

Universiteit
Leiden
The Netherlands

Healthcare information system engineering: AI technologies and open source approaches

Shen, Z.

Citation
Shen, Z. (2025, December 3). *Healthcare information system engineering: AI technologies and open source approaches*. Retrieved from https://hdl.handle.net/1887/4284431

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/4284431

Note: To cite this publication please use the final published version (if applicable).

Chapter 2

Federated Architecture for Multinational Clinical Trials

The Systematic Tool to Reduce Inappropriate Prescribing (STRIP) is a clinical intervention method crafted to deal with polypharmacy problems which are incurred by the concurrent use of multiple drugs. STRIP has been proven to be effective and is included in the Dutch national guideline for polypharmacy. To boost the usage of STRIP in clinical practices, a web application called the STRIP Assistant (STRIPA) was developed and further evaluated as user-friendly, efficient and effective by Dutch physicians. STRIPA has now evolved into a software tool that supports a large multinational randomized clinical trial (RCT). However, in order to successfully implement and use the application in such an RCT, several issues, including multilingual support, clinical data security, data accessibility and consistency, need to be addressed. In this chapter, we present an overhauled STRIPA prototype with a lightweight data integration component that supports multinational implementations, ensures data consistency across countries, and maintains data accessibility and security. The component includes a high-level information architecture, data models redesigned to generalize data entities from all countries, and the ETL processes that integrate diverse data sources and transfer data between databases.

This work was originally published as: Shen Z., Meulendijk, M. and Spruit, M. (2016). A Federated Information Architecture for Multinational Clinical Trials: STRIPA Revisted. *Proceedings of the 24th European Conference on Information Systems (ECIS)*, Prototypes. Paper 2. https://aisel.aisnet.org/ecis2016_prototypes/2

2.1 Introduction

Polypharmacy, defined as the chronic use of multiple medications by a patient, often leads to severe clinical problems or accidents if it is inappropriate, including adverse drug effects, under prescribing, overtreatment, low patient compliance and decreased drug adherence (Björkman et al., 2002; Claxton et al., 2001; Munger, 2010; Steinman et al., 2006; Wright et al., 2009). The elderly, generally adults aged over 65 years, are the main sufferers of polypharmacy (Munger, 2010). The demographic shift towards an elderly population among developed countries brings necessity and urgency to the implementation of an effective solution to reduce the risks and maximize the benefits of polypharmacy.

The STRIP assistant (STRIPA) is a web application that was built to help physicians to perform medication reviews in the case of polypharmacy. A standardized medication review method named Systematic Tool to Reduce Inappropriate Prescribing (STRIP) is incorporated into the application. Meulendijk et al. (2015b) presented the main components of the application, including functional architecture, user interface, decision rule engine, and data formats. It has gained recognition as a user-friendly, effective and efficient tool among Dutch physicians (Meulendijk et al., 2015a, 2016). Now, it is a part of a large-scale randomized clinical trial (RCT) that aims at investigating impacts of medication optimization in the multi-morbid elderly among four European countries: the Netherlands, Switzerland, Ireland and Belgium.

In order to implement and use the application across countries, relevant data from each country need to be collected and properly managed in the first place. These data contain medical data on medications, drug interactions and standardized medical terminology, and patient health records. Besides, usage data recorded in each country should be accumulated for further analysis. As the number of data sources grows, the complexity and diversity also increases. How to properly integrate and manage such diverse data sources becomes a challenge.

In this chapter we present an overhauled prototype that deals with the above data integration issues so as to support conducting medication reviews in multiple countries. The prototype, named as STRIPA.EU, expands the previous version of STRIPA with a data integration component which mainly consists of a number of databases and ETL processes that populate the databases. The component is able to integrate necessary heterogeneous data sources from different countries, easily update the data as data sources renew, and gather country-specific usage data together into a centralized database. In general, the application works independently as a cus-

tomized application for systematic medication reviews in each country. However, it also has a centralized database that maintains data consistency across countries and subsequently keeps research integrity and consistency of the large international RCT.

The remainder of the chapter is organized as follows. At first, related work and the detailed design objectives of the present research are discussed. Second, the data integration component that extends STRIPA to multiple countries is elaborated from three aspects: data representation, integration architecture and the ETL implementation. Preliminary results of the implementation of STRIPA.EU are briefly discussed in the following. The final section presents contributions of this research and possible future research aspects.

2.2 Background

2.2.1 Status Quo of Data Integration

Data integration aims at providing a unified view of heterogeneous data residing at different sources (Calì et al., 2002). There are three steps in data integration: extracting data from homogeneous or heterogeneous data sources, transforming data into a common structure and loading it into the final target (Muilu et al., 2007). These steps are commonly shortened as ETL. Data integration has become of great importance in the medical/clinical domain because of the tremendous increasing volume and complexity of data gathered at the medical community (Branson et al., 2008; Brazhnik and Jones, 2007). Many studies regarding integrating heterogeneous data sources in the areas of biomedical informatics and information technology have been conducted over the last decades (Karasavvas et al., 2004; Louie et al., 2007; A Seoane et al., 2013). However, there is little literature focusing on integrating clinical data sources, especially pharmaceutical data from various countries. But the abundance of information about drugs and its heterogeneous nature among countries does pose a challenge to data integration (Meulendijk et al., 2015b).

Louie et al. (2007) divided data integration into two axes: integration architecture, and data representation. Integration architecture refers to the final destination into which data and metadata are loaded. There are two main types of integration architecture: data warehouse and federated database system (FDB) (Louie et al., 2007). The need for combining various independent data sources into a secure, but easily accessible data architecture has given rise to the development of federated database systems (Calì et al., 2002; Ziegler and Dittrich, 2004). A FDB system has a reliable

infrastructure for data sharing and collaboration across disparate data sources while maintaining security and privacy locally. A number of aspects of such a system, such as privacy, security, access control and availability, have been well-studied both in research and in industry (Ansper et al., 2013).

In relational databases, data representation refers to a conceptual data model which represents the common schema of diverse data sources (Louie et al., 2007). For data integration, data models give a unified and structured view of the integrated and reconciled data sources, and also offer the elements for expressing the queries over integrated systems (Calì et al., 2002). Identifying such a data representation from heterogeneous data sources is mostly time consuming and requires extensive human interactions (Zhao and Ram, 2007). Over the last three decades numerous methods have been proposed to facilitate this process (Batini et al., 1986; Clifton et al., 1998; Passi1 et al., 2002; Ramesh and Ram, 1995; Zhao and Ram, 2007). Data models for STRIPA.EU are created by detecting schematic correspondences among data sources at both structural and semantic levels.

2.2.2 Design Objectives

As stated by Meulendijk et al. (2015b), data exchange and integration for STRIPA are difficult because of incompatible information systems and customized international classification standards for drugs and diseases in each country. In particular, since most countries use medical coding standards modified on the basis of international standards, or even completely disconnected ones, data encoded with country-specific codes are not fully interoperable. But decision rules of STRIPA are crafted with international codes, and they only recognize input encoded in such codes. Hence, semantic matching of data sources at instance level is necessary in data integration. Moreover, data of drugs and drug interactions from third-party companies vary a lot in terms of both schemas and formats. Last but not least, multilingualism of data sources adds another dimension of complexity to data integration. Table 2.1 demonstrates the diversity and complexity of data sources from different countries.

To overcome these issues, we developed a data integration component that is presented in three pillars: data representation, ETL processes and integration architecture. The component is added into STRIPA as a built-in tool. The design objectives when developing the new prototype were as follows:

(1) *to create a federated database system with a unified schema that supports implementation of the application in multiple countries*

(2) *to develop ETL processes that populate both a federated database and multiple localized databases in the predefined schema*

(3) *to regularly update and maintain medical data, including drugs and drug interactions, in each country*

(4) *to have an architecture which makes the application running in each country on customized local data sources and all the locally collected data remain under the control of the local authorities for the sake of privacy and security*

(5) *to create a unique identifier for each data item in the system*

Supplier	Database / Standard	Scope	Format	Country	Languages
Z-Index	G-Standard	Medications, Clinical Interactions	Fixed Width	Netherlands	Dutch
RIVM	ICD-10	Episodes	XML	Netherlands	Dutch
APB	Delphi-Care	Medications, Clinical Interactions	Fixed Width	Belgium	French, Dutch
FOD	ICD-10	Episodes	CSV	Belgium	French, Dutch
HCI Solutions	INDEX	Medications, Clinical Interactions	XML	Switzerland	German, French
DIMDI	ICD-10	Episodes	XML	Switzerland	German
HelixHealth	Safescript	Medications, Clinical Interactions	Fixed Width	Ireland	English
WHO	ICD-10	Episodes	XML	Ireland	English
MedDRA MSSO	MedDRA	Adverse Events	CSV	Netherlands, Belgium, Switzerland, Ireland	Dutch, French, German, English
Regenstrief	LOINC	Measures, Laboratory Tests	CSV	Netherlands, Belgium, Switzerland, Ireland	Dutch, French, German, English

Table 2.1: Data sources from different countries

2.3 Data Integration Component

Details of the data integration component that supports and realizes the design objectives are described in continuation. It is elaborated from three main aspects: data representation, integration architecture, and the ETL processes.

2.3.1 Data Representation

As discussed in the previous section, determining data models that represent a common schema of all heterogeneous data sources is crucial for data extraction and trans-

formation at the beginning of an ETL process. There are a great amount of methods or tools that have been developed for detecting schema-level correspondences. In this chapter we identified common schema elements (i.e., data entities, relations and attributes) by matching schema elements of all the available data sources at both structural and semantic level. Meanwhile, unnecessary elements are filtered out from the identified common schema according to the system requirements.

To begin with, data required for the application are summarized into two categories: medical knowledge and patient data. Medication data, medical codes and decision rules constitute the knowledge base of STRIPA. As the core data component, medication data contain all available drugs, drug interactions and adverse events in a country. Decision rules in the application are implemented based on established clinical guidelines. Hospitals register a variety of information about a patient on patient health records, but STRIPA only requires patients' medical data on treatments, diagnoses and measurements. Moreover, patient health records need to be encoded with standardized medical codes because both decision rules and drug interactions are composed with standardized codes and they can only recognize encoded patient's information.

As shown in Table 2.1, medication data are provided by many different data suppliers. By matching data entities of all data sources, four common data entities are derived: medication, product, substance and interactions. Medication stores generic drugs and their substances are saved in substance. Besides, each generic drug might have a list of products in the market and this information is recorded in product. Interactions between two generic drugs are put into interactions. Figure 2.1 shows the relationship between these data entities and critical attributes of each data entity are also listed.

Patient health records are represented by three data entities: diagnoses, measurements and treatments. Standardized medical codes that are utilized to encode patient health records include LOINC, SNOMED, modified and translated versions of ICD-10 in each country. They are represented with four data entities: episodes, dosage-units, dosage-intervals and measures. Episodes is used to instantiate diagnoses with ICD-10 codes. For instance, one record of patients' diagnoses refers to an episode identifier given the patient's number and a timestamp. Measures is extracted from LOINC and it relates to patients' measurements the same as episodes to diagnoses. Together with medications, dosage-intervals and dosage-units generate instances of treatments of patients. Figure 2.1 demonstrates how patient health records are modelled with three layers of data entities. Splitting patient health records into a set of

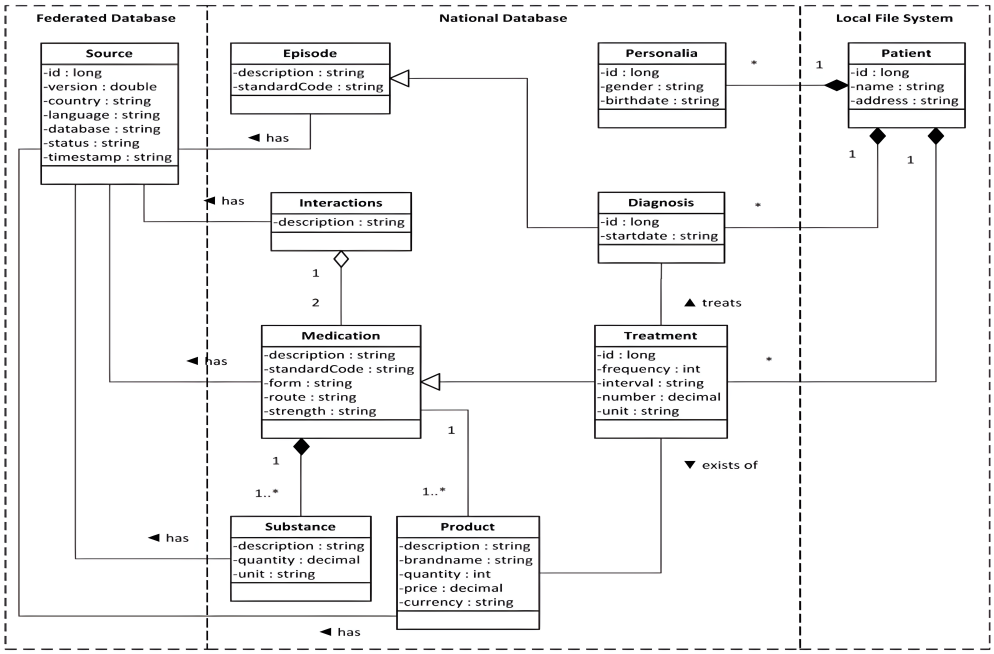


Figure 2.1: Class Diagram depicting main data models of STRIPA

data entities reduces data redundancy in the first place because of database normalization. It also provides an easy way of updating medical codes in each country. For example, adding new ICD-10 codes into episodes requires no modification for diagnoses and the rest of database.

Since each country regularly renews its medication data by adding new medications or removing old ones, medication data in the application need to be updated accordingly. In order to keep track of medication data during a series of updates, a unique identifier for each data item is created and recorded in sources before data items are added into relevant databases. Table sources also stores other metadata which keeps medical knowledge data manageable. Figure 2.1 shows attributes of sources and relationships between sources and other data entities. In addition, to maintain the privacy of patients patients' identification information, such as name, address, is stored as patient in local file system within hospitals, and the rest are put into personalia in national databases. They are linked with a unique patient identifier.

2.3.2 Integration Architecture

Integration architecture provides an outline of databases in a system from a data integration perspective. Data flows are also included to give a dynamic view of how data are transferred in-between databases. Building a reliable integration architecture is crucial for any system with distributed and autonomous databases in different locations. For STRIPA.EU a federated database system was developed instead of a data warehouse for a number of reasons. At first, unlike a data warehouse, the database in each country is not a subset of a centralized database, whereas each contains all data necessary for the application and takes nothing from the centralized one. Secondly, patient health records collected locally need to remain under the control of the local authorities for the sake of privacy and security. Only collected research data in each country are allowed to be transferred into the centralized database. Finally, a federated database system grants autonomy to the database in each country, which guarantees national databases having power to adapt themselves in response to changes.

Figure 2.2 depicts the integration architecture that was designed for STRIPA.EU. It is a federated database system which is composed of a federated database and a number of autonomous national databases. As shown in the figure, each country has its own national database, and private patient data is extracted and saved in the local file system within hospitals. Physical independence of the local file system prohibits unauthorized access and use of locally collected private patient data in each country. National databases are autonomous and independent of each other. The autonomy of national databases ensures that countries could customize their databases from various aspects such as schema and data instances. Because of the high independence of national databases, the application in each country has a high system availability. In particular, database breakdown in any country does not affect other countries. The functions of the federated database are twofold: storing and managing metadata for data in all countries, and gathering user log data for further user behaviour analysis (Pachidi et al., 2014). There are two ETL processes for the implementation of STRIPA.EU in each country. The first ETL process takes the aforementioned external data sources and transfers them into a national database while writing metadata into the federated database. Another ETL is designed to accumulate application usage data from all countries. A detailed description of ETL processes will be given in the following.

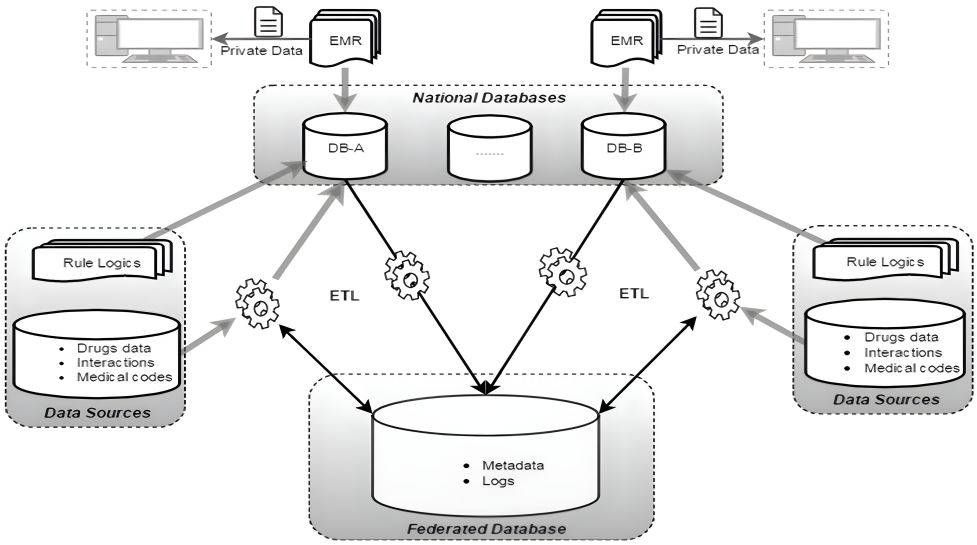


Figure 2.2: Information Architecture for data integration in STRIPA.EU

2.3.3 ETL

Determining data models and designing integration architecture only lay the groundwork for the federated database system. In order to provide the client side with a reliable data service, all relevant data need to be imported and heterogeneous data sources, including the most important medication data, need to be transferred using well-designed ETL processes. Therefore, ETL plays a decisive role in the implementation of STRIPA.EU.

There are a number of challenges ahead when it comes to constructing a successful ETL implementation: 1) significant dissimilarity between external data sources and target data structure; 2) external data sources are encoded in diverse formats, including CSV, fixed width and XML; 3) ETL needs to be autonomous, preferably just a click away. A powerful open source tool (Talend Open Studio for Data Integration) has been used to operationalize the ETL processes. It is capable of tackling a variety of data formats. And its graphic interfaces make ETL easier to follow and evaluate by non-IT experts. In the following paragraph an example of the ETL processes developed for medication data in the Netherlands is elaborated to give a glimpse of the complexity.

Medication data in the Netherlands are provided in positional flat files by the G-Standaard. There is a significant structural difference between data sources and the

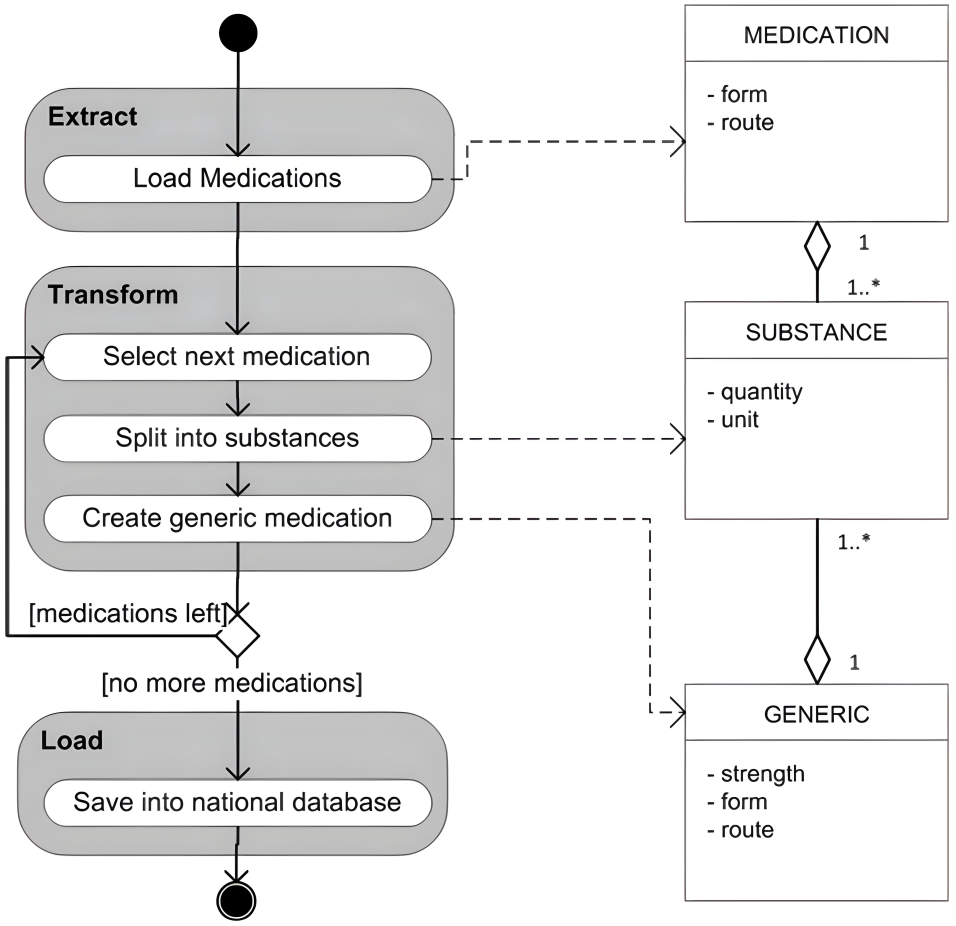


Figure 2.3: The ETL process of extracting medication data in the Netherlands

above data model. In specific, generic medications needed in the national database do not exist in data sources, whereas only medicinal products, coupled with product relevant information, such as product name, form, route, etc. are stored separately in different files. ETL should generate generic medications by grouping medicinal products based on their substances. Figure 2.3 conceptualizes the ETL process with the Process-Deliverables Diagram (van de Weerd and Brinkkemper, 2009).

As mentioned above, the regular updates of medication data require us to renew drugs and drug interactions within the system accordingly. And the updating process has been built as a part of the ETL implementation. The following process deliverable diagram illustrates the process of updating a database with its new released external

data sources. It is coded in a python script and automated with a batch file that loads and directly runs the script.

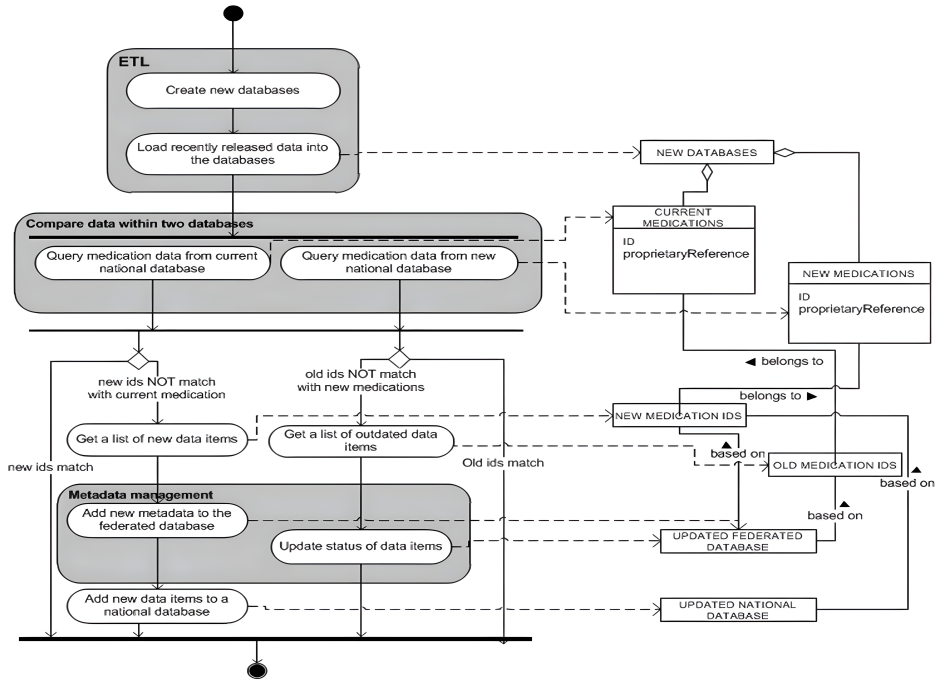


Figure 2.4: Process Deliverable Diagram depicting the data updating process

2.4 Evaluation & Contribution

Evaluation of the application has been carried out or is currently planned in multiple steps. At first, ETL processes are tested on data completeness, data consistency and data integrity. During the development of ETL processes, random manual comparisons of data between databases and data sources have been performed. More systematic testing tools and methods will be developed to assess ETL performance in terms of efficiency and maintenance before the implementation of the software. The next step of the evaluation focuses on the integration architecture. How well this architecture supports the implementation and use of the application in multiple countries is investigated. Security and privacy issues of the clinical trials are properly addressed with the architecture. Moreover, having integrated data sources from four

data sources in the given architecture shows its technical feasibility and suitability for such multinational clinical trials.

The successful development of STRIPA.EU and initial feedback collected from physicians suggest that the proposed data integration component fulfils the requirements of multinational clinical trials. These requirements, such as healthcare interoperability, privacy and security, are very common among many other large-scale clinical trials. Therefore, this leads us to believe that the federated information architecture and data integration methods have a good potential to be reused in other clinical trials.

2.5 Conclusion

This chapter presented a federated architecture for multinational clinical trials through the case study of STRIPA.EU. The proposed solution addresses key challenges in multinational clinical data integration, including multilingual support, data security, and consistency across countries. By implementing a federated database system with carefully designed data models and ETL processes, we achieved a balance between local autonomy and centralized coordination. The architecture successfully supports a clinical trail across four European countries while maintaining data privacy and security requirements.