CEUR-WS.org/Vol-2062
urn:nbn:de:0074-2062-0

DOLAP 2018

Design, Optimization, Languages and Analytical Processing of Big Data

Proceedings of the 20th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data co-located with 10th EDBT/ICDT Joint Conference (EDBT/ICDE 2018)

Vienna, Austria, March 26-29, 2018.

Edited by

Robert Wrembel *
Alberto Abelló **
Il-Yeol Song ***

* Poznan University of Technology, Poznan, Poland
** Universitat Politècnica de Catalunya - BarcelonaTech, Barcelona, Catalonia (Spain)
*** Drexel University, Philadelphia, USA

Table of Contents

2018-02-15: submitted by Robert Wrembel, metadata incl. bibliographic data published under Creative Commons CC0
2018-02-15: published on CEUR-WS.org |valid HTML5|

Towards a Security Reference Architecture for Big Data

Julio Moreno
GSyA Research Group, University of Castilla-La Mancha
Ciudad Real, Spain
Julio.Moreno@uclm.es

Manuel A. Serrano
Alarcos Research Group, University of Castilla-La Mancha
Ciudad Real, Spain
Manuel.Serrano@uclm.es

Eduardo Fernandez-Medina
GSyA Research Group, University of Castilla-La Mancha
Ciudad Real, Spain
Eduardo.FdezMedina@uclm.es

Eduardo B. Fernandez
Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University
Boca Raton, Florida
Fernande@fau.edu

ABSTRACT

Companies are aware of Big Data importance as data are essential to conduct their daily activities, but new problems arise with new technologies, as it is the case of Big Data; these problems are related not only to the 3Vs of Big Data, but also to privacy and security. Security is crucial in Big Data systems, but unfortunately, security problems occur due to the fact that Big Data was not initially conceived as a secure environment. Furthermore, this task is difficult due to the heterogeneous configurations that a Big Data system can have. One way to solve this problem is by having a global perspective, and in this way, a Reference Architecture (RA) is a high-level abstraction of a system that can be useful in the implementation of complex systems. Several initiatives have been made for obtaining a RA for Big Data like those from IBM, ORACLE, NIST or ISO, but none of them have their main focus on security. It is widely accepted that adding elements to address threats and facilitate the definition of security requirements to RA is a good starting point for solving these kind of threats and, in this way, converting RAs into Security Reference Architectures (SRAs). In the current paper, a SRA for Big Data is defined using UML models trying to ease secure Big Data implementations; allowing to apply security patterns in order to secure final Big Data systems.

1 INTRODUCTION

Companies are increasingly aware of Big Data importance [1]. For all of them, data are essential to conduct their daily activities and to help senior management to achieve business objectives and, as a result, take better decisions based on the information extracted from such data [22]. Big Data 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) [6]. These properties are known as the three Vs of Big Data [30].

New problems usually arise with new technologies, as it is the case of Big Data. These problems are related not only to the 3 Vs of Big data, but also to privacy and security. Big Data not only increases the scale of the problems related to privacy and security, as faced in the traditional management of security, but also adds new ones that should be addressed with different techniques and measures [36]. These security problems occur due to the fact that Big Data was not conceived initially as a secure environment [33], and therefore, the main security problems 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 [7].

Obtaining an adequate level of security in Big Data can influence its implementation in an institution because of the loss of reputation they could suffer or because they could receive financial penalties, due to regulations, in the case of data breaches; in fact, without a security guarantee, Big Data will not reach an appropriate level of acceptance [35]. Hence, it is important to have guidance, methodologies, and mechanisms to properly implement not only the Big Data system, but also its security. Big Data environments are very complex, so in order to address their security, we need to start from a global perspective. Security should be approached from high-level policies that can be mapped to the lower levels [13]. Different authors [2, 23] highlight that Reference Architectures (RA) have been shown to be valuable to guide security in different environments; for example, Cloud Computing [13] or Internet of Things [19].

An RA is an abstract software architecture that is based on one or more domains and with no implementation features [2]. Moreover, an RA should be expressed at a high level of abstraction, in order to be reusable, extendable, and configurable. This kind of architecture can be composed of different patterns to facilitate the implementation of the system and improve the addition of non-functional requirements [15]. Adding security patterns to control their identified threats, RAs become a Security Reference Architecture (SRA). In this way, a SRA is a high level architecture that incorporates a set of elements facilitating the definition of security requirements and allowing better understanding of security policies, threats, vulnerabilities, etc., and which can be used to describe a conceptual model of security for Big Data systems [21].

Among our main concerns in computer security, our current goal is to improve the security and trust of Big Data environments. In order to achieve that objective, our first step is the creation of a SRA for Big Data. To do that, we consider that security patterns have a primordial role in facilitating the implementation of security mechanisms in a Big Data ecosystem. Hence, we modified the RA proposed by the National Institute of Standards and Technology (NIST) for Big Data [26] to create a richer architecture, in which the relations between the different parts of Big Data are clearly exposed with a more granular detail. This enhanced RA will allow a better understanding of the Big Data ecosystem. In order to achieve that purpose, our reference architecture is specified by means of UML diagrams

[29]. Finally, along with the SRA, we created a partial example of how to apply our architecture; we have considered some of the different threats that can affect a Big Data system, and how the different components that take part in addressing them can be instantiated; for example, security patterns that can help in the solution of those problems.

We organize the content of the paper as follows: first, we show a section which explains the main properties of the NIST proposal of an RA for Big Data. After that, we present the components and structure of our SRA, together with an example of how to use security patterns to address threats in a particular Big Data project. Subsequently, we compare our proposal with the main Big Data RA proposals. Finally, we include a section in which conclusions and future work are discussed.

2 REFERENCE MODEL: NIST REFERENCE ARCHITECTURE FOR BIG DATA

For the last several years, the NIST has defined a RA for Big Data which has received the general consensus of the industry and scientific community [26]. With the release of last version on August 2017, 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. As a result, NIST produced the RA that can be seen in Figure 1. The architecture is divided into five different components that interact with each other and have different objectives. These components are:

- System Orchestrator (SO): This is one of the most important components of a Big Data ecosystem because it is the one in charge of defining and integrating the required data application activities into the ecosystem. The main purpose of this component is the configuration and management of the other components of the Big Data architecture. In an enterprise, this function is typically centralized and can be mapped to the traditional role of system governor which provides the supervision of the requirements and constraints that the Big Data must fulfill; for example, policies, architecture, or business requirements.
- Data Provider (DP): This component oversees feeding the Big Data ecosystem with new data. In order to accomplish that goal, the Data Provider has a collection of interfaces, or services, between the Big Data and the data sources. This set of interfaces acts like a gate between the outside world and the Big Data system.
- Big Data Application Provider (BDAP): The BDAP component provides a specific set of services along the data life cycle to meet the requirements established by the SO. It is important to highlight that its main purpose is to encapsulate the business logic and functionality to be executed by the architecture. In a regular Big Data scenario, there are several applications executing over the same data. As data propagates through the ecosystem, it is being processed and transformed in different ways to obtain valuable information from the data. In order to achieve that goal, the BDAP is composed of different services or activities that can be considered as the SaaS layer of the Big Data system. These activities are: collection, preparation, analytics, visualization, and access. Activities can be implemented as independent functions and deployed as stand-alone services. Furthermore, the activities can interact with the underlying Big Data Framework Provider, as well as with the Data Consumer, DP or even with each other.
- Big Data Framework Provider (BDFP): The BDFP component can be considered as the platform implementation of the Big Data logic. It supports the activities defined in the BDAP. In general, Big Data implementations are hybrids that combine multiple technologies. It has three main activities: infrastructure (virtual or physical), platform (how the data is distributed and organized), and processing (how data will be processed to support Big Data applications). In addition, the BDFP component also provides the support services for the system like communications or resource management.
- Data Consumer (DC): It is similar to the DP component. Usually the actor that interacts with this component is an end-user or another system. Similarly to the DP, it is composed of a set of interfaces between the end-user and the information.

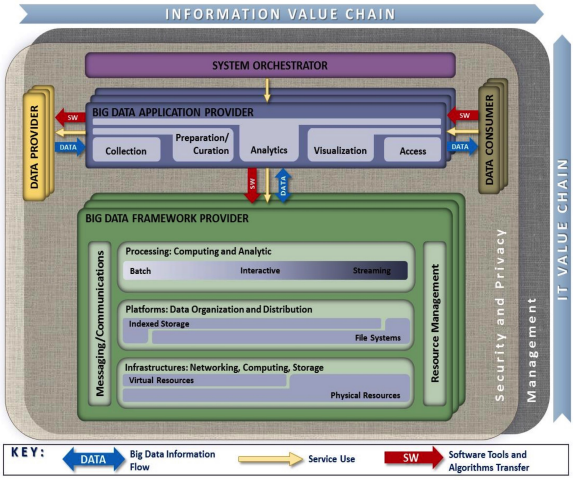

Figure 1: NIST proposal for a Big Data architecture [26]

The NIST proposal cannot be considered as a SRA, but it recognizes the importance of security and privacy in a Big Data environment. 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 [27].

From our point of view, this representation based on blocks is not expressive enough. 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 can make difficult the design and implementation of a Big Data ecosystem. Following the same approach, the ISO/IEC organization is also working in the creation of a RA for Big Data under the standard ISO/IEC 20547-3 [16]. Although, it is a work in progress, it is expected that it will follow a similar approach to the NIST proposal.

3 A SECURITY REFERENCE ARCHITECTURE (SRA) FOR BIG DATA

In this section, we will describe our SRA proposal which is structured using the same schema and components as the guidelines proposed by NIST. We consider that if our SRA is aligned with the RA proposed by NIST, it will be easier to implement. Furthermore,

this architecture highlights the importance of implementing security solutions based in concepts of the SRA.

3.1 System Orchestrator (SO)

The main purpose of this component is the enforcement of the different requirements that the Big Data ecosystem must address. Also, it organizes how the requirements are connected to all the components of the architecture; in this section, we will focus on the security requirements and the relation between them and the different components. Figure 2 shows the structure of our SO proposal. Due to the characteristics of this component, the security activities related to it are in general focused on the requirements and how to implement and monitor them. Those requirements must fulfill Big Data goals and should be aligned with the different business goals and company policies. In this concern, the role of the Security Administrator is crucial to ensure the observance of the security requirements. These security requirements must comply with the regulations affecting each Big Data ecosystem context. In fact, there are many other kinds of requirements that can address the needs of a Big Data ecosystem; for example, architecture, quality, or governance requirements.

There are many examples of security requirements that should be addressed in a Big Data context. Topics like data privacy and how to secure the Big Data architecture itself are the most addressed by researchers [25]. These problems can be tackled by using general mechanisms like user authorization and authentication, fraud detection, risk control, auditing, encryption, network access control, intrusion detection, or guarantee the quality and security of the data when they come from different data sources [3, 17, 20, 25, 32]. These are general security mechanisms but they must be modified to be applied to specific types of systems, based on possible threats.

As it is shown in Figure 2, these security requirements can be satisfied by means of different security solutions that follow the security policies of the company and have as main objective addressing threats to control vulnerabilities. An example of a security policy in a company can be the obligation of using secure communications, this policy can cause a security requirement in the Big Data environment that specifies that the data transfer between components must be secure. One way to approach requirement is by using authentication methods, the implementation of this security solution can be helped by means of the "Role-based access control" security pattern. These security solutions should be specifically implemented in the BDAP and BDFP components. However, these solutions are not easy to implement; thus, our model uses security patterns as a guidance. A security pattern is a solution to a recurrent problem that indicates how to defend against a threat, or a set of threats, in a concise and reusable way [12]. Patterns are abstract solutions that must be tailored to where they are applied. Furthermore, we can use misuse patterns [14] as a way to understand each attack and guide the application of the different security patterns that can be used to stop a threat. Moreover, the security metadata can be defined as a way to facilitate the coordination and realization of security requirements. Another topic covered by our architecture is the context of the asset; for example, the security considerations of a medical record, are totally different compared to the ones of a log file. It is important to evaluate the required security level for each asset.

3.2 Data Provider (DP)

The DP component creates an abstraction of the data sources considering their security metadata, if they exist. These metadata allow the DP to identify the types of access and analysis allowed by the data source and its security requirements. As we explained in section 2, the DP has a set of interfaces. Those interfaces must consider the constraints of each data source and also the different security policies and requirements specified by the SO. In this element, there may exist conflicts between the security requirements of the data source and the ones of the Big Data system itself. These clashes must be addressed in a way that satisfies both sides. The security and privacy issues of this component are mostly related to how to properly identify and validate the end point inputs. The DP interfaces must evaluate the provenance of the data source. It is a critical challenge in the data collection process knowing how to validate that a data source is not malicious and to filter out those which are [7].

In our SRA, the interfaces are connected with the Collector service of the BDAP that will be described in the next subsection. Figure 3 represents the DP component with its interfaces. In general, the elements that generally compose a data source, include: the data itself that can be structured, semi-structured, or unstructured; security requirements of the data source; and security metadata of the data source. Those elements are not represented in the diagram because we consider data source as an external agent of the Big Data system. Still it is important to know them to apply their constraints.

3.3 Big Data Application Provider (BDAP)

The BDAP component has the objective of meeting the requirements established by the SO, including its security and privacy requirements. To achieve that goal, the BDAP is composed of different services or activities that can be considered as the SaaS (Service as a System) layer of the Big Data ecosystem; in our case, we assume that, in general, Big Data is implemented on a Cloud platform, which will affect how the SRA is defined in the BDFP component. Figure 4 shows the different services that constitute this component, and also the BDAP Security Solution that must map the SO security solutions to these stages; for example, authorization may control here who can apply which operations to perform data analysis.

As it is represented in the diagram, not all the activities can communicate with each other, there is a sequential order of execution. This means that some of these activities are not mandatory in a Big Data ecosystem. The preparation step has the purpose of validating, cleaning and storing the data, but in a real-time scenario where the data should be analysed as soon as it gets into the system, this activity might be skipped. Something similar happens to the visualization step, if the data consumer is not a human end-user but another system, like a data warehouse or even another Big Data ecosystem, this activity may not be necessary.

Nevertheless, the other three activities are basic in a Big Data ecosystem: the collection activity acts like an ETL (Extract, Transform, and Load) process and combines sets of data of similar structure with the objective of unifying them; the analysis step includes a set of techniques to obtain valuable knowledge from data; for example, MapReduce algorithms and finally, the access activity has the purpose of communicating with the DC, acting like an interface between DC and visualization and analytics

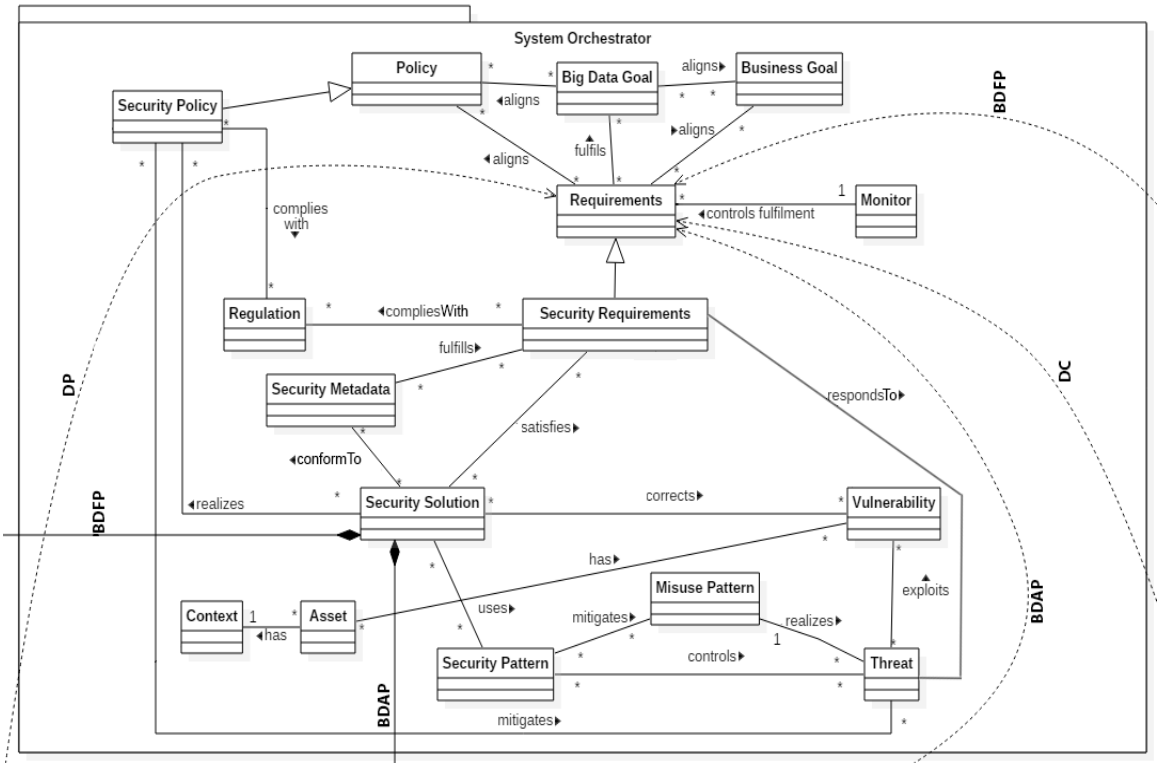

Figure 2: System Orchestrator (SO) diagram

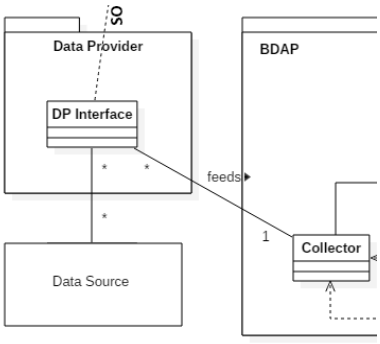

Figure 3: Data Provider (DP) diagram

activities. The relation between those different activities is represented in Figure 4 by dotted lines, because it is a temporary usage relation.

3.4 Big Data Framework Provider (BDFP)

In general, the BDFP component is composed of a set of clusters which, in turn, are composed of nodes. Those nodes can be deployed by means of Virtual Machines or Containers, which interact with the hardware itself and the OS.

The BDFP component in NIST is very abstract, with a lack of details in the subcomponents needed to perform its processes. Therefore, our proposal makes more emphasis in the different elements and how they are connected. Figure 5 depicts the different subcomponent of the BDFP. Our SRA highlights the idea of a Big Data ecosystem with the possibility of implementing the system with a Cloud environment and visualization techniques.

In regard to security and privacy issues, in this component the activities should be focused on the encryption and key management of the data, the isolation and containerization of process execution, authorization, authentication, audit logging, and how to secure the storage and the network. Those security issues should be addressed by means of the security solutions defined on the SO, which can be implemented in this level as BDFP security solutions. The SO security solutions are now mapped to data protection, including application of cryptography and specialized authorization mechanisms [8, 37].

3.5 Data Consumer (DC)

The DC component is, similarly to DP, composed by a set of interfaces. The interaction could include interactive visualization, report creation, or data drilling using business intelligence techniques. It is important to highlight that these interfaces must address the authorization and authentication function, in order to reach the goal that the DC matches the metadata related to the end-user and the security requirements and policies of the information.

Finally, Figure 6 summarizes our complete SRA for Big Data. In this figure, the relationships between the different components of the architecture can be seen in perspective. This figure is important to better understand the example which is presented in the following subsection.

3.6 Examples of Application of Security Patterns

As a way to show the usefulness of our SRA, we explain an example of how to employ security patterns using our architecture. We created the example by identifying some of the threats that

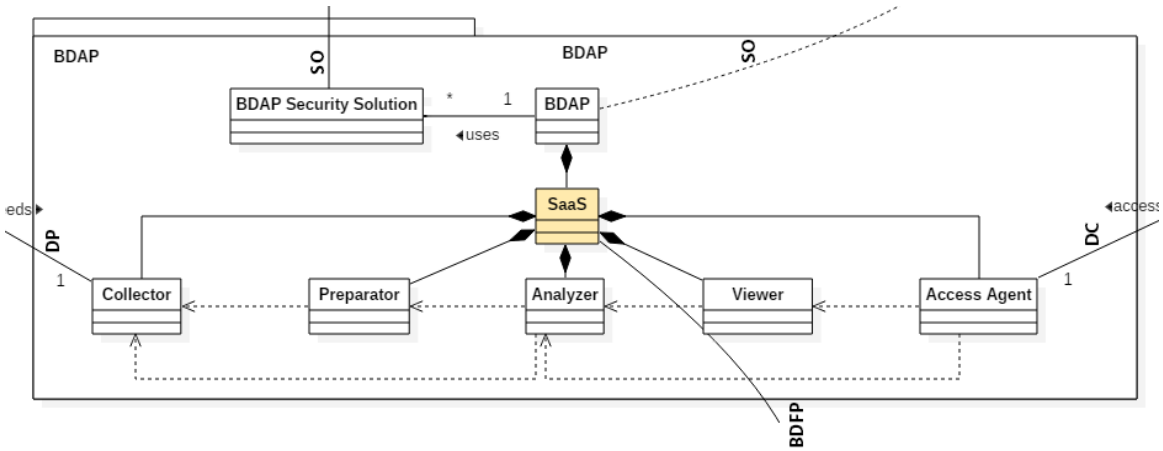

Figure 4: Big Data Application Provider (BDAP) diagram

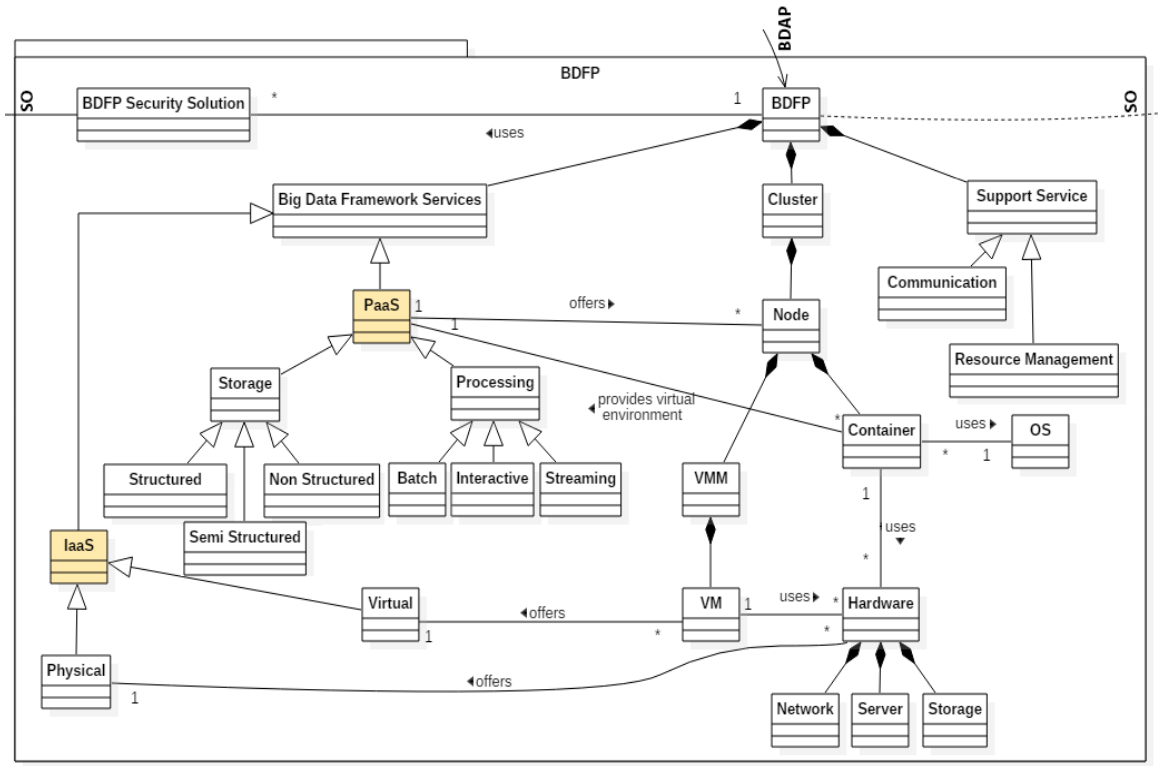

Figure 5: Big Data Framework Provider (BDFP) diagram

can be found in the different activities of the BDAP component. A systematic method for the enumeration of threats is shown in [12]. 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. The modification of these patterns, and the creation of new ones if needed, is beyond the purpose of this paper and is considered as future work. Table I summarizes some of the threats of each activity and the general patterns that can be applied to solve them. Those patterns are defined in [12].

As a way to better understand how to integrate the different components of our SRA and the security patterns, we will define how the threat TC1 can be addressed by using security patterns. We will use an object diagram to explain it, this diagram is shown in Figure 7. 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 TC1. In order to prevent that situation is necessary to implement a security solution. To facilitate the implementation of the solution, two security patterns can be used: Role-based access control and Authentication. However, this security solution will still have a high abstraction level due to the fact that it is defined in the SO component. Hence, a low level implementation of the security solution should be approached in the BDAP level, in this case, the TC1 can affect the different services provided by the BDAP, that

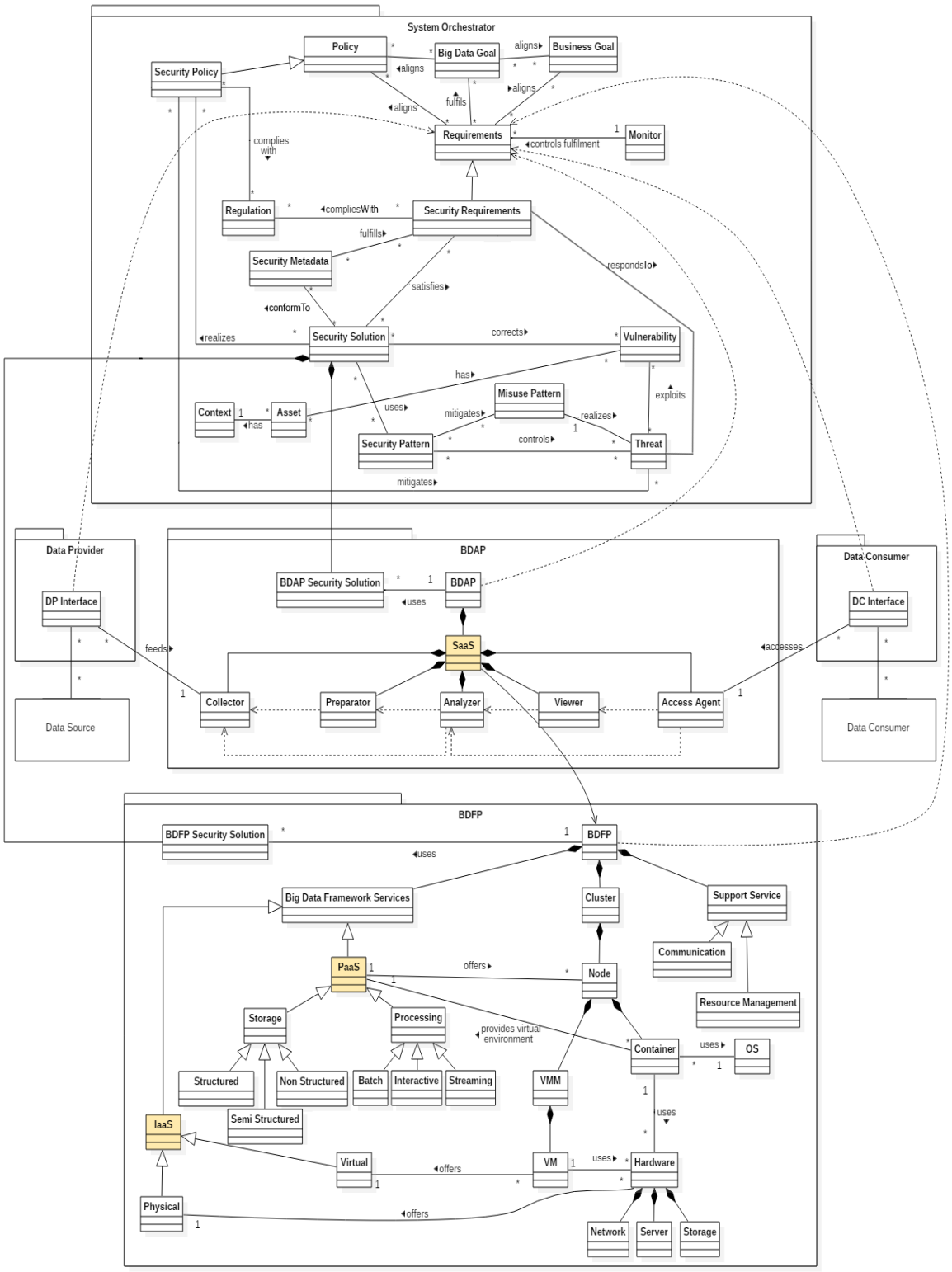

Figure 6: Big Data SRA complete diagram

Table 1: Identified threats and security patterns for the different activities

ID	Activity	Threat	Security Pattern
TC1	Common to all the activities	Data modified	Authentication, Role-based access control
TC2	Common to all the activities	Data destroyed	Authentication, Role-based access control
TC3	Common to all the activities	Data illegally read	Encryption, Role-based access control, Authentication
TC4	Common to all the activities	Unapproved change in activity function	Logger and Auditor, Controlled access session,Role-based access control, Authentication
TCo1	Collection	Malicious data source	Authentication
TP1	Preparation	Malicious filter	Logger and Auditor, Controlled access session, Role-based access control, Authentication
TA1	Analysis	Infer PII* from anonymized data	Encryption, Logger and Auditor, Multilevel security, Role-based access control, Authentication
TA2	Analysis	Malicious analysis algorithms	Logger and Auditor, Controlled access session, Role-based access control, Authentication
TV1	Visualization	PII* exposed due to high graphic granularity	Multilevel security, Authentication, Role-based access control
TAc1	Access	Several malicious access	Authentication, Role-based access control

*PII – Personal Identifiable Information

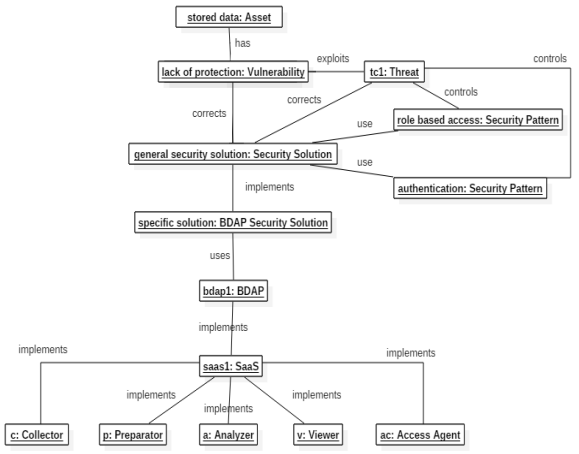

Figure 7: Using security patterns to address a specific threat

is the reason why the security solution should be implemented there and not in another component.

Furthermore, we will describe how to create an instance of the two different security patterns to secure the Collector subcomponent (Authentication and Role-based Access Control security patterns) by creating a partial example. In this example, we will focus on a Big Data system whose objective is to process tweets from the Twitter platform to analyse the general sentiment about a product. Figure 8 shows the object diagram for this example. The main component is what we want to protect, in this case: the tweets that have been obtained to be analysed.

The Authentication pattern allows us to verify the identity of the user by using a proof of identity and an authenticator. 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 system, the data scientist, the end user, and the data owner. As we explained before, this example is focused on the Collector phase, so the defined rights of the roles must consider this situation; for example, in this phase the end user should not have any rights over the data. Hence, the Figure 8 shows the different functions that the user can perform over the data according to their rights.

4 COMPARISON WITH OTHER PROPOSALS

There are not many reference architectures for Big Data systems; if we focus our architecture goal in security, there are even fewer. However, different authors and organizations have proposed different reference architectures for Big Data. In this section, we describe some of the most relevant proposals.

Demchenko et al. [11] propose a Big Data Framework Architecture that establishes the data lifecycle in 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 [28] the authors propose a complete architecture in terms of the relations between the different components; however, we found a lack of consideration given to security and privacy aspects. Klein et al. propose in [18] a specific reference architecture for Big Data in the national security domain. 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

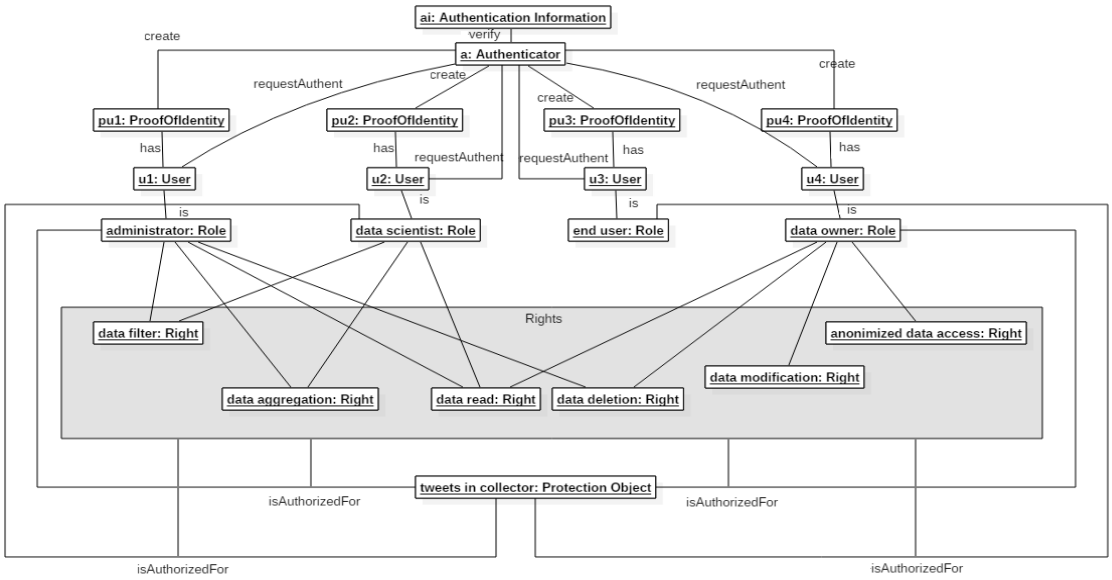

Figure 8: Application of Authentication and Role-based access control patterns

Table 2: Comparison between RAs

RA Proposal	Requirements concern	Security concern	Connection between components	Abstraction level
NIST	Medium	High	Low	High
Demchenko	Medium	Low	Medium	Medium
Klein	Low	Medium	Medium	Low
Pääkkönen and Pakkala	Medium	Low	High	Medium
SRA Proposal	High	High	High	Medium

case, requirements, and specifically the ones related to security, are the main part of the SO component.

Sqrrl [34] and BlueTalon [4] propose a Big Data model focused on data-centric security. Their purpose is to embed security information within the data itself. In the case of Sqrrl, they made emphasis in the access control in each field of data, and to do that they use a layered architecture built around the value or sensitivity of the data. On the other hand, BlueTalon includes in their proposal the concept of data lakes, a storage repository that holds a huge amount of raw data until it is needed. There are other proposals made by the main IT companies like Oracle [5], NTT data [10], IBM [9], Microsoft [24] or SAP [31]. Table II summarizes these RA and compares them with our SRA proposal. The criteria were selected based on a previous systematic mapping study that we carried out about security Big Data concerns [25]. As a side effect of this work, we detected some characteristics that usually are not considered in the different proposals and could be important to define a SRA.

Unlike the other proposals, our SRA has the requirements as the main factor to consider to properly implement a Big Data ecosystem, more specifically the security requirements that should be approached in this phase. Moreover, we have found in some proposals a lack of connection between the different components of the architecture, our SRA clearly specifies those relationships. Finally, our proposal has a medium abstraction level, due to the fact that we do not consider specific technology solutions or applications.

Although there are some SRAs for Cloud environments and some of their contributions could be useful to a Big Data environment, there are still some differences that are remarkable enough to create a SRA for Big Data. For example, there are some cases where the Big Data environment is supported by a Cloud infrastructure, in that case, the Big Data RAs must consider that possibility. In general, Cloud RAs are focused on the infrastructure, while a Big Data RA must contemplate also the services associated with the data analysis.

5 CONCLUSION AND FUTURE WORK

A more precise Reference Architecture (RA) is a better framework to guide the use of security mechanisms to provide a high level of security. Our Security Reference Architecture (SRA) subsumes the published RAs, including the proposals made by NIST, Oracle, NTT, and different researchers.

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 relationship between the different components and subcomponents is precisely defined. Also, thanks to this kind of diagram it is possible to apply different security patterns, which are usually described as UML models. Security patterns address recurrent security problems, we have defined some of the security patterns that can be implemented to protect the system against threats. Our SRA emphasizes the idea of a Big Data ecosystem by implementing the system using a Cloud Computing environment.

We have also listed some of the threats that can be found in a Big Data ecosystem; however, a deeper understanding of the different threats that can affect these systems it is needed. We will address this problem by creating different use cases and scenarios to identify those threats as in the method of [14].

Once we have the threats identified, we will find, adapt or create security patterns that can solve those problems. We consider these topics as the next steps to complete our SRA. Furthermore, it is important to perform an analysis of the different stakeholders that interact with the Big Data use cases.

ACKNOWLEDGMENTS

This work was funded by the SEQUOIA project (Ministerio de Economía y Competitividad and the Fondo Europeo de Desarrollo Regional FEDER, TIN2015-63502-C3-1-R).

REFERENCES

[1] Jacky Akoka, Isabelle Comyn-Wattiau, and Nabil Laoufi. 2017. Research on Big Data – A systematic mapping study. *SI: New modeling in Big Data* 54, Part 2 (Nov. 2017), 105–115. https://doi.org/10.1016/j.csi.2017.01.004

[2] Paris Avgeriou. 2003. Describing, Instantiating and Evaluating a Reference Architecture: A Case Study. *Default journal* (2003).

[3] E. Bertino. 2015. Big Data - Security and Privacy. In *2015 IEEE International Congress on Big Data*. 757–761. https://doi.org/10.1109/BigDataCongress.2015.126

[4] BlueTalon. 2016. BlueTalon Data-Centric Security Platform: Bringing Order to Data Security Chaos. (2016). http://bluetalon.com/data-centric_security/

[5] Doug Cackett. 2013. Information Management And Big Data A Reference Architecture. *Oracle, February* (2013).

[6] Min Chen, Shiwen Mao, and Yunhao Liu. 2014. Big data: A survey. *Mobile Networks and Applications* 19, 2 (2014), 171–209.

[7] Big Data Working Group Cloud Security Alliance (CSA). 2013. Expanded Top Ten Big Data Security and Privacy. (April 2013). https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Expanded_Top_Ten_Big_Data_Security_and_Privacy_Challenges.pdf

[8] Jason C. Cohen and Subrata Acharya. 2014. Towards a trusted HDFS storage platform: Mitigating threats to Hadoop infrastructures using hardware-accelerated encryption with TPM-rooted key protection. *Journal of Information Security and Applications* 19, 3 (2014), 224 – 244. https://doi.org/10.1016/j.jisa.2014.03.003

[9] IBM Corporation. 2014. IBM Big Data & Analytics RA. (2014).

[10] NTT DATA. 2015. NTT DATA BigData Reference Architecture. (2015). http://www.nttdata.com/global/en/shared/pdf/bigdata_reference_architecture.pdf

[11] Yuri Demchenko, Cees De Laat, and Peter Membrey. 2014. Defining architecture components of the Big Data Ecosystem. In *Collaboration Technologies and Systems (CTS), 2014 International Conference on*. IEEE, 104–112.

[12] Eduardo B. Fernandez. 2013. *Security patterns in practice: designing secure architectures using software patterns*. John Wiley & Sons.

[13] Eduardo B. Fernandez, Raul Monge, and Keiko Hashizume. 2016. Building a security reference architecture for cloud systems. *Requirements Engineering* 21, 2 (June 2016), 225–249. https://doi.org/10.1007/s00766-014-0218-7

[14] Eduardo B. Fernandez, Nobukazu Yoshioka, and Hironori Washizaki. 2009. Modeling misuse patterns. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*. IEEE, 566–571.

[15] Eduardo B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Madiha H. Syed. 2016. Modeling and Security in Cloud Ecosystems. *Future Internet* 8, 2 (April 2016), 13. https://doi.org/10.3390/fi8020013

[16] ISO/IEC. 2018. ISO/IEC CD 20547-3 - Information technology – Big data reference architecture – Part 3: Reference architecture. (2018). https://www.iso.org/standard/71277.html?browse=tc

[17] M. Kaushik and A. Jain. 2014. Challenges to big data security and privacy. *International Journal of Computer Science and Information Technologies (IJCSIT)* 5, 3 (2014), 3042–3043.

[18] John Klein, Ross Buglak, David Blockow, Troy Wuttke, and Brenton Cooper. 2016. A reference architecture for big data systems in the national security domain. In *Proceedings of the 2nd International Workshop on BIG Data Software Engineering*. ACM, Austin, Texas, 51–57.

[19] Srdjan Krco, Boris Pokric, and Francois Carrez. 2014. Designing IoT architecture (s): A European perspective. In *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 79–84.

[20] Guillermo Lafuente. 2015. The big data security challenge. *Network Security* 2015, 1 (Jan. 2015), 12–14. https://doi.org/10.1016/S1353-4858(15)70009-7

[21] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, and Dawn Leaf. 2011. NIST cloud computing reference architecture. *NIST special publication* 500, 2011 (2011), 292.

[22] V. Mayer-Schönberger and K. Cukier. 2013. *Big Data: A Revolution that Will Transform how We Live, Work, and Think*. Houghton Mifflin Harcourt. https://books.google.es/books?id=uy4lh-WEhhIC

[23] Nenad Medvidovic and Richard N. Taylor. 2010. Software architecture: foundations, theory, and practice. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*. ACM, 471–472.

[24] Microsoft. 2014. Microsoft Big Data Solution Brief. (2014). http://download.microsoft.com/download/f/a/1/fa126d6d-841b-4565-bb26-d2add4a28f24/microsoft_big_data_solution_brief.pdf

[25] Julio Moreno, Manuel A. Serrano, and Eduardo Fernández-Medina. 2016. Main Issues in Big Data Security. *Future Internet* 8, 3 (2016), 44.

[26] NIST NBD-WG. 2017. NIST Big Data Reference Architecture. (2017). https://bigdatawg.nist.gov/_uploadfiles/M0639_v1_9796711131.docx

[27] NIST NBD-WG. 2017. NIST Big Data Security and Privacy. (2017). https://bigdatawg.nist.gov/_uploadfiles/M0638_v1_4829021654.docx

[28] Pekka Pääkkönen and Daniel Pakkala. 2015. Reference architecture and classification of technologies, products and services for big data systems. *Big Data Research* 2, 4 (2015), 166–186.

[29] James Rumbaugh, Ivar Jacobson, and Grady Booch. 2004. *Unified modeling language reference manual, the*. Pearson Higher Education.

[30] S. Sagiroglu and D. Sinanc. 2013. Big data: A review. *Collaboration Technologies and Systems (CTS), 2013 International Conference on* (May 2013), 42–47. https://doi.org/10.1109/CTS.2013.6567202

[31] SAP. 2016. CIO Guide to Using the SAP HANA® Platform for Big Data. (Feb. 2016).

[32] B. Saraladevi, N. Pazhaniraja, P. Victer Paul, MS Saleem Basha, and P. Dhavachelvan. 2015. Big Data and Hadoop-A study in security perspective. *Procedia computer science* 50 (2015), 596–601.

[33] Priya P. Sharma and Chandrakant P. Navdeti. 2014. Securing big data hadoop: a review of security issues, threats and solution. *Int. J. Comput. Sci. Inf. Technol* 5 (2014).

[34] SQRRL. 2014. Big Data and Data Centric Security. (2014). http://sqrrl.com/media/Data-Centric-Security-WP-final-.pdf

[35] Bhavani Thuraisingham. 2015. Big data security and privacy. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*. ACM, 279–280.

[36] Hua Wang, Xiaohong Jiang, and Georgios Kambourakis. 2015. Special issue on Security, Privacy and Trust in network-based Big Data. *Information Sciences: an International Journal* 318, C (2015), 48–50.

[37] Jiaqi Zhao, Lizhe Wang, Jie Tao, Jinjun Chen, Weiye Sun, Rajiv Ranjan, Joanna Kołodziej, Achim Streit, and Dimitrios Georgakopoulos. 2014. A security framework in G-Hadoop for big data computing across distributed Cloud data centres. *J. Comput. System Sci.* 80, 5 (2014), 994 – 1007. https://doi.org/10.1016/j.jcss.2014.02.006 Special Issue on Dependable and Secure Computing.