IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 14, 2019, accepted July 11, 2019, date of publication July 17, 2019, date of current version August 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929330

Secure Development of Big Data Ecosystems

JULIO MORENO[1], EDUARDO B. FERNANDEZ[2], MANUEL A. SERRANO[3], AND EDUARDO FERNÁNDEZ-MEDINA[1]

[1]GSyA Research Group, University of Castilla-La Mancha, 13071 Ciudad Real, Spain
[2]Computer and Electrical Engineering and Computer Science Department, Florida Atlantic University, Boca Raton, FL 33431, USA
[3]Alarcos Research Group, University of Castilla-La Mancha, 13071 Ciudad Real, Spain

Corresponding author: Julio Moreno (julio.moreno@uclm.es)

This work was supported in part by the Ministerio de Economía y Competitividad and the Fondo Europeo de Desarrollo Regional (FEDER) through the ECLIPSE Project, in part by the Consejería de Educación, Cultura y Deportes de la Dirección General de Universidades, Investigación e Innovación de la JCCM, through the GENESIS Project under Grant SBPLY-17-180501-000202, and in part by the Programa Operativo Regional FEDER 2014/2020.

ABSTRACT A Big Data environment is a powerful and complex ecosystem that helps companies extract important information from data to make the best business and strategic decisions. In this context, due to the quantity, variety, and sensitivity of the data managed by these systems, as well as the heterogeneity of the technologies involved, privacy and security especially become crucial issues. However, ensuring these concerns in Big Data environments is not a trivial issue, and it cannot be treated from a partial or isolated perspective. It must be carried out through a holistic approach, starting from the definition of requirements and policies, and being present in any relevant activity of its development and deployment. Therefore, in this paper, we propose a methodological approach for integrating security and privacy in Big Data development based on main standards and common practices. In this way, we have defined a development process for this kind of ecosystems that considers not only security in all the phases of the process but also the inherent characteristics of Big Data. We describe this process through a set of phases that covers all the relevant stages of the development of Big Data environments, which are supported by a customized security reference architecture (SRA) that defines the main components of this kind of systems along with the key concepts of security.

INDEX TERMS Big Data, security by design, secure development, security patterns, security reference, architecture.

I. INTRODUCTION

Nowadays, companies are more aware of Big Data importance [1]. Data are crucial to conduct their daily activities and to help senior management to achieve business goals and, as a result, take better decisions based on the information extracted from such data [2]. The usage of a Big Data ecosystem implies a change compared to traditional techniques in three different ways: the amount of data (volume), the rate of generation and transmission of data (velocity) and the heterogeneity of the types of structured and unstructured data that it can handle (variety) [3]. These properties are known as the three Vs of Big Data [4]. This is the traditional definition of Big Data; however, different authors have added new V's to adapt its definition to the current state; for example, the veracity of the data, or the value obtained after performing the algorithms [5], [6]. A Big Data ecosystem can be defined as the set of different components that allow to store, process, visualize and deliver useful insights to target applications. Usually these components are very complex and need to work together in order to obtain valuable information [7].

The use of new technologies brings new opportunities and perspectives; however, they can also cause new problems, and Big Data is not an exception. These issues are related not only to the V's of Big data, but also to privacy and security. Big Data not only increases the size of the problems related to privacy and security, as faced in the traditional management of security, but also adds new threats and vulnerabilities that should be addressed with different techniques and measures [8]; for example, how to check the veracity of the data sources that feed the Big Data ecosystem [9]. Furthermore, these security problems are potentiated due to the fact that Big Data was not conceived initially as a secure environment [10], and therefore, the main security problems

The associate editor coordinating the review of this manuscript and approving it for publication was Zhitao Guan.

are related to the specific architecture of Big Data itself which makes it harder to protect the privacy of the data that it is being used [11].

For that reason, when a company decides to develop a Big Data ecosystem, it is important to consider these security and privacy issues which can affect how it is implemented. If these problems are not addressed properly, they can lead to difficulties that can affect the organization itself; for example, failure to comply with laws related to the context of the Big Data ecosystem may result in the loss of the company's reputation or even fines and lawsuits. Therefore, without guaranteeing its security, Big Data will not reach an appropriate level of confidence [12]. Hence, it is important to have methodologies, mechanisms, and guidance to properly implement not only the Big Data ecosystem, but also its security. In addition, security-by-design trends are becoming significant. This approach highlights the importance of tackling the security from the early stages of the design process [13].

Hence, the creation of a secure Big Data ecosystem is usually a very complex task that should be supported by guidance and methodologies to guarantee its success. Due to these problems, we have defined a process that integrates security aspects into the development of a Big Data ecosystem, and at the same time, considers its inherent characteristics. Our proposal is composed by twelve different phases covering the main stages of development, including analysis and design which, normally, are not sufficiently considered in this kind of scenarios. Moreover, it is important to highlight that a process of this kind should not be only a description of a set of activities; in fact, it should be supported by a conceptual framework that defines the main components of the system to develop [14]. In our case, we needed a metamodel that covers the main components of a Big Data ecosystem and, at the same time, incorporates security aspects into them; for this, we have defined a customized Security Reference Architecture (SRA) for Big Data [15]. This paper represents an evolution of that work, since it uses the architecture as a basis to build a secure process to develop Big Data ecosystems.

A SRA is usually defined as a high level architecture that incorporates a set of elements facilitating the definition of security requirements and allowing a better understanding of security policies, threats, vulnerabilities, etc., which can be used to describe a conceptual model of security for Big Data systems [16]. The use of an SRA allows a better control of the threats and vulnerabilities of the system, evaluating which can be stopped or mitigated from a risk assessment process. Therefore, the use of the components and concepts defined in the SRA can better support our process, while at the same time, we address the problem of the typical complexity of this type of systems. SRAs have become useful tools that allow a better understanding of complex systems [17], such as cyber-physical systems [18]. In this way, our SRA is designed to allow the use of patterns of different kinds to ease the implementation of the system and improve the addition of non-functional requirements [19]. In this case, we will focus on security patterns to ease the implementation of security mechanisms in Big Data ecosystems.

Additionally, this paper includes an example of how to apply our process following the components of our architecture; in this example, we have designed a Big Data ecosystem from scratch: first considering the requirements of it, and finally, implementing the security solutions that can tackle the different threats that can affect Big Data; for example, the security patterns that can help in the solution of those problems.

We organize the content of the paper as follows: in the first section, we explain some background, including the best known methodological security approaches for any kind of IT system; then we discuss the main proposals for Big Data reference architectures. As stated before, our process is supported by a SRA, for that reason, the next section explains it. Section 4 is focused on explaining the entire process including all its phases. Next section explains an example of how to use our SRA, following the previously defined process, to create a secure Big Data ecosystem. Finally, we present conclusions and future work.

II. BACKGROUND

As we described in the introduction, the use of a Big Data ecosystem brings new security problems that must be addressed. We carried out a study about the main security problems in Big Data ecosystems [20] that highlights that the main issues are related with data privacy and how to assure the Big Data architecture itself. These problems can be tackled by using general mechanisms like user authorization and authentication, fraud detection, risk control, auditing, encryption, network access control, or guarantee the quality of the data when it comes from an unreliable data source [21], [22]. However, these are general security mechanisms that must be modified in order to be applied in a specific context such as Big Data. For example, how to guarantee the data exchange in an edge computing context [23], how to ensure the cluster management to protect it from malicious access [24] or how to check the provenance of the data in an IoT scenario [25].

On the other hand, there are not many proposals that deal with the problem of security in Big Data from a methodological perspective. Therefore, to build a process to incorporate security in Big Data developments, we carried out a study of the main proposals for security methodologies. These methodologies are usually focused on software systems, so they cannot be fully used to deploy a secure Big Data ecosystem. On the other hand, we tried to find different alternatives to SRAs in Big Data ecosystems; however, none was found. Nevertheless, we were able to find some Reference Architectures (RA) that allow the abstraction of the main components of a Big Data environment. For that reason, this section is organized in two subsections: the first one explains the best known security approaches in development methodologies, and the second one, defines the main proposals of reference architecture for Big Data ecosystems.

A. SECURITY APPROACHES IN DEVELOPMENT METHODOLOGIES

There are many proposals related to how to address security in software development. In [14], the authors carried out a complete analysis of the quality of the main proposals in this topic. Therefore, in this subsection, we describe the main proposals.

Tropos [26] is a methodology that aims to build agent oriented software systems. This proposal is based on two main ideas: on one hand, the use of mentalistic notions; for example, goals or plans which are used in the entire process of software development. On the other hand, it highlights the importance of the early phases of requirements analysis. This allows a better understanding of the environment. Secure Tropos is an extension of this methodology that focuses on security goals and security requirements elicitation. It allows the integration of security concepts throughout the entire development process. For this, Secure Tropos uses an extended version of the i* language that includes concepts like goals or tasks [27].

SecureUML [28] is a modeling language for model-driven development that has the main purpose of securing distributed systems. Its approach is based on role-based access control. For this, it defines a meta-model that incorporates concepts like Roles, or Permissions. On the other hand, UMLSec [29] focuses on modeling security properties at the design stage by using a UML-based language. In order to support this purpose, it defines a UML extension with stereotypes, tagged values and constraints that allow the specification of security requirements. These two proposals are not competitors; on the contrary, they can complement each other, since UMLSec can help in defining the dynamic analysis, while SecureUML defines the static part of the security aspects to develop [30].

Another common perspective when facing the development of systems is to make use of the concepts proposed by Jackson [31], this approach is called problem-based frames. A problem frame is a mechanism for classifying problems that arise during development. This approach places special emphasis on the specification and definition of requirements. However, this approach is not usually used for security.

SERENITY [32] is a pattern-based methodology specially focused on Ambient Intelligence (Aml) systems. It is composed by two parts that cover the development and operation for the selection of security and dependability solutions. Its main characteristics are its dynamism, distribution and heterogeneity. SERENITY proposes a security goal approach which guides the discovery of requirements and the selection of patterns. The Secure Unified Process [33] incorporates security principles and disciplines into the Unified Process. The Unified Process can be considered as de facto standard for the software application development process. SysML-Sec [34] proposes a model-driven approach in which it proposes a stronger collaboration between the designers and the security experts; this approach is more oriented to systems where safety is an important requirement. ASE methodology [66] allows the incorporation of security mechanisms in distributed systems. To achieve this purpose it uses patterns.

All these approaches are too general and must be adapted to the specific context in which they will be applied. Furthermore, these proposals mainly focus on modeling software systems, while a Big Data ecosystem requires a double perspective: on the one hand, it is necessary to consider the services it will provide; on the other, the developer should not forget the infrastructure part of the system. Both layers will interact with each other and influence their development. However, we can learn a lesson from these proposals: most of them use UML as a way to express the particularities of the system they model and facilitate the incorporation of security concepts.

B. REFERENCE ARCHITECTURES FOR BIG DATA

An RA is an abstract software architecture that is based on one or more domains and with no implementation features. Moreover, an RA should be expressed at a high level of abstraction, in order to be reusable, extendable, and configurable [35]. Different authors and organizations have proposed different RAs for Big Data. As for standardization proposals, the RA defined by the NIST organization has gained relevance in this topic, therefore we will define its proposal in more detail. On the other hand, the ISO/IEC organization is currently working in the creation of a RA for Big Data under the standard ISO/IEC 20547-3 [36]. However, as it is a work in progress at the time of writing this article, it is not possible to comment much on its content.

For the last several years, the NIST has defined an RA for Big Data which has received the general consensus of the industry and scientific community [37]. With the release of last version on June 2018, this architecture collects many different ideas and features for creating a Big Data ecosystem. This set of features were extracted from the proposals of a Big Data architecture made by the main companies of the sector, such as, Oracle and IBM. The architecture is divided into five different components that interact with each other and have different objectives. In order to face the security problems, this architecture has a Security and Privacy Fabric that addresses the needs and solutions about this specific topic. In fact, there exists a specific volume about privacy and security in Big Data [38]. However, the NIST proposal cannot be considered as a SRA because it does not approach the security as a main requirement but as a fabric that is kept in the background. Here, the security concerns are addressed from a holistic perspective, rather than considering the security of each component of the Big Data ecosystem. From our point of view, this representation based on blocks is not expressive enough. Figure 1 represents the RA proposed by NIST. This kind of specification is too high level in terms of abstraction, it provides little emphasis on details of the subcomponents and how they are connected. This approach may not be expressive enough to assist in design and implementation of a Big Data ecosystem. Even though, this proposal has those

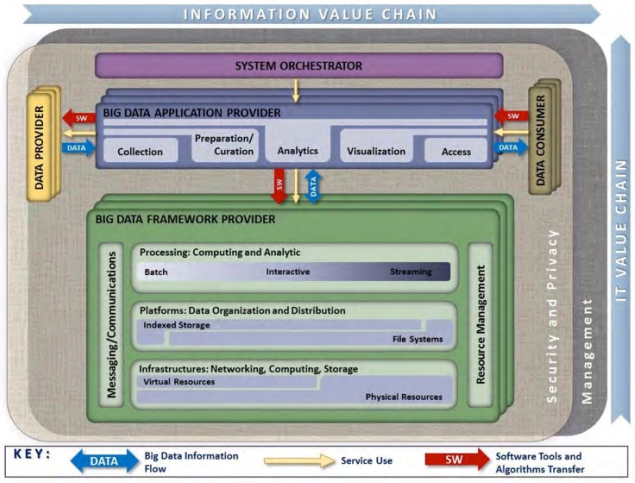

FIGURE 1. NIST proposal for a big data architecture.

issues, it is a well-conceived architecture that was the basis we used to create our own SRA.

Demchenko *et al.* [7] propose a Big Data Framework Architecture that establishes the data lifecycle of a Big Data ecosystem. As in the NIST approach, they use a block representation; but with a more detail in the relationships between the different components of the architecture. However, they address security in a very sketchy way and as an isolated feature, not really connected to the other components. In [39] the authors propose a complete architecture in terms of the relationships between the different components; however, we found a lack of consideration given to security and privacy aspects.

Klein et al. propose in [40] a specific reference architecture for Big Data to address the typical national defense requirements. Their architecture is very similar to the one proposed by NIST. Our goal is to obtain a better abstraction of the architecture, but still it is interesting how they address some concerns by using solution patterns. They highlight the importance of having a specific domain for the requirements. In our case, requirements, and specifically the ones related to security, are the fundamental pillars on which the SRA is based. Nadal *et al.* [41] propose a software reference architecture for semantic-aware Big Data ecosystems named Bolster, it follows the λ-architecture principles to which they add a semantic layer. They provide a very interesting approach; however, their proposal is more focused on the data lifecycle in a Big Data ecosystem; therefore, they do not have as an objective approaching security requirements. Following this approach, in [42] the authors propose a software architecture for Big Data that considers from the definition of requirements to their implementation. However, like other proposals, it contemplates security as a complement to be considered, not as a crucial feature for a successful implementation.

BlueTalon [43] proposes a Big Data model focused on data-centric security. Their purpose is to embed security information within the data itself. There are other proposals made by the main IT companies like Oracle [44], NTT data [45], IBM [46], Microsoft [47], or SAP [48] which are not focused on security, and are also aligned with their own technological stack.

III. SECURITY REFERENCE ARCHITECTURE (SRA) FOR BIG DATA

In this section, we briefly describe our SRA proposal which is based on the schema and components following the guidelines proposed by NIST. Our SRA is aligned with the RA proposed by NIST, so it can be easier to implement. Moreover, this architecture highlights the importance of implementing security solutions based on concepts of the SRA. We have created a SRA described by means of UML diagrams that try to facilitate the implementation of secure Big Data. We decided to use UML diagrams because we found a lack of proposals where the relationships between the different components and subcomponents are precisely defined. Also, thanks to this kind of diagram it is possible to apply different security patterns, which are usually described as UML models. As stated before, this SRA was more in-depth described in [15].

Our proposal focuses primarily on the requirements and security solutions that are described on the first component of our architecture: the System Orchestrator (SO). Thus, the requirements and security solutions are implemented in the other components of the SRA. It is important that these security requirements are aligned with the goals and policies of both the organization and the Big Data environment. These security requirements can be satisfied through different security solutions that follow the company's security policies and have the main objective of counteracting threats and controlling vulnerabilities. At this level, security requirements and solutions are still abstract objects that will be implemented in the rest of the components of the SRA. To facilitate their implementation, our SRA allows the use of security patterns. A security pattern is an abstract solution to a recurring problem that describes how we defend ourselves from a threat, or set of threats, in a concise and reusable form [49]. Therefore, it can be said that the SO is the most abstract of our architecture and will influence the implementation of the rest of components.

The next component of our architecture is the Big Data Application Provider (BDAP), which has the objective of satisfying the requirements established in the SO. To do this, the BDAP is composed of the different services offered by Big Data. In general, these services are five: collection (collecting the data that feed the analytics), preparation (cleaning or structuring the data to improve the results), analysis (algorithms to obtain valuable information from the data), visualization (representation of the data) and access control (who can access what data). It is not mandatory that all Big Data ecosystems provide all these services, there are some optional services such as the preparation or visualization of the results, which depending on the context may not be necessary. These services are implemented at the hardware level in the next component.

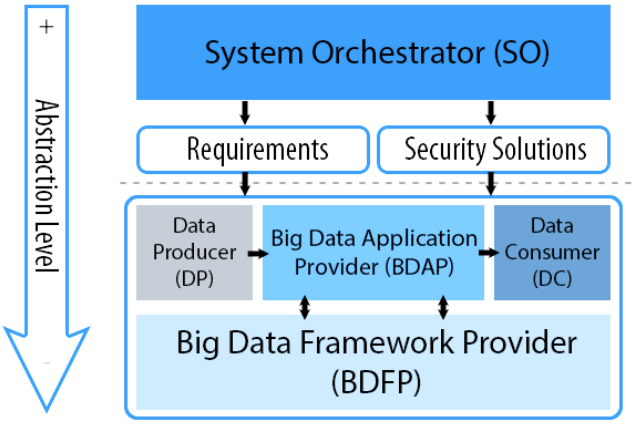

FIGURE 2. Main components of the SRA.

The Big Data Framework Provider (BDFP) supports the functionalities of the BDAP. In order to do this, it is usually composed of one or more clusters that in turn are composed of nodes. In addition to the hardware infrastructure, this component provides storage, processing and other services such as communications or resource management. Currently, many companies (especially small and medium) decide to outsource this part of the architecture by hiring a commercial cloud solution, on which they build their Big Data ecosystem.

Finally, the last two components of the SRA are the Data Producer (DP) and the Data Consumer (DC), which have a similar function, but at opposite edges of the architecture. On one hand, the DP is responsible for feeding data to the Big Data ecosystem, serving as a connection point with data sources, these data sources can be both structured and unstructured. On the other hand, the DC is the component that consumes the information generated by the Big Data ecosystem, serving as a connection point with the end user of the data. This end user does not have to be a physical person but can be another system. Fig.2 shows the structure of the SRA with its components and how they relate to each other.

IV. PROCESS TO INCORPORATE SECURITY TO BIG DATA DEVELOPMENTS

After describing the different components of our proposal, in this section, we describe how to properly use the SRA. Our process is composed by 11 phases, each of them is composed by different activities with input and output artifacts. The process follows the recommendations from the security-by-design culture [13] by considering the security since the early stages of the process, including security aspects during the whole process. As a result, this process can be considered as a guide of good practices that will improve the security of a Big Data ecosystem. In this section, we first define the process that we recommend following and then, we show an example of how to implement the security of a Big Data ecosystem from scratch.

A. PHASES OF THE PROCESS

When carrying out a Big Data project, it is important to highlight that it is quite different from a traditional software development. Big Data ecosystems are usually very complex systems where different technologies interact together to reach a goal. Furthermore, this kind of systems are normally implemented in companies where the time-to-market and the need to adapt to different changes is crucial. Moreover, many of those organizations are immersed in an internal cultural evolution in order to be more agile and innovative, such as DevOps movement [50].

Consequently, due to this pressure and the misunderstanding and misuse of the agile methodologies, the development of Big Data ecosystems usually does not make enough emphasis on the analysis and design phases, incrementing the technological debt.

For those reasons, our proposal tries to solve this problem by performing a light analysis and design phases. Therefore, our process for using the SRA has two different set of phases: on the one hand, analysis and design, and on the other hand, implementation.

The initial phases focus primarily on the definition of requirements, security solutions and risks, which will guide the implementation of the Big Data ecosystem on the second set of phases. These two set of phases are closely related to each other, since once the analysis and design are completed for the first time, it does not mean that the artefacts obtained are definitive. In fact, our process contemplates the possibility that during the implementation phase new requirements will emerge, and therefore, it will be necessary to go back to define those new requirements, the security solutions and the risks that are related to them. Once the process goes back, it does not mean that the whole process needs to be restarted; for example, if during the implementation of the Analyzer component a security requirement is discovered on how to guarantee the privacy of the sensitive data, then, the first three phases must be performed again, but the changes made may not affect the rest of the components; in contrast, if the changes do affect the other components, a new iteration of the implementation phase must be performed. Fig. 3 depicts the different phases of the process. In the following subsections, each phase will be defined. The artefacts involved in these phases are not described with an excessive level of detail so as not to overhead the scope of this paper.

1) PHASE 1: REQUIREMENTS DEFINITION

This phase is composed by four different activities that are shown in Table 1. The main goal of this phase is to obtain the requirements of the Big Data ecosystem. The first activity in this phase is the definition of Big Data goals, this is referred to as the main purpose of the Big Data ecosystem that will be implemented. Indeed, to have a useful and valuable system, these goals must be aligned with the policies and the business goals of the company. There are a few approaches that deal with the problematic of representing and obtaining goals;

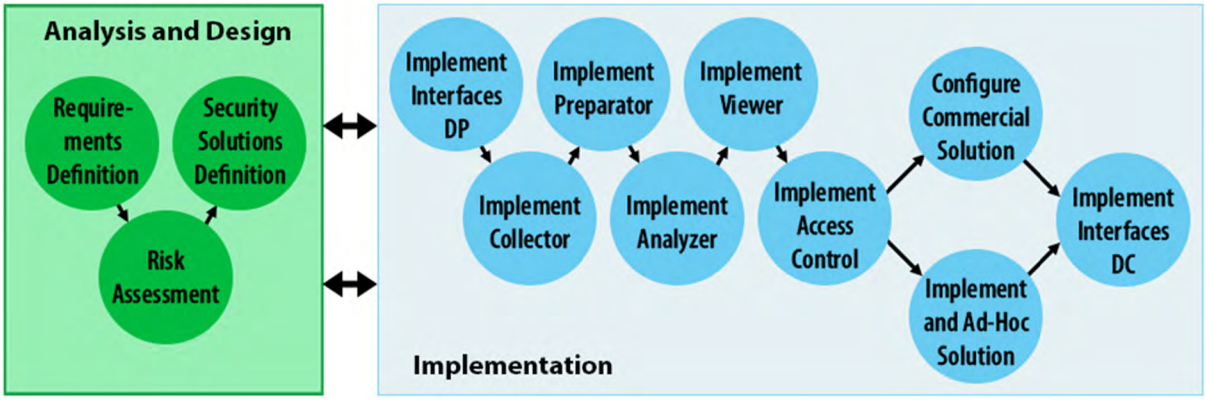

FIGURE 3. Process to implement the SRA.

TABLE 1. Activities of phase 1.

Activities	Description	Inputs	Outputs
Define Big Data Goal	Definition of Big Data goal and sub-goals that must be aligned with the business goals and polices.	Repository with the business policies Repository with the business goals	Big Data Goals
Define require-ments	Definition of requirements and security requirements based on the goal and sub-goals defined, and the context of the system.	Repository with the security policies of the company Big Data Goals Regulations	Requirements Security requirements
Select assets	Selection of assets that will address the previously defined requirements	Requirements Context of the company	Assets
Acquire assets	Acquisition of the selected assets	Requirements Context of the company	Assets

for example, the GORE (Goal-Oriented Requirement Engineering), i∗, or KAOS (Knowledge Acquisition in autOmated Specification). None of these methodologies are specific for Big Data ecosystems; however, they can be used to achieve this purpose [51]–[55]. Furthermore, the goals obtained can in turn be divided into more specific sub-goals, which can be represented by means of an AND-OR graph; this will allow a better understanding of the Big Data implementation.

The second activity is focused on the definition of the requirements, and more specifically the security requirements of the desired Big Data ecosystem. In order to do so, not only the goals and sub-goals obtained in the previous activity should be considered, but also the context of the company. The context of a company is a set of characteristics that can change the requirements of a Big Data; for example, the security requirements of a Big Data ecosystem implemented in a hospital should be especially strong in terms of privacy. Moreover, the context of the system also includes the different legal regulations that can affect the system, and therefore, its requirements. In order to properly specify the security requirements, there are a few methods that can be used; for example, UMLSec [29] is an UML extension focused on specifying security requirements regarding confidentiality, integrity, and availability to develop secure systems, or security uses cases [56] which represent scenarios focused on security issues. Based on the problem-frame approach, previously mentioned in section II.A, we can highlight the abuse-frames proposal [57], which introduces the concept of anti-requirement. An anti-requirement expresses the intentions of a malicious user, this can help with the definition of system threats. A more in-depth analysis of different security requirements definition methods can be found in [58].

Finally, the third and fourth activities are dedicated to the selection and acquisition of the assets that can approach the requirements defined in the previous activities. In general, there are six different types of assets that can be identified in a Big Data ecosystem: the hardware infrastructure, the services and applications, the data and metadata, the analytical resources, the security and privacy techniques, and the individuals and roles [37]. It is important to carry out a rigorous study of the different possibilities to decide which one is the option that best fits your Big Data requirements. The selection of assets will highly influence the implementation of the Big Data, so it is necessary to check the compatibility between the different elements before acquiring them. In some cases, the assets are already part of the company so there is no need to acquire them. A widely used solution for this type of problem are decision-making trees which allow the comparison of advantages and disadvantages between different possibilities.

At the end of this phase, a list of requirements and assets will be obtained. However, these are still tentative lists because they can be updated due to the emergence of new requirements as the different phases of the process progress. Obviously, in addition to security requirements, there are other types of requirements that are also crucial to implement

TABLE 2. Activities of phase 2.

Activities	Description	Inputs	Outputs
Define vulnerabilities	Definition of vulnerabilities that can affect the assets	Assets	Vulnerabilities
Define threats	Definition of threats that can exploit the vulnerabilities	Vulnerabilities	Threats
Risk Assessment	Definition of impact that the threats can have on the assets and the prioritization of risk based on that study	Threats Vulnerabilities Requirements Assets	Prioritized risks

the Big Data ecosystem and that are used following the common methods.

2) PHASE 2: RISK ASSESSMENT

The second phase has three activities that are shown in Table 2. The main goal of this phase is to define the risks that can affect the Big Data ecosystem. When we refer to risk we follow the traditional definition of risk: where a potential harmful event has a probability to happen and a potential severity over the elements involved [59]. Therefore, the first activity is the definition of the vulnerabilities that can affect the assets. The selected assets will probably have a set of vulnerabilities that are already identified by the community. These vulnerabilities can be exploited by threats. However, there are more threats that should be considered, for example, ENISA (European Union Agency for Network and Information Security) organization has created a list of the main threats that can be found in Big Data [60]. Another option is to consider the NIST Vulnerability and CVE database which contains a huge repository of threats and vulnerabilities (not only for Big Data environments). Also, there are different techniques that ease the discovery of threats like attack trees, misuse cases, or misuse activities [49].

Once all the risks of the Big Data ecosystem are identified, the risk assessment activity will focus on doing a quantitative and qualitative analysis of the risks. Therefore, based on that analysis a prioritized list of risks will be obtained. This list will allow stakeholders to decide how to deal with the risks, for instance, some risks are major and need to be prevented, and on the other hand, there are others that are not as important and accepted by the organization. The decision of this classification will also depend on the risk appetite of the company. There is not a specific method to deal with Big Data risks, however, there are many proposals for IT risk assessment in general that can be used; for example, MAGERIT [61], OCTAVE [62], CRAMM [63], CORAS [64], or ISO 31000 [59].

As it happened with the requirements, the discovery of new vulnerabilities, threats and risks is an on-going phase that can evolve during the process of implementing the Big Data ecosystem.

TABLE 3. Activities of phase 3.

Activities	Description	Inputs	Outputs
Find security solutions	Definition of the security solutions that can tackle the threats of the system	Threats Prioritized risks	Security solutions Security metadata
Select security patterns	Discovery of the security patterns that can ease the implementation of the security solutions	Security solutions	Security patterns to use
Find misuse patterns	Use of well-known misuse patterns that can guide the discovery of new threats to the Big Data ecosystem	Assets Threats Security patterns	Threats updated Vulnerabilities updated

3) PHASE 3: SECURITY SOLUTIONS DEFINITION

This phase is composed by three activities that are summarize in Table III. Their main purpose is defining the security solutions that will tackle the threats and risks defined in the previous phase. Also, the definition of these security solutions can lead to the creation of security metadata that can help in the implementation. However, the security solutions defined at this phase are still in a very abstract level, so they must be implemented in the lower levels of the architecture where the threats can actually affect the assets. There are methods to select the most appropriate security solutions; for example, in [65] the authors propose a mechanism that supports decision making to define the best set of security controls according to the family of standards in ISO/IEC 27000.

The second activity in this phase is the selection of security patterns, as stated before, security patterns are artefacts that realize security solutions. There are some methodologies proposed by the community to address the problem of applying security patterns in the implementation of an IT system [49], [66], [67]. In general, these methodologies propose a process to cover the security aspects that is similar to our approach, so they can be used together. However, it is possible that there is no security pattern that mitigates a specific threat or vulnerability; in that case, the security solution should be created from scratch. Another possibility is adapting security patterns from other fields.

Finally, the third activity can be considered as a way to improve the security aspects of the ecosystem since its main purpose is the identification of threats and vulnerabilities that were not considered before. Therefore, the use of misuse patterns is an interesting practice from the point of view of the security. It is based on the goals of the attacker related to the assets of the system, so it gives a new perspective. A misuse pattern defines an unauthorized use of an asset and how this

TABLE 4. Activities of phase 4.

Activities	Description	Inputs	Outputs
Identify data sources	Identification of the different data sources that will feed the Big Data ecosystem	Requirements Assets	Data source metamodel
Implement interfaces with data sources	Implementation of the "gates" between the data sources and the system	Data source metamodel Security requirements Requirements	DP interfaces

attack is performed. It also describes the countermeasures that can be used to reduce that risk [68]. To our knowledge, there are not specific misuse pattern for Big Data scenarios, however, it is possible to adapt the existing ones to this kind of environments or even create them [69].

4) PHASE 4: IMPLEMENT INTERFACES DP

From this phase onwards, the implementation of Big Data begins. The activities forming this phase are summarized in Table 4. The main goal of this phase is the description of the data sources that will feed the Big Data ecosystem, as well as the restrictions that must be applied due to the security requirements of the Big Data ecosystem and the data sources themselves. Therefore, the first step is the definition of the different data sources that will be used to meet the requirements defined. Usually a Big Data ecosystem will use different data sources to perform its analysis. Because of that, a good practice is the creation of a metamodel of the different data sources that depicts how data is connected and which data will be used. This metamodel will be used later during the implementation of the Collector component. Once the metamodel is completed, the access restrictions to the data must be implemented by using interfaces. These interfaces are software implementations of the security requirements of the Big Data ecosystem and the policies that the data sources can have.

This phase is highly connected with the implementation of the Collector, for that reason, sometimes it is possible to perform them at the same time. This approach allows a better alignment between these two components.

5) PHASE 5: IMPLEMENT COLLECTOR

This phase starts the implementation of the BDAP component. As we stated before, this component can be considered as the SaaS layer of the architecture. The activities that conform this phase are summarized in Table 5. Although, phase 5 has the main purpose of implementing the collection service, the first activity to perform is the definition of the data lake.

A data lake is a storage repository that holds a huge amount of raw data as it was generated, while it is still not necessary to process it. In general, data lakes store unstructured data but they can combine different kinds of data [70]. The data lake

TABLE 5. Activities of phase 5.

Activities	Description	Inputs	Outputs
Define the data lake	Definition of the data lake that will be used in the Big Data ecosystem	Data source metamodel Requirements Assets	Assets updated Data lake definitions
Implement the Collector component	Implementation of the Collector component, which is in charge of feeding the Big Data	Requirements Assets Data lake definitions	Data lake Registry of software
Implement security solutions for the Collector	Implementation of the security solutions defined during phase 3	Security solutions Security patterns Assets Data lake	Assets updated

is part of the Collector component, as it stores the raw data received from the data sources. For that reason, it is important that it is aligned with the defined requirements. Indeed, this data lake can be better managed if we use metadata to try to tackle the problematics of having a huge amount of disorganized data [71].

The second activity is focused on implementing the collection service which has the purpose of obtaining data from different data sources. Depending on the kind of data source that it is needed, you may need to use different applications. For example, if the data is stored in a relational database, it can be exported to the Big Data storage by means of Apache Sqoop. Collector also considers the data that must be analyzed in real-time, for instance, in a scenario where a log file requires to be processed and we need to store its changes. All this data will be stored in the data lake. Finally, the third activity is about applying the security solutions that were previously defined. In some cases, it will be possible to apply security patterns to ease this process, otherwise, security solutions must be implemented ad-hoc. Security of the data at this level is critical, and there are many issues to consider; for example, how to guarantee the confidentiality of the data or how to measure the acceptable level of privacy for a record.

6) PHASE 6: IMPLEMENT PREPARATOR

During this phase, the preparation service will be implemented. Usually, in these scenarios only a small part of the data is truly useful for the objective, so in order to properly analyze the data this phase is highly recommended. Furthermore, this service is highly related to one of the V's of Big Data: the value. The identification of the needed data is the first activity of this phase. In order to do it, it is important to consider which is the goal that wants to be achieved by the analysis. For that reason, the requirements are one of the inputs of this activity. As output, a repository of tagged data will be generated, where the data that will be used in the analysis phase is marked. This phase can be very important

TABLE 6. Activities of phase 6.

Activities	Description	Inputs	Outputs
Identify the important information from the data	Identification of the data that will be used to perform the analysis	Requirements Assets Data lake	Repository of tagged data
Define scripts to prepare the data	Implementation of the scripts that will prepare the data to be analyzed	Requirements Assets Repository of tagged data Collector's data	Scripts for data preparation Registry of software
Implement security solutions for the Preparator	Implementation of the security solutions defined during phase 3	Security solutions Security patterns Assets Scripts for data preparation	Assets updated

TABLE 7. Activities of phase 7.

Activities	Description	Inputs	Outputs
Design the algorithms for the data analysis	Definition of the algorithms and how they will be implemented	Requirements Assets	Big Data algorithms description
Implement the algorithms	Implementation of the algorithms to perform the analysis of the data	Analytical algorithms Repository of tagged data	Big Data algorithms implemented Registry of the expected information Registry of software
Implement security solutions for the Analyzer	Implementation of the security solutions defined during phase 3	Security solutions Security patterns Assets Data lake	Assets updated

in some scenarios in which the transformation of raw data into information (data wrangling) is crucial to perform the analysis.

After that, it is the moment to implement the different scripts that will transform the data to ease its analysis. Some techniques that can be used to prepare the data include the detection of missing values and outliers that can deteriorate the analysis of the data. Also, there are many commercial applications that focus on easing this data preparation. Finally, as it happened in the previous activity, it is necessary to apply the security solutions to this component. In this case, as preparation scripts can have access to personal data, it is important to control how they are implemented to guarantee that they are working as they should. Table summarizes the activities of this phase.

7) PHASE 7: IMPLEMENT ANALYZER

Phase 7 has as main purpose implementing the analysis service. In general, this service is the most important one in a Big Data ecosystem. This phase has three activities that are summarized in Table 7. First of all and based on the requirements and the assets already defined, it is important to determine how the desired insights will be produced. In other words, describe the algorithms and the technology to implement them. There are different ways to obtain value from the data, so the algorithms to use will be determined by the requirements about how to analyze it and which is the value that needs to be obtained. For example, it is possible to use an approach based on machine or deep learning techniques, without forgetting about the most knowledgeable way to perform analysis in Big Data: MapReduce (although nowadays is slowly falling into disuse) [72].

In terms of security problems, since a lot of environmental and human behavior is used by Big Data to obtain valuable insights, the main problem is about how to protect privacy. Many times, it is difficult to find a balance between obtaining useful information while guaranteeing the privacy of the users [73]. Another typical problem to consider in Big Data ecosystems is the inferred information from the data. In Big Data, it is possible to obtain sensitive information from data that did not have a special level of sensitivity. Therefore, these scenarios must be considered when approaching the security problems at this level.

8) PHASE 8: IMPLEMENT VIEWER

This phase has the main purpose of implementing the Viewer component. The visualization service provides representation of the information obtained. This phase has three activities that are explained in Table 8. This service is not mandatory for all cases; for example, if the information is consumed by another system the visualization component is not required. To decide which visualization technique is the most appropriate it is important to have a strong knowledge of the stakeholders that will use the information, in order to meet their requirements. In general, the visualization techniques can be divided into two categories: on one hand, the data can be visualized as a graph or a chart of any kind; on the other hand, data can also be represented by means of a dashboard, in this case, the information represented is more focused on the top management of an organization.

In terms of the security solutions, at this level it is important to worry about what information can the stakeholders look for. This issue will be largely considered in the next phase, although, there are still some details to be covered. For example, it is possible that because of the representation of information, a data scientist may infer personal information that needs to be protected. In this case, an additional layer of protection is needed to prevent this from happening.

9) PHASE 9: IMPLEMENT ACCESS CONTROL

Finally, at the end of this phase the BDAP can be considered as completed. This phase is focused on defining and implementing the access control rules and it is composed by two activities that are summarized in Table 9. The access control is a service that has the main goal of restricting read

TABLE 8. Activities of phase 8.

Activities	Description	Inputs	Outputs
Decide the best way to visualize the information	Definition of the methods of visualization that best fit the requirements	Requirements Assets Registry of the expected information	Rules for visualization
Implement visualization techniques	Implementation of the visualization methods decided, for example, graphs or dashboards.	Registry of the expected information Rules for visualization	Visualization techniques Registry of software
Implement security solutions for the Viewer	Implementation of the security solutions defined during phase 3	Solutions Security patterns Assets	Assets updated

TABLE 9. Activities of phase 9.

Activities	Description	Inputs	Outputs
Define access control rules	Definition of the rules for access control of each stakeholder	Requirements Assets Registry of expected information	Access control rules
Implement access control rules	Implementation of the access rules defined in the previous activity. This can be helped by using security patterns	Access control rules Security solutions Security patterns Registry of expected information	Access control implemented Registry of software

access to the information. In Big Data environments there are usually distinct stakeholders that must only access a part of the information. Indeed, this phase depends for its success on how well the requirements of the stakeholders were defined. Based on those requirements and the information obtained in the analysis service, the access control rules should be well-defined, and then implemented. This implementation is highly influenced by the technology that is being used, because each one has a different way to provide access control. For example, Apache Spark uses Kerberos to perform the authentication. Actually, these access control rules are the implementation of security solutions that were previously defined in the SO component (with a higher level of abstraction); this is why in this phase there is not a specific activity for that purpose. Indeed, this implementation can be also helped by using security patterns. In this phase, it is important to identify the different users and roles that will interact with the Big Data ecosystem, as this can also influence the definition and implementation of access control rules.

10) PHASE 10: IMPLEMENT BDFP COMPONENT

This phase is focused on implementing the BDFP component, although, there are two main ways to approach this: by implementing an ad-hoc solution that better meets the Big Data requirements or by using a commercial solution. For that reason, phase 10 can be divided in these two possibilities.

11) PHASE 10A: IMPLEMENT AN AD-HOC SOLUTION

The objective of this phase is to implement the necessary hardware architecture to perform the BDAP services. In order to achieve this purpose, this phase consists of five activities that focus on different parts of the BDFP component, these activities are summarized in Table 10.

The first activity consists of deploying the clusters and nodes that conform the Big Data ecosystem. In a Big Data context, a cluster is usually defined as a group of nodes that have different functions to achieve one main objective: to obtain valuable information from the data [74]. There are two node configurations that can be used depending on the needs of your project: on one hand, whether each node has a specific function, on the other hand, that all nodes will have a standard configuration that fits the requirements. A typical way to take advantage of the nodes of a Big Data environment are containers. A container is an aggregation of different technologies that exist in the operating system and that allow an application to run, usually a single process, within an operating system. In general, the container is completely linked to the life cycle of its process: when the container is started, the container process begins, when the process ends, so does the container. The container comprises only the application and its dependencies. It runs as an isolated process in user space on the host operating system, sharing the kernel with other containers. It therefore partly enjoys the resource isolation and resource allocation benefits of virtual machines, but is much more portable and efficient [75]. Nonetheless, these technologies have the purpose of doing a high-level management of the underlying hardware.

The second activity concerns the implementation of the storage system. In general, there are three different ways to store data in Big Data ecosystems depending on the data format and ecosystem requirements: structured, semi-structured, and non-structured. Structured storage can be considered as traditional relational databases; normally, in this type of storage, a language similar to SQL is used to query the data. On the other hand, unstructured data storage, generally known as NoSQL databases, are widely used in Big Data ecosystems. This type of storage has four different subtypes: graph databases (usually used to represent social network data), columnar databases (in these systems each key is associated with one or more attributes, unlike relational databases. They are suitable for analytical applications where many common operations are performed on the data), document databases (these databases store the data in document form; their main advantage is scalability), and key-value (similar to hash tables where each key is associated with a set of values). In addition, in this activity it is important to emphasize the possibility of tools that facilitate the realization of the consultations that can be made on the stored data; for example,

TABLE 10. Activities of phase 10a.

Activities	Description	Inputs	Outputs
Deploy cluster and nodes	Estimate the size and number of nodes that the Big Data needs and deploying it	Requirements Assets Registry of software Data lake definitions Scripts for data preparation Big Data algorithms Visualization techniques Access control rules	Registry of hardware resources
Implement storage system	Decision about the kind of storage system to use and its implementation	Requirements Assets Registry of software Data lake definitions	Storage system implemented
Implement processing engine	Decision about the kind of processing engine to use and its implementation	Requirements Assets Registry of software Scripts for data preparation Big Data algorithms	Processing engine implemented
Implement communication platform	Implement the commu-nication platform between the different components	Requirements Assets Registry of software Registry of hardware resources	Communication platform implemented
Implement resource management	Implement resource use control	Requirements Assets Registry of software Registry of hardware resources	Resource management implemented

it is possible to consult NoSQL databases with a language similar to SQL.

The third activity focuses on the implementation of the processing layer. In the context of Big Data, there are three different types of processing. Again, depending on the needs of your project, you should use the configuration that best suits your needs. Batch processing is usually related to the MapReduce paradigm, executing the different jobs in sequential mode. In addition, it writes to disk to store results between phases, which limits its speed. On the other hand, streaming processing manages data in real time. In the midst of these technologies is interactive processing, a possibility that is becoming increasingly relevant in Big Data environments [37]. These solutions allow queries to be made over the data while it is being retrieved.

TABLE 11. Activities of phase 10b.

Activities	Description	Inputs	Outputs
Decide a commercial virtual solution	Choose the IaaS provider that best meets the requirements of the Big Data ecosystem	Requirements Assets Registry of software Data lake definitions Scripts for data preparation Big Data algorithms Visualization techniques Access control rules	Assets updated Registry of hardware components
Configure the commercial virtual solution	Configure the selected provider to address the needs of the BDAP's services	Requirements Assets Registry of software Data lake definitions Scripts for data preparation Big Data algorithms Visualization techniques Access control rules	Assets updated Registry of BDFP components

The last two phases cover the functionalities known as support services. Communications functionality refers to how the different components or processes of the Big Data ecosystem communicate with each other. The other functionality is resource management. Its purpose is to control and manage how node resources are used. This functionality is especially important if a node configuration is used in which each node has different technologies in operation.

12) PHASE 10B: CONFIGURE COMMERCIAL SOLUTIONS

On the other hand, it is possible to abstract the technology and components of the Big Data ecosystem by using a cloud IaaS. These services can facilitate the implementation of the BDAP component making it transparent for the user. Therefore, this is a good option if you do not need an ad-hoc solution for your system or if you are a newcomer to Big Data ecosystems. This phase has two activities explained in Table 11.

First, there are many different vendors that could be considered. To choose the one that best suits the needs of the Big Data ecosystem, not only must the requirements be considered, but also the technologies that have been selected to implement the BDAP services. In addition, there are other criteria to take into account, for example, economic or reputational attributes of the provider. There are general techniques that can help in this decision, e.g. decision tree diagrams.

TABLE 12. Activities of phase 11.

Activities	Description	Inputs	Outputs
Identify data consumers	Identification of the different data consumers that will consume the insights produced by the Big Data	Requirements Assets	Data consumers
Implement interfaces with data consumers	Implementation of the "gates" between the data consumers and the system	Data source metamodel Security requirements Requirements	DC interfaces

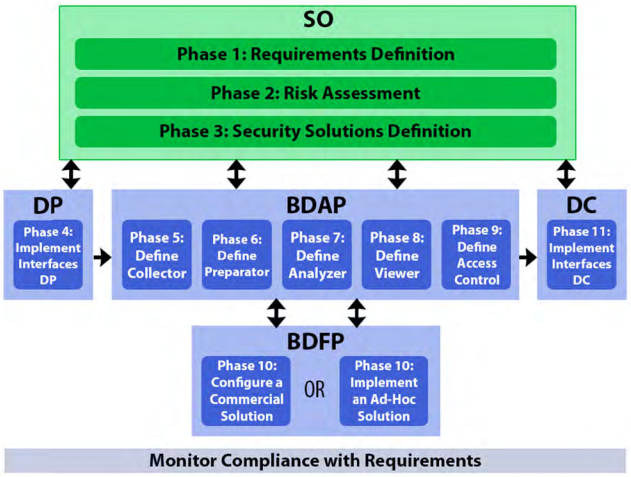

FIGURE 4. Process to use the SRA and its relationships with the main components.

Once the provider has been selected, there is another activity to be performed: configuring the IaaS. Depending on the selected provider, the configuration possibilities change. This activity should cover all the features necessary to support the BDAP, including the type of storage and the streaming engine. Typically, this type of IaaS includes a dashboard that facilitates monitoring of all system components and allows for flexible hardware configuration. This configuration phase can follow a flow of activities similar to that described in phase 10A.

13) PHASE 11: IMPLEMENT INTERFACES DC

This is the last phase to implement the Big Data ecosystem. The activities included in this phase are summarized in Table 12. The main goal of this phase is the description of the data consumers that will use the information produced by the Big Data ecosystem, as well as the restrictions that must be applied due to the security requirements of the Big Data ecosystem. Consequently, the first step is the definition of the different data consumers and the access constraints to the information. Usually a Big Data ecosystem has different stakeholders interested in accessing the information; however, depending on their roles they will have different restrictions. As it happened in the DP consumer, the use of different diagrams can help the implementation of this component, for example, UML sequence diagrams. Once this is completed, the access restrictions to the information must be implemented by using interfaces. These interfaces are the gates that protect the access to the insights generated in the Big Data ecosystem and can be considered as an implementation of the security solutions and requirements that were specified in the SO component.

Finally, the implementation of the Big Data ecosystem can be considered as completed. The operation of the system will be the next step; however, this is out of the scope of our SRA. Furthermore, it is important to highlight that there is a specific component that is in charge of monitoring that every requirement is covered in the implemented system. This functionality is not only limited to the implementation stage, but also, to the operation stage. Fig. 4 sums up the different phases of the architecture and places them along the architecture to ease its use.

B. EXAMPLE OF USE

As a way to show the usefulness of our SRA, we show an example of how to use the process to implement a Big Data ecosystem; in this example we want to emphasize the importance of the security patterns in our proposal. This example will be mainly focused in the BDAP component. To simplify the case, we consider that the first phase about requirements definition has already been done; as a result, a list of requirements is obtained derived from the Big Data goal, which is to detect incidents of racism from the tweets published on the Twitter platform. The security requirements are listed in Table 13. The column "Category" expresses the type of security requirement according to the classification made by the OWASP.[1] These categories include from application security to the context and regulations that can affect the ecosystem.

The next phase is the identification of the risks and security solutions that meet the requirements. In this case, we have identified some of the threats that can be found in the different activities of the BDAP component. A systematic method for the enumeration of threats is shown in [49]. Those threats can be addressed by means of security patterns, which, in some cases, should be modified from general security patterns to meet the Big Data inherent features. Those patterns help the implementation of security solutions that handle the threats. Table 14 summarizes some of the threats of each activity of the BDAP, related to the previously defined security requirements, and the general patterns that can be applied to solve them. Those patterns are defined in [49]. The other components of the SRA also are affected by different threats that must be controlled, but as we have already explained, in this example we will focus mainly on the BDAP component.

In order to better understand how to integrate the different components of our SRA and the corresponding security patterns, we will define how the threat TC3 can be addressed

[1] https://www.owasp.org/index.php/High_Level_Requirements_Categories

TABLE 13. List of security requirements derived from the big data goal.

ID	CATEGORY	Definition	Components involved
SR1	Application security	Input data must be validated	DP and BDAP
SR2	Application security	Data sources must be verified	DP
SR3	Encryption	Sensitive data must be encrypted	BDAP and BDFP
SR4	Application security	Mandatory authentication to access the data in its entire lifecycle	BDAP and BDFP
SR5	Application security	Provide access control mechanisms	BDAP and BDFP
SR6	Auditability	Audit all the actions performed on the data	All
SR7	Application security	Guarantee high-availability of the system	BDFP
SR8	Compliance	Comply with the GDPR regulations regarding the processing of personal data	All
SR9	Additional security considerations	Perform daily backups of the stored data	BDFP
SR10	Application security	The system cannot be accessed from outside the company	DC
SR11	Compliance	Data can be deleted at the request of its data owner	BDAP, BDFP and DC
SR12	Application security	Monitor possible inference of sensitive data derived from the results of the analysis	BDAP and DC
SR13	Application security	Verify the identity of the end-user	DC
SR14	Application security	Check that the developed scripts do not perform any malicious action on the data	BDAP

by using security patterns. In this scenario, we have the stored data as the main asset to protect, this asset has a vulnerability: it has no protection; this vulnerability could be exploited by a threat like TC3. In order to prevent that situation is necessary to implement a security solution. To facilitate the implementation of the solution, three security patterns can be used: Encryption, Role-based access control, and Authentication. However, we are still in an early step of the methodology (in the SO component, see Figure 2), so this security solution will be defined at a high abstraction level. Hence, a lower-level implementation of the security solution should be approached in the BDAP level, in this case, the TC3 can affect the different services provided by the BDAP; that is the reason why the security solution should be implemented there and not in another component.

TABLE 14. Identified threats and security patters for the different activities.

ID	ACTIVITY	Threat	Security patterns	Sec. Reqs.
TC1	Common to all the activities	Data modified	Authenticator, Role-based access control	SR4 SR5
TC2	Common to all the activities	Data destroyed	Authenticator, Role-based access control	SR4 SR5 SR11
TC3	Common to all the activities	Data illegally read	Encryption, Role-based access control, Authenticator	SR4 SR5
TC4	Common to all the activities	Unapproved change in activity function	Logger and Auditor, Controlled access session, Role-based access control, Authenticator	SR4 SR5 SR6
TC5	Common to all the activities	Fine for non-compliance with regulations	Encryption, Logger and Auditor	SR8
TCo1	Collection	Malicious data source	Authenticator	SR1 SR2
TP1	Preparation	Malicious filter	Logger and Auditor, Controlled access session, Role-based access control, Authenticator	SR14
TA1	Analysis	Infer Personally Identifiable Information (PII) from anonymized data	Encryption, Logger and Auditor, Multilevel security, Role-based access control, Authenticator	SR3 SR12 SR14
TA2	Analysis	Malicious analysis algorithms	Logger and Auditor, Controlled access session, Role-based access control, Authenticator	SR14
TV1	Visualization	PII* exposed due to high graphic granularity	Multilevel security, Authenticator, Role-based access control	SR12
TAc1	Access	Several malicious access	Authenticator, Role-based access control	SR4 SR5

Now the implementation phases can start (phases 4 to 11). Fig. 5 shows a simple view of how this example can be implemented by following our SRA. The tweets feed the Big Data ecosystem by using the Twitter REST API, that can be considered as the DP component of our SRA. Those tweets can be stored in a database, in this case, we have decided to use mongoDB as the Collector component. To do that, it is crucial to know the structure of the data that we are handling, in this case, the tweets. Usually, a tweet object follows a

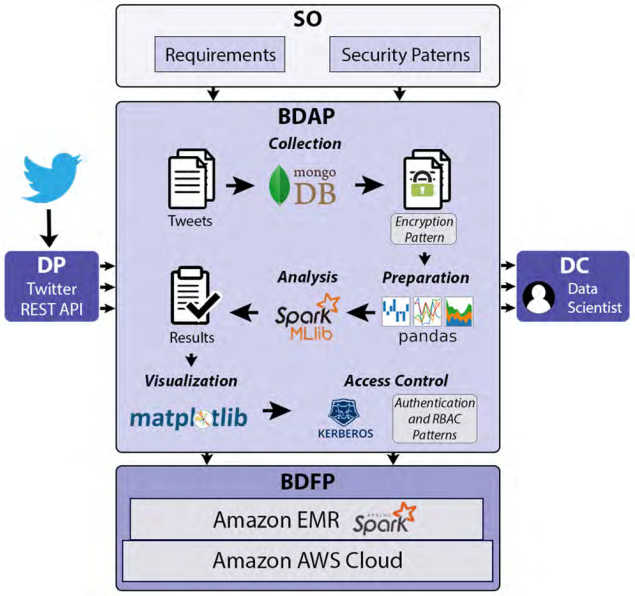

FIGURE 5. Example of use of the architecture.

JSON structure that contains different information about the user who published it (for example, the name, or location), and the tweet itself (for example, the time when it was created, or the mentions to other users). Some of this information can be sensitive information, in this case the information related to the user, therefore it must be protected. One way to protect it is to use an encryption scheme, which can be easier to implement when using the Encryption pattern.

The next step it is the preparation of the data, in this example, we do not really need all the data, the only information that we need are the text, the location, and the creation time of the tweet. There are a lot of libraries to perform this operation, we have decided to use the Pandas libraries. Once the data is ready, we can perform the analysis. For this example, we use machine learning techniques to discover when a racism event happens, so next time we can detect it before it happens.

We can implement those algorithms with the Apache Spark library for machine learning (Spark MLlib). Some results would be obtained from the execution of those algorithms, which can be represented by visual diagrams, so they can be easily understood. Matplotlib is our choice for that purpose. Finally, there are some restrictions about who can access which part of the results. The decisions made about which technology to use can cause the discovery of new threats and restrictions that must be addressed; the decision of using Spark creates the need of using another technology to provide access control. Kerberos allows us to define those rules of authentication for Apache Spark. This can be eased by using two different security patterns: Authenticator and Role-based Access Control. The Authenticator pattern allows us to verify the identity of the user by using a proof of identity and an Authenticator class that matches the proof to the Authentication data. On the other hand, as its name indicates, one of the most important things to implement the Role-based access control is to define the different roles. In this case, we have defined four roles: the administrator of the Big Data ecosystem, the data scientist, the end user, and the data owner. That data refers not only to the results obtained from the analytics, but also for the tweets stored during the collection phase.

All this architecture is supported by the BDFP component, which in this case, it its implemented by using an IaaS that virtualize all the resources. More specifically, in this example, the platform and services are provided by Amazon's AWS Cloud. Hence, in this case, the risks derived from the Cloud infrastructure are transferred to a third party. For example, SR7 on how to ensure the high-availability of the system should be covered in the Service Level Agreement (SLA) with the provider.

V. CONCLUSIONS AND FUTURE WORK

The development of a secure Big Data ecosystem is not a trivial project. In fact, it usually involves dealing with new security issues that had not previously been considered in other systems. In addition, such an ecosystem usually includes the use of different technologies that interact with each other, which complicates its implementation. For this reason, in this paper we present our proposal of a process to incorporate security to the development of a Big Data ecosystem. This process covers the typical phases of a development process from analysis to implementation. In addition, this process was conceived by considering the current scenario of companies, in which many of them are changing their internal culture to adopt concepts such as agile methodologies. This process is supported by a SRA that acts as a metamodel of the different components that usually conform a Big Data ecosystem allowing its abstraction, which will facilitate the development of such a complex environment. Finally, to illustrate this process, we have carried out an example of how to use the SRA that shows the main components of our SRA and how the security patterns can be applied to tackle the different threats that our ecosystem faces.

As future work, our proposal will be validated by means of a case study in a real environment that will allow us to elaborate a more complex example. On the other hand, the process will be refined through a more formal and detailed definition of all the phases and artefacts that conform our SRA. To this end, process modeling standards such as SPEM will be used.

REFERENCES

[1] J. Akoka, I. Comyn-Wattiau, and N. Laoufi, "Research on big data—A systematic mapping study," *Comput. Stand. Inter.*, vol. 54, pp. 105–115, Nov. 2017.

[2] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution that Will Transform how We Live, Work, and Think*. Boston, MA, USA: Houghton Mifflin Harcourt, 2013.

[3] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[4] S. Sagiroglu and D. Sinanc, "Big data: A review," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2013, pp. 42–47.

[5] M. Ali-ud-din Khan, M. F. Uddin, and N. Gupta, "Seven V's of big data understanding big data to extract value," in *Proc. Conf. Amer. Soc. Eng. Educ.*, Apr. 2014, pp. 1–5.
[6] Z. Sun, K. Strang, and R. Li, "Big data with ten big characteristics," in *Proc. 2nd Int. Conf. Big Data Res.*, Oct. 2018, pp. 56–61.
[7] Y. Demchenko, C. De Laat, and P. Membrey, "Defining architecture components of the big data ecosystem," in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2014, pp. 104–112.
[8] H. Wang, X. Jiang, and G. Kambourakis, "Special issue on security, privacy and trust in network-based big data," *Inf. Sci. Int. J.*, vol. 318, pp. 48–50, Oct. 2015.
[9] P. Jadon and D. K. Mishra, "Security and privacy issues in big data: A review," *Adv. Intell. Syst. Comput.*, vol. 841, pp. 659–665, Apr. 2019.
[10] P. P. Sharma and C. P. Navdeti, "Securing big data Hadoop: A review of security issues, threats and solution," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 2, pp. 2126–2136, 2014.
[11] B. D. W. G. Cloud Security Alliance (CSA). (20130.)*Expanded Top Ten Big Data Security and Privacy*. Accessed: Jan. 14, 2019. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Expanded_Top_Ten_Big_Data_Security_and_Privacy_Challenges.pdf
[12] B. Thuraisingham, "Big data security and privacy," in *Proc. 5th ACM Conf. Data Appl. Secur. Privacy*, Apr. 2015, pp. 279–280.
[13] V. Casola, A. De Benedictis, M. Rak, and E. Rios, "Security-by-design in Clouds: A security-SLA driven methodology to build secure cloud applications," *Procedia Comput. Sci.*, vol. 97, pp. 53–62, Jan. 2016.
[14] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "Assessing and improving the quality of security methodologies for distributed systems," *J. Softw. Evol. Process*, vol. 30, no. 11, 2018, Art. no. e1980.
[15] J. Moreno, M. A. Serrano, E. Fernandez-Medina, and E. B. Fernandez, "Towards a security reference architecture for big data," in *Proc. CEUR Workshop*, 2018, p. 2062.
[16] F. Liu, "NIST cloud computing reference architecture," *NIST Spec. Publ.*, vol. 500, p. 292, Sep. 2011.
[17] D. Garlan, *Documenting Software Architectures: Views and Beyond*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2010.
[18] V. M. Romero and E. B. Fernandez, "Towards a security reference architecture for cyber-physical systems," in *Proc. LACCEI Int. Multi-Conf. Eng., Educ. Technol.*, Jul. 2017, pp. 1–9.
[19] E. B. Fernandez, N. Yoshioka, H. Washizaki, and M. H. Syed, "Modeling and security in cloud ecosystems," *Future Internet*, vol. 8, no. 2, p. 13, Apr. 2016.
[20] J. Moreno, M. A. Serrano, and E. Fernández-Medina, "Main issues in big data security," *Future Internet*, vol. 8, no. 3, p. 44, Sep. 2016.
[21] E. Bertino, "Big data–security and privacy," in *Proc. IEEE Int. Congr. Big Data*, Jul. 2015, pp. 757–761.
[22] B. Saraladevi, N. Pazhaniraja, P. V. Paul, M. S. Basha, and P. Dhavachelvan, "Big data and Hadoop-a study in security perspective," *Procedia Comput. Sci.*, vol. 50, pp. 596–601, Apr. 2015.
[23] Z. Guan, Y. Zhang, G. Si, Z. Zhou, J. Wu, S. Mumtaz, and J. Rodriguez, "ECOSECURITY: Tackling challenges related to data exchange and security: An edge-computing-enabled secure and efficient data exchange architecture for the energy Internet," *IEEE Consum. Electron. Mag.*, vol. 8, no. 2, pp. 61–65, Mar. 2019.
[24] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, "Big data analysis-based secure cluster management for optimized control plane in software-defined networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 15, no. 1, pp. 27–38, Mar. 2018.
[25] M. S. Aktas and M. Astekin, "Provenance aware run-time verification of things for self-healing Internet of Things applications," *Concurr. Comput.*, vol. 31, no. 3, Feb. 2019, Art. no. e4263.
[26] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, "Tropos: An agent-oriented software development methodology," *Auton. Agents Multi-Agent Syst.*, vol. 8, no. 3, pp. 203–236, May 2004.
[27] H. Mouratidis and P. Giorgini, "Secure Tropos: A security-oriented extension of the tropos methodology," *Int. J. Softw. Eng. Knowl. Eng.*, vol. 17, no. 2, pp. 285–309, Apr. 2007.
[28] T. Lodderstedt, D. Basin, and J. Doser, "SecureUML: A UML-based modeling language for model-driven security," in *Proc. Int. Conf. Unified Modeling Lang.*, Sep. 2002, pp. 426–441, 2002.
[29] J. Jürjens, "UMLsec: Extending UML for secure systems development," in *Proc. Int. Conf. Unified Modeling Lang.*, Sep. 2002, pp. 412–425.
[30] R. Matulevićius and M. Dumas, "Towards model transformation between secureuml and UMLsec for role-based access control," in *Proc. DB&IS*, 2011, pp. 339–352.
[31] M. Jackson, *Problem Frames: Analyzing and Structuring Software Development Problems*. Boston, MA, USA: Addison-Wesley, 2001.
[32] D. Serrano, A. Maña, R. Llarena, B. G.-N. Crespo, and K. Li, "SERENITY aware system development process," *Adv. Inf. Secur.*, vol. 45, pp. 165–179, Mar. 2009.
[33] C. Steel, R. Nagappan, and R. Lai, "The alchemy of security design methodology, patterns, and reality checks," in *Proc. Core Secur. Patterns Best Practices Strategies J2EE, Web Services, Identity Manage.*, 2005, p. 1088.
[34] Y. Roudier and L. Apvrille, "SysML-Sec: A model driven approach for designing safe and secure systems," in *Proc. 3rd Int. Conf. Model-Driven Eng. Softw. Develop. (MODELSWARD)*, Feb. 2015, pp. 655–664.
[35] P. Avgeriou, "Describing, instantiating and evaluating a reference architecture: A case study," *Default J.*, vol. 342, pp. 1–24, Jun. 2003.
[36] *ISO/IEC CD 20547-3—Information Technology—Big Data Reference Architecture—Part 3: Reference Architecture*. Accessed: Nov. 1, 2017. [Online]. Available: https://www.iso.org/standard/71277.html?browse=tc
[37] NBD-WG, NIST. (2017). *NIST Big Data Reference Architecture*. Accessed: Oct. 8, 2017. [Online]. Available: https://bigdatawg.nist.gov/_uploadfiles/M0639_v1_9796711131.docx
[38] (2018). *NIST Big Data Interoperability Framework: vol. 4, Security and Privacy*. Accessed: Sep. 12, 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-4r1.pdf
[39] P. Pääkkönen and D. Pakkala, "Reference architecture and classification of technologies, products and services for big data systems," *Big Data Res.*, vol. 2, no. 4, pp. 166–186, Dec. 2015.
[40] J. Klein, R. Buglak, D. Blockow, T. Wuttke, and B. Cooper, "A reference architecture for big data systems in the national security domain," in *Proc. 2nd Int. Workshop BIG Data Softw. Eng.*, Austin, TX, USA, May 2016, pp. 51–57.
[41] S. Nadal, "A software reference architecture for semantic-aware big data systems," *Inf. Softw. Technol.*, vol. 90, pp. 75–92, Oct. 2017.
[42] I. Mistrik, R. Bahsoon, N. Ali, M. Heisel, and B. Maxim, *Software Architecture for Big Data and the Cloud*. Amsterdam, The Netherlands: Elsevier, 2017.
[43] BlueTalon. (2019). *BlueTalon Data-Centric Security Platform: Bringing Order to Data Security Chaos*. Accessed: Dec. 10, 2018. [Online]. Available: http://bluetalon.com/data-centric_security/
[44] D. Cackett, *Information Management and Big Data a Reference Architecture*. Redwood City, CA, USA: Oracle, 2013.
[45] N. DATA. (2018). *NTT DATA BigData Reference Architecture*. Accessed: Dec. 10, 2018. [Online]. Available: http://www.nttdata.com/global/en/shared/pdf/bigdata_reference_architecture.pdf
[46] I. Corporation. (2017). *IBM Big Data & Analytics RA*. Accessed: Jan. 16, 2019. [Online]. Available: https://www.ibm.com/cloud/garage/architectures/dataAnalyticsArchitecture/reference-architecture
[47] Microsoft. (2017). *Microsoft Big Data Solution Brief*. Accessed: Oct. 1, 2019. [Online]. Available: http://download.microsoft.com/download/F/A/1/FA126D6D-841B-4565-BB26-D2ADD4A28F24/Microsoft_Big_Data_Solution_Brief.pdf
[48] SAP ERP. (2016). *CIO Guide to Using the SAP HANAÂő Platform for Big Data*. Accessed: Sep. 1, 2019. [Online]. Available: https://www.sap.com/documents/2016/03/24d5e503-647c-0010-82c7-eda71af511fa.html
[49] E. B. Fernandez, *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns*. Hoboken, NJ, USA: Wiley, 2013.
[50] J. Carrasco, F. Durán, and E. Pimentel, "Trans-cloud: CAMP/TOSCA-based bidimensional cross-cloud," *Comput. Stand. Inter.*, vol. 58, pp. 167–179, May 2018.
[51] L. Liu, "Security and privacy requirements engineering revisited in the big data era," in *Proc. IEEE 24th Int. Requirements Eng. Conf. Workshops (REW)*, Sep. 2016, p. 55.
[52] H. Eridaputra, B. Hendradjaya, and W. D. Sunindyo, "Modeling the requirements for big data application using goal oriented approach," in *Proc. Int. Conf. Data Softw. Eng. (ICODSE)*, Nov. 2014, pp. 1–6.
[53] G. Park, L. Chung, L. Zhao, and S. Supakkul, "A goal-oriented big data analytics framework for aligning with business," in *Proc. IEEE 3rd Int. Conf. Big Data Comput. Service Appl. (BigDataService)*, Apr. 2017, pp. 31–40.
[54] N. Al-Najran and A. Dahanayake, "A requirements specification framework for big data collection and capture," in *Proc. East Eur. Conf. Adv. Databases Inf. Syst.*, Aug. 2015, pp. 12–19.

[55] I. Noorwali, D. Arruda, and N. H. Madhavji, "Understanding quality requirements in the context of big data systems," in *Proc. 2nd Int. Workshop Big Data Softw. Eng.*, Austin, TX, USA, May 2016, pp. 76–79.
[56] G. Sindre and A. L. Opdahl, "Eliciting security requirements by misuse cases," *Requir. Eng.*, vol. 10, no. 1, pp. 34–44, May 2005.
[57] L. Lin, B. Nuseibeh, D. Ince, and M. Jackson, "Using abuse frames to bound the scope of security problems," in *Proc. 12th IEEE Int. Requirements Eng. Conf.*, Sep. 2004, pp. 354–355.
[58] B. Fabian, S. Gürses, M. Heisel, T. Santen, and H. Schmidt, "A comparison of security requirements engineering methods," *Requir. Eng.*, vol. 15, no. 1, pp. 7–40, Mar. 2010.
[59] (2018). *Risk Management—Guidelines*. Accessed: Jan. 16, 2019. [Online]. Available: https://www.iso.org/standard/65694.html
[60] ENISA. (2016). *Big Data Threat Landscape and Good Practice Guide*. Accessed: Oct. 18, 2017. [Online]. Available: https://www.enisa.europa.eu/publications/bigdata-threat-landscape/at_download/fullReport
[61] M. V3, "Methodology for information systems risk analysis and management (MAGERIT version 3)," Ministerio de Hacienda y Administraciones Públicas, Madrid, España, Tech. Rep., 2012.
[62] C. J. Alberts and A. J. Dorofee, *Managing Information Security Risks: The OCTAVE Approach*. Old Tappan, NJ, USA: Addison-Wesley, 2002.
[63] *CRAMM v5.0, CCTA Risk Analysis and Management Method*, Insight Consulting, Leicester, U.K., 2003.
[64] R. Fredriksen, M. Kristiansen, B. A. Gran, K. Stølen, T. A. Opperud, and T. Dimitrakos, "The CORAS framework for a model-based risk management process," in *Proc. 21st Int. Conf. Comput. Saf., Rel. Secur.*, Sep. 2002, pp. 94–105.
[65] T. Neubauer, A. Ekelhart, and S. Fenz, "Interactive selection of ISO 27001 controls under multiple objectives," in *Proc. 23rd Int. Inf. Secur. Conf.*, 2008, pp. 477–492.
[66] A. V. Uzunov, E. B. Fernandez, and K. Falkner, "ASE: A comprehensive pattern-driven security methodology for distributed systems," *Comput. Stand. Inter.*, vol. 41, pp. 112–137, Sep. 2015.
[67] M. Schumacher, *Security Engineering with Patterns: Origins, Theoretical Models, and New Applications*. Berlin, Germany: Springer-Verlag, 2003.
[68] K. Hashizume, N. Yoshioka, and E. B. Fernandez, "Misuse patterns for cloud computing," in *Proc. 2nd Asian Conf. Pattern Lang. Programs*, 2011, pp. 12:1–12:6.
[69] E. B. Fernandez, N. Yoshioka, and H. Washizaki, "Modeling misuse patterns," in *Proc. Int. Conf. Availability, Rel. Secur.*, Mar. 2009, pp. 566–571.
[70] N. Miloslavskaya and A. Tolstoy, "Application of big data, fast data, and data lake concepts to information security issues," in *Proc. 4th Int. Conf. Future Internet Things Cloud Workshops*, Aug. 2016, pp. 148–153.
[71] C. Diamantini, P. L. Giudice, L. Musarella, D. Potena, E. Storti, and D. Ursino, "A new metadata model to uniformly handle heterogeneous data lake sources," *Commun. Comput. Inf. Sci.*, vol. 909, pp. 165–177, Sep. 2018.
[72] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: A survey," *J. Big Data*, vol. 2, no. 1, p. 21, Oct. 2015.
[73] K. Barker, "Privacy protection or data value: Can we have both?" in *Proc. Int. Conf. Big Data Anal.*, 2015, pp. 3–20.
[74] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, "The rise of 'big data' on cloud computing," *Inf. Syst.*, vol. 47, pp. 98–115, Jan. 2015.
[75] D. Steenken, S. Voá, and R. Stahlbock, "Container terminal operation and operations research—A classification and literature review," *Spectrure*, vol. 26, no. 1, pp. 3–49, Jan. 2004.

JULIO MORENO is currently pursuing the M.Sc. and Ph.D. degrees in computer science with the University of Castilla-La Mancha, Ciudad Real, Spain. He has the Spanish FPI Research and Investigation Fellowship. He is also a member of the GSyA Research Group, Information Systems and Technologies Department, University of Castilla-La Mancha. His research interests include data security and privacy and security architectures for Big Data ecosystems.

EDUARDO B. FERNANDEZ received the M.S. degree in electrical engineering from Purdue University and the Ph.D. degree in computer science from UCLA. He has been a Professor with the Department of Computer Science and Engineering, FAU, since 1984. He has published numerous papers as well as several books on computer security and software architecture. He has published numerous papers on authorization models, object-oriented analysis and design, cloud computing, and security patterns. He has written four books on these subjects, the most recent being a book on security patterns.

MANUEL A. SERRANO received the M.Sc. and Ph.D. degrees in computer science from the University of Castilla-La Mancha, Ciudad Real, where he is currently an Assistant Professor with the Escuela Superior de Informática. Regarding his research interests, he is working on cyber security (especially in Big Data and the IoT), data quality, software quality, and measurement and business intelligence. His scientific production is large, having published more than 50 papers in high-level journals and conferences. He has participated in more than 20 research projects, has conducted several invited speeches, and has worked in several transfer projects with companies. He has been teaching for nearly two decades at the university, especially in software engineering and programming subjects. He has supervised several final degree theses, final master works, and Ph.D. theses.

EDUARDO FERNÁNDEZ-MEDINA received the M.Sc. and Ph.D. degrees in computer science from the University of Castilla-La Mancha, Ciudad Real, Spain, where he is currently a Full Professor with the Escuela Superior de Informática (Computer Science Department).

His research interests are in the field of security in information systems, particularly in security in Big Data, cloud computing, and cyber-physical systems. He is a Co-Editor of several books and chapter books on these subjects and has published several dozens of papers in national and international conferences, such as BPM, UML, ER, ESORICS, and TRUSTBUS. He has authored more than 50 manuscripts in international journals, such as the *Decision Support Systems*, the *Information Systems*, the *ACM Sigmod Record*, the *Information Software Technology*, the *Computers and Security*, and the *Computer Standards and Interfaces*. He leads the GSyA Research Group, Department of Computer Science, University of Castilla-La Mancha, and belongs to various professional and research associations, such as ATI, AEC, and AENOR.