



## Data Life Cycle: Towards a Reference Architecture

Mohammed EL Arass<sup>1</sup>, Khadija Ouazzani-Touhami<sup>1,2</sup>, Nissrine Souissi<sup>1,3</sup>

<sup>1</sup>Mohammed V University in Rabat, EMI. Siweb team, Morocco, mohammed.elarass@gmail.com

<sup>2</sup>Ecole Nationale Supérieure des Mines de Rabat, Morocco, ouazzani@enim.ac.ma

<sup>3</sup>Ecole Nationale Supérieure des Mines de Rabat, Morocco, souissi@enim.ac.ma

### ABSTRACT

Data management becomes highly complex with the emergence of Big Data era. Different organizations lean to produce high quality frameworks to manage data throughout their lifecycle like the developed architecture for Big Data named NIST Big Data Reference Architecture (NBD-RA). This paper aims to extend NBD-RA by adding phases to its main component, Big Data Application Provider, in order to fit with Big Data requirements. Also, the enhanced version enriches the NIST architecture and could be an open reference architecture allowing companies that want to create value from their collected data “Big” and manage it in order to transform them into “Smart” Data. To achieve this purpose, we have followed a methodology that aims to study first the foundation of NBD-RA then identify and analyze the most relevant data lifecycles. Then we define the phases that enrich the NIST architecture. We validated the proposed architecture through a case study of a company that wants to manage the huge amount of information and events produced by all the IT infrastructure including designing, implementing and testing a security information and events system POC (Proof Of Concept) made up of a Big Data platform and open source security tools.

**Key words:** Big Data, Data lifecycle, Data Management, ELK, Process Cartography, NIST Big Data Architecture, Smart Data.

### 1. INTRODUCTION

Reference architectures generally serve as a basis for architecture proposals[1].

NIST Big Data Reference Architecture was the first attempt to design open reference architecture for Big Data [2]. In this paper, we opted to rely on Data LifeCycles (DLC) identified in [3], to extend the NIST architecture. The constitute Big data lifecycle phases presented in[3], [4]seems very complex since each phase is considered as one or more complex system, with operational and independent processes. The main objective of this paper is to propose an extended architecture to fit withthe requirements of Big Data and Smart Data. The proposed architecturewill solve the limitations on the huge volume of digital information that must be efficiently exploited in spite of the requirements [5]–[7].

To validate our proposed architecture, we used a case study of a company that we will designate by “Company” for confidentiality reasons. This company wants to manage the

huge amount of information and events produced by all its IT devices. For this purpose, we have designed, implemented and tested a security information and events management system POC to manage logs and events and other data from Big Data. Indeed, the “Company” existent systems can no longer manage data from Big Data because new mechanisms and functions are required to identify, process, and analyze information flows and security events[8]. These mechanisms are essential managing security incidents for more devices with significantly increased amounts of information and speed of information flows [9].

In the second section, we present the NIST Big Data Reference Architecture that constitutes a foundation for our proposed architecture. In the third section, we analyze the relevant data lifecycles through the literature and present Smart DLC. In the fourth section, we present the enhanced version of the Big Data Application Provider from NBD-RA. In the fifth section, we present an implementation of security information and events system POC to validate the proposed architecture. In the sixth and last section, we end with a conclusion and perspectives.

### 2. NIST BIG DATA REFERENCE ARCHITECTURE

In this section, we present the NIST Big Data Reference Architecture (NBD-RA) [2].

The goal of the NBD-PWG Reference Architecture as mentioned in [2]is to develop an open reference architecture for Big Data.

The NBD-RA as shown in figure 1 presents, five logical functional components connected by interoperability interfaces (i.e., services) under Big Data system. Two fabrics envelop the components, representing the interwoven nature of management and security and privacy with all five of the components.

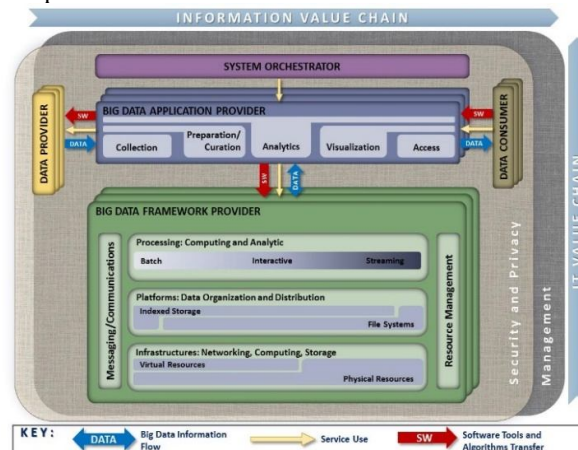


Figure 1: NBD-RA [1]

The NBD-RA in [2] is organized around five major roles and multiple sub-roles aligned along two axes representing the two Big Data value chains: the Information Value (horizontal axis) and the Information Technology (IT; vertical axis).

The five main NBD-RA roles, shown in figure 1, represent different technical roles that exist in every Big Data system. These roles are the following:

- **System Orchestrator:** provides all requirements must fulfil by the system including policy, governance, architecture, resources, business requirements, etc. Furthermore, monitoring or auditing activities to ensure that the system satisfies these requirements.
- **Data Provider:** afford data to internal use or to external services.
- **Big Data Application Provider:** executes the manipulations of the data lifecycle to meet requirements established by the System Orchestrator. Since the data and data processing is parallelized across resources the methods and techniques should be adapted onwards.
- **Big Data Framework Provider:** provide general resources or services to be used by the Big Data Application Provider in the creation of the specific application. There is a considerable number of new components from which the Big Data Application Provider can choose in using these resources and the network to build a specific system. This is the role that has seen the most significant changes because of Big Data. The Big Data Framework Provider consists of one or more instances of the three subcomponents: infrastructure frameworks, data platforms, and processing frameworks.
- **Data Consumer:** receives the value output of the Big Data system.
- The two fabric roles shown in figure 1 covering the five main roles are:
  - **Management:** Big Data demand a versatile system and software management platform for provisioning software and package configuration and management, along with resource and performance monitoring and management.
  - **Security and Privacy:** The Security and Privacy Fabric interacts with the System Orchestrator for policy, requirements, and auditing and also with both the Big Data Application Provider and the Big Data Framework Provider for development, deployment, and operation.

These two fabrics provide services and functionalities to the five main roles in the areas specific to Big Data and are crucial to any Big Data solution.

The DATA arrows in figure 1 show the flow of data between the system's main roles. Data flows between the roles either physically (i.e., by value) or by providing its location and the means to access it (i.e., by reference). The SW arrows show the transfer of software tools for the processing of Big Data. The Service Use arrows represent software programmable interfaces. While the main focus of the NBD-RA is to represent the run-time environment, all three types of

communications or transactions can occur in the configuration phase as well. Manual agreements (i.e., service-level agreements) and human interactions that may exist throughout the system are not shown in the NBD-RA. Within a given Big Data Architecture implementation there may be multiple instances of elements performing the Data Provider, Data Consumer, Big Data Framework Provider, and Big Data Application Provider roles. Thus in a given Big Data implementation, there may be multiple Big Data applications that use different frameworks to meet requirements. For example, an application may focus on gathering and analytics of streaming data and would use a framework based on components suitable for that purpose, while another application may perform data warehouse style batch analytics, which would leverage a different framework.

### 3. BIG DATA LIFE CYCLE

In this section, we seek in literature for other phases to complete those selected from NBD-RA. To do this, we are going to use the results of the DLCs literature review performed in [3].

In [3], 17 lifecycles of the literature were studied and separated into two types: DLCs proposed by renowned companies and work directly on the problem of data management in the Big Data context; and cycles from scientific research. For the second type, the most cited DLCs that are proposed in indexed and of high quality research work were chosen. In [10], a range of lifecycles related to the data of scientific research is presented. The objective is to analyze generic cycles that are not particularly interested in a specific domain.

Each cycle was analyzed individually to identify its advantages and disadvantages. At the end of this analysis, a synthesis of the DLCs studied was given. Particular attention has been taken to lifecycles that manipulate data in the Big Data context such as lifecycles Big Data and Hindawi [11], [12], to enable us to situate ourselves to these cycles. The analysis in [3] was carried out in two steps: a phase-oriented analysis and a relevant criteria-oriented analysis. Finally, 14 phases were retained: **Planning, Management, Collection, Integration, Filtering, Enrichment, Analysis, Access, Visualization, Storage, Archiving, Destruction, Security, and Quality**. The choice of these phases was the result of the synthesis of the literature review, which made it possible to identify the most relevant phases of each DLC studied.

Subsequently, these phases have been organized into three types of processes: Management, Operational and Support processes [13]. [4] proposed Smart DLC in the form of a process-oriented reference architecture as shown in figure 2. Tables 1, 2 and 3 describe for each Smart DLC process its objective, inputs, outputs, characteristics, rules, and actors.

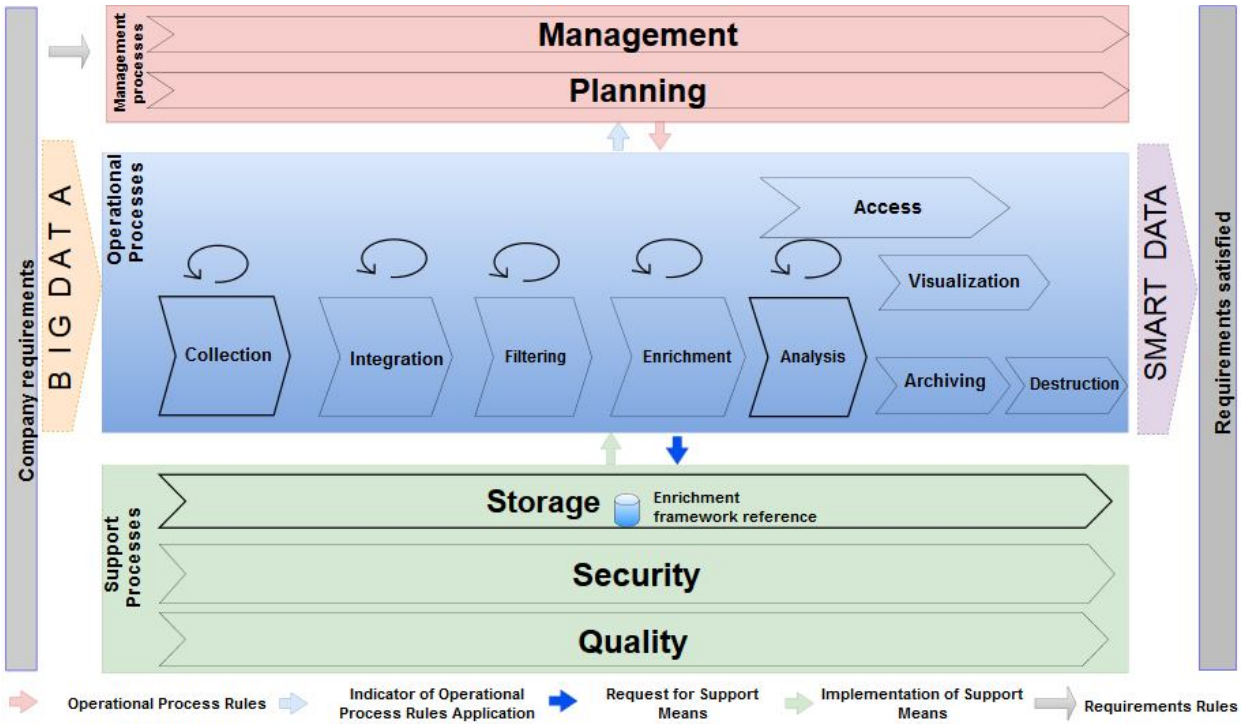


Figure 2: Smart DLC

Table 1: Smart DLC support processes details

Process	Objectives	Inputs	Outputs	Characteristics	Actors
Storage	<ul style="list-style-type: none"> <li>Store data throughout its lifecycle in order to have continuous traceability of data in each process and to know its state of progress</li> </ul>	<ul style="list-style-type: none"> <li>Raw data</li> <li>Integrated data</li> <li>Filtered data</li> <li>Information</li> <li>Knowledge</li> <li>Archived data</li> <li>Requests for storage</li> <li>Storage plan</li> <li>Storage rules</li> </ul>	<ul style="list-style-type: none"> <li>Raw data</li> <li>Integrated data</li> <li>Filtered data</li> <li>Information</li> <li>Knowledge</li> <li>Archived data</li> <li>Indicators of storage application</li> </ul>	<ul style="list-style-type: none"> <li>Transversal</li> </ul>	<ul style="list-style-type: none"> <li>Storage supervisor</li> </ul>
Quality	<ul style="list-style-type: none"> <li>Measure and control data quality throughout the cycle</li> <li>Provide for all realization processes quality means</li> </ul>	<ul style="list-style-type: none"> <li>Request for security means</li> <li>Quality plan</li> <li>Quality rules</li> </ul>	<ul style="list-style-type: none"> <li>Implementation of quality</li> <li>Indicators of quality application</li> </ul>	<ul style="list-style-type: none"> <li>Transversal</li> </ul>	<ul style="list-style-type: none"> <li>Support supervisor</li> </ul>
Security	<ul style="list-style-type: none"> <li>Measure and control data security throughout the cycle</li> <li>Provide for all realization processes security means to make data confidential.</li> </ul>	<ul style="list-style-type: none"> <li>Request for quality means</li> <li>Security plan</li> <li>Security rules</li> </ul>	<ul style="list-style-type: none"> <li>Implementation of security</li> <li>Indicators of security application</li> </ul>	<ul style="list-style-type: none"> <li>Transversal</li> </ul>	<ul style="list-style-type: none"> <li>Support supervisor</li> </ul>

Table 2: Smart DLC management processes details

Process	Objectives	Inputs	Outputs	Characteristics	Actors
Planning	<ul style="list-style-type: none"> <li>Define plans for each process</li> </ul>	<ul style="list-style-type: none"> <li>Company requirements</li> </ul>	<ul style="list-style-type: none"> <li>Plan for each process</li> <li>Data description</li> </ul>	<ul style="list-style-type: none"> <li>Transversal</li> </ul>	<ul style="list-style-type: none"> <li>Planner</li> <li>Planning team</li> </ul>
Management	<ul style="list-style-type: none"> <li>Manage all realization and support processes</li> <li>Validate all realization and support processes</li> </ul>	<ul style="list-style-type: none"> <li>Management plan</li> <li>All processes indicators</li> </ul>	<ul style="list-style-type: none"> <li>Rules for each process</li> <li>Orders for each realization process</li> </ul>	<ul style="list-style-type: none"> <li>Transversal</li> </ul>	<ul style="list-style-type: none"> <li>Smart DLC manager</li> <li>Management team</li> </ul>

**Table 3:** Smart DLC operational processes details

Process	Objectives	Inputs	Outputs	Characteristics	Actors
<b>Collection</b>	<ul style="list-style-type: none"> <li>• Collect all company data</li> <li>• Collect external data</li> <li>• Collect archived data</li> </ul>	<ul style="list-style-type: none"> <li>• Company data</li> <li>• External data</li> <li>• Collection rules</li> <li>• Collection plan</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Raw data</li> <li>• Indicators of collection rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Collection supervisor</li> </ul>
<b>Integration</b>	<ul style="list-style-type: none"> <li>• Provide a coherent pattern of data from multiple independent, distributed and heterogeneous sources</li> <li>• facilitate users accessing and querying</li> </ul>	<ul style="list-style-type: none"> <li>• Row data</li> <li>• Integration plan</li> <li>• Integration rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Integrated data</li> <li>• Indicators of integration rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Integration supervisor</li> </ul>
<b>Filtering</b>	<ul style="list-style-type: none"> <li>• Restrict the large data flow</li> </ul>	<ul style="list-style-type: none"> <li>• Integrated data</li> <li>• Filtering plan</li> <li>• Filtering rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Filtered data</li> <li>• Indicators of filtering rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Filtering supervisor</li> </ul>
<b>Enrichment</b>	<ul style="list-style-type: none"> <li>• Add information on collected data to improve their quality</li> </ul>	<ul style="list-style-type: none"> <li>• Filtered data</li> <li>• Enrichment data</li> <li>• Knowledge data</li> <li>• Enrichment plan</li> <li>• Enrichment rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Information</li> <li>• Enrichment data</li> <li>• Archived data</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Enrichment supervisor</li> </ul>
<b>Analysis</b>	<ul style="list-style-type: none"> <li>• Exploit and analyze data to draw conclusions and interpretations of decision-making</li> </ul>	<ul style="list-style-type: none"> <li>• Information</li> <li>• Implementation of anonymity</li> <li>• Analysis plan</li> <li>• Analysis rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Knowledge</li> <li>• Indicators of analysis rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Analysis supervisor</li> </ul>
<b>Access</b>	<ul style="list-style-type: none"> <li>• Provide an interface to the data consumer</li> </ul>	<ul style="list-style-type: none"> <li>• Knowledge</li> <li>• Access plan</li> <li>• Access</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Interface</li> <li>• Indicators of access rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Access supervisor</li> </ul>
<b>Visualization</b>	<ul style="list-style-type: none"> <li>• Display knowledge with a smart manner</li> </ul>	<ul style="list-style-type: none"> <li>• Knowledge</li> <li>• Visualization plan</li> <li>• Visualization rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Dashboards</li> <li>• Decisions</li> <li>• Indicators of visualization rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Visualization supervisor</li> </ul>
<b>Archiving</b>	<ul style="list-style-type: none"> <li>• Provide long-term storage of data for possible use</li> </ul>	<ul style="list-style-type: none"> <li>• Knowledge</li> <li>• Archiving plan</li> <li>• Archiving rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Archived data</li> <li>• Indicators of archiving rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Archiving supervisor</li> </ul>
<b>Destruction</b>	<ul style="list-style-type: none"> <li>• Delete the data when it is successfully used and will become useless and without added value.</li> </ul>	<ul style="list-style-type: none"> <li>• Destruction plan</li> <li>• Destruction rules</li> <li>• Security means</li> <li>• Quality means</li> </ul>	<ul style="list-style-type: none"> <li>• Indicators of destruction rules application</li> </ul>	<ul style="list-style-type: none"> <li>• Single</li> </ul>	<ul style="list-style-type: none"> <li>• Destruction supervisor</li> </ul>

#### 4. BIG DATA APPLICATION PROVIDER: ENHANCED VERSION

In this section, we enhance the Big Data Application Provider by adding new phases from the DLCs studied in the literature review.

We noted in [3] that the majority of phases are shared by most cycles, although their nomenclature is sometimes different. Phases are divided into sub-phases for some cycles; others are grouped together to form a single phase. For example, the collection phase includes the following phases: data reception, data creation, filtering, data integration, and anonymity. Some cycles introduce the visualization phase in the analysis phase; others detach it to have it as one phase.

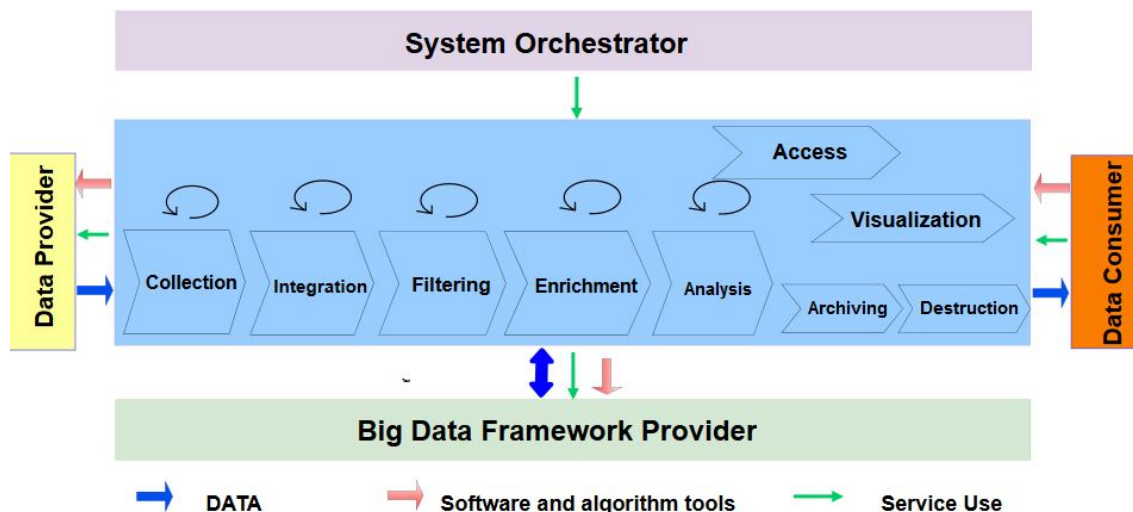
From the phases proposed in Big Data Application Provider from NBD-RA and the phases proposed in the DLCs, we note that the majority of phases are common. Table 4 summarizes common and specific phases of Big Data Application Provider and DLCs.

**Table 4:** Common and specific phases of NBD-RA and DLCs

Common phases of NBD-RA and DLCs	Specific phases of DLCs
Planning	Filtering
Management	Enrichment
Collection	Archiving
Integration	Quality
Analysis	
Visualization	
Destruction	
Access	
Storage	
Security	

To enhance the Big Data Application Provider, we started with the phases that were selected from the NBD-RA. Based on these phases, we added the other phases that were selected from the DLCs literature review. Hence, 9 phases were retained: **Collection, Integration, Filtering, Enrichment, Analysis, Visualization, Access, Archiving, and Destruction.**

Figure 3 illustrates the enhanced version of the Big Data Application Provider.



**Figure 3:** The enhanced version of Big Data Application Provide

#### 5. CASE STUDY

In this section, we present a validation of the proposed architecture through a case study of a Company that we designated “Company” for confidentiality reasons. The “Company” wants to manage the huge amount of information and events produced by all the computer park including:

- Network equipment: routers, switches...
- Servers: with Linux and Windows OS with several types of applications (domain controller, web, SGBD, messaging, IP telephony, videoconferencing, video surveillance ...)
- Machines: with only Windows OS

Our improved version of the NIST architecture will try to provide a solution to the following problematic of “Company”:

Despite the fact that “Company” has a monitoring solution of its IT park in particular:

- A SolarWinds platform that is a fairly common network monitoring tool that provides a real-time idea of a network equipment connectivity.
- A centralized supervision solution for firewall equipment.

However, these two systems do not protect this company against Cyber-type attacks. If the SolarWinds monitoring system just sends "ping" messages and wait for the answer to inquire about its connection status, the second system is more interesting in terms of log management and analysis but only for firewall equipment from the same constructor. The latter does not manage other types of firewalls and also other network and system equipment including routers, switches, servers, and machines. In addition, both systems are not designed to handle Big Data.

Our solution is to design, implement and test a Proof Of Concept (POC) of new security information and event



management system to allow the “Company” to centrally manage all security information and events of its IT park. Our validation will bring two major contributions:

- Design, implement and test a Proof Of Concept (POC) of a security event and information management system;
- Validate the enhanced version of the Big Data Application Provider of the NIST architecture proposed in this paper.

### 5.1 Motivation

Nowadays, cybersecurity incidents are observed more and more[14]. According to Microsoft, the potential cost of cyber-crime to the global community is a mind-boggling \$500 billion, and a data breach will cost the average company about \$3.8 million [15]. In addition, intrusion detection and network monitoring require intelligent and real-time data management [16]. The amount of data to manage is very large that a classical solution proves ineffective. Hence, Big Data solutions are needed in this context. Since our architecture was designed to be adapted to the Big Data context, we consider that it would be relevant to use it to provide security information and event managementsystem composed of several open source tools essential for network monitoring.

Data in IT flowsare unstructured and composed of several types (log, session, packet tcp, ip, etc.). While Big Data systems are able to process heterogeneous and large data, they also generate a significant amount of logs [17].Since the proposed architecture allows to transform Big Data to Smart Data, we found it is useful that a security information and event system implementation following the Big Data Application Provider enhanced version is a good way to validate the proposed architecture since the proposed POC allows to transform a huge amount of file logs to security alerts that are Smart Data in this case.

We believe that the proposed architecture can be considered as a basis for designing and implementing a security event and information system POC. To do this, we followed a rigorous method to validate both the proposed architecture and the designed POC.

### 5.2 Method

We followed a rigorous method for choosing tools that make up our system POC as well as for its design, implementation and functional tests as shown in figure 4.

- **Step 1:** we have identified several open source tools for each process in our architecture that meets at least one

security information and events management system functional requirement.

- **Step 2:**we have selected the tools that can integrate with each other. When we found several tools for the same function, we chose the most widespread tool that offers a great availability of documentation.
- **Step 3:** we designed the architecture of the POC tools to simplify its implementation later.
- **Step 4:** we implemented all the selected tools in a distributed Linux environment.
- **Step 5:** we have tested all the POC tools and their integration.

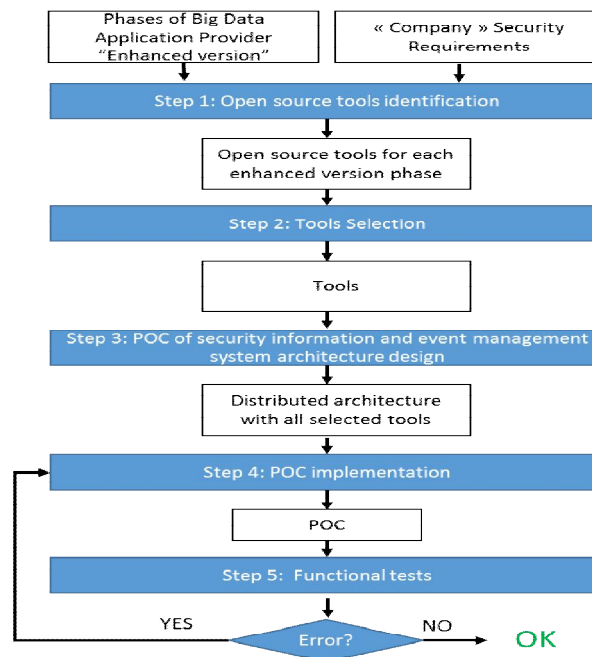


Figure 4: Method

### 5.3 Security Information and events management system construction

Often, a tool is composed of several components in which each one performs a function and thus concerns a process of the proposed architecture. For example Snort, consists of a packet decoder, a pre-processor, and a detection engine. Table 5 illustrates the correspondence matrix between the selected tools and the realization and support processes.

**Table 5:** Correspondence matrix between POC tools and enhanced version of Big Data Architecture Reference









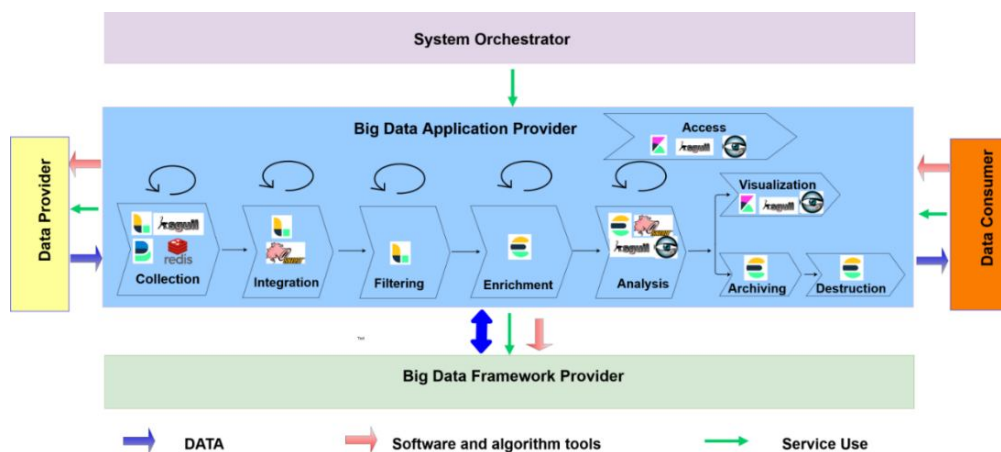
Tool	Definition	Logo	Collect	Integration	Filtering	Enrichment	Analysis	Access	Visualization	Archiving	Destruction
<b>Snort</b>	Open source intrusion detection and prevention system (NIDS & NIPS) maintained by CISCO [18].			✓			✓				
<b>Sguil</b>	Open source desktop application that offers an intuitive interface for events, session data and raw packets visualization. It was created by Network Security Analysts [19].		✓				✓	✓	✓		
<b>Bro</b>	Network analysis system. It provides a complete platform for analyzing network traffic. [20]			✓	✓	✓	✓	✓	✓		
<b>Redis</b>	Open source in memory database used to manage the data queue in memory. [21]		✓								
<b>Beats</b>	Open source agent platform installed on remote devices to send logs to Logstash.		✓								
<b>Logstash</b>	Dynamic open source pipeline, collects and integrates data simultaneously from a multitude of sources to transform them and send them to another storage system generally elasticsearch. [22]		✓	✓	✓						
<b>Elasticsearch</b>	Open source multi-entities RESTful distributed search and analysis engine. It manages a NoSql DB.						✓	✓		✓	✓
<b>Kibana</b>	Powerful and intuitive visualization tool for data found in elasticsearch.								✓	✓	

Figure 5 illustrates the location of each tool selected in the proposed architecture.

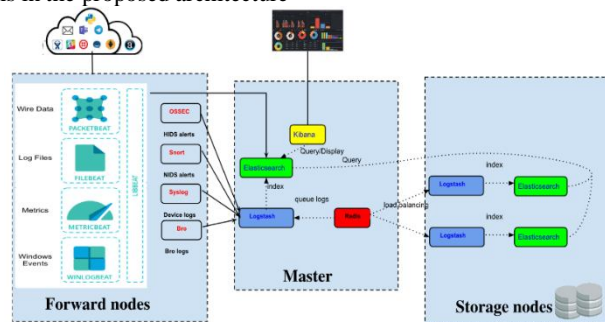


**Figure 5:** Location of open source tools in the proposed architecture

Tools that make up our security information event management system POC concern only the Big Data Application Provider enhanced version of the BDRA. For System Orchestrator, these are essentially planning processes that do not process the data but rather plan their collection, analysis, storage, etc. When we were in the tools implementation phase, we also rolled out these processes for each integrated tool.

**5.4 Implementation**

Before starting the POC implementation, we designed a distributed architecture in which our POC will be integrated. This distributed architecture shown in figure 6 will optimize its operation.



**Figure 6:** POC implementation architecture

Subsequently, we implemented all the selected tools in a distributed Linux environment with technical specifications. The hardware specifications are: a Master server with 8 CPU cores, 16 GB RAM and 1 TB disc space, Storage Node with 4 CPU cores, 8 GB RAM and 100

TB disc space, and Forward node with 2 CPU, 2 GB RAM and 500 Go disc space.

We performed functional tests of our POC to verify that all of its components work and interact with each other without errors. Figure 7 shows some alerts that were recorded by the filebeats tool installed on a device. Data shown in figure 7 represent Smart Data in our context. So, the POC implemented following the proposed architecture has allowed transforming from Big Data to Smart.

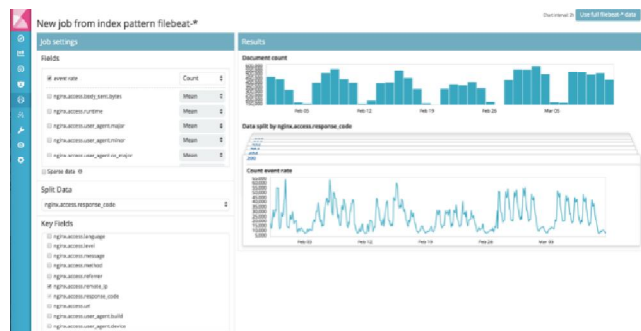


Figure 7: Visualization of alerts generated by the proposed POC

Among these alerts, our SIEM POC discovered these attacks:

- SQL injection attack
- Brute force attacks
- DDoS attacks

Also, the proposed POC identified the target and attacker machines. Following this, the company security team reacted to these alerts by unplugging the target machine and analyzing it and also by redefining the firewalls security policy. In addition, a general scan in all the “Company” machines was launched to see if any machine was infected.

## 6. CONCLUSION AND FUTURE WORKS

In this paper, we proposed an enhanced version of the BigData Application Provider to enrich and extend the NIST Big Data Reference Architecture.

The objective of the proposed architecture is to satisfy the requirements established in the System Orchestrator. The development of an ecosystem Big Data is not a trivial project because such an ecosystem usually includes the use of different technologies that interact with each other, which makes Big Data project more complicated. The new architecture supports effectively the company data management tasks and their transformations. It takes into account the existing situation and makes it possible to better anticipate the internal and external evolutions or constraints impacting the data lifecycle and, if necessary, relying on technological opportunities. Enriching the NIST architecture facilitates the company transformation and change.

We have followed all proposed Big Data Application Provider processes to design, implement and test a security information and events management system POC. This system has been used as a case study of “Company” that wants to manage the huge amount of information and events produced by all its IT park in order to transform them into smart security alerts. This validation has provided a double scientific contribution:

validate the extended version of the NIST architecture and validate the designed POC.

With this contribution, we have shown that our extended NIST architecture is the most suitable for Big Data architecture and makes raw and valueless data in Smart Data as defined in [23] especially those interested in cybersecurity issues. Companies wishing to evaluate their existing data lifecycle could use this architecture to check if it is adequate for the Big Data context. They can also, thanks to the proposed architecture, evaluate the intelligence level of their cycle. As future work, the performances of our proposal will be compared to other commercial SIEMs.

## REFERENCES

- 1 W. Chang, ‘NIST Big Data Reference Architecture for Analytics and Beyond’, in *Proceedings of the 10th International Conference on Utility and Cloud Computing*, New York, NY, USA, 2017, pp. 3–3 doi: 10.1145/3147213.3155013.
- 2 W. L. Chang, ‘NIST Big Data Interoperability Framework: Volume 6, Reference Architecture’, *Special Publication (NIST SP) - 1500-6*, Aug. 2017, Accessed: Apr. 12, 2018. [Online]. Available: <https://www.nist.gov/publications/nist-big-data-interop-erability-framework-volume-6-reference-architecture>.
- 3 M. El arass, I. Tikito, and N. Souissi, ‘Data lifecycles analysis: towards intelligent cycle’, presented at the Proceeding of The second IEEE International Conference on Intelligent Systems and Computer Vision, ISCV’2017, Fès 17-19 April, Fez, Morocco, 2017. , doi: 10.1109/ISACV.2017.8054938.
- 4 M. El arass and N. Souissi, ‘Data Lifecycle: From Big Data to SmartData’, in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, Marrakech, Oct. 2018, pp. 80–87 doi: <https://doi.org/10.1109/CIST.2018.8596547>.
- 5 D. Farge, ‘Du Big data au smart data : retour vers un marketing de l’émotion et de la confiance’, *LesEchos.fr*, 2015.
- 6 F. Meleard, ‘Smart data, l’avenir du contenu’, *Les echos.fr*, 2015.
- 7 F. Zaoui, ‘A Triaxial Model for the Digital Maturity Diagnosis’, *IJATCSE*, vol. 9, no. 1, pp. 433–439, Feb. 2020, doi: 10.30534/ijatcse/2020/60912020.
- 8 D. Ahamad, M. Akhtar, and S. A. Hameed, ‘A Review and Analysis of Big Data and MapReduce’, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1, pp. 1–3, 2019, doi: 10.30534/ijatcse/2019/01812019.
- 9 R. Zuech, T. M. Khoshgoftaar, and R. Wald, ‘Intrusion detection and Big Heterogeneous Data: a Survey’, *Journal of Big Data*, vol. 2, no. 1, Dec. 2015, doi: 10.1186/s40537-015-0013-4.
- 10 T. Wissik and M. Đurčo, ‘Research Data Workflows: From Research Data Lifecycle Models to Institutional Solutions’, in *Selected Papers from the CLARIN Annual Conference 2015, October 14–16, 2015, Wrocław, Poland*, 2016, pp. 94–107.



- 11 Y. Demchenko, C. De Laat, and P. Membrey, 'Defining architecture components of the Big Data Ecosystem', in *Collaboration Technologies and Systems (CTS), 2014 International Conference on*, 2014, pp. 104–112.
- 12 N. Khan *et al.*, 'Big data: survey, technologies, opportunities, and challenges', *The Scientific World Journal*, vol. 2014, 2014.
- 13 M. El arass, I. Tikito, and N. Souissi, 'An Audit Framework for Data Lifecycles in a Big Data context', in *2018 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, Tangier, Jun. 2018, pp. 1–5  
doi: <https://doi.org/10.1109/MoWNet.2018.8428883>.
- 14 R. K. Alqurashi, M. A. AlZain, B. Soh, M. Masud, and J. Al-Amri, 'Cyber Attacks and Impacts: A Case Study in Saudi Arabia', *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1, 2020, doi: 10.30534/ijatcse/2020/33912020.
- 15 M. John, '21 Interesting Cyber Security Statistics (2019)', 2019.  
<https://thebestvpn.com/cyber-security-statistics-2018/> (accessed Mar. 12, 2019).
- 16 Y. Keim and A. K. Mohapatra, 'Cyber threat intelligence framework using advanced malware forensics', *International Journal of Information Technology*, Feb. 2019  
doi: 10.1007/s41870-019-00280-3.
- 17 P. Wu *et al.*, 'Bigdata logs analysis based on seq2seq networks for cognitive Internet of Things', *Future Generation Computer Systems*, vol. 90, pp. 477–488, Jan. 2019, doi: 10.1016/j.future.2018.08.021.
- 18 CISCO, 'Snort website', 2019.  
<https://snort.org/documents> (accessed Mar. 08, 2019).
- 19 Network Security Analyst, 'Sguil - Open Source Network Security Monitoring', 2019.  
<http://bammv.github.io/sguil/index.html> (accessed Mar. 08, 2019).
- 20 Zeek, 'The Zeek Network Security Monitor', 2019.  
<https://www.zeek.org/> (accessed Mar. 09, 2019).
- 21 Redislabs, 'Redis', 2019. <https://redis.io/> (accessed Mar. 08, 2019).
- 22 S. J. Son and Y. Kwon, 'Performance of ELK stack and commercial system in security log analysis', in *2017 IEEE 13th Malaysia International Conference on Communications (MICC)*, Johor Bahru, Nov. 2017, pp. 187–190, doi: 10.1109/MICC.2017.8311756.
- 23 A. Lenk, L. Bonorden, A. Hellmanns, N. Roedder, and S. Jaehnichen, 'Towards a taxonomy of standards in smart data', in *Big Data (Big Data), 2015 IEEE International Conference on*, 2015, pp. 1749–1754.  
<https://doi.org/10.1109/BigData.2015.7363946>