Knowledge and Information Systems (2021) 63:2209–2230
https://doi.org/10.1007/s10115-021-01589-x

REGULAR PAPER

Improving security in NoSQL document databases through model-driven modernization

Alejandro Maté[1] · Jesús Peral[1] · Juan Trujillo[1] · Carlos Blanco[2] · Diego García-Saiz[2] · Eduardo Fernández-Medina[3]

Received: 30 October 2020 / Revised: 14 June 2021 / Accepted: 19 June 2021 /
Published online: 13 July 2021

Abstract
NoSQL technologies have become a common component in many information systems and software applications. These technologies are focused on performance, enabling scalable processing of large volumes of structured and unstructured data. Unfortunately, most developments over NoSQL technologies consider security as an afterthought, putting at risk personal data of individuals and potentially causing severe economic loses as well as reputation crisis. In order to avoid these situations, companies require an approach that introduces security mechanisms into their systems without scrapping already in-place solutions to restart all over again the design process. Therefore, in this paper we propose the first modernization approach for introducing security in NoSQL databases, focusing on access control and thereby improving the security of their associated information systems and applications. Our approach analyzes the existing NoSQL solution of the organization, using a domain ontology to detect sensitive information and creating a conceptual model of the database. Together with this model, a series of security issues related to access control are listed, allowing database

✉ Jesús Peral
jperal@dlsi.ua.es

Alejandro Maté
amate@dlsi.ua.es

Juan Trujillo
jtrujillo@dlsi.ua.es

Carlos Blanco
Carlos.Blanco@unican.es

Diego García-Saiz
Diego.Garcia@unican.es

Eduardo Fernández-Medina
Eduardo.Fdezmedina@uclm.es

1 Lucentia Research Group, Department of Software and Computing Systems, University of Alicante, Alicante, Spain

2 ISTR Research Group, Department of Computer Science and Electronics, University of Cantabria, Santander, Spain

3 GSyA Research Group, Institute of Information Technologies and Systems, Information Systems and Technologies Department, University of Castilla-La Mancha, Ciudad Real, Spain

designers to identify the security mechanisms that must be incorporated into their existing solution. For each security issue, our approach automatically generates a proposed solution, consisting of a combination of privilege modifications, new roles and views to improve access control. In order to test our approach, we apply our process to a medical database implemented using the popular document-oriented NoSQL database, MongoDB. The great advantages of our approach are that: (1) it takes into account the context of the system thanks to the introduction of domain ontologies, (2) it helps to avoid missing critical access control issues since the analysis is performed automatically, (3) it reduces the effort and costs of the modernization process thanks to the automated steps in the process, (4) it can be used with different NoSQL document-based technologies in a successful way by adjusting the metamodel, and (5) it is lined up with known standards, hence allowing the application of guidelines and best practices.

Keywords NoSQL databases · Security · Modernization process · Ontology

1 Introduction

Enormous amounts of data are already present and still rapidly growing due to heterogeneous data sources (sensors, GPS and many other types of smart devices). There has been an increasing interest in the efficient processing of these unstructured data, normally referred to as "Big Data", and their incorporation into traditional applications. This necessity has made that traditional database systems and processing need to evolve and accommodate them. Therefore, new technologies have arisen focusing on performance, enabling the processing of large volumes of structured and unstructured data. NoSQL technologies are an example of these new technologies and have become a common component in many enterprise architectures in different domains (medical, scientific, biological, etc.).

We can distinguish four different categories of NoSQL databases: (1) Key/Value, where data are stored and accessible by a unique key that references a value (e.g., DynamoDB, Riak, Redis, etc.); (2) Column, similar to the key/value model, but the key consists of a combination of column, row and a trace of time used to reference groups of columns (e.g., Cassandra, BigTable, Hadoop/HBase); (3) Document, in which data are stored in documents that encapsulate all the information following a standard format such as XML, YAML or JSON (e.g., MongoDB, CouchDB); (4) graph, the graph theory is applied and expanding between multiple computers (e.g., Neo4J and GraphBase).

One of the main challenges is that these new NoSQL technologies have focused mainly on dealing with Big Data characteristics, whereas security and privacy constraints have been relegated to a secondary place [1–3], thus leading to information leaks causing economic loses and reputation crisis. The main objective of our research deals with incorporating security in NoSQL databases, focusing on document databases as a starting point. In this way, this paper presents the first modernization approach for introducing security in NoSQL document databases through the improvement of access control.

The proposed approach consists of two stages: (1) the analysis of the existing NoSQL solution (using a domain ontology and applying natural language processing, NLP) to detect sensitive data and create a conceptual model of the database (reverse engineering); (2) the identification of access control issues to be tackled in order to modernize the existing NoSQL solution. At a later stage, which we will not see in this paper, different transformation rules for each detected security issue will be applied. These transformation rules will consist on

a combination of privilege modifications, new roles and the creation of views, which can be adapted by the database designer. Finally, the implementation of the transformation rules will be carried out. In order to evaluate our proposal, we have applied it to a medical database implemented using the document NoSQL database MongoDB.

The great advantages of our framework are that: (1) it takes into account the context of the system thanks to the introduction of domain ontologies; (2) it helps avoid missing critical security issues since the analysis is performed automatically; (3) it reduces the effort and costs of the modernization process thanks to the automated steps in the process; (4) it can be used with different NoSQL technologies in a successful way by adjusting the metamodel; and (5) it is lined up with known standards, hence allowing the application of guidelines and best practices.

The main contributions of this paper are summarized as follows:

- The first general modernization approach for introducing security through improved access control in NoSQL document databases. We focus on document databases although our proposal could be applied to other NoSQL technologies, such as columnar and graph-based databases.
- Our approach adapts to each domain by using an specialized ontology that allows users to specify the sensitive information.
- The automatic analysis of data and database structure to identify potential security issues.
- The generation of the security enhanced database model, including the automatic generation of a solution for each access control issue consisting of a combination of privilege modifications, new roles and views.

The remainder of this paper is organized as follows. In Sect. 2 the related work is shown. Following this, in Sect. 3, our framework for NoSQL modernization including security aspects is defined. In Sect. 4, our approach is applied to a case study within the medical domain. Section 5 presents a discussion and limitations of the present work. Finally, Sect. 6 explains the conclusions and sketches future works.

2 Related work

Given the multiple disciplines involved in our approach, there are several areas that must be considered as part of the related work.

With respect to security issues in NoSQL databases, different works have been developed. They address the problem of the inclusion of security policies in this kind of databases (usually applied in Big Data environments). However, these approaches rarely consider to apply this kind of policies at the different modeling stages [1–4] or to include security and privacy restrictions [3,5,6]. For all this, the works currently arise on this topic are proposals that lack adequate security in terms of confidentiality, privacy and integrity (just to mention a few properties) in Big Data domains [1,3,4,7].

Other approaches have achieved a proper security in the development of information systems, but they are not focused on NoSQL databases and their own security problems. In this sense, the most relevant proposals are listed below: (i) secure TROPOS is an improvement that provides security for TROPOS methodology [8]. It is focused on software development which uses intentional goals of agents. (ii) Mokum is an object-oriented system for modeling [9]. It is based on knowledge and facilitates the definition of security and integrity constraints. (iii) UMLsec evaluates general security issues using semantics and specifies confidentiality, integrity needs and access control [10]. (iv) MDS (model-driven security) is a proposal to

apply the model-driven approach in high-level system models [11]. It adds security properties to the model and automatically generates a secure system.

Focusing on Big Data systems we can conclude that the current proposals do not consider adequately the security concept in all stages. They provide partial security solutions such as: (i) anonymization and data encryption [12], (ii) description of reputation models [13], and (iii) authentication and signature encryption [14,15]. Furthermore, many deficiencies for the implementation of security features in NoSQL databases [6] or in Hadoop ecosystems [5,16–18] have been detected.

It is important to mention that our proposal follows the standards defined for Big Data systems. We have mentioned two main approaches: (1) the BIGEU Project (The Big Data Public Private Forum Project) from the European Union [19] which tries to define a clear strategy for the successful use and exploitation of Big Data in our society, aligned with the objectives of the Horizon 2020 program; (2) the NIST (National Institute of Standards and Technology, USA) standard [16] which proposes a reference architecture for Big Data, where it identifies a component corresponding to the Big Data application.

With respect to the BIGEU Project, our stages can be aligned with the ones used in the Big Data Value Chain. The NoSQL database analysis corresponds to data acquisition and analysis (making special emphasis in the use of ontologies and NLP techniques). Our modeling stage (conceptual model creation and security issues list) can be matched to the data curation and storage stages. Finally, the automatic generation of the solution is related to the data usage stage. Regarding the NIST architecture, different stages of the information value chain (collection, preparation/curation, analytics and access) are defined in the Big Data component (similar to the previously mentioned stages of BIGEU). The NIST big data reference architecture has been extended to integrate security issues [20]. Our approach is also aligned with this security reference architecture for big data, through components such as the security requirement, security metadata and security solution. We can conclude that the presented architectures take into account the specific characteristics of Big Data systems, oriented and directed by the data, defining the stages of its value chain. These standardization efforts are considered in our proposal that will be aligned with the aforementioned architectures.

Furthermore, with regard to conceptual modeling and semantics, several works present the use of ontologies in conceptual modeling. Weber presents how ontological theories can be used to inform conceptual modeling research, practice, and pedagogy [21]. The general formal ontology (GFO) is a foundational ontology which integrates objects and processes and is designed for applications of diverse areas such as medical, biological, biomedical, economics, and sociology [22]. A comparison between traditional conceptual modeling and ontology-driven conceptual modeling was made in the work of Verdonck et al. [23], demonstrating that there do exist meaningful differences between adopting the two techniques because higher-quality models are obtained using the ontology-driven conceptual modeling. However, to the best of our knowledge, here we present the first use of ontologies to improve the security in NoSQL databases by detecting sensitive information.

As shown, there have been advances in several areas that make possible to provide a modernization process to incorporate security in existing NoSQL databases. However, after carrying out this review of the literature related to these topics, it is evident that our research is the first modernization approach for introducing security in NoSQL databases which automatically identifies and enables security mechanisms that were not considered during the initial implementation. Furthermore, our approach adapts to each domain by using an ontology that allows users to specify what information can be considered sensitive or highly sensitive.

3 A modernization approach for NoSQL document databases

The proposed approach (see Fig. 1) consists on two stages: (1) the process of reverse engineering to create a conceptual model of the database and the analysis of the existing NoSQL solution to detect sensitive information; (2) the identification of security issues to modernize the existing NoSQL solution. These stages will be detailed in the following subsections.

3.1 Reverse engineering

The modernization process starts with a reverse engineering of the database. The aim of the reverse engineering process is to obtain an abstracted model that allows us to reason and identify security issues that would be the target of security improvements. In order to perform this process, we require a metamodel that represents the structures in the DB. One such metamodel is the one presented across Fig. 2.

The excerpt of the metamodel shown in Fig. 2 is divided into two parts. The upper part corresponds to the database structures and contains the main elements of NoSQL document databases, whereas the lower part corresponds to the security actions (modifications) to be made in order to improve access control. Since we work with MongoDB, the database metamodel is tailored to MongoDB structures and datatypes and is defined as follows:

The first element in the metamodel is the Database element, which acts as root of the model. A database may have a name and has associated any number of Collections (Views), Roles and Users.

Each Collection has a name and establishes an id that enables the identification of each document. A Collection can have several Documents, each of which may have several fields, some of which can be required for all the documents within the collection. Each Field can be a SimpleField, storing a basic type of value, or a compose field, storing other collections within it. Finally, each field may have a constraint applied to it.

Aside from collections, a MongoDB database may store views. A View has a name and a base collection (or view) over which a pipeline of projections and operations are performed.

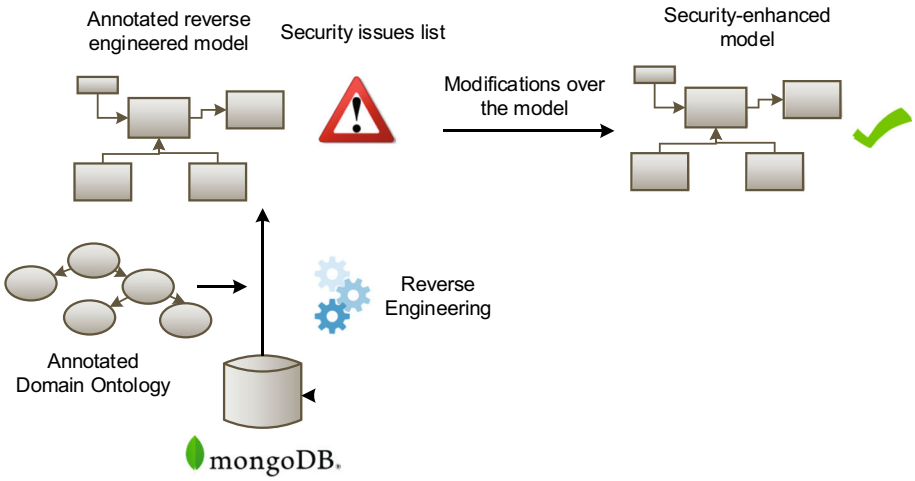

Fig. 1 Scheme of NoSQL modernization process

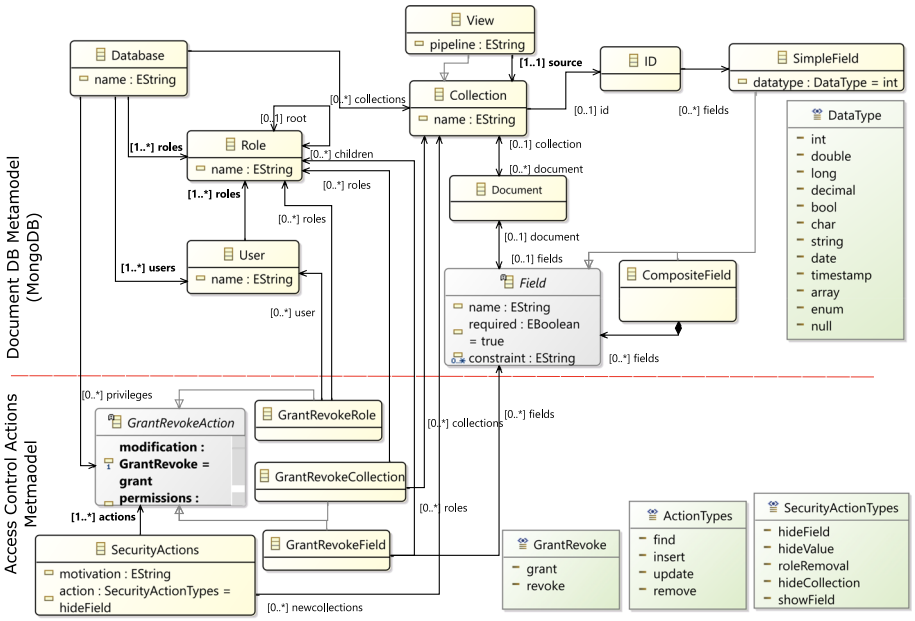

Fig. 2 Metamodel for Document-based Databases & MongoDB

Projections enable us to hide or show certain fields, and may involve conditions that determine whether an instance of the field is included or excluded. Moreover, a View can include aggregation operations to derive new dynamic fields obtained from the underlying collections.

To manage the access to the data stored within the different collections, the database may establish a Role-based system. A Role has a name and a series of privileges over a given Database and the different Collections or Views. These roles are assigned to users, with each User having at least one, and potentially multiple Roles.

In order to manage access control, privileges can be granted or revoked for each Role. Modifications over privileges in the database are covered in the lower part of our metamodel, which deals with access control. Our metamodel includes the four basic action privileges: Find, Insert, Update and Remove. These privileges can be granted over certain collections through the GrantRevokeCollection or, more in detail, over certain fields through the GrantRevokeField.

It is important to note that although at the conceptual level our model is generic and supports revoking field privileges, MongoDB does not implement field-level access control; thus, no privileges over individual fields can be directly established for a role or user.

In addition to grant and revoke actions, our model includes SecurityActions that represent high-level objectives, such as hiding fields or values, to be achieved through the creation of views, collections, grant and revoke actions.

The aim of the lower part of the metamodel is to make explicit the changes that will be performed over the database as a result of the security analysis. This also enables the addition or removal of any security actions before making effective the changes.

Using this metamodel as reference, the process for obtaining the reverse-engineered model is as follows:

1. First, a Database element is created with the name of the database that is reverse engineered.
2. Second, the list of collections is obtained by using the list_collection_names() method. For each collection, a Collection and a Document element are created and appended to the Database element. For simplicity, the Document element will hold the set of keys used in all the documents of the collection, as retrieving all the metadata of each specific document is unnecessary at this point.
3. Third, for each collection, the set of keys is retrieved and appended to the document of the collection. This can be done using multiple publicly known methods [24]. For each key, a Field element is added with its name set to the retrieved key. After all fields have been set, they are added to the collection and the next collection in the list is processed.
4. Fourth, after all collections have been retrieved, users and roles are retrieved using the *usersInfo* command. For each user a User is created along with its associated roles and privileges over collections. Once all the User and Role elements have been created, they are appended to the Database element.

After the reverse engineering process, we will have obtained a conceptual model of the database to be modernized. In order to identify potential security issues, we must perform an analysis that will start with an ontological analysis of the contents in the database. This analysis will allow domain experts to tag sensitive collections and properties that should be restricted, enabling the identification of new views, privileges, and roles that will need to be implemented or existing ones that should be adjusted.

3.1.1 Data analysis: the ontology

At this stage we will work with the source database in order to detect sensitive data. It is important to emphasize that the analysis stage can be applied to any NoSQL database technology used both in the previous phase of reverse engineering and in the next phase of identification of security issues.

Each field of the database is analyzed searching for sensitive information. One of the contributions of this proposal is the establishment of the security privileges needed to access each field of the data set. In order to define the different security privileges, we have followed the four levels of security clearance defined by U.S. Department of State[1]: unclassified, confidential, secret, and top secret, although the classifications used in other countries (Canada, UK, etc.) could be used. We have defined a mapping between the selected security clearance levels that we have called security levels (SL), and numerical values to facilitate subsequent calculations. Therefore, we will use SL=0 (unclassified, all people have access to the information); SL=1 (confidential, the persons registered in the system can access this information); SL=2,3 (secret and top secret, only certain people with specific profiles and characteristics can access this information).

In order to tag the different SL of the database fields to allow their access, we have used NLP techniques and lexical and ontological resources. In our approach, we have used the lexical database WordNet 3.1[2] which contains semantic relations (synonyms, hyponyms, meronyms, etc.) between words in more than 200 languages. In addition, it was necessary

[1] https://www.state.gov/security-clearances (visited on April, 2021).

[2] http://wordnetweb.princeton.edu/perl/webwn (visited on December, 2019).

the help of experts in the specific domain of the source database (in our case study, medical domain) and in the domain of data protection.[3]

The labeling process consists of two steps. In the first step, the lexical resource is enriched with information related to sensitivity. Therefore, WordNet was enriched by adding the SL to all the concepts; initially all WordNet concepts are labeled with SL = 1. Next, the specific domain expert (a physician, in our case scenario) will update the WordNet concepts related to his/her domain, distinguishing between SL = 2 (concepts that have sensitive information such as those related to the treatment or diagnosis of the patient) and SL = 3 (they have very sensitive information, for example, the concepts related to the medical specialty Oncology). Finally, the data protection expert will carry out the same process distinguishing between SL = 2 (for example, the patient's address) and SL = 3 (such as race or religious beliefs among others).

In the second step, each database field is labeled with a specific SL. Thus, all the field values are consulted in WordNet. Basically, two cases can occur distinguishing two types of restrictions: (1) all values have the same level of security (e.g., SL = 3) and, consequently, this SL is assigned to the field (security constraints); (2) the field values have different SL (e.g., SL = 2 and SL = 3) and, consequently, distinctions within the same field will be made (fine-grain security constraints). The restrictions are explained below:

1. Security constraints. They are defined at the field level. Security constraints are specified when the information contained in a field (after the processing of all instances) has the same level of security; that is, there is no information within the field that is more sensible than another one. For example, a race field is sensible. The information which contains is always sensible regardless of the values of the field. Therefore, a specific security level of SL = 3, 'top secret', might be required for queries.
2. Fine-grain security constraints. They are specified at the field content level. These constraints are described to define different security privileges of a field depending on its content. For example, to query a field which represents the medical specialty of patients might require a generic SL = 2. However, patients with terminal diseases might require a higher SL (for example, SL = 3). Thus, a user with SL = 3 could see all the patients (including those with terminal diseases), whereas the users with lower SL only could see patients with non-terminal diseases.

A more detailed description of the entire process will be explained in Sect. 4.3 with our case study.

3.2 Identification of security issues: security improvements

Once we have performed the data analysis using the expert tagging and the ontology, we will have a set of fields (properties in the MongoDB metamodel) tagged with different security levels. In order to identify potential security issues automatically, we will proceed as follows:

First, for each collection C, a security level array S_c will be defined, including a triple *(attribute, security level, condition)* for each attribute in the collection. The security values will be defined according to the security levels specified in the previous step, such that:

$$S_C = (\langle a_1, s_1, c_1\rangle, \langle a_2, s_2, c_2\rangle \ldots \langle a_n, s_n, c_n\rangle) \quad (1)$$

[3] This person will be the responsible for processing the data of the organization or company (according to the General Data Protection Regulation, European Union, [25,26] is the person who decides the purpose and the way in which the organization's data are processed).

In this sense, if a field presents multiple security levels, it will be repeated, and the associated condition stored for later use. If no condition is specified, c_i will be null. For each array, the maximum and minimum value will be calculated. If they are equal, the array will be compressed into a single number that represents the security level of the entire collection. An example of (1) would be $Patient$, where each value represents the security level s_i of an attribute a_i (the names have been omitted for simplicity):

$$S_{Patient} = \begin{bmatrix} 1 & 2 & 1 & 1 & 2 & 3 & 2 \end{bmatrix}$$

Afterward, we will obtain the role access matrix RA of the system, composed by role access arrays. A role array i, contains the name of the role r_i and the associated set of permissions A_i to access each collection. There is a role array for each role in the system except for the admin role, which forcibly has access to everything. Therefore, the role access matrix is defined as:

$$RA = (\langle r_1, A_1 \rangle, \langle r_2, A_2, \rangle \ldots \langle r_n, A_n \rangle), Admin \notin R \tag{2}$$

where A_i is a list of pairs $\langle c, p \rangle$, denoting that the role has access to the collection named c if $p = 1$ or that his access is restricted if $p = 0$. The role access matrix (2), such as the one shown in the following for the medical database, will be used in conjunction with the security level arrays to determine the security level of each user and role.

$$RA_{MedicalDB} = \begin{bmatrix} & Patient & Admission \\ Role\_1 & 1 & 1 \\ Role\_2 & 1 & 0 \\ Role\_3 & 0 & 1 \\ \ldots & \ldots & \ldots \end{bmatrix}$$

The user access matrix UA is obtained in a similar fashion, using the permissions of users in the systems instead of the roles.

Once we have the security arrays and the access matrices, we obtain the extended access matrix for roles REA and users UEA by multiplying the access level of each role/user with the security level array of each collection. Essentially each row of the REA matrix will be:

$$REA_i = (\langle r_i, A_1 \rangle \times S_{c1}, \ldots, \langle r_i, A_n \rangle \times S_{cn}) \tag{3}$$

An example of the REA matrix for the medical database is as follows:

$$REA_{MedicalDB} = \begin{bmatrix} & type & time & medical & medical2 & name & \ldots \\ Role\_1 & 2 & 1 & 2 & 3 & 1 & \ldots \\ Role\_2 & 0 & 0 & 0 & 0 & 1 & \ldots \\ Role\_3 & 2 & 1 & 2 & 3 & 0 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & & \end{bmatrix}$$

The extended access matrices are used as input for the analysis. The security issues identified during the process will depend on the choice of the user across three different security policies:

1. Exclusive access to highly sensitive data per user (only one user can access to each highly sensitive field/collection)
2. (Default) Exclusive access to highly sensitive data per role
3. Without exclusive access

Taking into account the selected policy, we will analyze the accessibility of each field as follows:

1. For each set of rows that are equal to each other within the role matrix, we will identify these roles as "Duplicate Roles", suggesting the removal of duplicates in the security issues list. The users that had their roles removed will be assigned the remaining role. Accordingly, the security action that summarizes this process, role removal, will include eliminating duplicate roles as well as corresponding grant role actions.
2. For security level 1 collections and fields, we will identify any 0's in the column of the matrix. Since these fields are considered as "generally accessible", we will report the list of roles without access for review, suggesting that access is given to them in the security issues list. The security action will be tagged as show field, with the corresponding grant collection actions.
3. For security level 2 collections and fields, we will identify any columns without 0's. For these collections and fields, a warning will be included in the security issues list. In order to deal with this security issue we will proceed depending on whether the entire collection has a security level higher than 1 or if higher security is set only for selected fields:
   (a) If the entire collection has a security level higher than 1, then the suggested modification will be the removal of all the access rights and the creation of a new role with access to all level 1 collections as well as the restricted collection. The associated security action, hide collection, will include the revoke access action to all roles and will grant access to the new role to be created.
   (b) If only selected fields are affected, then the suggested modification will be to create a new View that contains level 1 fields. All existing roles with access to the collection will have their access removed and will be granted access to the view instead. Finally, a new role will be created with access to all level 1 collections as well as the restricted collection. This actions will be aggregated into the hide field security action.
4. For security level 3 collections and fields, first, we will identify any columns with two or more non-zero values. Since these are highly sensitive data, it is expected that only the admin and a specific role have access to them. Therefore, a warning will be included in the security issues list. If the exclusive role access policy has been selected, then the removal of all access rights from existing roles will be suggested, creating a new one with exclusive access as in the previous step. These actions will be related to a hide field or hide collection security action. Second, we will identify any rows with two or more level 3 field access. If these fields pertain to different collections, it would mean that a single role has access to sensitive information from multiple collections. Therefore, in order to improve security, a warning will be included in the security list and the role will keep its access only to the first collection. A new role for each collection with highly sensitive information that cannot be accessed by any other roles will be added. These actions will be related to a hide field security action.
5. In the case of fine-grained constraints, (i.e., fields that contain multiple security levels depending on their contents) such as the "medical_specialty field", a new View will be created for the lower security level using the $redact operator from MongoDB and the condition previously stored. In this way, the original collection will retain all data, including highly sensitive information, while the other two views allow access to generally available and sensitive information, respectively. In these cases an additional role will be created that has access to all level 1 collections as well as the restricted view but not to the original collection. All these actions will be related to the hide value security action.

For users in the database, steps 2–4 will be repeated, identifying users without access to general collections as well as sensitive data to which all users have access. In addition, if the exclusive user access policy has been selected, step 5 will be repeated using the extended user access matrix, suggesting the removal of the rights of all users with access to highly sensitive data so that the database administrator can choose which users should maintain the access.

4 Case study

In order to prove the validity of our proposal, we have applied it to a case study within the medical domain. In the following subsections, we present: the source data, the reverse engineering process to extract the original model, the process to identify sensitive data in order to obtain the security recommendations, and finally, the generation of the security-enhanced model.

4.1 Source data

With respect to the database selection, we have used the structured data extracted from patients with diabetes used in the study developed by Strack et al. [27], extracted from the Health Facts database (Cerner Corporation, Kansas City, MO), a national data warehouse that collects comprehensive clinical (electronic medical) records across hospitals throughout the USA. It contains personal data of the patients and all the information related to their admission in the hospital.

The Health Facts data we used were an extract representing 10 years (1999-2008) of clinical care including 130 hospitals and integrated delivery networks throughout the United States. The database consists of 41 tables in a fact-dimension schema and a total of 117 features. The database includes 74,036,643 unique encounters (visits) that correspond to 17,880,231 unique patients and 2,889,571 providers.

The data set was created in two steps. First, encounters of interest were extracted from the database with 55 attributes. Second, preliminary analyses and preprocessing of the data were performed resulting in only these features (attributes) and encounters that could be used in further analyses being retained, in other words, features that contain sufficient information. The full list of the features and their description is provided in [27]. This data set is available as Supplementary Material available online,[4] and it is also in the UCI Machine Learning Repository.

Finally, the information from the mentioned database for "diabetic" encounters was extracted. In this way, 101,766 encounters were identified related to diabetic patients. These data were used in our experiments.

4.2 Reverse engineering

Using the information of diabetic patients, we created an initial MongoDB database replicating the structure of the dataset. The database contains two collections. The first one, "Admission", stores all the information regarding the admission of patients. This information includes sensitive information such as drugs that have been administrated to the patient

[4] http://dx.doi.org/10.1155/2014/781670 (visited on December, 2019).

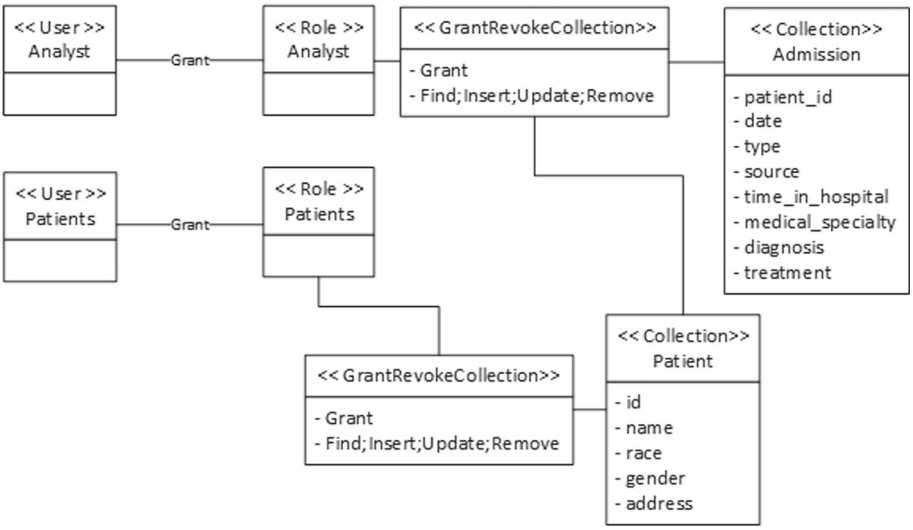

Fig. 3 Initial database model

or the medical specialty corresponding to their case. The second one, "Patient", stores information regarding patients such as name, gender, race, age or address.

Together with the "Admission" and "Patient" collections, two users with their corresponding roles are created in the database: first, the "Analyst" user and role, with access to every collection and field in order to manage the database as would be expected in a Data Warehouse-style database; second, a user and role for querying information about patients "Patients" that has access to the 'Patient" collection only.

Using the reverse engineering process presented in Sect. 3.1, we obtain the model shown in Fig. 3.

As can be seen in Fig. 3, security-wise this would be a poor database model for general use. There is little security beyond one user having access to only one collection, and there is no discrimination on whether fields are sensitive or not. Therefore, to show how our process modernizes database security, our next step will be to be analyze the data at hand in order to annotate the model, identify security issues, and generate security recommendations.

4.3 Security recommendations extraction

In our example, considering our data model, we analyzed the different fields in order to establish the security constraints.

We applied the process defined in Sect. 3.1.1. The first step consists on the enrichment of the lexical resource adding the SL to the concepts. As we have previously mentioned we have used WordNet 3.1. Initially, the lowest security level (SL = 1) was assigned to all the WordNet terms. These initial values will be modified by an expert according to the specific domain we were working on.

In our case scenario an expert in medical domain (a physician) will update the concept security levels by distinguishing sensitive information (SL = 2) and very sensitive information (SL = 3). For instance, the concepts related to the patient's treatment and the respective medicaments will be treated as sensitive information (SL = 2). On the other hand, the concepts

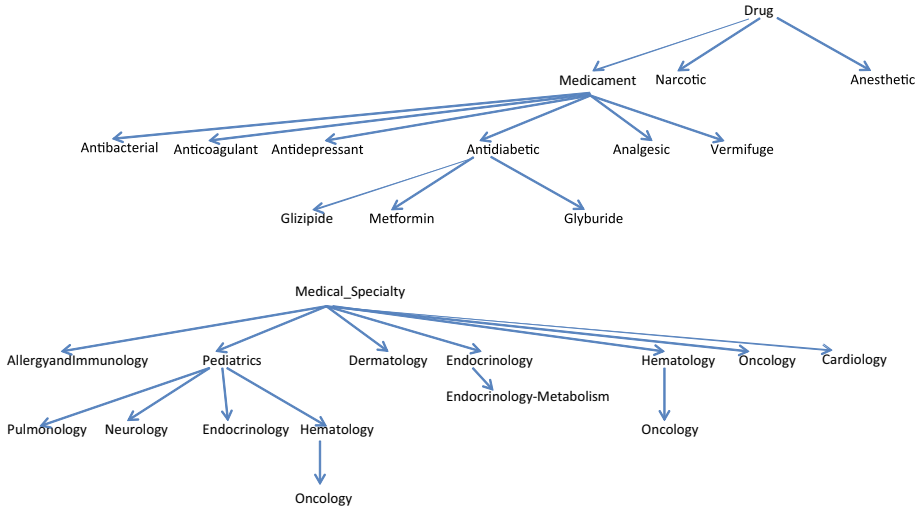

Fig. 4 A fragment of treatment and medical specialty ontologies

related to the patient's medical specialty will have sensitive information (SL = 2). However, the physician will distinguish the most sensitive specialties (Oncology, etc.) assigning them the highest security level (SL = 3). These new security levels established by the expert will be updated in WordNet in the following way: the concepts with security levels 2 or 3 are searched in WordNet. When they are found, the new security level is modified and it is propagated to all their children concepts in the tree structure. This process ends when a leaf is reached or concepts with security levels greater than the new one are found (this indicates that the concept security level has been previously modified in the sub-tree). If a concept is not found in WordNet, it will be enriched with the new concept (the expert will indicate where it will be inserted).

In Fig. 4, two fragments of WordNet related to the fields treatment and medical specialty are shown. On the one hand, the expert will assign the SL = 2 to the concept *Medicament* which will be propagated to their children (Antibacterial, Anticoagulant, Antidiabetic, Metformin, Glyburide, etc.). On the other hand, he/she will assign the SL = 2 to the concept *Medical_Specialty* which will be propagated to their children (Pediatrics, Dermatology, Hematology, Cardiology, etc.), except the concept Oncology that will have the SL = 3.

A similar process will be carried out by the data protection expert to update the SL in all the concepts of his/her domain with sensitive information. For example, the expert will assign the SL = 3 to the concept *Race* which will be propagated to the concepts *Asian*, *Caucasian*, *Hispanic*, etc.

In order to check that the labeling of the SL carried out by the experts is valid, the Cohen's kappa coefficient has been calculated to measure the agreement between them [28]. Thus, two experts in the medical domain and two experts in data protection carried out the labeling of the SL of the concepts of their respective domains. The results obtained of kappa coefficient for each domain were higher than 0.75, which are considered excellent.

In the second step, each database field was labeled with a specific SL. Next, we will introduce some examples of the two kinds of constraints that we have previously defined. Suppose we are analyzing the treatment field. In advance, we ignore the details of the information stored in this field but its values are already known. After having consulted each one of the

```
> db.wordnet.find({id:"99999999"}).pretty()
{
        "_id" : ObjectId("5ad8436ca270769f86b3f0e7"),
        "id" : "99999999",
        "nombre" : "Example",
        "definicion" : "Definition",
        "level" : "1"
}
```

Fig. 5 Example of enrichment of WordNet

values of the treatment field in WordNet, they are all assigned SL = 2. An example of the use of security constraints would be, given the previous preconditions, setting SL = 2 for the mentioned field.

On the other hand, to illustrate an example of a fine-grain constraint, suppose the field medical specialty is now analyzed. This field describes, with 84 values, the specialty of the patient: "dermatology", "endocrinology", "pulmonology", "oncology", etc. These values are searched for in WordNet having, in most instances, SL = 2. However, it can be seen that some patients have the value of "oncology" which is the most sensible (SL = 3). According this, a new recommendation will be created for the following stage of modeling.

The final result of the analysis, after applying the constraints set, was:

Patient collection Race: SL = 3; Address: SL = 2; Remaining fields: SL = 1.

Admission collection Treatment: SL = 2; Medical specialty: SL = 2 (in case of oncology: SL = 3); Diagnosis: SL = 2; Remaining fields: SL = 1.

It can be highlighted that the application of NLP techniques is very useful in database fields where natural language is the way to explain concepts or ideas. For example, in the medical domain, fields containing information about the encounter with the patient, the diagnosis, or the medication are very usual. After carrying out an analysis of these textual fields, both the lexical-morphological (POS tagging) and partial syntactic (partial parsing), we can identify the main concepts of the text. The application of these NLP techniques also contributes to dealing effectively with natural language issues, such as ambiguities, ellipses or anaphoric expressions. Once these key concepts are extracted, the process defined above is carried out by assigning a security level to a text field.

Furthermore, it is interesting to note that a similar problem is tackled by the responsible for data protection and privacy of an organization or company about document anonymization or text sanitization. In these cases, entity recognition techniques (NER, Named Entity Recognition) from NLP are used to identify sensitive entities or words in order to anonymize them by generalizing to broader concepts.

With regard to the implementation, WordNet has been converted into JSON format that is compatible with our database engine. Each of the terms of the fields is searched in WordNet to assign them a level of security. If a concept does not exist in WordNet, it can be included as shown in Fig. 5.

In our example, we have focused on the treatment and medical specialty fields to show the aforementioned constraints. An automatic process is carried out to establish the field security level. In Fig. 6, the functions to extract the security levels of the Admission collection are shown.

The result of this stage is a list of security recommendations to be taken into account in the following stage of modeling. For instance, we have obtained these two recommendations related to the mentioned fields: (1) the treatment field has the SL = 2 (security constraint); and (2) the medical_specialty field has the SL = 2, and if it is Oncology: SL = 3 (fine-grain security constraint).

```
db.system.js.save(
 {
 _id:"getLevelAdmission",
 value: function(){
 var n=db.Admission.find().toArray();
 setLevelAdmission(n[0].patient_id);
 setLevelAdmission(n[0].date);
 setLevelAdmission(n[0].type);
 setLevelAdmission(n[0].source);
 setLevelAdmission(n[0].time_in_hospital);
 setLevelAdmission(n[0].medical_specialty);
 setLevelAdmission(n[0].treatment[0].medicament);
 return ("Done")
 }
 }
 );

db.system.js.save(
 {
 _id:"setLevelAdmission",
 value: function(value, position){
 var n=value;
 var code = db.wordnet.find({"nombre":n}).toArray();
 var idp;
 code.forEach(function(i){idp=i.level;});
 return (db.admissionLevelSecurity.insert({"level":idp}));
 }
 }
 );
```

Fig. 6 Extraction of security levels of Admission collection

4.4 Security-enhanced model

Using the information from the previous step, we analyze the access levels using the steps described in Sect. 3.2.

In our case at hand, we have two roles not including the admin. With the following access levels:

$$RA_{MedicalDB} = \begin{bmatrix} & Patient & Admission \\ Analyst & 1 & 1 \\ Patients & 1 & 0 \end{bmatrix}$$

Combining this information with the security levels identified in the previous step, we obtain the extended access matrix. For the sake of brevity, all access levels 0 ("Patients" role does not have access to the "Admission" collection) and 1 (general access for registered users) are omitted in the paper:

$$REA_{Admission} = \begin{bmatrix} type & specialty & specialty\_onc & diagnosis & treatment \\ 2 & 2 & 3 & 2 & 2 \end{bmatrix}$$

$$REA_{Patient} = \begin{bmatrix} & race & address \\ Analyst & 3 & 2 \\ Patients & 3 & 2 \end{bmatrix}$$

Using the extended access matrix for roles, our approach identifies the following issues and recommendations:

- The "address" field in "Patient" can be accessed by all roles (no 0s in the column). The suggested modification is to create a new view "ViewSL1_Patient" removing all security level 2 and 3 fields. The "Analyst" and "Patients" roles have their access to the collection "Patient" removed and are granted instead access to "ViewSL1_Patient". A new role, "Patients_SL3" is created with access to the original collection.
- No further SL = 2 issues are detected. The access matrix is updated.
- The "race" field in "Patient" could initially be accessed by all roles (no 0s in the column). With the modifications made no further changes are needed to deal with this issue.
- The "Analyst" role could initially access highly sensitive information (multiple security level 3 fields) from different collections. With the modifications made no further changes are needed to deal with this issue.
- The "medical specialization" field in "Admission" has two security levels (2 and 3), yet only one user role exists "Analyst", which has access to all the information. The suggested modification is to create a new view "ViewSL2_Admission" using the $redact operator to remove the information related to oncology patients. A new role "Admission_Oncology" is proposed, which has access to the original collection. "Analyst" role has its access revoked and instead is granted access to the new view. All these operations are related to a hide value security action. The access matrix is updated.

The resulting model summarizing the new structure of the database and the actions to be taken is shown in Fig. 7, where the new roles and views are highlighted in gray color. Additionally, since all the new elements and modifications are related to their corresponding security action, it is easy to locate and remove undesired changes by removing the corresponding security action.

As a result of the analysis, our approach suggests the creation of two new roles, "Patients_SL3" and "Admission_Oncology", that have exclusive access to highly sensitive information. Existing roles (and therefore users) have their access revoked and can be granted access again by assigning them the new roles. In this way, the database now has a security hierarchy that ensures data are adequately protected without duplicity of roles.

5 Discussion and limitations

Our proposed approach allows users to be aware and introduce security mechanisms into existing NoSQL document databases. Our approach provides users with a clear view of the main components in their document database, not only warning users about potential security flaws, but also providing the mechanisms to tackle them. Still, there are some limitations that must be taken into account when applying the proposal.

First and foremost, the specific implementation of the reverse engineering process is dependent on the MongoDB API. While MongoDB is the most popular document database available, the reverse engineering process (i) depends on the evolution of the API and (ii) would need to be adapted for other NoSQL document databases. Nevertheless, the constructs used in our proposal (Collections, Fields, Roles, etc.) are generic, and the rest of the process including the ontological and security analysis can be applied to any document-oriented database since they are independent of the specific technology used. As such, our proposal would be applicable to other popular NoSQL document databases such as Apache CouchDB or Amazon DynamoDB by updating the API calls used during the process.

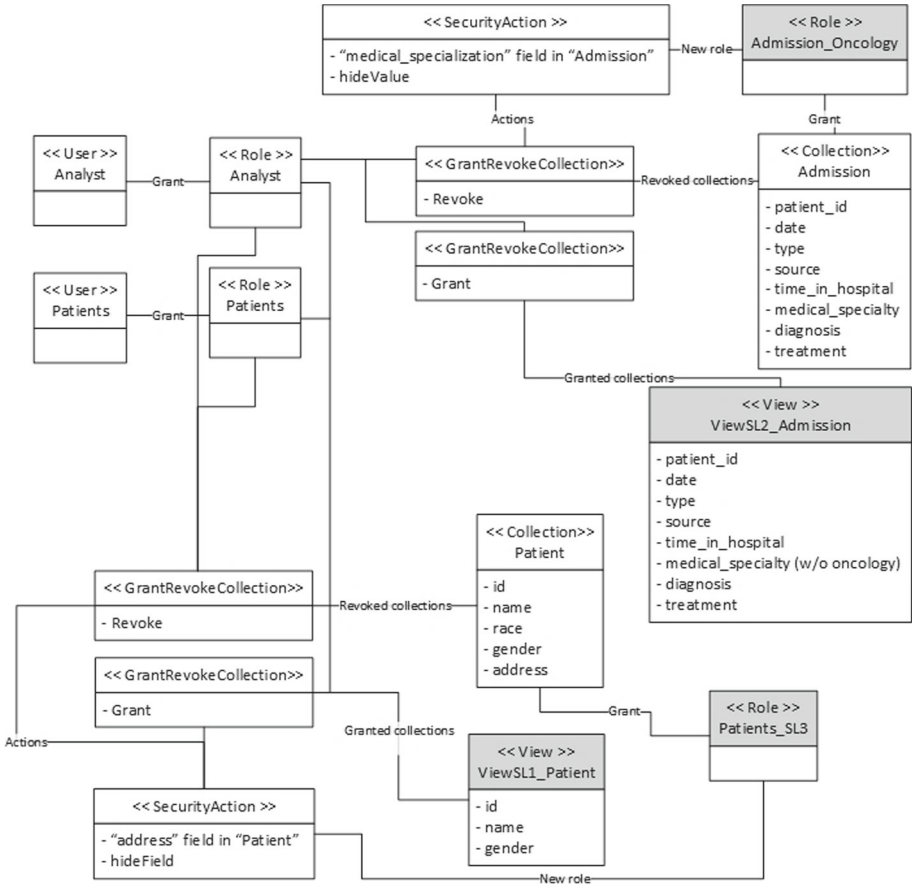

Fig. 7 Security enhanced database model

The case for other NoSQL technologies (key-value, columnar graph, etc.) is different however. As database structures differ more and more, more profound changes are required, not only in the analysis process but also in the structural part of the metamodel. As such, it is expected that the process requires certain effort to be adapted for example to columnar NoSQL databases, where the way that information is stored maintains certain similarities but structures are different. More radical changes would be needed in the case of graph NoSQL databases, where not only the structure is completely different but also the emphasis is put on the relationships. Therefore, in these cases the process would need to be entirely redone from scratch.

Second, the current reverse-engineering process does not obtain an exact model of the database. Most notably, it does not differentiate on its own between Collections and Views in the database to be modernized. This is due to limitations in the current version of the MongoDB API. Nevertheless, the analysis process is the same, since both elements need to be checked for security issues. Furthermore, the modernization itself involves modifications that do not remove or alter existing views and collections, only creates new ones and alters permissions that users have over those that already exist.

Third, the process could be optimized by carrying out the ontological analysis at the same time that the reverse engineering process is performed, thereby increasing the performance by reducing the number of reads over the database. However, this would imply coupling both processes and making the ontological analysis dependent on the specific database technology used. As such, we have preferred maintaining decoupled both steps in the process, making it easier to adapt the process to other technologies, including non document-oriented database technologies.

Fourth, the proposed process focuses on security issues related to access control. Thus, other security issues such as vulnerability to attacks, weak user passwords, etc., are considered out of the scope of the proposal. Therefore, these issues would need to be tackled by existing approaches that model attack scenarios and test the security of user accounts.

6 Conclusions

In this paper, we have proposed the first modernization approach for introducing security in NoSQL document databases improving the security of their association information systems and applications. It identifies and enables security mechanisms that had been overlooked or not even been considered during the initial implementation. Our approach adapts to each domain by using an ontology that allows users to specify what information can be considered sensitive or highly sensitive. Then, it automatically analyzes the data and the database structure to identify security issues and propose security mechanisms that enable fine-grained access control, even when by default this level of security is not supported by the existing technology. As such, our approach can be adapted to any domain and reduces the effort and knowledge required to introduce security in NoSQL document databases.

As part of our future work, we plan to cover the entire cycle, automatically deriving the code that is required to modify the database. Furthermore, we plan to expand our approach to other NoSQL technologies, such as columnar and graph-based databases.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was supported in part by the Spanish Ministry of Science, Innovation and Universities through the Project ECLIPSE under Grants RTI2018-094283-BC31 and RTI2018-094283- B-C32. Furthermore, it has been funded by the AETHER-UA (PID2020-112540RB-C43) Project from the Spanish Ministry of Science and Innovation.

References

1. Michael K, Miller KW (2013) Big data: new opportunities and new challenges [guest editors' introduction]. Computer 46:22–24
2. Kshetri N (2014) Big data' s impact on privacy, security and consumer welfare. Telecommun Policy 38:1134–1145
3. Thuraisingham B. Big data security and privacy. In: Proceedings of the 5th ACM conference on data and application security and privacy, pp 279–280

4. Toshniwal R, Dastidar KG, Nath A (2015) Big data security issues and challenges. Int J Innov Res Adv Eng 2:15–20
5. Saraladevi B, Pazhaniraja N, Paul PV, Basha MS, Dhavachelvan P (2015) Big data and hadoop—a study in security perspective. Procedia Comput Sci 50:596–601
6. Okman L, Gal-Oz N, Gonen Y, Gudes E, Abramov J (2011) Security issues in nosql databases. In: Proceedings of the 10th IEEE international conference on trust, security and privacy in computing and communications. IEEE, pp 541–547
7. RENCI/NCDS, Security and privacy in the era of big data. White paper (2014)
8. Compagna L, El Khoury P, Krausová A, Massacci F, Zannone N (2009) How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. Artif Intell Law 17:1–30
9. van de Riet RP (2008) Twenty-five years of mokum: for 25 years of data and knowledge engineering: Correctness by design in relation to mde and correct protocols in cyberspace. Data Knowl Eng 67:293–329
10. Schmidt H, Jürjens J (2011) UMLsec4UML2-adopting UMLsec to support UML2. Technical report, Technische Universitat Dortmund, Department of Computer Science
11. Basin D, Doser J, Lodderstedt T (2006) Model driven security: from uml models to access control infrastructures. ACM Trans Softw Eng Methodol 15:39–91
12. Lafuente G (2015) The big data security challenge. Netw Secur 2015:12–14
13. Yan S-R, Zheng X-L, Wang Y, Song WW, Zhang W-Y (2015) A graph-based comprehensive reputation model: Exploiting the social context of opinions to enhance trust in social commerce. Inf Sci 318:51–72
14. Wei G, Shao J, Xiang Y, Zhu P, Lu R (2015) Obtain confidentiality or/and authenticity in big data by id-based generalized signcryption. Inf Sci 318:111–122
15. Hou S, Huang X, Liu JK, Li J, Xu L (2015) Universal designated verifier transitive signatures for graph-based big data. Inf Sci 318:144–156
16. NIST, Nist big data interoperability framework: Volume 4, security and privacy, NIST Big Data Public Working Group (2017)
17. O'Malley O, Zhang K, Radia S, Marti R, Harrell C (2009) Hadoop security design. Technical report, Yahoo, Inc
18. Yuan M (2012) Study of security mechanism based on hadoop. Inf Secur Commun Privacy 6:042
19. Cavanillas JM, Curry E, Wahlster W (2016) New horizons for a data-driven economy: a roadmap for usage and exploitation of big data in Europe. Springer, Berlin
20. Moreno J, Serrano MA, Fernandez-Medina E, Fernandez EB (2018) Towards a security reference architecture for big data. In: Proceedings of the 20th international workshop on design, optimization, languages and analytical processing of Big Data (DOLAP)
21. Weber R (2003) Conceptual modelling and ontology: possibilities and pitfalls. J Database Manag 14:1–20
22. Herre H (2010) General formal ontology (gfo): A foundational ontology for conceptual modelling. In: Theory and applications of ontology: computer applications. Springer, pp 297–345
23. Verdonck M, Gailly F, Pergl R, Guizzardi G, Martins B, Pastor O (2019) Comparing traditional conceptual modeling with ontology-driven conceptual modeling: an empirical study. Inf Syst 81:92–103
24. Object Rocket (2019) Get the Name of All Keys in a MongoDB Collection
25. EU, Regulation (European Union) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation), Official Journal L 119, 04/05/2016, p 1–88, 2016
26. EU, Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications), Official Journal L 201, 31/07/2002, pp 37–47, 2002
27. Strack B, DeShazo JP, Gennings C, Olmo JL, Ventura S, Cios KJ, Clore JN (2014) Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. BioMed Res Int 2014:781670
28. Smeeton NC (1985) Early history of the kappa statistic. Biometrics 41:795

Alejandro Maté received the degree in computer science engineering and the M.Sc. degree in computer science technology from the University of Alicante, in 2010, and the Ph.D. degree, in 2013. He was abroad for over two years as part of a Postdoctoral Researcher position in Italy from 2014 to 2016 and worked for Lucentia Lab as a Business Intelligence & Big Data Architect as part of a Torres Quevedo grant from 2016 to 2017. From 2017 to 2019, he was an Assistant Professor with the Department of Software and Computing Systems. Since 2019, he has been an Associate Professor with the University of Alicante. Throughout his career, he has collaborated with several research groups across the globe, most notably including the Requirements Engineering group led by John Mylopoulos at the University of Trento, Italy, and the Software Engineering group led by Eric Yu at the University of Toronto, Canada. His research interests include BI and Analytics, ranging from the definition of strategic plans and key performance indicators to the extraction of insights by means of dashboards and algorithms. As a result, he has published over 50 articles related to BI and analytics. Most of these articles are published in high-impact international conferences (e.g., ER, CAiSE, and RE) and JCR journals (Information Systems, Future Generations Computer Systems, and Information and Software Technology). Nevertheless, his career has not been limited to the research field. In the professional department, he has developed analytic systems and software for several national and international projects. Among these projects, we can find European Research Council grants (Lucretius) and large-scope national projects from private initiatives (LPS-Bigger). The novelty of the algorithms developed granted him the Best Demonstration Award at the IBM conference CASCON in Canada. He is currently working on several projects related to eHealth and the Internet of Things (IoT), combining real-time analytics, and artificial intelligence.

Jesús Peral is an Associate Professor at the Department of Software and Computing Systems in the University of Alicante. He obtained his Ph.D. in Computer Science from the University of Alicante (2001). He has been the director of the Ph.D. Program in Computer Science (2002-2009) and the secretary (2009-2013) of the Department of Software and Computing Systems. His main research topics include Natural Language Processing, Data Integration, Information Extraction, Information Retrieval, Question Answering, data warehouses and Business Intelligence applications. He has participated in numerous national and international projects, agreements with public/private companies related to his research topics. He has been the lead investigator on one National and one Regional Research Projects. He has advised 5 Ph.D. students, and he has published more than 50 papers in Journals and Conferences related to his research interests (26 JCR papers). He has also been the Co-editor of 3 special issues in different JCR journals. Finally, he has participated in program, organizing committees and reviewer of national and international conferences and journals (more than 30 participations).

Juan Trujillo (Member, IEEE) received the Ph.D. degree, in 2001. Since 2001, he has been leading the Business Intelligence and Big Data research in the department and has also been the Founder and the Director of the Lucentia Research Group since 2008. He is currently a Full Professor with the Department of Software and Computing Systems, University of Alicante. His research interests include business intelligence applications, big data processing and analytics, data warehouses, decision support systems, and artificial intelligence. He has advised 12 Ph.D. students, and he is the author of more than 200 conference papers, many of them in ERA A conferences, such as ER, UML, and DAWAK or CAiSE, and more than 60 JCR articles, such as Data & Knowledge Engineering (DKE), Decision Support Systems (DSS), iSoft, and Information Science (IS or InfSci). He has also been the Co-editor of 11 special issues in multiple JCR journals, including Data & Knowledge Engineering (DKE), Decision Support Systems (DSS), and Computer Science & Information Technology (CS & IT). He has also been the PC-Chair in multiple international events, such as ER'18, ER'13, DOLAP'05, and DAWAK'05-'06. He was a Senior Editor of the Q1 JCR journal Decision Support Systems (DSS) until 2017. It is also noteworthy the high impact of his publications in the field of BI, which have led him to become one of the most cited authors in the area, having articles with 152, 128 or 98 citations, positioned in the 3rd and 8th rank of the list of most downloaded articles in journals, such as Data & Knowledge Engineering (DKE). One of his articles appears as the most cited in the Data & Knowledge Engineering (DKE) journal during a five year period. He is the most cited Researcher in the Technical School of Computer Science (EPS) in the UA and he is between the top 50 Researchers in Spain, considering all the disciplines within the Computer Science area, and he is within the 20 Top international researchers in his main areas, such as conceptual modeling, data warehouses or business intelligence (Font: Google Scholar).With regard to Technology Transfer, he owns eight Intellectual Property Registers (IPR) and is the Co-Founder of the Lucentia Lab S.L. Spin-off in April 2015, an EBT company participated by the University of Alicante. He has been the Principal Investigator (PI) of a high number of National and Regional, and even International Research Projects. Furthermore, he is also a Very Active International Researcher, and he is also a member of several international associations, participating in the meetings of NESSI, PLANETIC, and BDVA (Big Data Value Association) among others. It is worth noting that this international activity and networking has materialized in the participation of several H2020 European projects, such as SAMNIC, SAFERPLAY (PI: Juan C. Trujillo), E4Children (PI: Juan C. Trujillo), and Lucretius (ERC Advanced Grant), where he is also an Active Researcher. Finally, he holds the international credential Project Management Professional (PMPr) for project management awarded by the prestigious Project Management Institute (PMIr).

Carlos Blanco has a Ph.D. in Computer Science from the University of Castilla-La Mancha (Spain). He is working as a lecturer at the Science Faculty at the University of Cantabria (Spain) and is a member of several research groups: GSyA (University of Castilla-La Mancha) and ISTR (University of Cantabria). His research activity is in the field of Security for Information Systems and its specially focused on assuring Big Data, Data Warehouses and OLAP systems by using MDE approaches. He has published several international communications, papers and book chapters related with these topics (DSS, CSI, INFSOF, ComSIS, TCJ, ER, DaWaK, etc.). He is involved in the organization of several international workshop (WOSIS, WISSE, MoBiD) and has served as reviewer for international journals, conferences and workshops (INFSOF, CSI, DSS, TCJ, ARES, ER, DaWaK, SECRYPT, etc.).

Diego García-Saiz obtained this PhD. in Computer Science in 2016. Today he is a Professor at the Software Engineering Department the University of Cantabria (Spain). His research activity is focused in the Data Management and Analysis field, with several papers published in impact-factor journals, book chapters and conferences. Also, he works in the Software Engineering arena. He has participated in various national and European public research projects.

Eduardo Fernández-Medina holds a PhD. and an MSc. in Computer Science from the University of Castilla-La Mancha. He is a Full Professor at the Escuela Superior de Informática of the University of Castilla-La Mancha in Ciudad Real (Spain) (Computer Science Department, University of Castilla La Mancha, Ciudad Real, Spain), his research activity being in the field of security in information systems, and particularly in security in big data, Cloud Computing and cyber-physical systems. Fernández-Medina is co-editor of several books and chapter books on these subjects and has published several dozens of papers in national and international conferences (BPM, UML, ER, ESORICS, TRUSTBUS, etc.). He is author of more than fifty manuscripts in international journals (Decision Support Systems, Information Systems, ACM Sigmod Record, Information Software Technology, Computers & Security, Computer Standards and Interfaces, etc.). He leads the GSyA research group of the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain and belongs to various professional and research associations (ATI, AEC, AENOR, etc.).