



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÀSTER EN INGENIERÍA INDUSTRIAL

DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL SECTOR

Autor: Karim Kadbey Nasser-Eldine
Director: Carlos Martín Orozco

Madrid
Enero 2019

Karim
Kadbey
Nasser-Eldine

**DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL
SECTOR**



AUTHORIZATION FOR DIGITALIZATION, STORAGE AND DISSEMINATION IN THE NETWORK OF END-OF-DEGREE PROJECTS, MASTER PROJECTS, DISSERTATIONS OR BACHILLERATO REPORTS

1. Declaration of authorship and accreditation thereof.

The author Mr. /Ms. KARIM KADBEY NASSER-ELDINE

HEREBY DECLARES that he/she owns the intellectual property rights regarding the piece of work: DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL SECTOR that this is an original piece of work, and that he/she holds the status of author, in the sense granted by the Intellectual Property Law.

2. Subject matter and purpose of this assignment.

With the aim of disseminating the aforementioned piece of work as widely as possible using the University's Institutional Repository the author hereby **GRANTS** Comillas Pontifical University, on a royalty-free and non-exclusive basis, for the maximum legal term and with universal scope, the digitization, archiving, reproduction, distribution and public communication rights, including the right to make it electronically available, as described in the Intellectual Property Law. Transformation rights are assigned solely for the purposes described in a) of the following section.

3. Transfer and access terms

Without prejudice to the ownership of the work, which remains with its author, the transfer of rights covered by this license enables:

- a) Transform it in order to adapt it to any technology suitable for sharing it online, as well as including metadata to register the piece of work and include "watermarks" or any other security or protection system.
- b) Reproduce it in any digital medium in order to be included on an electronic database, including the right to reproduce and store the work on servers for the purposes of guaranteeing its security, maintaining it and preserving its format.
- c) Communicate it, by default, by means of an institutional open archive, which has open and cost-free online access.
- d) Any other way of access (restricted, embargoed, closed) shall be explicitly requested and requires that good cause be demonstrated.
- e) Assign these pieces of work a Creative Commons license by default.
- f) Assign these pieces of work a HANDLE (*persistent URL*). by default.

4. Copyright.

The author, as the owner of a piece of work, has the right to:

- a) Have his/her name clearly identified by the University as the author
- b) Communicate and publish the work in the version assigned and in other subsequent versions using any medium.
- c) Request that the work be withdrawn from the repository for just cause.
- d) Receive reliable communication of any claims third parties may make in relation to the work and, in particular, any claims relating to its intellectual property rights.

5. Duties of the author.

The author agrees to:

- a) Guarantee that the commitment undertaken by means of this official document does not infringe any third party rights, regardless of whether they relate to industrial or intellectual property or any other type.

- b) Guarantee that the content of the work does not infringe any third party honor, privacy or image rights.
- c) Take responsibility for all claims and liability, including compensation for any damages, which may be brought against the University by third parties who believe that their rights and interests have been infringed by the assignment.
- d) Take responsibility in the event that the institutions are found guilty of a rights infringement regarding the work subject to assignment.


6. Institutional Repository purposes and functioning.

The work shall be made available to the users so that they may use it in a fair and respectful way with regards to the copyright, according to the allowances given in the relevant legislation, and for study or research purposes, or any other legal use. With this aim in mind, the University undertakes the following duties and reserves the following powers:

- a) The University shall inform the archive users of the permitted uses; however, it shall not guarantee or take any responsibility for any other subsequent ways the work may be used by users, which are non-compliant with the legislation in force. Any subsequent use, beyond private copying, shall require the source to be cited and authorship to be recognized, as well as the guarantee not to use it to gain commercial profit or carry out any derivative works.
- b) The University shall not review the content of the works, which shall at all times fall under the exclusive responsibility of the author and it shall not be obligated to take part in lawsuits on behalf of the author in the event of any infringement of intellectual property rights deriving from storing and archiving the works. The author hereby waives any claim against the University due to any way the users may use the works that is not in keeping with the legislation in force.
- c) The University shall adopt the necessary measures to safeguard the work in the future.
- d) The University reserves the right to withdraw the work, after notifying the author, in sufficiently justified cases, or in the event of third party claims.

Madrid, on 31..... of December....., 2018

HEREBY ACCEPTS


Signed..... Karim Kadbey Nasser-Eldine.....

Reasons for requesting the restricted, closed or embargoed access to the work in the Institution's Repository

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
**DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE
FINANCIAL SECTOR** en la ETS de Ingeniería - ICAI de la Universidad
Pontificia Comillas en el
curso académico 2018/2019 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos. El Proyecto no es
plagio de otro, ni total ni parcialmente y la información que ha sido tomada
de otros documentos está debidamente referenciada.

Fdo.: Karim Kadbey Nasser-Eldine

Fecha: 31/ 12/ 2018

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Carlos Martin Orozco

Fecha: 31/ 12/ 2018



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO KARIM KADBey NASSER-ELDINE

DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL SECTOR

Autor: Karim Kadbey Nasser-Eldine
Director: Carlos Martín Orozco

Madrid
Enero 2019

Karim
Kadbey
Nasser-Eldine

**DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL
SECTOR**



DESARROLLO DE DATA LAKES Y SUS APLICACIONES EN EL SECTOR FINANCIERO

Autor: Kadbey Nasser-Eldine, Karim.

Director: Martin Orozco, Carlos.

Entidad Colaboradora: Management Solutions.

RESUMEN

El objetivo de este trabajo de fin de máster es definir la arquitectura del modelo de datos basado en un data lake implementado en una entidad financiera, así como describir el proceso que un usuario del modelo de datos de la entidad financiera debe seguir para solicitar la ingestión de datos en el data lake de dicho modelo de datos y solicitar el acceso a datos ingeridos en dicho data lake.

Para ello, este trabajo de fin de máster se dividirá en cinco partes. La primera parte explicará el concepto de Big Data. La segunda parte presentará Hadoop, sus componentes, así como algunos componentes de su ecosistema. La tercera parte explicará lo que es el Reglamento General de Protección de Datos, así como afecta a las empresas. La cuarta parte definirá el concepto de data lake, así como sus ventajas. Finalmente, la última parte de este trabajo de fin de máster definirá la arquitectura del modelo de datos basado en un data lake implementado en la entidad financiera, así como los procesos que deben seguir los usuarios de dicho modelo de datos para solicitar una ingestión de datos en el data lake dicho modelo de datos y solicitar el acceso a los datos ingeridos en dicho data lake.

Big Data

Vivimos en una sociedad dirigida por datos donde la mayoría de los servicios con los cuales interactuamos diariamente recogen todos los datos que los estamos alimentando.

Las empresas están siempre buscando formas de utilizar todos los datos que están recopilando para poder tomar las mejores decisiones para mejorar sus negocios, generar más ganancias y ser más eficiente que sus competidores.

Los datos recogidos por las empresas son en esencia datos brutos que necesitan pasar a través de un proceso con el fin de ser significativos y valiosos para dichas empresas.

El Modelo DICS (Datos, Información, Conocimiento y Sabiduría) describe cómo los datos brutos recogidos por las empresas pueden transformarse en información, y luego conocimiento y, finalmente, sabiduría.

El Modelo DICS es una pirámide jerárquica, compuesta por, en orden, los cuatro niveles siguientes:

- Datos: Los datos brutos recogidos por las empresas.
- Información: Se procesan los datos brutos recogidos por las empresas. Las relaciones entre los datos se revelan y se analizan para llegar a información.
- Conocimiento: Se procesa la información adquirida desde el nivel anterior, guiada a través de medidas específicas, con el fin de encontrar y entender patrones.
- Sabiduría: Los conocimientos adquiridos en el nivel anterior se aplican y se implementan y se determinan principios.

Después de que los datos brutos recogidos por las empresas se convierten en sabiduría, las empresas pueden aplicar esa sabiduría adquirida en resultados útiles y así tomar las decisiones adecuadas para mejorar sus negocios, generar más rentabilidad y ser más eficiente que sus competidores.

Sin embargo, los datos digitales generados hoy en día y disponibles para ser recogidos por las empresas tienen un gran volumen, aumentando drásticamente cada año, una amplia variedad de formatos (los datos generados pueden ser estructurados, semi-estructurados o no estructurados) y una alta velocidad (los datos se generan con una alta velocidad y requieren un análisis en tiempo real), haciendo muy difícil su captura, almacenamiento, procesamiento y análisis con herramientas y tecnologías tradicionales en el tiempo necesario para que los datos generados sean útiles para las empresas y por lo tanto la necesidad de algunas herramientas y tecnologías especiales llamadas herramientas y tecnologías de Big Data.

Big Data se refiere sobre todo a la recogida de datos que tienen:

- Un gran volumen,
- Una gran variedad,
- Una alta velocidad de crecimiento,

resultando muy difíciles su recogida, almacenamiento, procesamiento y análisis, utilizando herramientas y tecnologías tradicionales dentro del tiempo necesario para que estos datos sean útiles.

Herramientas y Tecnologías de Big Data

Hadoop y su ecosistema son una colección de software aptos para manejar grandes volúmenes de datos con varias formas de estructuras que permiten a la empresa recoger, almacenar, proteger, procesar, analizar y utilizar estos datos en el tiempo necesario para que estos datos sean útiles.

En esencia, los componentes principales de Hadoop son:

- HDFS (Hadoop Distributed File System): Encargado de almacenar datos en Hadoop.
- MapReduce: Encargado de procesar los datos almacenados en Hadoop.
- YARN (Yet Another Resource Negotiator): Encargado de la administración de los recursos, así como la planificación de Jobs para las diferentes aplicaciones procesando datos almacenados en Hadoop.

Algunos de los componentes del ecosistema de Hadoop son:

- Encargados de almacenar datos en Hadoop: HBase, Kudu.
- Encargados de recoger e ingerir datos en Hadoop: Sqoop, Flume.
- Encargados de proteger el acceso a los datos y los metadatos almacenados en Hadoop: Sentry.
- Encargados de procesar, analizar y visualizar los datos almacenados en Hadoop: Pig, Hive, Impala, Mahout, SOLR, Spark, HUE.

Por lo tanto, Hadoop y su ecosistema se pueden utilizar para implementar el modelo de datos de la empresa, permitiéndola recopilar, almacenar, proteger, procesar, analizar y utilizar todos los datos digitales generados en gran volumen, alta variedad y alta velocidad, en el tiempo necesario para que los datos sean de utilidad para la empresa.

Reglamento General de Protección de Datos

El mundo está convirtiéndose en cada vez más digitalizado hasta tal punto que casi cada parte de la vida de las personas puede ser digitalizada y así más y más información personal de las personas está siendo disponible para ser recogida, almacenada, procesada, analizada, utilizada y algunas veces incluso intercambiada por las empresas.

Muchas empresas tienen por lo tanto datos de información personal del cliente altamente sensible que se asocian con un riesgo significativo de ser abusados o robados.

Por lo tanto, el Reglamento General de Protección de Datos, también conocido como RGPD, fue acordado por el Parlamento y el Consejo Europeo como la principal ley que regula como las empresas deben utilizar y proteger los datos personales de los ciudadanos de la Unión Europea.

El RGPD fue adoptado oficialmente por el Parlamento y el Consejo Europeo en abril de 2016 siguiendo un período de post adopción de dos años, convirtiéndose en ejecutable el 25 de mayo de 2018.

El RGPD afecta a todas las empresas que operan en la Unión Europea o manejen datos relacionados con los ciudadanos de la Unión Europea.

Para proteger la información personal de los ciudadanos de la Unión Europea de cualquier riesgo de ser abusados o robados de las empresas que los recogen, el RGPD estableció varios requisitos para ser respetados por las empresas que manejan información personal de los ciudadanos de la Unión Europea.

Los requisitos principales del RGPD incluyen las siete áreas siguientes:

- **Consentimiento:** Las empresas necesitan obtener el consentimiento de la persona de manera clara para almacenar y usar sus datos personales especificando qué tipo de datos de carácter personal está almacenado y por qué se almacenan.
- **Notificación de infracción:** En caso de infracción de datos personales, la empresa tiene que informar a la persona afectada de dicha infracción dentro de las 72 horas después de ser consciente de ello.
- **Derecho al acceso:** Los individuos tienen derecho a obtener la confirmación de la empresa de si sus datos personales están siendo procesados.

- Derecho a ser olvidado: Los individuos pueden tener sus datos personales borrados del sistema de la empresa.
- Portabilidad de datos: Los individuos tienen derecho a obtener y reutilizar sus datos personales almacenados por la empresa para sus propios fines.
- Privacidad de diseño: Las empresas deben implementar las medidas técnicas e infraestructurales adecuadas para incluir la protección de datos a partir del inicio de sus sistemas de procesamiento de datos.
- Delegado de Protección de Datos: Las empresas que almacenan o procesan grandes volúmenes de datos personales sensibles deben designar a un Delegado de Protección de Datos.

Las empresas que no cumplen con la RGPD sufrirán consecuencias graves ya que podrían ser multadas con hasta un 4% de su facturación global anual o hasta 20 millones de Euros, el que sea mayor, por incumplimiento.

Utilizando Hadoop y su ecosistema, así como algunas de las herramientas de Cloudera (una empresa que además de ofrecer paquetes con su propia distribución del ecosistema de Hadoop, proporciona algunas herramientas que hacen más fácil para la empresa gestionar sus datos) para implementar su modelo de datos, las empresas son capaces de cumplir con los requisitos del RGPD.

Data Lake

Un data lake es sobre todo un repositorio de almacenamiento que contiene un gran volumen de datos brutos en su formato nativo. Por lo tanto, un data lake puede contener los siguientes tipos de datos diferentes:

- Datos estructurados,
- Datos semiestructurados,
- Datos no estructurados.

Mediante la implementación de un modelo de datos basado en un data lake, todos los datos brutos que podría necesitar la empresa son recogidos y almacenados en su formato original en un único repositorio dentro de la empresa.

Puesto que en un data lake todos los datos brutos recopilados por la empresa se almacenan en un repositorio único dentro de la empresa, la empresa puede procesar y analizar todos los datos juntos y así encontrar patrones, detectar tendencias, así como encontrar asociaciones entre todos sus datos, convirtiéndolos en información que les ayudará a tomar las mejores decisiones para mejorar sus negocios, generar más ingresos y ser más eficiente que sus competidores.

Puesto que un data lake tiene la capacidad de almacenar todos los tipos diferentes de datos recogidos por la empresa en su formato original, estos datos son por lo tanto rápidamente disponibles para la empresa que los recoge pudiendo acceder a ellos, procesarlos y analizarlos en tiempo real con el fin de encontrar patrones, identificar tendencias, así como encontrar asociaciones entre estos datos en tiempo real convirtiéndolos en información que les ayudará a hacer predicciones y decisiones proactivas para mejorar sus negocios, generar más ingresos y ser más eficientes que sus competidores.

Puesto que el data lake de la empresa contiene todos los datos recogidos por la empresa, es más fácil para la empresa administrar, gobernar y proteger sus datos, y por lo tanto cumplir con regulaciones tales como la Regulación General de la Protección de Datos y evitar multas.

Modelo de Datos basado en un Data Lake en una Entidad Financiera

La arquitectura del modelo de datos basada en un data lake implementado en la entidad financiera utilizando los componentes de Hadoop, algunos de los componentes del ecosistema de Hadoop, así como una de las herramientas de Cloudera es la siguiente:

- Los componentes encargados de ingerir los datos en el data lake: Sqoop, Flume.
- Los componentes encargados de almacenar los datos en el data lake: HDFS, HBase, Kudu.
- El componente encargado de indexar y buscar los datos almacenados en el data lake: SOLR.
- Los componentes encargados de la gestión y la protección de los datos en el data lake: Sentry, Cloudera Navigator.
- El componente encargado de la administración de los recursos y la planificación de Jobs al procesar los datos almacenados en el data lake: YARN.
- Los componentes encargados de procesar, analizar y visualizar los datos almacenados en el data lake: MapReduce, Pig, Hive, Impala, Mahout, Spark, HUE.

El proceso que debe seguir un usuario del modelo de datos de la entidad financiera para solicitar la ingestión de datos en el data lake de dicho modelo de datos es el siguiente:

1. Rellenar la "Data Ingestion Request";
2. Enviar la "Data Ingestion Request" al equipo de Data Management;
3. Entrega de la "Data Ingestion Request" al equipo de Data Ingestion por el equipo de Data Management;
4. Datos ingeridos por el equipo de Data Ingestion;
5. Usuario se asegura de que la ingestión se ha realizado correctamente.

El proceso que debe seguir un usuario del modelo de datos de la entidad financiera para solicitar el acceso a datos ingeridos en el data lake de la entidad financiera utilizando uno o varios de los componentes del ecosistema de Hadoop es el siguiente:

1. Rellenar la "Data Access Request";
2. Enviar la "Data Access Request" al equipo de Data Management;
3. Entrega de la "Data Access Request" al equipo de Data Access por el equipo de Data Management;
4. Acceso concedido por el equipo de Data Access;
5. Usuario se asegura de que el acceso se ha concedido correctamente.

DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL SECTOR

Author: Kadbey Nasser-Eldine, Karim.

Director: Martin Orozco, Carlos.

Collaborating Entity: Management Solutions.

ABSTRACT

The aim of this thesis is to define the architecture of the data model based on a data lake that is implemented in a financial entity as well as to describe the process a user of the financial entity's data model should follow in order to request the ingestion of data in said data model's data lake and to request the access to data ingested in said data lake.

Therefore, this thesis will be divided into five parts. The first part of this thesis will explain the concept of Big Data. The second part will present Hadoop, its components as well as some components of its ecosystem. The third part will explain what the General Data Protection Regulation is and how it affects companies. The fourth part will define the concept of data lake as well as its advantages. Finally, the last part of this thesis will define the architecture of the data model based on a data lake that is implemented in the financial entity as well as the processes to be followed by the users of said data model to request an ingestion of data in the data model's data lake and to request the access to data ingested in said data lake.

Big Data

We live in a data driven society where most of the services that we interact with on a daily basis collect all the data that we are feeding them.

Companies are always looking for ways to use all the data that they are collecting in order to be able to make the best decisions to improve their businesses, generate more profit and be more efficient than their competitors.

The data collected by companies are in essence raw data that need to go through a process in order to be meaningful and valuable data for the company.

The DIKW Model (Data, Information, Knowledge and Wisdom Model) describes how the raw data collected by companies can be transformed into information then knowledge and finally wisdom.

The DIKW Model is a hierarchical pyramid, composed of, in order, the following four levels:

- Data: The raw data collected by companies.
- Information: The raw data collected by companies is processed. Relationships between the data are revealed and analyzed in order to reach information.
- Knowledge: The information acquired from the previous level is processed, guided through specific measures, in order to find and understand patterns.
- Wisdom: The knowledge acquired from the previous level is applied and implemented and principles are determined.

After the raw data collected by the companies are converted into wisdom, the companies can apply that wisdom gained into actionable results and thus make the appropriate decisions to improve their businesses, generate more profit and be more efficient than their competitors.

However, the digital data being generated nowadays and available to be collected by companies have a large volume, drastically increasing every year, a wide variety of format (the data generated could be structured, semi-structured or unstructured) and a high velocity (the data is generated with a high-speed and requires a real-time analysis), making it very difficult to be captured, stored, processed and analyzed using traditional technologies and tools within the time necessary for the data generated to be useful for the companies and hence the need for some special technologies and tools called Big Data technologies and tools.

Big Data refers, fore and foremost, to the collection of data which:

- Large volume,
- High variety,
- High speed of growth,

makes it very difficult to be collected, stored, processed and analyzed, using traditional technologies and tools within the time necessary for that data to be useful.

Big Data Technologies and Tools

Hadoop and its ecosystem are a collection of software that are suited to handle high volumes of data with various forms of structures allowing the company to collect, store, secure, process, analyze and use that data within the time necessary for that data to be useful.

At its core, Hadoop's main components are:

- HDFS (Hadoop Distributed File System): In charge of storing data in Hadoop.
- MapReduce: In charge of processing data stored in Hadoop.
- YARN (Yet Another Resource Negotiator): In charge of managing the resources as well as scheduling jobs for the different applications or workloads processing data stored in Hadoop.

Some of Hadoop's ecosystem components are:

- In charge of storing data in Hadoop: HBase, Kudu.
- In charge of collecting and ingesting data into Hadoop: Sqoop, Flume.
- In charge of securing the access to the data and metadata stored in Hadoop: Sentry.

- In charge of processing, analyzing and visualizing the data stored in Hadoop: Pig, Hive, Impala, Mahout, SOLR, Spark, HUE.

Therefore, Hadoop and its ecosystem can be used to implement the data model of the company allowing it to collect, store, secure, process, analyze and use all the digital data being generated in large volume, high variety and high velocity, within the time necessary for said data to be useful for the company.

General Data Protection Regulation

The world is becoming increasingly digitized to the point where nearly every part of people's life can be digitized and thus more and more of people's personal information is available to be collected, stored, processed, analyzed, used and sometimes even traded by companies.

Many companies are thus holding highly sensitive customer personal information data which is associated with a significant risk if abused or stolen.

Therefore, the General Data Protection Regulation, also known as GDPR, was agreed upon by the European Parliament and Council as the main law regulating the way companies should use and protect European Union citizens' personal data.

The GDPR was officially adopted by the European Parliament and Council in April 2016 but followed a two years post adoption period and became enforceable on 25 May 2018.

The GDPR affects all companies that either operate in the European Union or handle data related to citizens of the European Union.

In order to protect European Union citizens' personal information from any risk of being abused or stolen from the companies that collect them, the GDPR has established several requirements to be respected by companies handling European Union citizens' personal information.

The GDPR's major requirements include the seven following areas:

- **Consent:** Companies need to obtain the individual's consent in an upfront way to store and use his personal information specifying what type of personal data is being stored and why it is being stored.
- **Breach Notification:** In the event of a personal data breach, the company must inform the individual concerned of said breach within 72 hours after becoming aware of it.
- **Right to Access:** Individuals have the right to obtain confirmation from the company of whether their personal data are being processed.
- **Right to be forgotten:** Individuals can have their personal data erased from the company's system.
- **Data Portability:** Individuals have the right to obtain and reuse their personal data stored by the company for their own purposes.
- **Privacy by Design:** Companies should implement the appropriate technical and infrastructural measures to include data protection from the beginning of their data processing systems.

- Data Protection Officers: Companies that store or process large volumes of sensitive personal data must appoint a Data Protection Officer.

Companies that do not comply with the GDPR will suffer serious consequences as they could be fined up to 4% of their total worldwide annual turnover of the preceding financial year or up to 20 Million Euros, whichever one is higher, for non-compliance.

By using Hadoop and its ecosystem as well as some of Cloudera's ready tools (an enterprise that in addition of providing packages with its own distribution of Hadoop's ecosystem, provides some ready tools that makes it easier for the company to manage their data) to implement their data model, companies are able to comply with the GDPR.

Data Lake

A data lake is foremost a storage repository that holds a large volume of raw data in its native format. Therefore, a data lake can contain all the following different types of data:

- Structured data,
- Semi-structured data,
- Unstructured data.

By implementing a data model based on a data lake, all the raw data that the company might need is collected and stored in its original format in a single repository within the company.

Since in a data lake all the raw data collected by the company are stored in a single repository within the company, the company can process and analyze all their data together and thus find new patterns, spot new trends as well as find new associations between all their data turning it into information that will help them make the best decisions to improve their businesses, generate more revenue and be more efficient than their competitors.

Since a data lake has the ability to store all the different types of data collected by the company in its original format, it is thereby quickly available for the company to access it, process it and analyze it in real time in order to find patterns, spot trends as well as find associations between that data in real time turning it into information that will let the company make predictions and proactive decisions to improve their businesses, generate more revenue and be more efficient than their competitors.

Since the company's data lake contains all the data collected by the company, it is easier for the company to manage, govern and secure their data and thus comply with regulations such as the General Data Protection Regulation to avoid penalties.

Data Model based on a Data Lake in a Financial Entity

The architecture of the data model based on a data lake implemented in the financial entity using Hadoop's components, some of Hadoop's ecosystem components as well as one of Cloudera's ready tools is the following:

- The components in charge of ingesting the data into the data lake: Sqoop, Flume.
- The components in charge of storing the data in the data lake: HDFS, HBase, Kudu.
- The component in charge of indexing and searching the data stored in the data lake: SOLR.
- The components in charge of managing and securing the data stored in the data lake: Sentry, Cloudera Navigator.
- The component in charge of managing the resources and scheduling jobs when processing the data stored in the data lake: YARN.
- The components in charge of processing, analyzing and visualizing the data stored in the data lake: MapReduce, Pig, Hive, Impala, Mahout, Spark, HUE.

The process a user of the financial entity's data model should follow in order to request an ingestion of data in said data model's data lake is the following:

1. Fill in the "Data Ingestion Request";
2. Submit the "Data Ingestion Request" to the Data Management Team;
3. Submission of the "Data Ingestion Request" to the Data Ingestion Team by the Data Management Team;
4. Data Ingested by the Data Ingestion Team;
5. User makes sure that the ingestion has been properly done.

The process a user of the financial entity's data model should follow in order to request the access to some data ingested in the financial entity's data lake using one or several of Hadoop's ecosystem components should be the following:

1. Fill in the "Data Access Request";
2. Submit the "Data Access Request" to the Data Management Team;
3. Submission of the "Data Access Request" to the Data Access Team by the Data Management Team;
4. Access granted by the Data Access Team;
5. User makes sure that the access has been correctly granted.



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
MÀSTER EN INGENIERÍA INDUSTRIAL

DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL SECTOR

Autor: Karim Kadbey Nasser-Eldine
Director: Carlos Martín Orozco

Madrid
Enero 2019

Karim
Kadbey
Nasser-Eldine

**DEVELOPMENT OF DATA LAKES AND THEIR APPLICATION IN THE FINANCIAL
SECTOR**



Acknowledgment

I would like to express my sincere gratitude to every person who contributed to the success of my thesis and helped me in finalizing it.

Firstly, I would like to thank Management Solutions, and more specifically Carlos Martín Orozco, Jiajie Yan Wu and Sergio González Madroñal whom without their continuous help, supervision and support this project couldn't have been done.

I would also like to express my gratitude to Universidad Pontificia Comillas (ICAI) who, through their highly qualified and professional academic expertise, and through their support during my years of study, contributed to improve and develop my academic skills.

Finally, I would like to thank my parents who gave me constant support and encouragement which helped me in carrying out this thesis.



TABLE OF CONTENTS

<i>Table of Contents</i>	<i>1</i>
Chapter 1 Introduction	5
1.1 State of the Art	5
1.2 Possible Solutions	7
1.3 Motivation	9
1.4 Objectives of the Project	10
1.5 Work Methodology	10
Chapter 2 Big Data	13
2.1 What is Data?	13
2.2 What is Big Data?	13
2.2.1 What is a Dataset?	14
2.2.2 What is a Relational Database?	15
2.2.3 What is Big Data?	16
2.2.4 The Sizes in Big Data.....	17
2.3 The Value of Big Data	18
2.3.1 DIKW Model	19
2.3.2 Examples	27
2.4 Big Data Types	28
2.4.1 Structured Data	28
2.4.2 Unstructured Data	30
2.4.3 Semi-Structured Data	31
2.5 The “V” Model	31
2.5.1 Volume.....	32
2.5.2 Velocity.....	33
2.5.3 Variety.....	34
2.5.4 Veracity	35



Chapter 3	<i>Big Data Technologies And Tools</i>	37
3.1	Hadoop's Core Components	37
3.1.1	Open-Source Software	39
3.1.2	Job	39
3.1.3	Hadoop Cluster	39
3.1.4	HDFS (Hadoop Distributed File System)	40
3.1.5	MapReduce	43
3.1.6	YARN (Yet Another Resource Negotiator)	44
3.2	Hadoop's Ecosystem	44
3.2.1	Store	45
3.2.2	Integrate	47
3.2.3	Security	48
3.2.4	Process, Analyze and Serve	49
Chapter 4	<i>General Data Protection Regulation</i>	53
4.1	Definition and Overview	54
4.2	Personal Information under the General Data Protection Regulation	55
4.3	General Data Protection Regulation's Requirements	56
4.3.1	Consent	57
4.3.2	Breach Notification	57
4.3.3	Right to Access	57
4.3.4	Right to be forgotten	57
4.3.5	Data Portability	58
4.3.6	Privacy by Design	58
4.3.7	Data Protection Officers.....	58
4.4	Impact of the General Data Protection Regulation on Businesses	59
4.5	General Data Protection Regulation, Hadoop And Cloudera	60
Chapter 5	<i>Data Lake</i>	63
5.1	Definition, Overview and Advantages	64
5.2	Examples	66
5.2.1	Example 1	66
5.2.2	Example 2	67
5.2.3	Example 3	68



Chapter 6	<i>Data Model based on a Data Lake in a Financial Entity</i>	69
6.1	Architecture of the Data Model based on a Data Lake	71
6.1.1	Ingestion	72
6.1.2	Store	72
6.1.3	Index & Search	74
6.1.4	Manage & Secure	74
6.1.5	Resource Management	75
6.1.6	Process & Analyze	76
6.2	Data Ingestion and Data Access Requests	77
6.2.1	Data Ingestion Request	77
6.2.2	Data Access Request	81
6.3	Data Ingestion and Data Access Problems	85
6.3.1	Data Ingestion Error	85
6.3.2	Data Access Error	88
Chapter 7	<i>Conclusion</i>	93
Chapter 8	<i>Bibliography</i>	95



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Table of Contents



Chapter 1 INTRODUCTION

We live in a data driven society where most of the services that we interact with on a daily basis collect all the data that we are feeding them.

Companies all over the world, including financial entities, are always looking for ways to store, process and analyze all the data that they are collecting in order to find patterns, trends and associations between said data and turn it into information that will help them make the best decisions to improve their businesses.

The aim of this thesis is to define the architecture of the data model based on a data lake that is implemented in a financial entity as well as to describe the process a user of the financial entity's data model should follow in order to request the ingestion of data in said data model's data lake and to request the access to data ingested in said data lake.

1.1 STATE OF THE ART

The mere collection of data by companies is not enough to help them take decisions to improve their businesses.

As a matter of fact, all the data collected by companies are in essence raw data that needs to be processed and analyzed in order to find patterns, spot trends as well as find associations between said data to turn it into information that will help companies make the best decisions in order to improve their businesses, generate more revenue and be more efficient than their competitors.

In other words, the raw data collected by companies needs to be processed, analyzed and understood in order for said data to be meaningful and valuable for the companies collecting it.

However, the world is becoming increasingly digitized and thus the digital data being generated and available to be collected by companies have:

- A large volume that is drastically increasing every year;
- A wide variety, that is to say, the data being generated has a wide range of format (in other words, the data generated can be structured data, semi-structured data or unstructured data);



- A high velocity, that is to say, the data is being generated with a high speed and requires a real-time analysis;

all of which makes it very difficult for companies to capture that data, store it, process it and analyze it using traditional technologies and tools within the time necessary for that data to be useful for them.

Therefore, the need for special technologies and tools called Big Data technologies and tools is necessary in order for the companies to be able to collect, store, process and analyze the digital data being generated and available to be collected.

On the other hand, if all the data collected by the company is not stored in the same place within the company, it will make it very difficult for the company to process and analyze its data stored in different places together, making it very difficult to find patterns, spot trends as well as find associations between said data.

Therefore, the need for a single storage repository within the company is necessary to be able to store all the data collected.

Due to the large volume of data being generated and available to be collected by the company as well as its high velocity and wide variety, this single storage repository needs to be highly scalable, able to store all the different types of data generated as well as able to store all the data collected in its original format allowing the company to access it, analyze it and process it in real-time.

The world is becoming increasingly digitized to the point where nearly every part of people's life can be digitized and thus more and more of people's personal information is available to be collected, stored, processed, analyzed, used and sometimes even traded by companies.

Therefore, many companies hold highly sensitive customer personal information data which is associated with a significant risk if abused or stolen.

The General Data Protection Regulation, also known as GDPR, was agreed upon by the European Parliament and Council as the main law regulating the way companies should use and protect European Union citizens' personal data.

The General Data Protection Regulation was officially adopted by the European Parliament and Council in April 2016 but followed a two years post adoption period and became enforceable on 25 May 2018.

The General Data Protection Regulation affects all companies that either operate in the European Union or handle data related to citizens of the European Union.

In order to protect European Union citizens' personal information from any risk of being abused or stolen from the companies that collect them, the General Data Protection Regulation has established several requirements to be respected by companies handling European Union citizens' personal information.



The GDPR's major requirements include the seven following areas:

- Consent: Companies need to obtain the individual's consent in an upfront way to store and use his personal information specifying what type of personal data is being stored and why it is being stored.
- Breach Notification: In the event of a personal data breach, the company must inform the individual concerned of said breach within 72 hours after becoming aware of it.
- Right to Access: Individuals have the right to obtain confirmation from the company of whether their personal data are being processed.
- Right to be forgotten: Individuals can have their personal data erased from the company's system.
- Data Portability: Individuals have the right to obtain and reuse their personal data stored by the company for their own purposes.
- Privacy by Design: Companies should implement the appropriate technical and infrastructural measures to include data protection from the beginning of their data processing systems.
- Data Protection Officers: Companies that store or process large volumes of sensitive personal data must appoint a Data Protection Officer.

Companies that do not comply with the GDPR will suffer serious consequences as they could be fined up to 4% of their total worldwide annual turnover of the preceding financial year or up to 20 Million Euros, whichever one is higher, for non-compliance.

Therefore, companies need to manage, govern and secure their data in order to comply with the General Data Protection Regulation to avoid penalties.

1.2 POSSIBLE SOLUTIONS

Hadoop and its ecosystem are a collection of software that are suited to handle high volumes of data with various forms of structures allowing the company to collect, store, secure, process, analyze and use that data.

At its core, Hadoop's main components are:

- HDFS (Hadoop Distributed File System): In charge of storing data in Hadoop;
- MapReduce: In charge of processing data stored in Hadoop;
- YARN (Yet Another Resource Negotiator): In charge of managing the resources as well as scheduling jobs for the different applications or workloads processing data stored in Hadoop.



Some of Hadoop's ecosystem components are:

- In charge of storing data in Hadoop:
 - o HBase,
 - o Kudu.
- In charge of collecting and ingesting data into Hadoop:
 - o Sqoop,
 - o Flume.
- In charge of securing the access to the data and metadata stored in Hadoop:
 - o Sentry.
- In charge of processing, analyzing and visualizing the data stored in Hadoop:
 - o Pig,
 - o Hive,
 - o Impala,
 - o Mahout,
 - o SOLR,
 - o Spark,
 - o HUE.

Hadoop and its ecosystem can thus be used to implement the data model of the company allowing it to collect, store, secure, process, analyze and use all the digital data that is being generated in large volume, high variety and high velocity within the time necessary for said data to be useful for the company.

A data lake is fore and foremost a storage repository that holds a large volume of raw data in its native format. Therefore, a data lake can contain all the following different types of data:

- Structured data,
- Semi-structured data,
- Unstructured data.

Thus, all the raw data that the company might need is collected and stored in its original format in a single repository within the company.

By implementing a data model based on a data lake, the company is hence able to:

- Store all the raw data collected in a single repository within the company.
The company is thus able to process and analyze all its data together and find patterns, spot trends as well as find associations between all its data.
- Store all the raw data collected in its original format in one single repository.



Since the data collected by the company is stored in its original format, it is thereby quickly available for the company to access it, process it and analyze it in real time and thus find patterns, spot trends as well as find associations between that data in real time.

By implementing the data model based on a data lake using Hadoop and its ecosystem, the company is able to:

- Have a scalable storage repository as Hadoop and its ecosystem provides scalable storage infrastructure;
- Store all the data collected by the company, whether it is structured data, semi-structured data or unstructured data, in one single repository.

Since the company's data lake contains all the data it collected, it is easier for the company to manage, govern and secure its data and thus comply with regulations such as the General Data Protection Regulation to avoid penalties.

Moreover, by using Hadoop and its ecosystem as well as some of Cloudera's ready tools (an enterprise that in addition of providing packages with its own distribution of Hadoop's ecosystem, provides some ready tools that makes it easier for the company to manage their data) to implement its data model based on a data lake, the company is able to comply with the General Data Protection Regulation.

1.3 MOTIVATION

The aim of this thesis is to define the architecture of the data model based on a data lake that is implemented in a financial entity as well as to describe the process a user of the financial entity's data model should follow in order to request the ingestion of data in said data model's data lake and to request the access to data ingested in said data lake.

By implementing a data model based on a data lake using Hadoop's components, some of Hadoop's ecosystem components as well as some of Cloudera's ready tools, the financial entity is able to:

- Collect, store, secure, process and analyze all the data that it might need from the data generated and available to be collected with a large volume, high velocity and wide variety;
- Store all the collected data, whether it is structured data, semi-structured data or unstructured data, in its original format in one single scalable storage repository;
- Comply with the requirements of the General Data Protection Regulation and avoid penalties.



By establishing a process that users of the financial entity's data model must follow in order to request an ingestion of data in said data model's data lake, the financial entity will make sure that its data lake contains all the data the users of its data model need.

By establishing a process that users of the financial entity's data model must follow in order to request an access to some data ingested in said data model's data lake, the financial entity will make sure that the data ingested in its data lake is only accessible by the users that need it.

1.4 OBJECTIVES OF THE PROJECT

The main objective of this thesis is to define the architecture of the data model based on a data lake that is implemented in a financial entity as well as to describe the process a user of the financial entity's data model should follow in order to request the ingestion of data in said data model's data lake and to request the access to data ingested in said data lake.

In order to reach this objective, the following specific goals had to be achieved:

- Study the different Big Data tools and technologies that allow the financial entity to collect, store, process, analyze, secure, use and visualize the large amount of digital data being generated at a high velocity and with a wide variety of format;
- Identify how the GDPR can affect the way a financial entity, that handles European Union citizens' personal data, collects, stores, processes, analyzes, uses and visualizes that data;
- Define the architecture of the data model based on a data lake that is implemented in the financial entity;
- Describe the process a user of the financial entity's data model should follow in order to request the ingestion of data in said data model's data lake;
- Describe the process a user of the financial entity's data model should follow in order to request the access to some data ingested in said data model's data lake.

1.5 WORK METHODOLOGY

In order to reach the goals specified in the previous sub-section, the following tasks were carried out:

- Become familiarized with the concept of Big Data;



- Become familiarized with the concept of Data Lake;
- Study the effect of the GDPR on companies (including financial entities);
- Study the functionality of the different Big Data tools and technologies;
- Define the architecture of the data model based on a data lake that is implemented in a financial entity;
- Define the process a user of the financial entity's data model should follow in order to request the ingestion of data in said data model's data lake;
- Define the process a user of the financial entity's data model should follow in order to request the access to some data ingested in said data model's data lake.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Introduction



Chapter 2 BIG DATA

Big Data is nowadays everywhere. Most people get confused when they hear the term “Big Data” and start to wonder why this term is finding its way into businesses worldwide.

This first section of this thesis is dedicated to give a brief explanation of Big Data and thus assumes no prior knowledge of Big Data whatsoever.

At first, a definition of Data and Big Data will be given. These definitions will be followed by some explanations on why companies should use Big Data in their businesses. The three existing types of Big Data will be defined afterwards. Last but not least, the “V” model of Big Data will be presented.

2.1 WHAT IS DATA?

The first question that comes to mind while reading this section is “What is Data”? Or more specifically “What does the author of this thesis mean when saying Data?”

More than one definition of the term “Data” is available. However, Data is basically all the information (such as facts, numbers, measurements, statistics, etc.) collected from the past, to be examined, considered and used in the future to make better decisions.

Since we live in a digital world, a more relevant definition of data would be (especially in this thesis), all the digital information collected and stored in a device (such as a computer) to be used (transmitted or processed) in the future.

2.2 WHAT IS BIG DATA?

In order to understand correctly the definition of the term “Big Data”, this subsection will be divided into four parts.

In the first part, the definition of a dataset will be given. The second part will provide the definition of a database, or more precisely, of a relational database. The third part will present the definition of the term “Big Data”. In the fourth part, different units of measurement of data used in Big Data will be introduced.



2.2.1 WHAT IS A DATASET?

In the world of Big Data, the term “Dataset” is often mentioned and heard, but the question here is: What is exactly a Dataset?

A dataset is simply a collection of related data with a specific purpose that might be accessed in combination, managed as a whole entity and thus manipulated as a unit by a device (such as a computer).

To better understand the meaning of a dataset, the following examples of datasets generated from different sources are given:

- Social Media:

- All the posts of a social network posted in Spain for a period of one week constitute one dataset.

All these posts represent one collection of data with one focal point and thus represent one dataset.

This dataset can be analyzed with the purpose of answering some questions, for example the following questions: What were people talking about? What were people interested in? What products or services are being mentioned the most?

After analyzing this dataset, some results are generated. These results could be compared afterwards with the results of analyzing other datasets.

- Weather:

- All the data collected by a weather forecasting channel for a week constitute a dataset.

The weather forecasting channel might want to collect and analyze all the data of several weeks, each week representing one dataset, and compare the different datasets with each other.

- Supermarket:

- All the sales for one month would be a collection of data about the supermarket’s sales for this month and thus constitute a dataset.

The supermarket might want to collect and analyze all the sales of several months, each month representing one dataset, and compare the different datasets with each other.

- Emails:

- All the emails sent by someone during a specific month constitute a dataset.

This dataset can be analyzed with the purpose of answering some questions, for example the following questions: How many emails



were sent out of the country from this person? How many emails did this person sent to another specific person?

After analyzing this dataset, some results are generated. These results could be compared afterwards with the results of analyzing other datasets.

- All the emails sent by everyone in a company during a specific month constitute a dataset.

This dataset can be analyzed with the purpose of answering some questions, for example the following questions: How many emails were sent to clients? How many emails were sent to a specific client?

After analyzing this dataset, some results are generated. These results could be compared afterwards with the results of analyzing other datasets.

- All the emails sent form a country during a specific month constitute a dataset.

This dataset can be analyzed with the purpose of answering some questions, for example the following questions: How many emails were sent to other countries? How many emails were sent to other continents? How many emails were sent to another specific country?

After analyzing this dataset, some results are generated. These results could be compared afterwards with the results of analyzing other datasets.

2.2.2 WHAT IS A RELATIONAL DATABASE?

The term “Database”, or more precisely the term “Relational Database”, is a fundamental term in the digital word of Big Data and can thus be mentioned and heard a lot. So, the question here is “What is a Database?” and more precisely “What is a Relational Database?”

A Database is for and foremost an organized collection of data that can be easily accessed, managed and updated. That is to say, a Database, is a place where data is stored and kept in an organized form.

In a Database, data is organized and displayed in defined columns, rows and tables. Most Databases contain several tables, each table may include several fields and rows.

A Relational Database, is a Database where a well-defined relationship between the Database’ tables is established. That is to say, Database tables communicate



with each other and share information, and thus facilitate the access, organization and reporting of the data.

2.2.3 WHAT IS BIG DATA?

The fundamental definition of this sub-section is the definition of the term “Big Data”, and thus the question here is “What is Big Data?”

For and foremost, Big Data, refers to the collection of data, that is to say a dataset, or combinations of datasets which:

- Large volume,
- High variability,
- High speed of growth,

make it very difficult to be captured, managed, processed and analyzed, using traditional technologies and tools (such as relational databases), within the time necessary for them to be useful.

However, even if Big Data is for and foremost a collection of a large volume of data with high variability and high speed of growth, it is also the following things:

- Big Data is a Technology: Since it is very difficult to capture, manage, process and analyze Big Data using traditional technologies and tools, some special technologies and tools will be needed (Big Data technologies and tools) such as Hadoop (Hadoop and Hadoop’s ecosystem will be explained in the second section of this thesis);
- Big Data is a Strategy: Big Data is definitely a strategy as a company might develop a Big Data centric approach, which means that the company is going to focus their business on using Big Data in order to help create greater efficiency, better business, more revenue, new products, better products, new services, better services, etc.;
- Big Data is an Industry: There are companies, such as Cloudera, that provide Big Data technologies to their customers. They provide Big Data services to their customers in order to help them develop their Big Data strategies and use Big Data technologies.

All these interactions form an industry and thus Big Data is, without doubt, an industry;

- Big Data is a Process: In order to get the knowledge needed from the large amount of data, the large amount of data should go through a process using Big Data technologies.

Big Data technologies should be used to collect, process, analyze and understand the large amount of data.



Big Data is thus definitely a process, as it could also refer to the process of understanding the large amount of data.

What is Big Data?

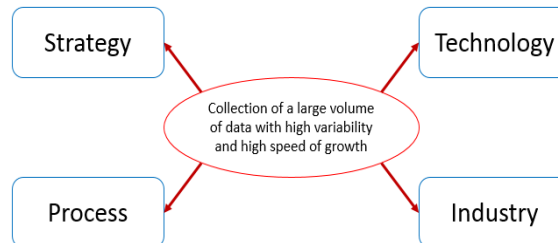


Figure 1: What is Big Data?

2.2.4 THE SIZES IN BIG DATA

Big Data represents a large volume of data. However, a question that a lot of people might ask themselves is how large, in numbers, should be the volume of data to be considered Big Data, which is equivalent to the question “How Big is Big?”

Well, the term Big in Big Data is a relative term and there is no numeric definition for the term Big in Big Data.

A dataset is considered Big Data if it is too large in size to be evaluated by traditional tools and technologies (such as relational databases). However, the size in numbers depends on the data forming the dataset and on what should be done with said data.

Nonetheless, in general scale, the sizes that are most commonly used in Big Data are the following:

- Petabyte: A Petabyte is equal to 1000 Terabytes.

Even though not many people have heard of the Petabyte unit, usually most people have heard of the Terabyte unit as it is the new standard for laptop disc space.

An example to see how large a dataset could be is the following:

- o As of January 2013, Facebook users had uploaded an estimated of 357 Petabytes of storage.
- Exabyte: An Exabyte is equal to 1000 Petabytes.
The following example is to see how big an Exabyte is:



- In 2006, the combined space of all computer hard drives in the world was approximately 160 Exabytes.
- Zettabyte: A Zettabyte is equal to 1000 Exabytes.

The following example is to see how big a Zettabyte is:

- In 2003, Mark Liberman calculated that the storage required for all human speech ever spoken was 42 Zettabytes if digitalized as 16 kHz 16-bit audio.

1 Kilobyte	=	1000 Byte
1 Megabyte	=	1000 Kilobyte
1 Gigabyte	=	1000 Megabyte
1 Terabyte	=	1000 Gigabyte
1 Petabyte	=	1000 Terabyte
1 Exabyte	=	1000 Petabyte
1 Zettabyte	=	1000 Exabyte
1 Yottabyte	=	1000 Zettabyte

Table 1: Sizes in Big Data

2.3 THE VALUE OF BIG DATA

The following important question should be answered at this point: “What is the Value of Big Data?”. This question is equivalent to the question “Why and How should companies use all the data that they are collecting to improve their businesses?”

We live today in a data driven society. That is to say, most of the services we interact with on a daily basis, collect all the data that we are feeding them to be able to make better future decisions in order to improve their businesses.

In other words, companies are always looking for ways to use and analyze all the data that they are collecting in order to make the best future decisions for their businesses and be more efficient than their competitors; for example, provide the best user experience, develop and provide the best new product, establish more efficient methods of production and manufacturing, make the best decisions on what product to deliver, make the best decisions on what experience to deliver, etc.

In order for companies to be able to effectively use all the data that they are collecting in order to make better future business decisions, or more precisely, in order to make the best future business decisions, they have to understand the data that they are collecting, understand how to use all the data that they are collecting



and ask themselves how can their businesses get better as a result of their knowledge on all the data that they are collecting.

The need to look deeply for companies into understanding the massive amounts of data that they are collecting all day, every day, is the nature of Big Data. This is a challenge to a lot of companies as they assume that the only thing they need in order to make the best future business decisions is to collect data (and the more data they collect, the better) and to use Big Data technologies. This is not true, since data is only raw material, and all the data they collect should go through a process before having value and before being used to make better future business decisions.

If companies successfully analyze the large amount of data they are collecting, it will show them patterns, trends and associations between the data that they are collecting. This in return will help the companies make the best future business decisions and create new company goals. Companies can thus, based on the results from all the data they collected, analyzed and understood, make the best future business decisions and create new goals. The use of companies of the results of all the data they collected, analyzed and understood, to make better future business decisions and create new goals is the importance of Big Data and is the reason why Big Data is so powerful.

In order to better understand the process that the data collected by a company should go through in order to help the company make the best future business decision, the DIKW (Data-Information-Knowledge-Wisdom) model will be presented followed by some examples on how the use of Big Data and the DIKW model could improve the business of a company.

2.3.1 DIKW MODEL

The data collected by a company is in essence raw material that needs to be processed in order to make it meaningful and valuable for the company.

The DIKW Model (Data, Information, Knowledge and Wisdom Model), also known as DIKW Hierarchy or DIKW Pyramid, describes how data is transformed into wisdom. More precisely, the model describes how the data collected by a company can be processed and transformed into information, then knowledge, and finally wisdom.

The term “DIKW” stands for:

- “D” for Data,
- “I” for Information,
- “K” for Knowledge,
- “W” for Wisdom.



The DIKW model represents the relationships between data, information, knowledge and wisdom and is a hierarchical pyramid, that is to say, the pyramid is rigidly built up by the four following blocks:

- The first block is Data;
- The second block is Information;
- The third block is Knowledge;
- The fourth and final block is Wisdom.

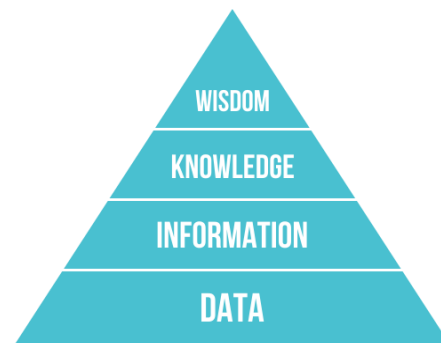


Figure 2: DIKW Pyramid

Starting from the first block of the pyramid, corresponding to the collection of data, the higher we go up in the pyramid, the more questions we answer about the initial data collected and the more value we add to the initial data collected. That is to say, the higher we go up in the pyramid, the more our data will have meaning and context, and thus the more knowledge and insights we will get out of it. When we get to the last block of the pyramid, we will have turned the knowledge and insights we got out of our data into learning experiences that will guide our future decisions and actions.

The DIKW model can be seen from two different angles:

- Contextual concept,
- Understanding perspective.

The DIKW model can also be viewed in terms of time.

In context, the DIKW model can be seen as a hierarchical model with the four following phases:

- Phase 1 (Data): Phase of gathering parts;
- Phase 2 (Information): Connection of raw data parts;
- Phase 3 (Knowledge): Formation of a whole meaningful contents;
- Phase 4 (Wisdom): Conceptualize and joining of whole meaningful contents.

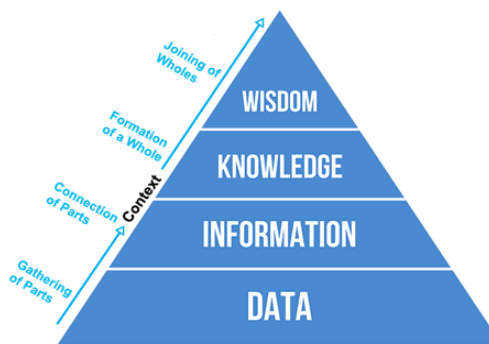


Figure 3: DIKW Pyramid viewed in a Contextual Concept

In understanding, the DIKW model can be seen as a process with the four following steps:

- Step 1: Researching & Absorbing;
- Step 2: Doing;
- Step 3: Interacting;
- Step 4: Reflecting.

Even though these steps connect the DIKW pyramid, no step is linked to a particular block of the DIKW model.

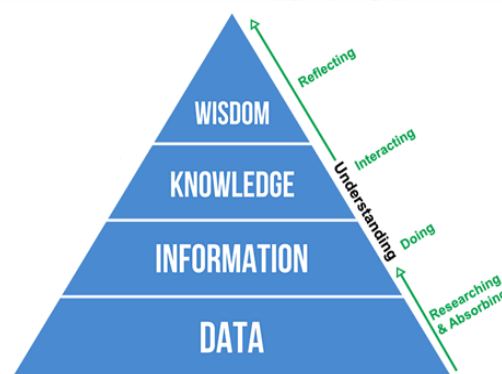


Figure 4: DIKW Pyramid viewed in an Understanding Perspective

In terms of time, the DIKW model can be seen divided into two parts:

- The Past (Data, Information, Knowledge),
- The Future (Wisdom).

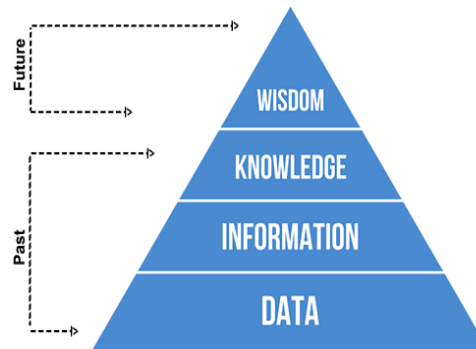


Figure 5: DIKW Pyramid represented in terms of Time

Each individual block of the DIKW model will be now explained separately, starting from the first one (Data) till the last one (Wisdom).

2.3.1.1 “Data” level of DIKW Model

The first level of the DIKW pyramid is Data.

The collection of raw data is the starting point to be able to get a meaningful result that will help companies take the best future decision for their businesses in the end.

The collection of raw data is the collection of all the facts, companies are able to collect, in a raw form such as measurements, records, logging, etc.

The raw data is collected in mass and may thus include both useful and not useful contents. Therefore, the raw data alone do not present any meaningful result, nor answer any question, nor draw any conclusion.

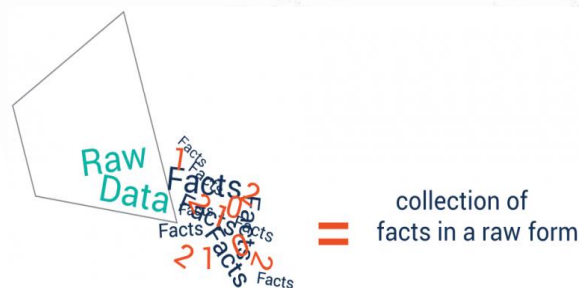


Figure 6: Collection of Raw Data



To be able to better understand the transformation of data into meaningful results using the DIKW model, the following example will be taken and discussed in each of the following levels (Information, Knowledge and Wisdom) of the DIKW pyramid:

A Spanish airline company collects, as raw data, the number of users that visit its website on a daily basis.

As raw data, the company collected that, on a daily basis, 700 users visit its website.

2.3.1.2 “Information” level of DIKW Model

The second level of the DIKW pyramid is Information.

In the Information level of the DIKW model, the raw data collected is processed and analyzed.

Data processing is done in order to give a meaning to the raw data collected in the first level of the DIKW model. Data processing may include several different operations such as:

- Aggregation: The combination of different sets of data and revelation of relationships in the collected data.
- Validation: The verification that the collected data is relevant and accurate.

By processing the data collected, and thus defining relational connections between the data collected, the data is given a meaning, which makes the data easier to be measured, visualized and analyzed, by the company, for a specific purpose.

After the collected data has been processed in the Information level of the DIKW pyramid, it will be analyzed in order to find answers to the following questions:

- Who?
- What?
- When?
- Where?

The answers of these questions will extract valuable information from the data and thus make the data more useful for the company.

In other words, the output of this second level of the DIKW pyramid, is that, the raw data collected in the first level of the DIKW pyramid became information that answers the questions Who, What, When and Where.

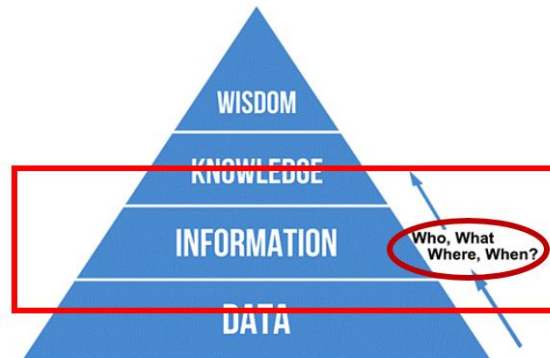


Figure 7: Questions of the Information level

Continuing with the example of the Spanish airline company:

The generic raw data “700 users visit the website on a daily basis” collected by the company, is processed and analyzed in the Information step of the DIKW model to get, for example, the following answers:

- 500 users search for national flights,
- 200 users search for international flights,
- 40% of the users are in the age group of 18-20 years,
- 30% of the users are in the age group of 25-35 years,
- 60% of the users visit the website between 8:00pm and 10:00pm.

2.3.1.3 “Knowledge” level of DIKW Model

The third level of the DIKW pyramid is Knowledge.

Information is turned into knowledge when it is not only seen as a description of collected facts but when the ways to apply it to achieve certain established goals is understood.

The Knowledge level of the DIKW model aims to try to find the answer to the “How” question. In this level, specific measures are presented and the information acquired from the second level of the DIKW model is used to answer that question based on these measures.

In other words, the information acquired from the Information level of the DIKW model can be guided through specific measures in order to answer the How question of the Knowledge level of the DIKW model. Information is thus converted into Knowledge as output of this third level of the DIKW pyramid.

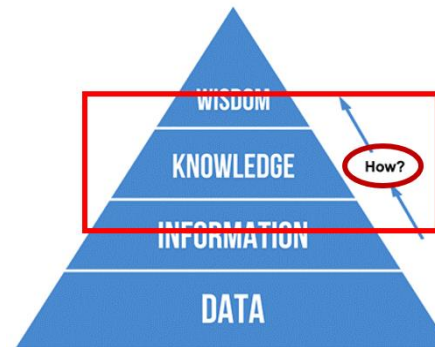


Figure 8: Question of the Knowledge level

Continuing with the example of the Spanish airline company:

A How question that the Spanish airline company might be interested in finding the answer to is the following:

- How do young users in the age group of 18-25 years use our website for national flights?

2.3.1.4 “Wisdom” level of DIKW Model

The fourth and last level of the DIKW pyramid is Wisdom.

Wisdom is reached when knowledge and insights gained from information is used to take proactive decisions or improvement decisions. That is to say, wisdom is knowledge applied in actions.

In the Wisdom level of the DIKW model, the knowledge acquired in the third level of the DIKW pyramid is applied and implemented.

The Wisdom level of the DIKW model is the last level of the DIKW pyramid and answers the “Why” question.

The DIKW model is a hierarchical pyramid in which the data collected by the company is converted into actionable results based on wisdom.

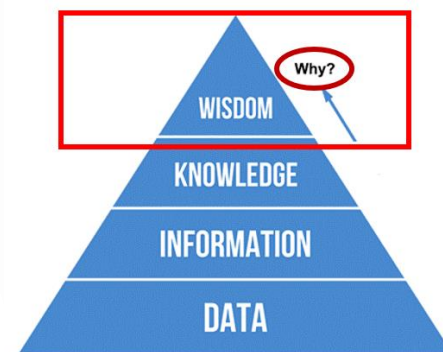


Figure 9: Question of the Wisdom level



Continuing with the example of the Spanish airline company:

An example of wisdom gained by the Spanish airline company is that 70% of young users visit the website to be able to find the best national flight in order to travel with their school or university friends.

2.3.1.5 Summary of the DIKW Model

The DIKW model is a hierarchical pyramid where all the data collected by the company can be converted into wisdom in order for the company to take the best decision to improve its business.

The DIKW model is composed of, in order, the four following levels:

- Data: Raw data collected by the company;
- Information: The data collected in the previous step is processed. Relationships between the data are revealed and analyzed in order to reach information. This level answers the “What”, “Where”, “When”, and “Who” questions;
- Knowledge: Information is processed, guided through specific measures, in order to find and understand patterns and answer the “How” question;
- Wisdom: Knowledge is applied and implemented, and principles are determined. The “Why” question is answered.

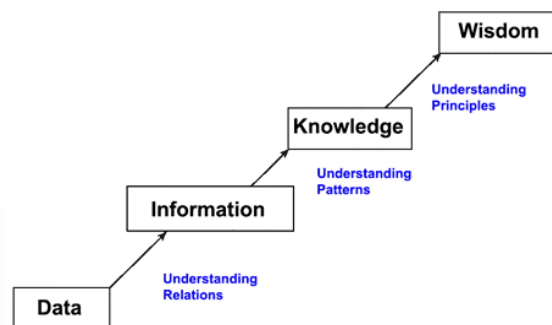


Figure 10: DIKW Model - From Data to Wisdom

The DIKW model converts the raw data collected by the company into wisdom adding value to that data. The raw collected data alone cannot help the company make the best future decisions for its business. However, when the data is converted into information then knowledge and finally wisdom, the company can make proactive and improvement decisions in its business applying the wisdom gained by the data it collected into actionable results.



2.3.2 EXAMPLES

Two examples of how the collection of data and the use of the DIKW model could help a company improve its business are given below. In other words, two examples of how the use of Big Data combined with the DIKW model methodology could help a company improve its business are given below.

2.3.2.1 Example 1

A Supermarket that collects all their sales data converts it into wisdom in order to make improvements that will benefit their business using the DIKW model.

The Supermarket, by collecting its sales data, knew that in the last three months they sold 3000 salted deserts.

After processing their data and analyzing it, they found out that 30% of their salted deserts sales were popcorn.

By processing that information, the Supermarket found similar patterns between their popcorn sales and weather patterns and thus asked themselves how do their customers buy popcorn.

The Supermarket found out that the day before a major storm was announced, their customers tend to buy popcorn.

The Supermarket started placing popcorn in front of the store and next to the cashiers right before a major storm was scheduled to hit. By doing that, the Supermarket' sales went up.

2.3.2.2 Example 2

A Mobile Phone Company that collects all their sales data converts it into wisdom in order to increase their sales using the DIKW model.

The Company, by collecting its sales data, knew that in the last four months they sold 500 mobile phones.

After processing their data and analyzing it, they found out that 70% of their mobile phones sold had a memory superior to 8 GB.

By processing that information, the Company found similar patterns between their mobile phones sales and their earphones sales.

The Company found out that 80% of their customers that buy mobile phones with a memory superior to 8 GB, bought, from them, in the same day, earphones.

The Company started selling packages including a mobile phone and an earphone. By doing that, the Company' sales increased.



2.4 BIG DATA TYPES

Big Data is composed of the three following main types of data:

- Structured Data,
- Unstructured Data,
- Semi-Structured Data.

The existing digital data is divided approximately as follows:

- Structured Data: 20% of the existing digital data;
- Unstructured Data: 80% of the existing digital data with this percentage increasing year over year;
- Semi-Structured Data: 5-10% of the existing Structured/Unstructured digital data.

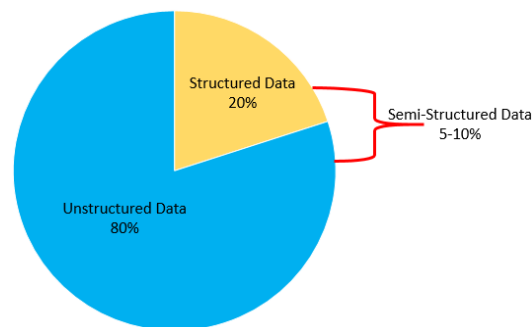


Figure 11: Big Data Types

In this sub-section, a definition as well as examples of each type of data that composes Big Data will be presented.

2.4.1 STRUCTURED DATA

2.4.1.1 Definition

The term “Structured Data” usually refers to data that has an established length and format.

Structured Data is usually stored in databases and is thus displayed in defined columns and rows.

Structured Data could be numbers, dates and strings (groups of words and numbers).



2.4.1.2 Sources

Structured Data can be created in the two following ways:

- Machine-generated or Computer-generated data: Data that is created by a device, machine, computer or sensor without any human intervention.
- Human-generated: Data supplied by humans in interaction with computers or other devices.

2.4.1.3 Examples

Examples of Machine-generated or Computer-generated data:

- Sensor data:
 - o RFID (Radio Frequency ID) tags: Using tiny computer chips to track items at a distance, a company evaluating its supply chain, could track containers from one location to another and collect the results.
 - o GPS (Global Positioning System) data: Using GPS in smartphones, a company could, by collecting all the GPS data from smartphones, try to understand its customers' behaviors in many new ways.
- Web log data: Collecting the activity data of, for example, servers, networks and applications when operating to predict security breaches or deal with service-level agreements.
- Point-of-sales data: When swiping the bar code of any product being purchased in a supermarket, all the data associated with that product is being generated.

Examples of Human-generated data:

- Input data: Data that a human inserts into a device (for example: name, age, income, non-free-form survey responses, etc.). This data could be used by companies to understand their customers' behaviors.
- Click-stream data: Data that is generated when a link is clicked on a website. This data could be used by companies to understand their customers' behaviors and patterns.
- Gaming-related data: Data recording the moves made in a game. This data could be used by companies to understand their end users' moves through a gaming portfolio.



Figure 12: Examples of Structured Data

2.4.2 UNSTRUCTURED DATA

2.4.2.1 Definition

Unstructured Data is the opposite of Structured Data. Basically everything that is not Structured Data is considered Unstructured Data.

Unstructured Data does not follow any specified format nor has any identifiable internal structure (it does not carry any tags (metadata about the data) nor has any established schema nor ontology nor glossary nor consistent organization).

Unstructured Data does not fit neatly in a relational database.

2.4.2.2 Sources

Unstructured Data can be created in the two following ways:

- Machine-generated or Computer-generated data: Data that is created by a device, machine, computer or sensor without any human intervention.
- Human-generated: Data supplied by humans in interaction with computers or other devices.

2.4.2.3 Examples

Examples of Machine-generated or Computer-generated data:

- Satellite images: Weather data, Google Earth, etc.
- Scientific data: Seismic imagery, Atmospheric data, etc.
- Digital surveillance photographs and videos: Security surveillance, Traffic surveillance, etc.
- Sensor data: Vehicular sensors, Oceanographic sensors, etc.

Examples of Human-generated data:

- Company internal documents: Free text files produced in a company, presentations, etc.



- Social Media: Data generated from social media platforms (for example: Facebook).
- Website: Data from any site generating unstructured content (for example: Youtube, Instagram).

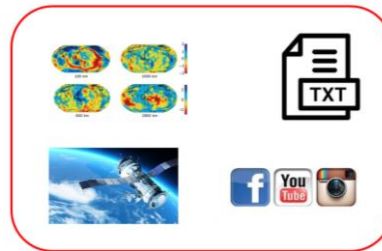


Figure 13: Examples of Unstructured Data

2.4.3 SEMI-STRUCTURED DATA

2.4.3.1 Definition

Semi-Structured Data falls between Structured Data and Unstructured-Data.

The line between Semi-Structured Data and Unstructured Data is a little blurry.

Semi-Structured Data does not fit neatly in a relational database but has some organizational structure (such as a known schema) or contains markers (such as tags) separating semantic elements and enforcing hierarchies of records and fields within the data.

2.4.3.2 Example

The most common example of Semi-Structured Data are emails.

Even though emails lack formal structure, they contain tags and a known structure which separates semantic elements.

2.5 THE “V” MODEL

Big Data is, for and foremost, a collection of data which large volume, high variability and high speed of growth makes it difficult to be captured, managed, processed and analyzed using traditional technologies and tools (such as relational databases) within the time necessary for the data to be useful.

In this sub-section, Big Data will be broken down into the four following dimensions:

- Volume: Size of Data;
- Velocity: Analysis of Data;
- Variety: Different types of Data;
- Veracity: Uncertainty of Data.

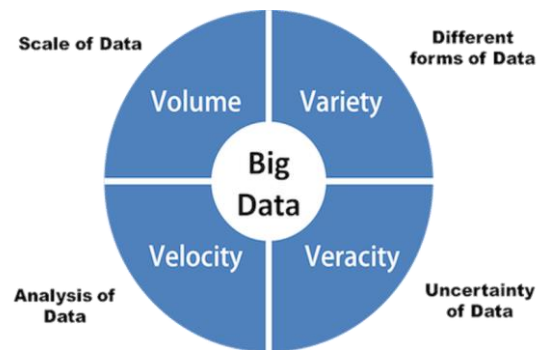


Figure 14: The 4 V's of Big Data

A definition as well as examples of each dimension will be presented.

2.5.1 VOLUME

2.5.1.1 Definition

Big Data represents a large volume of data.

However, there is not a specific size of data that qualifies it as Big Data. Data is considered Big Data if it is, or becomes, too large to be handled by traditional technologies or tools (such as relational databases).

The total amount of digital data generated and being collected is huge and is drastically increasing every year.

2.5.1.2 Examples

The following examples show how big the amount of digital data generated and available to be collected is and will be:

- More than 10 billion messages per day are sent via Facebook.
- The Facebook like button, in 2015, was approximately clicked 4.5 billion times per day.
- An estimate of 40 Zettabytes of data will be created by 2020, which is an increase of 300 times from 2005.
- An estimate of 2.5 Quintillion Bytes of data are being created each day.

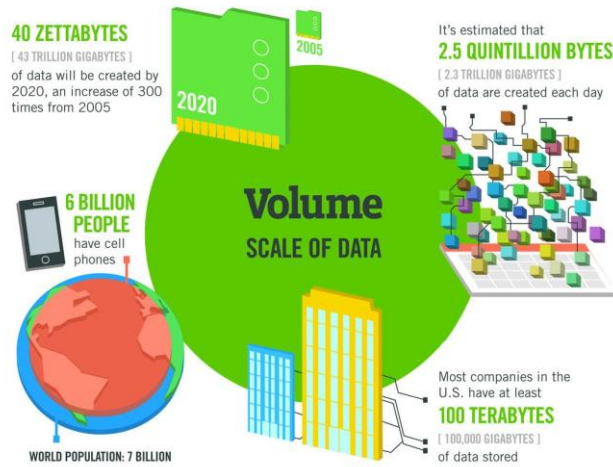


Figure 15: Big Data Volume

2.5.2 VELOCITY

2.5.2.1 Definition

Velocity refers to the frequency of arriving data that needs to be processed.

Velocity is the speed referring to how fast new data is generated and collected as well as how fast it should be analyzed and utilized.

Velocity is crucial for businesses that require real-time analysis of data.

2.5.2.2 Examples

The following examples show how crucial velocity can be for businesses that require real-time analysis of data:

- Credit card fraud detection systems: If a credit card is owned by someone who lives in Madrid but that credit card is suddenly used in Brazil to purchase an item, the system should be able to detect and flag that purchase in real-time.
- Modern Cars Sensors: Modern cars have close to 100 sensors that monitor items such as fuel level and tire pressure. This monitoring requires real-time analysis of data in order not to put the driver of the car in danger.

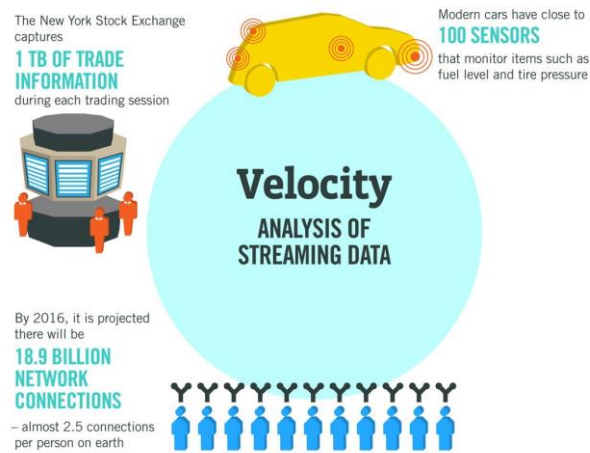


Figure 16: Big Data Velocity

2.5.3 VARIETY

2.5.3.1 Definition

Variety is due to the fact that the digital data that is being generated can be in different forms. The three different types of data that can be generated are:

- Structured Data,
- Unstructured Data,
- Semi-Structured Data.

These three data types have been defined in the previous sub-section.

2.5.3.2 Examples

The following examples will show how different sources of data are generating different forms of data:

- An estimate of 1 billion pieces of content are shared on Facebook every day.
- An estimate of 269 billion emails are sent per day.
- In 2011, the global size of data in healthcare was estimated to be 150 Exabytes.

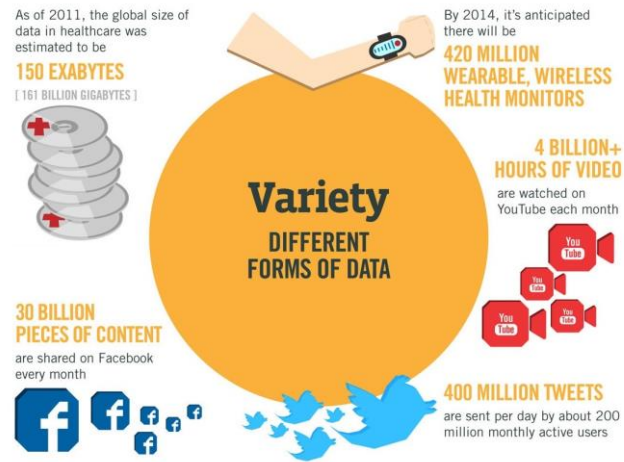


Figure 17: Big Data Variety

2.5.4 VERACITY

2.5.4.1 Definition

Veracity refers to the accuracy, relevancy and representativeness of the data collected.

There are always inherent discrepancies in all the data collected which generate a veracity problem.

Usually, since the size of the data collected is big, a lot of the veracity problem in the data collected tends to get eliminated. However, veracity in the data collected still remains a problem.

2.5.4.2 Examples

The following examples will show how veracity can be a problem in the data collected:

- **Twitter hashtag:** A company that is collecting twitter hashtags in order to analyze them might have a veracity problem as the hashtag might not accurately reflect what is in the Tweet. Even though, a Twitter hashtag usually reflects what is in the Tweet, sometimes it may not.
- **Analyzing social media chatter to develop a new product:** A company decides to analyze a social media chatter for the past year in order to determine which products are buzzing and what people are actually looking for in the product to be able to develop the best new product. If the product is a fast-moving product like mobile devices, the period chosen to analyze the social media chatter might be too long and the data collected might thus be relatively inaccurate.

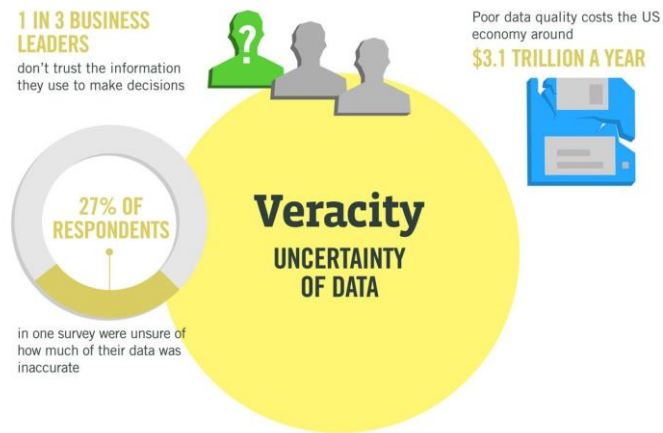


Figure 18: Big Data Veracity



Chapter 3 BIG DATA TECHNOLOGIES AND TOOLS

We live in a data driven society where all the services we interact with on a daily basis, collect all the data that we are feeding them.

Companies all over the world collect all the data they can in order to understand, process, analyze and use that data in order to be able to make the best decisions to improve their businesses.

However, the digital data being generated nowadays and available to be collected has:

- A large volume that is drastically increasing every year,
- A high variety,
- A high velocity,

that makes it very difficult to be captured, managed, processed and analyzed using traditional technologies and tools (such as relational databases), within the time necessary for them to be useful and thus the need of some special technologies and tools such as Hadoop and its ecosystem.

This second section of this thesis is dedicated to give a functional explanation on Hadoop's core components as well as functional explanations on some of the components that constitute its ecosystem.

At first, a brief definition of Hadoop as well as a functional explanation on the components that constitute Hadoop's core architecture will be given. Functional explanations of some of the components that constitute Hadoop's ecosystem will be presented afterwards.

3.1 HADOOP'S CORE COMPONENTS

Hadoop is a collection of open-source software designed to store and process large amount of data efficiently.

Hadoop is suited to handle high volumes of data with various forms of structures allowing to collect, process and analyze efficiently that data.



At its core, Hadoop was composed of the two following main components:

- HDFS (Hadoop Distributed File System): In charge of storing the data within the Hadoop cluster;
- MapReduce: In charge of processing the data as well as managing the resources within the Hadoop cluster.



Figure 19: Hadoop Core Components v1

However, this architecture of Hadoop's core components had some limitations. Even though the limitations of this architecture of Hadoop's core components are not in the scope of this thesis, the following limitation will be mentioned as an example:

HDFS was directly linked to MapReduce and that limited Hadoop's user to running MapReduce jobs only.

In order to confront the limitations of this architecture of Hadoop's core components, the following new component was added:

- YARN (Yet Another Resource Negotiator): In charge of managing the resources within the Hadoop cluster and job scheduling.

YARN will take over the functions of managing the resources within the Hadoop cluster and job scheduling from MapReduce.

Hadoop's main components are now the following:

- HDFS (Hadoop Distributed File System): In charge of storing the data within the Hadoop cluster;
- YARN (Yet Another Resource Negotiator): In charge of managing the resources within the Hadoop cluster and job scheduling;
- MapReduce: In charge of processing the data.

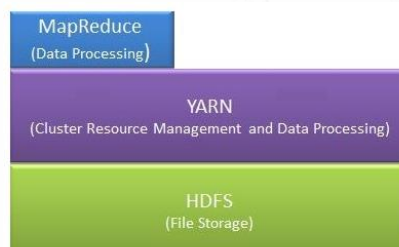


Figure 20: Hadoop Core Components v2

Going back to the example of the limitation previously mentioned:



The addition of YARN ended the strict reliance of HDFS on MapReduce and opened up Hadoop to other processing engines than MapReduce.

In order to understand correctly the definition of Hadoop and the functionality of Hadoop's core components, this sub-section will be divided into several parts.

At first, a definition of an open-source software will be given followed by the definition of a job. The composition of a Hadoop cluster will be presented afterwards. Last but not least, the description of Hadoop's core components will be presented starting with HDFS (Hadoop Distributed File System), followed by MapReduce and finally Yarn (Yet Another Resource Negotiator).

3.1.1 OPEN-SOURCE SOFTWARE

An open-source software is a computer software that is distributed with its source code available for modification. It usually includes a license for programmers in order to be able to study and change the software in any way they like, for example, by fixing bugs, improving functions or adapting the software in order to suit their own needs.

3.1.2 JOB

A job is a unit of work or task that a computer program executes based on some predefined order.

Job scheduling is the process of coordinating the execution of the jobs a system needs to execute.

3.1.3 HADOOP CLUSTER

A computer cluster is composed of various computers that are connected and work together in such a way that they could be seen as a single system.

Clustering is mainly used to:

- Parallel processing: Splitting a process into various parts that are executed on different processors simultaneously;
- Load balancing: Improving the workloads' distribution over various computers;
- Fault tolerance: Enabling the system to properly continue to operate if one or more of its components fail.

Power is added to the cluster by adding a new computer to the cluster, making it relatively easy to add power to the cluster.



A Hadoop cluster is a type of computer cluster.

A Hadoop cluster is composed of various nodes that are connected through a network.

A node is typically used to describe a machine or computer that is present within a cluster.

A Hadoop cluster can also be seen as a collection of racks.

A rack is a collection of 30 or 40 nodes that are physically stored close together and are all connected to the same network switch.

A Hadoop cluster is designed for storing and analyzing large volumes of data in a distributed computing environment.

Distributed computing refers to a model where components of a software system are shared among various computers in order to improve efficiency and performance.

Hadoop is thus designed to be deployed on a large cluster of computers that are connected through a network and is composed of the two following types of nodes:

- Master nodes: Master nodes host the services that control Hadoop's storage and processing;
- Slave nodes: Data is stored and processed in the slave nodes.

Hadoop clusters are:

- Highly scalable: If a cluster needs more processing power due to the growing volumes of data, additional nodes could be added to the cluster in order to increase throughput;
- Highly resistant to failure: When data is stored into the cluster, each piece of data is copied into various nodes of the cluster ensuring that the piece of data is not lost in case a node fails.

Hadoop clusters are also known to increase the speed of data analysis applications.

3.1.4 HDFS (HADOOP DISTRIBUTED FILE SYSTEM)

Hadoop Distributed File System, also known as HDFS, is in charge of storing data using a distributed environment. That is to say, data is dispersed and stored over many machines (known as nodes in a distributed environment) which are connected with each other through a network. In other words, HDFS is in charge of storing data in the Hadoop cluster.

HDFS is Hadoop storage layer.

HDFS was designed to be able to store and handle very large amount of data in an efficient manner.



Figure 21: HDFS Symbol

When data is stored in HDFS, it is divided into smaller chunks, known as blocks, and each block of data is stored on separate Slave Nodes within the Hadoop cluster.

The data storage unit in HDFS is thus called a block and represents 64 MB in size.

For example:

When 1 GB of data is going to be stored in HDFS, it is divided into 15 blocks of 64 MB and 1 block of 40 MB.

Each one of these blocks will be stored on separate Slave Nodes within the Hadoop cluster.

By dividing the data that is going to be stored in HDFS into several blocks before storing it, the total time to store the entire data in HDFS is significantly reduced. As a matter of fact, the total time to store the entire data in HDFS is approximately equal to storing one part, known as block, of the data (the biggest block, generally equal to 64 MB) in HDFS since all the parts of the data, known as blocks, will be stored simultaneously on different Slave Nodes within the cluster.

On the other hand, when the blocks of data are going to be stored in HDFS, each block of data is replicated several times and stored on different Slave Nodes that are present within the Hadoop cluster. The copies of the block of data are known as replicas.

The number of replicas of each block of data depends on the replication factor. The replication factor, by default, is equal to 3.

Example:

If the default replication factor is set, that is to say if the replication factor is set to 3, each block of data will be replicated 2 times, having then 3 copies of each block of data on three different Slave Nodes within the Hadoop cluster.

Since the data that is stored in HDFS is distributed across the Hadoop cluster, HDFS takes care of the networking required by the nodes within the Hadoop cluster in order to communicate and perform the storage and the retrieval of data properly within the Hadoop cluster.



Only one Master Node is active in a Hadoop cluster.

The Master Node is the overall coordinator of HDFS. Its main duties are to allocate, maintain and report the locations of the data stored in HDFS.

The Master Node is in charge of storing the data in HDFS by providing storage space for each piece of data in the Slave Nodes of the Hadoop cluster.

The Master Node gives instructions to the Slave Nodes and the Slave Nodes report the updates on the data to the Master Node.

The Master Node keeps track of all the copies of every block of data stored in HDFS. By doing so, the Master Node has the complete picture of the entire storage available in the Hadoop cluster.

The Master Node maintains the list of all the files stored in HDFS and the actual location of each block of data that constitute the file in the distributed system.

The Master Node maintains thus a table indicating:

- Each block of data's allocation in the various Slave Nodes,
- Each replica of the block of data's allocation in the different Slave Nodes,
- The related access information of every block of data.

The complete information about all the files that are stored in HDFS is known as metadata (the metadata specially contains the name of the file, the location of the file, the access rights, etc.). The metadata contains the information related to which blocks stored in HDFS belongs to which file.

The Master Node contains:

- The fsImage file: The fsImage file stores the complete snapshot of HDFS' metadata;
- The edit log file: The edit log file contains all the modifications done to HDFS' metadata.

A major duty of the Master Node is to provide the block of data information to the related job executions.

The Master Node is thus the only node in the Hadoop cluster which has the complete information about everything in HDFS and is therefore the Single Point Of Failure (also known as SPOF). For that reason, if the Master Node of the Hadoop cluster fails, the complete Hadoop cluster will come to halt. To avoid such scenario, another node in the Hadoop cluster should act up as a Backup or Secondary Master Node.

This Backup or Secondary Master Node should contain and maintain the fsImage file and the edit log file of the Master Node of the Hadoop cluster in order to maintain a consistent copy of the complete information about everything in HDFS.

In order to maintain the fsImage file and the edit log file of the Master Node of the Hadoop cluster, the Backup or Secondary Master Node will periodically retrieve the edit log file from the Master Node, update its edit log file and merge its updated edit log file with its fsImage file. By doing that, the Backup or Secondary Master Node will always have a consistent copy of the complete



information about everything in HDFS and is thus able to take over the charge of the Master Node of the Hadoop cluster in case the Master Node of the Hadoop cluster fails.

3.1.5 MAPREDUCE

MapReduce is in charge of processing the data that is stored in the Hadoop Distributed File System.

MapReduce is a programming model used to process a large dataset by breaking it into smaller pieces of datasets and processing each piece of dataset in parallel and with the same algorithm. Since the smaller pieces of datasets are processed simultaneously, the total amount of time necessary to process the large dataset is significantly reduced.

As HDFS divides a large dataset into smaller blocks of data before storing it, MapReduce is therefore a processing model that pairs with HDFS as it could be directly used to process the large dataset already divided into smaller blocks of data.

MapReduce is therefore a programming model used for processing datasets that are stored in a distributed environment, that is to say datasets that are stored in the Slave Nodes across Hadoop's cluster.



Figure 22: MapReduce Symbol

The key concept in MapReduce is “divide and conquer”.

When a MapReduce job is submitted by the Client to the Master Node of the Hadoop cluster, what especially happens in the Hadoop cluster is the following:

- 1- Determination of the exact datasets that need to be processed from the blocks of data stored in the Slave Nodes within the Hadoop cluster. That is to say, determination of the exact location of the datasets that should be processed within the blocks of data stored in Hadoop cluster's Slave Nodes.
- 2- Slave Nodes will run the specified job against each dataset that needs to be processed until all the datasets that should be processed have been processed. Each Slave Node running the specified job on a dataset is doing a mapper task (This corresponds to the mapping part of MapReduce).
- 3- Slave Nodes will organize and reduce (in other words, combine) the output of the mapper tasks until all the results of the mapper tasks are combined into one single result obtaining therefore the output of the job submitted by



the Client. Each Slave Node organizing and reducing the output of the mapper tasks is doing a reducer task (This corresponds to the reduce part of MapReduce).

3.1.6 YARN (YET ANOTHER RESOURCE NEGOTIATOR)

Yet Another Resource Negotiator, also known as YARN, is in charge of managing the resources within the Hadoop cluster as well as scheduling the jobs in the Hadoop cluster.

YARN is responsible to allocate the resources within the Hadoop cluster for the different applications or workloads running in the Hadoop cluster. YARN is also responsible to schedule the execution of the tasks on the different nodes within the Hadoop cluster.



Figure 23: YARN Symbol

The Master Node of the Hadoop cluster is in charge of the submission and the scheduling of jobs in the cluster. The Master Node of the Hadoop cluster is also in charge in monitoring the jobs and allocating the resources.

The Slave Nodes in the Hadoop cluster interact with the Master Node in order to run tasks and track resource usage.

The Slave Nodes periodically report their statuses to the Master Node of the Hadoop cluster. If a Slave Node does not respond as expected, the Master Node will reassign the job that is assigned to that Slave Node to other available Slave Nodes in the Hadoop cluster.

3.2 HADOOP'S ECOSYSTEM

Users of Hadoop's core components needed to ingest and store their data into Hadoop, secure their data stored in Hadoop as well as process, analyze and use their data stored in Hadoop. Hadoop's core components had thus some missing capabilities and therefore an ecosystem of components was needed and started to get build around Hadoop's core components.

In this sub-section, some of the components of Hadoop's ecosystem will be presented.

In order to understand correctly the functionality of these components, this sub-section will be divided into four parts. In the first part, some of Hadoop's



ecosystem storage layers will be presented. In the second part, some of Hadoop's ecosystem ingestion mechanisms will be presented. In the third part, a component of Hadoop's ecosystem in charge of securing the data that is stored in Hadoop's ecosystem will be presented. In the fourth and last part, some of the components of Hadoop's ecosystem in charge of processing and analyzing the data stored in Hadoop's ecosystem will be presented.

3.2.1 STORE

In this part of this sub-section, the following two storage layers of Hadoop's ecosystem will be presented:

- HBase,
- Kudu.

3.2.1.1 HBase

HBase is a non-relational, highly distributed database that runs on top of HDFS and is able to store large amount of distributed data.



Figure 24: HBase Symbol

A distributed database is a database that is stored across a computer cluster.

A non-relational database is a database that does not use the tabular relation used in relational databases to store and retrieve the data. Instead, non-relational databases use a storage model optimized for the specific requirements of the type of data being stored.

3.2.1.2 KUDU

KUDU is a storage layer in Hadoop's ecosystem that has a database-like semantic and a data model similar to the relational data model. It was designed and implemented to fill the gap between HDFS and HBase.



Figure 25: KUDU Symbol



HBase provides fast read/write random access to the data stored in it. That is to say, individual rows of data stored in HBase can be found, written and mutated efficiently.

However, HBase provides a pretty bad performance for scanning large amounts of data stored in it.

On the other hand, HDFS provides a very good performance for scanning large amounts of data stored in it.

However, HDFS does not have the capability of fast random access to the data stored in it. That is to say, HDFS is not suitable for reading data from a random position within a file stored in it and is best suited for reading data either from the beginning or the end of a file stored in it.

HDFS has a Write Once, Read Often model of data access. In other words, the content of a file stored in HDFS cannot be modified, other than adding new data at the end of the file.

In other words, HDFS has a good performance for scanning large amount of data stored in it but does not have the capability of fast read/write random access to the data stored in it. On the other hand, HBase has a bad performance for scanning large amount of data stored in it but has the capability of fast read/write random access to the data stored in it efficiently.

In order to fill the gap between HDFS and HBase, a new storage layer in Hadoop's ecosystem was needed that provides:

- Good performance for scanning large amount of data stored in it;
- Good performance for read/write random access to the data stored in it.

KUDU was thus designed and implemented. KUDU fills the gap between HDFS and HBase. KUDU provides a good performance for scanning large amount of data stored in it (but not as good as HDFS) as well as a good performance for random read/write access to the data stored in it (but not as good as HBase).

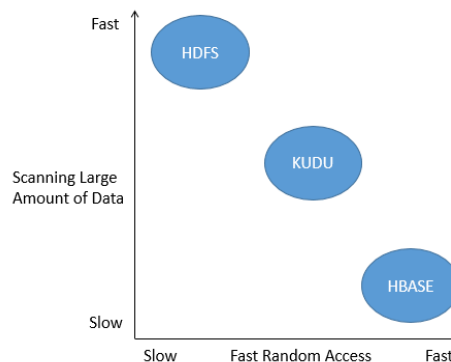


Figure 26: HDFS vs. KUDU vs. HBase



3.2.2 INTEGRATE

In this part of this sub-section, the following two components of Hadoop's ingestion layer will be presented:

- Sqoop,
- Flume.

3.2.2.1 Sqoop

Sqoop is designed for efficiently transferring data from relational databases to Hadoop as well as from Hadoop to relational databases.



Figure 27: Sqoop Symbol

Sqoop can transfer data from relational databases to HDFS or HBase but not to KUDU as well as transfer data from HDFS or HBase to relational databases but not from KUDU.

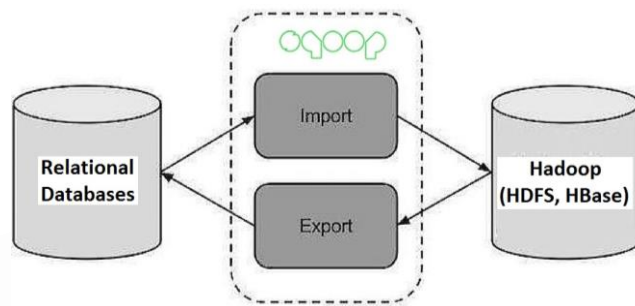


Figure 28: Sqoop's Ingestion Flow

3.2.2.2 Flume

Flume is designed to efficiently and continuously aggregate and move large amounts of streaming data (such as data logs) from different sources into Hadoop.



Figure 29: Flume Symbol

Flume runs one or more agents. Each agent must have at least one of each of the three following components:

- Source: Get data and send it to the channels;
- Channel: Conduit between sources and sinks. Hold data queues which is useful when the input flow rate exceeds the output flow rate;
- Sink: Take the data from channels, process the data and deliver the data to Hadoop.

The unit of data processed by Flume is called an event.

Flume can deliver data to HDFS, KUDU or HBase.

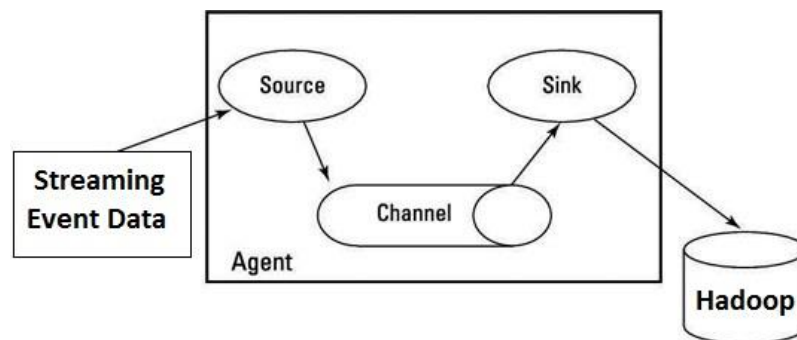


Figure 30: Flume's Ingestion Flow

3.2.3 SECURITY

In this part of this sub-section, the following component of the Hadoop's ecosystem in charge of securing the accesses to the data and metadata stored in Hadoop's ecosystem will be presented:

- Sentry.

3.2.3.1 Sentry

Sentry is a role-based authorization module for Hadoop providing the ability to control and impose precise levels of privileges and permissions on the data and metadata stored in Hadoop for identified users and applications.

In other words, Sentry defines the accesses permissions to the data and metadata stored in Hadoop to Hadoop's users and applications by defining what users and applications can do with the data stored in Hadoop.



Figure 31: Sentry Symbol

A role-based authorization module is an authorization module that:

- Creates roles and grants privileges to these roles;
- Joins users and group of users to the roles.

Users and group of users are not granted privileges and permissions directly but acquire them through their role or roles.

In Hadoop's ecosystem, groups are created and identified users are assigned to these groups. These groups are called Active Directories. Identified users of Hadoop's ecosystem are thus assigned to Active Directories.

Sentry first creates roles and grants privileges and permissions to these roles. Afterwards, Sentry joins identified users and Active Directories (authentication entities) to the roles (authorization entities).

3.2.4 PROCESS, ANALYZE AND SERVE

In this part of this sub-section, the following components of Hadoop's ecosystem in charge of processing and analyzing the data stored in Hadoop's ecosystem will be presented:

- Pig,
- Hive,
- Impala,
- Mahout,
- SOLR,
- Spark,
- HUE.

3.2.4.1 Pig

Pig is a high-level scripting language used with Hadoop.



Figure 32: Pig Symbol



Programmers encounter difficulties writing MapReduce tasks because they need Python or Java programming knowledge.

Pig enables programmers to write MapReduce programs without knowing Java nor Python. Programmers can write MapReduce programs in Pig using Pig Latin and run their program against the data stored in Hadoop's ecosystem.

Pig Latin is Pig's SQL-like scripting language and usually attracts developers familiar with scripting languages and SQL.

3.2.4.2 *Hive*

Hive is designed to facilitate reading, writing and managing large datasets stored in Hadoop's ecosystem using SQL.



Figure 33: Hive Symbol

Hive allows its users to run SQL, which is converted to MapReduce, against the data stored in Hadoop's ecosystem.

3.2.4.3 *Impala*

Impala is a SQL query engine for data stored in Hadoop's ecosystem.



Figure 34: Impala Symbol

Impala allows its users to query and search the data stored in Hadoop's ecosystem in real time using easy SQL commands.

Impala works around MapReduce for direct access to the data stored in Hadoop's ecosystem. Impala is thus faster than Hive in querying data stored in Hadoop's ecosystem.



3.2.4.4 Mahout

Mahout is a library of scalable machine-learning algorithms running on top of Hadoop MapReduce.



Figure 35: Mahout Symbol

Machine learning is a discipline of artificial intelligence allowing machines to automatically learn from experience and without being explicitly programmed. Machines will thus improve their performance based on their previous results.

Mahout aims to automatically find meaningful patterns in the data stored and thus turning data into information in a faster and easier way.

3.2.4.5 SOLR

SOLR is a platform for rapid indexing and searches of data stored in Hadoop's ecosystem.



Figure 36: SOLR Symbol

SOLR allows the indexing of all the data stored in Hadoop's ecosystem which is very useful when the data is going to be searched.

The data being ingested in Hadoop's ecosystem through Sqoop or Flume could be routed to pass by SOLR first in order to do the indexing on the fly. However, SOLR could also index that data in batches.

3.2.4.6 Spark

Spark is an engine designed for handling large volumes of data processing and analytics.



Figure 37: Spark Symbol

Spark is designed to run programs against data faster than MapReduce and might be able to replace MapReduce in the future.



Other than being designed to accelerate analytics on the data stored in Hadoop's ecosystem, Spark also provides stream processing (processing the data directly as it is produced or received instead of storing the data before processing it) as well as a fully-featured machine learning library.

3.2.4.7 HUE

HUE (Hadoop User Experience) is a web user interface for Hadoop.



Figure 38: HUE Symbol

Hue allows technical and non-technical users to query, analyze and visualize the data stored in Hadoop's ecosystem using some of Hadoop's ecosystem components.



Chapter 4 GENERAL DATA PROTECTION REGULATION

We live in a data driven society where all the services we interact with on a daily basis collect all the data that we are feeding them.

Companies all over the world are collect all the data they can in order to understand, process, analyze and use that data to be able to study customers' behaviors in order to make the best decisions to improve their businesses.

The world is becoming increasingly digitized to the point where nearly every part of people's life can be digitized and thus more and more of people's personal information is being collected, stored, processed, analyzed, used and sometimes even traded by companies.

Many companies are therefore holding highly sensitive customer personal information data which is associated with a significant risk if abused or stolen. The General Data Protection Regulation was consequently introduced to specify how customer's personal information data collected by companies should be used and protected by these companies.

This third section of this thesis will explain what the General Data Protection Regulation is as well as introduce its major requirements and their impact on businesses. Additionally, the way companies will be able to adapt to some of the General Data Protection Regulation's requirements using some of Hadoop's ecosystem components as well as one of Cloudera's ready tool (an enterprise that, in addition of providing packages with its own distribution of Hadoop's ecosystem, provides some ready tools that makes it easier for companies to manage their data) will be explained.

In the first part of this section, a definition and an overview of the General Data Protection Regulation will be given. The definition of personal data under the General Data Protection Regulation will be explained afterwards. Subsequently, the General Data Protection Regulation's major requirements will be presented. Then, the way the General Data Protection Regulation will impact companies' businesses will be described. Last but not least, the way companies will be able to conform to some of the General Data Protection Regulation's requirements by using some of Hadoop's ecosystem components as well as one of Cloudera's ready tool will be explained.



4.1 DEFINITION AND OVERVIEW

The General Data Protection Regulation, also known as GDPR, is a regulation in the European Union law on data protection and privacy that provides a set of standardized data protection laws for the collection and processing of personal information of individuals within the European Union.

The General Data Protection Regulation was agreed upon by the European Parliament and Council in April 2016 as the main law regulating how companies use and protect European Union citizens' personal data.

The General Data Protection Regulation was officially adopted by the European Parliament and Council in April 2016 but followed a two-year post adoption period and became enforceable throughout the European Union on 25 May 2018.

The General Data Protection Regulation affects all companies that either operate in the European Union or handle data related to citizens of the European Union. That is to say, the GDPR applies to any business involved in collecting and processing personal data of citizens of the European Union in the context of selling goods and services to citizens of the European Union regardless whether the company is located within the European Union or not.

Therefore, the GDPR will affect all the companies that have businesses in the European Union as well as companies based outside the European Union but that store and process personal data about European Union citizens. The United Kingdom's government said it will also apply the GDPR regulation in British law regardless of how the Brexit deal turns out. Some companies based outside of the European Union but that operate worldwide are going to apply the GDPR regulation to all their users around the world whether or not they are European Union citizens.

With the General Data Protection Regulation, businesses will have to take the individuals' consent for using their personal data. The GDPR gives the individual the right to find out whether its personal data is being processed, where it is being processed and for what purpose it is being processed. Under the GDPR, individuals can choose to move their personal data across different IT environments as well as object to having it processed for direct marketing purposes. Individuals are also entitled to have their personal data erased or not spread any further including potentially stopping third parties from processing their personal data. Under the GDPR, companies have to protect and keep the individuals' personal data safe and have to inform them within 72 hours if any breach of their personal data occurs.



4.2 PERSONAL INFORMATION UNDER THE GENERAL DATA PROTECTION REGULATION

The ultimate goal of the GDPR is to protect European Union’s citizens’ personal information that is collected by companies.

So, the question here is “What is defined as Personal Information?” or more precisely “What is considered Personal Information under the GDPR?”

Personal information means any information related to an individual who can be identified, directly or indirectly, in particular by reference to that information.

In other words, personal information is any information that can be used to identify an individual.

Personal information under the GDPR includes anything from an individual’s name, address, social security number, to his location on an online identifier like an IP address or browser cookies that can track the individual’s web activity. An individual’s physical identity, physiological identity, genetic identity, mental identity, economic identity, cultural identity or social identity also falls under the definition of personal information under the GDPR.



Figure 39: Personal Information under the GDPR



4.3 GENERAL DATA PROTECTION REGULATION'S REQUIREMENTS

In order to protect European Union's citizens' personal information from any risk of being abused or stolen from the companies that collect them, the GDPR has established several requirements to be respected by companies that have businesses in the European Union or are based outside of the European Union but store personal information about European Union citizens.

The GDPR's major requirements include the seven following areas:

- Consent,
- Breach Notification,
- Right to Access,
- Right to be forgotten,
- Data Portability,
- Privacy by Design,
- Data Protection Officers.

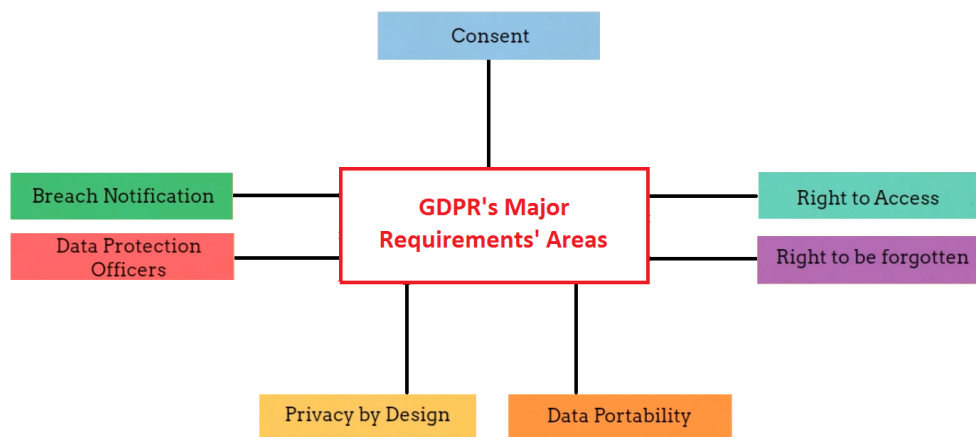


Figure 40: GDPR's Major Requirements' Areas

This sub-section will go through each of these seven areas.



4.3.1 CONSENT

Companies need to obtain the individual's consent if they want to store and use his personal information specifying to him what type of personal data is being stored and why it is being stored.

To obtain the individual's consent to store and use his personal data, companies cannot use unreadable terms and conditions filled with legalese. In other words, companies that want to store and use an individual's personal information, have to be upfront about it when asking him the permission to do so.

It should also be as easy to withdraw consent as it is to give it.

4.3.2 BREACH NOTIFICATION

In the event of a personal data breach, the company has to inform the individual concerned of said breach within 72 hours after becoming aware of it.

That is to say, if the company's system was broken into and any of the individual's personal information was stolen or if the company loses any of the individual's personal information, the company has to notify the individual concerned within 72 hours after becoming aware of it.

4.3.3 RIGHT TO ACCESS

Individuals have the right to obtain confirmation from the company of whether their personal data are being processed.

Every individual has the right to obtain from the company processing his personal data:

- The purpose of the processing;
- The categories of personal data concerned.

The company should provide an electronic copy for free to the individual concerned.

4.3.4 RIGHT TO BE FORGOTTEN

When the individual's personal data stored by a company is no longer relevant to its original purpose, the individual can have the company erase his personal data from their system and end its distribution.



4.3.5 DATA PORTABILITY

Individuals are allowed to obtain their personal data stored by the company from said company, as well as reuse their personal information for their own purposes by transferring it across different IT environments.

4.3.6 PRIVACY BY DESIGN

Companies should implement the appropriate technical and infrastructural measures to include data protection from the beginning of their data processing systems.

An example of technical measure that could be used by the company for data protection is the pseudonymisation of personal data.

Pseudonymisation of personal data means the processing of personal data in a way that the personal data cannot be attributed anymore to a specific individual without the use of additional information, as long as the additional information is stored separately and is subject to technical and organizational measures to make sure that the personal data could not be linked to a specific individual.

Another example of technical measure that could be used by the company for data protection is the encryption of personal data.

4.3.7 DATA PROTECTION OFFICERS

Companies that store or process large volumes of sensitive personal data must appoint a Data Protection Officer (DPO).

Some of the responsibilities of a Data Protection Officer are the following:

- To advise and give recommendations to the company and its employees on the importance of applying the GDPR requirements;
- Preparing the company's employees involved in data processing;
- Perform regular audits to ensure the application of the GDPR requirements by the company and its employees and deal with possible problems proactively;
- Serving as a contact between the company and the GDPR Supervisory Authorities (SAs).

Each member of the European Union should have at least one Supervisory Authority. The Supervisory Authority is an independent public authority who should mainly monitor the application of the GDPR.



4.4 IMPACT OF THE GENERAL DATA PROTECTION REGULATION ON BUSINESSES

The General Data Protection Regulation, agreed upon by the European Parliament and Council in April 2016 as the main law regulating the way companies should use and protect European Union citizens' personal data, came into effect on 25 May 2018.

The General Data Protection Regulation will affect all the companies that have businesses in the European Union as well as companies based outside the European Union but that store and process personal data about European Union citizens.

In order to correctly comply with the GDPR, companies handling European Union citizens' personal data should:

- Be able to prove that they have obtained the consent from all the individuals, whose personal information are stored and used, to do so in an upfront way;
- Make sure to have the right procedure in place to detect, report, and investigate a personal data breach within 72 hours;
- Be able to provide an electronic copy to individuals whose personal information are being processed, stating the categories of personal data concerned and the purpose of the processing;
- Make sure to have the necessary technology to be able to quickly remove all the personal data stored of an individual upon his request;
- Make sure to have the necessary technology to be able to search and find all the personal data stored of an individual upon his request;
- Implement the appropriate technical and infrastructural measures (such as pseudonymisation of personal data and encryption of personal data) to include personal data protection in their data processing systems;
- Appoint a Data Protection Officer if required.

Companies that do not comply with the GDPR will suffer serious consequences as they could be fined up to 4% of their total worldwide annual turnover of the preceding financial year or up to 20 Million Euros, whichever one is higher, for non-compliance.



4.5 GENERAL DATA PROTECTION REGULATION, HADOOP AND CLUDERA

Cloudera is an enterprise that provides companies packages with its own distribution of Hadoop and Hadoop's ecosystem components in a platform that allows companies to deploy and manage Hadoop and Hadoop's ecosystem components in order to collect, store, process, analyze, manipulate and visualize their data and to keep their data secure and protected.

In addition of providing companies with its own distribution of Hadoop and Hadoop's ecosystem components, Cloudera also provides companies some ready tools (such as Cloudera Navigator) that makes it easier for companies to manage, govern, secure and protect their data.

Cloudera Navigator is a ready tool provided by Cloudera that enables companies to manage and secure their data collected and stored in Hadoop and its ecosystem.



Figure 41: Cloudera Navigator Symbol

Companies handling European Union citizens' personal information are able to conform to some of the GDPR's requirements by using Hadoop, some of Hadoop's ecosystem components as well as some of Cloudera's ready tools when collecting, storing, processing, analyzing, manipulating and visualizing their data.

By using Kudu, companies are able to:

- Quickly update individual records stored;
- Quickly erase individual records stored.

By using SOLR, companies are able to:

- Index and quickly search the data stored.

By using Sentry, companies are able to:

- Control and impose precise levels of privileges and permissions on the data and metadata stored for users and applications.

By using Cloudera Navigator, companies are able to:

- Classify/tag and track personal data;
The company can thus determine exactly where specific individual personal data reside and apply the appropriate controls or produce reports for audit.
- Detect and analyze breaches;



- Encrypt and anonymize data.

By using Cloudera Navigator and Sentry, companies are able to:

- Tag the data that can be accessed and impose a time limit for such access.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

General Data Protection Regulation



Chapter 5 DATA LAKE

Companies are always looking for ways to process and analyze all the data that they are collecting in order to find patterns, trends and associations between said data and thus turn it into information that will help them make the best decisions to improve their businesses, generate more revenue and be more efficient than their competitors.

However, if all the data collected by the company is not stored in the same place, it will make it very difficult for the company to find patterns, trends and associations between the data that is stored in different places. Therefore, the need for a single storage repository within the company to store all the data that they are collecting is necessary.

On the other hand, the world is becoming increasingly digitized and thus the digital data being generated and available to be collected by companies have:

- A large volume that is drastically increasing every year;
- A high velocity, that is to say, the data is being generated with a high speed and requires a real-time analysis for it to be useful for the company collecting it;
- A wide variety, that is to say, the data being generated has a wide range of format (in other words, the data generated can be structured data, semi-structured data or unstructured data).

Thus, the need of a scalable storage repository where the company can store different types of data that allows the company to make real-time analysis of the data collected.

Companies handling European Union citizens' personal data have to comply with the General Data Protection Regulation that became enforceable on 25 May 2018 as the main law regulating the way companies should use and protect European Union citizens' personal data. Companies that do not comply with the General Data Protection Regulation could be fined up to 4% of their total worldwide annual turnover of the preceding financial year or up to 20 Million Euros, whichever one is higher, for non-compliance.

Having a single storage repository for all the data collected by the company will make it easier for the company to manage, govern and secure their data making it easier for the company to comply with the General Data Protection Regulation and avoid penalties.

This fourth section of this thesis will give an overview of the concept “Data Lake” as well as some advantages on using a data model based on a data lake. Additionally, some examples will be given on how companies using a data model based on a data lake have been able to generate more revenue and improve their businesses.

In the first part of this section, the definition of a data lake followed by an overview and some advantages on using a data model based on a data lake will be presented. Some examples on how companies using a data model based on a data lake were able to improve their businesses, generate more revenue and be more efficient than their competitors will be given afterwards.

5.1 DEFINITION, OVERVIEW AND ADVANTAGES

A data lake is fore and foremost a storage repository that holds a large volume of raw data in its native format. Therefore, a data lake can contain all the following different types of data:

- Structured data;
- Semi-Structured data;
- Unstructured data.

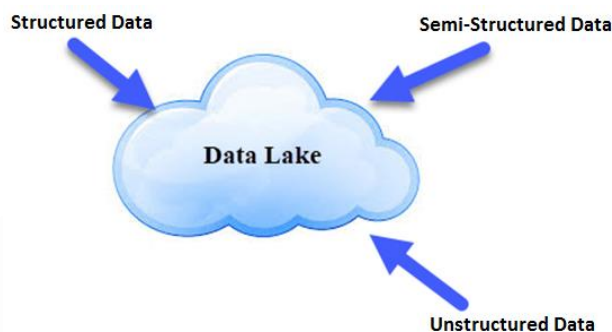


Figure 42: Types of data in a Data Lake

A data lake can be seen as a real lake with rivers flowing into the lake. The rivers carry the data from the sources (generating the data) into the lake. The lake is then a large container holding all the data received from the rivers. In other words, a data lake contains all the raw data collected by the company.

Therefore, all the raw data, whether it is structured data, semi-structured data or unstructured data, that the company might need is collected and stored in its original format in a single repository within the company.

The raw data collected by the company needs to be processed and analyzed in order to find patterns, trends and associations between said data to turn it into



information that will help the company make the best decisions to improve its businesses, generate more revenue and be more efficient than its competitors.

Since in a data lake all the raw data collected by the company is stored in a single repository, the company can access all its data more easily, process it, analyze it and find patterns, trends and associations between all its data.

More specifically, since a data lake has the ability to store all the different types of data collected by the company (whether it is structured data, semi-structured data or unstructured data) in a single repository, the company can, by processing and analyzing its different types of data together, find new patterns, trends and associations between these different types of data that a priori did not seem to be linked together making the company understand new things about its business, its customers, its competition and its market and thus making the company get more value from its collected raw data.

The data generated with a high speed and collected by companies that operate in industries where success depends on acting quickly requires a real-time analysis for it to be useful for the companies collecting it.

Since the raw data collected by companies is stored in the data lake in its original format, it is thereby quickly available for the companies collecting said data to access it, process it and analyze it in real time in order to find patterns, spot trends and find associations between that data in real time and thus turning it into information that will let the companies make predictions and proactive decisions that will improve their businesses, generate more revenue and be more efficient than their competitors.

The world is becoming increasingly digitized and thus the digital data being generated and available to be collected by companies has a large volume that is drastically increasing every year.

In order for companies to be able to store all the data that they are collecting in a single storage repository and be able to, in some cases, hold on to that data for a very long time, this single storage repository needs to be scalable.

In order for the data lake to be a scalable storage repository, it should be built using some special tools and technologies that can provide scalable storage infrastructure for the data collected such as Hadoop and its ecosystem.

When a company wants to process or analyze a large volume of stored data, it is better for the company to be able to run the program against the data where the data is physically located instead of moving that large volume of data around.

By using some special tools and technologies such as Hadoop and its ecosystem, it is possible for the company to process and analyze its data stored in its data lake by running the program where the data is physically located instead of moving that data around.



If all the data that is collected by the company is not stored in the same place, it will be hard for the company to manage, govern and secure its data as well as comply with regulations to avoid penalties.

Since the company's data lake contains all the data collected by the company, it is easier for the company to manage, govern and secure its data as well as comply with regulations such as the General Data Protection Regulation to avoid penalties.

5.2 EXAMPLES

In this sub-section, three examples will be presented showing how companies using a data model based on a data lake were able to improve their businesses, generate more revenue and be more efficient than their competitors.

5.2.1 EXAMPLE 1

A movie theatre company owns a movie theatre in a big mall downtown. Just in front of that movie theatre, the company also owns a small place with some arcade games where customers waiting for their movie to start could spend some time and money playing on these arcade games.

In order for a customer to see a movie or play with any of the arcade games of this company, he has to fill in an electronic form with some personal details (his name, family name, email and age). The customer is thus handed a fidelity card that he has to fill with money in order to be able to buy a movie ticket or pay to play an arcade game.

The company had the objective to increase the sales of its movie tickets by the end of the following month.

In order to do that, the company started analyzing the number of viewers that each movie had during the first week it was projected, offering for the second week discounts on the movies that had less viewers than expected during their first week.

Even though this approach boosted their movie ticket sales in the past, it was not enough for the company to increase their movie ticket sales and reach the objective set for the following month.

Therefore, the company invested in infrastructure and implemented a data model based on a data lake enabling the company to store all the data that they collect from their customers in one single place.

By doing that, the company was able to gather and combine in one single repository all the following data collected from their customers:



- The personal information filled by their customers when applying for the fidelity card;
- The day and time each customer bought a movie ticket as well as which movie he is going to watch;
- The day and time each customer played an arcade game as well as which arcade game he played.

By storing all their customers' data in one single repository, the company was able to, by processing and analyzing all their data together, find the following new relationship between their data:

- On Saturdays, some customers played with the arcade games from 12:00pm till 2:00pm and then from 5:00pm till 6:00pm, watching a movie between 3:00pm and 5:00pm.

The company realized that these customers prefer playing arcade games than watching movies. However, since the arcade games close from 2:00pm till 5:00pm, these customers buy a movie ticket for the session of 3:00pm till 5:00pm only to kill time. As these customers were not really picky on which movie they were going to watch and were only watching said movie to kill time until the arcade games reopen, the company started offering special discounts to these customers for the Saturday's 3:00pm session on movies that had less viewers than expected.

By doing that, the company boosted its movie ticket sales and reached the objective set for the following month.

5.2.2 EXAMPLE 2

A Supermarket that collects and stores in separate places its sales data and the weather forecast (in order to know when they have to cover their parking lot) decided to invest in infrastructure and implement a data model based on a data lake enabling the Supermarket to store all the data that they collect in one single place.

By storing all their data in one single repository, the Supermarket was able to, by processing and analyzing all their data together, find similar patterns between their popcorn sales and the weather forecast.

The Supermarket found out that the day before a major storm was announced, their customers tend to buy popcorn.

Therefore, the Supermarket started placing popcorn in front of the store and next to the cashiers right before a major storm was scheduled to hit.

By doing that, the Supermarket's sales increased.



5.2.3 EXAMPLE 3

A Supermarket that collects and stores its sales data as well as its customer's data decided to invest in infrastructure and implement a data model based on a data lake.

By doing that, the Supermarket was able to collect social media information on their customers and store that data in the lake in its original format. That data was thus quickly available for the Supermarket to access it, process it and analyze it in real time and hence understand what their customers might want to buy at that moment.

Therefore, the Supermarket was able to send to each customer personalized discounts on some products they might want to buy at that moment.

By doing that, the Supermarket's sales increased.



Chapter 6 DATA MODEL BASED ON A DATA LAKE IN A FINANCIAL ENTITY

Financial entities are always looking for ways to process and analyze all the data that they are collecting in order to find patterns, trends and associations between said data and turn it into information that will help them make the best decisions to improve their businesses.

By building a data model using Big Data technologies and tools, financial entities will be able to collect, store, process and analyze the data that is being generated and available to be collected with a:

- Large volume,
- High velocity,
- Wide variety.

By building a data model based on a data lake using Big Data technologies and tools, financial entities are able to:

- Collect all the data that they might need, whether it is structured data, semi-structured data or unstructured data and store all of it in one single repository;

By doing that, financial entities will be able to gain more insight into their data as they are able to process and analyze all their data together.

By gaining more insight into their data and thus gaining deeper insight into customers' data and behavior, financial entities can find new patterns, spot new trends as well as find new associations between their data. All this information will help financial entities understand new things about their businesses, their customers, their competition and their market. Therefore, financial entities will be able to make the best decisions to improve their businesses, manage their risk, develop new products and services, improve their customers' experience and promote their satisfaction and thus their retention as well as generate more revenue and new streams of revenue through, for example, personalized offers for their customers, targeted cross-sell and improved customer service, and hence be more efficient than their competitors.

- Store all the data that they are collecting in its original format in one single repository.



By collecting all the data they need and storing it in their data lake in its original format, financial entities will gain faster insight into their data by accessing to it, processing it and analyzing it in real time and thus find patterns, spot trends and find associations between their data in real time turning said data into information that will help them detect and prevent fraud in real time, manage their loan risk in real time, as well as take proactive decisions to improve their businesses.

By performing real time analysis of customers' information and behavior (such as spending patterns, product portfolio, bank interactions, credit information and social media), financial entities can get faster insight into their customer's information and behavior and therefore deliver real time offers.

On the other hand, financial entities handling European Union citizens' personal data must comply with the General Data Protection Regulation that became enforceable on 25 May 2018 as the main law regulating the way companies should use and protect European Union citizens' personal data. Companies that do not comply with the General Data Protection Regulation could be fined up to 4% of their total worldwide annual turnover of the preceding financial year or up to 20 Million Euros, whichever one is higher, for non-compliance.

Building a data model based on a data lake using Big Data technologies and tools will allow financial entities to manage, govern and secure their data and therefore make it easier on financial entities to comply with the General Data Protection Regulation and avoid penalties.

This fifth section of this thesis will define the architecture of the data model based on a data lake that is implemented in the financial entity. The way users of the financial entity's data model can request an ingestion in its data lake or request the access to data ingested in said data lake will be described afterwards. Finally, some problems related to the ingestion of data in the financial entity's data lake as well as some problems related to the accesses granted on said data as well as their solutions will be presented.

In the first part of this section, the architecture of the data model based on the data lake that is implemented in the financial entity will be described. The process that a user of said data model should follow to request either an ingestion of data in the data model's data lake or an access to some data ingested in said data lake will be illustrated afterwards. Last but not least, some post-implementation problems as well as their solutions will be presented.



6.1 ARCHITECTURE OF THE DATA MODEL BASED ON A DATA LAKE

In this sub-section, the architecture of the data model based on a data lake that is implemented in the financial entity will be presented.

This data model based on a data lake is implemented using Hadoop's components, some of Hadoop's ecosystem components as well as one of Cloudera's ready tools for the following main reasons:

- Hadoop and its ecosystem are suited to handle large volumes of data with various forms of structures allowing to collect, store, process, analyze and secure said data efficiently;
- By using some of Hadoop's ecosystem components as well as one of Cloudera's ready tools, financial entities can comply with the General Data Protection Regulation's requirements when handling European Union citizens' personal data;
- Hadoop and its ecosystem provide a highly scalable storage infrastructure;
- When processing and analyzing the data stored, Hadoop and its ecosystem run the program where the data is physically located instead of moving large amount of data around.

In order to present the architecture of the data model based on a data lake that is implemented in the financial entity, this sub-section will be divided into the following parts:

- Ingestion: This part will present the components in charge of ingesting the data into the data lake;
- Store: This part will present the components in charge of storing the data in the data lake;
- Index & Search: This part will present the component in charge of indexing and searching the data stored in the data lake;
- Manage & Secure: This part will present the components in charge of managing and securing the data in the data lake;
- Resource Management: This part will present the component in charge of managing the resources and scheduling jobs when processing the data stored in the data lake;
- Process & Analyze: This part will present the components in charge of processing and analyzing the data stored in the data lake.



6.1.1 INGESTION

In order for the financial entity to be able to collect and ingest the data in its data lake, the two following components of Hadoop's ecosystem are implemented:

- Sqoop,
- Flume.

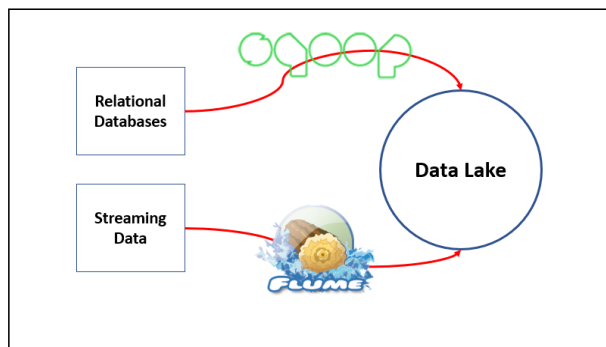


Figure 43: Collect & Ingest Data in the Data Lake

Sqoop will be used by the financial entity to collect and ingest data from relational databases into its data lake.

Flume will be used by the financial entity in order to collect and ingest streaming data into its data lake.

By using Sqoop and Flume the financial entity can collect structured, semi-structured as well as unstructured data and ingest said data into its data lake.

Ingestion of:	Data From Relational Databases	Streaming Data
Sqoop	✓	
Flume		✓

Figure 44: Sqoop vs. Flume

6.1.2 STORE

In order for the financial entity to be able to store all the data it collected in its data lake, the following components of Hadoop and its ecosystem are implemented:

- HDFS,



- HBase,
- Kudu.

These three components constitute the storage layer of the financial entity.
By using HDFS, HBase and Kudu, the financial entity can store in its data lake all the data it collected, whether it is structured, semi-structured or unstructured data.

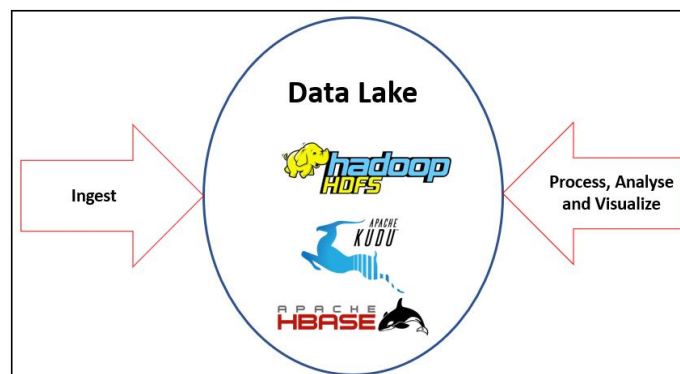


Figure 45: Data Storage

Each one of these three components provides some advantages and some disadvantages for the financial entity by storing the data collected in it:

- HDFS:
 - Provides a very good performance for scanning large amount of data stored in it.
 - Is not suitable for read/write random access to the data stored in it.
- HBase:
 - Provides a very good performance for read/write random access to the data stored in it.
 - Provides a very bad performance for scanning large amount of data stored in it.
- Kudu:
 - Provides a good performance for scanning large amount of data stored in it.
 - Provides a good performance for read/write random access to the data stored in it.



Data Model based on a Data Lake in a Financial Entity

	Scanning Large Amount of Data	Read/Write Random Access to the Data
HDFS	Very Good	✗
HBase	Very Bad	Very Good
KUDU	Good	Good

Figure 46: HDFS vs. HBase vs. Kudu

6.1.3 INDEX & SEARCH

In order for the financial entity to be able to index and search all the data stored in its data lake, the following component of Hadoop's ecosystem is implemented:

- SOLR.

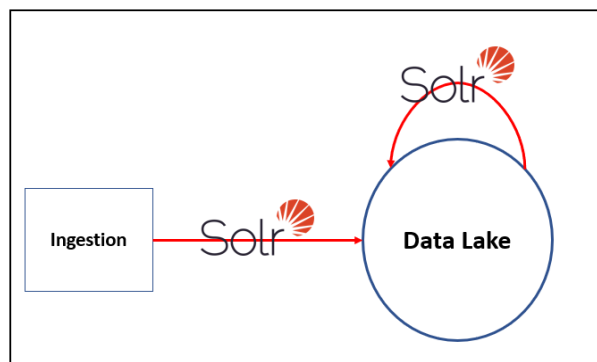


Figure 47: Index & Search the Data Stored

SOLR allows the financial entity to index all the data stored in its data lake which will be very useful when the data is going to be searched.

6.1.4 MANAGE & SECURE

In order for the financial entity to be able to manage, govern, secure and protect the data stored in its data lake, the following component of Hadoop's ecosystem is implemented:

- Sentry,

as well as the following Cloudera's ready tool:

- Cloudera Navigator.

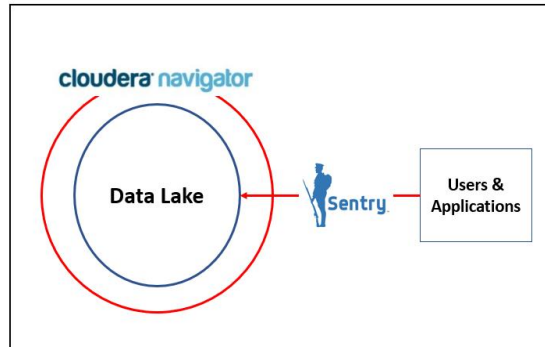


Figure 48: Manage, Govern, Secure and Protect the Data

By using Sentry, the financial entity can:

- Control and impose precise levels of privileges and permissions on the data and the metadata stored in its data lake for identified users and applications.

By using Cloudera Navigator, the financial entity can:

- Classify/tag and track personal data;
- Detect and analyze breaches;
- Encrypt and anonymize data.

By using Sentry and Cloudera Navigator, the financial entity can:

- Tag the data that can be accessed and impose a time limit for such access.

6.1.5 RESOURCE MANAGEMENT

To manage the resources and schedule jobs when the data stored in the financial entity's data lake is processed and analyzed, the following Hadoop's component is implemented:

- YARN.

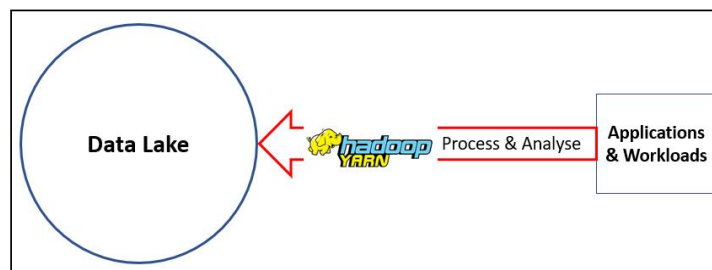


Figure 49: Manage Resources in the Data Lake

YARN is responsible of allocating the resources as well as scheduling the execution of jobs for the different applications or workloads processing and analyzing the data stored in the financial entity's data lake.

6.1.6 PROCESS & ANALYZE

In order for the financial entity to be able to process, analyze and visualize the data stored in its data lake, the following components of Hadoop and its ecosystem are implemented:

- MapReduce,
- Pig,
- Hive,
- Impala,
- Mahout,
- Spark,
- HUE.

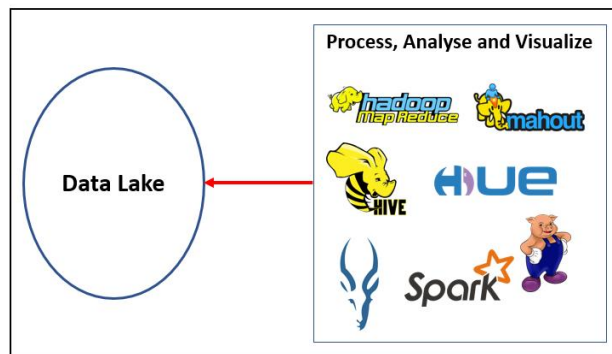


Figure 50: Process, Analyse and Visualize the Data

By using these components, the financial entity can process, analyze and visualize the data stored in its data lake:

- MapReduce is a programming model in charge of processing the data stored in the data lake.
- Pig is a high-level scripting language that can be used to write MapReduce programs to run against the data stored in the data lake without knowing Java nor Python.
- Hive allows to run SQL (which is converted to MapReduce) against the data stored in the data lake to read, write and manage said data.
- Impala allows to query and search the data stored in the data lake in real time (working around MapReduce and accessing directly the data) using easy SQL commands.
- Mahout is a library of scalable machine-learning algorithms running on top of MapReduce aiming to automatically find meaningful patterns in the data stored in the data lake.
- Spark allows to process and analyze large volumes of data stored in the data lake.



Spark allows to run programs against the data stored in the data lake faster than MapReduce. Spark also provides stream processing of the data collected by the financial entity as well as a fully-featured machine learning library.

- HUE is a web user interface that allows to query, analyze and visualize the data stored in the data lake using some of Hadoop's ecosystem components.

6.2 DATA INGESTION AND DATA ACCESS REQUESTS

In order for the financial entity's data model to contain all the data its users need as well as to protect the accesses to the data ingested in its data lake, the two following processes have been put in place and will be illustrated in this sub-section of this thesis:

- The first one describes the process that a user of the financial entity's data model should follow in order to request an ingestion of data in the financial entity's data lake;
- The second one describes the process that a user of the financial entity's data model should follow in order to request an access to some data ingested in the financial entity's data lake.

It goes without saying that if a user of the financial entity's data model needs to request an access to some data that is not yet ingested in the financial entity's data lake, he should:

1. Firstly, request the ingestion of needed data in the financial entity's data lake;
2. Then, request the access to said data in the financial entity's data lake.

This sub-section will therefore be divided into the two following parts:

1. Data Ingestion Request,
2. Data Access Request.

6.2.1 DATA INGESTION REQUEST

The user of the financial entity's data model can request the ingestion in said data model's data lake of either:

- Streaming data;
Streaming data are ingested in the financial entity's data lake using Flume.
- Data stored in databases;



Data stored in databases are ingested in the financial entity's data lake using Sqoop.

In both cases, the process a user of the financial entity's data model should follow in order to request an ingestion of data in said data model's data lake is the following:

1. Fill in the "Data Ingestion Request": The user requesting the ingestion of the data in the financial entity's data lake should fill in all the fields that concern him in the "Data Ingestion Request".
2. Submission to the Data Management Team: The user should submit the Data Ingestion Request filled to the Data Management Team in the financial entity.
3. Request Review & Decision: The Data Management Team will review the Data Ingestion Request sent by the user and make sure that all the fields that should be filled in by the user are correctly filled and that the ingestion of the data requested by the user is relevant to his needs.

If the Data Management Team approves the ingestion requested by the user, they should fill in all the fields that concern them in the Data Ingestion Request.

However, if the Data Management Team does not approve the ingestion requested by the user, one of the two following scenarios could occur:

- Data Ingestion Request wrongly filled: In case the Data Ingestion Request was not correctly filled by the user, the Data Management Team will send back said request to the user specifying to him the fields that are to be corrected;
 - Data Ingestion Request rejected: In case the Data Management Team consider that the ingestion of data requested by the user is not relevant to his needs, they will reject his request explaining to him the reason of that rejection.
4. Submission to the Data Ingestion Team: After approving the Data Ingestion Request, the Data Management Team should submit said request to the team responsible for the ingestions of data in the financial entity's data lake.

If the Data Ingestion Team have any doubt concerning the Data Ingestion Request, they should contact:

- The user for any doubt relevant to the information filled in by the user;
 - The Data Management Team for any doubt relevant to the information filled in by the Data Management Team.
5. Data Ingested: After the ingestion is realized, the Data Ingestion Team should:
 - Fill in the fields relative to them in the Data Ingestion Request;
 - Notify the user of the completeness of the requested ingestion.



6. User test: After being notified by the Data Ingestion Team of the completeness of the ingestion he requested, the user should make sure that the ingestion has been properly done.

In case the ingestion has occurred properly as he requested, the user should notify the Data Management Team as well as the Data Ingestion Team that the ingestion was successful. The Data Management Team will subsequently fill in the post-ingestion fields in the Data Ingestion Request.

If the ingestion has not occurred properly as he requested, the user should contact the Data Ingestion Team notifying them the reason why the ingestion has not occurred properly. The Data Ingestion Team will then proceed to correct the ingestion.

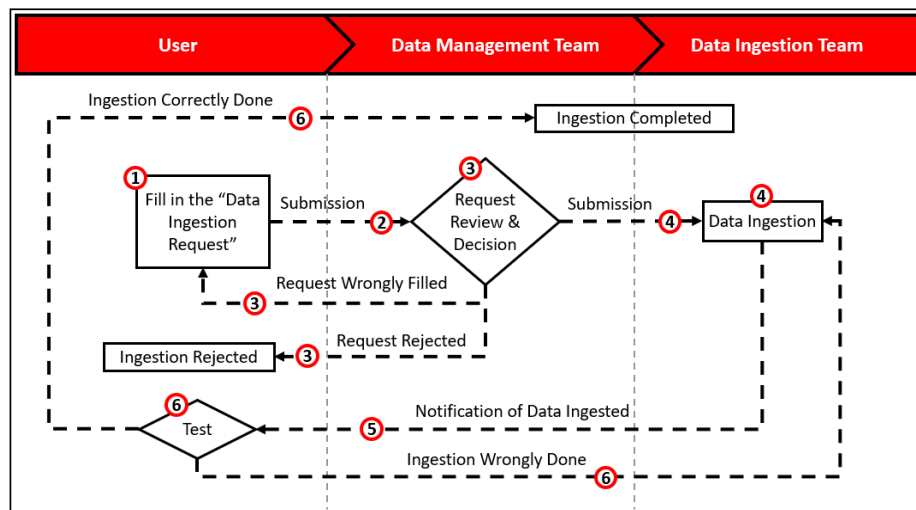


Figure 51: Data Ingestion Request Flow

The Data Ingestion Request is divided into the four following parts:

- The fields that should be filled in by the user requesting the ingestion;
- The fields that should be filled in by the Data Management Team before the ingestion has occurred;
- The fields that should be filled in by the Data Ingestion Team;
- The fields that should be filled in by the Data Management Team after the ingestion has occurred.

The fields of the Data Ingestion Request that should be filled in by the user requesting the ingestion are the following:

- Name - User: Name of the user or users requesting the ingestion of the data in the financial entity's data lake;
- User Department: Department of the user or users requesting the ingestion of the data in the financial entity's data lake;
- Person of Contact: The user or users to contact in case of any doubt relevant to the information filled in by the user;



Data Model based on a Data Lake in a Financial Entity

- Source: The source from where the data should be taken;
If the data is stored in databases, Sqoop will be used to ingest the data in the financial entity's data lake. However, if the data is streaming data, Flume will be used to ingest the data in the financial entity's data lake.
- Destination: Where the data should be stored in the financial entity's data lake;
- Description of the Information Ingested: The user should describe in two or three lines the content of the information he needs to ingest in the financial entity's data lake;
- Reason of Ingestion: The user should, in two or three lines, give the reason or reasons behind his need of ingesting the data in the financial entity's data lake;
- Frequency: The frequency the data should be ingested in the financial entity's data lake:
 - o One-off: The ingestion only occurs once;
 - o Yearly ingestion specifying the date: The ingestion will occur once a year on the date specified by the user;
 - o Monthly ingestion specifying the date: The ingestion will occur once a month on the date specified by the user;
 - o Weekly ingestion specifying the date: The ingestion will occur once a week on the date specified by the user;
 - o Daily ingestion: The ingestion will occur once a day.
- Time period of the ingestion: If the ingestion is not a one-off ingestion, the user should specify the date the recurrent ingestion should be stopped;
- GDPR Flag: The user should specify if the data ingested contain personal information.

The fields of the Data Ingestion Request that should be filled in by the Data Management Team pre-ingestion are the following:

- Authorization or Rejection: The Data Management Team should specify if the ingestion requested by the user is authorized or rejected;
- Date of Authorization or Rejection: The Data Management Team should indicate the date the user's ingestion request has been accepted or rejected;
In case of acceptance, this date would also be the date the Data Ingestion Request is sent to the Data Ingestion Team.
- Reason of Rejection: In case of rejection of the user's ingestion request by the Data Management Team, the latter should specify the reason or reasons behind said rejection;
- Name - Data Management Team: The name of the person from the Data Management Team accepting or rejecting the user's ingestion request.
This person will be contacted for any doubt relevant to the information filled in by the Data Management Team.



The fields of the Data Ingestion Request that should be filled in by the Data Ingestion Team are the following:

- Name - Data Ingestion Team: The name of the person from the Data Ingestion Team realizing the ingestion requested by the user;
This person will be contacted for any doubt relevant to the ingestion of the data, requested by the user, in the financial entity's data lake.
- Date of Ingestion: The date the ingestion or the first ingestion (in case the ingestion is not a one-off ingestion but a recurrent one) has been completed.

The field of the Data Ingestion Request that should be filled in by the Data Management Team post-ingestion is the following:

- Date of Approval by User: The date the ingestion or the first ingestion (in case the ingestion is not a one-off ingestion but a recurrent one) has been verified and approved by the user that requested said ingestion.

6.2.2 DATA ACCESS REQUEST

When the user of the financial entity's data model wants to request the access to some data stored in the financial entity's data lake, he should specify not only which data he wishes to access but also with which component of Hadoop's ecosystem used in the financial entity's data model he wishes to access said data.

The component used in the financial entity's data model to grant accesses to the data stored in the financial entity's data lake to users is Sentry.

Sentry is a role-based authorization module; in other words, a user or a group of users are not directly granted accesses to the data stored in the financial entity's data lake but acquire them through their role or roles.

Therefore, in order to grant accesses to the data stored in the financial entity's data lake to a user or a group of users requesting such accesses, the following process should be followed:

1. Create a role;
2. Grant the accesses requested by the user or users to this role;
3. Check if the user or users requesting the accesses to the data stored in the financial entity's data lake already form a group (called an Active Directory (AD) group);
4. If the user or users requesting the accesses to the data stored in the financial entity's data lake already form an AD group, join that group to the role created.

If the user or users requesting the accesses to the data stored in the financial entity's data lake do not already form an AD group, assign the user or users to a new AD group and then join that AD group to the role created.



The process a user of the financial entity's data model should follow in order to request the access to some data ingested in the financial entity's data lake using one or several of Hadoop's ecosystem components should be the following:

1. Fill in the "Data Access Request": The user requesting the access to data stored in the financial entity's data lake using a Hadoop's ecosystem component or components should fill in all the fields that concern him in the "Data Access Request".
2. Submission to the Data Management Team: The user should submit the Data Access Request filled to the Data Management Team in the financial entity.
3. Request Review & Decision: The Data Management Team will review the Data Access Request sent by the user and make sure that all the fields that should be filled in by the user are correctly filled and that the access to the data requested by the user is relevant to his needs.

If the Data Management Team approves the access requested by the user, they should fill in all the fields that concern them in the Data Access Request.

However, if the Data Management Team does not approve the access requested by the user, one of the two following scenarios could occur:

- Data Access Request wrongly filled: In case the Data Access Request was not correctly filled by the user, the Data Management Team will send back said request to the user specifying to him the fields that are to be corrected;
 - Data Access Request rejected: In case the Data Management Team consider that the access to the data requested by the user is not relevant to his needs, they will reject his request explaining to him the reason of that rejection.
4. Submission to the Data Access Team: After approving the Data Access Request, the Data Management Team should submit said request to the team responsible for the accesses to the data stored in the financial entity's data lake.

If the Data Access Team have any doubt concerning the Data Access Request, they should contact:

- The user for any doubt relevant to the information filled in by the user;
 - The Data Management Team for any doubt relevant to the information filled in by the Data Management Team.
5. Access Granted: After the access is granted, the Data Access Team should:
 - Fill in the fields relative to them in the Data Access Request;
 - Notify the user that the access to the data he requested has been granted.
 6. User test: After being notified by the Data Access Team that the access to the data he requested has been granted, the user should make sure that, by



Data Model based on a Data Lake in a Financial Entity

using the component or components of Hadoop's ecosystem that he specified in his request, the access to the data he requested has been properly granted.

In case the access has been properly granted as he requested, the user should notify the Data Management Team as well as the Data Access Team that the access was rightly granted. The Data Management Team will subsequently fill in the post-access fields in the Data Access Request.

If the access has not been properly granted as he requested, the user should contact the Data Access Team notifying them the reason why the access has not been correctly granted. The Data Access Team will then proceed to correct the problem.

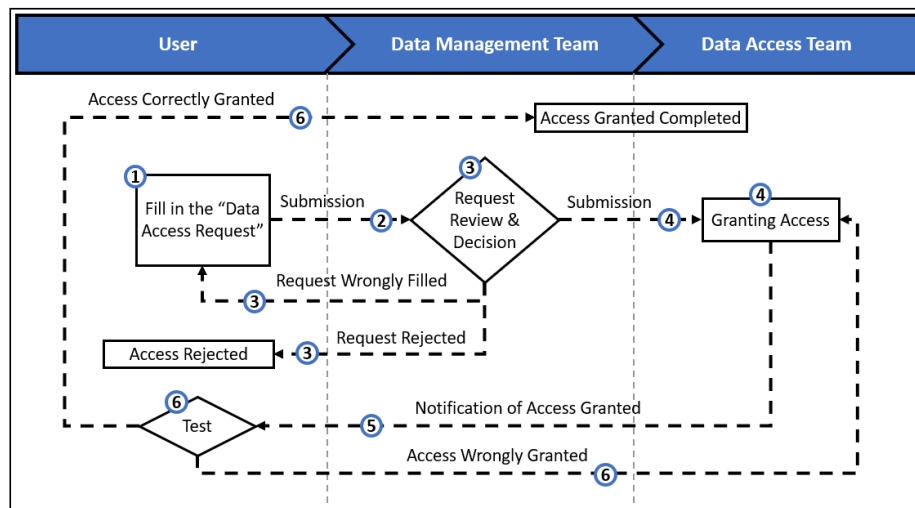


Figure 52: Data Access Request Flow

The Data Access Request is divided into the four following parts:

- The fields that should be filled in by the user requesting the access;
- The fields that should be filled in by the Data Management Team before the access has been granted;
- The fields that should be filled in by the Data Access Team;
- The fields that should be filled in by the Data Management Team after the access has been granted.

The fields of the Data Access Request that should be filled in by the user requesting the access to the data are the following:

- Name - User: Name of the user or users requesting the access to data stored in the financial entity's data lake;
- User Department: Department of the user or users requesting the access to data stored in the financial entity's data lake;
- Person of Contact: The user to contact in case of any doubt relevant to the information filled in this part of the Data Access Request;



Data Model based on a Data Lake in a Financial Entity

This user will also be the one to notify the Data Access Team and the Data Management Team if the access has been properly granted or not.

- Data Access: The data, stored in the financial entity's data lake, the user or users wish to access;
- Component Used: The component or components of Hadoop's ecosystem that the user or users wish to use to access the data;
- Reason to Access: The user or users should, in two or three lines, give the reason or reasons behind this request;
- Access Limited or Unlimited in Time: The user or users should specify if the access requested should be granted for a limited or unlimited time period.

The fields of the Data Access Request that should be filled in by the Data Management Team before the access has been granted are the following:

- Authorization or Rejection: The Data Management Team should specify if the access requested by the user is authorized or rejected;
- Date of Authorization or Rejection: The Data Management Team should indicate the date the user's access request has been accepted or rejected;
In case of acceptance, this date would also be the date the Data Access Request is sent to the Data Access Team.
- Reason of Rejection: In case of rejection of the user's access request by the Data Management Team, the latter should specify the reason or reasons behind said rejection;
- Name - Data Management Team: The name of the person from the Data Management Team accepting or rejecting the user's access request.

This person will be contacted for any doubt relevant to the information filled in by the Data Management Team.

The fields of the Data Access Request that should be filled in by the Data Access Team are the following:

- Name - Data Access Team: The name of the person from the Data Access Team granting the access requested to the user or users;
This person will be contacted for any doubt relevant to the access granted to the user or users to the data in the financial entity's data lake.
- Date of Access: The date the access requested has been granted to the user or users to the data in the financial entity's data lake.

The field of the Data Access Request that should be filled in by the Data Management Team after the access has been granted is the following:

- Date of Approval by User: The date the access granted to the user or users has been verified and approved by said user or users.

The user that should communicate this approval to the Data Access Team and the Data Management Team is the one whose name appears in the field "Person of Contact" in the Data Access Request.



6.3 DATA INGESTION AND DATA ACCESS PROBLEMS

After a requested access to data stored in the financial entity's data lake has been granted and validated by the user of the financial entity's data model or after the first ingestion of a recurrent requested ingestion of data in the financial entity's data lake has been done and approved by the user of the financial entity's data model, some incidents could occur.

In order for the financial entity to resolve these incidents, the two following processes have been put in place and will be illustrated in this sub-section of this thesis:

- The first one describes the process that a user of the financial entity's data model should follow in order to communicate an incident that occurred in a recurrent ingestion after the first one has been done properly;
- The second one describes the process that a user of the financial entity's data model should follow in order to communicate an incident that occurred in an access after it has been granted properly to the user.

This sub-section will therefore be divided into the two following parts:

1. Data Ingestion Error,
2. Data Access Error.

6.3.1 DATA INGESTION ERROR

The process a user of the financial entity's data model should follow in order to notify an incident that has occurred in a recurrent ingestion after the first one has been done properly and approved by the user that has requested it, is the following:

1. Fill in the "Data Ingestion Error" request: The user notifying the incident should fill in all the fields that concern him in the "Data Ingestion Error" request.
2. Submission to the Data Management Team: The user should submit the Data Ingestion Error request filled to the Data Management Team in the financial entity.
3. Request Review & Decision: The Data Management Team will review the Data Ingestion Error request sent by the user to make sure that:
 - o All the fields that should be filled in by the user are correctly filled;



- Comparing with the relevant Data Ingestion Request previously verified and approved, the incident reported by the user is truly an ingestion error.

If the Data Management Team approves the incident reported by the user, they should fill in all the fields that concern them in the Data Ingestion Error request.

However, if the Data Management Team does not approve the incident reported by the user, one of the two following scenarios could occur:

- Data Ingestion Error request wrongly filled: In case the Data Ingestion Error request was not correctly filled by the user, the Data Management Team will send back said request to the user specifying to him the fields that are to be corrected;
- Data Ingestion Error request rejected: In case the Data Management Team consider that, comparing with the relevant Data Ingestion Request previously verified and approved, the incident reported by the user is not an ingestion error, they will reject his request explaining to him the reason of that rejection.

4. Submission to the Data Ingestion Team: After approving the Data Ingestion Error request, the Data Management Team should submit said request to the team responsible for the ingestions of data in the financial entity's data lake.

If the Data Ingestion Team have any doubt concerning the Data Ingestion Error request, they should contact:

- The user for any doubt relevant to the information filled in by the user;
- The Data Management Team for any doubt relevant to the information filled in by the Data Management Team.

5. Ingestion Error Corrected: After the ingestion error has been corrected, the Data Ingestion Team should:

- Fill in the fields relative to them in the Data Ingestion Error request;
- Notify the user that the incident he reported has been resolved.

6. User test: After being notified by the Data Ingestion Team that the incident he reported has been resolved, the user should make sure that the ingestion error has been corrected.

In case the ingestion error has been successfully corrected, the user should notify the Data Management Team as well as the Data Ingestion Team that the incident he reported has been successfully resolved. The Data Management Team will subsequently fill in the post-resolution fields in the Data Ingestion Error request.

If the ingestion error has not been successfully corrected, the user should contact the Data Ingestion Team notifying them the reason why the

incident he reported has not been successfully resolved. The Data Ingestion Team will then proceed to correct the incident.

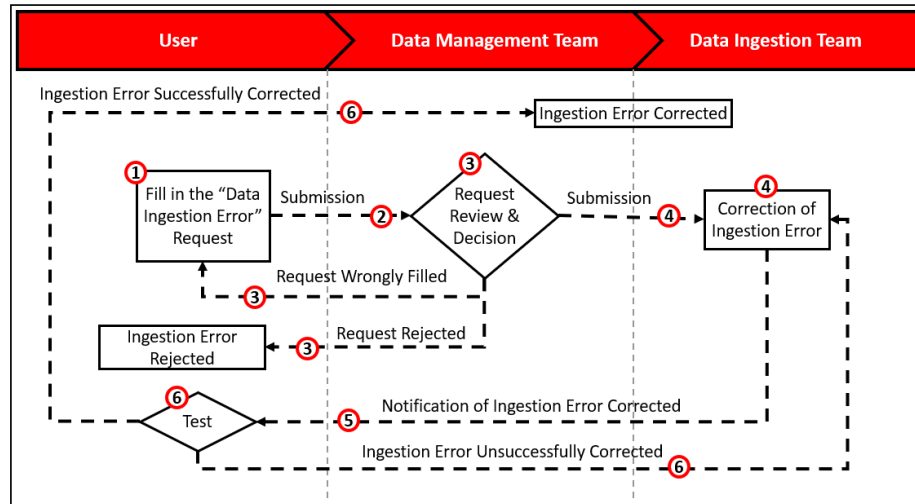


Figure 53: Data Ingestion Error Flow

The Data Ingestion Error request is divided into the four following parts:

- The fields that should be filled in by the user notifying the incident;
- The fields that should be filled in by the Data Management Team before the incident has been resolved;
- The fields that should be filled in by the Data Ingestion Team;
- The fields that should be filled in by the Data Management Team after the incident has been resolved.

The fields of the Data Ingestion Error request that should be filled in by the user notifying the incident are the following:

- Name - User: Name of the user reporting the ingestion error;
- User Department: Department of the user reporting the ingestion error;
- Ingestion Description: The user should describe the ingestion where the error has occurred;
 The accuracy of this description is very important for the Data Management Team to find the relevant Data Ingestion Request.
- Incident Description: The user should describe the incident he is reporting.

The fields of the Data Ingestion Error request that should be filled in by the Data Management Team before the incident has been resolved are the following:

- Authorization or Rejection: The Data Management Team should specify if the user's incident report is authorized or rejected;
- Date of Authorization or Rejection: The Data Management Team should indicate the date the user's incident report has been accepted or rejected;

In case of acceptance, this date would also be the date the Data Ingestion Error request is sent to the Data Ingestion Team.



- Reason of Rejection: In case of rejection of the user's incident report by the Data Management Team, the latter should specify the reason or reasons behind said rejection;
- Name - Data Management Team: The name of the person from the Data Management Team accepting or rejecting the user's incident report.
This person will be contacted for any doubt relevant to the information filled in by the Data Management Team.

The fields of the Data Ingestion Error request that should be filled in by the Data Ingestion Team are the following:

- Name - Data Ingestion Team: The name of the person from the Data Ingestion Team resolving the incident reported by the user;
This person will be contacted for any doubt relevant to the resolution of the incident.
- Date of Correction: The date the incident has been resolved.

The field of the Data Ingestion Error request that should be filled in by the Data Management Team after the incident has been resolved is the following:

- Date of Approval by User: The date the resolution of the incident has been verified and approved by the user that reported said incident.

6.3.2 DATA ACCESS ERROR

The process a user of the financial entity's data model should follow in order to notify an incident that has occurred in a previously granted access to data stored in the financial entity's data lake, is the following:

1. Fill in the "Data Access Error" request: The user notifying the incident should fill in all the fields that concern him in the "Data Access Error" request.
2. Submission to the Data Management Team: The user should submit the Data Access Error request filled to the Data Management Team in the financial entity.
3. Request Review & Decision: The Data Management Team will review the Data Access Error request sent by the user to make sure that:
 - o All the fields that should be filled in by the user are correctly filled;
 - o Comparing with the relevant Data Access Request previously verified and approved, the incident reported by the user is truly an access error.

If the Data Management Team approves the incident reported by the user, they should fill in all the fields that concern them in the Data Access Error request.

However, if the Data Management Team does not approve the incident reported by the user, one of the two following scenarios could occur:



- Data Access Error request wrongly filled: In case the Data Access Error request was not correctly filled by the user, the Data Management Team will send back said request to the user specifying to him the fields that are to be corrected;
 - Data Access Error request rejected: In case the Data Management Team consider that, comparing with the relevant Data Access Request previously verified and approved, the incident reported by the user is not an access error, they will reject his request explaining to him the reason of that rejection.
4. Submission to the Data Access Team: After approving the Data Access Error request, the Data Management Team should submit said request to the team responsible for the accesses to the data stored in the financial entity's data lake.

If the Data Access Team have any doubt concerning the Data Access Error request, they should contact:

- The user for any doubt relevant to the information filled in by the user;
 - The Data Management Team for any doubt relevant to the information filled in by the Data Management Team.
5. Access Error Corrected: After the access error has been corrected, the Data Access Team should:
- Fill in the fields relative to them in the Data Access Error request;
 - Notify the user that the incident he reported has been resolved.
6. User test: After being notified by the Data Access Team that the incident he reported has been resolved, the user should make sure that the access error has been corrected.

In case the access error has been successfully corrected, the user should notify the Data Management Team as well as the Data Access Team that the incident he reported has been successfully resolved. The Data Management Team will subsequently fill in the post-resolution fields in the Data Access Error request.

If the access error has not been successfully corrected, the user should contact the Data Access Team notifying them the reason why the incident he reported has not been successfully resolved. The Data Access Team will then proceed to correct the incident.

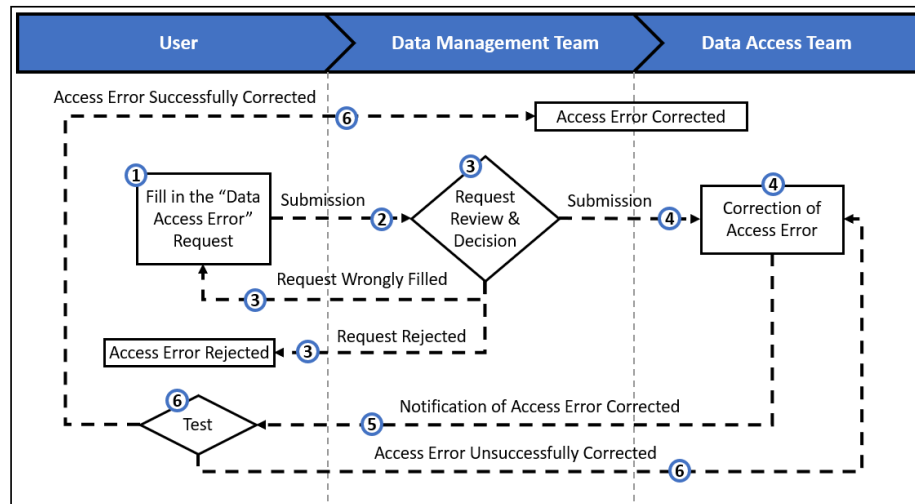


Figure 54: Data Access Error Flow

The Data Access Error request is divided into the four following parts:

- The fields that should be filled in by the user notifying the incident;
- The fields that should be filled in by the Data Management Team before the incident has been resolved;
- The fields that should be filled in by the Data Access Team;
- The fields that should be filled in by the Data Management Team after the incident has been resolved.

The fields of the Data Access Error request that should be filled in by the user notifying the incident are the following:

- Name - User: Name of the user reporting the access error;
- User Department: Department of the user reporting the access error;
- Access Description: The user should describe the access where the error has occurred;

The accuracy of this description is very important for the Data Management Team to find the relevant Data Access Request.

- Incident Description: The user should describe the incident he is reporting.

The fields of the Data Access Error request that should be filled in by the Data Management Team before the incident has been resolved are the following:

- Authorization or Rejection: The Data Management Team should specify if the user's incident report is authorized or rejected;
- Date of Authorization or Rejection: The Data Management Team should indicate the date the user's incident report has been accepted or rejected;
 In case of acceptance, this date would also be the date the Data Access Error request is sent to the Data Access Team.

- Reason of Rejection: In case of rejection of the user's incident report by the Data Management Team, the latter should specify the reason or reasons behind said rejection;



Data Model based on a Data Lake in a Financial Entity

- Name - Data Management Team: The name of the person from the Data Management Team accepting or rejecting the user's incident report.

This person will be contacted for any doubt relevant to the information filled in by the Data Management Team.

The fields of the Data Access Error request that should be filled in by the Data Access Team are the following:

- Name - Data Access Team: The name of the person from the Data Access Team resolving the incident reported by the user;

This person will be contacted for any doubt relevant to the resolution of the incident.

- Date of Correction: The date the incident has been resolved.

The field of the Data Access Error request that should be filled in by the Data Management Team after the incident has been resolved is the following:

- Date of Approval by User: The date the resolution of the incident has been verified and approved by the user that reported said incident.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Data Model based on a Data Lake in a Financial Entity



Chapter 7 CONCLUSION

The financial entity, by implementing a data model based on a data lake using Hadoop's components, some of Hadoop's ecosystem components as well as one of Cloudera's ready tools, is able to:

- Collect, store, secure, process and analyze all the data it might need from the data generated and available to be collected with a large volume, high velocity and wide variety within the time necessary for said data to be useful for the financial entity;
- Store in one single scalable storage repository all the collected data;
By storing all the collected data together, the financial entity is able to process and analyze all its data together and hence find patterns, spot trends as well as find associations between all its data turning it into information that will help the financial entity make the best decisions to improve its businesses.
- Store the collected data in its original format;
By storing the collected data in its original format, it becomes quickly available for the financial entity to access it, process it and analyze it in real time and thus find patterns, spot trends and find associations between its data in real time turning it into information that will help the financial entity make predictions and take proactive decisions.
- Manage, govern and secure its data and thus comply with regulations such as the General Data Protection Regulation and avoid penalties.

The financial entity, by establishing processes to be followed by the users of its data model in order to request an ingestion of data in its data model's data lake as well request the access to data ingested in said data lake, ensures that:

- Its data lake contains all the data the users need;
- The data ingested in its data lake is only accessible by the users that need it.

As the digital world is always evolving, it is very important for the financial entity to always be up to date on the new technologies and tools that are being developed in order to always have the most efficient data model.





Chapter 8 BIBLIOGRAPHY

- [1] “DIKW Model” [Online]. Available: <https://www.certguidance.com/explaining-dikw-hierarchy/> . [Last Access: July 2018]
- [2] “DIKW Model” [Online]. Available: <https://ontotext.com/knowledgehub/fundamentals/dikw-pyramid/> . [Last Access: July 2018]
- [3] “Types of Big Data” [Online]. Available: <https://www.knowledgehut.com/blog/big-data/types-of-big-data>. [Last Access: July 2018]
- [4] “Big Data” [Online]. Available: <https://www.upgrad.com/blog/what-is-big-data-types-characteristics-benefits-and-examples/>. [Last Access: July 2018]
- [5] “The “V” Model” [Online]. Available: <http://www.ibmbigdatahub.com/infographic/four-vs-big-data> . [Last Access: July 2018]
- [6] Tom White, Hadoop: The Definitive Guide, Fourth Edition, O’Reilly Media, Inc., 2015.
- [7] “Hadoop” [Online]. Available: <https://searchdatamanagement.techtarget.com/definition/Hadoop>. [Last Access: July 2018]
- [8] “Hadoop” [Online]. Available: <https://readwrite.com/2018/06/26/complete-guide-of-hadoop-ecosystem-components/>. [Last Access: July 2018]



- [9] “Cloudera” [Online]. Available:
<https://www.cloudera.com/documentation/enterprise/5-8-x/topics/introduction.html> . [Last Access: July 2018]
- [10] “Hadoop” [Online]. Available:
<https://intellipaat.com/blog/what-is-hadoop/>. [Last Access: July 2018]
- [11] “Hadoop’s Ecosystem” [Online]. Available:
<https://data-flair.training/blogs/hadoop-ecosystem-components/>. [Last Access: July 2018]
- [12] “Hadoop Cluster” [Online]. Available:
<https://data-flair.training/blogs/hadoop-cluster/>. [Last Access: July 2018]
- [13] “Sqoop” [Online]. Available:
<https://sg.com.mx/revista/55/exportando-sistemas-relacionales-sistemas-columnares-distribuidos>. [Last Access: July 2018]
- [14] “Flume” [Online]. Available:
<https://flume.apache.org/FlumeUserGuide.html>. [Last Access: July 2018]
- [15] “General Data Protection Regulation” [Online]. Available:
<https://gdpr-info.eu/> . [Last Access: August 2018]
- [16] “General Data Protection Regulation” [Online]. Available:
https://www.iocea.com/blog/gdpr_enforcement/. [Last Access: August 2018]
- [17] “General Data Protection Regulation” [Online]. Available:
<https://www.eugdpr.org/> . [Last Access: August 2018]
- [18] “Data Lake” [Online]. Available:
<https://blogs.oracle.com/bigdata/whats-a-data-lake> . [Last Access: August 2018]
- [19] “Data Lake” [Online]. Available:
<https://searchaws.techtarget.com/definition/data-lake> . [Last Access: August 2018]
-