# Towards NoSQL graph based master data management systems: building a generic and collaborative solution

Arnaud Castelltort, Cédric Fauvet, Johanna Guidoni, Anne Laurent, Michel Sala

Int. J. Emerg. Sci., 4(3), 87-102, September 2014
ISSN: 2222-4254


# Towards NoSQL Graph Based Master Data Management Systems: Building a Generic and Collaborative Solution

A. Castelltort[1], C. Fauvet[2], J. Guidoni[1], A. Laurent[1] and M. Sala[1]

[1] LIRMM – Univ. Montpellier 2 – CNRS UMR 5506,
161 rue Ada, 34 095 Montpellier, France
[2] Neo Technology Inc, South Koreya

arnaud.castelltort@lirmm.fr, Anne.Laurent@lirmm.fr

**Abstract.** Maintaining clean data in organizations is crucial and strategic. It has been indeed proven that poor quality data can lead to efficiency and/or financial damages. For this purpose, organizations are now relying on the so-called *Master Data Management systems*. These systems aim at providing organizations with tools for building robust sets of information acting as the *truth*. They are for instance used for maintaining up-to-date information about the entities, customers and products of a company. Many tools have recently been proposed for managing master data, but they do not allow to consider the three key points we address in our work, namely genericity, graph-orientation and collaborative solutions. In this paper, we present the KEBENARAN system that integrates all these three concepts. We show the feasibility of such an approach through the implementation of a prototype.

**Keywords:** Master Data Management, Collaborative Systems, NoSQL Graph Databases.

## 1 INTRODUCTION

In our word of pervasive computing, data are the cornerstone and appear in all steps of the organization workflows, from their acquisition to their use in daily life and high-level decisional processes.

Information systems have been intensively mutating in the last decades. The Web 2.0 practices have permanently affected the architecture of information systems by increasing the interoperability.

In such a situation, data quality matters more than ever. The cost of poor data quality, as estimated in many work from the literature, is huge [1, 2]. As reported in [3], *data quality best practices boosts revenue by 66%. On the opposite, the cost of bad or dirty data exceeds $600 billion for US businesses annually. Poor data across businesses and the government costs the US economy $3.1 trillion a year*.

There have been many examples where the presence of erroneous data has a significant impact. In December 2005, the *Tokyo Stock Exchange* which is the second Japanese banking group lost 286 million of euros because of a typo.

In Europe, Airbus has had difficulties with its industrialization phase because of a problem with connecting electric cables of the fuselage: the design had been carried out with different software versions for the French and the German part!

Thus, the quality of the data is at the heart of the problem of organizations. These examples have led organizations to consider at every level, from the board of directors to operational

employees to become more and more aware of the importance of data governance. However, the process of data governance is tedious.

The rest of the paper is organized as follows. Section 2 defines the problem statement and the different existing solutions proposed to solve it. It also discusses the limitations of these approaches. Sections 3 and 4 introduce our proposal for an innovative solution that addresses limitations detailed in Section 2. Section 5 concludes and presents the future work.

## 2 PROBLEM STATEMENT

### 2.1 Information System: a Silo Approach

Information systems have become a key component in organizations during the last decades. Software units have been designed for every activity. Such developments have created systems in silo that all have their own treatments and their own data. When units have to exchange data, point-to-point integrations are created that have led to the so called spaghetti pictures of information exchange depicted in Figure 1[1].

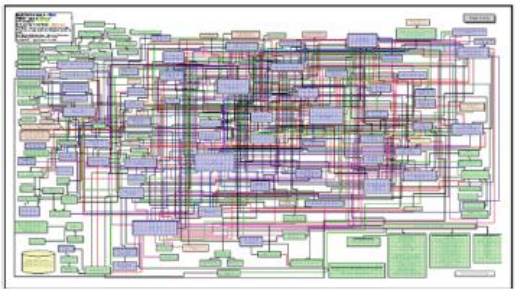

**Figure 1.** Point-to-point Information Exchanges between Software in an IS

The lack of orchestration in the information system induces a decline in the quality of data. Solving the problem of data quality starts by answering this question: How to ensure coherence of the information system?

For now, organizations use two different approaches:

- Based on a technical point of view, they use Enterprise Service Bus (ESB) technology to simplify the interactions between applications;
- Based on a governance point of view, they use Master Data Management (MDM) approaches proposed by the software industry, whose objective is to guarantee data consistency between the various components of the information system.

### 2.2 Enterprise Service Bus (ESB)

As point-to-point exchanges are difficult to maintain and lead to bad quality of data, systems have been proposed to optimize the exchanges. The Enterprise Service Bus (ESB) architecture [4] works as shown by Figure 2.

[1] blogs.informatica.com

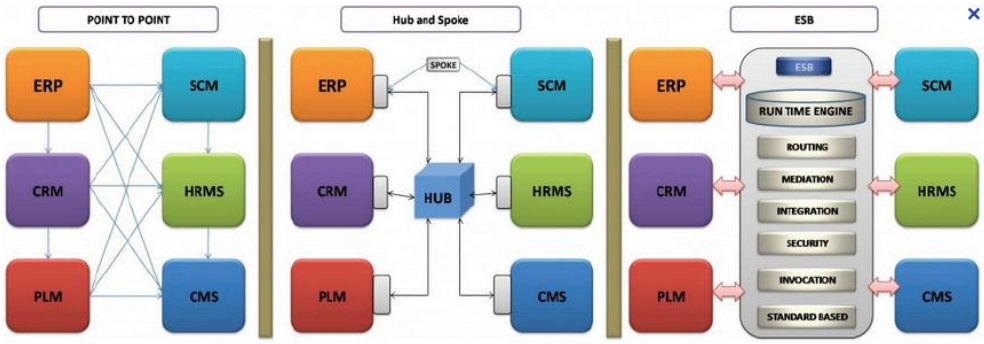


**Figure 2.** From Point-to-point to ESB.

ESB have contributed to the enhancement of data quality but are known to be insufficient to guarantee it. Organizations need to manage information about key data that is used in operational and decisional processes. For instance, the list of employees, the list of suppliers, customers, etc. Managing these key data amounts to create a *Single Source of Truth*.

### 2.3 Master Data Management (MDM)

Vendors have developed Master Data Management solutions (also known as MDM) that have been designed to tackle with this challenge [5, 7]. These solutions rely on a hierarchical vision of the data governance, as depicted in Figure 3.

They aim at providing organizations with tools for building robust sets of information acting as the truth. They are for instance used for maintaining up-to-date information about the entities, customers and products of a company. Gartner has estimated the revenue for these solutions to $1,9 billion in 2012, now 21% increase compared to 2011 and predicted to be a $3,2 billion market in 2015.

**How it Works:** MDM mainly relies on hierarchical processes. The data to maintain are shared among several data stewards who are responsible for their data elements [8]. Data quality is cut into subdomains which data stewards are responsible to maintain. This top-down approach implies that these data stewards act as relays who show a certain push towards looking at the data experts. Data stewards then rely on the other members from the organization to ensure data quality in a top-down approach, as displayed on Figure 3.

**MDM Solutions Overview:** Many solutions are proposed. Some of the main companies proposing MDM solutions are Microsoft [9], Talend [10], Oracle [11, 12], SAP [13, 14], Tibco Software, IBM [15, 16], Orchestra Networks [17], Informatica [18] and Information Builders [19]. These solutions can be compared according to some criteria that are recalled in Figure 4. In this comparison, the process orchestration stands for the capacity to manage a workflow. Data integration is related to data synchronization. Modelization is related to the possibility for users to create models, thus being linked to genericity of the solution. Security is related to the management of user roles and access restrictions.

Organizations must decide what to manage in their MDM. Most of the time, such decisions take time and require the help of specialist IT consultant. Every change in the organizations or in the choices is then difficult to maintain.

However, we point out three limitations of the existing MDM systems: non- hierarchical organization support, weak relationship handling and a high cost of personalization over time.

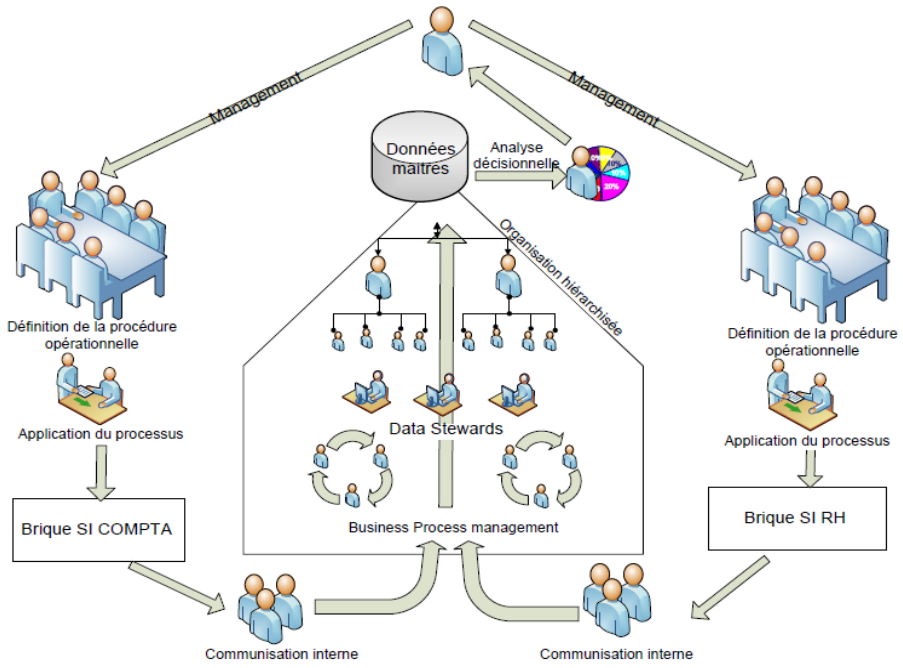


**Figure 3.** Hierarchical Organizations and Data Stewards.

| Features | Microsoft | Oracle | Orchestra Networks | SAP | IBM | Tibco Software | Semarchy | Talend | Informatica | Information Builders |
|---|---|---|---|---|---|---|---|---|---|---|
| Data Quality Management | X | X | X | X | X | X | X | X | X | X |
| Process Orchestration | X | | X | | X | X | X | X | X | X |
| Data Integration | X | X | X | | X | X | X | X | X | X |
| History Tracking | X | X | X | | X | | X | X | X | X |
| Modelization Service | X | X | | | X | | | X | X | |
| Security | | | | | X | | X | | | |
| View Creation | X | | X | | X | | X | X | X | |
| Use of ETL | X | | | X | X | | | X | | |
| Relevant GUI | X | X | X | X | X | X | | X | X | |
| Versioning syst. | X | | X | | | X | X | X | | |
| User Experience | | X | | | | | | | X | |
| Use of Cloud | | X | | | | | | | X | |
| Open Source | | | | | | | | X | | |

**Figure 4.** Comparing MDM Solutions

**Non-hierarchical Organization Support:** MDM solutions are designed for hierarchical organizations (Figure 5). In such structures, the authority of persons from high level

departments allows to designate persons responsible for controlling the process of cleaning data.

However, more and more organizations are organized in a non-hierarchical manner. This is the case for example for extended and matrix-based organizations [20] (universities, holding companies and their subsidiaries/franchisees/joint venture partners, public administrations, etc.).

In matrix-based organizations (Figure 6), a vertical structure (functional structure) combines with a horizontal structure (project structure), two divisions may overlap on the same hierarchical level [21]. Cross functional managers and managers work without direct line authority. Such organizations thus require a combination of cooperation and no conflicts between the units.

Then, non-hierarchical (e.g., network-based and matrix-based) organizations cannot implement the MDM solutions as their functioning cannot be reduced to a pyramidal vision. All actors communicate only rarely and are independent from each other which make it therefore impossible to consolidate designations of data stewards. For example, if we consider the case of a hospital facility where all actors are multidisciplinary and the authority is not partitioned. In this situation, a physician may receive orders by a committee of physician. All business actors may decide not to adhere to the implementation of a conventional MDM project.

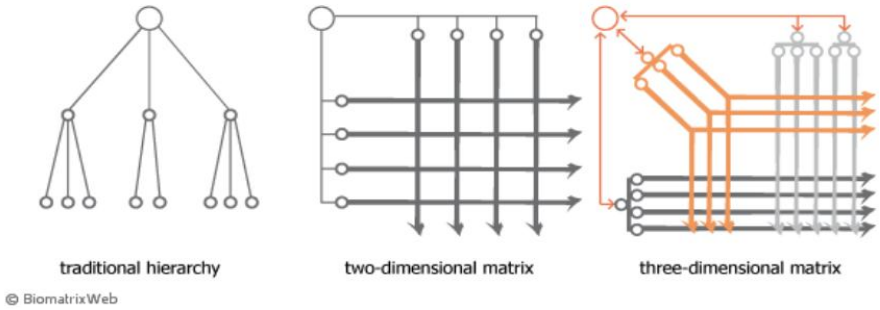


**Figure 5.** Types of Organization

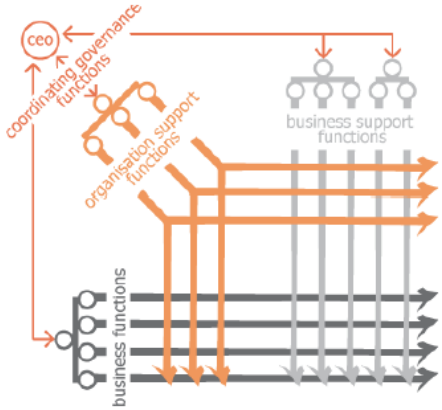


**Figure 6.** An Example of a Matrix-Based Organization

Moreover, the turnover and modification activities constantly change which amplifies the phenomenon of change.

Finally, users are less and less happy with being taken aside from such processes, as they become more and more used to collaborate for building and sharing information (Social networks, Wikipedia, OpenStreetMap, etc.).

**Weak relationship handling:** Although relations can be managed in some MDM solutions, we claim that the focus is not enough put on this type of data.

we claim that existing MDM are well-designed for managing simple information such as the lists of suppliers and customers, but do not enough focus on complex information such as the links between objects. In real applications such as maintaining information on a university, it is indeed important to record the relationships between students and the teaching units they are registered for.

**High cost of personalization over time:** MDM solutions are often designed to be deployed with the help of specialists who have to come and analyze the target organizations to set up a personalized solution.

Solutions are not flexible. They often require external consultants to be launched and maintained. In a typical project implementation of an MDM, organization uses consultants modeling all the structure and functioning of the organization and then offering a unique and rigid solution. This is costly, time consuming and difficult to apply.

Moreover, existing MDM systems require that experts parameterize them to fit the targeted organization design (e.g., administrative structure, type of hierarchical relations between employees). This process is almost impossible to be managed by the organizations themselves. Beyond the cost of such a process, it is important to highlight that any change in the organization structure requires another costly process. The cost of implementing MDM solutions in organizations is thus enormous.

Moreover, the MDM system must be personalized to _t the organization design (e.g., administrative structure, type of hierarchical relations between employees).

### 2.4 RUNNING EXAMPLE

**Example Presentation** We consider here a very simple example of an organization which will be considered in the rest of this paper.

- Several types of structures are defined: public units, research laboratories, engineering schools, private companies.
- Three types of people are defined: PhD candidates, teachers and engineers;
- Several structures are considered: the LIRMM lab, the POLYTECH school, the NeoTechnology company.
- Several types of relations are considered: collaborates, teaches, works;
- Five people are considered: Arnaud, Cédric, Anne and Julien.

It should be noted that these objects are defined at several levels. First, the objects are defined, namely structures, people and relations. Second, some types of structures, of relations and of persons are defined (*e.g., research laboratory, collaborates, teacher*) Moreover, some data are provided (*e.g.*, Arnaud, LIRMM).

This example is shown in Figure 7 where some information is forgotten for the sake of readability.

**MDM Solutions Limitation on the Example:** In this example, the aim is to define and maintain the data structures (e.g., types of structures, types of persons).

This raises some limitations that existing MDM could hardly address.

- **Non-hierarchical organization support.** Many persons have information that they could share in order to improve the system. However, persons and structures are not linked in a hierarchical manner (a research laboratory is neither superior nor inferior to the engineering school). Thus data stewards will have difficulties to work.

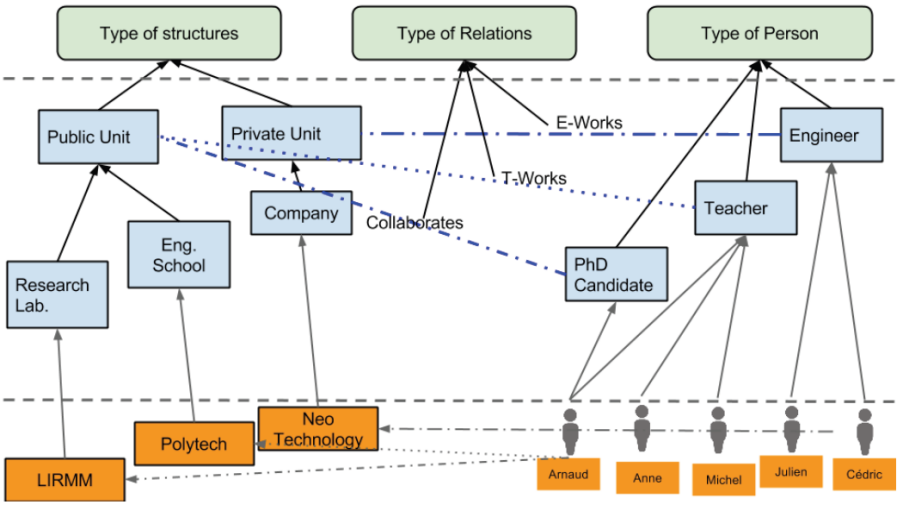


**Figure 7.** Running Example.

- **Weak relationship management.** In such a problem, even if it is important to keep the list of structures and persons up-to-date, one of the main concerns is to maintain current types of links and links between these objects. For instance, the role of one person in a research laboratory could evolve (adding a link of type IsMemberOf), the relations between Arnaud and NeoTechnology could evolve (for instance if Arnaud is hired), etc.
- **Personalization over time.** The system needs to be flexible. Genericity is important. Every definition is indeed in its own context. For instance, in another organization, no research laboratory would be to design. It also needs to be evolving, as both data and metadata can evolve. It may for instance be the case that the definition of the structures changes.

As described above, master data management is crucial. However, existing MDM solutions reach their limits as they do not address some of the specificities of data and of organizations. We thus propose a solution that addresses the three above-mentioned concerns: managing matrix-based organizations through collaborative solutions, dealing with links between objects and designing a generic solution that users can personalize and maintain.

## 3 KEBENARAN: AN INNOVATIVE MDM

We propose to build a collaborative and innovative graph based MDM system called KEBENARAN (Figure 8). KEBENARAN means “Truth” in Indonesian.

This section offers a synoptic of the KEBENARAN system.

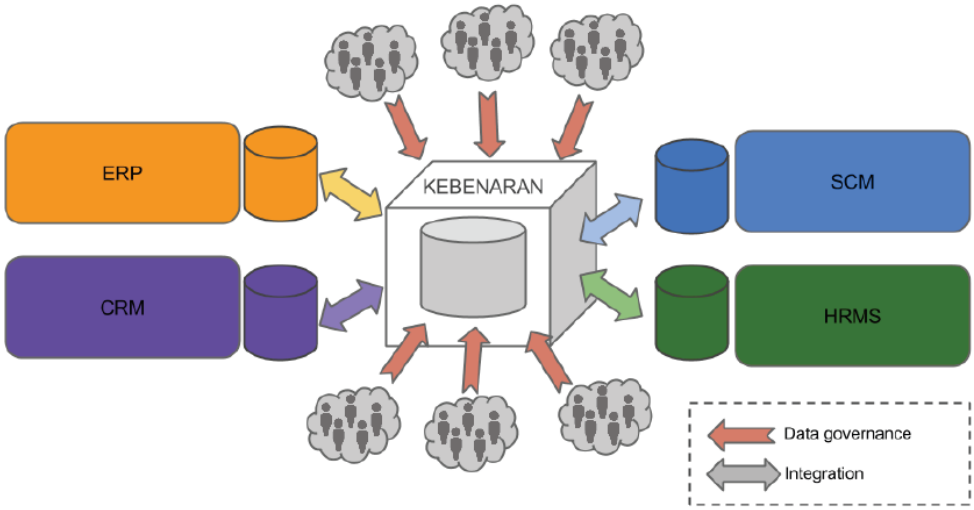


**Figure 7.** Kebenaran Illustration

### 3.1 An Innovative Approach

The approach taken in this paper is based on a logical alternative to the logic MDM, which involves valuing community behavior within complex organizations and organizing processes ensure reliability to produce skill levels appropriate to their data management issues.

The main idea to address this problem is not to define an MDM project involving leaders for decisions and for consultants or IT implementation, but rather the reverse. This means that we define from the outset of the project community management rules to delegate the implementation community of actors trades under the authority responsible.

Although MDM offer relevant solutions, they also often fail to fully meet the expectations. However, turning down to experts is not efficient as they do not always feel being as involved and motivated as they should be. Being aware of the importance of data quality and following new practices induced by the arrival of an MDM solution is not an easy task.

Crowdsourcing has proven to be more efficient (e.g., Wikipedia) for involving people, although all information cannot be shown to anybody and although all collaborators cannot act with the same reliability in sensitive domains.

In fact, Community-based mechanisms based on a set of open communities (Wikipedia, OpenStreetMap [22], etc.) which produce data quality / reliability that are often very good and up-to-date.

In addition, collaborative solutions such as social networks have led to new forms of collaboration. We thus claim that such collaborative solutions are interesting for MDM in the context of matrix-based organizations.

However, Community-based mechanisms are not yet well-established in organizations for data quality purposes.

### 3.2 Key concepts

In our work, we address the challenge of building a bottom-up approach based solution that overcomes the cost of change management and that answers the three limitations of existing MDM stated above. Our solution is meant to comply with three main features, being generic, graph-oriented, and collaborative:

1. Collaboration is proposed to address the issue of dealing with non-hierarchical organizations. One of the challenges of this high level of abstraction is to give a high complexity level to the project conception. The system must not only handle the data but also all the setting that describes the data and the organization.
2. Genericity is proposed to address the issue of personalizing and maintaining the solution. There is a great diversity in the way of achieving a process of data validation. KEBENARAN follows the crowdsourcing concept and provides an agile system able to display personalized views on every object, by computing points of views oriented. On our proposition, we introduce the concept of surrounded objects/data.
3. Graphs are proposed to address the issue of dealing with relations between objects. We claim that the model must be graph oriented, because this problem can neither be treated efficiently as a tree nor in a table-centric modelization.

### 3.3 A Generic Project

We rely on the paper from [23]. As described above, extended organizations require innovative solutions in order to deal with their particular structuration. By managing such a complex framework, our system will be able to deal with all kinds of organizations to increase the reliability. Reliability must be understood as a process requiring several key concepts that are introduced below.

The KEBENARAN project involves the establishment of benchmarks which completeness and quality must ensure and improve the level of digital services. There are four types of such master data pieces:

- Repository of persons (e.g., employees, students, customers),
- Repository of structures (e.g., departments, units),
- Repository of resources (e.g., building),
- Repository of services.

In this system, every person, every structure, every jurisdiction, every department, every resource is uniquely identified and their information are subject to continuous reliability processes.

Some other concepts may be added, as for instance skills for talent management.

It should be highlighted that these concepts are not considered as operating independently but are rather linked through relations. In particular, persons are connected to structures, resources, and services, while resources are connected to structures, etc.

Regarding persons, the connections are controlled with a role management. Several roles are defined. Depending on the role, the person has more or less possibilities to read and write the data associated with the object being considered.

Some roles have been predefined, namely the tutor, responsible, manager, member, associated, denied. The possibilities to access the information are detailed into arbitrage, validation, writing, alert, monitoring, and reading.
Several members and responsible can be defined for an object, thus leading to conflicts. The tutor is then the only person qualified to arbitrate. This empowers him/her to resolve or mediate conflicts when other persons do not agree.
As every object has one and only one tutor, the system cannot enter in deadlock situations.

For every object, communities are thus built by considering the group of all persons connected to an object. The information on an object that a user is authorized to access and the access level depend on his/her role regarding this object. For every object and for every link between objects, information is contained in the so-called surrounded objects. This information is split into information pieces ranging from highly sensitive and private information to public information.

For instance, the example from the running example can be exploited by considering the fact that Arnaud is linked to himself with the role of responsible, Arnaud is linked to the LIRMM lab. With the role of member, Arnaud is linked to Polytech with the role of associated. Anne is linked to Arnaud with the role of tutor, Julien and Michel are linked to Arnaud with the role of member of his community, and Cédric is linked to Arnaud with the role of associate in his community.

Arnaud and Anne will thus have access to all the information about Arnaud, including sensitive information. Michel and Julien will have access to less information, while Cédric will have access to very few information. As Arnaud has relations with Polytech, NeoTechnology and LIRMM, Arnaud belongs to these communities with different roles which form communities as shown on Figure 9.

Data access restrictions are then computed regarding the roles as shown on Figure 10.

To support genericity, it is necessary to build a flexible data model. Because the data is itself the schema in graph databases, it is not necessary to predict all possible data and relationship permutations ahead of time. Likewise, it is not necessary to hold all data records accountable to the same set of constraints. Instead, the database has the flexibility to accurately represent each and every piece of data as it truly is, in such a way that all of the connections between data are represented with precise fidelity.

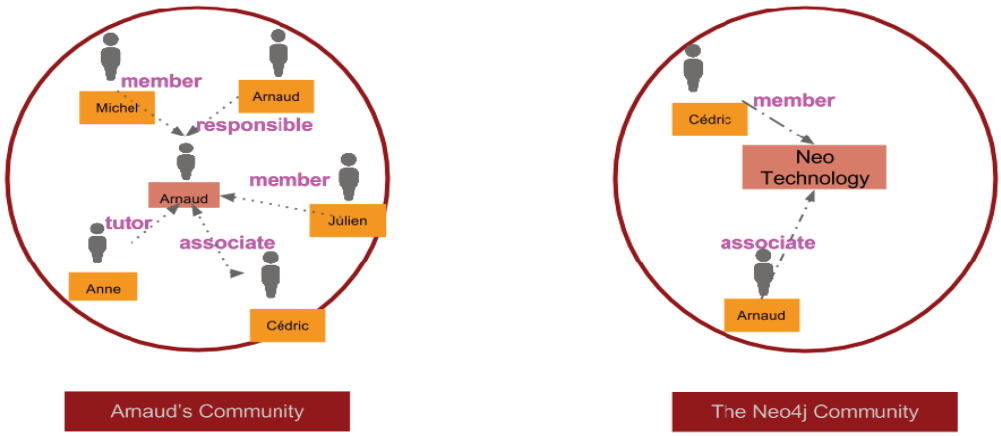


**Figure 9.** Running Example: Communities.

### 3.4 A Collaborative System

KEBENARAN proposes an innovative model hybridizing collaborative croud-sourcing solutions and classical MDM systems.

In many cases, model representation is constrained to a tree-like model because tools to manage this kind of model (e.g., LDAP (SLAPD) and Active Directory) have been available for a long time. It should be noticed that a tree is a particular graph with strict constraints applied on it.

When modeling real business processes, it is often necessary to overcome these constraints. This section introduces two illustrations of how KEBENARAN overcomes such constraints.

- Role management: complex organizations circuits and systems must be innovative as actors are heterogeneous when considering both their nature and their purposes. Actors are indeed not all belonging to the same units, and can play several roles in the organization. Depending on the roles, they can act as both supervisors of some units, members of other ones, etc, making it impossible to develop a tree-like description of the organization. This is described in [23].
- Complex access rules: one of the key concepts of KEBENARAN is the surrounded objects. Every object is provided with a personalized view built by considering both the vision taken from this object and its direct vicinity to deduce the associated access rights. The use of a graph structure and traversals build a full permissions-structure for any managed object with exclude and include overriding possibilities. This results in a dynamic construction of Access Control Lists (ACLs) based on the position and context of the surrounded object.

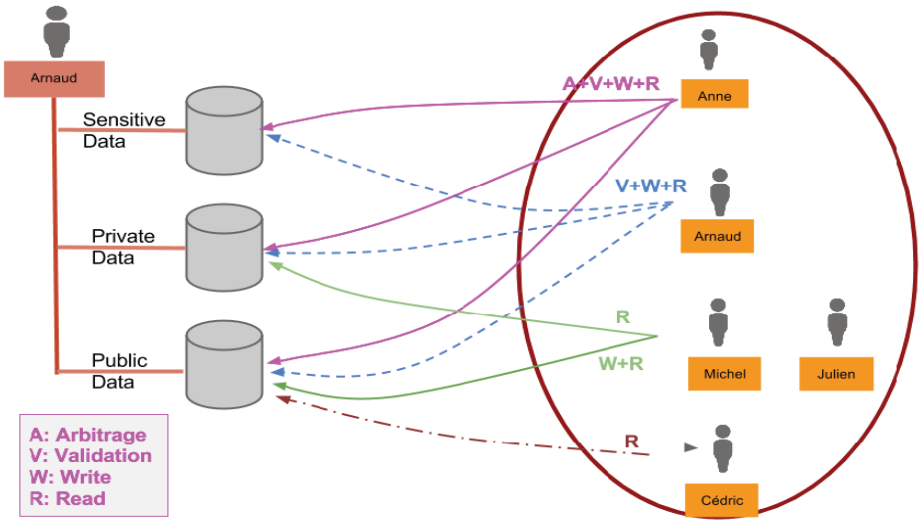


**Figure 10.** Running Example: Data Access Restrictions.

Dealing with surrounded objects allows every user to be provided with a personalized view of the data. The user's work is facilitated as he will be able to focus on his scope very efficiently, as well as objects will be efficiently provided with all their vicinity. Considering such surrounded objects also aims at reinforcing the system security and efficiency by maintaining the user in his authorized area.

All objects are provided with a community which represents all the people involved in it, at several levels from full implication (implying the right to act on the information) to low implication (implying the right of read only) and even interdiction (implying that the people is prevented from becoming a member of the community).

### 3.5 A Graph-Based System

Graph databases [24] offer the opportunity to avoid a lot of technical problems faced in MDM implementations. We explain here why we used a graph database to persist the information for the system in further details.

One of the big challenges with Master Data Management is building and maintaining an accurate model of complex hierarchical data sets. Master data such as organization and product master are inherently shaped like graphs: deep hierarchies with top-down, lateral and diagonal connections. Managing such data models with a relational database results in complex and unwieldy code that is slow to run, expensive to build, and time-consuming to maintain.

Hierarchy management in Master Data Management (MDM) and even Hierarchical Data Matching do not align completely with the real world when it is about handling complex relational data models and hierarchical approaches.

Some of the greatest technical problems faced in MDM implementations are allowing iterative data model development (largely not possible with relational solutions), and having a simple way of working with inherently complex data. A graph approach significantly reduces project complexity. Also, graphs offer:

- Semantic fit: hierarchies are easy to represent, mapped hierarchies are complex graph;
- Time variance and sandbox: need to keep versions over time (at the granularity of seconds) - approved versions co-exist with what-if data;
- Business rules: complex access rules, roles plus graphs queries; validation complexity;
- Real time: hierarchy changes to data warehouse and reports.

## 4 PROTOTYPE

We explore here our solution that tackles the above-mentioned challenges. In this section, we draw an overview of the system.

### 4.1 Architecture Overview

Figure 11 displays the architecture of KEBENARAN. KEBENARAN is placed as a middleware in the information system, operating with all components in order to achieve its goals.

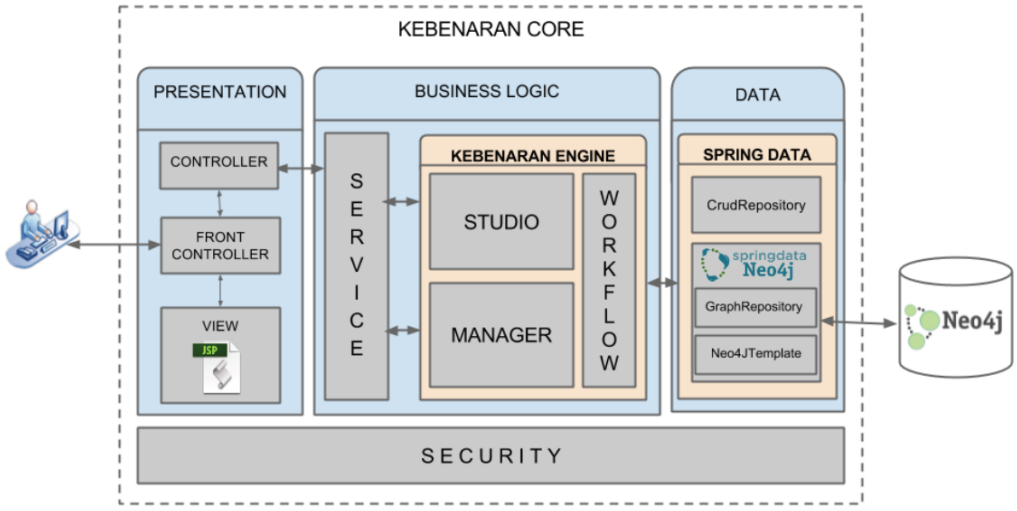


**Figure 11.** Architecture of KEBENARAN.

The internal architecture of KEBENARAN is split into three levels, so as to address its three main features as shown in Figure 12 and Figure 13.

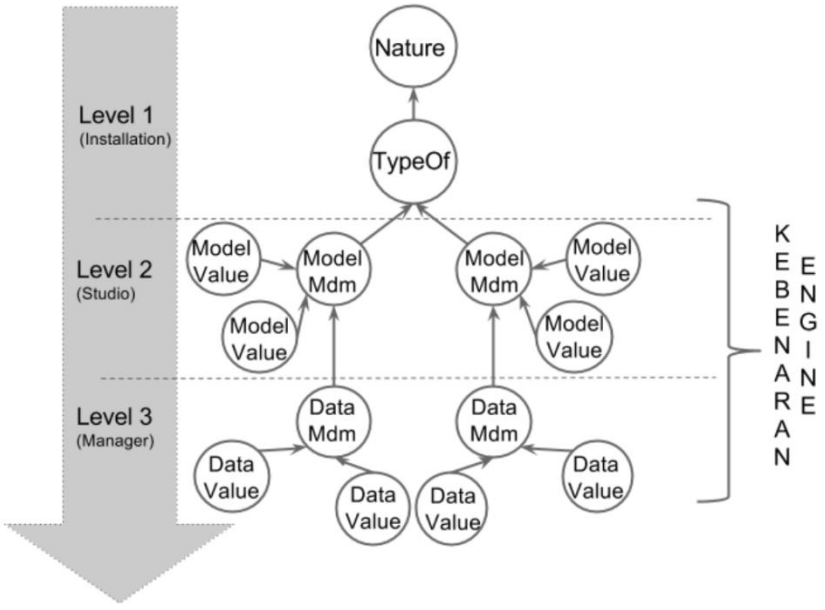


**Figure 12.** Modelization Layers of KEBENARAN

The **STUDIO** provides a high-level modelization tool, so as UML profiles can do, in order to make users able to modelize any organization. In this part, objects are the ones provided in the key concepts (persons, structures, resources).

The **MANAGER** allows to design the organization itself. For instance in the case of the UM2, it is the part where the users will define that there are laboratories, teaching units as administrative structures, research projects, bachelors and masters as functional structures, professors and students as persons, buildings and teaching hours as resources etc.

The **DATA** part contains the data themselves, for instance the LIRMM lab and the Polytech teaching unit, and the Arnaud person who is both member of LIRMM and POLYTECH.

For all these parts, the data are highly graph-oriented. The particularity is that it is that these graphs are structured in levels and acyclic.

One of the main originality of KEBENARAN is that the MANAGER part is also offered to users who want to act and make it more reliable, as well as the **DATA** part which is classically considered as the part to work on.

A prototype has been implemented and is shown in figures Figure 14 and 15.

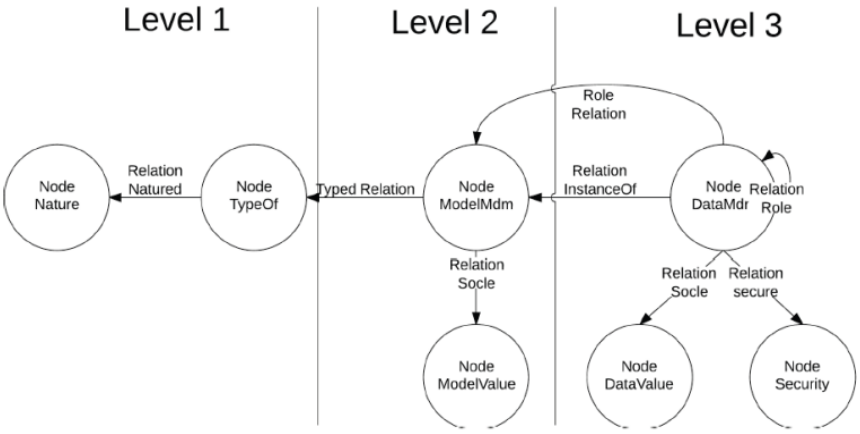


**Figure 13.** Modelization Layers

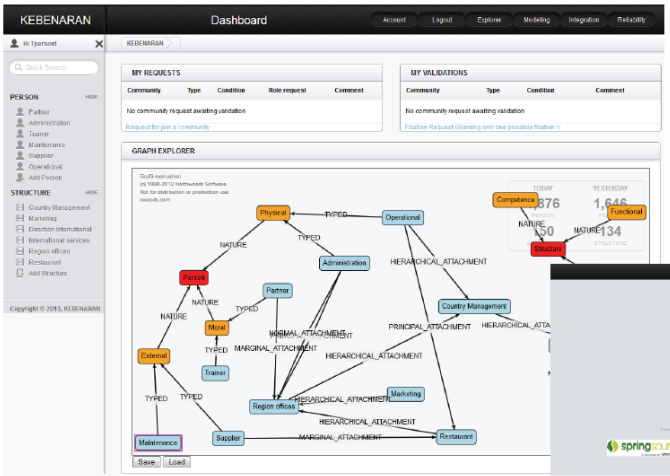


**Figure. 14.** The KEBENARAN System

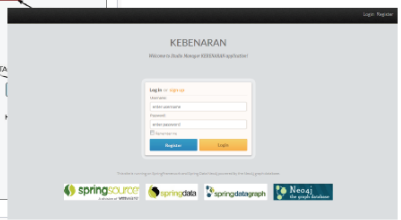


**Figure. 14.** Logging into KEBENARAN

## 4.2 Graph Databases

KEBENARAN heavily relies on relationships: a data is linked to some people, structures, resources, etc.

In a relational model, each link between data is materialized by a join operation. In a graph database, the act of traversing over an edge is also the act of joining. However, what makes this more efficient is that traversing from one vertex to another one is computed a constant time operation.

As shown in [25], basic queries are often at least of complexity O(log2(n)) for relational databases as data must be retrieved and joined, where n stands for the number of persons in the database. This complexity is dramatically reduced
(down to O(1)) in graph databases as connected data can be directly accessed without costly join operations.

Thus, traversal time is defined solely by the number of elements touched by the traversal. This is independent of the size/topology of the graph as a whole.

That is to say that, if a graph database is populated of 10 billion of nodes or 10 nodes, the time to accomplish the node traversal for a constant set of traversal concerned data is the same.

Neo4j has been chosen to store the KEBENARAN data. Neo4j is the world leader graph database system, for the following reasons:

- Neo4j is the first graph database ranked as being already adopted as shown in [26];
- Neo4j provides both functional and declarative traversal capacities;
- Neo4j is one of the most advanced graph databases with extra features that open new possibilities for further work;
- Neo4j is a java-based open source project so the code is available and can be upgraded by the community;
- Neo4j provides excellent capacities to query over labels;
- Neo4j offers ACID capabilities.

As Neo4J is a graph database, MDM solutions based on it will thus run on less hardware with better performances. They will also be more easily adaptable to change than their relational counterparts, reducing both capital and operational expenses. Naturally, MDM applications are mission critical systems, requiring full support for distributed transactions, ACIDity, and highly availability. Neo4J supports all of these, with high-availability clustering and online backups. As a native graph database, Neo4j is built from the ground to be fully ACID. Since our choices, some MDM systems based on Neo4j (e.g., CISCO [27]) were released, showing the interest of our approach.

## 5 CONCLUSION

In this paper, we introduce an innovative MDM system relying on Neo4j. This system offers three main innovations: a graph-oriented approach of master data, a generic solution suitable to any organization and a collaborative system.

We show that Neo4j is efficient for building such robust and relevant MDM solutions. We demonstrate that several types of organizations can be managed with our system and that processes easily integrate the collaborative concepts.

Further work includes the release candidate version of our solution, together with the study of the many research and development avenues open by our work.

First, we will explore how open data and linked open data can be connected to our solution. This work will rely on a study of the general issue of managing distributed data and distributed sub-systems that must be reconciled. In extended organizations, it will be often the case that several systems will work in a parallel manner in local sub-organizations. Reconciling them is a key challenge. The integration of KEBENARAN in existing information systems is also a

challenge to be addressed, so as to facilitate it as far as possible. Moreover, KEBENARAN's ETL will be discussed in future work.

## REFERENCES


1. Haug, A., Zachariassen, F., van Liempd, D.: The costs of poor data quality. Journal of Industrial Engineering and Management **4**(2) (2011) 168-193.
2. Friedman, T., Smith, M.: Measuring the business value of data quality. Gartner (2011).
3. InsightSquared: 7 facts about data quality. (2012).
4. Chappell, D.: Enterprise Service Bus. O'Reilly Media, Inc. (2004).
5. Otto, B., Reichert, A.: Organizing master data management: Findings from an expert survey. In: Proc. of the 2010 ACM SAC, ACM (2010).
6. Berson, A., Dubov, L.: Master Data Management And Customer Data Integration For A Global Enterprise. McGraw-Hill Education (India) Pvt Limited (2007).
7. Loshin, D.: Master Data Management. Morgan Kaufmann (2009).
8. Marco, D.: Building and Managing the Meta Data Repository. Wiley (2000).
9. Jeremy Kashel, T.K., Bullerwell, M.: Microsoft SQL Server 2008 R2 Master Data Services. Packt Publishing (2011).
10. Talend: Open studio for mdm, http://tinyurl.com/nnrav9p (2013).
11. Oracle: Master data management, http://tinyurl.com/kacketb (2013).
12. David Butler and Bob Stackowiak: Master Data Management. White Paper (2012).
13. SAP: Netweaver master data management, http://tinyurl.com/ply7tt4 (2013).
14. Loren Heilig, Ste_en Karch, Oliver B • ottcher,Christiane Hofmann and Roland Pfennig: NetWeaver Master Data Management. Galileo Press (2007).
15. IBM: Master data management, http://tinyurl.com/oegkgsu (2013).
16. Jan-Bernd Bracht, Joerg Rehr, Markus Siebert and Rouven Thimm: (Smarter Modeling of IBM InfoSphere Master Data Management Solutions).
17. Networks, O.: Meet ebx5, http://tinyurl.com/lj_266 (2013).
18. Informatica: Master data management, http://tinyurl.com/pxv393d (2013).
19. InformationBuilders: Master data management, http://goo.gl/q5wzh3 (2013).
20. Ross, J., Weill, P., Robertson, D.: Enterprise Architecture As Strategy: Creating a Foundation for Business Execution. Cambridge, Harvard Business School Press (2006).
21. Newell, M., Grashina, M.: The Project Management Question and Answer Book. Amacom, New York, NY, USA (2003).
22. Sehra, S.S., Singh, J., Rai, H.S.: Assessment of openstreetmap data - a review. Int. Jal of Computer Applications **76**(16) (2013).
23. Buffenoir, E., Bourdon, I.: Reconciling complex organizations and data management: the panopticon paradigm. CoRR **abs/1210.6800** (2012).
24. Sakr, S., Pardede, E., eds.: Graph Data Management: Techniques and Applications. IGI Global (2011).
25. Rodriguez, M.A., Neubauer, P.: The graph traversal pattern. CoRR **abs/1004.1001** (2010).
26. Board, T.T.A.: Technology radar, http://goo.gl/q5wzh3 (May 2013).
27. Malhota, P.: Hierarchy management with neo4j - CISCO. In: GraphConnect. (2013).