
POLITECNICO DI TORINO

DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING (DAUIN)

Master's Degree in Computer Engineering

Master's Degree Thesis

# Implementation of policies in Data-Centric Security Solutions: A Case Study

Author: Luca Ponchione

Advisor: Paolo Ernesto Prinetto
Co-Advisors: Nicoló Maunero, Luca Marcelli

April 2023

# Abstract

One of the most crucial aspects that a company must take into account is the data it has to manage. So as organizations' endpoints evolve and the use of cloud solutions increases, it is important to introduce a new approach to improve the security and privacy of the data being handled. In addition to providing these capabilities, the paradigm shift to cloud solutions allows for easier data management and lower costs for organizations, which translates into gains in both economic terms and efficiency for both the provider and the user of the service. The application environments of Data-Centric Security (DCS) are varied and range from the military to healthcare to IoT device management, even being able to intersect with each other. The very fact of applying security to the data, as well as to the platform or device in use, makes the whole system more flexible and because of this, the applicability varies greatly. Metadata is fundamental to practical application, whether data or users, in the former case describing the resource by making its object explicit, defining its structure and relationships with other resources, indicating who can use it, how and for how long, while in the latter case, they assign roles or attributes to enable the implementation of Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC). DCS makes it easy for companies to implement the European GDPR in a way that ensures that citizens have fair and correct use of their data throughout the entire lifecycle: by implementing security policies and using metadata they can ensure these capabilities. GDPR represents in the field of regulations a very modern, user-centred approach, and although it has weaknesses that can be improved, its strengths summarized by the seven basic principles it represents have been copied worldwide. Cryptography does not become second nature because it always retains the key role of ensuring confidentiality, data integrity, authentication, access control and non-repudiation, so the use of Data-Centric Security should be seen as an additional layer of security to the entire system. Nowadays there are several frameworks and tools on the market to take full advantage of the potential of this data-centric approach, and I can interface with other existing tools to have 360-degree control over data management to ensure data integration, data governance and monitoring. This thesis work outlines the practical applications of DCSs and technical information on the frameworks and platforms that adopt them, along with the various hardware components, such as logical gateways and proxies, and application models, such as Extract, Transform and Load (ETL), to effectively apply the DCS paradigm. The topic covered in this thesis comes from a 900-hour internship at PricewaterhouseCoopers Business Services S.r.l. during which the topic of Data-Centric Security was addressed using Apache Ranger, a framework for enabling, monitoring and managing comprehensive data security on the Hadoop platform. Central to the use of this tool has been the analysis,

implementation and testing of security policies of access control, dynamic data masking and row-level filtering used as a base to build more complex ones, both with automatic tools using REST APIs and with the User Interface (UI). The thesis also discusses the architecture of Apache Ranger itself with the various components with which it can interact, of which only Hive was used in the project, to get a lower-level understanding of how the framework works showing its inner functioning.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

The world of information is constantly evolving, so it is necessary to stay abreast by adopting useful approaches to simplify the management of the data itself and its security. While it used to be thought to protect the network in which the data resides, today we prefer to protect the data directly: this is the so-called data-centric approach. Underlying this thinking is the idea of bringing together the pillars of security controls, such as access management, Data Loss Prevention and archiving: they are no longer left at the extremes of the network perimeter but are brought closer to the data itself [38].

In recent years, there has been a paradigm shift due to the increased use of cloud solutions by companies: thus, there has been a shift from owning computing systems to buying computing services. This has enabled organizations to gain important benefits, including decreased costs and greater ease and flexibility in managing the huge data streams under their control. From the point of view of securing resources, it is necessary to go beyond a preventive approach, relying also on an investigative approach: this means that it is not enough to use better encryption algorithms or firewalls on networks, but also to focus on analysis tools, IDS and logs, to ensure that security or privacy risks do not violate policies [42].

The cloud environment is not the only one affected by this change, but the growing development of the Internet of Things has prompted a focus on data-centric security in this area as well, so that appropriate policies can be implemented to protect data, throughout its entire lifecycle [39]. Indeed, it should be noted that Data-Centric Security should not be relegated to single domains, but can be placed side by side with a wide variety of environments, from the military to healthcare. This flexibility is allowed precisely because we focus on data security and privacy, regardless of the architecture and structure around it. Moreover, the cloud approach guarantees the merit of leaving the service provider the task of managing the security of the platform, the scalability of resources and the size of the storage, not to mention the economic benefits of not having to deal directly with these factors.

Having understood the starting point, it becomes clear how the most vulnerable party is the user himself because once his data is shared, he has no idea who is handling it and how. The properties of transparency, verifiability, and accountability need to be ensured so that the data are the focus of attention and security and so are the users [20]: all of this should not be done only by looking at it from a strictly economic point of view so as not to

ruin the reliability of the company, but it should be implemented with modern data use in mind while respecting the rights of the citizen, as required by the 2016 European GDPR [75]. Precisely for this reason, the changes described so far should not be interpreted as an isolated phenomenon, but must be seen accordingly with the sharp increase in digital data in circulation, the management of which is a source of problems for companies, that must guarantee privacy and security.

The basic idea of the network itself is to share data across it, and cloud solutions help to achieve this purpose by putting scalability and manageability on the plate. From this model, it is, therefore, necessary to note how the concept of data ownership is separate from that of data administration. The security issue is not only about the fact that the data cannot be shown to unauthorized parties by guaranteeing its confidentiality, but it is also necessary to assure its integrity so that no one, not even the manager of the cloud or storage where the data resides, can alter it. In this way, using Data-Centric Security (DCS) the data describes and protects itself [14], ensuring its security even in more decentralized environments [20]. It is therefore essential to reconcile the classic cryptographic practices, updated in the face of the modern calculation power of computers, and the data-centric approach, which therefore serves as an additional level of security and not a replacement. Cryptography remains at the heart of the importance of ensuring the security of transmissions and the architecture that resides around the data since the paradigm shift is focused on the data and not the system it resides. From this point of view, the introduction and use of metadata become fundamental, not only as ancillary information to the data itself but as information containing data to manage the security and privacy of the resource. The metadata can be divided into descriptive, structural and administrative according to the use that is made of it.

The following work is based on the topics covered during the internship at PricewaterhouseCoopers Business Services S.r.l.[1]: the purpose of this thesis is to show the benefits of data-centric security, as well as some of its shortcomings, its implementation architecture and possible future developments. In particular, it also describes the operation of Apache Ranger, a framework used during the project, starting from its architecture with all its internal components, up to the logic of implementation of security policies, achievable both through a graphical interface (UI) and through REST API, with the basic methods GET, POST, DELETE and PUT. The use of security policies was necessary to ensure access control to the client's corporate systems, dynamic data masking to comply with GDPR rules and row-level filters based on the user's legal entity. To have a more operative vision of the situation, the problems present in the life cycle of the management of the data are addressed, therefore the processing, the storage and the analysis going to speak about the frameworks that currently go to work on these topics, ensuring the properties of Data Integration, Data Governance and Monitoring. The main Big Data technologies discussed are Hadoop, Spark, Flink and Hyarcks.

This document is organised as follows: in Chapter 2, the specific topics covered are explained to provide a technical basis for the subsequent sections, addressing cryptography, GDPR and some Data-Centric Security notions; in Chapter 3, the state of the art of the data-centric security paradigm is presented, along with currently adopted frameworks and

[1] https://www.pwc.com/it/it/

their properties to solve the main problems encountered; in Chapter 4, the work done during the internship is addressed together with the tools used and the architecture of the environment that are illustrated; in Chapter 5, possible future developments and problems of the data-centric paradigm are discussed.

# Chapter 2

# Background

This chapter discusses the main issues useful for a better understanding of the thesis, to provide a foundation of the necessary topics: the central theme of Data-Centric Security is explained along with a smattering of Cryptography and some background on the European Data Protection Regulation (GDPR).

## 2.1 Data-Centric Security

The focus of the discussion in this thesis is Data-Centric Security (DCS), which is designed to solve the problems described in Chapter 1. Thus, it can be summarized that DCS is used to secure data at all times based on its value, regardless of how it is saved or transmitted.

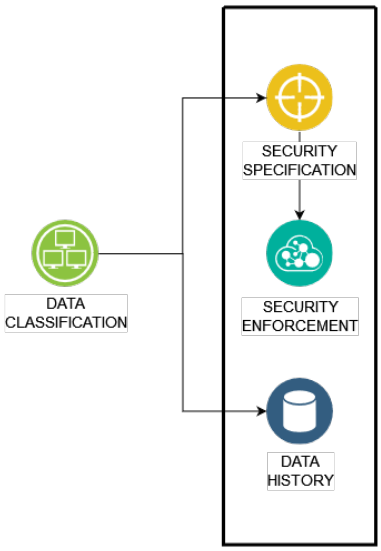


Figure 2.1. Conceptual division of Data-Centric Security (DCS)

Conceptually, Data-Centric Security can be divided into four parts, as described in

Figure 2.1 [59]:

- **Data Classification** is specific to individual environments and organizations, and it is important to use standard approaches to simplify data management
- **Security Specification** is critical to meet legal requirements by following directives governing the flow of data
- **Security Enforcement** complements the previous point by implementing the identified specifications in a way that achieves the security goals
- **Data History** provides a clear and effective way to manage the entire data lifecycle, enabling quality assurance and value in the form of metadata

Having talked about data so far, it is also necessary to consider that it comes from multiple sources, such as laptops, mobile phones, clouds and so on, so that it is easy to understand how its management is highly complex and onerous. Classifying data is not just about deciding whether it can be public or should be confidential, but it is important to know where it comes from and where it is stored [38].

A practical implementation of Data-Centric Security can be done through various methods such as **Gateways**, **Extract, Transform and Load (ETL)** and **Reverse Proxy**.

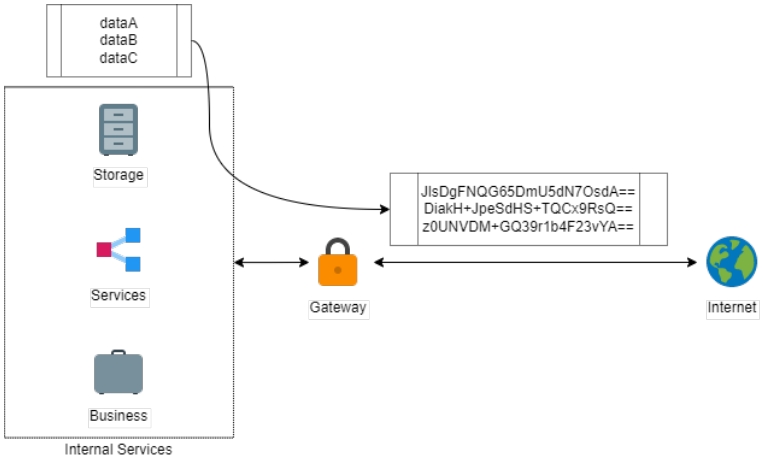


Figure 2.2. Example of gateway Functionalities

Gateways are electronic systems whose purpose is to communicate with external systems, that is any application on the Internet that offers services via a protocol (e.g., HTTP), and process data for analysis, aggregation or security measures. They are useful in this case to manipulate/obfuscate data when directed to third-party service providers (e.g., applying masking by encrypting it) [56], as shown in Figure 2.2.

Extract, Transform and Load (ETL) is the process of collecting data from an unlimited number of sources and their subsequent organization and centralization in a single repository ensuring the masking and encryption of sensitive data. Figure 2.3 illustrates the ETL

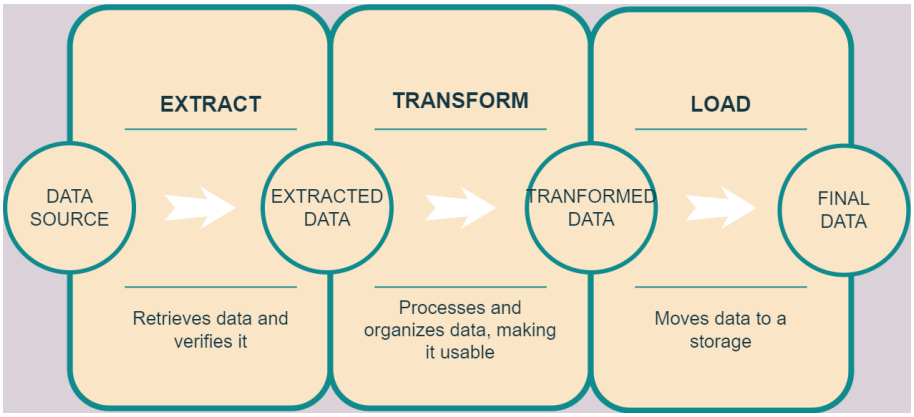


Figure 2.3. ETL scheme

schema, wherein the extraction part, data is obtained from different sources as full or differential copies. Before being loaded into storage, the data is transformed, which consists of two main operations: filtering or cleaning operations, such as searching for primary key violations, and transformation operations, such as data aggregation [72].

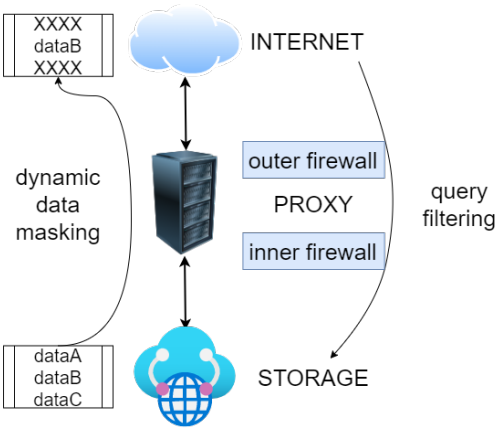


Figure 2.4. Example of Reverse Proxy Functionalities

Reverse Proxy is similar to gateways but enables operations such as dynamic masking (on-the-fly) directly between the Internet and database, where the latter is not altered [38]. This acts as a shield against all access that is executed, allowing the filtering of requests made by a user (e.g., SQL query) and thus isolating the server to provide an additional layer of security by using an inner firewall and an outer firewall with the reverse proxy placed in the Demilitarized Zone (DMZ) [65].

## 2.2 Cryptography

The two strong points for data security are encryption and masking: while the first provides protection, both in transit and in storage, the latter adds a layer of obfuscation. Whether you are going to deploy an authentication model or want to inherently protect data, cryptography is not an area of secondary importance. It permits implementing confidentiality, data integrity, authentication, access control and nonrepudiation [80]. Starting with the definition of plaintext, which is a sequence of characters in an alphabet, it is important to make an initial subdivision of cryptosystems into stream and block cyphers. While the former goes character by character as a continuous stream of data (much used in World War II crypto machines), the latter focuses on blocks of characters depending on the algorithm used: generally, these are considered cryptographically stronger [64]. This is not to argue that stream algorithms are to be considered obsolete, as they possess a great advantage in execution speed [36].

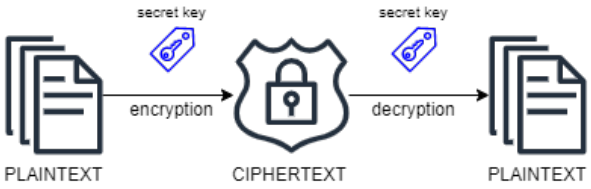


Figure 2.5. Concept of Symmetric Encryption

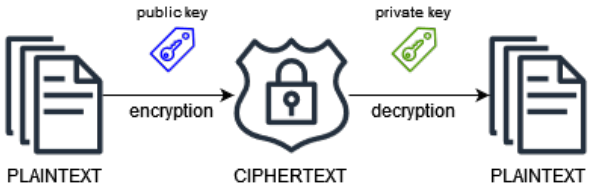


Figure 2.6. Concept of Asymmetric Encryption

No less important is the division between symmetric and asymmetric encryption: symmetric because the secret key is the same for encrypting and decrypting and must be shared between sender and receiver as shown in Figure 2.5, while asymmetric because there are two different keys for encrypting and decrypting, a public key and a private key, as can be seen in Figure 2.6. Both have pros and cons, but in general, symmetric encryption is almost 1,000 times faster but poses a problem for the security of secret key sharing [80]. The most widely adopted algorithms for symmetric encryption are the Advanced Encryption Standard (AES) (block cyphers) and RC4 (stream cypher), which are used in various protocols such as SSH File Transfer Protocol (SFTP), File Transfer Protocol Secure (FTPS) and HyperText Transfer Protocol over Secure Socket Layer (HTTPS) [73]. For asymmetric encryption, the most widely used algorithms are RSA and Digital Signature Algorithm

(DSA), adopted for models such as Digital Signature and protocols including Pretty Good Privacy (PGP) and Transport Layer Security (TLS) [6].

| | Algorithm | Status |
|---|---|---|
| Encryption | 3DES Encryption | Disallowed after 2023 |
| | AES-128 | Acceptable |
| | AES-256 | Acceptable |
| Digital Signatures | ECDSA | Acceptable with the key length greater than 224 bits |
| | RSA | Acceptable with the key length greater than 2048 bits |
| Key Derivation Functions | HMAC using any approved hash function | Acceptable |
| | CMAC using AES | Acceptable |
| Hash Functions | SHA-1 | Disallowed with exceptions |
| | SHA-2 Family | Acceptable |
| | SHA-3 Family | Acceptable |

Table 2.1. NIST Cryptographic Key Management Guidance

Table 2.1 shows an important aspect to consider, namely how it is necessary to be up-to-date on the security of the algorithms used and the lengths of the related keys. This is because the robustness of algorithms in terms of security is not absolute, but decreases over time as computers become more powerful or as they are broken. For example, NIST, National Institute of Standards and Technology, periodically provides to issue guidance on how to properly implement key management and use algorithms that adequately protect data [15]: in addition, in January 2023, the United States passed legislation [1] to migrate federal systems to post-quantum cryptography (PQC), which can withstand attacks from the now upcoming quantum computers [61].

## 2.3 GDPR

The development of the General Data Protection Regulation (GDPR) with effective operation in 2018 was a key step in underscoring the importance of data in the digital world [75], and thus the need to create rules to enable public administrations and companies to protect users, i.e., those directly affected. Ensuring privacy and security in data management is the cornerstone principle of this regulation, and its importance has not been isolated to the European Union (EU) alone but has resonated globally: countries such as India [57], Brazil [26], and South Korea [17] have introduced new regulations taking their cues from the GDPR.

Figure 2.7 shows the seven basic principles that constitute the GDPR [75]:


Figure 2.7. The seven basic principles of the GDPR

- **Lawfulness, Fairness and Transparency** describe how data handling should be lawful, fair and transparent to the user.
- **Integrity and Confidentiality** represent the need to process data while ensuring its security
- **Accountability** specifies the responsibility of the data controller to ensure that all principles are complied with
- **Data Minimization** lays out the need to take only the data that are required for the given purpose
- **Purpose Limitation** indicates that when the user shares data should be informed of the purpose for which they will be processed
- **Accuracy** identifies the need to maintain data up-to-date and correct
- **Storage Limitation** depicts the necessity of storing identifying data only for the specific time needed

Focusing on a more practical point of view, it can be seen from a 2020 case study derived from the use of a data mining tool on corporate privacy policies that in general companies have improved their policies to comply with the GDPR, showing that this regulation has had a real impact on the security and privacy of European citizens' data. In any case, the GDPR should not be seen as an endpoint, but seen as a starting point toward improving the regulation itself, starting with actually allowing users to manage, modify, and delete their data [81] and resolving disputes with non-European states such as in the case of the

United States (US), where the 2020 ruling by the Grand Chamber of the European Court established how the US is deficient in the protections established by European laws [70].

# Chapter 3

# Related Work

The benefits of Data-Centric Security should not only be observed in the privacy that is achieved on user data, but also in the security of the organization in which it is used. No less important to consider is the cost of implementing and managing the architecture used to effectively implement the Zero Trust Model (ZTM): this approach is to consider any network, whether internal or external, to be trustless. Any access to the organization's resources is a potential threat, so it must always be authorized and controlled [18]. Other strategies for applying ZTM can also be considered, such as micro-fragmentation, the purpose of which is to divide the network into smaller logical segments so that only authorized endpoints can access resources [51]. This leads to greater granularity and thus more checkpoints to authenticate and authorize, but mainly for large companies, there comes a threshold beyond which costs become too high as seen in Figure 3.1.

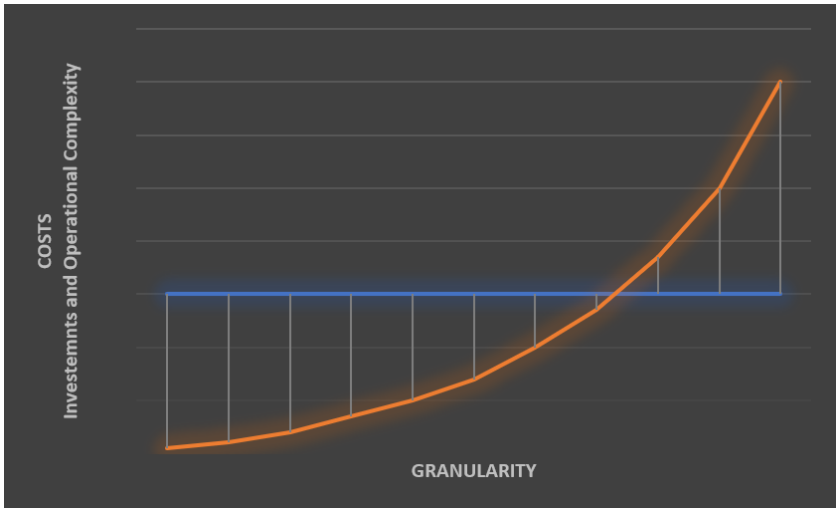


Figure 3.1. Qualitative representation of the relationship between cost and granularity

Therefore, using Data -Centric Security not only allows for high granularity since security is applied to the individual data, but this ensures that costs are not exponential

because equipment upgrades and operational complexity are no longer necessary [50].

The U.S. multinational Gartner Inc. [1] which has always been active in research and analysis in the field of information technology, believes that the use of Data-Centric Security has countless benefits, including:

- Mitigating data breaches
- Allowing organizations to use IT services and vendors safely
- Managing stored data and how it is protected
- Complying with regulations
- Assessing the risks to data, then prioritizing protection

Indeed, the best practices suggested for the optimal use of DCS are [37]:

- Ensure protection from unauthorized access by implementing a security infrastructure whose purpose is to provide security and privacy for data
- Have continuous control over how data is used by setting up an auditing system
- Manage resource access permissions using encryption keys, properly handled
- Use secure deletion techniques for data no longer used, such as crypto-shredding and physical destruction of storage

Data-Centric Security is a paradigm that is not isolated to a specific environment but can be generalized to even very different domains with in common the interest of protecting the data they have to manage. In the modern world it is mainly adopted in the generic cloud environment, but more specifically also in the Internet of Things (IoT) field, it can be a strong point for related security [39]. IoT steadily increases, thus growing the total amount of data, but it is necessary to note that there is a huge weakness: IoT devices do not very often follow the principle of security-by-design, starting from electric vehicle chargers to smart thermostats, this is because manufacturers aim to save on cost and production time [5]. This is precisely why adopting a Data-Centric Security approach is an optimal solution that can circumvent the weaknesses of today's IoT devices.

While IoT simplifies many daily actions, security and privacy must also be considered: focusing on the applications it can have on healthcare, whose information can range from blood pressure to heart rate that, being sensitive data, must be kept safe. Currently in this field, the key points to be addressed to reach a Data-Centric Security and privacy model for intelligent IoT (Figure 3.2) are as follows [39]:

- Define the characteristics of data-centric IoT after the examination of various standards
- Define a threat model based on IoT feature analysis

[1] https://www.gartner.com/en

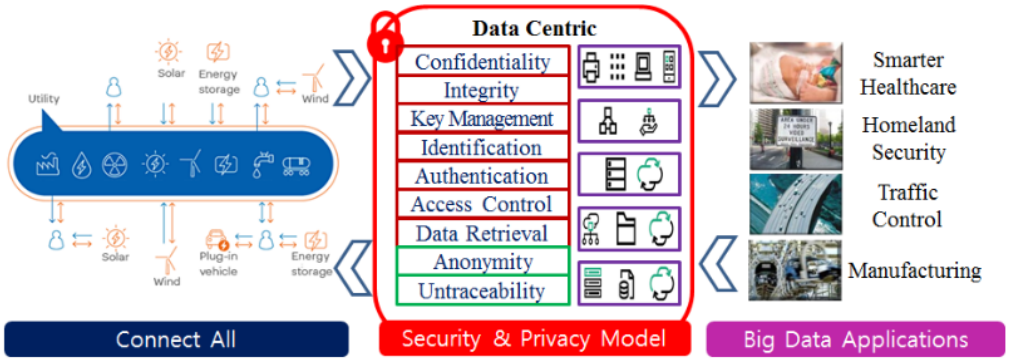


Figure 3.2. Conceptual diagram of Data-Centric security and privacy model for intelligent IoT

- Develop a data-centric privacy and security model according to IoT characteristics studies
- Develop an attribute-centred access control method emphasizing the importance of least-privilege access
- Use homomorphic encryption to ensure confidentiality and privacy [67]
- Use data-centred anonymity and untraceability techniques by eliminating per-session connectivity and proposing selective anonymity by focusing on data users' privileges

Notice also the importance of the centrality of data not only from a security perspective but also for a universal and predictive digital health ecosystem, as envisioned by the United Nations Sustainable Development Goal (SDG) number 3 [32]. This would support national and international management of health risks [82].

No less crucial is the military, for which the need to maintain high standards of security and privacy is ever-present. It can be seen that in every field commercial IoT components play an important role due to their low cost, so even in the military environment they have gained relevance in recent times. But while in this respect they allow for intensive use due to price effectiveness, the same cannot be said about the security adopted on these devices due to both the previous reason and the user-friendliness that accompanies them, which is often at odds with military security standards. The very use of public servers that these devices rely on is a source of vulnerability. One possible approach at this point is to strengthen the infrastructure with the use of stronger cryptographic keys along with digital signature and homomorphic encryption: at the same time using data-centric security can also have an important impact. If encryption protects information at a low level indiscriminately, DCS directly protects the data by applying security rules according to its type [76].

From a practical point of view, an example of DCS implementation is Content-based Protection and Release (CPR), developed by NATO [77]. This is based on the use of two

Attribute-Based Access Control (ABAC): release policy and protection policy. While the former manages who can access the content, the latter determines how the information should be secured. The main purpose of this model is to allow sensitive data to be shared with third-party organizations or coalition partners without violating confidentiality. Authorization decisions are then made based on user attributes, which can range from identity to military rank.

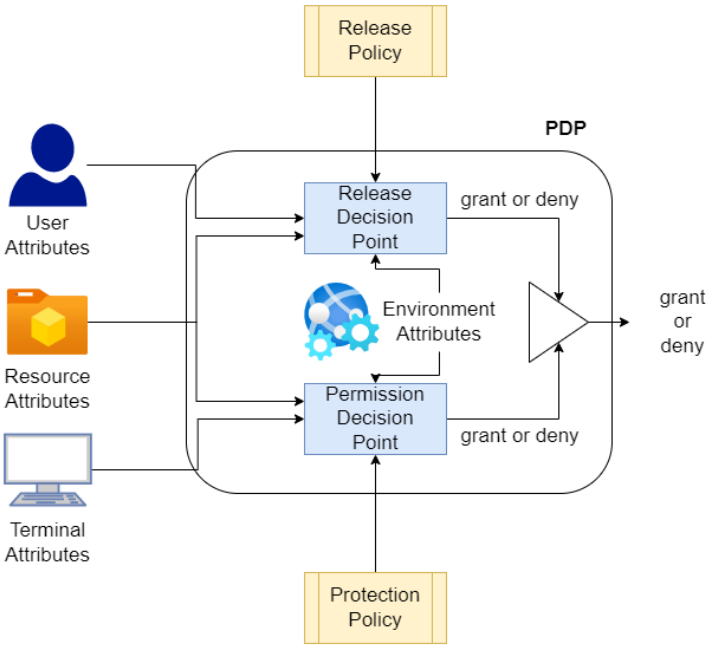


Figure 3.3. CPR Model

The Policy Decision Point (PDP) that fully describes the RCP model is represented by Figure 3.3: on the one hand, it is checked that the user, resources and environmental attributes comply with the release policy, while on the other hand, it is checked that the terminal, resources and environmental attributes respect the protection policy. The PDP allows the user access to the resource only if both conditions are *"grant"*. Applying this model in the field, DCS can be used, for example, in NATO's Passive Missile Defense (PMD) system [12], the purpose of which is to minimize the impact of a missile attack: by simulating the attack, the consequences of the impact and the trajectory can be predicted. At this point, it is critical to share this information with coalition members to attempt to intercept the missile and with civilian organizations such as the International Committee of the Red Cross (ICRC) to mitigate damage in the event of a successful attack. The issue of authorization is of utmost importance: if a NATO user can afford to access all the information, the same cannot be said for an ICRC user who will not be able to know the trajectory, thus being able to deduce where the attack came from. The rules on how users manage, store, and transmit data are described by terminal attributes [13].

Expanding the theme to devices used daily by most citizens, Data-Centric Security also applies well to the topic of smartphones, the use of which continues to grow with 6.259 million users in 2021 [66]. The main threats to smartphones are not only malware such as viruses and ransomware but also, for example, in case they are stolen or lost, the data inside them must continue to be protected. Equally, consideration must be given to the transmission of data in addition to its proper storage: DCS helps here to ensure confidentiality, integrity and privacy for smartphones. Compared to PCs and laptops, these devices tend to suffer from the fact that they are less secure [48], but they are also portable and interconnected to the environment around us, which is also one of their merits: the variety of connections they can entertain are many and can be threat vectors, such as WI-FI [45], Bluetooth [31], mobile data connection [27] and NFC [3].

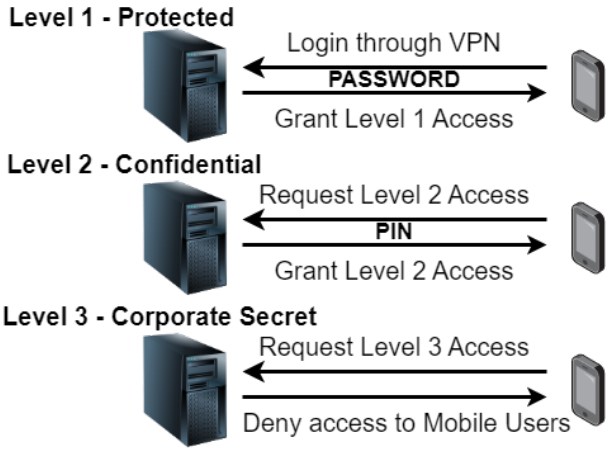


Figure 3.4. Example of user authentication through a smartphone

A more practical example can be explained through the user's action of logging in from a smartphone via a VPN to the corporate network: thinking with the standard model, the user will log in via single-factor authentication, then via a password. At this point, he will have permission to access information on the corporate network. In a more realistic environment, the company will have its information systems divided into multiple layers like in Figure 3.4 based on data confidentiality: thus the user will need pin codes or physical tokens to access higher layers. The authentication phase itself is thus a point of vulnerability: passwords and codes must pass through the network and VPN, which, while cryptographically secure, equally raises security questions since this encrypted information is still exposed. The flexibility of this model is also suboptimal because a file containing only some confidential data will have to be moved entirely to a higher security level. This can be avoided through the use of DCS, which acts directly on the data and allows greater flexibility in this aspect. So the user will no longer need to use passwords or physical tokens, but internally within the company will be associated with a role (e.g., tax audit) and a permission level (e.g., confidential), and then a decision point in the corporate network will decide based on this information and implemented policies whether or not there is permission to access the particular resource [24].

Since recent years, the meaning of the Fourth Industrial Revolution has become increasingly important: if the First led agricultural-craft societies to become industrial societies with the introduction of the steam engine, the Second was characterized by the advent of electricity production and petroleum, while the Third enabled the automation of production processes through information technology, leading to economic development and social progress. These are united by the centrality of energy as the basis of industrial development. Although this importance is still not to be disregarded, the Fourth Industrial Revolution considers an interconnection between real life and digital life from which various opportunities can be derived:

- Lower barriers between inventors and markets, from which a prime example is modern 3D printing, sharply reducing costs for small startups and speeding up time
- More importance for Artificial Intelligence (AI), which can be relied upon to solve complex systems, and according to multinational consulting firm McKinsey & Company, AI is expected to enable the automation of half of all jobs, creating new opportunities for the labour market [47]
- Integration between different technologies, facilitated by the introduction of innovative technologies to bring together different technical and scientific disciplines
- Improving lives through robotics that can wedge, play music and drive cars
- Everyday interconnection with the Internet, which, thanks to IoT devices, makes it possible to automate virtually every field around us, giving rise to smart cities [78]

Instead, all of this is characterized by the centrality of data and, no less important, the amount of data that organizations will have to manage, more commonly known as Big Data. Moving to this high level of abstraction, data is a strategic asset increasingly at the centre of global economic processes, and proper governance is needed from both public and private entities to best manage it. Not to be excluded is the definition of what resides around the data, such as metadata, which is all that information that indicates its content, structure, management and conditions of use [62]. In particular, metadata has the utility of containing:

- the nature, semantics, and quality of the data required by the agent processing them
- the volume, speed, and variety of data to be handled
- security requirements to manage authentication, integrity, confidentiality, and availability; reliability requirements against system failures
- metrics to be taken into account to dynamically reconfigure resources

By allowing the data to be described and contextualized, metadata are therefore critical in the application of Data-Centric Security approaches. There are three types of metadata according to their function [33]:

- **Descriptive Metadata** describes the resource by facilitating its search and uses, making explicit its subject matter

- **Structural Metadata** provides information to locate the resource within the storage, defining its structure and relationships with other resources
- **Administrative and Management Metadata** makes explicit the treatments applied to the resource or those to be applied, ensuring its integrity and authenticity over time, such as access rights

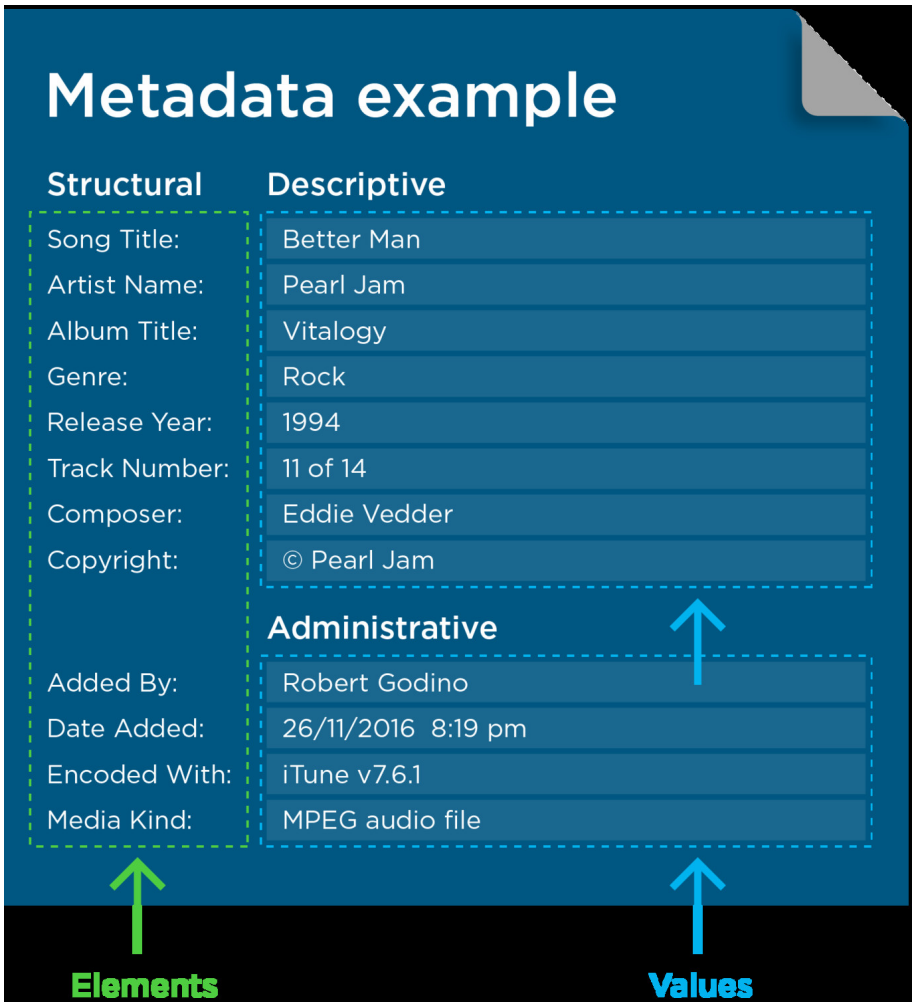


Figure 3.5. Metadata Example

Figure 3.5 clearly shows an example of metadata applied to a MPEG file [28].

The term Big Data itself emphasizes the enormous amount of data that organizations have to manage, in parallel with its scalability , heterogeneity and distribution. The platform used to manage this data must be able to meet these characteristics, having its architecture designed with the consideration that resources are organized into physical or virtual nodes, for data distribution and task execution, and communication networks, to make them communicate with each other.

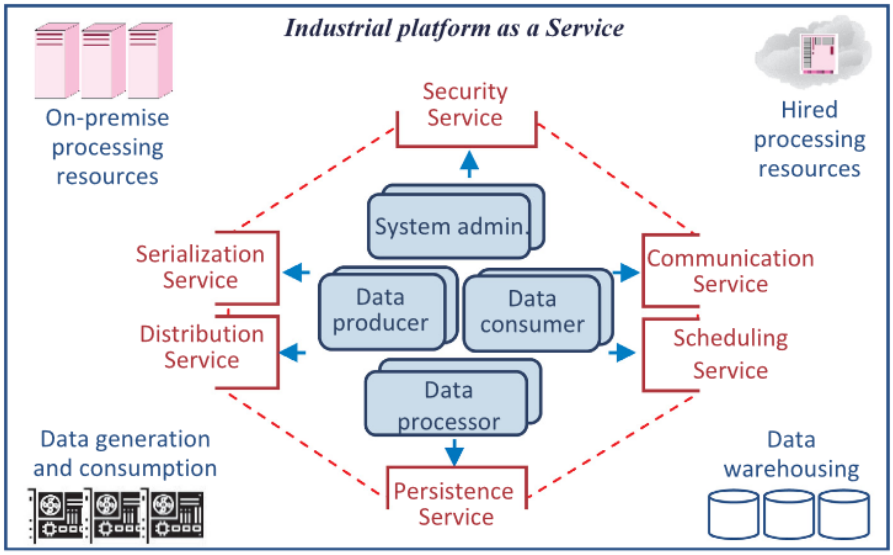


Figure 3.6. Platform as a Service of Big Data

The following six services described in Figure 3.6 are needed to best manage such a platform:

- **Serialization Service:** defines a language to make the information structure (schema) formal, the latter being available during read/write operations to/from the environment and for transfers within the platform
- **Distribution Service:** provides secure access to shared resources in the distributed system, offering metadata useful for the configuration and coordination of distributed services. It also assigns each resource a key/value duality and a synchronization system to inform all entities around the resource
- **Communication Service:** using a publisher/subscriber strategy records and sends the topics within the platform, that is, the flow of instances describing the same type of information. Its main role is to define data partitions distributed on free nodes
- **Scheduling Service:** schedules task execution in nodes dealing with execution queues and latency
- **Persistence Service:** manages the persistence of resources according to the requests of the agent that is processing them
- **Security Service:** fundamental to ensure security properties, such as authentication, integrity, confidentiality and availability. Being a data-centric approach, these must be defined on the data through specific metadata that indicate who can access those resources, in what way and with what restrictions.

It is important to note that the presented platform is decentralized to achieve both horizontal and vertical scalability[46].

Having talked about all these environments of uses, it is necessary to dwell on how users, whether military or users of IoT device services can access the data they need: one of the answers lies in the cloud solution, defined as "being able to access files, data, programs and third-party services ... via the Internet that are hosted by a third-party provider" [40]. Note how it is increasingly common for organizations to leverage existing cloud services of other companies instead of owning their computing systems, see in the public the case of Italy's National Strategic Pole that relies on TIM, Leonardo and Sogei's solution [34] and in the private sector the U.S. bank Goldman Sachs that relies on Amazon's cloud [29].

This paradigm shift leads to infrastructure as a service and frees organizations from the burden of directly managing data allocation and security. This model leads to net cost advantages, but in parallel, it poses a problem with data ownership and transparency of cloud services [43], especially in government and banking. Also from a legal point of view, there is uncertainty about the actual ownership of data when using a cloud solution, thus in what limits the cloud provider can use the third-party data [60]. Certainly having a model such as data-centric security that takes into account who, how and when used the resource can be a great asset in providing transparency to cloud users.

The main benefits to be gained from using emerging cloud computing technologies are [16]:

- **Economic Benefits** economic benefits are given by using the services on a pay-as-you-go basis, eliminating operational and ownership costs and circumventing the difficulties of managing an in-house solution that requires specialized human resources
- **On-demand scalability of computing resources** makes it possible to request the services needed for one's business based on the organization's available budget and performance and storage needs
- **Secured Platform** implementing cloud-level encryption, secure and up-to-date software, security audits and so on, preventing the organization from incurring high costs to manage cybersecurity
- **Massive Storage** allows the use of hundreds of Gigabytes accessible anywhere without space and time constraints
- **Facilitating collaborative practice** by offering a centralized system for accessing data by the organization and its users, also offering services to work together on the same resources facilitating the work of employees and students for companies and universities

Table 3.1 [16] shows how the services offered by cloud providers are highly differentiated from each other because the uses can be varied. From the Service Models there are three options as in Figure 3.7:

- **Infrastructure as a Service (Iaas):** the client has the availability of computer services such as storage, networking, and computing through which it can install and manage virtual machines including operating systems and applications that the user has control over, as opposed to the infrastructure that it does not handle

| Cloud Provider | Cloud Service | Service Model | Service Function |
|---|---|---|---|
| Amazon | EC2 | IaaS | Server |
| Amazon | S3 | IaaS | Storage |
| Google | GAE | PaaS | Development Environment |
| Microsoft Corp | Window Azure | IaaS | Storage |
| Microsoft Corp | Office 365 | SaaS | Office Suite |
| Salesforce | Salesforce Service Cloud | SaaS | Custom Relationship Management |
| CDC Software APTEAN | Pivotal CRM | SaaS | Business Customer Relationship |
| eBid Systems | ProcureWare | SaaS | Procurement System |
| Procore | Procore Construction Project Management Software | SaaS | Project Management System |
| e-Builder | e-Builder | SaaS | Construction Management Software |
| Oracle | Aconex | SaaS | Project Management System |
| Amazon | AWS EMR | SaaS | Hadoop Framework |

Table 3.1. Cloud Providers and Service Function

- **Platform as a Service (Paas):** without having to directly manage the production environment allows you to run or build applications and deploy them over the network. The customer no longer has control of the operating system, storage, and hardware but handles the applications and their configurations
- **Software as a Service (Saas):** enables applications that operate on the cloud platform, focusing on the end-user interface, where users can use and manage the cloud-built software without having to deal with the infrastructure, network, storage, and even specific applications [49]

To ensure security, it is essential to perform constant controls, which are divided into:

- **Preventive Controls** are used to mitigate the occurrence of a threat with, for example, improved cryptographic techniques and firewalls
- **Detective Controls** are used to identify the occurrence of a security or privacy threat in violation of policies with, for example, an Intrusion Detection System (IDS), logs, and analysis tools

Keeping in mind the importance of both approaches, Detective Controls are important in Data-Centric Security to ensure data auditing: indeed, they are useful for possible post-mortem forensic investigations of cloud services, ensuring accountability of the data throughout its lifecycle and keeping track of all events that characterize the data during this period [42].

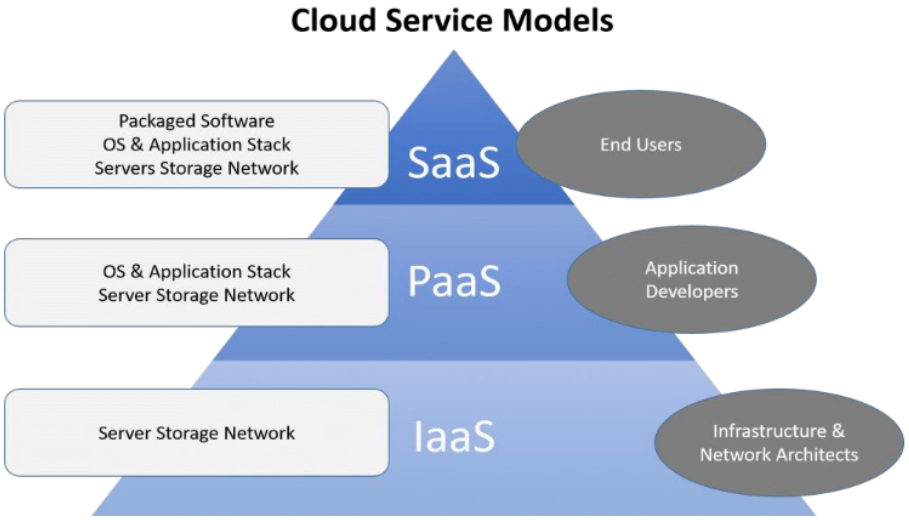


Figure 3.7. Cloud Service Models

In general, companies using Big Data technologies experience problems such as:

- managing resources with heterogeneous data organized in structured, unstructured, and semi-structured ways
- the need to adopt machine learning or graph computing algorithms in addition to simple SQL queries to obtain knowledge from large heterogeneous datasets
- continuously receive streams of data to be processed in (near) real-time time

Current frameworks allow these challenges to be solved, and we can divide the services offered into:

- **Batch Processing** long-term computing of large amounts of data, usually performed through Extract, Transform and Load (ETL) processes, data aggregation and machine learning models. Hadoop, which through its MapReduce implementation allows computation to be distributed across a cluster with several nodes, and Hyracks and Spark, used for their in-memory computational speed, can be used for this purpose
- **Streaming Processing** data are processed in a defined period, usually of a millisecond or microsecond duration. This is where tools such as Spark Streaming, which receives live data streams and splits them into smaller batch operations that are easier to handle but risk introducing overhead as seen in Figure 3.8, and Flink, which allows no task-scheduling overhead, ensuring a real-time scenario, are useful
- **Generic Storage** is provided by HDFS (Hadoop Distributed File System), which is capable of handling structured, unstructured, and semistructured data; Hyracks,

which through AsterixDB allows storage, indexing, querying, and analysis of large amounts of data and connects to HDFS; Spark, which allows interfacing to different file systems such as HDFS, Cassandra, Amazon S3; and Flink, which can connect to different heterogeneous resource sources and HDFS

- **Data Analytics** provided by YARN/Hadoop, which can support projects to manage and process the flow of data such as Giraph, Pig, Hive, Mahout, and HBase; Spark, which allows the implementation of various solutions such as ETL, Machine Learning (MLib), Stream Processing (Spark Streaming), and Graph Computing (GraphX); Flink, which offers several libraries with high-level APIs including Complex Event Processing (CEP), Machine Learning (FlinkML), to Graph Analysis (Gelly); Hyracks, which offers low-level APIs and contains several analysis interfaces such as SQL (Hivesterix), XQuery (Apache VXQuery), and Graph (Pregelix)

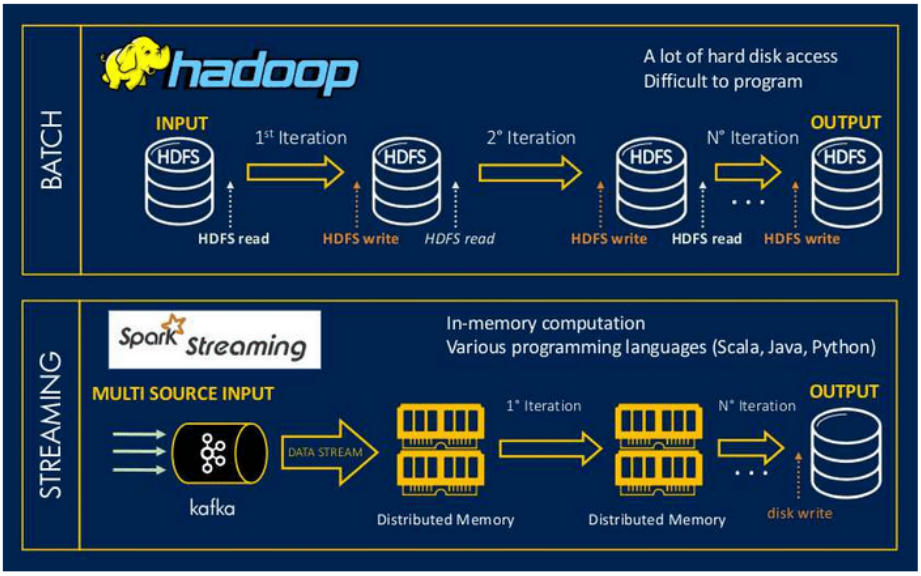


Figure 3.8. Comparison between batch processing and streaming processing

These points are summarized in Table 3.2.

Organizations also face problems such as:

- integrate different sources of Big Data in a way that shows them transparently to users
- keep data up-to-date, correct, complete, secure, and understandable
- monitor data and resources to get an overview of the performance of the entire system

These are grouped into three categories concerning how the platforms behave:

| | Hadoop | Spark | Flink | Hyracks |
|---|---|---|---|---|
| Processing Type | Batch | Mini-batch | Streaming, Batch | Batch |
| Generic Storage | HDFS | no primary storage | no primary storage | AsterixDB |
| Data Analytics | SQL, ML, Graph | ETL, ML, Graph | ML, CEP, Graph | SQL, XQuery, Graph |

Table 3.2. Summary of Big Data technologies discussed

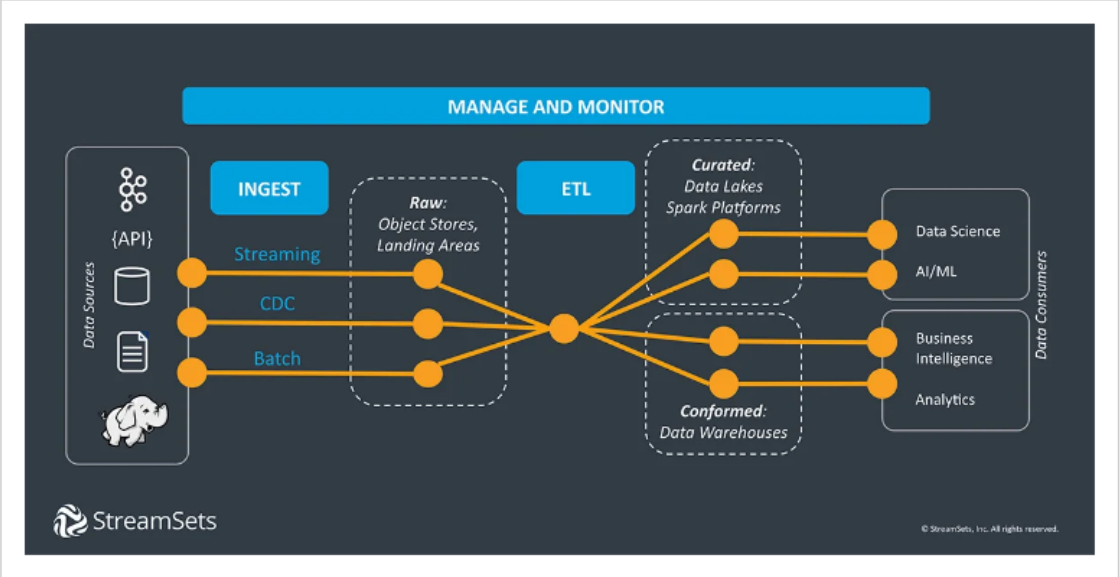


Figure 3.9. Data Integration Process

- **Data Integration** groups the difficulties in bringing together disparate data sources to provide the user with a unified view. It is formed by platforms such as Hadoop Data Platform (HDP), which through its partnership with Talend allows for Big Data integration by natively supporting Hadoop and including HDFS, HBase, Pig, Sqoop and Hive without having to write code; Oracle Data Integrator (ODI), where the user can create a stream from data sources of different technologies including relational databases, applications, XML, JSON, Hive tables, Hbase, and HDFS files; BigDataEurope (BDE) Platform, which offers a Semantic Data Lake called Ontario, i.e., a repository that can process and analyze datasets in their original formats. This allows the data to be mapped ensuring a total view of the data, after which the data can be extracted, queried and analyzed as if one were using a high-level language. A representation of this process is shown in Figure 3.9 [23]

- **Data Governance** represents the set of decision-making and accountability rights for information-related processes, which are then executed taking into account agreed-upon models that describe who can perform what, with what information, and when. Must be also considered the policies that affect the optimization, privacy and monetization of Big Data. The frameworks that offer these properties are Cloudera Navigator Data Management, a component for Hadoop that guarantees compliance, data governance and auditing for large organizations; Apache Atlas and Apache Ranger, used for HDP, adopt data classification and security policy strengthening: the first allows you to have a complete view of data movements through Apache Storm, Kafka, Falcon and Hive, while the second provides a centralized security platform for Hadoop. Atlas and Ranger manage dynamic runtime policies that prevent violations

- **Monitoring** allows you to evaluate the entire system proactively by monitoring data, resources and applications through services such as dashboards, alerts and reports. The frameworks that offer this service are Cloudera Manager, which allows you to control the health and performance of the various components and jobs running on clusters; BDE, which divides the monitoring of resources and system status, to control the condition of components or servers such as CPU usage, memory usage and I/O networks, and to monitor the status of an application; BDE, which can leverage the Docker Stats, cAdvisor, Prometheus tools, InfluxDB and Grafana; Apache Ambari, which for HDP manages the planning, installation and secure configuration of clusters, achieving easier maintenance and management. Figure 3.10 [44] depicts these key monitoring points

These properties are summarized in Table 3.3 [63].

| | HDP | BDE | Cloudera |
|---|---|---|---|
| Data Integration | Talend, ODI | Ontario, Semagrow | Talend |
| Data Governance | Atlas, Ranger | No support | Cloudera Navigator |
| Monitoring | Ambari | Prometheus, ELK stack | Cloudera Manager |

Table 3.3. Summary of Big Data platform studied

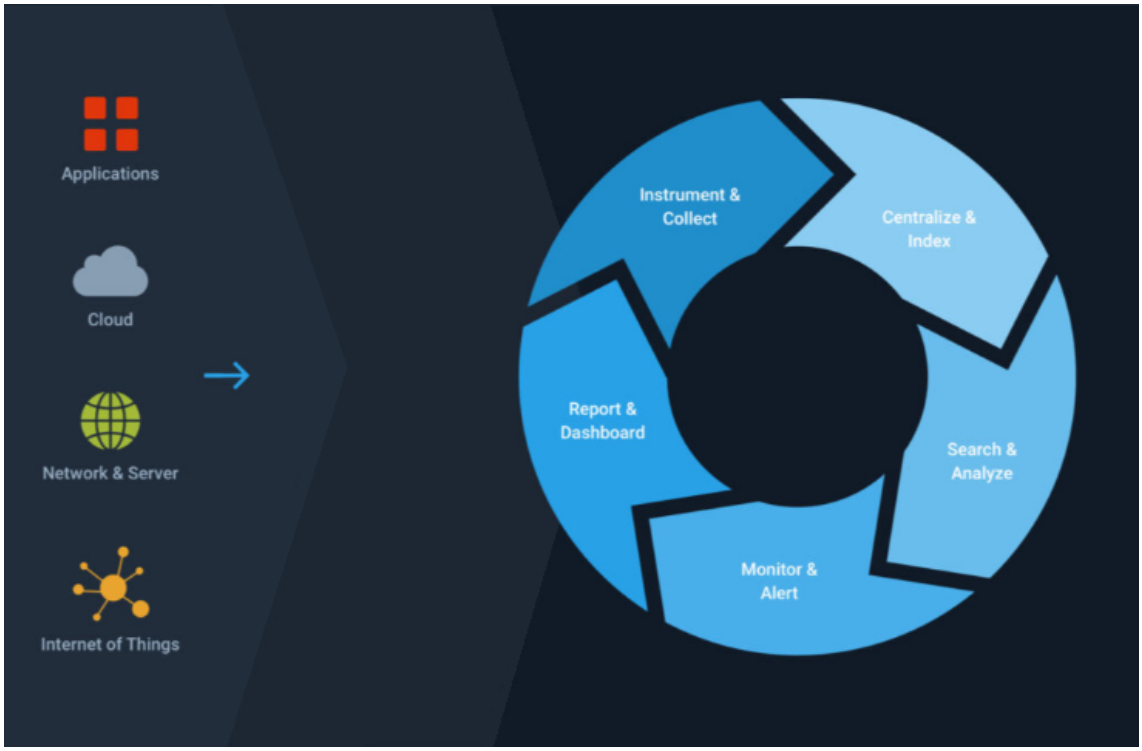


Figure 3.10. Monitoring Process

# Chapter 4

# Contribution

The work in this thesis focuses on the 900-hour internship at Pricewater-houseCoopers Business Services S.r.l held between March and September 2022. As a consulting firm, the project of interest was to define the client's security perimeter and the implementation and testing of security policies to manage user access to corporate databases. The approach used is Data-Centric Security, implemented through the use of specific tools and frameworks: in this case, Apache Ranger, a framework to enable, monitor and manage comprehensive data security across the Hadoop platform, was used.

## 4.1 Implementation Requirements

The tasks performed can be broken down into three phases. In the first phase, we analyzed the framework documentation and obtained a physical copy to understand its operation. This allowed us to determine the most appropriate logic to implement the client's required policies in Apache Ranger. The second phase involved implementing security policies for the client's data in their databases using automated in-house tools to minimize errors resulting from manual actions. Lastly, the third phase focused on testing the implementation of each policy using other automated tools to ensure accuracy. During the policy implementation phase, we used an automated tool to efficiently and accurately manage the task, minimizing the possibility of human error. Similarly, to verify that the implemented policies were correctly applied to each data item, we employed an additional automated tool to test their functionality [21].

The framework analysis phase focused heavily on the features provided by this tool to properly implement the five types of policies to be applied to resources:

- **Access Control List (ACL)** access to the database is restricted only to users who belong to a group with database visibility enabled. If a user belongs to the group that is authorized to access a database (e.g. DB X) and queries DB X, they will be able to access it and receive an OK message. However, if the same user belongs to the group that is authorized to access DB X but queries a different database (e.g. DB Y), they will not be able to access DB Y and will receive an error message (Figure 4.1)

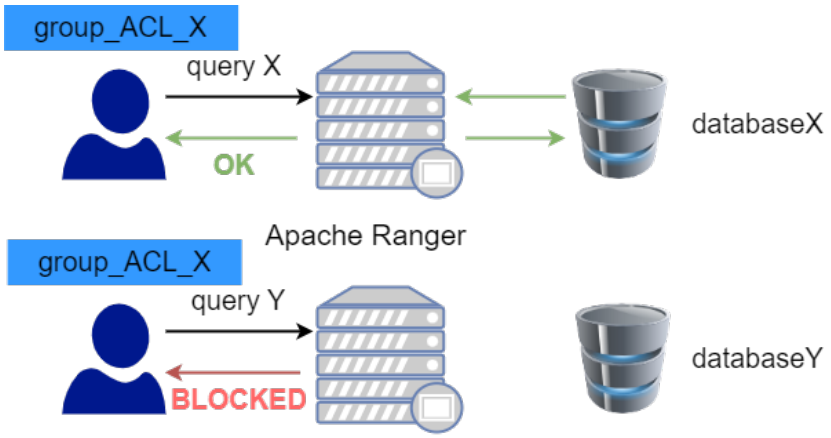


Figure 4.1. Policy ACL Scheme

- **Masking GDPR Data** In the database, values in a particular column or field are masked for all records that contain information classified as GDPR. If a user belongs to an ACL group for a database (e.g. DB X) and queries that database without belonging to the group with visibility rights over GDPR data for that database, the fields that contain GDPR data will be dynamically masked before being sent to the user. However, if a user belongs to both an ACL group for DB X and a group with visibility rights to GDPR data for that database, then they will be able to see all GDPR data in plain text when they query the database (Figure 4.2)

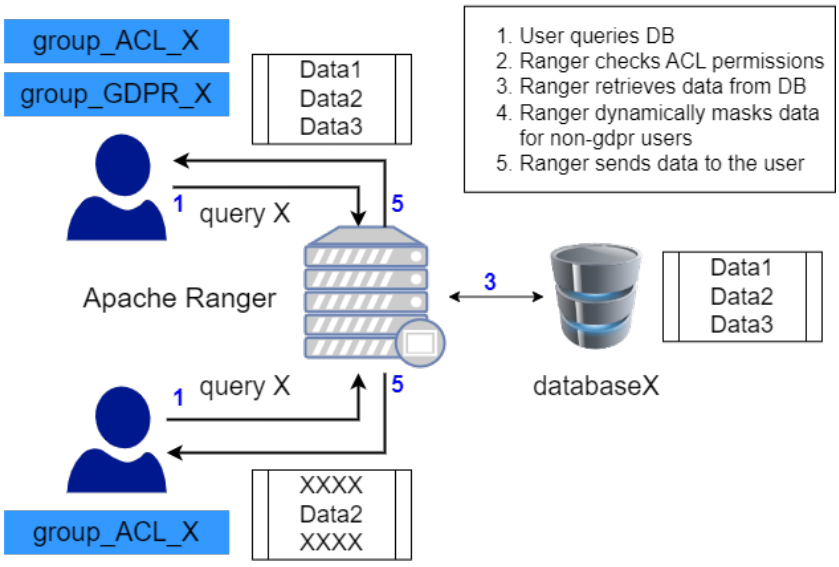


Figure 4.2. Policy GDPR Scheme

- **Row-Level Legal Entity Filtering** applies to data that belongs to specific legal

entities. To view this data, a user must belong to a group that is related to the database. If a user belongs to both the ACL group and the LE group of a database, they can view data that is not classified as a legal entity resource, as well as data that belongs to their legal entity. However, any remaining data that is protected by the policy and belongs to other legal entities will be filtered out. When retrieving data from the database, Ranger queries it with a new query that is adapted to filter the related legal entity (Figure 4.3)

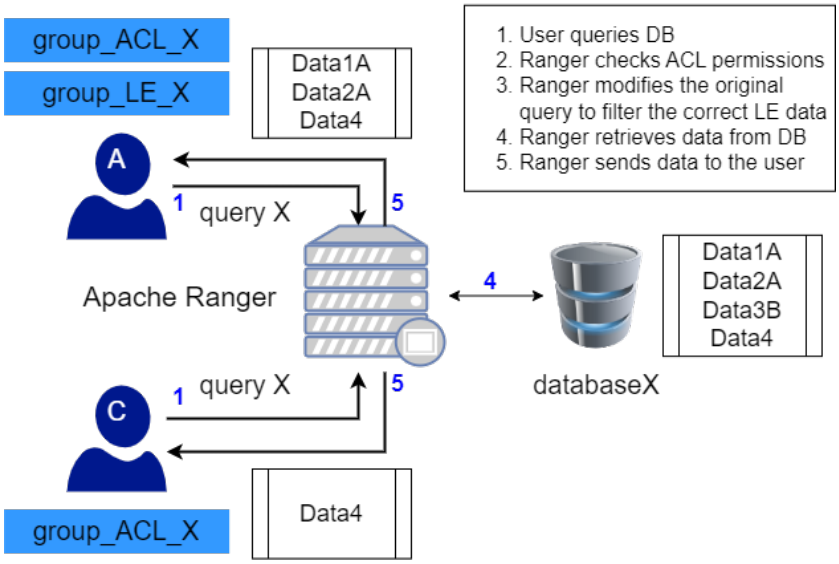


Figure 4.3. Policy Legal Entity Scheme

- **Reserved Data** access to database tables that contain confidential data is restricted to users who belong to the Confidential Data group associated with the database. If a user belongs to both the ACL group and the Restricted Data group, they can query the tables related to this database policy and see all the data in plain text. However, if the user does not belong to the Restricted Data group, they will not be able to access the tables and will receive a table access error (Figure 4.4)
- **Profiled Data** this policy applies to data that belongs to two different user entities, and visibility cones need to be applied to view this data. To access the profiled data, a user needs to belong to a database-related group of at least one of the two types. To simplify things, a single policy is created to handle both visibility cones, and it filters data based on the user's group membership. If a user belongs to both the ACL group and one Profiled group of a database (e.g., entity A), he can view data that is not classified as profiled in the two entities and data that belongs to the entity his group belongs to. However, any remaining data that's protected by the policy and belongs to the other entity will be filtered out. If the user belongs to the groups of both entities, he can see all related data. If they don't belong to either of these groups, he won't have visibility to this data. (Figure 4.5)

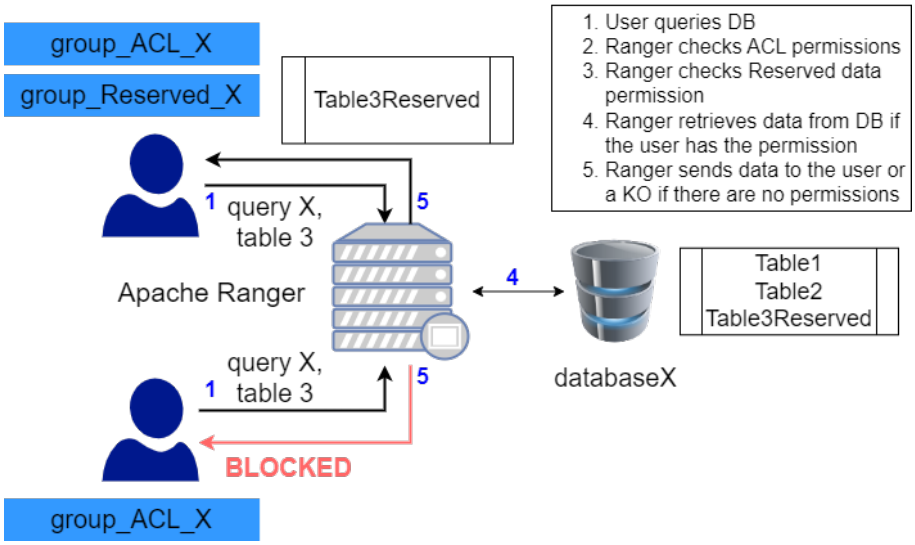


Figure 4.4. Policy Reserved Scheme

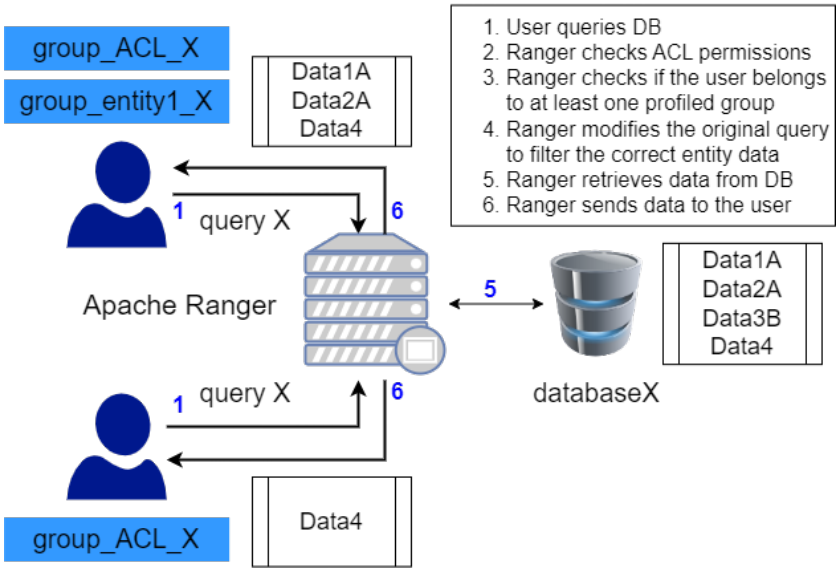


Figure 4.5. Policy Profiled Scheme

During the analysis phase, our team had to implement customer-required policies for database access. To achieve this, for the ACL policy we applied a default deny approach to all users and groups trying to access the databases, except for the specific group that has access to the data for each database. To ensure compliance with GDPR regulations, we implemented dynamic masking through generic masking on resources classified as GDPR, with the exception of the group authorized to view the data in plain text. Row-level

filtering for Legal Entities was also applied by filtering the content of a specific column in tables related to each Legal Entity and only applied to the specific Legal Entity group related to the database.

## 4.2 Apache Ranger Introduction

Apache Ranger is a centralized security management framework that supports unified authorization and auditing. It manages fine-grained access control over Hadoop and related components, such as HDFS, Hive, HBase, and Kafka. The Policy-Based Access Control (PBAC) access permission model provides unified permissions and policy synchronization, decision-making logic, and permission control for components. Additionally, it offers unified audit for user access and policy applications, a management interface for users and policy management, and log audit pages. In the context of the internship project, Apache Ranger was used to implement and manage various policy types required by the client.

Ranger's goals are to:

- manage all tasks to administer security centrally via User Interface (UI) or REST APIs
- manage through centralized administration permissions for actions and/or operations on components on which the framework is implemented
- support various authorization methods, including role-based access control and attribute-based access control
- have centralized auditing to better manage user access logs and administrative actions for all components

From these points it is clear that the great strength of Ranger is to manage all services centrally: this is done both through UI to make the User Experience (UX) simple, and through REST APIs to allow the use of automated tools for the entire management of security administration. The methods used are:

- **GET** The GET method is called to read a given resource
- **POST** The POST method is called when it is necessary to add a child resource to the resource collection
- **DELETE** The DELETE method is called to delete a given resource
- **PUT** The PUT method is called when it is necessary to modify a single resource, which is already part of the resource collection

## 4.3 Apache Ranger Architecture

From Figure 4.6 we observe the architecture of the Apache Ranger framework consisting mainly of three components:

- **Ranger Administration Portal** is a graphical interface that enables users to create and update policies, which are then saved to the policy database. The plugins embedded within each cluster component regularly poll these policies. Additionally, the portal includes an audit server that collects and sends audit data from the plugins to be stored in Hadoop Distributed File System (HDFS) or a relational database.
- **Ranger Plugins**, Java programs which embed within processes of each cluster component: when a request is sent from one of these components, the related plugin evaluates the request according to security policies. Another operation of the plugins is to forward to the audit server the information of the users who made the requests
- **User Group Sync**, namely that Apache Ranger allows synchronization of users by pulling them from Unix, LDAP, or Active Directory (AD). User and group information is stored in the Ranger portal

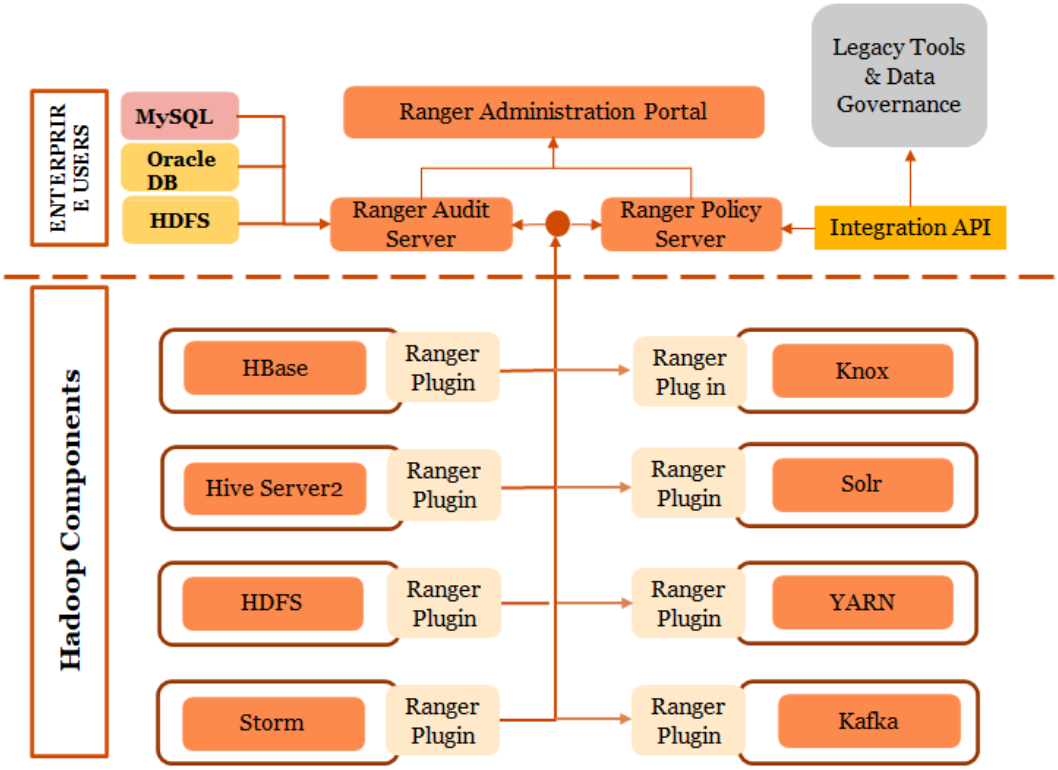


Figure 4.6. Apache Ranger Architecture

Apache Ranger provides centralized security policy management and auditing for all HDP components and manages access control across the following systems:

- **Apache Hadoop HDFS**, a distributed file system that manages large datasets running on commercial hardware. It is used to increase the size of a single Apache Hadoop cluster to hundreds (and even thousands) of nodes [22]
- **Apache Hive**, a distributed, fault-tolerant data warehouse system that enables analytics at a massive scale and facilitates reading, writing, and managing petabytes of data residing in distributed storage using SQL [8]
- **Apache HBase**, a distributed and scalable store for random and real-time read/write access to big data. Apache HBase is an open-source, distributed, versioned, non-relational database modelled after Google's Bigtable, a distributed Storage System for Structured Data [7]
- **Apache Storm**, a free and open-source distributed real-time computation system. Apache Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop did for batch processing [2]
- **Apache Knox**, an Application Gateway for interacting with the REST APIs and UIs of Apache Hadoop deployments. The Knox Gateway provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters [41]
- **Apache Solr**, an open-source enterprise search platform that is highly reliable, scalable and fault tolerant, providing distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration and more. Solr powers the search and navigation features of many of the world's largest internet sites [11]
- **Apache Kafka**, an open-source distributed event streaming platform used by thousands of companies for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications [9]
- **Apache NiFi**, a powerful and reliable system to process and distribute data that supports scalable directed graphs of data routing, transformation, and system mediation logic [10]
- **YARN**, a code package manager that allows to split the project into sub-components, use and share code with other developers from around the world in a quick, secure and reliable manner [79]

In the project, only Hive was used. However, Ranger plugins are installed on all HDP components. These plugins intercept authorization requests that are related to the component being queried. This is shown visually in Figure 4.7.

In Figure 4.8, we can see the components that make up the Ranger operation architecture. The Tag Sync Component is responsible for populating the tag store from an external system like Apache Atlas or a file. It does this by running a daemon process, which updates the tag store whenever tags are added, updated, or deleted from resources in Apache Atlas or file. This synchronization ensures that tags are defined consistently across systems and can be used to define cross-cutting policies. The tag represents a set of resources, even across databases, specified under a unique identifier (tag) and its use is

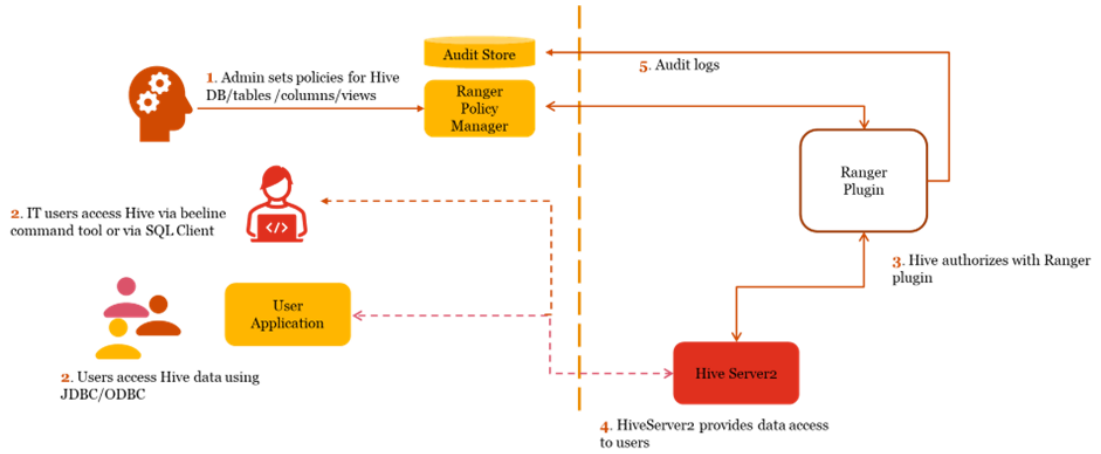


Figure 4.7. Apache Ranger Auditing Model

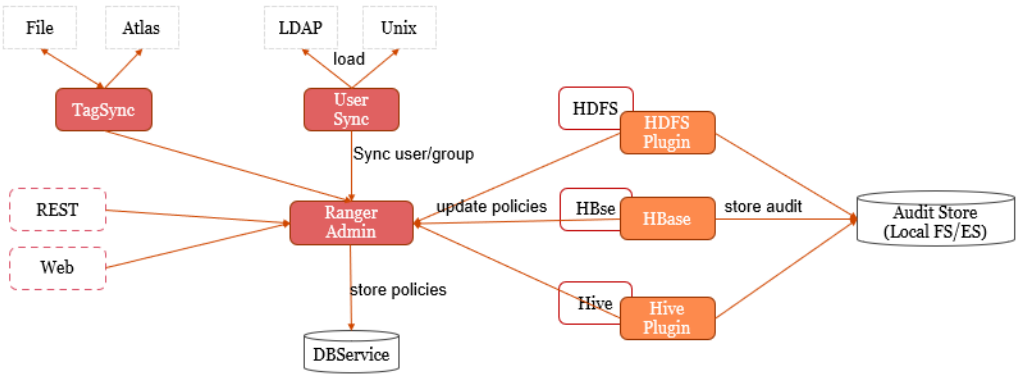


Figure 4.8. Apache Ranger Components

presented in chapter 4.4. The User Sync Component is another module that runs a daemon process. It retrieves user and group information from Active Directory or Unix and synchronizes them to the Ranger Admin. This enables the security administrator to apply policies to users and groups. The User Sync Component obtains user information periodically (default every 5 minutes) and pushes it to the Ranger Admin. The Ranger Admin is the core of the Ranger infrastructure. It is accessed through a web interface or a REST API and is used to create and update policies. Policies are stored in a policy database and are polled at regular intervals by plugins within each component. The Ranger Admin also includes an audit server that collects audit data from the plugins and stores it in HDFS or a relational database.

## 4.4 Policy Logic

Ranger manages access control through a user interface that ensures consistent policy administration across Hadoop's data access components. Security administrators can define security policies at the database, table, column, and file levels, and can administer permissions for specific LDAP-based groups or individual users so that they can be applied consistently across the entire HDP stack. Rules based on dynamic conditions such as time or geolocation can also be added to an existing policy rule . Once a user has been authenticated, his or her access rights must be determined. Authorization defines the user's access rights to resources. For example, a user may be authorized to create policies and view reports, but not to edit users and groups. Ranger can be used to set up and manage access to Hadoop services. It is also possible to create tag-based services and add access policies to those services. Using tag-based policies allows you to control access to resources across multiple Hadoop components without creating separate services and policies in each component. The mode of operation is then given in input by policy control and verification of data access request using Policy-Based Access Control (PBAC) logic, while in the output the data is masked based on the visibility cones related to the user accessing the data. Ranger allows policies to be implemented according to 2 methodologies:

- Resource-Based Policy
- Tag-Based Policy

Resource Based Policy is vertical on the resources to be protected and is effective if the goal is to secure well-defined DBs/tables/columns from unauthorized access. Similar to Resource-Based Policy, Ranger provides another security policy management methodology that allows the system administrator to leverage tag-based rather than resource-based logics: the tag represents a set of resources, even across databases, specified under a unique identifier (tag). Practical use of realization of these policies can be seen for the former in Figure 4.10[52], where for each policy must be specified the database, the table and the column, while for the latter in Figure 4.9[53], where all the resources are grouped in the tag "PII" and all the rules are applied directly to them.

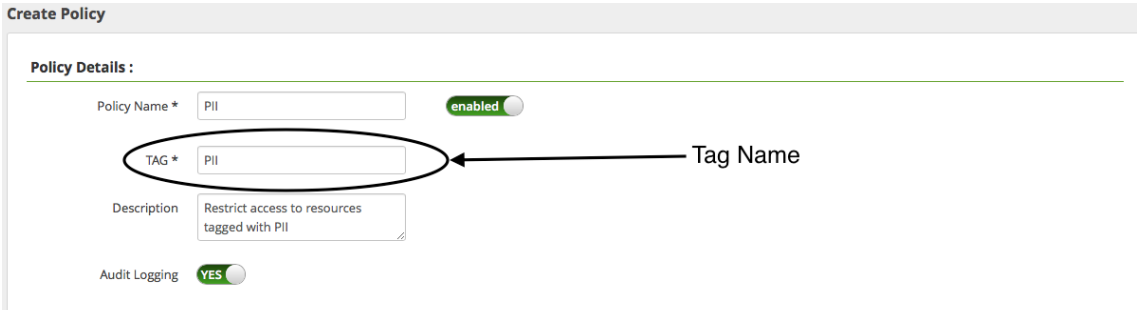


Figure 4.9. Example of Tag-Based Policy Implementation

In the analysis phase for policy implementation, the execution flow by which Ranger addresses these types should not be underestimated, and as shown in Figure 4.11, deny

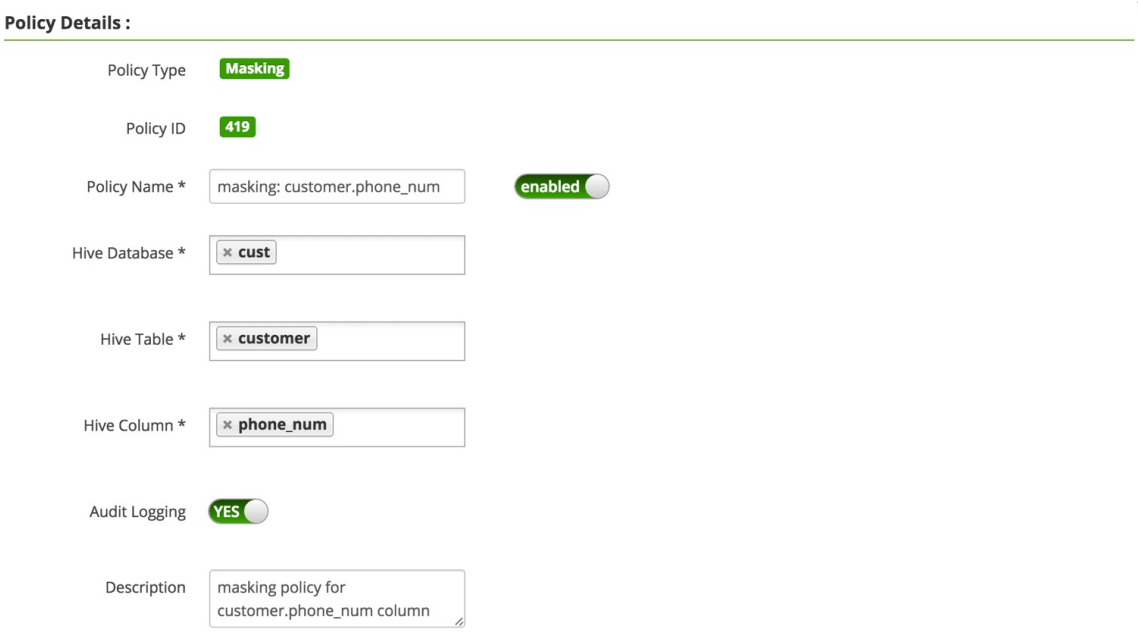


Figure 4.10. Example of Resource-Based Policy Implementation

policies are considered first and then allow policies, where specifically Tag-Based Policies are checked first and then Resource-Based Policies [53].

Among the key features of the Cloudera Ranger solution, we can implement Access Control policies for the entire Hadoop cluster. In this case, Ranger provides the ability to indicate, which:

- Db/tables are impacted by the policy
- Who is impacted by the policy
- Permissions of those impacted by the policy (select, create, etc.)

Masking policies can be implemented only if there is already an Access Control policy that regulates the impacted resources, and as can be seen in Figure 4.12, the user interface allows to choose the groups/users to which policies should be applied, the type of access and the Masking Option used, selected from nullify, hash, redact, partially mask, unmasked, date and custom. Ranger provides the ability to indicate:

- Db/tables are impacted by the policy
- Who is impacted by the policy
- Masking function to be applied (e.g., cast, substr, custom)

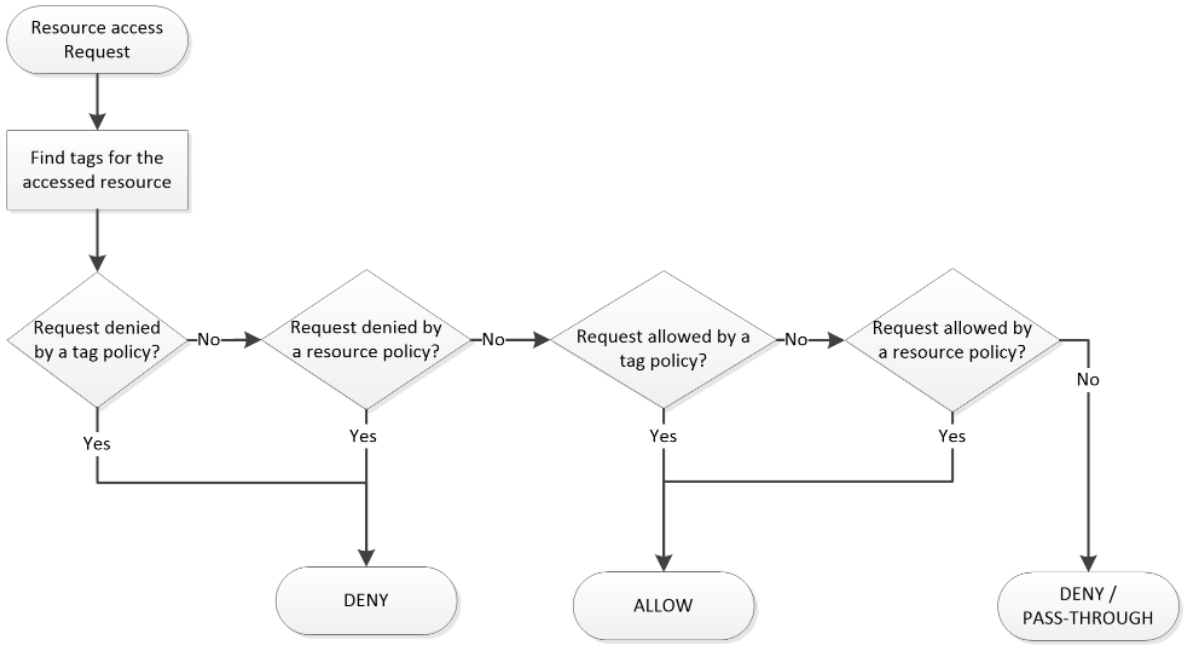


Figure 4.11. Policy Evaluation Flow

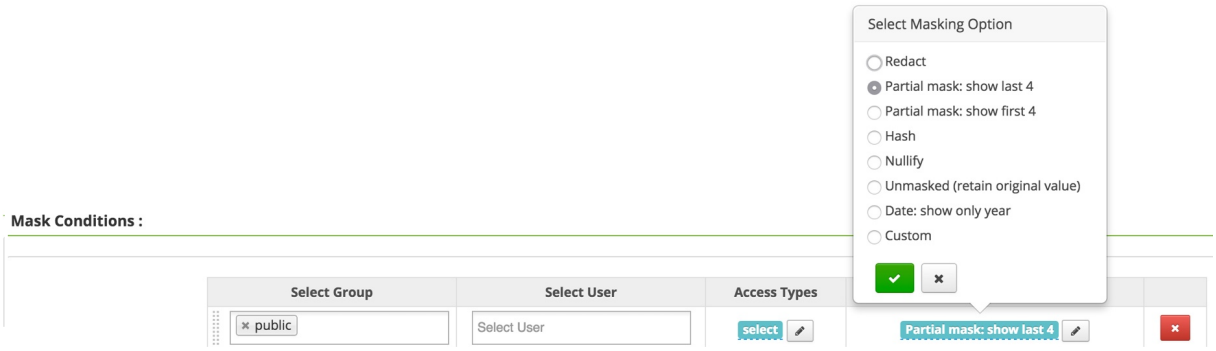


Figure 4.12. Example of Dynamic Data Masking Policy Implementation

The scenario could arise where, a malicious user uses SQL clauses to attempt to bypass the security features of the tool and still obtain, through Reverse Engineering techniques [19], the information behind the masking. Ranger in this regard knows the clauses that could generate data exfiltration and always returns zero results (e.g. select * from table1

where masked field like '%A%'). Row Filtering policies in Ranger can only be implemented if there is already a policy of Access Control that regulates impacted resources, and as can be seen in Figure 4.13, the user interface allows to choose the groups/users to which policies should be applied, the type of access and the filter that is used. Ranger provides the ability to indicate:

- Db/tables are impacted by the policy
- Who is impacted by the policy
- Row filtering function to be applied

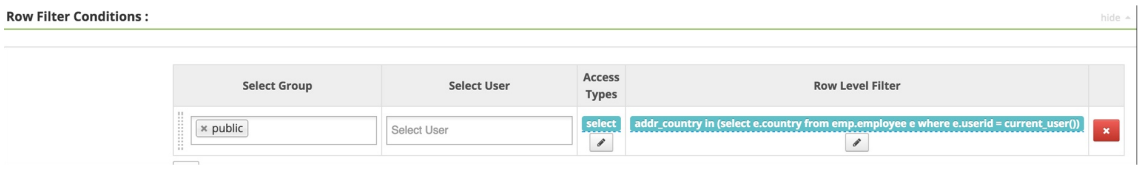


Figure 4.13. Example of Row-Level Filter Policy Implementation

Policies are managed through Ranger relying on the Policy-Based Access Control Model, but although there are open-source standards such as XACML (eXtensible Access Control Markup Language) [58], the framework prefers to focus on a secrecy approach without revealing its source code [68].

Often Ranger's native features are not sufficient to cover the requirements expressed during the analysis phase. Ranger, therefore, provides the ability to integrate customizations developed ad hoc. Suppose we manage some user attributes on a file or DB external to the Cloudera cluster. Ranger, unfortunately, does not have the native capabilities to connect to and retrieve data from external sources. To bypass this limitation, the following capabilities are in place:

- **Context Enricher**: Any java class that extends the RangerAbstractContextEnricher class. It has the main task of enriching the user's context, here it is possible to develop the logic that retrieves and processes data from external sources. The output is shared with the RangerAbstractConditionEvaluator class.
- **Condition Evaluator**: Any java class that extends the RangerAbstractContextEnricher class It has the main task of returning, based on input received from the RangerAbstractContextEnricher and following defined policies, a boolean value indicating whether or not the policy should be applied to the user.

Data-Centric Security is a valuable approach to enhancing data security and privacy by directly protecting the resource. However, users interact with this model through a single point, such as Apache Ranger, and so it is crucial to ensure the tool's security. Malicious users often bypass canonical security systems to target weaker and unconsidered steps in the security process. For example, the Man in the Browser Attack (MITB) [25] exploits

social engineering techniques [4] to deceive users and modify/carry information from official website pages to illegally obtain data, including passwords. Even banks are vulnerable to these attacks [71]. Therefore, we must remain vigilant and prioritize safety in all its forms. Even Apache Ranger has received security patches over time to become a better product, such as these concrete examples [74]:

- CVE-2019-12397 vulnerability to cross-site scripting (XXS) [55]
- CVE-2018-11778 vulnerability to stack buffer overflow [54]

In addition to continually considering the importance of correcting programs to be increasingly resistant to attack, the role the human user plays in the process must be taken into account: vulnerability to social engineering techniques such as phishing or spearing is given to the attention and preparation of the user who must always be alert to possible attacks of this kind.

Aside from continuously improving program resistance against attacks, it's important to acknowledge the human user's role in the process. Social engineering techniques, such as phishing [30] and spear phishing [35], can exploit vulnerabilities in users who are not attentive or prepared. Therefore, users must remain vigilant and alert to possible attacks of this kind to minimize risks to the system.

# Chapter 5

# Conclusions and Future Work

While the protection and privacy of data are still guaranteed by the use of cryptographic techniques and continuous updates in the strength of cryptographic algorithms, it is equally important to focus on the use of different layers of security and other approaches that can reduce the risk caused by malicious sources. Using a battlefield as a comparison, the use of multiple lines of defence allows the security of the counterpart to be more reliable so that even if the first line is penetrated others maintain the defence. Data-Centric Security is precisely one such approach that can be adopted to enable organizations to provide security and privacy to resources under management simply and cost-effectively. From a more practical point of view, it is essential to focus on metadata, that is, all the information that accompanies the data and defines its description, structure, and details for the administration and management of the resource: it is precisely the latter that makes it possible to implement Data-Centric Security with the use of access and display policies. The name itself hints at how this paradigm shift brings the data to the centre of attention and precisely because of this metadata can guarantee its security and privacy within the organisation's environment.

At the market level nowadays, some various tools and frameworks allow implementing the data-centric approach effectively by offering Platform-as-a-Service (Paas): here the focus was on the description of the architecture and functionality of Apache Ranger, as it was used during the project carried out for the curricular internship in PricewaterhouseCoopers Business Services S.r.l. to analyze, implement and test five different types of security policies using its graphical user interface (GUI) or automated testing tools that leverage the REST API. In the various frameworks commonly used nowadays, there are not all the same architectures, but it is necessary to have different solutions to the problem by dividing them according to the services performed and the technologies offered: the former move between processing (batch and streaming), generic storage and data analytics, while the latter focus on data integration, data governance and monitoring. Indeed, these properties allow for tools and frameworks with a common line focused on cloud management and a data-centric approach, but each going to look at a different slice of market needs. This is also evident by going to look at how the information environment is increasingly permeated by cloud solutions that provide users with a constant data connection and sharing, the sources of which are the varied amounts of IoT devices that surround our lives, the data with which the public administration represents citizens and those that are shared

with private companies using their services (e.g. Netflix) or products (e.g. Apple), and so on.

The need to constantly increase the security standards used is not only linked to a proportional growth in risks given by the evolution of threats but also by a sharp increment in the amount of data in circulation that directly implies a wider risk surface open to attacks. Data should not be treated as a simple sequence of bits because it is essential to take into account its intrinsic value and the importance and meaning it holds with all the implications it can have: if the semantics of data is analyzed, it is clear how they are the transposition into the digital of users. Therefore, legal protections for the use of this information must be ensured as has been transposed by European nations following the 2016 GDPR, this is not only because of a mere legal duty but because guaranteeing the privacy and security of data is a right contained in Article 12 of the United Nations Universal Declaration of Human Rights: letting these resources be unconditionally managed can result in a democratic risk if used unproperly for a political purpose. Privacy is not just an empty word that we exploit to defend ourselves from intrusive cookies or to remain anonymous on the web, but it has a fundamental role in modern society that provides the individual citizen with a guarantee against interference in his or her private life: even more significant to note is the parabolic evolution of the economic value of data in the present day, which as The Economist points out has surpassed that of wheat and oil [69], stimulating the malicious use of data and forcing organizations to use ever-higher security systems.

Currently, Data-Centric Security represents a breakthrough to be taken into account regarding the cyber resilience of systems, whether public or private. It should also be considered how this approach is independent of the network environment around it, which may be on-premise, cloud or hybrid, going by the fact that in general the purpose of cyber attacks is not to attack the network itself but to steal or manipulate the data passing through it. In an ever-changing world this is a huge plus point as it ensures that usage is not focused on one specific technology but instead is highly flexible to support its security at the resource level without worrying about what will be the main technology used in the future.

Thus, if we look to the future, we should not focus primarily on improving this paradigm, but on ensuring that it will be applied in as many solutions as possible, to make security management simpler and cheaper. In the field of cybersecurity, there is no 100 percent chance of guaranteeing the security of an environment, so even this approach can have some weaknesses: unintentional loss of information due to human error or improper implementation of the design, unauthorized eavesdropping on networks or data breaches, poisoning of resource fairness, malicious activities that alter the proper functioning of the model such as Distributed Denial of Service (DDoS) Attacks and code injections in the internal systems. Special attention must therefore be paid to all phases of system design and implementation on which the data-centric approach is applied to try to avoid human error. No less important are the checks to be made when the system is in operation to monitor and control that the defence resists DDoS attacks, data breaches, and code injections.

The struggle between attackers and defenders in the security of data is often portrayed as a simple Tolkienian battle between good and evil, but the reality is far more complex. It is essential to stay vigilant and up-to-date to maintain an adequate level of security, as protecting data means defending the rights and privacy of users. To quote Machiavelli, "si

vis pacem, para bellum" - if you want peace, prepare for war - this endless conflict requires constant preparation and readiness.

# Bibliography


[1] https://www.congress.gov/bill/117th-congress/house-bill/7535/actions. Accessed: 2023-1-10.

[2] URL: https://storm.apache.org/.

[3] Sajeda Akter, Sriram Chellappan, Tusher Chakraborty, Taslim Arefin Khan, Ashikur Rahman, and A. B. M. Alim Al Islam. «Man-in-the-Middle Attack on Contactless Payment over NFC Communications: Design, Implementation, Experiments and Detection». In: *IEEE Transactions on Dependable and Secure Computing* 18.6 (2021), pp. 3012–3023. DOI: 10.1109/TDSC.2020.3030213.

[4] Aisha Suliaman Alazri. «The awareness of social engineering in information revolution: Techniques and challenges». In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015, pp. 198–201. DOI: 10.1109/ICITST.2015.7412088.

[5] Tejasvi Alladi, Vinay Chamola, Biplab Sikdar, and Kim-Kwang Raymond Choo. «Consumer IoT: Security vulnerability case studies and solutions». In: *IEEE Consumer Electronics Magazine* 9.2 (2020), pp. 17–25.

[6] Jason Andress. «Chapter 5 - Cryptography». In: *The Basics of Information Security*. Ed. by Jason Andress. Boston: Syngress, 2011, pp. 63–80. ISBN: 978-1-59749-653-7. DOI: https://doi.org/10.1016/B978-1-59749-653-7.00005-0. URL: https://www.sciencedirect.com/science/article/pii/B9781597496537000050.

[7] *Apache HBase – Apache HBase™ home*. URL: https://hbase.apache.org/.

[8] *Apache Hive*. URL: https://hive.apache.org/.

[9] *Apache kafka*. URL: https://kafka.apache.org/.

[10] *Apache Nifi*. URL: https://nifi.apache.org/.

[11] *Apache Solr*. URL: https://solr.apache.org/.

[12] Alessandro Armando, Matteo Grasso, Sander Oudkerk, Silvio Ranise, and Konrad Wrona. «Content-based information protection and release in NATO operations». In: *Proceedings of the 18th ACM symposium on Access control models and technologies*. 2013, pp. 261–264.

[13] Alessandro Armando, Sander Oudkerk, Silvio Ranise, and Konrad Wrona. «Formal modelling of content-based protection and release for access control in NATO operations». In: *International Symposium on Foundations and Practice of Security*. Springer. 2014, pp. 227–244.

[14] Amandeep Singh Arora, Linesh Raja, and Barkha Bahl. «Data Centric Security Approach: A Way to Achieve Security & Privacy in Cloud Computing». In: *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)*. Malaviya National Institute of Technology, Jaipur (India): SSRN, 2018. URL: http://dx.doi.org/10.2139/ssrn.3168615.

[15] Elaine Barker and Allen Roginsky. *Transitioning the Use of Cryptographic Algorithms and Key Lengths*. en. 2019. DOI: https://doi.org/10.6028/NIST.SP.800-131Ar2.

[16] Sururah A Bello, Lukumon O Oyedele, Olugbenga O Akinade, Muhammad Bilal, Juan Manuel Davila Delgado, Lukman A Akanbi, Anuoluwapo O Ajayi, and Hakeem A Owolabi. «Cloud computing in construction industry: Use cases, benefits and challenges». In: *Automation in Construction* 122 (2021), p. 103441.

[17] Paul Breitbarth. «The impact of GDPR one year on». In: *Network Security* 2019.7 (2019), pp. 11–13.

[18] Christoph Buck, Christian Olenberger, André Schweizer, Fabiane Völter, and Torsten Eymann. «Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust». In: *Computers & Security* 110 (2021), p. 102436. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102436. URL: https://www.sciencedirect.com/science/article/pii/S0167404821002601.

[19] Gerardo Canfora, Massimiliano Di Penta, and Luigi Cerulo. «Achievements and Challenges in Software Reverse Engineering». In: *Commun. ACM* 54.4 (2011), 142–151. ISSN: 0001-0782. DOI: 10.1145/1924421.1924451. URL: https://doi.org/10.1145/1924421.1924451.

[20] Lingfeng Chen and Doan Hoang. «Active data-centric framework for data protection in cloud environment». In: *ACIS 2012 Proceedings*. 71. 2012, pp. 1–12. URL: https://aisel.aisnet.org/acis2012/71.

[21] Maria Odea Ching. *Apache Ranger - Introduction*. 2021. URL: https://ranger.apache.org/.

[22] *Cos'è l'hdfs? Apache Hadoop Distributed File System*. URL: https://www.ibm.com/it-it/topics/hdfs.

[23] *Data Integration: Tools, techniques, key Concepts| streamsets*. URL: https://streamsets.com/learn/data-integration/.

[24] Ali Dehghantanha, Nur Izura Udzir, and Ramlan Mahmod. «Towards data centric mobile security». In: *2011 7th International Conference on Information Assurance and Security (IAS)*. 2011, pp. 62–67. DOI: 10.1109/ISIAS.2011.6122796.

[25] Timothy Dougan and Kevin Curran. «Man in the browser attacks». In: *International Journal of Ambient Computing and Intelligence (IJACI)* 4.1 (2012), pp. 29–39.

[26] Abigayle Erickson. «Comparative Analysis of the EU's GDPR and Brazil's LGPD: Enforcement Challenges with the LGPD». In: *Brook. J. Int'l L.* 44 (2018), p. 859.

[27] Simon Erni, Martin Kotuliak, Patrick Leu, Marc Roeschlin, and Srdjan Capkun. «AdaptOver: Adaptive Overshadowing Attacks in Cellular Networks». In: *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*. MobiCom '22. Sydney, NSW, Australia: Association for Computing Machinery, 2022, 743–755. ISBN: 9781450391818. DOI: 10.1145/3495243.3560525. URL: https://doi.org/10.1145/3495243.3560525.

[28] Robert Godino. *An introduction to metadata and taxonomy*. 2016. URL: https://www.linkedin.com/pulse/introduction-metadata-taxonomy-robert-godino.

[29] *Goldman Sachs on AWS*. 2007. URL: https://aws.amazon.com/it/solutions/case-studies/innovators/goldman-sachs/.

[30] Surbhi Gupta, Abhishek Singhal, and Akanksha Kapoor. «A literature survey on social engineering attacks: Phishing attack». In: *2016 International Conference on Computing, Communication and Automation (ICCCA)*. 2016, pp. 537–540. DOI: 10.1109/CCAA.2016.7813778.

[31] Gary Helm and MD Minhaz Chowdhury. «Security Issues of Mobile Devices: A Survey». In: *2021 IEEE International Conference on Electro Information Technology (EIT)*. 2021, pp. 014–020. DOI: 10.1109/EIT51626.2021.9491840.

[32] Philippa Howden-Chapman, José Siri, Elinor Chisholm, Ralph Chapman, Christopher NH Doll, and Anthony Capon. «SDG 3: Ensure healthy lives and promote wellbeing for all at all ages». In: *A guide to SDG interactions: from science to implementation. Paris, France: International Council for Science* (2017), pp. 81–126.

[33] *I metadati: Cosa Sono e perché sono fondamentali*. 2022. URL: https://www.doxee.com/it/blog/regtech/i-metadati-cosa-sono-e-perche-sono-fondamentali-per-la-conservazione-dei-documenti/.

[34] *Il cloud sicuro per l'Italia Digitale*. URL: https://www.polostrategiconazionale.it/.

[35] Ankit Kumar Jain and Brij B Gupta. «Feature based approach for detection of smishing messages in the mobile environment». In: *Journal of Information Technology Research (JITR)* 12.2 (2019), pp. 17–35.

[36] Lin Jiao, Yonglin Hao, and Dengguo Feng. «Stream cipher designs: a review». In: *Science China Information Sciences* 63.3 (2020), pp. 1–25.

[37] Kevin Jones. *What is DCAP Gartner?* 2019. URL: https://www.hackercombat.com/what-is-dcap-gartner/.

[38] AH Kabir. «Data Centric Security». In: *National Security Institute J* (2015), pp. 21–33.

[39] Hyunsung Kim et al. «Research issues on data centric security and privacy model for intelligent internet of things based healthcare». In: *ICSES Trans. Comput. Netw. Commun* 5 (2019), pp. 1–3.

[40] Eric Knorr and Galen Gruman. «What cloud computing really means». In: *InfoWorld* 7.20-20 (2008), pp. 1–17.

[41] *Knox Gateway*. URL: https://knox.apache.org/.

[42] Ryan K. L. Ko, Markus Kirchberg, and Bu Sung Lee. «From system-centric to data-centric logging - Accountability, trust & security in cloud computing». In: *2011 Defense Science Research Conference and Expo (DSR)*. 2011, pp. 1–4. DOI: 10.1109/DSR.2011.6026885.

[43] Ryan KL Ko, Bu Sung Lee, and Siani Pearson. «Towards achieving accountability, auditability and trust in cloud computing». In: *Advances in Computing and Communications: First International Conference, ACC 2011, Kochi, India, July 22-24, 2011, Proceedings, Part IV 1*. Springer. 2011, pp. 432–444.

[44] Rajesh Kumar. *List of log monitoring tools | scmgalaxy - devopsschool.com*. 2017. URL: https://www.devopsschool.com/blog/top-5-log-monitoring-tools/.

[45] Ahmed Y Lotfy, Alaa M Zaki, Tarek Abd-El-Hafeez, and Tarek M Mahmoud. «Privacy Issues of Public Wi-Fi Networks». In: *The International Conference on Artificial Intelligence and Computer Vision*. Springer. 2021, pp. 656–665.

[46] Patricia López Martínez, Ricardo Dintén, José María Drake, and Marta Zorrilla. «A big data-centric architecture metamodel for Industry 4.0». In: *Future Generation Computer Systems* 125 (2021), pp. 263–284. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2021.06.020. URL: https://www.sciencedirect.com/science/article/pii/S0167739X21002156.

[47] James Manyika, Michael Chui, Mehdi Miremadi, Jacques Bughin, Katy George, Paul Willmott, Martin Dewhurst, et al. «Harnessing automation for a future that works». In: *McKinsey Global Institute* (2017), pp. 2–4.

[48] Teodor Mitrea and Monica Borda. «Mobile Security Threats: A Survey on Protection and Mitigation Strategies». In: *International conference KNOWLEDGE-BASED ORGANIZATION* 26.3 (2020), pp. 131–135. DOI: doi:10.2478/kbo-2020-0127. URL: https://doi.org/10.2478/kbo-2020-0127.

[49] Chnar Mustafa Mohammed, Subhi RM Zeebaree, et al. «Sufficient comparison among cloud computing services: IaaS, PaaS, and SaaS: A review». In: *International Journal of Science and Business* 5.2 (2021), pp. 17–30.

[50] Trevor J. Morgan. *Starting with data-centric security: A cost-benefit analysis*. URL: https://insights.comforte.com/starting-with-data-centric-security-a-cost-benefit-analysis.

[51] Muhammad Mujib and Riri Fitri Sari. «Design of implementation of a zero trust approach to network micro-segmentation». In: *International JOURNAL of Advanced Science and Technology* 29.7 Special Issue (2020), pp. 3501–3510.

[52] Madhan Neethiraj. *Row-level filtering and column-masking using Apache Ranger policies in Apache Hive*. 2016. URL: https://cwiki.apache.org/confluence/display/RANGER/Row-level+filtering+and+column-masking+using+Apache+Ranger+policies+in+Apache+Hive.

[53] Madhan Neethiraj. *Tag based policies*. 2015. URL: https://cwiki.apache.org/confluence/display/RANGER/Tag+Based+Policies.

[54] *Nvd - cve-2018-11778*. en. https://nvd.nist.gov/vuln/detail/CVE-2018-11778. Accessed: 2023-1-29.

[55] *Nvd - cve-2019-12397*. en. https://nvd.nist.gov/vuln/detail/CVE-2019-12397. Accessed: 2023-1-29.

[56] Juan D. Parra Rodriguez, Daniel Schreckling, and Joachim Posegga. «Addressing Data-Centric Security Requirements for IoT-Based Systems». In: *2016 International Workshop on Secure Internet of Things (SIoT)*. 2016, pp. 1–10. DOI: 10.1109/SIoT.2016.007.

[57] Deva Prasad M and Suchithra Menon C. «The Personal Data Protection Bill, 2018: India's regulatory journey towards a comprehensive data protection law». In: *International Journal of Law and Information Technology* 28.1 (Feb. 2020), pp. 1–19. ISSN: 0967-0769. DOI: 10.1093/ijlit/eaaa003. eprint: https://academic.oup.com/ijlit/article-pdf/28/1/1/33387032/eaaa003.pdf. URL: https://doi.org/10.1093/ijlit/eaaa003.

[58] Carroline Dewi Puspa Kencana Ramli, Hanne Riis Nielson, and Flemming Nielson. «The logic of XACML». In: *Science of Computer Programming* 83 (2014). Formal Aspects of Component Software (FACS 2011 selected & extended papers), pp. 80–105. ISSN: 0167-6423. DOI: https://doi.org/10.1016/j.scico.2013.05.003. URL: https://www.sciencedirect.com/science/article/pii/S0167642313001238.

[59] Stefan Ransom and Christian Werner. «Towards Data-Centric Security in Ubiquitous Computing Environments». In: *2009 20th International Workshop on Database and Expert Systems Application*. 2009, pp. 26–30. DOI: 10.1109/DEXA.2009.22.

[60] Chris Reed. «Information Ownership in the Cloud». In: *Cloud Computing Law (2nd Edn, OUP 2021)* (2021).

[61] Redazione RHC. *Il pre-quantum computing si avvicina. Gli USA Danno priorità alla migrazione a standard PQC*. Accessed: 2023-1-10. 2023. URL: https://www.redhotcyber.com/post/il-pre-quantum-computing-si-avvicina-gli-usa-danno-priorita-alla-migrazione-a-standard-pqc/.

[62] Lucia Sardo and Paul Gabriele Weston. «Metadati». In: *Metadati* (2017), pp. 1–123.

[63] Ticiana L Coelho da Silva, Regis Pires Magalhães, Igo Ramalho Brilhante, José AF de Macêdo, David Araújo, Paulo AL Rego, and Aloisio Vieira Lira Neto. «Big Data Analytics Technologies and Platforms: A Brief Review.» In: *LADaS VLDB* (2018), pp. 25–32.

[64] Gustavus J. Simmons. «Symmetric and Asymmetric Encryption». In: *ACM Comput. Surv.* 11.4 (1979), 305–330. ISSN: 0360-0300. DOI: 10.1145/356789.356793. URL: https://doi.org/10.1145/356789.356793.

[65] Peter Sommerlad. «Reverse Proxy Patterns.» In: *EuroPLoP*. 2003, pp. 431–458.

[66] Aug Statista Research Department 22. *Smartphone subscriptions worldwide 2027*. 2022. URL: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/.

[67] K. Suveetha and Manju Thomas. «Ensuring Confidentiality of Cloud Data using Homomorphic Encryption». In: *Indian Journal of Science and Technology* 9 (Mar. 2016). DOI: 10.17485/ijst/2016/v9i8/87964.

[68] Anne M. Tall and Cliff C. Zou. «A Framework for Attribute-Based Access Control in Processing Big Data with Multiple Sensitivities». In: *Applied Sciences* 13.2 (2023), p. 1183. ISSN: 2076-3417. DOI: 10.3390/app13021183. URL: http://dx.doi.org/10.3390/app13021183.

[69] *The world's most valuable resource is no longer oil, but Data*. 2017. URL: https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

[70] Xavier Tracol. «"Schrems II": The return of the privacy shield». In: *Computer Law & Security Review* 39 (2020), p. 105484.

[71] Nattakant Utakrit. «Review of browser extensions, a man-in-the-browser phishing techniques targeting bank customers». In: (2009).

[72] Panos Vassiliadis, Alkis Simitsis, and Spiros Skiadopoulos. «Conceptual Modeling for ETL Processes». In: *Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP*. DOLAP '02. McLean, Virginia, USA: Association for Computing Machinery, 2002, 14–21. ISBN: 1581135904. DOI: 10.1145/583890.583893. URL: https://doi.org/10.1145/583890.583893.

[73] JC Villanueva. «An introduction to stream ciphers vs. Block ciphers». In: *JSCAPE LLC* 12 (2015).

[74] *Vulnerabilities found in Ranger*. URL: https://cwiki.apache.org/confluence/display/RANGER/Vulnerabilities+found+in+Ranger.

[75] Ben Wolford. *What is GDPR, the EU's new Data Protection Law?* 2022. URL: https://gdpr.eu/what-is-gdpr/.

[76] Konrad Wrona, Angel de Castro, and Bogdan Vasilache. «Data-centric security in military applications of commercial IoT technology». In: *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. 2016, pp. 239–244. DOI: 10.1109/WF-IoT.2016.7845511.

[77] Konrad Wrona and Sander Oudkerk. «Content-Based Protection and Release Architecture for Future NATO Networks». In: *MILCOM 2013 - 2013 IEEE Military Communications Conference*. 2013, pp. 206–213. DOI: 10.1109/MILCOM.2013.44.

[78] Min Xu, Jeanne M David, Suk Hi Kim, et al. «The fourth industrial revolution: Opportunities and challenges». In: *International journal of financial research* 9.2 (2018), pp. 90–95.

[79] *YARN*. URL: https://yarnpkg.com/.

[80] Muneer Bani Yassein, Shadi Aljawarneh, Ethar Qawasmeh, Wail Mardini, and Yaser Khamayseh. «Comprehensive study of symmetric key and asymmetric key encryption algorithms». In: *2017 International Conference on Engineering and Technology (ICET)*. 2017, pp. 1–7. DOI: 10.1109/ICEngTechnol.2017.8308215.

[81] Razieh Nokhbeh Zaeem and K. Suzanne Barber. «The Effect of the GDPR on Privacy Policies: Recent Progress and Future Promise». In: *ACM Trans. Manage. Inf. Syst.* 12.1 (2020). ISSN: 2158-656X. DOI: 10.1145/3389685. URL: https://doi.org/10.1145/3389685.

[82] Arnob Zahid, Jennifer Kay Poulsen, Ravi Sharma, and Stephen C Wingreen. «A systematic review of emerging information technologies for sustainable data-centric health-care». In: *International Journal of Medical Informatics* 149 (2021), p. 104420.