of the operational costs, interoperability with international registries, scalability with respect to the different types of participants, and safe user authorization procedures modeled on the specific organization of the Italian National Health Service.

More in detail, in this article we focus on the data layer design approach, the modularity of the data structures, the data validation processes, enhanced with the production of automatic feedback for improving data quality, and on the adopted role-based access control management that characterized the design and implementation process of the RIPI

Platform. Also, we emphasize the cloud-oriented architecture of the platform, based on micro-services, and the related cost efficiency aspects. Finally, we present the early evaluation results, and we discuss the impact of the design and implementation choices, with the aim of providing references based on our experience for the future development of healthcare registries.

The article is organized in sections dealing with the context, the RIPI architecture, and the design and pilot implementation of the platform, with a particular attention to data formats and structures, to the descriptions of each component and to the main implementation details.

2 | THE CONTEXT

The RIPI platform was designed to support multiple registries, each one managing the procedures related to the implant of specific classes of devices. Currently, the following four registries are included:

- RIAP: The Italian Arthroplasty Registry.
- RIDEP: The Italian Implantable Cardioverter-Defibrillator and Pacemaker Registry.
- RIDIS: The Italian Spinal Implants Registry.
- RIVAC: The Italian Heart Valves Registry.

Each registry is organized as a federation of regional registries coordinated, at national level, by the ISS.

One of the main goals of RIPI platform is to enable operations on reliable and high-quality data, to monitor surgery outcomes, conduct survival analysis, enable performance assessment of devices and procedures, and support other actions, like fast traceability of patients, which is required in case of, for example, device recalls.[10] RIPI participants can be grouped in two categories: (1) Regional Coordination Centers and Hospitals, which periodically feed the registries with clinical data, and (2) manufacturers of implantable devices, which periodically update the registries with data about implantable devices available on the market. The collection of data to be hosted by the RIPI platform started in 2006 with RIAP. Since then, the set of participants continuously increased over the years, especially after the 2017 law that established RIPI. To support the increasing number of participants, the RIAP platform was updated several times, introducing features that were taken as a reference to design the RIPI platform.

2.1 | Overview and limitations of the RIAP platform

To design the RIPI platform, the main problems and limitations of the RIAP platform concerning its architecture and data flows were analyzed thoroughly.

The RIAP platform is composed of the following three software applications:

- **SOnAR**, for handling transmission of clinical data from Regional Coordination Centers to the registry. The acronym stands for Automatic Online Synchronization of Hospitalizations.
- **RaDaR**, for allowing Regional Coordination Centers that do not have their own registry data collection tools to collect them in hospitals and structures where surgeries are performed. The acronym stands for Hospitalizations Data Collection
- **OrtMeDIC**, for managing the dictionary of implantable devices (RIAP-DM Dictionary), a dataset updated by the manufacturers of implantable devices containing all the devices available on the market. OrtMeDIC is employed by RaDaR to allow the choice of devices used in surgeries.

Figure 1 provides a high-level representation of the RIAP platform architecture, including software applications, participants, and the main data flows (depicted by arrows).

Clinical data transmitted to the registry by a participant is the result of the integration of the Hospital Discharge Record, which is produced every time a patient is discharged from a hospital, with a minimum set of variables, called Minimum Data Set (MDS), that describe in detail the joint replacement surgeries the patient underwent during the hospitalization. Such data is needed by RIAP and is not available in the Hospital Discharge Record.[10] The format used

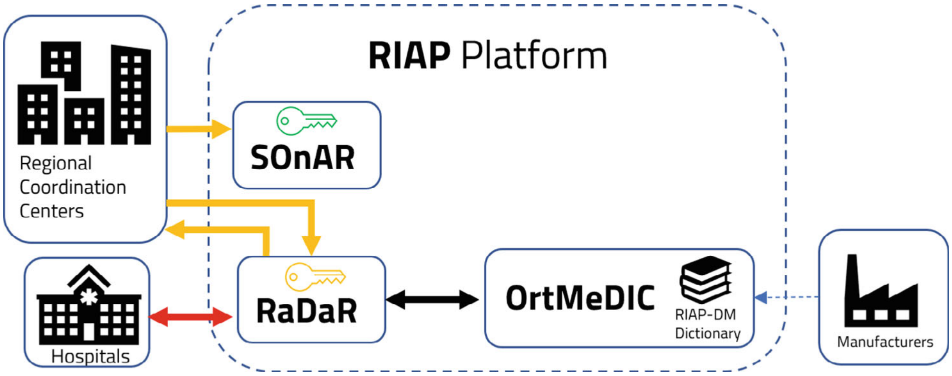


FIGURE 1 The RIAP platform architecture

for data exchange is *comma-separated values (csv)*. The integrated data are then flattened into *csv* files, which comply with a specific structure, as defined in the file structure technical documentation,[11] while transmission procedures and quality checks comply with specific rules, as defined in the technical procedure documentation.[12,13] Regional Coordination Centers that do not use RaDaR exploit their own data collection systems, thus registry data are extracted by these systems, transformed (if needed) to comply with the RIAP data exchange format and integrated with the MDS. Differently, when RaDaR is used, registry data is extracted directly from it. As data are received by SOnAR, it undergoes a series of semi-automated quality checks, as well as ad-hoc format standardization, before being ingested into the RIAP database.

The RIAP platform shows the following major limitations:

- It was designed around a single specific data domain (i.e., the one of data about joint replacements).
- Its main components were designed following a monolithic architecture approach.
- It did not offer interoperability with external registries.
- Ad-hoc data transformation is required in order to process data received from the participants.

Various of the above limitations arise from the fact that the RIAP platform was designed on the early RIAP data collection requirements, without considering possible future integrations and/or interactions with other registries. For example, RaDaR and OrtMeDIC were designed exclusively for managing data collection and flows of joint replacements clinical data and device catalogs. The main three components of the RIAP platform are monolithic in their architecture and are loosely integrated. As a matter of fact, a direct interaction only exists between RaDaR and OrtMeDIC. In the current architecture, RaDaR uses OrtMeDIC to query the RIAP-DM Dictionary for devices that can be associated to surgeries. The loose integration also reflects on the different authentication procedures required to access the applications, since RaDaR and SOnAR use different user databases.

As for data exchange, SOnAR only supports the reception of *csv* files, whose structure is not constrained or defined/validated through some schema, but it is only specified by the technical documentation. Because of this, format and structure of data received from participants are not guaranteed to adhere to specifications, thus may require ad-hoc transformation. Also, only semi-automatic tools are available to this aim. Similarly, data quality checks are semi-automated, meaning that checking procedures are triggered and supervised by humans.

In addition to data received from Regional Coordination Centers, RIAP is fed with extended datasets of Hospital Discharge Records received from the Ministry of Health, that are used for data quality check and research purposes. Currently, this data feeding is manually operated, as there is no integration with Nuovo Sistema Informativo Sanitario (NSIS[14]), the official data flow from the Ministry of Health. Although this is not currently considered as a requirement, the automation of this data feeding may be required in future.

Other limitations of the current architecture concern OrtMeDIC. Specifically, its data schema only supports the identification data of implantable devices for joint replacements, without offering the possibility to store additional technical data characterizing such devices. Also, it was designed to be queried for data, but not to query other systems for automatic updates of its contents. Both these features are required in the new RIPI platform.[6]

3 | OVERVIEW OF THE NEW RIPI PLATFORM

The design process of the RIPI platform targeted the following main purposes:

- Adopting well-structured data formats and designing automating data quality check procedures.
- Designing an integrated authentication and authorization system modeled on the current federated structure of the Italian National Health Service.
- Integrating the RIPI registries within the Italian National Health Service and enabling interoperability with international registries.
- Designing a cloud-based system, with a modular plug-in based architecture, for scaling with respect to the increasing number of participants and registries and for optimizing the operating costs.

In the next sections, we focus on the design choices to pursuit the above objectives.

3.1 | The architecture of the new RIPI platform

A high-level representation of the functional architecture of the RIPI platform, showing the four currently hosted registries and the main supported data flows, is depicted in Figure 2. Independent data flows are represented through different colors, which will be detailed later in this section.

The design choices about the software architecture for supporting the RIPI functional requirements led to the solution depicted in Figure 3. The figure depicts the main software components and the different data flows. The latter are represented using the same colors as in Figure 2, in order to show the data flows from both the points of view of functional architecture and software architecture.

The software component architecture is based on five macro-components:

1. **Authentication**, for the generation and verification of user credentials, and for enabling user authentication.
2. **AmAGeT** (Administration, Authorization and Territorial Management—"Amministrazione, Autorizzazione e Gestione Territoriale"), for managing authorization of operations based on a role based access control (RBAC) mechanism.

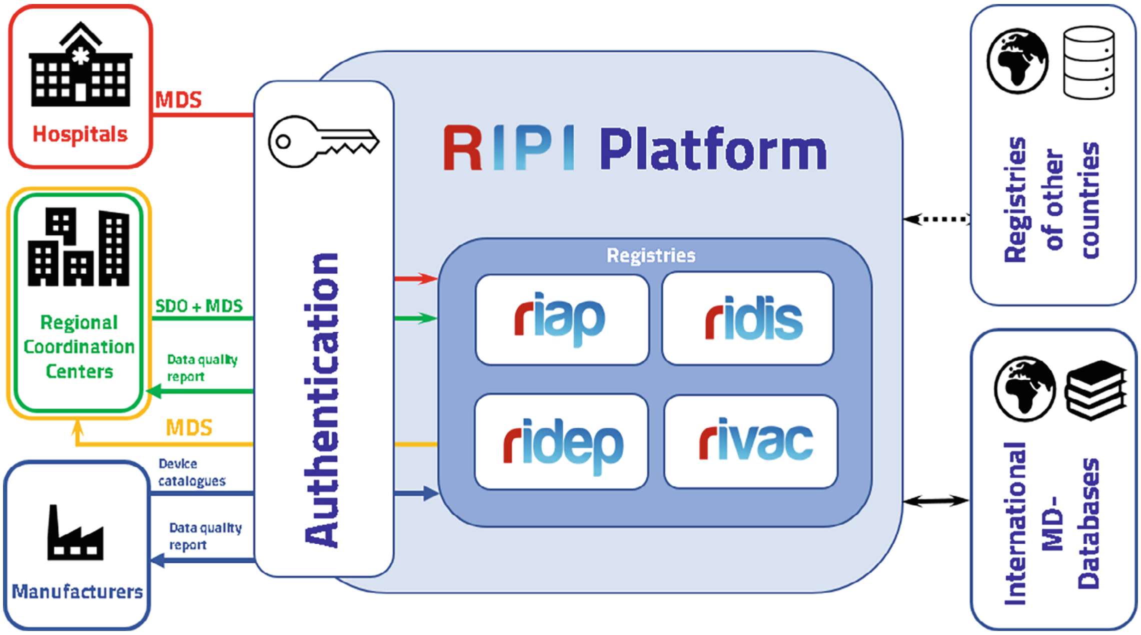


FIGURE 2 High-level representation of the functional architecture draft of the RIPI platform

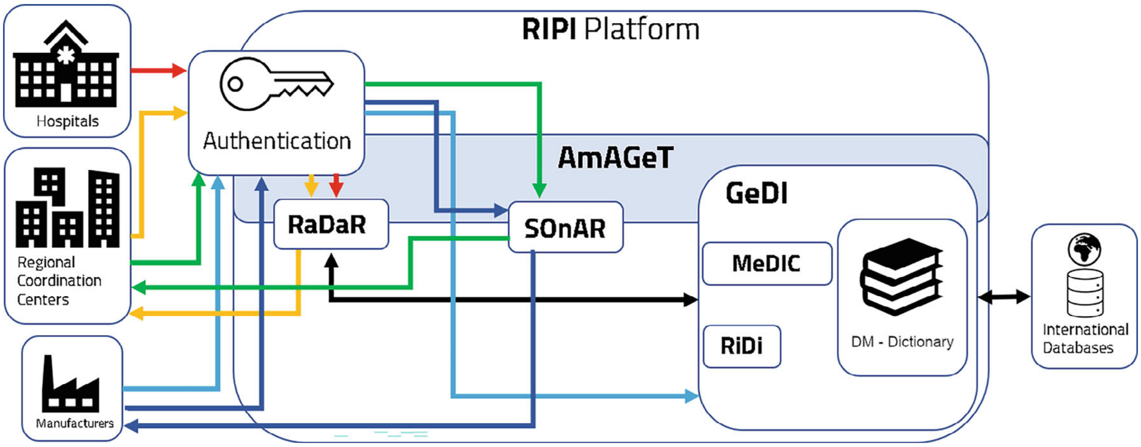


FIGURE 3 Software component architecture of the RIPI platform

The component manages authorization following a hierarchy model which reflects the current federated structure of the Italian National Health Service.

3. **RaDaR** (Hospitalization Data Collection—"Raccolta Dati Ricoveri"), for collecting data of implant and explant surgeries from authorized surgeons and operators. Although it is functionally similar to its RIAP counterpart, this component has been completely redesigned, supporting structured data formats and all the RIPI registries including RIAP.
4. **SOnAR** (Automatic Online Synchronization of Hospitalization—"Sincronizzazione Online Automatica Ricoveri"), for allowing the clinical data transmission from Regional Coordination Centers. For compatibility with existing data formats, the new SOnAR component supports both RIAP *csv* file format as well as new structured data formats, that will be presented in Section 5. When received, correctness of the file formats is checked for adherence to the data format specifications. In addition to clinical data transmission, SOnAR also supports the upload of commercial catalogs by manufacturers of implantable devices.
5. **GeDI** (Management of Implantable Devices—"Gestione dei Dispositivi Impiantabili"), for managing the reference Medical Devices Dictionary (DM-Dictionary). The new component design also includes, as sub-components, the new implementation of OrtMeDIC, now called MeDIC and RiDi, offering a web service-based interface for accessing the DM-Dictionary. Also, GeDI is the component in charge of enabling the interoperability with international registries, as mentioned in Section 1.

Data flows depicted in Figures 2 and 3 are detailed below

- Red arrows represent the data flow that starts from hospitals, from where surgeons and other authorized operators insert into RaDaR data about implant and explant surgeries.
- Yellow arrows represent the data flow through which Regional Coordination Centers retrieve data about surgeries registered in RaDaR in a given period. This data, which is only surgical and anonymized, is downloaded and linked with the corresponding Hospital Discharge Record (SDO). Typically, this happens yearly, and it is early part of a process that continues with the green arrows.
- Green arrows represent the data flow of the process started from Regional Coordination Centers that feed registries with the previously mentioned linked data, receiving back a report that provides a feedback about structural, syntactic and semantic correctness of received data. This feedback can help the senders in improving the quality of data and the efficiency of the overall data transmission process.
- Blue arrows represent the data flow from manufacturers. They are allowed to access SOnAR to upload files containing the updated versions of the catalogs of implantable devices available on the market. After uploading files, similarly to what happens with linked clinical data, SOnAR returns a feedback report about data correctness.

- Light blue arrows represent the data flow from the manufacturers for querying data about catalogs they have previously uploaded.
- Black arrows represent a data flow between RaDaR and GeDI. This flow includes automated queries from RaDaR to GeDI and their responses for collecting updated identification and characterization data about the devices used in surgeries.

The five macro-components showed in Figure 3 implement independent services, and interact between them to cover the whole set of RIPI functional requirements. As it will be described in detail in Section 7, the whole platform implementation is based on a combination of services and micro-services.

4 | SELECTED DESIGN APPROACHES

Data are the main asset of a medical registry. For this reason, data quality plays a critical role. Also, RIPI registries deal with health and safety data of patients. Thus, data security and confidentiality represent key requirements. Also, data manipulation has to strictly adhere to specific data format rules and has to be compliant with strict regulations.

Basically, given the importance of data and its treatment, a Data Oriented Design approach[15] has been selected. Following the traditional Object-Data-Model approach,[16] data structure should have followed the one of the objects which model the reality of the domain. Thus, a specular Data-To-Object model approach has been used, leading the objects to be modeled on the data structures that they represent. The choice of this paradigm implied a preliminary phase of revision and formalization of data formats, that resulted in a complete redesign of data formats and structures, which have been modeled using XML schemas (XSD). This formalization step, along with the produced documentation, provided the foundations for the rest of the project. The detail about this formalization process are provided in section 5.

From a security standpoint, the RIPI platform has been designed following a security-by-design approach. The user authorization process has been designed following an RBAC security model,[17] combined with a variation of the Biba security model[18] to preserve integrity and confidentiality of information. Variations applied to the Biba model consisted in changing the no write-up and no read-down policies to no write-up and relaxed no read-down. The result of this combination is a security model that allows users to have full access to data belonging to users of lower levels in their scope, while the access to data to higher-level users is not allowed. This choice represents the foundation of the user hierarchy in the platform that has been designed to strictly resemble the hierarchy of the federation behind the Italian National Health Service. As in the Health Service, each user in the system, according to its role, has well-defined responsibilities on scopes (territories, structures, and units) assigned, as well as all the subordinate users in the assigned scope. In particular, the adopted security model is such that in every component of the platform, RBAC is locally enforced through roles which are specific for each component. Such roles give users specific authorizations within the component. When registering, users are assigned one or more roles on one or more territories in the AmAGeT component. The roles assigned in AmAGeT are then mapped to specific roles of the other components. Such a structure was introduced to be easily updated following the evolution of the platform. The security model is described more in detail in Section 6.1.

5 | DATA FORMATS AND STRUCTURES

By the adoption of a Data Oriented Design, data representation has been revised and redesigned adopting the structured XML format, thus using XSD schemas. Clearly, this offers the advantages of allowing structure and syntactic correctness to be verified against schemas using widely available tools. Correctness checks are well standardized compared to the previous used *csv* data format, which typically require ad-hoc designed processes. Also, the verification of XML files can be done both at registry-side and participant-side, thus increasing the robustness of data exchange processes.

Data representation is based on two XSD schemas, the RIPI XSD schema for clinical data and an XSD schema for data about devices. In addition to this, it is possible to define custom schemas to enable interoperability with different external registries. More details are given in the next sections.

5.1 | The RIPI clinical data format

A key data flow in RIPI is represented by clinical data (depicted with green arrows in Figure 3). Data are periodically received by Regional Coordination Centers, feeding the RIPI registries. They result from the linkage of anonymized data about surgeries of patients, called MDS, with the patient hospital discharge record (SDO). The MDS is in fact an extension of the SDO that includes the minimum amount of data about surgeries to the purposes of the registry, following a *data minimization* principle, as required by the EU General Data Protection Regulation[19] (GDPR). It is possible to consider the RIPI data format as a structured and generalized evolution of the flat data format used by RIAP, based on a simple concatenation of variables of the SDO followed by the variables of the MDS. The concatenation is used to compose the *csv* file. The RIPI platform will support these files for compatibility purposes, only for RIAP data, in order to allow a smooth transition to the new data format. The new data format devised for RIPI puts together two XSD schemas, the MDS schema that describes the MDS (registry) part of data, and the SDO schema for the SDO part of data, based on the XSD schema the Ministry of Health uses to collect SDOs at national level.[20] With this modeling approach, the overall data structure is an extension of the data structure employed by the Ministry of Health, and guarantees interoperability with the national data flows of NSIS. A graphical representation is reported in Figure 4.

The structure based on two different XSD schemas also reduces the effort behind the linkage procedure. Indeed, Regional Coordination Centers and the other participants have to select, from the SDO XML files they already periodically transmit to the Ministry for other purposes, only the SDOs related to RIPI surgeries and build the MDS part, without having to extract SDO and MDS and assemble them in the RIAP custom format. In the future, this will allow to completely overcome the linkage process. The MDS schema is designed to be modular, in the sense that the structure of data collected by each registry can be defined through an independent “module.” More in detail, each module is an element that contains the specific MDS (sub-)schema for data of a specific registry. This approach favors the independence of the overall part of the MDS with each specific registry schema requirements.

The MDS schema is shown in Figure 5.

5.2 | The DM-Dictionary data format

As discussed, blue arrows in Figure 3 represent the data flow from manufacturers of implantable devices, that periodically send the catalogs of the implantable devices available on the market. The catalogs are used to keep the DM-Dictionary updated. This dictionary is managed by GeDI and used by RaDaR to associate devices to the surgery data. For the participants that do not use RaDaR, the DM-Dictionary can be accessed through the RiDi web service-based interface. As a response to each periodical update of the device catalogs, manufacturers receive an automatically generated report about the quality of data they send. Indeed, all sent data are automatically checked for structural correctness, as for clinical data. Also, it is checked whether the devices in the catalogs are included in the databases of the Ministry of Health, which contain all the devices authorized on the market.

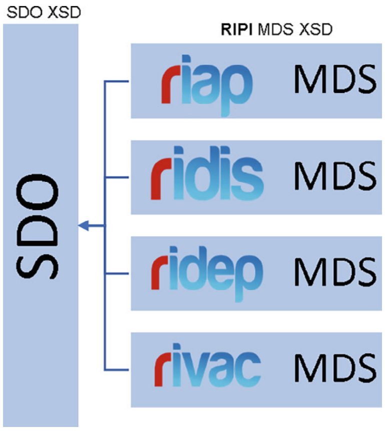


FIGURE 4 The MDS as modules extending the SDO

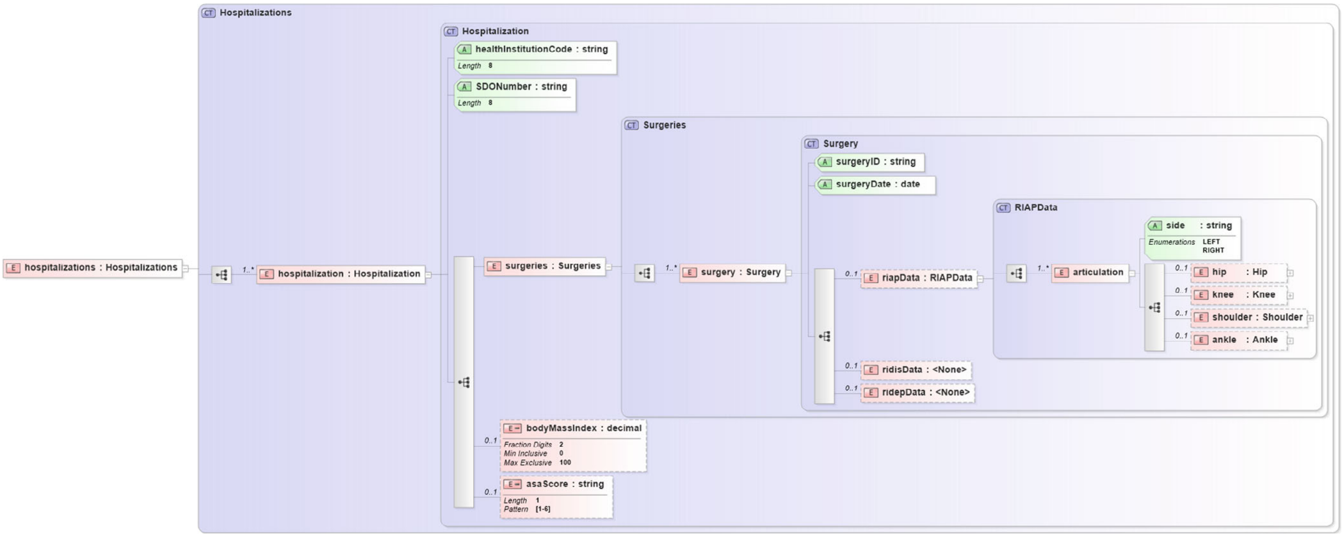


FIGURE 5 The MDS schema with RIDIS and RIDEP modules under construction, as of November 2021

An XSD schema is also used by Manufacturers for the device catalog data(Figure 6). These data are uploaded in SOnAR and are automatically verified, returning also in this case a feedback report. SOnAR also allows to upload in legacy CSV or spreadsheet formats, transparently converting data to XML for validation.

The schema used for manufacturers constitutes a reference on how data about implantable devices are handled throughout the platform. In fact, it represents the superset of the information registered for any device. Subsets of this schema are used to represent the structure of data that associates the devices to surgeries and the response from the RiDi Web Service, which is able to return the same set of attributes in both in XML and JSON, depending on the choice of the client.

The XSD schema used by manufacturers offers a modular structure. When transmitting information about a device of interest for a given registry, manufacturers can specify the technical information of interest for the registry. Currently, only RIDEP uses device characterizations. However, it will be possible to support any additional device by simply adding the related section to the schema, that will be unique for all types of devices and backward compatible with previously produced data. It's worth noticing that with such a schema, manufacturers can either communicate that a class of devices is available on the market (by giving the common identification information like UDI-DI or the legacy GTIN-EAN barcode) or the punctual list of the devices that they put on the market (by specifying single devices either through their UDI-PI or through identifiers about the legacy lot, expiration date, serial number and barcode).

5.3 | The interoperability data exchange format

As mentioned in Section 1, the RIPI platform supports interoperability with other international registries, allowing to exchange data about implantable devices. Following the international cooperation agreement already signed by RIAP,[6] it has been chosen to enable interoperability through a shared database which contains all the technical data required for comparative evaluation of devices. Each international registry has a local replica of the database, which is lazily synchronized with the other replicas in case of missing or outdated data. Again, the exchange data structure has been defined through an XML schema. The structure of this schema, shown in Figure 7, includes a fixed structure for the device identification data (the *deviceID* element), while the device-specific characterization data is stored within the *deviceDescriptionJSON* element. Since this element contains data in JSON format, the characterization is generic and in fact agnostic of the type of device described, thus making the XML schema independent from the specific characterization data of whichever device. The *deviceID* element) has been built based on the structure for device identification adopted by NJR (mentioned in Section 1).

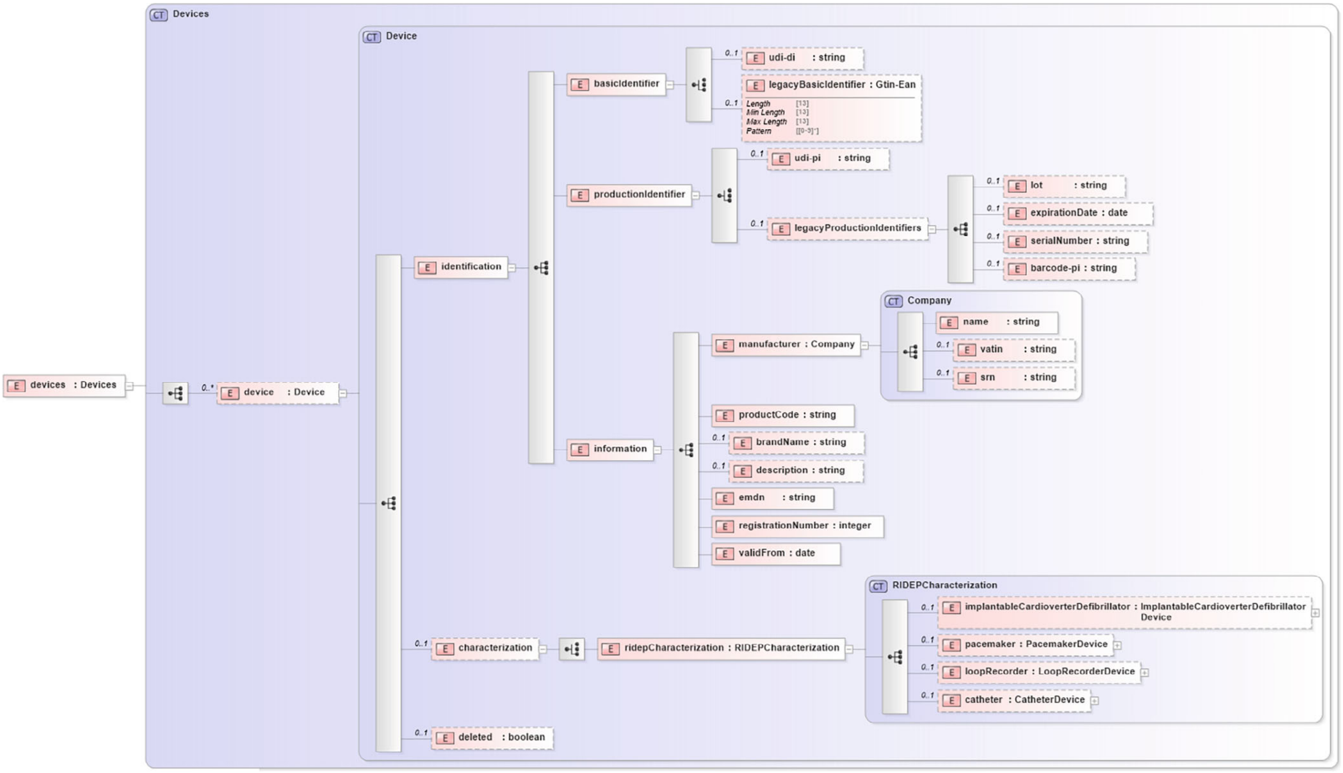


FIGURE 6 The DM-Dictionary data schema with the module for characterizing RIDEP devices

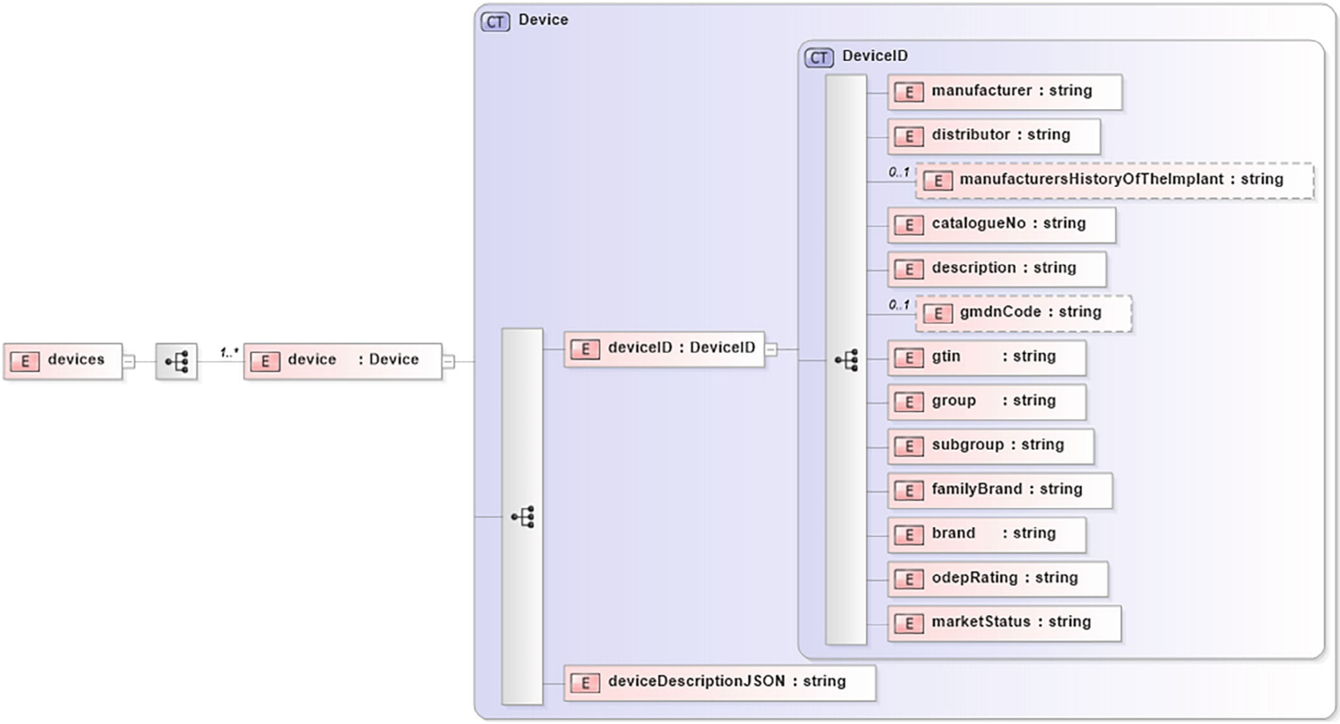


FIGURE 7 The schema defining the structure for the data exchanged

6 | THE PLATFORM COMPONENTS

This section provides specific details about the various components of the new RIPI platform, as evidenced in Figure 3, highlighting the main features and the key aspects.

6.1 | AmAGeT

AmAGeT is in charge of managing user roles based on their actual position in the user hierarchy of the Italian National Health Service. The position in the hierarchy establishes the operations allowed or denied for the user. From a modeling point of view, a role is associated to a component, and is denoted as C_i, which indicates a role of level i for component C. The level indicates the position of the role in the user hierarchy for the component it refers to. As the level increases, the role is placed lower in hierarchy, hence the user is authorized to execute a more restricted set of operations. Such a role modeling approach allows an easy extension, when needed, with new roles for the components to be added at the bottom of the hierarchies without affecting the existing ones.

Roles in the system are assigned to the users as follows. Let's consider a new user that has a certain role in the National Health Service, for example a Regional Referee for a certain Regional Healthcare Service. The user registration process requires the personal data of the user, its actual role in the health system and the scope for the role. The scopes include *Regional Coordination Centers*, *local health authorities*, *healthcare structures* and *operative units*, on which users can assume the declared role. Registration requests are reviewed to ensure that they are legitimate, so that the access is possibly granted only to entitled users. For this purpose, authorities of the declared scope may be involved in the process. The review is performed by the authorities for the scope to ensure that the user is who claims to be. If the review is positive, the user is assigned a role in AmAGeT on the declared scope, corresponding to its role in the Italian National Health Service.

Once the role in AmAGeT is assigned, established mappings (as shown in Figure 8) are used to propagate the assigned role to corresponding roles in the other components. This enables every registered user to operate across the whole platform. In conclusion, the actions each user is initially allowed to perform in each component depends on the role assigned in AmAGeT, that in turn follows from the role the user has in the national hierarchy.

In the proposed design, the registration, as well as the login action, is meant to be handled by the Authentication component, while AmAGeT is focused mainly on enforcing the role-based access model. It is worthy to mention that the system was designed to be flexible. Initially, privileges are granted through mapping, but also later manual assignments of roles in single components is possible to accommodate future requests based on the evolution of system requirements over time.

The Italian National Health Service is organized in Regional Coordination Centers, every region containing many local health authorities.[21] Every local health authority includes healthcare structures, each of them containing several operative units. Surgeries are performed in the operative units. We have considered every entity as possibly having one or more referees interacting with the RIPI platform. The model of the hierarchy of the Italian National Health Service would have been incomplete if we had not considered that users have roles and authority on a given scope, which in fact is the representation of a territorial entity in the health system. The model of the hierarchy and the federation of the Italian National Health Service in AmAGeT(Figure 9) has been designed relying on two families of roles:

1. **Administrators**, R_0, who having level 0 have full operational power on the entire system.
2. **Territorial referees**, R_i, where the decreasing level of i progressively reduces the operational privileges of the role. Due to the hierarchy model, given two roles R_i and R_j, with $i < j$, R_i is higher in hierarchy than R_j.

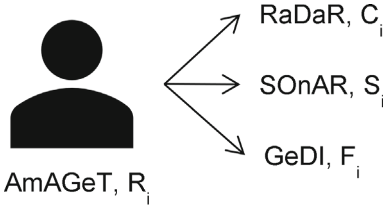


FIGURE 8 The mappings for the propagation of roles

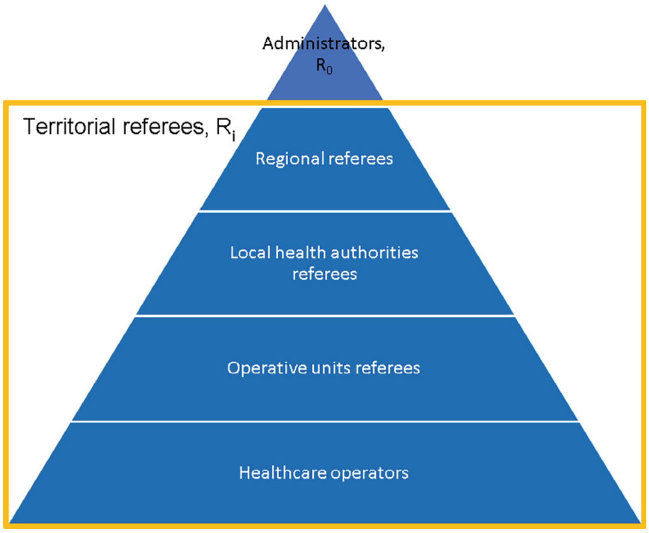


FIGURE 9 Graphical representation of the overall hierarchy of roles

Other than Administrators, that in the proposed model are *above* the hierarchy, as they operate on the entire system regardless of the scope, in the resulting hierarchy model of the health system, the highest role is R_1 , which corresponds to a regional referee. The regional referee is a user that has the operative control of a territory which corresponds to a region. As the level index increases, the role is lower in the hierarchy, with level $i + 1$ having assigned an operational scope (set of territories, local health authorities, healthcare structures, operative units) at most equal (but likely smaller and included) to the scope assigned to level i . Health operators, are by choice assigned the highest level, meaning that they have the lowest territorial authority but are still assigned to the operative units in which they work.

In this model, users are allowed to *delegate* their actual roles to other users. Thus, similarly to what happens in real-world scenarios, each user can delegate, or in other words legitimate, one or more people to do its job. When a user is delegated for a role by a delegating user, the delegated one assumes the role of the delegating user in the system, but it cannot further delegate other users on the role it has been delegated. In this way, users have the autonomy to manage their scopes, without the need of external intervention for everyday operation.

To demonstrate the efficacy and flexibility of the adopted authorization model, we can consider an example scenario in which the platform starts operating, and has initially a single Administrator user. Administrators have privileges on the entire platform, since they act on behalf of national entities, like ISS or the Ministry of Health. Administrators can define other Administrators, as well as scopes to create a map of the territorial entities and assign scopes to one or more referees. From this point on, each scope is managed by its referee. For example, supposing that in the initial configuration the Administrator has created Regional Coordination Centers and has assigned them to the appropriate regional referees, the regional referees will autonomously manage their territories, by creating the local health units and assigning them their referees. The same management strategy applies going down through the hierarchy. Even the registration requests of new users are managed by the appropriate referee, depending on the role requested by the new user at registration. For example, if a health operator registers in the platform as an operator in the operative unit B of the structure A, its request will be processed for approval and role assignment by the referee of the operative unit B, which supposedly knows the operators of its operative unit. If requests are not processed within acceptable time, they escalate the hierarchy bottom-up, until a referee either approves or reject the request. This choice allows to ensure that requests are served as soon as possible by people knowing as much as possible about the involved scope, avoiding the possibility of inadvertently authorized users.

6.2 | RaDaR

RaDaR enables the reception and collection of data about surgeries, including data related to implanted devices. Conceptually, it is the evolution of previous RaDaR application, which has been completely redesigned for supporting novel functionalities. In particular, RaDaR supports the export in the XML formats described previously in Section 5.2, and uses another component called BRAVa (which will be described in Section 6.4) to guide the choices of the operators depending on the values they specify during data insertion. BRAVa is also used by SOnAR, as it will be described later. RaDaR is meant to be used mainly by health operators in selected structures to collect registry data right after surgeries. The health operators can access the full set of functionalities in their scope, for example the structure they are assigned to, while users with higher roles can access a subset of functionalities, still in their scope, for monitoring purposes.

The workflow for data reception in RaDaR is such that health operators access the component to register the RIPI data after they have performed a surgery of interest of the registry (it is identified by red arrows in Figure 3). Each surgery is identified by a progressive number on the health discharge record and the healthcare structure code, which is unique at national level and is automatically assigned considering the scope of the healthcare operator that is inserting the data. Yearly, referees access RaDaR to export data pertaining to their scope and link each record with the corresponding health discharge record (this flow is identified by yellow arrows in the figures in Section 3.1). Linked data are then uploaded in SOnAR and checked for quality (identified by green arrows). The quality check is performed to ensure that it has not been altered during the process of linkage, since the MDS part of the data has been generated by RaDaR following the correct rules in terms of data structure and semantics.

As part of the registry data, RaDaR requires the operators to specify the devices that have been implanted during the procedure. Implanted devices are selected from the set of devices contained in the DM-Dictionary. The dictionary, managed through the GeDI component, is built and updated from the catalogs uploaded by the manufacturers in SOnAR. If it is not possible to select a device in RaDaR because it is not available in the dictionary, the operator should file a request for its insertion and wait for the device to be inserted in the DM-Dictionary. This is required to avoid that clinical records remain in a pending state. This choice prevents the manual insertion of information about devices, that often result in low quality data due to inaccuracies, mistyping, or wrong data.[10]

6.3 | SOnAR

SOnAR allows Regional Coordination Centers to send data resulting from linking of SDO data and MDS data, as described in Section 5.1. The new version of SOnAR evolves the core functionality the previous version in RIAP platform, acting as a communication gate for all the RIPI registries, while also allowing manufacturers of implantable devices to upload their updated catalogs. SOnAR automatically performs data validation based on schemes, also applying syntactic and semantic checks to evaluate the quality of data and returning feedback reports. Checking and validation capabilities still rely on BRAVa. SOnAR supports the upload of clinical data in the new XML formats, as well as in the previous CSV format, to allow a smooth format transition for participants. The same holds for manufacturer catalogs that are supported in both XML and CSV. Independently of the type and format, all received data are automatically checked and validated, producing a feedback report.

6.4 | BRAVa

As already mentioned, BRAVa (Business Rules Automatic Validator) is a component designed to perform data validation and quality check, while also representing a source of knowledge about the structure of RIPI data. For what concerns validation, XML schemas allow the verification of the structure of XML files and the membership of the values specified for each element with respect to their domains defined either by enumeration or regular expression. However, in many cases, this is not enough for verifying the correctness. There are cases in which the value of a variable should not only belong to the domain, but it should also be correct with respect to the values assumed by other variables within the same record. As an example, let's consider the case of a dataset in which each record is a tuple $(x \in X, y \in Y, z \in Z)$. Using an XML schema, data validation would allow to check whether for each record $x \in X, y \in Y, z \in Z$. However, suppose that each record to be correct have to comply with business rules, for example like the following ones:

- $x = a \Leftrightarrow y = c$
- $x = b \Leftrightarrow (y = d) \wedge (z = e)$

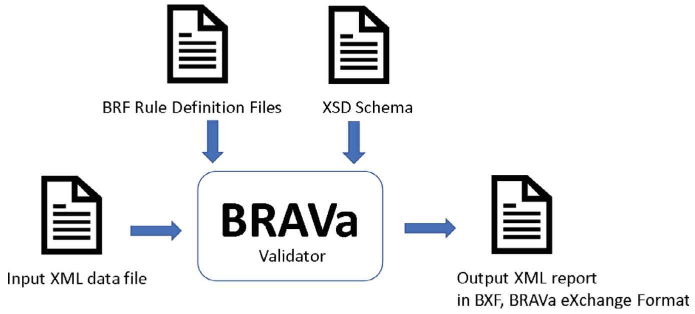


FIGURE 10 Inputs and outputs of BRAVa

with $a, b \in X$, $c, d \in Y$, and $e \in Z$. A record $R = (a, d, e)$ would be accepted by the XML validation even if not compliant with the business rules. In RIPI, business rules ensures coherency of values of the different variables. Thus, BRAVa uses a rule-based validation engine that, in addition to data validation through XML schemas, also checks business rules on variables. A representation of inputs and outputs of BRAVa is shown in Figure 10.

Business rules are contained within XML files and are written in the BRAVa Rules Format (BRF). BRF is defined through a dedicated schema, which is shown in Figure 11.

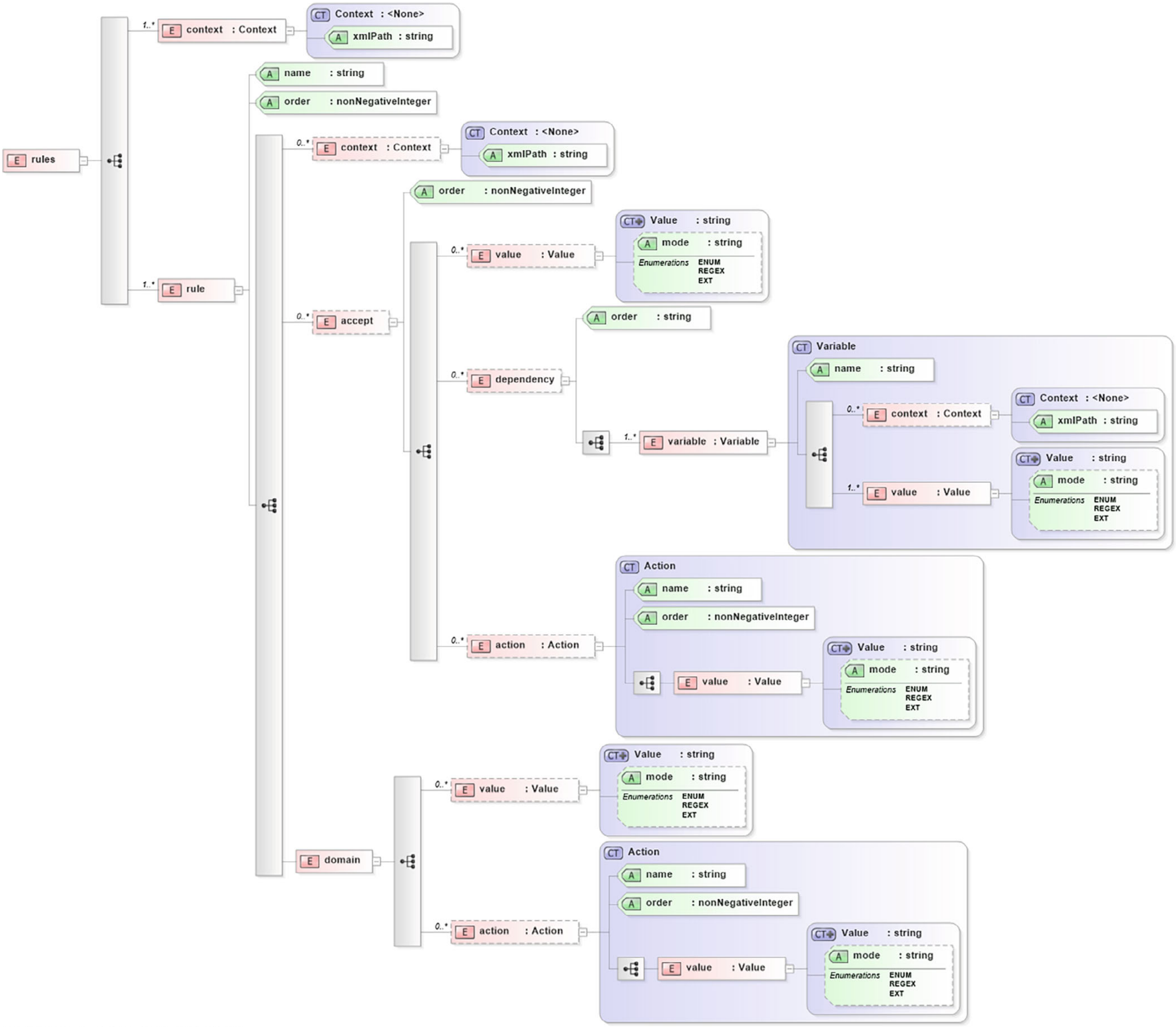
The schema was designed to describe the validation process of an input data file over an ordered set of rules. Each set of rules is defined in a context on which it will be evaluated. The context is the XML path of the root of the sub-trees where the rules apply. In other words, it is the XML path to each record to be validated. Each rule in the set applies to a variable (identified by its *name*), which can be located in the data file at its own context. The context of a rule extends the general context by indicating the subtree in each record where the variable should be evaluated. Each rule is defined by an *accept* and a *domain* element. The domain element describes all the values in the domain of the variable, and the actions to be executed if the value of the variable belongs to the domain. The accept element describes the business rules to be checked on the variable by saying that certain values are admissible if the dependencies, evaluated in order, are satisfied. If this is true, then an action is executed.

The definition of the domain within the rules for each variable may seem a duplication with respect to what is already provided by XSD schemas. However, this apparent duplication allows to use the schema for describing business rules in more general cases with respect to the analysis of data files. An example of this is the modeling of other rules in the system, like rules for constraining possible inputs as graphical forms are filled. In other words, the BRF allows to specify rules in virtually any system avoiding hard coding, thus allowing easy and fast update and maintenance operations. While being initially designed for BRAVa, it has been decided to widely employ BRF throughout the RIPI platform to give a common language for managing rules.

Accessing the definitions of all data structure and business rules, BRAVa plays the role of a system-wide authority. Thus, in addition to data validation, it can be trusted for queries about data structure or business rules. It offers a web service that replies to queries like the following ones:

- *Given the variable v, from which variables does the value of v depends from*? Considering the previous example about the business rules, if asked for dependencies of variable x, BRAVa, would respond with the tuple $\{y, z\}$, while if asked for the dependencies of variable y the answer would be $\emptyset$.
- *Given the variables v, t, and u, if $v = v'$ and $t = t'$ which are the possible values for variable u*? Considering again the previous example, if queried by asking the possible values of x for $y = d$ and $z = e$ the answer would be $x = b$ while for $y = c$ and $z = e$ the answer would be $\emptyset$.

Among the possible ways to employ BRAVa queries, they can be used to constrain the content of specific fields in forms of the graphical interfaces for ensuring the insertion of correct values. For example, operators inserting clinical records in RaDaR could only insert specific values depending their previous selections (e.g., choosing an allowed surgery indication following the specified procedure type).



Liquid Studio 2021 - Developer Bundle 19.0.11.10915

FIGURE 11 XSD schema for the BRF

6.5 | GeDI

GeDI plays the role of a system-wide authority for data about implantable devices, since it manages the DM-Dictionary. The other components of the RIPI platform can access the DM-Dictionary using MeDIC, a sub-component of GeDI. It allows to, for example, reference and register devices used in surgeries, and enables operators and device manufacturers to query the DM-Dictionary ensuring data privacy based on the user role. For example, manufacturers are allowed to access only data about devices marketed by their brand. GeDI also provides through RiDi a web service-based interface, which allows registered and authenticated external applications to query the DM-Dictionary. RiDi is a REST Web Service that returns a common set of data about devices in both XML and JSON. GeDI also is in charge of managing the interaction with the Web Services dedicated to the access to databases shared for interoperability purposes.

Overall, GeDI manages of three different data flows regarding implantable devices:

1. **Update**, by the manufacturers, of the devices stored in the DM-Dictionary. Manufacturers upload data in SOnAR, data is verified and prepared for review and finally update the DM-Dictionary. The data exchange format is the one described in Section 5.2.

2. **Exchange** of technical information of devices for interoperability, following the specifications reported in Section 5.3.
3. **Share** the identification information of devices through the RiDi web service, whose outputs in XML format include a subset of the data as specified in Section 5.2.

7 | IMPLEMENTATION DETAILS AND EARLY EVALUATION

The first architecture draft of the RIPI platform appeared in a report published by ISS in May 2021.[22] The report was published for providing a reference for participants and for other registries.[7] Then, the RIPI platform started being implemented as a proof-of-concept, in particular to evaluate the effectiveness of the design choices. As for development tools, Liquid Studio[23] has been selected for designing and managing XML and JSON data structures and schemas, and all the components have been developed relying on .Net framework. As for hosting infrastructures, Microsoft Azure has been selected as a cloud platform for the deployment of all platform components and services.

7.1 | The early implementation of BRAVa

As mentioned, the collection of data to be hosted by the RIPI platform started in 2006 with RIAP. Over the years, data has been collected from heterogeneous sources, using different format and tools. This resulted in data with differentiated quality levels. Because of this, there was the need to assess quality and reliability of collected data and this led to the design and first implementation of BRAVa. Originally, BRAVa was a C# console application that featured all the validation and dependency querying functionalities. This early implementation allowed extensive testing of its core functionalities with real data. Given the flexibility of the tool, data used for the first tests came from the National Breast Implants Registry (RNPM) dataset. RNPM is a project of the Ministry of Health whose data collection platform has been realized in collaboration with RIPI. It was known that its dataset, collected through a preliminary version of the platform, had many inconsistencies in both domains and business rules. The dataset, including about 3000 records and exported in XML format, was processed by BRAVa using a BRF written for the RNPM data structure. Upon analyzing data, BRAVa returned a report that allowed to discover and locate in the dataset all the records with anomalies, totally 847, providing a detailed description for each anomaly. Information in the report allowed to successfully remove all the anomalies, consequently improving the quality of data.

7.2 | SOnAR

SOnAR has been implemented on top of an evolution of the early implementation of BRAVa. SOnAR is the gateway for all the data entering the registry. Its integration with BRAVa allows to analyze received data and to generate detailed data quality reports, thus ensuring that registries receives well structured and verified data. SOnAR was designed to be deployed in a cloud environment.

As said, Microsoft Azure has been used as a hosting platform. The workflow supported by SOnAR and its sub-components involves a data processing flow that is triggered by an initial interaction from the user, via a specific web interface, ultimately allowing the on-demand activation of SOnAR sub-components. More in detail, the SOnAR architecture features a minimal web server with a front-end, listening for incoming requests, and a modular back-end composed of many basic serverless modules (microservices), that implement the business logic for data processing. Depending on the uploaded data, only some specific back-end microservices are required. Accordingly, only the required ones can be activated on demand. As a further advantage, it has been possible to implement the SOnAR back-end through a set of Azure Functions, while the front-end through an Azure App Service. This choice, although may introduce a certain latency upon reactivating idle components, greatly improves the cost efficiency. Indeed, as discussed, the required microservices can be activated only when really needed, thus being billed only for their actual active time. Additionally, serverless computing removes the burden of the infrastructure sizing, since vertical scaling is automatically managed by the cloud service platform, based on the actual computing demand.

The modules depicted in Figure 12, in darker blue with respect to the front-end, are the following ones:

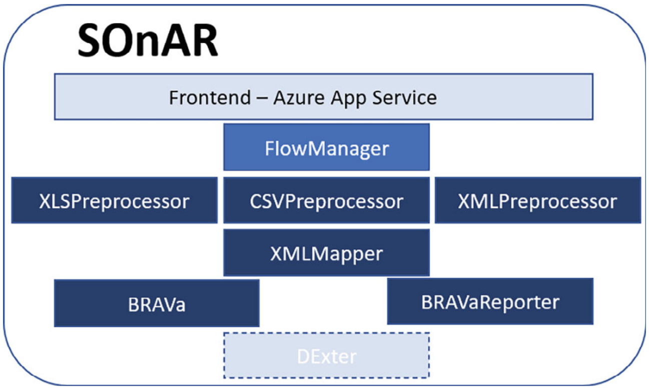


FIGURE 12 Graphical representations of the modules of SOnAR

- FlowManager: A message router based on queues to allow components to coordinate and communicate with each other.
- XLSPreprocessor: It transforms input Excel spreadsheets into CSV files.
- CSVPreprocessor: It analyzes the structure of CSV files given as input and transform it to a reference structure depending on the content of the file (e.g., a RIAP CSV is transformed toward its reference format as in record plots).
- XMLMapper: It transforms a CSV file to an XML in the appropriate format. The code for this component is generated by Liquid Studio 2021 Developer Edition, that allows to design complex ETL workloads graphically and generate the corresponding C# code.
- BRAVa: It validates data in XML format and produces a report.
- BRAVaReporter: It applies record ID to each error found in the report.

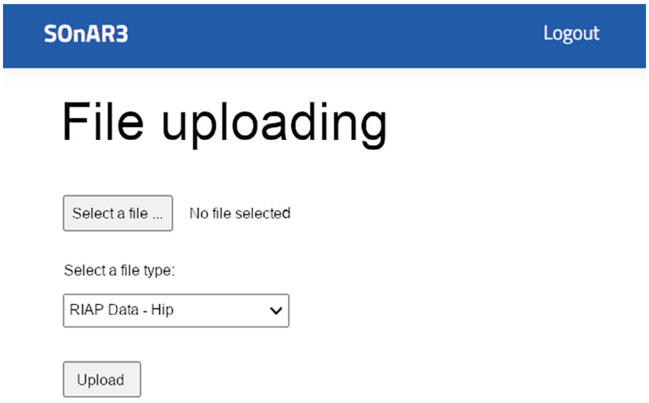
Upon splitting the back-end into several modules, coordination and synchronization issues aroused. To this aim, facilities provided by Queue-triggered Azure Functions have been used. Each module has a dedicated message queue, on which it listens for messages triggering the operations. Upon terminating an operation, each module writes a message on the queue of the flow manager. The flow manager, triggered by the incoming message, dispatches it to the appropriate destination module. Messages exchanged between the different modules use JSON, and contain the following data about the action they trigger on the destination module:

- *Blob storage*: The name of the blob storage from which the input must be retrieved.
- *Source component*: The identifier of the module that sent the message.
- *Objects*: A list of objects to be used by the destination module. Each object has a format and a type that can be either *data* or *report*.

When a message enters the queue of the flow manager, it is parsed and its content is evaluated against a message routing table. The routing table is composed of a series of rows, each one defining a route for the messages. A route is composed of the following elements:

- Message source component
- Object format
- Object type
- Destination queue

The process of matching the correct route is similar to the one employed in network routers. Upon the first match, the object reference is forwarded to the destination queue. This strategy allows to dispatch multiple objects within a message



File uploading

No file selected

Select a file type: RIAP Data - Hip

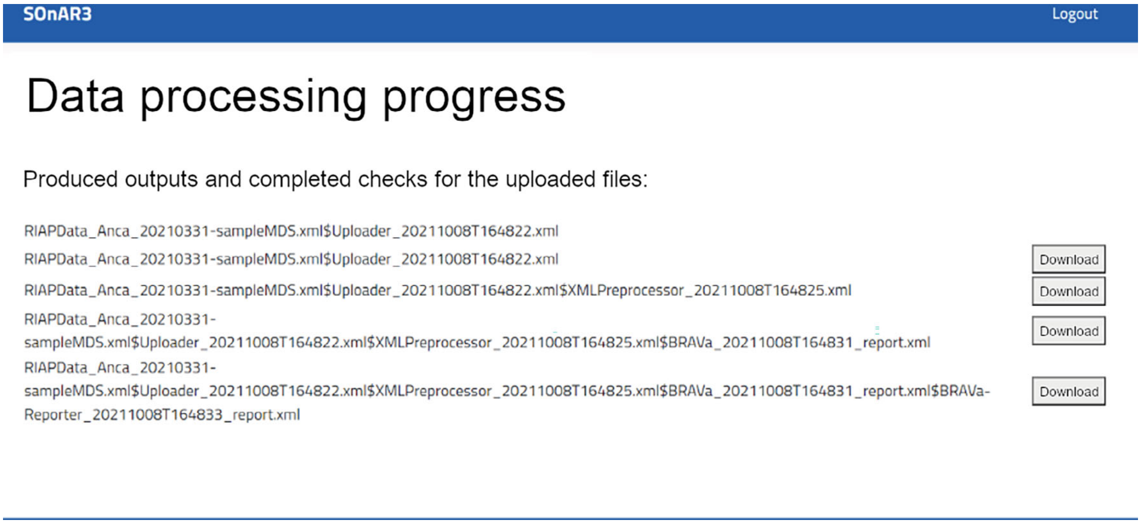
FIGURE 13 The upload interface of SOnAR3

toward different destinations. To better understand the mechanisms, let's consider the typical case of a user uploading XML data to SOnAR.

1. The user logs into the platform with the assigned credentials.
2. After successful login, in the main page of SOnAR, the user indicates the type of the file and uploads it through the web interface (Figure 13). Supposing this is a hip data file, the uploaded file, that was originally named *file.xml* is automatically renamed *RIAPData_Anca_file.xml*. In other words, the file name is prepended the composite information on the file type. Upon uploading, the file is written on the centralized blob storage, and a message (*BlobStorage*, *Uploader*, [(*RIAPData_Anca_file.xml*, *xml*, *data*)]) is written in the queue of the flow manager.
3. The flow manager is triggered by the message written on its queue and performs routing. In particular, it parses the message, finds an appropriate route on basis of the content and then forwards the message toward the proper module, in this case *XMLPreprocessor*
4. XMLPreprocessor is triggered by the message received on its queue. As a consequence, it retrieves the *RIAPData_Anca_file.xml* from BlobStorage and applies some transformations to make sure that the XML file is well-formed. As an output, the module produces a *RIAPData_Anca_file_XMLPreprocessor.xml* file and writes the message (*BlobStorage*, *XMLPreprocessor*, [(*RIAPData_Anca_file_XMLPreprocessor.xml*, *xml*, *data*)]) on the queue of the flow manager.
5. The flow manager is triggered by the message written on its queue by XMLPreprocessor and performs routing. The proper destination of the message in this case is BRAVa.
6. BRAVa is triggered by the message received on its queue. As a consequence, it retrieves the *RIAPData_Anca_file_XMLPreprocessor.xml* file from BlobStorage, the associated XSD schema and the appropriate rule files. The XML data file is processed against the schema and the rules, then BRAVa produces the feedback report, identified as *RIAPData_Anca_file_XMLPreprocessor_BRAVA_report.xml*. The report is provided as an output, while the XML data file is forwarded to the next module. As a result, the message written to the queue of the flow manager is (*BlobStorage*, *BRAVa*, [(*RIAPData_Anca_file_XMLPreprocessor.xml*, *xml*, *data*), (*RIAPData_Anca_file_XMLPreprocessor _BRAVA_report.xml*, *xml*, *report*)]).
7. The flow manager is triggered by the message written on its queue by BRAVa and performs routing. The appropriate route for the content in this case is BRAVaReporter.
8. BRAVaReporter is triggered by the message received on its queue. As a consequence, it retrieves both the given objects and applies the transformations, by replacing the line number for each error encountered with the ID of the surgery it refers to.

The front-end of SOnAR shows the progress of the overall data processing, allowing to download the output produced by each processing step. Figure 14 shows a snapshot of the SOnAR user interface while processing data.

All the queues adopt a FIFO policy for preserving the processing order in the workflow for each uploaded file. Also, it is worthy to note that the adopted approach enables pipeline processing of messages, thus increasing the parallelism level.



Data processing progress

Produced outputs and completed checks for the uploaded files:

RIAPData_Anca_20210331-sampleMDS.xml$Uploader_20211008T164822.xml

RIAPData_Anca_20210331-sampleMDS.xml$Uploader_20211008T164822.xml

RIAPData_Anca_20210331-sampleMDS.xml$Uploader_20211008T164822.xml$XMLPreprocessor_20211008T164825.xml

RIAPData_Anca_20210331-sampleMDS.xml$Uploader_20211008T164822.xml$XMLPreprocessor_20211008T164825.xml$BRAVa_20211008T164831_report.xml

RIAPData_Anca_20210331-sampleMDS.xml$Uploader_20211008T164822.xml$XMLPreprocessor_20211008T164825.xml$BRAVa_20211008T164831_report.xml$BRAVa-Reporter_20211008T164833_report.xml

FIGURE 14 The interface of SOnAR3 showing the progress of a computation

Finally, since the approach is based on a routing table, modifications to the table allow to easily redefine the workflow, for example, as a consequence of the addition or the removal of modules in the architecture.

The example above describes what happens upon receiving an XML file. In general, if an Excel or *csv* file is uploaded, it undergoes some preprocessing before being transformed into XML format, and is processed as described. Excel files are unpacked into a single *csv* file for each sheet in the file by XLSPreprocessor. Then *csv* files, after some mappings to respect reference formats performed by CSVPreprocessor, are passed to the XMLMapper that generates the corresponding XML file for the rest of the processing.

By using the early implementation of SOnAR, it has been possible to analyze for the first time the entire RIAP dataset of hip surgeries, and to quantify the extent of data inaccuracies due to the heterogeneity of data sources. Considering about 304,000 hip records collected since the starting date of the project up to 2018, around 133,000 records showed anomalies. Data was used to train a neural network that, given the MDS variables describing the hip surgery, could predict the fixation of the acetabular component of the hip prosthesis. By training such neural network on the two different datasets, the full dataset and a filtered dataset, from which records with anomalies were removed, showed an increase of the model accuracy spanning from 76% on the full dataset to 92% on the filtered dataset, in fact making data potentially usable in the context of prediction. These results have been presented in Bacocco et al.[24]

8 | LESSONS LEARNED

In this section, we provide some overall observations about what we have learnt from the our experience, in particular as a result of the various choices made to design and implement the RIPI architecture, and the experimental results we achieved. We highlight some of the concrete advantages we observed, some drawbacks, and the potential advantages expected in perspective for the future evolution of the RIPI platform.

The RIPI platform was designed by capitalizing the experience gained in the development and managing of the RIAP platform, extending it to the larger and more complex context of RIPI. Since data represent the main asset of a medical registry and data quality is a crucial aspect for its success, a Data-Oriented Design approach was selected, considering it as an important foundation for the whole platform implementation. Although this choice required an effort for investigating suitable data formats and software data structures, it allowed to ensure the preservation of the whole established data organization along all data processing phases, since software components and data validation procedures were designed around data organization. Moreover, it contributes to both preserve data quality and improve it when incoming data has quality issues. Thanks to the effort in decoupling the design of software components from the data structures, the future extension of the platform to new data domains will not require to modify the design and the implementation of the components, but only to add the associated data definitions. This means that, the introduction of new data domains will only require to define the custom modules of the MDS part of the RIPI clinical data format. The adoption of structured data formats simplified also the design of automated data quality check procedures, to be executed upon receiving data. Early experiments with automated data quality checks on real datasets provided first positive results. In this context, the decoupling between software components and data demonstrated to improve also the flexibility of the proposed solution, when working with different data domains. Even if these design choices required an additional design effort in the early stages, we were repaid by the advantages we have already observed in the data definition process for the different registries considered by RIPI. Indeed, we reasonably expect the future inclusion of new registries to be similarly straightforward. We also expect that such design choices, that provide higher level of data quality, maintainability and extensibility of the platform, will enhance future maintainability of both data structures and software products.

Another foundation of the project is the security-by-design approach. This choice led to the definition of a specialized RBAC authorization system, tailored to the hierarchical structure of the Italian National Health Service. Even if it required an effort to take into account both legislative requirements and registry functional requirements, it allowed to implement an authorization model in which each user is allowed to autonomously administer its scope within the platform (e.g., Regional Coordination Centers, local health authorities, healthcare structures, and operative units), not requiring any special intervention of the Administrators. This facilitates the assignment of the proper roles and scope to users, corresponding to their effective role they have in the Italian National Health Service, thus making the system more efficient and easy to manage for both users and administrators.

The monolithic design of the RIAP platform components had required extensive testing and maintenance operations for each of them. To overcome these issues, the RIPI platform has been based on an orchestration of independent services and micro-services, each of them having well-defined interfaces and functionalities. These components are loosely coupled allowing simplified maintenance processes, that only require unit and integration tests for each updated component. Moreover, most of the logic for handling data transformation was implemented with the support of specialized tools, like Liquid Studio, that allow to generate the code behind data transformations using a graphical environment based on functional blocks to be connected to build up the desired flows. This choice allowed a rapid prototyping and debugging of data transformations and conversions, by focusing more on the desired results than on the production of code. On the other hand, the logic behind BRAVa was developed and tested following a more traditional approach that resulted more demanding, but essential to realize a general-purpose tool not tied to a particular scenario.

In conclusion, redesigning the new platform, beyond overcoming various limitations and drawbacks of the RIAP platform, has provided additional functionalities and a basic infrastructure ready to support the implementation of future additional registries pertaining to RIPI.

9 | RELATED WORKS

Over the last years, the exploitation of digital registries to assess medical devices, drugs safety, and, more in general, healthcare treatments and procedures, has become a worldwide trend. Gliklich et al.[5] provide a reference handbook describing the best practices for design, operation, analysis, and evaluation of patient registries. The authors survey the different medical applications for which registries can be helpful. They note the potential advantages offered by registries that make widespread use of electronic health record systems. However, they also point out that various interoperability challenges still remain. Also, the authors highlight that medical device registries are essential for the identification and the study of medical devices outcomes. This kind of registries are used for many purposes, including short- and long-term surveillance, fulfillment of postmarket observational study commitments for regulatory bodies, and comparative safety and effectiveness assessments, including those in under-studied subpopulations.

Surveys about the currently available registries in Europe for implantable medical devices are provided by Niederländer at al..[3] and Lübbeke et al.[25] the last one focusing specifically on hip and knee prostheses.

However, all the authors focus either on classifying data contained in the different registries or on the medical aspects and impacts of the employment of such registries and do not deal with the technical aspects in realizing effective and interoperable registries, especially in scenarios with various and independent participants. Aim of this article was to specifically target technical issues and challenges, presenting a design perspective and implementation details of interest for computer scientists, engineers and practitioners in the area of information systems.

Registries of medical devices are part of the broader world of information systems for eGovernment,[26] in which it is well known that healthcare plays an important role. In particular, the RIPI platform project is an independent contribution that might be considered in the process going on in the Italian Public Administration for the integration of different local and regional information systems (which are currently very tight to local and regional operation contexts) through modern service-based architectures.[27] Similar initiatives can be found also in other national contexts.[28]

Outside the medical field, the term *registry* has been used in the world of semantic web services.[29] In this sense, the concept of medical registries integration, as proposed by the RIPI platform project, shares some similarities with the concept of *service registries federation*.[30] However, it must be taken into account that the approach adopted by RIPI focuses more on data integration than on service integration. Indeed, the work by Murakami et al.[31] applies this concept to the so-called language grid,[32] which is composed by both services and data registries for the creation of value-added multi-lingual support applications. The work by Baladrón et al.[33] follows a similar approach for service registries in the field of telecommunications.

The work presented in this article shares some similarities with the work by Ebad,[34] which analyzes the case of a platform for managing EHR (Electronic Health Records) in a structure in Saudi Arabia. The RIPI platform manages an anonymized subset of data of health records of patients, extended with registry-specific data, but the goals of both projects are similar. In both cases the implementations had the objectives of improving existing data collection and management procedures, as well as reaching higher standards in data quality, with data coming from heterogeneous sources. The work by Ebad[34] analyzes the failure of the described project, due to limited integration with other existing systems, limited training of the users and scarce modularity of the system. The RIPI platform was designed to be modular and extensible from the beginning, and for what concerns the user experience, many of the usability paradigms that were widely tested and refined for the RIAP platform, have been adopted with few adaptations to ensure users a smooth transition from the replaced platform.

10 | SUMMARY AND CONCLUSIONS

This article presented the architecture and the implementation details of a new platform supporting the data collection of the RIPI, which includes a set of digital healthcare registries each one targeting a specific category of implantable medical devices. We discussed the key aspects and choices that guided the design and pilot implementation of the RIPI platform, like the selected data layer design approach, the modularity of the data structures, the data validation processes, enhanced with the production of automatic feedback for improving data quality, and the adopted role-based access control management. Also, we focused on cost efficiency aspects, supported by the design of a cloud-oriented architecture based on micro-services. Finally, we presented the first evaluation results, and we discussed the impact of the choices adopted for the design and the pilot implementation of the RIPI platform, with the aim of providing references based on our experience for the future development of healthcare registries.

Data represent the main asset for a medical device registry. Data treatment ensuring quality, safety and integrity, as well as the privacy of patients, should be a foundation for its realization. This was the basic principle inspiring the design of the RIPI platform. Thus, the main architectural choices had the goal of achieving efficiency and efficacy in data treatment, while guaranteeing the above prerequisites. Consequently, data modeling represented a key aspect of the RIPI platform design process. Indeed, the defined data schemas and data transmission procedures, as well as the automatic generation of feedback reports about structural, syntactic and semantic correctness of transmitted data will allow to greatly improve the quality of the hosted registry data.

The selected data modeling approach used for the RIPI data format, designed as an extension of the data structure employed by the routinely hospital discharge data collection of the Italian Ministry of Health, might allow, in the future, to consider interoperability with the data flows of the new national health information system (NSIS). Furthermore, it minimizes the number of variables to be collected, being some of them already routinely collected by the hospital discharge data flow.

With the aim to optimize and simplify the platform operation, a unique formalism for writing all system rules was introduced. It provides a way to describe the business rules data have to comply with, and to describe rules that establish or modify the behaviors of the platform components. Eventually, this makes straightforward defining and updating the rules, and enables easy and fast maintenance operations.

Based on the above-mentioned formalism, the data validation engine BRAVa, with its rule format (the BRAVa Rules Format) was designed for defining rules and for automating rule checking. First tests performed with BRAVa demonstrated the effectiveness of the designed solution, which was able to detect and correct various anomalies within the real data sets used in the tests.

Furthermore, also the tests performed on SOnAR, paved the way toward a systematic approach for assessing and improving the registry data quality. Due to their essential role, both BRAVa and SOnAR can be considered the cornerstones of the platform.

More in detail, SOnAR represents a web front-end for BRAVa, making the latter suitable to support the healthcare registries implemented in the RIPI platform. SOnAR makes the outputs and reports of BRAVa usable for those feeding the registries. By this way, SOnAR contributes to the virtuous circle of creating value for them, since it allows to establish a direct feedback channel between the system and the users that become aware of the quality of the uploaded data and can possibly commit toward its improvement. In addition to this, SOnAR was designed to include subcomponents in charge of transforming unstructured data, respecting specific formats, to a well-defined structured format. This makes data ready for a reliable automated quality assessment and BRAVa is the engine that effectively analyzes data quality in a reasonably short time (minutes) over historical databases by associating to each record its provenance information. This implies that data becomes usable for research purposes, with the awareness of any possible weakness in them. Such weaknesses may be acceptable in certain scenarios while, in others, they can introduce unacceptable deviations in the input datasets and consequently in the results. A solution to overcome this issue is to filter out problematic records and consequently to work on cleaner datasets and eventually obtain more reliable results.

The adopted authorization model is another key point of the RIPI platform, as it reflects the organization of the Italian National Health Service,[21] allowing every scope that in the platform maps entities of the National Health Service to be managed by its actual managers.

Overall, the design process of the RIPI platform led to a modular and extensible architecture. The platform is designed to allow interoperability with national information systems and registries of other countries as required, for example, by RIAP within the agreement signed with the UK National Joint Registry (NJR).[6] Also, its native cloud-oriented design makes it ready to exploit the elasticity offered by cloud platforms in term of computing resources. Indeed, the possibility to resize on-demand the underlying resource pools allows to scale with respect to an increasing number of users.

Building a modular, extensible, interoperable and cloud-oriented platform has both pros and cons in comparison with more traditional, manual approaches to healthcare registries. To build up an infrastructure that is fully modular, from the points of view of both data and structures of the components, requires an additional effort at least in the design stages, where all the aspects of the project must be conceived in that way. This approach, however, produces an infrastructure that is easier to realize because it is based on modules whose design has been deeply analyzed and consolidated in the previous steps, and because these modules can be realized and tested separately, and then integrated and tested together. Moreover, the same advantages may be applied when the maintenance processes are considered, since they are supported by extensive documentation and often localized to one or more modules with a relatively simple structure. By this way, cost for maintenance is reduced over time. Finally, the modular structure allowed many of the components to be implemented using serverless computing facilities, which bill the client only for the time in which they are actually used to actively process data using, in this way, only the right amount of computational resources.

Overall, the following factors might be considered to assess the potential impact and the success of the presented technologies and techniques:

- Capability to collect and manage a larger volume of data with the same, or even less effort spent using the traditional manual solutions.
- Capability to process a larger volume of data in less time. This can allow registries based on the proposed technology to quickly detect issues and alert clinicians, authorities and patients.
- Capability to improve the data quality, by applying automatic techniques which can detect potential anomalies of data in terms of, for example, syntactical and semantical (business rules) factors. This capability, compared to the manual approaches, concurs in reducing the effort in data analysis, and increases the accuracy of the results. Furthermore, manual solutions often require the assessment of data quality is performed by humans which, even if well trained, may introduce bias in the process or, with large amounts of data, may fail in detecting subtle issues. With the proposed

automated approach, controls are thoroughly applied as prescribed by defined rules, following a systematic and standard process.

- Overall, a registry that can reach its surveillance and alerting purposes faster and with more efficiency may represent a significant decisional support tool for policy makers, in particular e-Government like contexts.

In Italy, several procedures adopted by Public Administrations produce data that can be used to extract anonymized statistics. A recent example of such procedures is the Italian FOIA—Freedom of Information Act (see https://foia.gov.it/strumenti/indicazioni-operative-registro-accessi). However, the products of these procedures are currently stored and exported by several different software systems with specific formats, often without considering application of business rules. In this sense, several solutions proposed in this paper, such as the authentication module, can be reused in any other e-Government application. Furthermore, as local Public Administrations are releasing more and more open data, often following different formats, a similar approach can be used to automatically create open data portals at the national and international levels.

Currently, the RIPI platform represents both an added value for the national healthcare system and a foundation for a more complex infrastructure that can further evolve over the years. In conclusion, other than the contribution provided by the design process of the platform for its main purposes, the experience presented in this article may represent a reference for the future design and the evolution of healthcare registries,[35] at least in the framework defined by the Italian law. More in general, although many of the presented components fit exactly in the setting of the registries that are included in RIPI, a particular effort was put in designing them as more general as possible, consequently making them suitable for any other data collection scenario. As a first example, BRAVa can analyze any kind of data, given that the appropriate rules are provided. With the appropriate rules, BRAVa can also act as the source of information on structure and values for any kind of data. Another example can be the design of AmAGeT: indeed, it can be easily extended to any hierarchy, independently of its size, remaining unchanged the main assumptions. AmAGeT, practically answering to questions like "I have role X, can I do Y?," can represent the authority on roles of any system, not only limited to data collection scenarios.

In the end, it is possible to conclude that the technologies presented in this paper are widely applicable to different contexts and they may fit in any scenario where:

- Data must be collected synchronously and asynchronously from heterogeneous sources and a strong data quality evaluation is required upon data collection.
- A structured role-based access control is required to discipline the access of users to data and functionalities depending on their roles.
- Scalability and interoperability are key requirements.

All these features indeed describe typical e-Government settings, where a considerable number of different contributors generally feed nation-sized databases.

ACKNOWLEDGMENT

This study was coordinated by the Italian National Institute of Health (ISS) and its realization was possible thanks to the contribution from the General Directorate of Medical Devices and the Pharmaceutical Service at the Ministry of Health. Open Access Funding provided by Universita degli Studi di Roma La Sapienza within the CRUI-CARE Agreement.

AUTHOR CONTRIBUTIONS

Duilio Luca Bacocco: Conception or design of the work, data collection, data analysis and interpretation, drafting the article. **Eugenio Carrani:** Conception or design of the work, data collection, data analysis and interpretation, drafting the article, critical revision of the article, final approval of the version to be published. **Bruno Ciciani:** Critical revision of the article, final approval of the version to be published. **Pierangelo Di Sanzo:** Drafting the article, critical revision of the article. **Francesco Leotta:** Drafting the article, critical revision of the article. **Marina Torre:** Conception or design of the work, data collection, data analysis and interpretation, drafting the article, critical revision of the article, final approval of the version to be published.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Duilio Luca Bacocco https://orcid.org/0000-0003-0593-5457
Eugenio Carrani https://orcid.org/0000-0002-5362-1439
Bruno Ciciani https://orcid.org/0000-0003-3190-2333
Pierangelo Di Sanzo https://orcid.org/0000-0001-6136-6303
Francesco Leotta https://orcid.org/0000-0001-9216-8502
Marina Torre https://orcid.org/0000-0003-3851-187X

REFERENCES

1. Lübbeke A, Silman A, Prieto-Alhambra D, Adler A, Barea C, Carr A. The role of national registries in improving patient safety for hip and knee replacements. *BMC Musculoskeletal Disorders*. 2017;18:414. doi:10.1186/s12891-017-1773-0
2. Graves SE. The value of arthroplasty registry data. *Acta Orthopaedica*. 2010;81(1):8-9. doi:10.3109/17453671003667184
3. Niederländer C, Wahlster P, Kriza C, Kolominsky-Rabas P. Registries of implantable medical devices in Europe. *Health Policy*. 2013;113(1):20-37. doi:10.1016/j.healthpol.2013.08.008
4. Gliklich RE, Dreyer NA. *Registries for Evaluating Patient Outcomes: A User's Guide*. 2nd ed. Agency for Healthcare Research and Quality (US); 2010.
5. Gliklich RE, Dreyer NA, Leavy MB. *Registries for Evaluating Patient Outcomes: A User's Guide*. 3rd ed. Agency for Healthcare Research and Quality (US).; 2014.
6. Torre M, Franzò M, Carrani E, et al. A new collaboration on the horizon: the National Joint Registry (NJR) and the Italian Arthroplasty Registry (RIAP) towards an agreement upon a common component database and device classification systems harmonisation. Proceedings of the 8th International Congress of Arthroplasty Registries. International Society of Arthroplasty Registries; 2019; Leiden.
7. Decreto del Presidente del Consiglio dei Ministri 3 marzo 2017, Identificazione dei sistemi di sorveglianza e dei registri di mortalita', di tumori e di altre patologie. (17A03142), (GU Serie Generale n.109 del 12-05-2017); 2017.
8. Torre M, Carrani E, Franzò M, et al. Il Registro Italiano delle Protesi Impiantabili: una nuova realtà per la sicurezza del paziente. *Boll Epidemiol Naz*. 2021;2:16-23. doi:10.53225/BEN_015
9. Torre M, Laricchiuta P, Luzi I, et al. *Italian Arthroplasty Registry Project. Better data quality for better patient safety. Fourth Report 2017 – Addendum*. Il Pensiero Scientifico; 2018.
10. Torre M, Luzi I, Carrani E, Leone L, Romanini E, Zanoli G. *Progetto Registro Italiano Artroprotesi - Primo Report – Idea, Sviluppo e Avvio*. Il Pensiero Scientifico; 2014.
11. Carrani E, Franzò M, Bacocco D, Cornacchia A, Torre M. Tracciati Record RIAP; 2020. Accessed September 27, 2020. https://riap.iss.it/riap/it/strumenti/documenti-tecnici/tracciati-record-riap/
12. Carrani E, Cornacchia A, Bacocco D, Franzò M, Torre M. Procedura operativa organizzazione dei file e trasmissione dati riap; 2020. Accessed September 27, 2020. https://riap.iss.it/riap/it/strumenti/documenti-tecnici/organizzazione-file-e-trasmissione-dati-riap/
13. Carrani E, Cornacchia A, Torre M. Procedura controllo di qualità dei dati; 2020. Accessed September 27, 2020. https://riap.iss.it/riap/it/strumenti-documenti-tecnici-controllo-di-qualita-dei-dati-riap/
14. Ministero della Salute, Nuovo Sistema Informativo Sanitario - NSIS; 2020. Accessed September 19, 2020. http://www.salute.gov.it/portale/temi/p2_4.jsp?lingua=italiano&tema=Piani,20finanziamenti20e20monitoraggio20del20SSN&area=sistemaInformativo
15. Fabian R. *Data-Oriented Design: Software Engineering for Limited Resources and Short Schedules*. 2018. https://dataorienteddesign.com/dodbook.pdf
16. Urban S, Dietrich S. Object data models. In: Liu L, Tamer Özsu M, eds. *Encyclopedia of Database Systems*. Springer; 2009.
17. Ferraiolo D, Cugini J, Kuhn DR. Role-based access control (RBAC): features and motivations. ACSAC; 1995:241-48.
18. Aaron EC. Biba integrity model. In: Encyclopedia of Cryptography and Security, ed. *Encyclopedia of Cryptography and Security*. Springer; 2011:81.
19. Garante Privacy. GDPR - Regolamento 2016/679; 2020. Accessed December 16, 2021. https://www.garanteprivacy.it/regolamentoue
20. Ministero della Salute, La scheda di dimissione ospedaliera (SDO); Accessed September 15, 2020. http://www.salute.gov.it/portale/temi/p2_6.jsp?lingua=italiano&id=1232&area=ricoveriOspedalieri&menu=vuoto
21. Ministero della Salute, Sistema Sanitario Nazionale; Accessed September 28, 2020. http://www.salute.gov.it/pianoNazionaleIntegrato2015/dettaglioPianoNazionaleIntegrato2015.jsp?cap=capitolo2&sez=pni-cap2-autoritacompetenti&id=1104
22. Bacocco D, Carrani E, Torre M. Infrastruttura informatica del Registro Italiano Protesi Impiantabili: case-study per la progettazione della piattaforma di raccolta dei dati. *Rapporti ISTISAN*. 2021;21/8. https://www.iss.it/rapporti-istisan/-/asset_publisher/Ga8fOpve0fNN/content/rapporto-istisan-21-8-infrastruttura-informatica-del-registro-italiano-protesi-impiantabili-case-study-per-la-progettazione-della-piattaforma-di-raccolta-dei-dati.-duilio-luca-bacocco-eugenio-carrani-marina-torre
23. Liquid Technologies. Liquid studio; 2021. Accessed December 16, 2021. https://www.liquid-technologies.com/xml-studio
24. Bacocco D, Carrani E, Cornacchia A, Torre M. BRAVa: improving the quality of RIAP data through automated validation. Proceedings of the 10th International Congress of Arthroplasty Registries, 2nd Virtual Congress. International Society of Arthroplasty Registries; 2021; Copenhagen, Denmark.
25. Lübbeke A, Silman A, Barea C, Prieto-Alhambra D, Carr A. Mapping existing hip and knee replacement registries in Europe. *Health Policy*. 2018;122(5):548-557.

26. Viscusi G, Batini C, Mecella M. *Information Systems for eGovernment: A Quality-of-Service Perspective*. Springer Science & Business Media; 2010.
27. Mecella M, Batini C. Enabling italian e-government through a cooperative architecture. *Computer*. 2001;34(2):40-45.
28. Kovac M. E-health demystified: an e-government showcase. *Computer*. 2014;47(10):34-42.
29. Paolucci M, Kawamura T, Payne TR, Sycara K. Importing the semantic web in UDDI. Proceedings of the International Workshop on Web Services, e-business, and the Semantic Web; 2002:225-236; Springer.
30. Baresi L, Miraz M. A distributed approach for the federation of heterogeneous registries; 2006:240-251; Springer.
31. Murakami Y, Tanaka M, Lin D, Ishida T. Service grid federation architecture for heterogeneous domains; Proceedings of the 2012 IEEE 9th International Conference on Services Computing; 2012:539-546; IEEE.
32. Ishida TL. anguage grid: an infrastructure for intercultural collaboration; 2006:5-pp; IEEE.
33. Baladrón C, Aguiar J, Carro B, et al. User-centric future internet and telecommunication services. *Towards Future Internet*. 2009;1:217-226.
34. Ebad SA. Healthcare software design and implementation—A project failure case. *Softw Pract Exp*. 2020;50:1258-1276. doi:10.1002/spe.2807
35. Abbonizio F, Arcieri R, AICE, Giampaolo A. Registro Nazionale Coagulopatie Congenite. Rapporto 2019. Rapporti ISTISAN 2021(21/15).

How to cite this article: Bacocco DL, Carrani E, Ciciani B, Di Sanzo P, Leotta F, Torre M. Design and implementation of the new Italian healthcare digital interoperable registry for implantable medical devices. *Softw Pract Exper*. 2022;1-25. doi: 10.1002/spe.3130