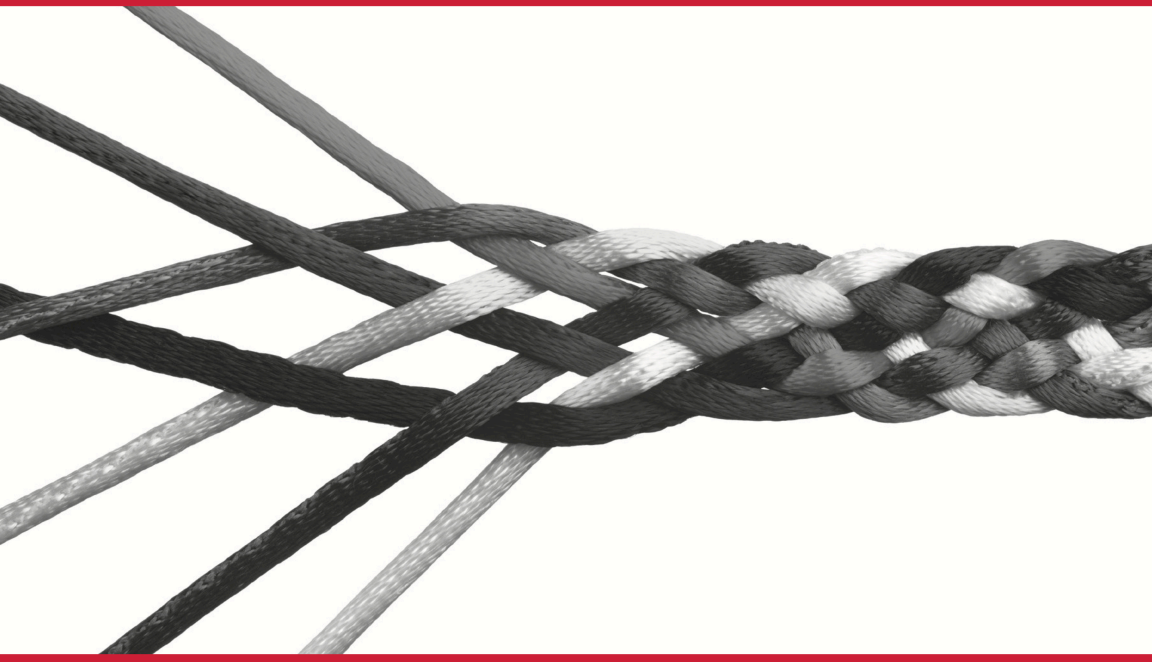
O'REILLY®

Compliments of MarkLogic®

Understanding Data Governance

Practices and Frameworks for Regulatory Compliance and Security



Federico Castanedo

Your data deserves better.

There's nothing wrong with your data that a better database can't fix.

- Rely on a 100% trusted, enterprise-grade multi-model database
- Load your data *as-is*, no upfront data modeling required
- Unify your structured and unstructured data
- Build and deploy your apps in days, not months

www.marklogic.com

MarkLogic®

Understanding Data Governance

Practices and Frameworks for Regulatory Compliance and Security

Federico Castanedo

Beijing • Boston • Farnham • Sebastopol • Tokyo O'REILLY®

Understanding Data Governance
by Federico Castanedo

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*http://oreilly.com/safari*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Shannon Cutt
Production Editor: Melanie Yarbrough
Copyeditor: Octal Publishing Services
Proofreader: Charles Roumeliotis

Interior Designer: David Futato
Cover Designer: Karen Montgomery
Illustrator: Rebecca Demarest

October 2017: First Edition

Revision History for the First Edition
2017-10-05: First Release

See *http://oreilly.com/catalog/errata.csp?isbn=9781491990780* for release details.

978-1-491-99076-6

[LSI]

Table of Contents

Understanding Data Governance

Introduction

From the moment organizations begin collecting information, *governance* becomes crucial: what information should we gather, and how do we ensure that information is accurate and current? What are we allowed to do with the information we gather? How do we store it, and for how long? Who can see it? And finally, how is it updated?

With the digitization of virtually everything, the importance of governance rises as we need *more* context around data that drives decision-making. *Context* includes lineage and provenance: who created the data? Where did it come from? Has it been versioned? Is it accurate? For example, when considering data quality, using out-of-date information to make critical, organizational decisions is very risky. If several data sources have similar data, how do you decide which data source is the "golden source"?

Lack of governance increases risk to the organization. In fact, the financial crisis of 2008 occurred largely because of a lack of regulation around the *quality* of the data. As legislation tightens, stricter reviews of data look to reduce risk. These reviews follow through in all industries, particularly around security and privacy.

Data governance is supposed to include all of the processes that ensure data assets are formally managed throughout an organization; in other words, the policies developed to define accuracy, accessibility, consistency, relevance, completeness, and management of the organization's data assets.

What really happens is policies are determined in a business suite, passed on to department heads, which are passed on to technology teams, and then implemented by developers. As you can imagine, this process has been failing for years and is only getting worse. In fact, with the General Data Protection Regulation (GDPR), it makes organizations accountable to every single citizen with whom they manage personal data as each citizen can demand precise control over the usage of their data. Therefore, broad-brush policies do not work in this case; fine-grained yet scalable control of data driven by policies that can be implemented at the data level.

Data-driven governance is a new approach in which governance is established at the data level, not at the system level via some separate platform wholly apart from the data. When governance is established at the data level, database logic takes care of the specific constraints given by each of the customers to their data or by the specific regulation. In contrast, when governance is established at the code level, it is necessary to develop specific code to ensure the required constraints. Focusing on data makes automation and enforcement easier. In this report, we describe how you can establish data governance at the data level as well as the practices and frameworks for regulatory compliance and security.

Every organization must make several key decisions about their data. For instance, it is necessary to define how data is stored, archived, backed up, and protected. Lawyers are brought in to understand rules and regulations, which can and often do, frequently change. Moreover, a set of procedures must be developed to define how data—and metadata—will be accessed by authorized personnel. Organizations must also be able to audit their security policies. An organization must know the authenticity of their metadata (provenance, structure, context, and so on) so that it can determine the validity of the data to which it refers. For example, a forecast team might use a max flow number to decide when flooding is imminent. But, what happens if there are bottlenecks in the river where the *local* max flow number is less than the *overall* max flow number. Source and context are critical in determining data quality and appropriateness.

Lastly, a set of controls and audit procedures must be established to ensure ongoing compliance with government regulations. If rules change, how can it be ensured that stakeholders will be notified so they can take appropriate action.

Data governance is used to ensure that input data meets precise requirements, such as specific business rules, data definition, and data integrity constraints in the data model. However, this often is not the case. For example, when data is extracted from source systems and massaged for its target usage, attempts to get clean and consistent data often fail. Why? Because each system will have different standards on how the data is stored, inputted, and managed. When these datasets are brought together these differences in quality and consistency will be multiplied resulting in either poor-quality output, or significant manual effort to fix. Data governance is also needed in order to minimize the cost of data silos, because it defines a clear procedure on how to share data across different departments. However, minimizing the cost of data silos is a difficult task. Again, this is because each silo was built for its specific purpose, with the data modeled against those requirements—often in incompatible ways with the other silos.

Let's take a step back and consider how data silos commonly arise such as when companies grow, when new departments create isolated databases, or when acquisitions take place that make it necessary to integrate redundant subsystems. The common practice of having one database for every application and the emergence of third-party, cloud-based technologies means schemas are organic and unique. Without any canonical standard, the integration of data from all these individual silos makes data governance difficult to achieve.

Taking a Step Back

There are two accepted models that should be considered concerning governance: Confidentiality, Integrity, and Availability (CIA) and Authentication, Authorization, and Accounting (AAA).

For instance, in cybersecurity, practitioners often talk about a triad of basic security principles: confidentiality, integrity, and availability. This expands upon the narrow view of cybersecurity as "keeping the bad guys out." It is just as important to ensure that data is protected from unauthorized users, and that the good guys can rely on the data they find.

Broadly speaking, we can define the CIA triad as:

Confidentiality
: Data is secured for authorized users and not exposed to unauthorized parties or systems.

Integrity
: Data is trusted by authorized users and has not been altered by unauthorized parties or systems.

Availability
: Data is accessed by authorized users and cannot be "brought down" by unauthorized parties or systems.

Again, broadly speaking here are the definitions for AAA:

Authentication
: Identifying that only the correct individuals (legitimate users) are accessing a given system or network.

Authorization
: Identifying what commands that user can do (role-based permissions).

Accounting
: Measurement of what the user has done. (Sometimes this is called auditing)

Although CIA is used more broadly, AAA is usually more directly applicable to how systems are actually designed. The principles of AAA, then, when implemented correctly, support the overall achievement of the CIA principles.

What this report will show is that there is technology that can finally marry policies to the data and offer fine-grained access controls. Further, these controls must travel with the data—no matter where the data is moved to!

The Rise of the Data Lake

One of the methods used to try to overcome disparate data is the creation of *data lakes* and *data warehouses*. A data warehouse is a structured repository that contains a collection of all an organization's data. A data lake is similar to a data warehouse, but can contain other input data types besides relational data (rows and columns), such as semi-structured data (CSV, logs, XML, JSON), unstructured data (emails, documents), and binary data (images,

audio, video). The purpose behind a data lake is to have all different types of information in a single repository, accessible to different types of users.

Developing a successful data governance strategy requires careful planning, the right people, and the appropriate tools and technologies. It is necessary to have a global view of all the company's data assets and implement the required policies across all the data.

Finally, successful compliance consists of establishing the adequate procedures and policies in order to guarantee that everyone in the organization acts in accordance with the regulatory framework and organizational needs. The practice of compliance appeared in the financial sector due to the large amount of regulations required historically by this sector. After the financial crisis happened in 2008 due to the low-quality subprime loans, the stronger focus on risk prevention made regulatory compliance increasingly complex. However, compliance and data security are mandatory in any regulated industry, so it is important to integrate the regulatory framework in order to minimize updates and simplify maintenance processes.

Within the compliance framework, we need to consider not only the legal regulations, but the internal policies, the customer's and provider's agreements, and the ethical frameworks of the organization itself. Because compliance affects every area and department, it must be integrated into the organization's environment. In the case of multinational organizations, compliance is even more important, because it is necessary to take into account the regulations of each country in which they operate, and regulation can be very different. A flexible framework that incorporates metadata can address the governance, risk, and compliance (GRC) necessary to meet the needs of the varying regulations no matter the jurisdiction.

A Flexible Framework

In the following sections, we explore these concepts in greater depth.

Data Governance Policies

Any large organization—a corporation, university, hospital, state or local government—that must make data available to third parties or

share it among their departments and with governments, faces the challenge of defining its data governance policies.

Data governance policies typically affect the following areas:

Security
: As mentioned previously, CIA and AAA are crucial. On the one hand, maybe you shouldn't be showing your data to everyone, but, on the other, you need to share it. So, it is necessary to implement the right security policies, which can include anonymization or redaction of data.

Life cycle
: Maybe you need to save the customer data for almost 20 years. For example, MiFID II requires that companies must make records available to their customers for a retention period of five years and for up to seven years for regulators. And under the US Law Enforcement 28 CFR 23 "Criminal Intelligence Systems Operating Policies" certain intelligence data are only allowed to be stored for five years unless it has been updated. On the other hand, the EU GDPR mandates data can be stored while consent has been granted (which can be revoked), assuming no other regulation overrides this rule (which the MiFID II exactly would do). So, understanding how each regulation interplays is critical.

Compliance
: It is necessary to be able to audit or account for how data was used. For example, in healthcare under HIPAA, you can request a report on when and why your health information was shared for certain purposes.

Deletion
: If your customer asks you to—or if the regulators require you to delete information. For example, in the EU there is the so-called "Right to be forgotten" that EU citizens can exercise as well as requesting that all their personal data is deleted.

Moreover, there are country-specific, pan-regional, and industry-specific regulations that elevate the data governance strategy conversation from the CTO/CDO agenda to their peers in the GRC leadership. Even public-sector institutions are not immune to regulatory compliance and the constraints it imposes on their data governance strategy.

In Australia, for example, a number of agencies (primarily the National Archives of Australia) imposed security and life cycle, retention, and disposal obligations on public sector entities with regard to keeping records. The requirements that Australian Government agencies need to meet in relation to records management derive from multiple sources. Access to, preservation, and destruction of information created and received when undertaking Australian Government business is governed by the Archives Act Law of 1983.

A key focus of the Archives Act is authorization of the disposal or destruction of records by the Archives approving records authorities. Records management obligations are also contained in other Acts, including the Freedom of Information Act 1982 and the Financial Management and Accountability Act 1997 (FMA Act).

The Archives Act also has a key role in establishing standards and providing guidance and assistance to agencies in managing their records management responsibilities. An important piece of guidance issued by the Archives Act is Check-up 2.0, which establishes the minimum requirements for information and records management. These requirements cover agencies' information and records management arrangements and practices, including frameworks, records' creation, capture and destruction, and business systems.

Retention and sentencing/disposal of records is governed by instruments called *records authorities*. A records authority is a formal instrument, published by NAA, that defines the retention periods and consequent disposal actions authorized for classes of records described in the authority. Following a particular period of time, disposal action includes destruction or transfer to the Archives. Records authorities typically apply to the core business records of a single agency or body, whereas general records authorities, such as the Administrative Functions Disposal Authority (AFDA), normally apply to Australian Government agencies and bodies.

A number of other Australian Government agencies issue policies, standards, and guidelines relevant to the management of records, including electronic records. For example, the Attorney General's office has established the Protective Security Policy Framework (PSPF), which outlines mandatory minimum security requirements for all agencies, and has implications for records management.

Sometimes, however, these policies are in conflict with one another. What if the regulators say you must hang on to a record, but the customer says delete it?

Along with all of these requirements and requests, you also need to be able to do the following:

- Design and operate a management system to assure that data deliver value and are not a drain on resources.
- Designate who can do what to the organization's data, and how.
- Ensure standards and regulations are set and met.
- Support a high-level, strategic view across the organization.
- Ensure management system can be updated to address new policies and amendments
- Incorporate new data sources as policies evolve.
- Ensure that data is reliable, appropriate, and usable in support of existing and new policies.

Additionally, note that policies can include life cycle data management, master data management, data quality (for example, ensuring that data matches certain patterns), access controls, privacy, security, provenance and lineage, availability, and organizational-specific policies.

Data Governance Practices

Data governance is about *policy execution*, the ability to represent arbitrarily complex policies as well as track and enforce those policies. When we say *policies*, we mean any kind of policy, not just those used for government regulation. For example, we might want to have documents containing the string "future trade" separated and stored in long-term storage. Also, from a privacy perspective, policies could be about data access, data anonymization, or encryption.

Modern data environments generate and store all kinds of data, and data variety is a salient feature of current data lake environments, which manage and integrate diverse types of data: structured, unstructured, and semi-structured. Some companies need to keep these data for years. Additionally, metadata can be used to evaluate and define rules that ensure data governance constraints, so the abil-

ity to keep metadata with the data to search and query it is a very powerful capability.

Data governance is often managed by using a report of the data and looking manually for clear errors or alarms. However, the following questions may be difficult to answer with this procedure: is this the right data? How do we know? Is it all of the data?

An organization may also have a separate governance platform in which the policies are listed, and it might include a data catalog, monitoring, and measurement. The main drawback of this model is that the data governance platform is disconnected from the data. A better option integrates data governance within the data itself (a.k.a. data-driven governance).

Data Governance Technologies

Current technology should support data governance policies and practices and simplify the process of data governance. Data is usually stored in a database management system, which also supports queries over stored data. Traditional databases are based on the relational model and are known as Relational Database Management Systems (RDMS). In RDMS-type databases, data is presented to the user as *relations* stored in a tabular form; that is, in a collection of tables wherein each table consists of a set of rows and columns. Recent database technologies known as NoSQL provide a way to store and retrieve data that can be modeled in a nontabular form.

It can be very difficult to manage policies when the *data* and the *schema* needs to change frequently. For instance, although you might be able to add access controls on a given column—or add a specific column for lineage—how do you add lineage to each *row* (i.e., where does a specific row come from)? It becomes very difficult to update the relational schema and store that information. In general, traditional relational databases are not good for data governance because you need to define your schema in advance.

NoSQL databases can be divided into the following types: key–value, column, document-oriented, and graph-based. NoSQL document-oriented databases provide a more natural way of modeling governance, particularly as data and data structures change. A multimodel database is a more general NoSQL approach and it can store, index, and query data in one or more of the previous models. The

flexible nature of a document/graph (and therefore multi-model) database make it more natural to address data governance because governance attributed can be easily added, removed, or modified to each data item. Although each NoSQL database has its strengths, if you are dealing with multistructured data, the multi-model, document-oriented approach is best, as it gives the most flexibility with the least amount of lift.

Treating Each Row as a Document

In the book *Building on Multi-Model Databases*, we learn that we can easily store an *entity* (a customer, a transaction, a vendor, a material, and so on) as a document. Each attribute of that entity (no matter how many tables they are spread across) such as element values for XML or property values for JSON, will be indexed and nested under the parent. This requires a denormalization of all entities—or a no-op transform when migrating from relational.

Envelope pattern

In multi-model databases, a simple yet powerful pattern has emerged that MarkLogic calls the *envelope pattern*.

Here's how it works: take your source entity and make it the subdocument of a parent document. This parent document is the "envelope" for your source data. As a sibling to your source data within the envelope, you add a header section where you start to add your standardized attribute names, attribute values, and attribute structure. In our example, we've standardized the "Zip" and "Postal" attributes as "Zip" properties in both envelopes (see Figure 1-1). As a result, we can now issue a structured query of the type "Show me all customers having a zip code equal to 94111" and get both entities back.

Figure 1-1. The envelope pattern in action

The benefits to this approach are numerous:

- We retain our source entity *as is*. This is a big benefit for highly regulated environments that come in and request to see the entities as they were originally imported into our database.
- We standardize only what we need, when we need it. We don't require all the mappings up front. Now we iteratively manage our data in place.
- We can continue to update our model later without reingesting.
- We can divide and conquer the standardization process across development.

We can add all sorts of information to envelopes, such as source, data, lineage, permissions—anything you want (Figure 1-2).

Figure 1-2. Adding more information to the envelope

Figure 1-2 demonstrates that you can do the following:

- Preserve and query lineage, provenance, and other metadata.
- New schemas don't affect existing schemas or applications.

What about relationships?

In the relational world, we would have joins that would link a row to other rows; for example, associating a customer to an order. Ironically, relational is really bad at modeling relationships! However, in multi-model, customer and order data would now be represented by documents (instead of a row). So how would we "join" these two? You could use foreign key joins, but in multi-model there is a better way: *semantic triples*.

Triples have their name because they consist of a Subject (entity), Predicate (the association), and an Object (another entity). SPARQL is the query language that allows you to query this type of data. Further, by having a multi-model database that allows semantic associations between documents, we can now add context to data—as metadata. And that metadata can be easily amended over time as it grows and changes. MarkLogic is the reference for multi-model databases, and it can use semantics and an envelope pattern to support data governance.

A multi-model database can support a document model, key–value pairs, semantics, and search (see Figure 1-3). In MarkLogic, docu-

ments can consist of JSON and XML, RDF, triple models in the semantic world, or graph-based models.

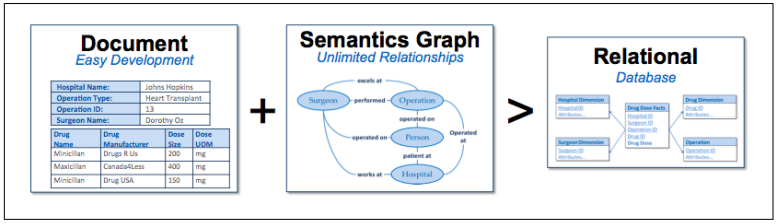


Figure 1-3. A multi-model database (from the healthcare scenario) for which a document model and a semantic graph allow for a strong data governance (lineage, history, life cycle, and security). Image courtesy of MarkLogic.

In the envelope pattern, core content and headers are kept separated. Headers contain metadata about how data sources are ingested, such as timestamp, data source, batch job, or external manifest associated, as well as data type, validation, or quality issues detected, including the URI of a binary document if the (XML/JSON) document is associated with a binary. Core content is the raw content ingested, which typically is unchanged other than to convert text formats to XML, JSON, or RDF.

True multi-model databases have the ability to store multiple types of data in the same system that has unified data governance, management, and access. And if you are storing it, you also must be able to search it. The composability of a search is crucial; you must have a database that handles different data models and indexes them so that you can run combined queries of text, SPARQL, XQuery, and SQL from industry standard APIs such as REST, JavaScript, and Java.

What about permissions?

So, we have the entity–including all its rich attributes, now modeled as a document. In our envelope, we can include where this information came from, when it arrived, and any associations to other entities. What about permissions? What about our tenants of AAA?

This is the beauty of handling governance in multi-model. We can decide that different aspects of the data within a document can be made available, whereas other attributes can not.

For example, in a patient record, we could decide that a patient's name would be accessible by only the practitioners and the accounting department. But the patient history would never be available to accounting. Further, we could decide that a researcher could have access to everything but the person's identifying information, name, social, and address *number*, but for research purposes, we might want to know the street name and zip.

This capability is called *fine-grained access controls*, or *element level* security. This means that every element within the document can have unique permissions on it—depending on the organizational policies.

Further, if the data is moved into a data lake built on multi-model, the *permissions stay with the data*. This happens because a multi-model database that uses sophisticated indexes can store these permissions. The query runs against the index, not the data itself. So when the query is run, it checks to see if the user has the permission to even see the index!

Flexible Data Governance Framework

The data landscape is evolving, incorporating increasingly large volumes of data. In the case of NoSQL databases, the data can be composed of structured, unstructured, geospatial, and linked open data, among others. Although the NoSQL movement is widely established, there is a tremendous amount of information residing in relational databases in companies nowadays. As previously mentioned, relational databases lack the agility needed for integrating or upgrading data into common data governance framework. So, what do you do with data in these traditional silos?

A flexible data governance framework prevents situations in which complex engineering systems have several disconnected pieces with expensive hardware. Often, the disconnected pieces are built through the integration of several open source projects. The problem with these hand-crafted, engineered architectures is that you must also develop the necessary glue to hold together the different pieces. Most of the time, each technological component of this architecture has been tested alone in order ensure the complete data pipeline works. In contrast, a unified data governance architecture has the benefit that is has been already tested on the end-to-end use case.

A flexible data governance framework should be able to ingest data without needing a long Extract-Transform-Load (ETL) process. Data (or is metadata) should move to a flexible framework in order to model the required data governance policies. This flexible model must also support schema-free databases where timely data from multiple and disparate sources, that are usually not normalized, are stored in any schema. The ability to support semantic relationships is another important characteristic. Relationships can be modeled using graphs, RDFs, triples, or ontologies, and are valuable for lineage support. Figure 1-4 shows an example of how relationships among different entities can be defined using RDF triples.

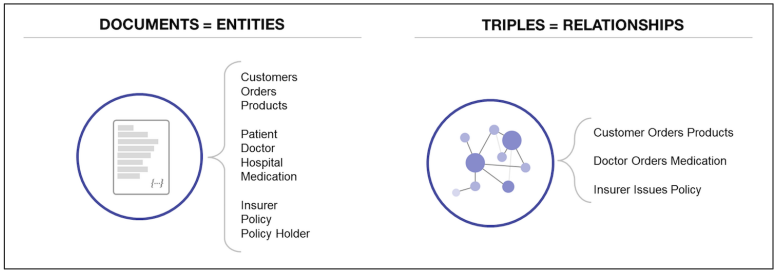


Figure 1-4. A multi-model database system that stores all the entities as documents and the relationships as triples

The capacity to ingest data without needing a long ETL process, the use of metadata, schema-free databases, and the ability to support semantic relationships simplify the implementation of data governance and regulation processes within an organization.

Search with Unified Index

As we noted, the document model has the advantage of organizing data in a more natural manner. For example, all the information about a customer can be stored in one document, instead of scattered over dozens of relational database tables.

However, key to any multi-model approach is to ensure that a document model coupled with search means that the metadata is preserved, the ontology and history of the data are preserved, and data can be utilized immediately in supporting policies, whether the policies are for data life cycle management or GRC. It's imperative that your clustered DBMS create one integrated suite of indexes on data as it is ingested to allow fast query of that data. This allows a single,

composable query to quickly retrieve data across all the data models, simultaneously.

As authors Pete Aven and Diane Burley wrote in *Building on Multi-Model Databases*:

> The beauty of indexing in a document store is that it also provides for structured search, indexing the inherent structure of the document in addition to the text and values within the document. That is to say, documents have plenty of structure, including titles, section headings, book chapters, headers, footers, contract IDs, line items, addresses, and subheadings, and we can describe that structure by using JSON properties or XML elements. A multi-model database creates a generalized index of the JSON and XML values for that structure and indexes every value and every parent–child relationship that has been defined. In other words, the document model operating in a multi-model database, where anything can be added to any record (and is immediately queryable in a structured way) is far more powerful and flexible than the related-table model of a relational database for storing rich representations of entities. It also allows us to store, manage, and query relationships between those entities.

With a multi-model database, we can also extend the generalized document text and structure index with special purpose indexes such as range, geospatial, bitemporal, and triples.

Making Rational Policy

The main challenge with data governance is that "policy" is a vague term. You can have all sorts of policies—you can have geo-data policies, in-country processing policies, and so on. You can have different data life cycle policies; for instance, data might need to be deleted upon the request of customers, or a regulator may demand you preserve data for 10 years.

With the previous constraints, it was impossible to build a single relational schema to centrally handle changing data and policies. Semantics provides a great way to model such a schema so that you can infer which policy takes precedence; they are terrific for expressing complex metadata in a very flexible way. Semantics provide another level of capability to make inferences, such as this one: Carlos is a citizen of Spain; Spain is a member of the EU; so any EU policy applies to Carlos. In addition, enveloping can be used to dynamically wrap different forms of data (Spain = España = Espagne).

You might also have policies that are internal, though not necessarily regulatory. For instance, you might have an internal policy that states there must be a Recovery Time Objective (RTO) with a gap of no more than 24 hours for one kind of data, and no more than a month for another. Because it's impossible to predict what you will want to do tomorrow with 100% certainty, it is important to have a flexible way to incorporate constraints in the future, without incurring any new development to the system.

Another challenge in data policy involves dealing with near-real-time (or run the business) and historical (observe the business) data, and how policy execution should be carried out together with operations. If policy is integrated at the database level instead of at the application level, the execution of the specific policy will be easier because it requires less coding effort to implement. Furthermore, measuring and monitoring become trivial when all the data and the policies are together, reducing a major cost.

Of course, to centrally manage data policy at the database level, the database must ensure data quality and data consistency and prevent data loss, so ACID transactions are a must. And because availability is one of the three pillars of the CIA triad of security, proper governance also requires that the database offer High Availability and Disaster Recovery (HA/DR).

Finally, in addition to data quality and availability, it is also important to make sure the data is *accessible* for developers. MarkLogic exposes data using a variety of standard programming APIs so that solutions developers can create and execute policies utilizing all of the security and data protection capabilities in MarkLogic. You can create policies such as backup, retention, data access, data life cycle (with tiered storage) and authentication utilizing existing MarkLogic APIs.

Data Security and Regulations

In the past decade, almost every industry found it necessary to transform their internal processes in order to respond to an increased focus on Governance, Risk, and Compliance (GRC). The global economic crisis of 2008 resulted in new risk regulations for banks. Consumer privacy laws in the European Union introduced regulations for companies conducting business on the web (such as the EU GDPR). Additionally, health-care organizations are continu-

ously challenged by new privacy regulations because they are changing the way they deliver patient care. The trend is to combine all related touchpoints (hospital, prescription, doctors) of a patient that are stored in different databases.

Data security is also becoming critical. According to a recent report from PWC, the number of security incidents in 2015 increased by 38% compared to 2014. This escalation is reflected in the amount of money spent on security: worldwide security spending has been estimated at $76.9 billion in 2015, and is projected to grow to $170 billion by 2020, according to Gartner. Moreover, each cyber incident costs U.S. companies a reported $7.1 million on average or $221 per record, and 63% of organizations are deploying new IT prior to having appropriate data security measures in place. The May 2017 WannaCry ransomware attack where attackers demanded bitcoin payments is a good example of how a massive cyberattack can affect business on a global scale.

A good data governance model should cover data security in order to avoid putting data at risk. In the context of regulation, this means that you need to ensure that your data is secure, and you need to trust your data.

The MarkLogic security model (depicted in Figure 1-5) provides fine-grained authorization with Role-Based Access Control (RBAC), in which each user is assigned to any number of roles, and these roles have associated permissions and privileges. The permission controls what documents the user can read, insert, and update, and the privileges control what actions the user can perform. To access documents stored in MarkLogic, a user has to have an authorized role. This fine-grained access control can operate at scale with huge amounts of data, and the security does not affect the high transaction performance of the database.

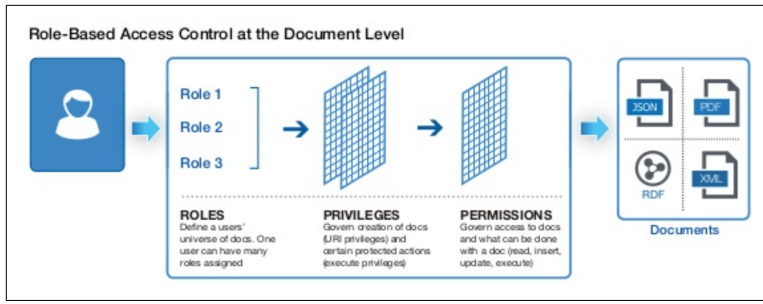


Figure 1-5. The MarkLogic security model

According to the Institute of International Finance (IIF), a financial institution can spend over $1 billion every year on regulatory compliance and controls. For more information on the cost of compliance and the related challenges, refer to the Thomson Reuters report. In another report, the Spanish bank BBVA recently estimated that on average financial institutions have 10 to 15 per cent of their staff dedicated to this area. To simplify the process and reduce costs, several start-ups that enable simple, reliable, and cost-effective regulatory compliance through the use of new technology have emerged. The companies that focus on the automation of manual process to ensure regulatory compliance are known as *regtech*. For instance, Suade, a Microsoft ventures startup, wants to help big and small banks adapt to changes in regulation and become more cost effective by providing a platform that keeps banks in line with the latest requirements.

Regtech companies also assist financial institutions in ensuring employees are following the required compliance. For example, the IBM Surveillance Insight for Financial Services identifies individual employees who might pose financial risks. The conclusions are based on a "cognitive reasoning engine" developed using artificial intelligence to analyze data and detect signals. Sybenetix is another company using novel technologies such as cloud computing and artificial intelligence to help the financial services industry with conduct monitoring and detecting suspicious trading.

Some of the biggest challenges of the financial industry include modeling, scenario analysis, and forecasting, processes on which banks spend a lot of money in order to complete regulators' stress tests. Citigroup, for instance, uses the artificial intelligence system

from Ayasdi to help it pass the U.S. Federal Reserve's stress test, having failed it the previous year.

Along with the innovations of regtech startups that are aiming to disrupt financial services regulation process, it is important to have the technical capacity to couple internal data with regulatory policies. A multi-model database like MarkLogic can help to drive down the cost of regulatory and compliance by simplifying and standardizing compliance processes, and reducing the need for manual and duplicate checks. Because in a multi-model database you can keep the data in its original source form, it is possible to eliminate some unnecessary ETL process. By storing all entities (customers, financial instruments) as documents and wrapping the entity in an envelope, we can add anything else we may want to capture along in the envelope, such as metadata or semantic triples.

Summary

Data governance is necessary in order to meet strategic business requirements. Through the definition of a set of policies, data governance ensures data is managed effectively, reduces regulatory compliance risk, and improves confidence in operational and management decisions.

Further, regulated industries such as finance and healthcare spend a tremendous amount of money on regulatory compliance and controls that are constantly changing. In a governance "arms war" a flexible platform that help define governance policies easily at scale helps in reducing compliance costs.

Flexible and novel data governance platforms should support the integration of data of different types (JSON, XML, tabular, and so on) and from different sources (Hadoop, RDBMS, filesystems, and so on). Since different types of input data can be stored, some ETL processes can be removed. These platforms must provide an easy way to retrieve and query specific parts of the data and it is common to have a SQL interface that also allows the user to interact with this data.

The ability to load different types of input data into a unified data model without having to model it in advance is a huge advantage, and can be achieved using a multi-model database. Moreover, having text search capabilities within the platform becomes very useful.

Finally, the ability to model relationships using RDF triples is another important feature to capture complex interactions easily.

About the Author

Federico Castanedo is the Lead Data Scientist at Vodafone Group in Spain, where he analyzes massive amounts of data using artificial intelligence techniques.

Previously, he was Chief Data Scientist and cofounder at Wise Athena, a startup that provides business value through artificial intelligence.

For more than a decade, he has been involved in projects related to data analysis in academia and industry. He has published several scientific papers about data fusion techniques, visual sensor networks, and machine learning. He holds a PhD in Artificial Intelligence from the University Carlos III of Madrid and has also been a visiting researcher at Stanford University.