Security by Design for Big Data Frameworks over Cloud Computing

Feras M. Awaysheh*† *Member, IEEE*, Mohammad N. Aladwan†, Mamoun Alazab‡ *Senior Member, IEEE*, Sadi Alawadi§, José C. Cabaleiro†, and Tomás F. Pena† *Senior Member, IEEE*
*Data Systems Group Institute of Computer Science, University of Tartu, Estonia
†Centro Singular de Investigación en Tecnoloxías Intelixentes, University of Santiago de Compostela, Spain
‡College of Engineering, IT and Environment at University of Charles Darwin, Australia
§ Department of Information Technology, Division of Scientific Computing, Uppsala University, Sweden

Abstract

Cloud deployment architectures have become a preferable computation model of Big Data (BD) operations. Their scalability, flexibility, and cost-effectiveness motivated this trend. In a such deployment model, the data is no longer physically maintained under the users direct control, which raises new security concerns. In this context, BD security plays a decisive role in the widespread adoption of cloud architectures. However, it is challenging to develop a comprehensive security plan unless it is based on a preliminary analysis that ensures a realistic secure assembly and addresses domain-specific vulnerabilities. This study presents a novel Security-by-Design framework for BD frameworks deployment over cloud computing (BigCloud). In particular, it relies on a systematic security analysis methodology and a completely automated security assessment framework. Our framework enables the mapping of BigCloud security domain knowledge to the best practices in the design phase. We validated the proposed framework by implementing an Apache Hadoop stack use case. The study findings demonstrate its effectiveness in improving awareness of security aspects and reducing security design time. It also evaluates the strengths and limitations of the proposed framework, from which it highlights the main existing and open challenges in the BigCloud related area.

Index Terms

Security-by-Design, Big Data, Cloud-computing security, Data protection, Reference Architecture, Security Components Diagram, Security Analysis Pattern.

Managerial relevance statement

This paper aims to provide the first security-by-design framework of big data operations over cloud computing (BigCloud). The proposed framework provides software engineering, security engineering, and system engineering functionalities to deploy a secure BigCloud solution with an integrated four knowledge domains to guide the security in the design phase. A reference architecture, security component diagram, security analysis pattern, and security features selection are the main framework components. The proposed approach introduces a new workflow for engineering big data operations where security modeling and engineering are fully integrated into the targeted environment's software engineering processes. A series of quantitative methods has also been presented for the practical use-case of Apache Hadoop 3.0, the big data platform's de-facto. An example of an application to a real BigCloud project by implementing security-by-design system architecture that considers security granularity is also provided. The proposed approach is expected applicable to a wide set of big data practitioners and guides security-by-design as a mainstream development approach. Overall, the paper's findings pave the way for a wide range of revolutionary and state-of-the-art enhancements and future trends within the BigCloud deployment models.

I. Introduction

In this new digital era, many companies use cloud technology to store, process, and analyze petabytes of structured and non-structured data relating to their business and customers [1] [2]. Advancements in cloud computing technology have shaped the modern application delivery model [3]. The advantages of adopting cloud computing are inarguable due to its great potential to provide affordable and straightforward access to substantial computing power. Big data (BD) frameworks over cloud computing (BigCloud) promote this trend —with the potential of higher substantial scalability and elasticity than traditional models [5]. As the outsourced data may contain confidential information, such as financial records, proprietary research data, healthcare data, or government information, data security becomes even more critical.

Data security and privacy are among the clouds leading next-decade research directions [4]. This direction aims to maintain the efficiency of sustainable BD operations over cloud systems. The main security challenge pertains to the clients trust in data transfer in and out of the cloud environment —as well as the storing and processing of critical data within an off-premise data center. Some fundamental characteristics of the cloud (such as multitenancy and virtualization) ensure better utilization of resources but make it challenging to deliver secure computation. Other security threats associated with this adoption include

Feras M. Awaysheh is with the Centro Singular de Investigacin en Tecnoloxas Intelixentes (CiTIUS), University of Santiago de Compostela, 15782 Santiago de Compostela, Spain, and Data Systems Group, University of Tartu, Narva mnt 18, Tartu 51008, Estonia (* corresponding author email: feras.awaysheh@ut.ee)

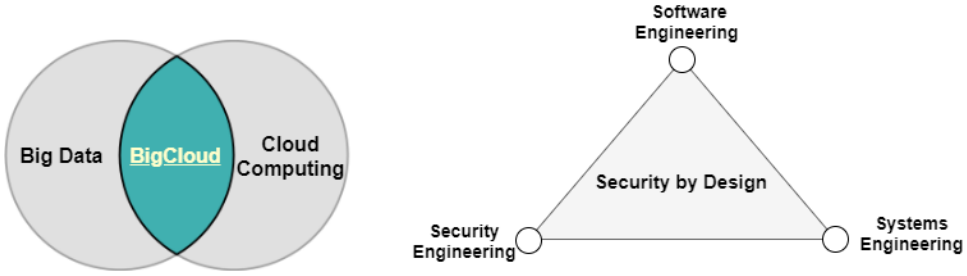

Fig. 1. The paper domain

privacy, integrity, confidentiality, and the availability of stored data, which are magnified by the properties of BD (i.e., volume, velocity, and variety) systems [1].

The security of BigCloud is a growing concern, first, due to the increased exposure of the data to potential attackers; second, due to the broad attack surface of a cloud environment; third, due to the interaction of several security frameworks within different layers of protection, especially when it comes to the development of applications in this environment. Beyond these technical issues, we argue that the Security-by-Design principle for BigCloud systems is poorly understood and rarely practiced. A recently published study by Sequeiros, Joo BF, et al. highlights security-by-designas one of the main existing challenges and open issues in the cloud environment [13]. This study reports on the IaaS cloud, where both the user and the service provider share the responsibility for their BD stages, i.e., data-at-rest, data-in-transit, and data-in-process security. In this context, security-by-designis a holistic and anticipatory approach that ensures meeting security requirements from the system's early conception. It includes methodical and systematic security procedures to ensure that these requirements are complete, consistent, and easy to measure and evaluate in later system development stages.

Our research domain is distributed over two main computer science fields; big data and cloud computing, referring to BigCloud. Also, it employes software engineering, security engineering, and system engineering for the implementation of BigCloud solution as represented in Figure 1. At its core, this approach has the description and proposal of workflows for engineering and development of applications and systems where security modeling and engineering are fully integrated into the software engineering processes. For this reason, we propose a framework (in Figure 2) that supports the secure deployment of BD frameworks during all the development phases of a BigCloud solution. These stages start from the specification of its high-level reference architecture, up to identifying the deployment configuration that best fits its requirements in the component diagram and analysis pattern. In this regard, the framework illustrates security components that ensure the representation of the problem domain (i.e., security, reliability, and privacy). This representation serves as a mechanism to transfer knowledge of security modeling and software engineering tools to the BigCloud domain. It also serves as a knowledge capture, which contains domain knowledge (e.g., using cases and scenarios) and the solution knowledge (e.g., mapping current technologies). On the same note, this study took from the security by designing a scheme to eliminate the chasm between software engineering and security engineering to enable secure BigCloud design.

A. Contribution

In this article, a BigCloud security-by-design framework is proposed, whose main components ate a reference architecture, a security component diagram, and a security analysis pattern. To the best of our knowledge, other studies do not seem to approach these topics with an in-depth investigation as this study. The proposed methodology supports BD developer in building and deploying a secure cloud application while taking into account the potential security issues from the beginning of the development process to reduce the risks associated with existing vulnerabilities and threats. This work, hence, contributes to the adoption of secure IaaS cloud models as scalable BD deployment architectures by taking the following measures:

- We define a security-by-design development process that foregrounding the security components and essential qualities of a BigCloud framework;
- We provide a formulation of BigCloud deployment architecture that explicitly takes into account the Apache Hadoop 3.0 security ecosystem and that allows utilizing the BD state-of-the-art technologies that express security best practices;
- We map the current security technologies onto concerns and defining domain concepts based on the grouping of relevant of these concerns into a security-by-design model that facilitates the installation and configuration of secure BigCloud systems;
- We demonstrate how to utilize the security-by-design process as an effective medium to create and evaluate secure BigCloud systems with a sufficient trust level.

B. Organization

The structure of this paper is as follows: Section II presents a general review of the papers scope and background alongside the methodology and related work. Section III outlines the reference architecture as part of the cloud security management.

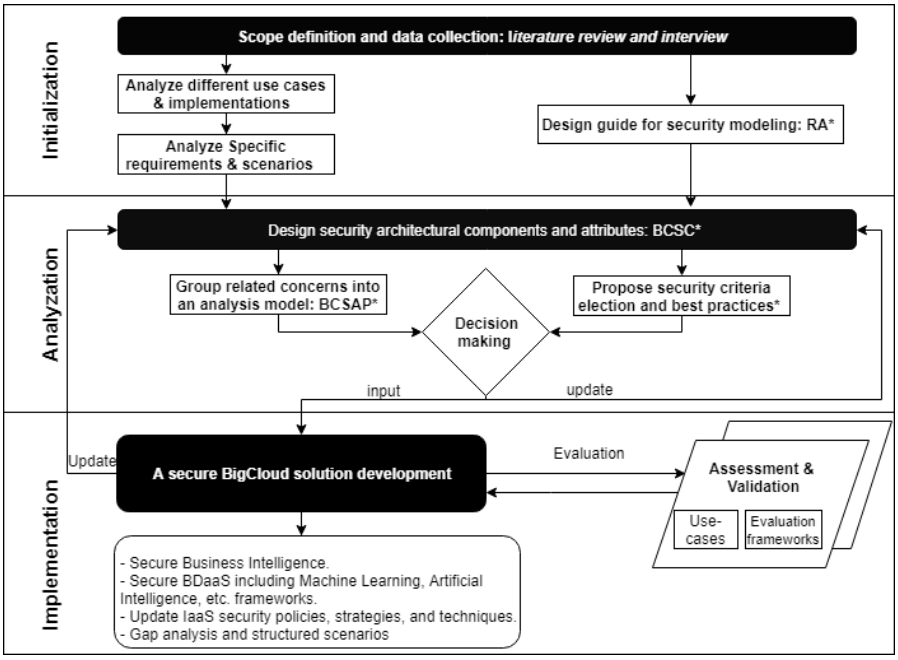

Fig. 2. A graphical representation of systematic research methodology for modeling and implementing a security solution for BigCloud adoption. The abbreviations and (*) signs represent the paper scope, and they are further described in Table 1.

Section IV provides a component diagram of BigCloud security processes and attributes. Further, Section V discusses security-related considerations, including service-delivery, security, and data service security. Section VI provide an analysis pattern and selection approach by mapping the knowledge domain to the solution domain. Finally, we draft our conclusions in Section VII by proposing recommendations, open challenges, and future work.

II. Background

Big data platforms use an architectural pattern that guides data-intensive solutions to create, organize, and reuse their computing components. Meanwhile, cloud computing is a set of enabling technologies that provide broader services and more flexible solutions for enterprises to deploy their frameworks. This section discusses the relationship between BD platforms and cloud computing service providers. Further, it introduces the methodology and motivation behind this work. Finally, it discusses related work.

A. Big Data over Cloud computing (BigCloud)

Historically, BD deployment architectures have been designed as shared-nothing architecture with enough capacity to meet peak demands. However, this architecture could result in the system underutilizing its capacity that organizations must still pay for. On the other hand, once the systems capacity has been reached, a significant investment in time, resources, and money to expand it is expected. Modern industry and academia require utility services, through which they can scale capacity vertically and horizontally on demand and pay only for what they use.

The advent of cloud-based clusters promotes implementing a cloud solution that supports BD operations. This approach grants a practical solution that not only tackles this challenge but also enhances the systems scalability, reduces maintenance cost, and increases the efficiency of resource management. Over the years, cloud service providers have offered a wide range of BD-supporting services spanning from storage to processing and analyzing vast amounts of datasets. Examples include public-service providers (e.g., Amazon EMR [6], Microsoft Azure HDInsight [7], and Google Cloud Dataproc [8]) and private vendors (e.g., Cloud-era [9], Hortonworks [10] , and MapR [11]).

B. Motivation and Methodology

This work aims at facilitating the realization of secure BD systems in the IaaS cloud model. When a BigCloud system is realized, important security considerations arise. These security factors include the architectural design of the system and the underlying security technologies and policies/services. IaaS continues to be the fastest growing model [12] and the most desired by BD implementers. This study analyzes security services used in Iaas cloud environments and describes BD security items and relationships amongst them. It discusses security systems oriented to BigCloud design in order to present their glossary

TABLE I
REPRESENTING A SUMMARIZATION OF THE RESEARCH SCOPE MAPPED TO THE SECTION ARCHITECTURE.

Abbreviation	Knowledge Domain	Description	Section
BCRA	Preparation	Reference Architecture with different service layers and security services that describes the security issues addressed by the study.	Section numberIII
BCSC	Initialization	BigCloud Security Component that reviews the primary components as of security design pattern with security attributes and constraints affect the security problem.	Section number IV
BCSAP	Examination	BigCloud Security Analysis Pattern that reviews the bases for building a secure ecosystem during the analysis phase and describes the basic structure and risks to be considered while applying the solution using a UML diagram.	Section number VI-A
SCE	Selection	Security Criteria Election that reviews the various criteria that influence security in a BigCloud context, and describes different ways a security pattern may be implemented and deployed.	Section number VI-B

and landscape techniques and to define research gaps and best practices. Figure 2 describes the methodology employed in this study to deliver a generic BigCloud security reference model.

In detail, this study contributes to the BigCloud security deployment body of knowledge by, first , extensively examining the building blocks of the cloud security stack for supporting BD science. In addition, it classifies the different layers of security based on their supported service models into a reference architecture. Second, it examines the vulnerabilities associated with BigCloud adoption by providing the security components of a secure design pattern and its attributes. Third, it provides various insights into BigCloud security specifications by refining the cloud context-pattern into a novel security analysis pattern. This pattern maps the current technologies to the solution domain by extending the CIA (Confidentiality, Integrity, and Availability) triad. Next, they study analyzes and classifies the state-of-the-art security frameworks available today mostly as open-source for a detailed criteria election. Finally, it highlights some open challenges and recommendations for both service providers and customersfor a comprehensive discussion towards achieving the vision of providing a secure BigCloud service. To facilitate using the systematic research methodology, we summarize the proposed models and patterns in Table I, which consists of the knowledge domain, its model description, and its designated section within this study.

C. Related Work

While cloud security is a well-established domain, no work seems to focus on integrating security aspects within the software development of cloud applications [13]. Table II presents a comparison of five different models of the state-of-the-art security-by-design studies. In particular, we compare each study's advantages and limitations in delivering security-by-design in a cloud-enabled environment to the proposed framework in our research. This section also discusses security modeling and solutions devoted to risk analysis and security assessment tasks within the BigCloud environment.

The literature has suggested different Cloud-driven meta-models to support cloud application management. In [14] , Hamdaqa et al. propose a service-oriented architecture that captures design elements, configuration rules, and a semantic interpretation of cloud applications in a meta-model. Their work meets the goals of this paper in that both works aim to standardize cloud-modeling language by drafting reference models. However, the proposed model of this study formalizes the cloud securitys vocabulary and semantics, which assists in developing a secure BD service-oriented model and suitable cloud runtime security support. In the context of the cloud-security meta-model, the authors in [15] present an integrated domain-specific language coupled with a basic security model that promotes the designers modeling effort. Furthermore, the work by [16] provides two use-case studies to verify the usability of their meta-model. Nevertheless, none of the previous studies consider BD-specific security requirements of IaaS cloud deployment architecture proposed by this study.

A recently established NIST BD security Sub-Working Group (NBDs-WG) [17] addresses the importance of security and privacy measurements, definitions, requirements, and characteristics of BD systems. The mutual relationship amongst BD technologies and model-driven engineering (represented by software engineering) is investigated in [18]. In [19], Pekka and Pakkala analyze published implementations of BD architectures (e.g., Netflix, Linkedin, and Facebook) to draft a reference architecture. In doing so, they aim to map different BD solutions that facilitate designing BD systems and create a classification of BD technologies, products, and services. In [20], the authors extensively illustrate BD ecosystem components based on the

TABLE II
A COMPARISON OF RELATED WORK ADVANTAGES AND LIMITATIONS IN IMPLEMENTING SECURITY BY DESIGN.

Paper	Year	Domain Focus	Planning & Modeling	Classification	Assessment	Requirement	Implementation
J. Sequeiros et. al. [13]	2020	Edge-to-Cloud	✓	✓	✓	✓	X
V. Casola et. al. [21]	2020	Cloud	✓	✓	✓	X	X
A. Chattop et. al. [22]	2020	Auton-Vehicle	✓	X	X	✓	✓
D. Polverini et. al. [24]	2018	Data Storage	✓	✓	X	X	X
V. Casola et. al. [23]	2018	Cloud	X	X	✓	X	✓

NBD interoperability framework. Their architecture framework consists of BD infrastructure, BD analytics, data structures and models, BD lifecycle management, and BD security. However, they do not investigate cloud-specific security requirements, components, or delivery within an IaaS cloud as this study proposes.

Security-by-designgets momentum with cloud computing as a security enabling tool to integrate security engineering in the software development and engineering processes [13]. Based on security service level agreements, Valentina Casola et al. [21] proposed automated security-by-design for cloud applications. Their solution relies on a risk analysis process and a completely automated security assessment phase to assess security requirements. On the same note, the autonomous vehicles' security requirements using security-by-design was reported in [22]. The authors proposed implementing security objectives and the necessary control measures from the risk mitigation techniques and adversarial model perspective. Data privacy and protection in the information and communication systems and infrastructure were discussed in [24] to shed light on the need for improved security by design. Meanwhile, the definition of a security-by-design development process of multi-cloud application deployment was proposed in [23] by proposing optimal deployment identification.

Overall, none of those above mentioned studies rigorously consider the BigCloud security specifications, which shape the security deployments in the design phase. Hence, we argue that the Big Data security deployment requirements have not been thoroughly investigated. Moreover, current literature is still missing an in-depth analysis and systematical methodology to apply security-by-design over BigCloud deployment architectures. To cope with those limitations and fill this research gap, this paper tries to put together the most innovative results and introduce a BD security-driven process to optimize the deployment of BigCloud components in the cloud environment. Our study distinguishes itself by proposing the most up to date and comprehensive discussion of the BD security requirements a the IaaS cloud deployment architecture. As previously discussed, our framework relies upon the adoption of four phases, (i) preparation phase, using a novel reference architecture with different service layers and security services; (ii) initializing phase, using a security component diagram that reviews the primary BigCloud security components; (iii) examination phase, using analysis pattern that describes the basic structure while applying the solution; (iv) selection phase, using a novel security criteria election approach that influences security in a BigCloud context.

III. BIGCLOUD SECURITY REFERENCE ARCHITECTURE

Many security threats regarding BigCloud platforms can be mitigated using traditional security processes and techniques. However, some security threats require cloud-specific solutions. BD frameworks have different security vulnerabilities and may be exposed to various threats. Thus, in addition to setting BigCloud security service requirements and data storage components, it is significant to define whose responsibility is to protect them. Therefore, we specify a vocabulary of design elements associated with BigCloud actors (system components) by presenting the BigCloud Reference Architecture (BCRA), which outlines cloud applications' main components. In general, the reference architecture is a template solution of an interconnected set of clearly defined concepts consisting of a domain-specific ontology. BCRA summarizes the relationship between the security service and other cloud services as well as their functions. Further, the BCRA model defines a set of implementational requirements and characteristics that can be used for orchestrating a secure BigCloud ecosystem. Therefore, it relates to companion security requirements and features that are the basis for designing a reliable BigCloud implementation. Figure 3 illustrates the five major cloud actors of the BCRA framework, excluding the client itself: service delivery, management, auditing, data, and security services.

It is worth mentioning that our reference architecture meets the NIST work in [17]. Both reference architectures share common features, as both of them are not tied to any specific cloud service provider or service model. Both architectures

represent a set of actors and functions used in big data over the cloud computing definition. However, NIST work targeting BD operations in the BigCloud environment in a generalized manner without providing a design definition of specific BD-related domains. Doing so can guide a high-level conceptual model for decision-making effort, but not developing a comprehensive BD solution. For instance, our RA proposes different BD services and their specifications, e.g., data services that include data lifecycle, which defines a solution and its implementation. Hence, BigCloud RA is tailored to facilitate the discussion of BD requirements, design structures, and operations inherent in the cloud deployment model. Besides, our work reports on a set of views and descriptions that are the basis for discussing security-by-design characteristics, uses, and standards for BigCloud deployment. Section III-B further presents the characteristics used, as well as standards for describing BigCloud security in detail, whereas Section IV and V present BigCloud security-specific elements and considerations, respectively.

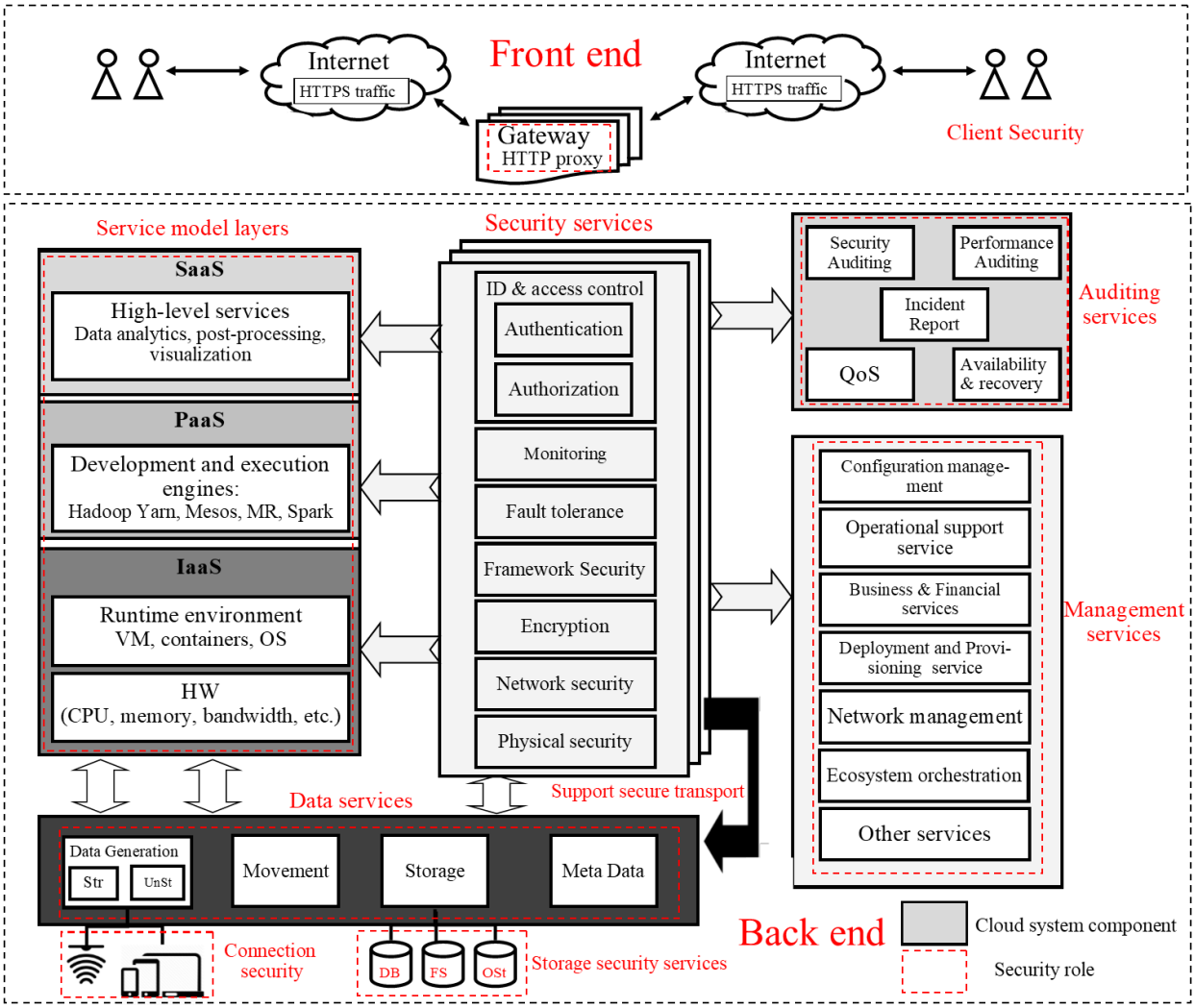

Fig. 3. The reference architecture as part of the security management of BigCloud specification development.

A. *BigCloud reference architecture components*

1) Client Security: An entity (organization or user) that has a formal contract or arrangement to maintain a business relationship with a cloud provider to use IT resources and other services made available by the provider. The cloud client security complements the providers security and components. The client accesses the service by applying a session that defines interaction security, using service level agreement (SLA) and policies. The session establishes client permissions and log method, and it even configures session timeout values. These sessions have the effect of mirroring services across all layers and system components. Assuring the sessions availability regardless of whether there is an attack (e.g., denial of service) or a system failure is in the BigCloud service providers interest, along with securing access, user identification, and authentication. Providing the needed level of training and awareness among users (such as strong passwords) are considered a common interest for both the client and provider.

2) Service delivery: This represents the three types of cloud delivery models in the form of layer abstractions: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Figure 2 shows how each layer defines its role (operation and function) within the BD stack. These roles require input and interaction between customers and the service provider. The service delivery layers serve as a simplified translation of business demand into technology and operational capabilities. The service delivery model is shaped and presented from a BD viewpoint. It focuses on the BD framework service delivery as cloud-service abstraction layers.
   In an IaaS deployment, the capabilities are split into hardware, network, and runtime environment. Fundamental computing resources, such as CPU, RAM, and bandwidth, are at the base of the model. While the runtime environment can deploy and run arbitrary software that includes operating systems (OS), virtual machines (VM), and containers, the client does not control the underlying infrastructure but has limited control of select networking components (e.g., host firewalls and virtual networks). Modern BD schedulers utilize the containers as a runtime environment of their applications [26], [27].
   PaaS deployment describes the relationship between the cloud provider and the cloud client. The capability granted to the client is to deploy applications using libraries, programs, services, and tools established by the provider. The development environment consumes the runtime services (VM, etc.) from the previous layer, so the user has administrative rights over deployed applications and configuration settings for the application-hosting environment. For instance, a client can select the BD platform (e.g., Apache Hadoop [25] and the execution engine (e.g., MapReduce [28] for a batch query, Spark [29] for micro-batch, or Storm [30] for real-time processing).
   Finally, the SaaS deployment layer provides high-level capabilities such as post-processing operations and visualization capabilities. However, the client does not manage or control the cloud infrastructure and the development or runtime capabilities. For this reason, only the cloud service provider is responsible for efficiency and security.
3) Management services: The capabilities that enable the management of the service-delivery model. Typically, these are services to which the provider connects rather than the client. They refer to a set of services designed to ensure that other cloud components are working optimally for BD framework operations. Moreover, management services present an entity that manages the operation and interaction between the client and the cloud provider . Thus, it is critical to maintain the same security levels for the service delivery layer as for large-scale security monitoring [31] and continuous system-security auditing [32], while retaining authentication and authorization access control. The cloud provider must assure and maintain overall proactive security governance of management services. Different BD frameworks can be implemented to harness management services For instance, Apache Zookeeper[1] for BD ecosystem orchestration and Apache Ambari[2] for cluster deployment and provision service.
4) Auditing services: This includes the assessment of cloud services, operations, performance, and security auditing of cloud implementation [31]. It also assures system availability, quality of service, and recovery plans. Security auditing defines and reports on security policies (e.g., password complexity levels). Furthermore, it evaluates recovery policies and the quality of security services while maintaining the reporting of security incidents. A multi-replica dynamic auditing of public multi-tenant data storage on cloud computing is re-ported in [34].
5) Data services: The underlying data service provides storage capacities on demand, either within virtual disk drives using a hypervisor and containers or with direct access to physical storage. The tasks associated with these services include all data stagesfrom data collection (generating structured or unstructured data) to data in rest within file systems (FS), databases (DB), and object storage (OSt) as illustrated in the Figure 3. These services also include data movement, also known as data placement, from storage to virtual machines and vice versa and other data operations and processing services, such as storing meta-data. The importance of these services is magnified in data-intensive batch-based systems (e.g., Hadoop MapReduce). Since data must materialize in storage before the process can begin, these services must provide the capability to backup and restore data by establishing data protection policies at the service layer.
6) Security services: They define a broad set of technologies, policies, and controls deployed to protect data, services, applications, and the associated infrastructure resources of cloud computing. By managing the on-going delivery of security, these services represent the capabilities of the security life cycle. The RA verifies that the BigCloud security is a cross-cutting interest that influences all the components in the model.

B. BigCloud Security Characteristics

Due to its inherently remote operations, resources co-tenancy, distributed management, and administrative control, ensuring the privacy of BD workloads while outsourcing computation is crucial. Customers do not have direct control over the systems that consume their data because of the clouds black-box nature. The following are the most pressing challenges in assessing data protection before a move to the public cloud:

- Data residency:This refers to the physical geographic location of the data stored in the cloud . In conventional BD systems, such as on-premise clusters, the geolocation of data is always known and, thus, controlled. When deploying a BigCloud

[1]https://zookeeper.apache.org
[2]https://ambari.apache.org

system, the physical location of the data is no longer known or fully trusted. Data residency also includes data flow, file locations, and data input/output.

- Data privacy: This describes the ability to limit data sharing in BigCloud systems, including third parties through an organization or individuals. Maintaining an appropriate data privacy level can be achieved by exploring various technologies and tools, including encryption [36] and virtual mapping [35]. Other solutions include modifying policies and legislation to prevent unauthorized access or use of data. However, defining legal ownership, responsibilities, and privileges of data between owner and data custodian can alleviate privacy threats.
- Data ownership: A serious concern within BigCloud data processing is data ownership. When clients transfer their data to the cloud, the primary processor of that data is then not the physical owner but the provider. Consequently, a new threat parameter is raised regarding trust in that provider. Clients cannot be sure how the cloud system manipulates their data or whether the processing complies with their demands.

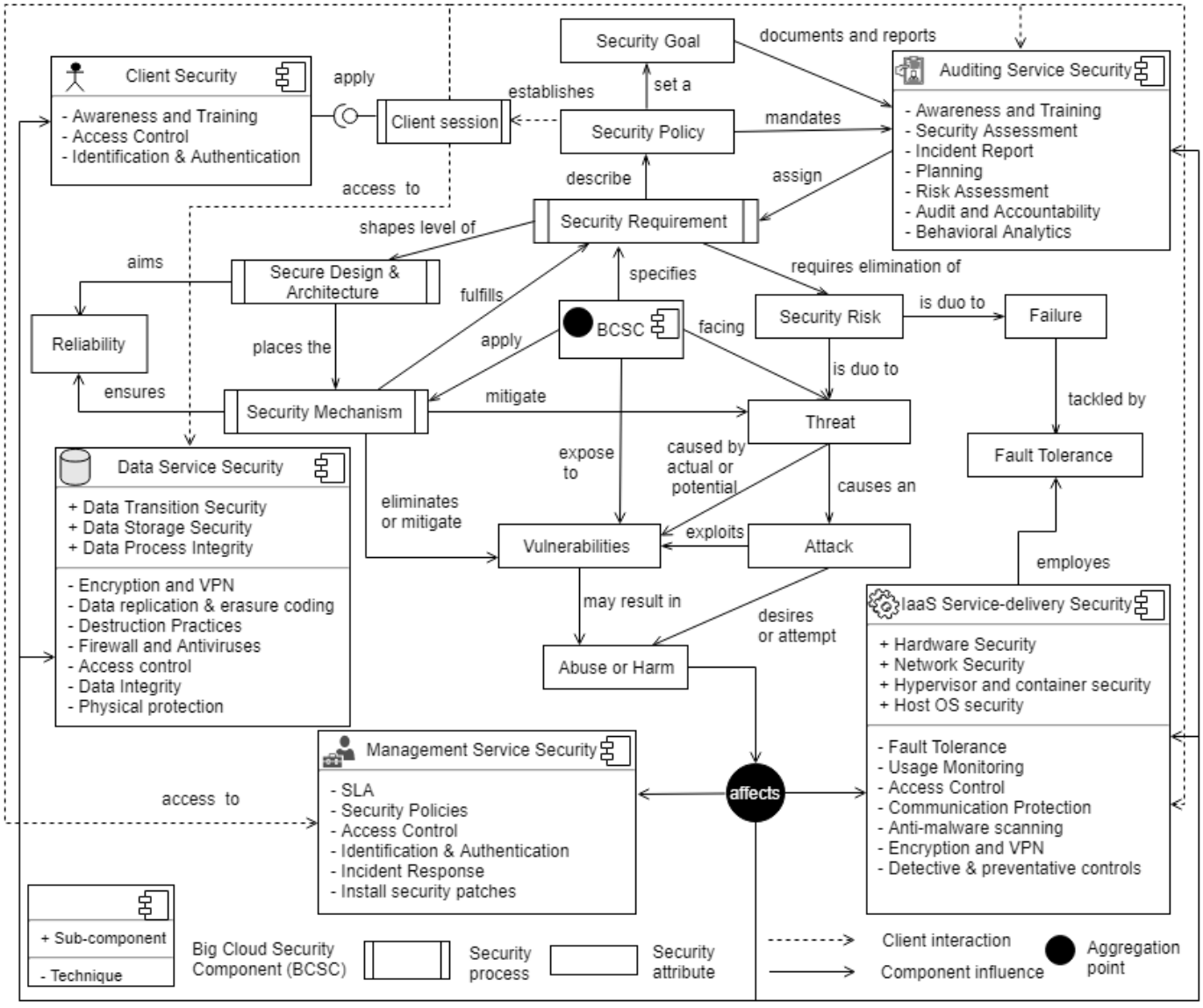

Fig. 4. A novel BigCloud Security Component (BCSC) diagram as a structure of software design pattern with associated security attributes.

IV. BigCloud Security Component Model

Any software system consists of a vocabulary of design elements, a set of configuration rules, and a semantic interpretation to refer as component model (or diagram in UML). The technology-agnostic BigCloud Security Component (BCSC) model represented in Figure 4 is a logical extension of BD application security in cloud computing definition. As highlighted earlier, BCSC is a generic, high-level conceptual model that facilitates the understanding of successful implementation of trusted BD in a cloud environment. From this perspective, it summarizes operational intricacies and component interaction of BigCloud

security. The BCSC does not represent the system architecture of a specific cloud vendor. Instead, it is a framework for describing, evaluating, and developing a system-specific architecture using a shared cloud security component of reference models, along with their activities and functions. This actor-based model is intended to serve designers by representing the overall view of roles and responsibilities for the assessment and management of risk by implementing security components and controls.

The BCSC model specification relies on the predefined BigCloud RA components described in Section III By analyzing these components, we conceptualized a BCSC model to represents the implementation of security-specific techniques and sub-components. The proposed model demonstrates the security process and attributes (actions) that influence, and are influenced by, the BCSC and describes the structural relationships among them. In fact, the BCSC model is a logical design constructed with replicable items (i.e., the sub-components and security techniques can be modified according to use-case). This approach ensures reuse and substitutes these components and actions within any BigCloud application security design. It also offers the architects a reference model to verify that the installed security plan/design meets their system security functionality. Moreover, it can be utilized as a communication tool for various development groups, as well as project stakeholders and implementation staff, as it provides a high-level, architectural view of BigCloud security. This model assists considerably in formalizing the implementation roadmap for the security integration.

The BCSC model consists of five main components: client, data service, management service, auditing service, and IaaS service-delivery security. Sections V-Aand V-B further discuss security-related considerations of targeted sub-components. The BCSC model is also composed of three security processes and two aggregation points. These processes (described in Section V-C) represent a set of techniques, tools, and methodologies to achieve their goals. On the other hand, the aggregation points facilitate the diagram with a decent and uncomplex view. The BCSC aggregation point, located in the center of the model, resembles integral components (i.e., it can be implemented among all of them). The relationship between components and other entities is represented by two independent arrows, where the arrowhead connects with the provider. The continuous arrow resembles the entities interactions (or influence), while the dashed arrows represent the clients interaction.

V. BigCloud Security-related considerations

This study offers an extensive analysis of BigCloud IaaS service delivery and data-service security. However, due to the vast research area, an in-depth examination of management, auditing and client service security are not within this research scope. These items offer material for future work or an open research direction. Herein, we examine the implication and remediation of the most relevant security components in the BigCloud paradigm: IaaS service-delivery security and data-service security. This section discusses other security considerations and processes when implementing a BD system within a cloud development model. These security threats mainly originate from issues such as multi-tenancy, loss of control over data, and trust [42].

A. IaaS Service-delivery Security

The IaaS service delivery represents the technology stack in which each layer provides services to the layer above. The reference model categorizes security services among the IaaS layer to a runtime environment or hardware and network components. The BCSC diagram specifies these components in detail by charting out the IaaS service-delivery layer as follows:

- Hardware security: Hardware resources (e.g., CPU caches, GPUs, and RAM) deliver their services in a scalable way by sharing infrastructure. The underlying resources that provide this infrastructure were not often designed to offer robust isolation features for multi-tenant architecture. A virtualization hypervisor or container mediates access between guest operating systems, and these computing resources are utilized to address this issue. Security measures should still be employed to ensure that individual customers do not impact the processes of other tenants operating on the same cloud provider.
- Network security: Provides the network connection that supports IT activity, which includes network fabric, virtual local area networks (VLAN), connectivity, and segmentation. Network services are responsible for delivering clients data to storage capacities and linking system components. They also support a secure movement of BD meta-data and pass the workload to the process units among all service-delivery layers. Connection security is a critical factor in securing the delivery of services; network and communication carriers provide the distribution of any BigCloud services . Due to its significance, the network architecture design should treat client connections with a minimal level of trust. Clients will always access cloud services using a remote network connection. A set of security measures should be employed to mitigate channels (and network services such as DNS) that transmit data to and from cloud structures. Secure sockets layer, transport layer security encryption, and VPN technologies are examples. Firewalls should also deny any attempts to access a BigCloud service from a session that should not be connected to that service.
- Hypervisor and container security: Virtualization technology is a technique that allows multiple OS running concurrently on a host environment. It is also a resource abstraction component that ensures efficient and reliable usage of underlying physical resources, including computation and storage. This resource abstraction acts as a security component by itself, using proper configuration and permissions. A cloud provider would utilize hypervisors or containers for resource pooling. Doing so provides and manages secure access (among other advantages) to its physical computing resources. The

TABLE III
COMPARISON OF KEY SECURITY MECHANISMS OF DATA STAGES

Data Security Lifecycle	Access Control	Data Integrity	Data Destruction	Physical Protection	Erasure Coding	Encryption	Firewall and Antimalware
Data storage	√	√	√	√	√	√	√
Data transfer	√	√				√	√
Data process	√	√			√	√	

security aspect behind this subcomponent refers to access control (ensuring authorized access to services, data, and other components) and usage monitoring. The security topology should not expose user interface service functionality to non-privileged users. Malware scanning should follow that access control, ensuring comprehensive security monitoring of the whole environment [58] [59] [60].

- Host OS security: Operating System security commonly involves configuring the host OS that supports the virtualization environment. As with all OS configurations, a fundamental approach is to reduce the attack surface to an acceptable level. For instance, OS images used by a cloud provider can introduce risks to the cloud client when using pre-owned virtual machines. The main threat arises in uploading images with built-in Trojans. Thus, the authentication level to minimize the risk will depend on the overall risk strategy and threat surface model.

B. Data Service Security

Data service security includes data protection and monitoring in the three stages of the data-security lifecycle data-at-rest, data-in-motion, and data-in-useas follows:

- Data storage (data-at-rest) security: The primary capabilities of data storage include managing the storage required by BigCloud frameworks. However, modern cloud storage components can implement backups (including virtual storage). These backups may consider a remediation technique that promotes working with the process to create snapshots at regular intervals. Another way that the storage component can collaborate with the hypervisor or container is to allow for workload migration and storing metadata among host compute nodes. Data storage security services must cover file systems, databases, and object storage security scanning (data content discovery) to identify and locate sensitive content (e.g., credit card numbers). This method supports data compliance and auditing efforts by providing comprehensive reporting on the effectiveness of data storage protection mechanisms; it also gui des decisions on security measurements for implementing data encryption (disk-level encryption) and masking, removing, or warning the file owner. In general, data-at-rest is considered more vulnerable than data-in-transit [37]. However, Hadoop normal mode does not provide encryption functionalities at their Distributed File System (HDFS), which leads to the generation of Hadoop security-complementing ecosystems.
- Data transfer (movement) security:: Data transfer can be classified, based on the connection domain zone into internal and external data movement. First, internal data transfer occurs between storage capacities and processing units. This communication usually takes place at the platform layer. The BigCloud should consider the internal network as an untrusted network alongside the Internet. Hence, all data transfers (including the meta-data) should be handled with the same level of minimal trust. However, ensuring a high-level of security requires a) encapsulating the data workloads; b) sniffing the traffic on the network using proxies (to identify the content); and c) monitoring, reporting, and blocking abnormal bandwidth usage (using central policies) based on the traffic type.
  Second, the external data transfer occurs between the client and the BigCloud provider. Here, the network acts as an intermediary that provides data transport using different communication methods, from dedicated network channels to the open Internet. Using the Internet is still the dominant pattern as it cuts costs. In this case, it is the clients responsibility to recognize the full set of security measurements to secure the data migrated to the cloud, as data can be intercepted in transit. On the other hand, the cloud may require the network provider to provide secure connections between it and its clients to reduce vulnerabilities (e.g., man-in-the-middle attack) in Internet transmission channels to a minimum. The network service provider should maintain security control points, maintain security testing, and prevent suspected tasks.
- Data processing security: This refers to securing the processing environment. High-reliability data execution may be achieved by I a) harnessing robust distributed file systems permissions and b) enforcing isolation among computing instances, workloads, and applications. Therefore, it is ideal to protect platform/application configuration file(s) with appropriate access control. Doing so will prevent the attacker from modifying these critical settings. According to a recent classification of malware attacks in IaaS execution environments [53] , 71% of these attacks target the hybervisor denial-of-service. In contrast, fault tolerance is the most important aspect when discussing data processing security. To support high reliability and availability of BD operations, data blocks used to be duplicated across multiple nodes. This traditional approach was costly and returned with a moderate performance in massive operation scales [46] , which lead to the advancement of modern large-scale distribution storage systems with erasure coding techniques. This storage technique

provides the same level of fault tolerance with much less storage space and has been implemented with the HDFS [38]. Providing cell-level encryption for HBase on the runtime was proposed in [53] by Intel.

C. Security Processes

1) Security mechanisms: Table III maps the data life-cycle stages to the primary implemented security mechanisms. The table illustrates that data-at-rest is considered the most vulnerable, so it requires a larger number of security mechanisms. Both data-replication and erasure-coding techniques consider fault tolerance utilities while an encryption and protection of the integrity of data in the transition stage is expected. The use of an adequate data sensitization technique to deliberately, permanently, and irreversibly remove or delete data after ending the service contract must be set at the service-level agreement. Preserving composable security for high-level abstractions of data analytics and mining is also of security interest. On the other hand, secure computations in distributed data-processing frameworks that are deployed over decentralized clouds (e.g., edge and fog clouds) should be considered within the security design. In the meantime, these architectures demand real-time security and compliance monitoring.

As mentioned earlier, Hadoop standard security configurations may lead to several vulnerabilities. This issue can be tackled by utilizing the latest Hadoop 3.0 secure model, which consists of a service level of authentication and authorization [43]. The security issues of BD authentication are extensively discussed in [47]. Traditional data encryption may be employed at different layers according to Hadoop 3.0 [44] namely, application-level, database-level, filesystem-level, and disk-level encryption, in which HDFS-level encryption is placed between the database and file-system-level encryption. Accordingly, HDFS continues to provide reliable performance, while BD frameworks run safely over encrypted data. This encryption level limits the runtime level attacks as the OS interacts with encrypted data blocks. Besides, several BD security frameworks may be utilized in synergy towards a comprehensive security solution [48].

Other BD-specific tools and techniques to improve the security ecosystem may include, but are not limited to the following:

- Apache Knox gateway [49] over HTTP/HTTPs, which provides perimeter security with REST API authentication and control access gateway utility for Hadoop clusters and ecosystems. Apache Knox may also provide end-to-end wire encryption using a key-store to hold the SSL certificate and SSO plugin [50];
- Apache Ranger [51], a security orchestration framework with centralized administration and User Interface (UI) to enable, monitor, and manage data security across the Hadoop Yarn clusters;
- Apache Sentry [52], which provides the ability to establish fine-grained (role-based privileges) authorization on both users and applications data and metadata within Hadoop clusters.

2) Security design and architecture: It is essential to address the BigCloud-specific security demands to completely illustrate the various security components in a conceptual context. These demands may be summarized as follows:

- Continuous vulnerability assessment and remediation;
- Data recovery capability;
- Maintenance, monitoring, and analysis of audit logs;
- Automation data protection.

After identifying the security component and functional requirements for the adoption of BD in the Cloud, it is of research interest to highlight security design guidelines when prototyping a BigCloud system:

- Component-based architecture: Quickly add new behaviours.
- Highly available: Scale to very serious workloads.
- Fault tolerant: Isolated processes avoid cascading failures.
- Recoverable: Failures should be easy to diagnose, debug, and rectify.
- Broad network access.
- Decreased visibility and control by client.
- Dynamic system boundaries and commingled roles/ responsibilities between client and provider.

3) Vulnerabilities: These are exploitable system bugs, and they can be exposed remotely across all cloud-service delivery layers. Attackers mainly target the vulnerabilities within the operating system (system kernel, libraries, and application tools). Hence, all services, components, and data face significant risk. Plenty of remediation mechanisms (spanning from planning a secure design to performing compliance testing to validate the security measurements) may be implemented. Moreover, modeling risk patterns and vulnerability scanning, followed up by installing security patches, can mitigate security gaps, as appropriate risk patterns can capture most vulnerabilities [39]. According to [45], over public clouds, Hadoop suffers from an overloaded authentication key and the lack of fine-grained access control at the data access level.

VI. Structured selection of BigCloud Security services

Security election is a control element that shapes policies, practices, procedures, and responsibilities of the IaaS cloud provider. Aiming to better understand this process, we propose the BigCloud security analysis pattern that captures an abstraction of threats using several attributes, behaviors, and expected interactions. These entities are employed to achieve security goals and provide general design guidance to eliminate the introduction of vulnerabilities.

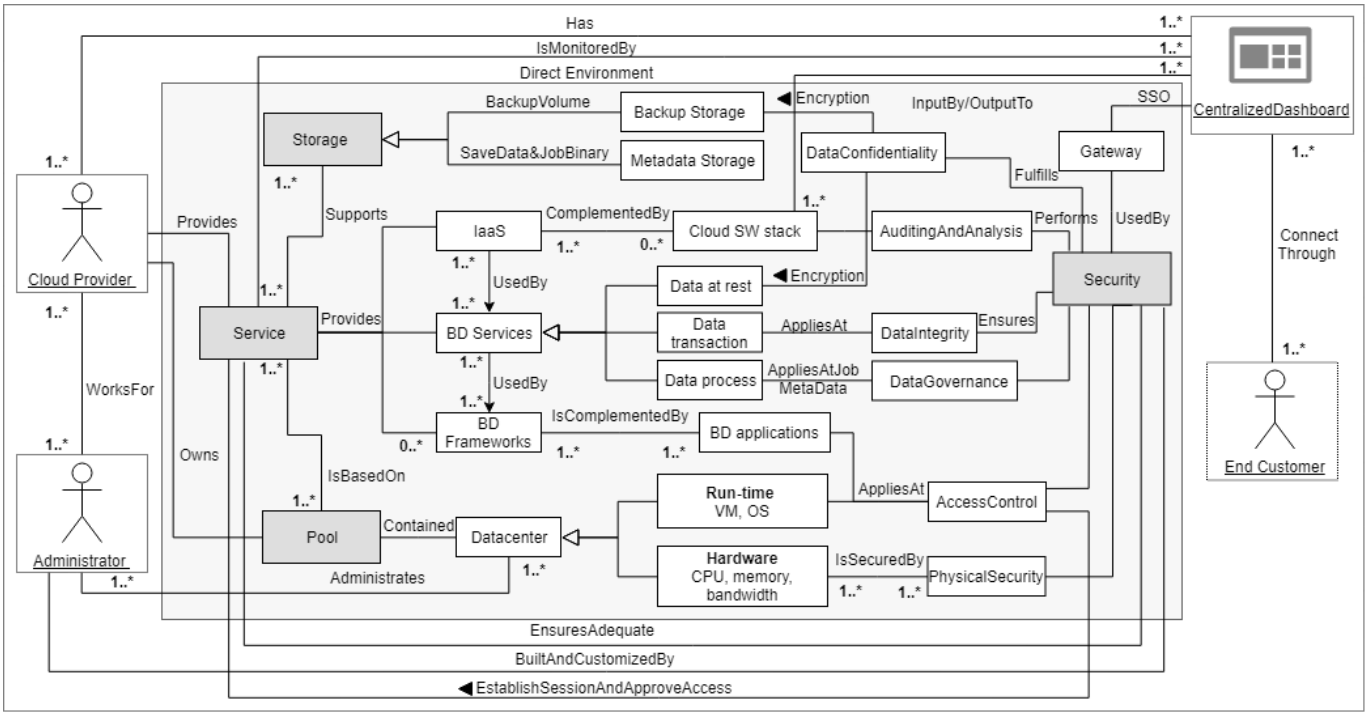

Fig. 5. BigCloud Security Analysis Pattern (BCSAP).

TABLE IV
PATTERN RELATION BCSAP TO THE CLOUD PATTERN FROM [54].

Direction	BigCloud to Cloud Pattern
Relation Type	Refines
Reasoning	The services deployed in a BigCloud can be created or composed based on cloud patterns. Thus, the information in the cloud can be translated as refinements descriptions of the service in our BCSAP model. Specific technologies that form the core of the BigCloud resources pool (e.g., storage) are mainly based and rely on the cloud-architecture. Moreover, the cloud architecture is essential for the mapping of BigCloud security specification design.

A. BigCloud Security Analysis Pattern

Figure 5 presents the elements and concepts of a secure BigCloud model and the relations between these components. We propose the BCSAP patterns for a structured domain knowledge election using the context-election pattern proposed in the cloud system analysis pattern [54]. Furthermore, the relations between existing context patterns and the BCSAP are defined in Table IV. The BCSAP shows a meta-model that forms a uniform basis for the current and future BigCloud security deployment. The generalization of its elements creates the basis of a pattern language that ships other deployment architectures (e.g., fog and mobile edge-cloud security).

B. BigCloud Security Requirements Election

Hosting data off-premises increases the number of potential security risks. By considering the security as mentioned in the BCSC, we divided the security issues, requirements, and characteristics of BigCloud secure deployment. The following sections give details. Figure 6 is a model designed to guide policies for information security life cycle within BigCloud systems.

In this model, the input phase is a set of security use cases, requirements, and patterns that realize a well defined and structured security business intelligence and policies. The security method is a description of a process for accomplishing a security objective, usually expressed in terms of tasks and their dependencies. Hence, we refer to different defense security layers (from the gateway to the core defense) as security methods, which may explain as follows:

1) Security initialization: This policy establishes the level of security design by describing security requirements through analyzing the security goals and use cases. This process is implemented using a service-level agreement (SLA) with an IaaS client

2) Gateway: In a cloud deployment architecture where a multi-tenant environment is a dominant model, it is critical to control the clients access to the internal cloud entities (resources and services) by defining which users and groups have access to a specific entity. In the case of BigCloud, these entities are represented in the direct environment. A cloud gateway in this context is a single sign-on layer that links the client requests, off a dashboard, to the authorized entity of a cloud SW stack securely and efficiently. This layer verifies the external clients access to the system using their user ID and passwords. Every client username and IP address has to be in the clients host file (/etc/hosts) or DNS table, and it has to match the clients given password. This process may also include Apache Knox , a unified gateway framework for Hadoop services and ecosystems that can be utilized as a SSO gateway. When connecting to a BigCloud cluster, there are two methods of authenticating the access. The first is a simple username/password identification approach. The second is an authentication using Kerberos protocol (authentication based on tokens). Each client and service must be authenticated by Kerberos keytab file (binary containing the information needed to log) to initialize trust between a client/application and the BigCloud components. Authentication for access to the Hadoop services web console requires enabling HTTP SPNEGO protocol as a backend for Kerberos credentials. Thus, the two approaches prevent unauthorized access to the stored data .

3) Access Control: In a BigCloud-based Business Intelligence environment, several user roles need to be enforced at the service level [64]. These roles must be provisioned dynamically to ensure large-scale participation while maintaining access control [61], [62]. This process improves security controls for authentication and authorization and enforces access discussions to meet BigCloud regulatory compliance. For instance, after users log-on to the cluster, the system must assign authorizations (i.e., access rights over a given service). The system manages access in the context of a specific service, resource, and data functionality provided by the cloud service provider. BigCloud should support a robust set of role-level security that can be utilized to configure the right level of application authorization for different user types, such as defining the users and groups who are authorized to make service calls to cloud storage service [63]. The call will pass the authorization check only if the user making call belongs to an authorized service entity. In general, the BigCloud platform security model supports three levels of permissions within IaaS:

- Application level control which users and groups are able to create, modify, and publish data within a BD application run within a specific execution engine (e.g., Hadoop). A client can submit jobs and query results of a predefined framework with limited access to the data.
- Framework level controls which users and groups are able to deploy, configure, and administrate a BD framework (e.g., MapReduce, Spark, Storm) over the given cloud instances. A client may access the runtime variables, paths, add/remove processing features, and change the scheduler (e.g., Fair or capacity) and the resource manager (Hadoop Yarn, Apache Mesos, etc.).
- Runtime environment level controls which users and groups are able to query and manage the runtime environment (VM, containers, OS, etc.). However, the client does not control the underlying infrastructure but has limited control (based on the SLA policy) to select networking components (e.g., virtual networks) and to select the OS and VM capacities and configurations.

4) Data Governance: BD sources and types can vary in their nature with multiple data processing patterns generally formulated as trees, graphs, or workflows. BigCloud should enable a client to maintain high data quality throughout the complete lifecycle of the datawith flexible mechanisms that store and access such data sources independently from their specific format. Moreover, metadata formalisms should be defined and used to describe the relevant information associated with data sources (e.g., location, type, format), enabling their access, use, and administration. A common platform is also essential for metadata exchange and storage within the different elements. This design will assist in supporting policies consistently across the BigCloud components. Apache Atlas provides data governance capabilities for the Hadoop stack and helps in searching, classifying, and managing data.

5) Data Integrity: Storing clients critical data over a cloud model requires robust data integrity and availability mechanisms. Cloud clients want to ensure that BigCloud provides appropriate data privacy and integrity between all components of the system as well as the data source with which they communicate. Supporting the appropriate security for these connections is imperative by ensuring adequate consistency and accuracy of data-in-transit. A block of data fetched from the storage file system or DB (e.g., an HDFS DataNode) could arrive corrupted due to faults in the storage device, network faults, or buggy software, as well as abuse or attack. Several approaches are implemented to tackle this issue, such as checksum checking on the transit data or wire encryption . Wire-securityfor data transfer between web console and clientmay be managed via Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) cryptographic protocols HTTP communications. The cloud provider can configure BigCloud Business Intelligence so that all communication between every component of the cloud system, as well as with the web-client traffic is secured using TLS/SSL.

Another implementation of end-to-end encryption relies on providing secure communication over public networks. In this case, all REST APIs offered by BigCloud components (like Apache HBase, Hive, and Oozie) are enforced to pass cryptographic protocols. Doing so requires creating a key store to hold the TLS/SSL certificate and set up environment variables. It takes

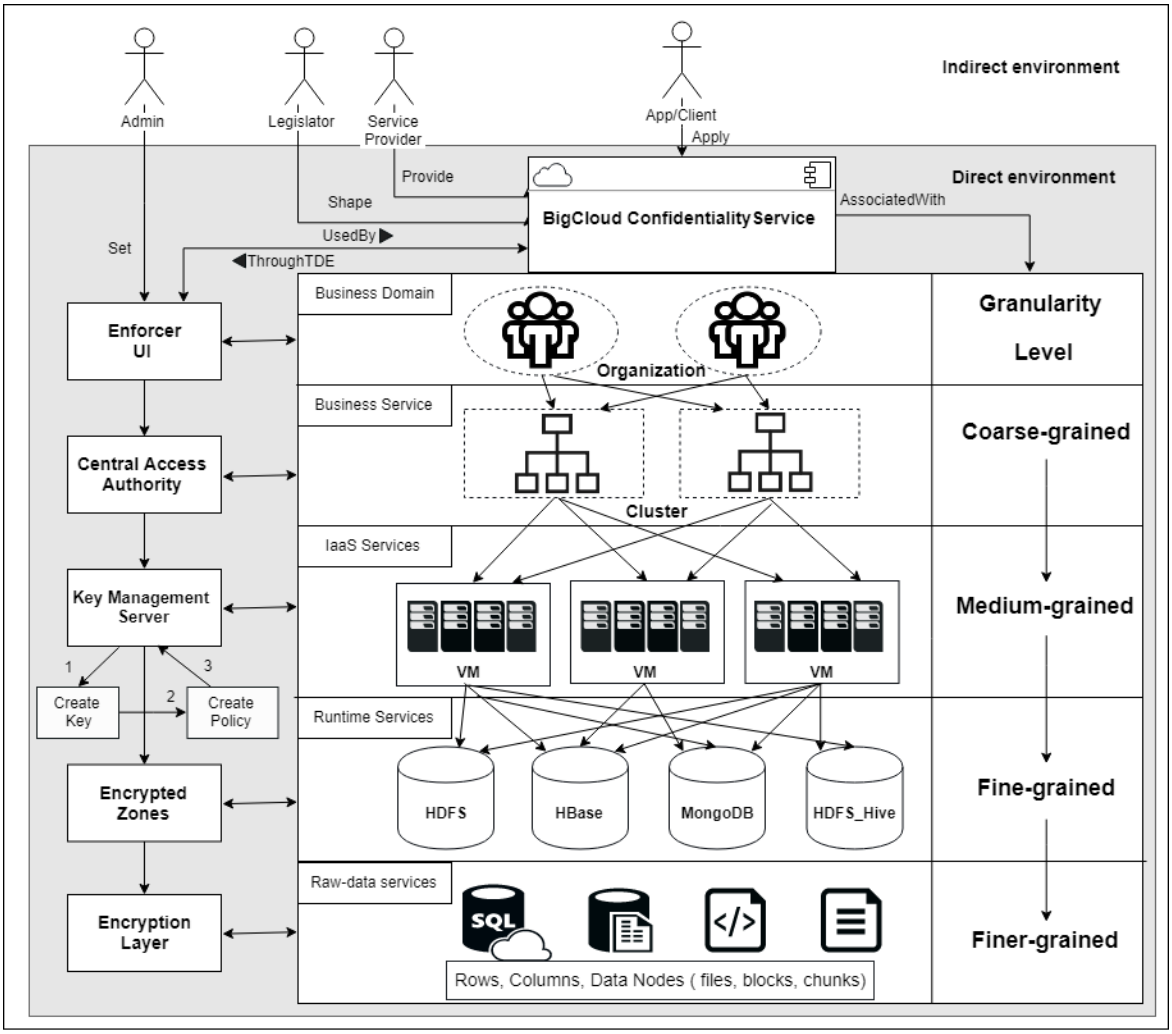

Fig. 6. Transparent Data Encryption Analysis Pattern with Encryption Zones Granularity in a BigCloud System Architecture.

two stages to set up a secure connection. The first uses digital signatures and asymmetric cryptography for authentication, while the second stage is for data transmission. Figure 7 summarize client connection to BigCloud services using a public network. After a secure session is established (steps one and two), both the client (e.g., result query) and BD frameworks (e.g., HBase data fetching) may access the data securely. The same approach applies to authenticate the internal components of communication upon an SLA policy. For instance, SSL certification to secure connection between the access control and the data storage requires either a self-signed or an authority-signed certificate. Thus, admins need to configure SSL on REST server (e.g., SSL-server.xml file) and configure a universal key-store to hold the SSL certificates.

6) Data Confidentiality: This requirement assures that a given stored data cannot be reached by any client/application except those who hold permission. Thus, data confidentiality in general aims at preventing protected data from being inappropriately accessed. It preserves authorized restrictions on data access, including job metadata . Cryptographic encapsulation enforcement by using distributed cryptographic protocols, such as PKI and identity/attribute-based encryption, is a common trend. This security layer also includes validity and recoverability approaches as Hadoop 3x starts utilizing erasure coding for fault tolerance. However, aiming for data confidentiality, Hadoops HDFS implement end-to-end encryption with so-called Transparent Data Encryption (TDE) [57]. These HDFS encryption sets are at the file-level of on-disk data and are stored as NameNode metadata. Further, HDFS TDE operations rely on encryption zone level of all components of a path, which means all files designated zones are encrypted on disk. In context, transparent at-rest encryption implies that the client/application access data without being aware the data was encrypted. It also indicates that data is automatically encrypted and decrypted on-the-fly as it is read or written. In However, it is not meant to hide sensitive data (e.g., data masking technique). Nevertheless, security policies like masking can be implemented on top of TDE data as a post-decryption.

Figure 6 shows the confidentiality layer components, stages, and granularity levels within a BigCloud ecosystem. Ensuring adequate services and data confidentiality while the client is signed into a privileged BigCloud session requires several layers of control. These layers for controlling data confidentiality are separated across four essential strategies. Table V discusses

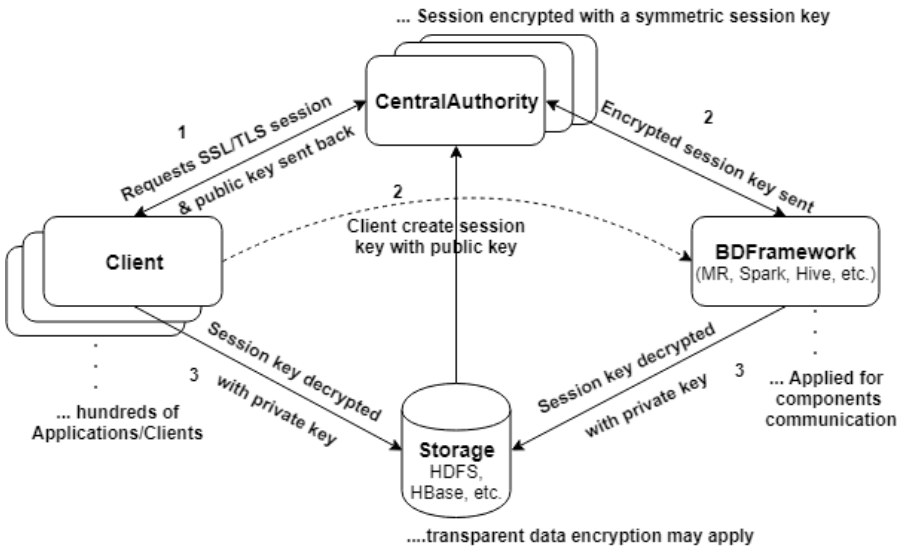

Fig. 7. Implementing data encryption discussion over BigCloud components.

these strategies by grouping similar techniques and mechanisms into the same layer of granularity. The table further specifies both the solution domain (threat category) and the solution limitation over a BigCloud security stack.

On the other hand, TDE encrypts HDFS data at rest (on disk) using an interaction of multiple components and security keys. Next, we illustrate these components and stages by defining these components and encryption steps:

- Enforcer UI is a panel board that is a subset of the general management panel to provide connectivity among the security services and customers. It also acts as a registration authority for all of the external logins and all REST APIs. As a first confidentiality stage for verifying clients calls, it employs either simple username/passwords or a third-party protocol such as Kerberos authentication. Additionally, system admins manage the process (create, edit, and delete policies) for the clients, groups, and applications that can use the service through the UI tool.
- Central Access Authority (CAA) is a policy-based authority that keeps issuing the encryption service that controls the client access for BigCloud customers. However, it represents the primary stage of the data confidentiality process that releases access policies and functionalities by matching each user/group with its granted permission key and enforcing encryption discussions.
- Key Management Service (or Server) (KMS) is a validation entity that approves client/application reading and writing permissions to the service (encrypted zone). Upon passing the KMS, it affords the master key to encrypt or decrypt the data. In this context, all of the policy creation, encryption, and decryption processes of data encryption key zones are managed in the KMS layer. Apache Ranger may utilize as a third-party KMS.
- Encrypted Zone (EZ) is a unique file path (a directory or a database) the contents of which are transparently encrypted. When establishing a new encryption zone, a single encryption key is associated with each of these zones. Moreover, the content files within these zones hold a private data encryption key. These keys are never handled directly by the CAA, as they only see a stream of encrypted chunks.
  1) Create Key: After creating the targeted EZ, the admin creates a key for each particular zone (EZK).
  2) Create Policy: The admin launch policy against each EZK, which spawn service inclusion (who can read/write to the EZ) and add clients/applications to that policy.
- Encryption Layer: The client/application informs the CAA it wants to write a file (e.g., SQL client accessing Hive) to a particular EZ. The CAA requests the KMS to return an encrypted data encryption key from the key store by establishing a trusted connection between the server and the key management server. The client may use that key to write/encrypt to the EZ and read/decrypt from the file. The CAA stores the encrypted data encryption key in the metadata store.

It worth mentioning that the CAA does not control directly the data encryption keys (encrypted data in the files), but it uses an encrypted data encryption key that can only be decrypted by the clients data encryption.

Confidentiality must be maintained throughout the complete data lifecycle. However, herein we highlight the BigCloud confidentiality challenges in terms of data halt, where confidentiality is delivered typically via data encryption techniques . Table V represents a comparison of data confidentiality granularity with related approaches to data security within the BigCloud system. Medium-grained encryption (MGE) enforces the decision of which files and directories to encrypt on clients behalf (i.e., clients discretion)thereby protecting the swap space, OS, containers, and temporary files as well. MGE, however, does not replace the fine-grained encryption (FGE) in all scenarios. The VM encryption may be employed in conjunction with the

TABLE V
BIGCLOUD DATA CONFIDENTIALITY GRANULARITY.

Less complex deployment ↑

Confidentiality granularity	Description	Threat category	Limitation
Corase-grained	Limit the system access (e.g., cluster) in a single access point as in SSO using username/ password authentication or Kerberos protocol, etc.	Unauthorized access to services and components including external attackers.	Addresses limited threats and doesn't consider internal attackers.
Medium-grained	Encryption over VM and full disk encryption, the entire run-time environment within the VM are encrypted	Unauthorized access to VM runtime contents and guard against physical attacks including privileged users.	Lacks safeguards against advanced threats and meet minimum auditing requirements
Fine-grained	Encryption over data nodes and tables, like HDFS, DataNodes files and databases,but not the whole file system, databases, or machine	Unauthorized access to designated data path in a file directory or database, including malicious insiders.	Doesn't support central orchestration across multiple storage's systems
Finer-grained	represents the smaller entities that can be controlled by a system admin.This layer may comprise a particular file, column, or row in data storage.	Prevents attacks at the filesystem-level and OS-level as the OS and disk interacts with encrypted data only including malicious DBAs and SQL-injection attacks.	Doesn't provide security over metadata configuration files and could lead to design complexity.

↓ High secure deployment

file-based encryption, seeking secure multi-layer encryption implementation.

On the other hand, FGE management operates over individual files, directories, and tables (i.e., accessible HBase/Hive DB table columns, Kafka queues, and HDFS file-level of acces s). With the ability to encrypt each of its components with a separate encryption key, the FGE provides flexible policy decisions and high-performance encryption. Therefore, FGE provides greater overall protection as it stays encrypted through the rest of the layers. However, this protection is at the cost of increased complexity (i.e., it is more comfortable to encrypt a hard drive than a specific cell for instance). FGE has to be associated with a robust access control mechanism and enabled wire encryption.

HDFS is a Java-based framework that requires a JVM environment and contact with the Linux kernel file system before reaching the stored data in the disks. Hence, the machine kernel security (of OS-level) is concerned with security design when processing sensitive data over a public cloud. Utilizing TDE could potentially cause slight performance degeneration as additional process layers are attached. However, doing so is justified and acceptable compared to potential threats. Alternatively, direct access to the multi-tenant database may be restricted to specific admins (e.g., DBA). The clients service calls may be routed through an intermediate business layer, which enforces security checks, including means for protecting personal identity and proprietary information. Moreover, visibility labels may be utilized by tagging cells in a table (e.g., Apache HBase) and controlling access to them. This method restricts the access to specific subsets of labeled data in a fine-grained manner.

In Table V, the coarse-grained encryption is the easiest to implement and manage, and it is the most flexible security approach. This layer limits the system access (e.g., cluster) in a single access point, such as an SSO using username/password authentication or Kerberos protocol. In contrast, the medium-grained encryption works on the runtime environment (i.e., VM access control and encryption). This reasonable compromise shifts the complexity of a solution to the required level of isolationespecially when implemented with other confidentiality layers. The next two layers (fine-grained) are the most secure yet complex approaches to acquire; they require very detailed policy definitionsincluding the DB , data node, and even control decisions on the file paths and the specific rows, columns, and cells of the target storage.

Azure Disk Encryption allows the encryption of IaaS VM disks by leveraging the volume encryption for the OS and data disks. This solution also ensures that all data on the VM disks are encrypted at rest in the medium-grained stage of cloud storage.

7) Auditing and Analysis: To cope with the modern security demands of large-scale distributed clusters, as in a BigCloud. Any security architecture should be able to perform security auditing and analysis at the service level. Security auditing and analysis aggregates log files and reports and provides a robust audit capability within different components of the BD ecosystem. This layer may also afford information by performing security and risk assessments, tracking data pipeline audit logs, and examining behavioral analytics to meet their compliance demands within BigCloud. Examples of this may include incident reporting, behavioral and data activities analytics, daemon (processes starting under the framework and running in the background) logs, and risk assessment of the system components regularly. This feature does not only identifying security issues but provides a sophisticated alert engine that identifies security vulnerabilities and shows insights. Apache Eagle is an open source analytics solution for the Hadoop frameworks and applications [65].

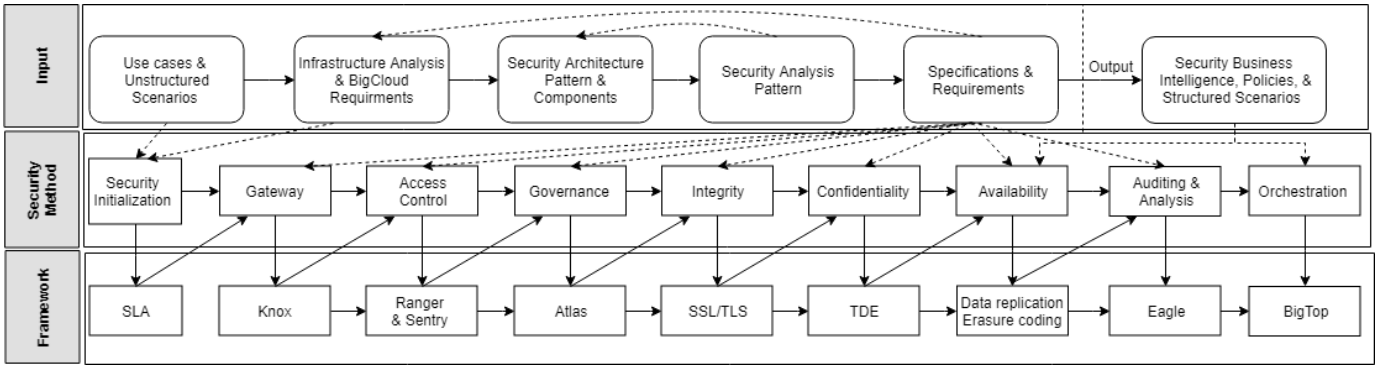

Fig. 8. Mapping the BCSAP knowledge domain to the Apache BD security framework domain.

8) Orchestration and Automation: With the increasing number of different security frameworks, policies, and products in a cloud stack, the connection and integration of these tools is a cornerstone behind inclusive security. This process is called security orchestration, and it brings together these various technologies to work in harmony for the benefit of its customers. It is crucial to standardize and model security to enable inter-operability among the various security subcomponents and products. This aids in supporting the heterogeneity of security deployment over IaaS using various security layers and tools. By bringing together security components consistency, it improves efficiency and effectiveness of security management and the processes surrounding them.

Moreover, the synchronization of the security ecosystem helps security admins and clients to make more informed decisions and aids in better specification development. Security orchestration involves advanced automation procedures by assembling security alerts across the ecosystem. By enabling universal alert repository, security implementers may execute automatable mitigation policies and standardized reactions scenarios. This feature strengthens the overall security operations and supports the right incident response. Apache Knox provides a common platform for frameworks interaction by abstracting the policy exchange. Likewise, Apache BigTop [56] equips the stack with comprehensive packaging, testing, and configuration of big data frameworks. Bigtop supports a wide range of components/ tools continuous integration using Jenkins server. Putting all of the previous in context, Figure 8 represent modeling the security requirements election by instantiating the BCSAP and Apache security frameworks with core components highlighted.

VII. Summary and Open Challenges

Security-by-design can be defined as the process of analyzing all security solutions and select the most suitable approach by taking into account the security requirement, performance, and cost to be implemented in the early stages of the system deployment. It is, indeed, a software engineering approach that embeds various approaches that include, but is not limited to, security analysis patterns, security requirement modeling, threat modeling, attack graphs, and security components diagrams. However, despite the importance of BigCloud security by design, it has not been significantly explored in the literature.

In this work, we argue that the realization of BigCloud security-by-design as a cross-layer objective in the system deployment life-cycle will shape the design of robust security architecture and guide its applications' security development as a mainstream development approach. Hence, BigCloud applications are composed of a robust security and privacy architecture that enforces the required authentication, authorization, data confidentiality, data integrity, data availability, data privacy, auditability, and non-repudiation requirements from the foundation secure.

To cope with the challenges mentioned earlier, we propose a systematic research methodology for the security of BigCloud adoption. By capturing the methodology stages, we design four primary models that guide any BigCloud solution's security deployment. First, we design a reference architecture to summarize the relationship between the security service and other cloud services and their functions. Second, we offer a solution to the main research question, which deals with BD deployment's security elements over the IaaS cloud model. Our solution proposes designing a security component model consisting of main actors that emphasize the separation of concerns for the service functionality (data service security, IaaS security, etc.) and non-functional security requirements at the beginning of the design. Third, we propose a security analysis pattern that refines the cloud context-pattern [54] in synergy to an extended CIA triad [55]. It provides a set of guidelines for structuring BD specifications, which relates a cloud design to its security environment. Finally, we suggest a structured election method for BigCloud-specific security selection. It delivers various insights regarding the latest ongoing developments and cutting-edge frameworks by mapping each security domain to its solution knowledge.

A. Recommendations

1) BigCloud Provider: Deploying BD frameworks in a cloud environment, whether private or public, demands proactive thinking regarding security ramifications. Security is magnified when considering the impact on clients sensitive data. This is

especially true when IaaS cloud providers control the underlying infrastructure (storage, servers, and networks), while clients have no control over these assets. This shift of responsibility requires providing capabilities to assure the functional properties of BigCloud security and the trust concern between the BD owner and the IaaS cloud providers. These concerns are based on a lack of control, visibility, and governance while outsourcing the clients data computation. Ensuring the security of BD frameworks over cloud deployment architectures is a keystone to sustaining the porting of BD applications to cloud deployment architectures.

A substantial effort has been made to solve the problem of cloud quality-of-service evaluation [40] . Data security and reliability are first-class considerations that play an essential role in most cloud-computing contexts. However, there is a remarkable research gap regarding the evaluation of the BD security service within the cloud [41]. A security evaluation framework can maximize the level of trust (between resource provider and user) and minimize the risk to an acceptable level. Hence, BD application implementers not only have a clear sense of whether the provided service security level is high or low but can also assist in improving the trust level among them. In this regard, a security evaluation framework that identifies unimproved gaps (according to security control elements) serves to converge BD operations to vast cloud environments. Clients will consider an IaaS cloud provider trustworthy if they fulfill the security requirements of a rigorous security evaluation framework.

It is still challenging to put forward such an improvement plan unless it is based on the results of security analysis that establishes clear security components and requirements. The BCSC model accommodates the previous security evaluation framework requirements. Cloud adopters, who are involved in the development of BD solutions, may leverage the BCSC to perform a security analysis that maps the installed/needed security components. Further, the BCSC guides the security designers in selecting the required security controls that best suit their demands. Overall, an evaluation framework with interest in BigCloud security would help in the following:

- Meeting client satisfaction: BigCloud service providers can provide adequate information regarding their system security, which indeed raises client satisfaction.
- Improving BigCloud security services: Security evaluation can play a vital role in meeting client demands by providing the IaaS cloud with security improvement initiatives and gap analysis.
- Managing security risk: Security evaluation results can guide providers in detecting unimproved security gaps between their current IaaS cloud state and the ideal security state.
- Guide porting new BD execution environments: A secure BD execution environment would serve to converge BD operations with the vast cloud paradigms (e.g., edge cloud, decentralized cloud, etc.).
- Gaining competitive advantages: IaaS cloud providers could use the results of the security evaluation framework to remain competitive in the market.

2) BigCloud Client: In an IaaS model, security is a mutual responsibility, so the client must pay attention to gain security-by-design approval. In this regard, client awareness is critical to achieve this goal. It is recommended to involve developers in threat modeling, creating an easily legible sense of awareness. Also, to carry out manual checks of the security installations and test for certain security vulnerabilities. It is also important to point out that expanding such an effort to include data privacy is also considered a recommended practice.

Password management, typically, clients passwords are managed using an encryption mechanism or digitally over plugins and extensions. The clients have to ensure that cloud providers protect their identity correctly and enforce strong and unique passwords. Other features used for automating the filling of the password and sharing credentialswhich are integrated to the browser's function over services (e.g., XML or REST) and allow users to change and randomize passwordsare required to be testified by the client for better cloud service selection. For instance, Azure Active Directory uses REST instead of the traditional LDAP, which is meant to run SaaS applications and provide identity management services. Meanwhile, AWS provides fine-grained access control to AWS resources.

B. Open Challenges

Cloud computing is being steadily adopted as one of the dominant paradigms of BD platforms. The cloud's new concepts —such as computation outsourcing, resource sharing, and external data warehousingincrease privacy concerns and security threats. BD frameworks, as an emerging technology domain, lack model-driven engineering to secure IaaS clouds. The software development methodology offered in this paper focuses on creating conceptual models that abstract the security-solutions domain —delivering reference modules, basic requirements, main characteristics, and best practices for securing BD-cloud operations. These concerns and security threats have been addressed in this research by drafting security models for BD cloud adoption. This study proposes a component model that manages to standardize terminology, define key components and their relationships, collect relevant solution patterns, and categorize existing technologies. It also presents a reference architecture for big data systems focused on addressing IaaS cloud deployment architectures' security concerns. This research demonstrates how to use this model for the development of practical security solutions.

The lack of specialized threat and attack modeling tools is a leading deterrent in realizing the secure BigCloud vision. In this context, the attack/threat modeling aims at facilitating the simulation, instantiation, and optimization of the system security to

design domain-specific attack languages that simulate possible attacks and threats graphs. As its primary target, this challenge should have the description and proposal of workflows for engineering and development of applications and systems where security modeling and engineering are fully integrated into the software engineering processes. It will also enable developing domain-specific knowledge that allows for more reliable environments. Another goal is developing ontological assumptions for the underlying vulnerabilities and deterrents.

C. Future work

It is worth pointing out that our study targeting IaaS Cloud service providers. The main threats and vulnerabilities affecting other cloud services models like PaaS and SaaS have not been considered and are out of this paper's scope. Adding these levels may introduce additional un-predictable risks and should be investigated in compliance with the principles of BigCloud Security-by-Design. The methodology's effectiveness depends on continually updating the threats and vulnerabilities and taking into account a more comprehensive set of assets. A future work to extend this paper finding could be investigating the BD frameworks over both PaaS and BDaaS. In this regard, BD as a service and storage deployment model, gains momentum recently by providing perceptive insights into BD that drive business intelligence and other applications for a viable advantage [66]. End-to-End security and security-by-design seem to be imperative in such a multi-tenant environment. End-to-End security and security-by-design seem to be imperative in such a multi-tenant environment.

We also plan to extend the models to provide different security engineering aspects for threat modeling and attack graphs. To develop this work with the security evaluation theory, it is expected to define and score the risks based on multiple criteria and weights; based on specific application domains like data streaming and batch query. Finally, to magnify the paper findings' effectiveness, it is recommended to include more threats and vulnerabilities and consider a broader set of assets in the design phase of the BigCloud deployment. Future work for this study involves a cloud security evaluation framework for BD applications. It includes refining the syntax and defining the semantics of the proposed reference model and mapping the reference architecture to different cloud security architectures and big data frameworks.

Also, in future developments of the BigCloud Security-by-Design, we plan to extend the model using a meta-language for threat modeling and attack Simulations [67]. The advantage of such domain-specific attack language is creating a generic attack logic that facilitates modeling and instantiation of BigCloud solution. Another future work is to extend this work by comprehensively evaluating BigCloud security to ensure data privacy protection, the integrity of information, and the availability of resources [68].

ACKNOWLEDGMENT

This work has received financial support from the Ministry of Science and Innovation of Spain (PID2019-104834GB-I00, AEI/FEDER, EU), the Consellería de Cultura, Educación e Ordenación Universitaria of the Xunta de Galicia (accreditation 2019-2022 ED431G-2019/04 and reference competitive group 2019-2021, ED431C 2018/19) and the European Regional Development Fund (ERDF), which acknowledges the CiTIUS-Research Center in Intelligent Technologies of the University of Santiago de Compostela as a Research Center of the Galician University System. Also, Dr. Awaysheh acknowledge support from the European Social Fund via IT Academy programme.

REFERENCES

[1] I. A. T. Hashem, et al., "The rise of "Big Data" on cloud computing: Review and open research issues," Information systems, vol. 47, pp. 98-115, 2015.
[2] Awaysheh, Feras M., Toms F. Pena, and Jos Carlos Cabaleiro. "EME: An Automated, Elastic and Efficient Prototype for Provisioning Hadoop Clusters On-demand." CLOSER. 2017.
[3] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," Future Generation Computer Systems, vol. 79, pp. 849-861, 2018.
[4] R. Buyya, et al., "A manifesto for future generation cloud computing: Research directions for the next decade," ACM Computing Surveys (CSUR), vol. 51(5), 2018.
[5] Awaysheh, Feras M., et al. "Next-generation big data federation access control: A reference model." Future Generation Computer Systems (2020).
[6] Amazon EMR Web service manage cluster Cloud platform [Online]. Available: https://aws.amazon.com/emr/ Last access: 20/11/2018
[7] Microsoft AzureHDInsight [Online]. Available: https://azure.microsoft.com/ Last access: 14/10/2020
[8] Google Cloud Dataproc [Online]. Available: https://cloud.google.com/dataproc/ Last access: 14/10/2020
[9] Cloudera Big Data cloud service provider [Online]. Available: https://www.cloudera.com/ Last access: 14/10/2020
[10] Hortonwork [Online]. Available: https://hortonworks.com/ Last access: 14/10/2020
[11] MapR [Online]. Available: https://mapr.com/ Last access: 14/10/2020
[12] S. S. Manvi, and G. K. Shyam, "Resource Management for Infrastructure as a Service (IaaS) in cloud computing: A survey," Journal of network and computer applications, vol. 41, pp. 424-440, 2014.
[13] Sequeiros, Joo BF, et al. "Attack and System Modeling Applied to IoT, Cloud, and Mobile Ecosystems: Embedding Security by Design." ACM Computing Surveys (CSUR) 53.2 (2020): 1-32.
[14] M. Hamdaqa, T. Livogiannis, and L. Tahvildari, "A reference model for developing cloud applications," CLOSER 2011, pp. 98-103.
[15] K. Kritikos and P. Massonet, "An integrated meta-model for cloud application security modelling," Procedia Computer Science, vol. 97, pp.84-93, 2016.
[16] T. Xia, H. Washizaki, T. Kato, et al., "Cloud Security and privacy metamodel: Metamodel for security and privacy knowledge in cloud services," Proc. 6th Int. Conf. on Model-Driven Engineering and Software Development, pp. 379-386, 2018.
[17] NIST Big Data Working Group (NBD-WG). [Online]. Available: http://bigdatawg.nist.gov/ Last access: 14/10/2020
[18] T. Arndt, "Big Data and software engineering: prospects for mutual enrichment," Iran Journal of Computer Science, vol. 1, pp. 3-10, 2018.

[19] P. Pääkkönen and D. Pakkala, "Reference architecture and classification of technologies, products and services for Big Data systems," Big Data Research, vol. 2(4), pp. 166-186 2015.

[20] Y. Demchenko, C. De Laat, and P. Membrey, P., "Defining architecture components of the Big Data Ecosystem," Int. Conf. on Collaboration Technologies and Systems (CTS), pp. 104-112, 2014.

[21] Casola, Valentina, et al. "A novel Security-by-Design methodology: Modeling and assessing security by SLAs with a quantitative approach " Journal of Systems and Software 163 (2020): 110537.

[22] Chattopadhyay, Anupam, Kwok-Yan Lam, and Yaswanth Tavva. " Autonomous vehicle: Security by design." IEEE Transactions on Intelligent Transportation Systems (2020).

[23] Casola, Valentina, et al. "Security-by-design in multi-cloud applications: An optimization approach." Information Sciences 454 (2018): 344-362.

[24] Polverini, Davide, et al. "Resource efficiency, privacy and security by design: a first experience on enterprise servers and data storage products triggered by a policy process." Computers & Security 76 (2018): 295-310.

[25] D. Cutting, M. Cafarella, Apache Hadoop, http://hadoop.apache.org, accessed: 14/10/2020, 2006.

[26] V. K. Vavilapalli, A. C. Murthy, et al., "Apache Hadoop YARN: Yet Another Resource Negotiator," Proc. of the 4th annual Symposium on Cloud Computing, ACM, p. 5, 2013.

[27] B. Hindman, et al., "Mesos: A platform for fine-grained resource sharing in the data center," NSDI, vol. 11, 2011.

[28] J. Dean, S. Ghemawat, "MapReduce: simplified data processing on large clusters," Comms. of the ACM, vol. 51(1), pp. 107-113, 2008.

[29] M. Zaharia, et al., "Apache Spark: a unified engine for Big Data processing." Comms. of the ACM, vol. 59(11), pp. 56-65, 2016.

[30] A. Toshniwal, et al., "Storm@ Twitter," Proc. of the 2014 ACM SIGMOD Int. Conf. on Management of Data. ACM, 2014.

[31] S. Marchal, et al., "A Big Data architecture for large scale security monitoring," IEEE Int. Congress on Big Data, 2014.

[32] C. Liu, et al., "Authorized public auditing of dynamic Big Data storage on cloud with efficient verifiable fine-grained updates," IEEE Trans. on Parallel and Distributed Systems, vol. 25(9), pp. 2234-2244, 2014.

[33] S. Lins, S. Schneider, and A. Sunyaev, "Trust is good, control is better: Creating secure clouds by continuous auditing," IEEE Trans. on Cloud Computing, vol. 6(3), pp. 890-903, 2018.

[34] C. Liu, et al. "MUR-DPA: top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud". IEEE Trans. on Computers, vol. 9, pp. 2609-2622, 2015.

[35] C. Hongbing, et al., "Secure Big Data storage and sharing scheme for cloud tenants," China Communications, vol. 12(6), pp. 106-115, 2015.

[36] Y. Li, et al., "Intelligent cryptography approach for secure distributed Big Data storage in cloud computing," Information Sciences, vol. 387, pp. 103-115, 2017

[37] L. Hao and D. Han, "The study and design on secure-cloud storage system," Int. Conf. on Electrical and Control Engineering, 2011.

[38] Apache Hadoop 3.0 [Online]. Available: https://hadoop.apache.org/docs/r3.0.0/hadoop-project-dist/hadoop-hdfs/HDFSErasureCoding.html Last access: 14/10/2020

[39] A. Palm, Z. Á. Mann, and A. Metzger, "Modeling data protection vulnerabilities of cloud systems using risk patterns," International Conference on System Analysis and Modeling, pp. 1-19, 2018.

[40] H. Alabool, A. Kamil, N. Arshad, and D. Alarabiat, "Cloud service evaluation method-based multi-criteria decision-making: A systematic literature review," Journal of Systems and Software, vol. 139, pp.161-188, 2018.

[41] D. Gonzales, J. M. Kaplan, E. Saltzman, Z. Winkelman, and D. Woods, "Cloud-trust: A security assessment model for infrastructure as a service (IaaS) clouds". IEEE Transactions on Cloud Computing, vol. 5(3), pp. 523-536, 2017.

[42] A. Gholami and E. Laure, "Big Data Security and privacy issues in the cloud," International Journal of Network Security and its Applications (IJNSA), January 2016.

[43] Hadoop in Secure Mode, [Online]. Available: https://hadoop.apache.org/docs/r3.1.1/hadoop-project-dist/hadoop-common/SecureMode.html Last access: 14/10/2020

[44] Transparent Encryption in HDFS, [Online]. Available: https://hadoop.apache.org/docs/r3.1.1/hadoop-project-dist/hadoop-hdfs/TransparentEncryption.html Last access: 14/10/2020

[45] X. Yu, P. Ning, and M. A. Vouk, "Enhancing security of Hadoop in a public cloud," 6th Int. Conf. on Information and Communication Systems (ICICS), 2015.

[46] M. Xia, M. Saxena, M. Blaum, D. Pease, "A tale of two erasure codes in HDFS," FAST, pp. 213-226. 2015.

[47] N. Abdullah, A. Hakansson, and E. Moradian, "Blockchain based approach to enhance Big Data authentication in distributed environment," 9th Int. Conf. on Ubiquitous and Future Networks (ICUFN), 2017.

[48] Awaysheh, Feras, et al. "Big data security frameworks meet the intelligent transportation systems trust challenges." 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, 2019.

[49] Apache Knox, EST API and Application Gateway for the Apache Hadoop Ecosystem [Online]. Available: https://knox.apache.org/ Last access: 14/10/2020

[50] Awaysheh, Feras M., et al. "Poster: A pluggable authentication module for big data federation architecture." Proceedings of the 24th ACM Symposium on Access Control Models and Technologies. 2019.

[51] Apache Ranger, Comprehensive security managment for Enterprise Hadoop, [Online]. Available: https://ranger.apache.org Last access: 14/10/2020

[52] Apache Sentry, fine grained role based authorization, [Online]. Available: https://sentry.apache.org/ Last access: 14/10/2020

[53] N. Rakotondravony, B. Taubmann, W. Mandarawi, E. Weishäupl, P. Xu, B. Kolosnjaji, M. Protsenko, H. De Meer, and H. P. Reiser. "Classifying malware attacks in IaaS cloud environments." Journal of Cloud Computing 6.1 (2017): pp, 26.

[54] Beckers, Kristian, et al. "Pattern-based support for context establishment and asset identification of the ISO 27000 in the field of cloud computing." 2011 Sixth International Conference on Availability, Reliability and Security. IEEE, 2011.

[55] Krutz, R. L., and Vines, R. D. (2010). Cloud security: A comprehensive guide to secure cloud computing. Wiley Publishing.

[56] Apache Bigtop, comprehensive packaging, testing, and configuration of the Hadoop stack, [Online]. Available: https://https://bigtop.apache.org/ Last access: 14/10/2020

[57] Parmar, Raj R., et al. "Large-scale encryption in the Hadoop environment: Challenges and solutions." IEEE Access 5 (2017): 7156-7163.

[58] Alazab, Mamoun, et al. "A hybrid wrapper-filter approach for malware detection." Journal of networks 9.11 (2014): 2878-2891.

[59] Alazab, Mamoun, et al. "Malicious spam emails developments and authorship attribution." 2013 Fourth Cybercrime and Trustworthy Computing Workshop. IEEE, 2013.

[60] Tran, Khoi-Nguyen, Mamoun Alazab, and Roderic Broadhurst. "Towards a feature rich model for predicting spam emails containing malicious attachments and urls." (2014).

[61] Gupta, Maanak, et al. "An Attribute-Based Access Control for Cloud-Enabled Industrial Smart Vehicles." IEEE Transactions on Industrial Informatics (2020).

[62] Gupta, Maanak, et al. "Dynamic Groups and Attribute-Based Access Control for Next-Generation Smart Cars." U.S. Patent Application No. 16/811,165. 2020.

[63] Pooranian, Zahra, et al. "LEVER: Secure Deduplicated Cloud Storage with EncryptedTwo-Party Interactions in Cyber-Physical Systems." IEEE Transactions on Industrial Informatics (2020).

[64] Gupta, Maanak, Farhan Patwa, and Ravi Sandhu. "POSTER: Access control model for the Hadoop Ecosystem." Proceedings of the 22nd ACM on Symposium on Access Control Models and Technologies. 2017.

[65] Apache Eagle, open source security analytics, [Online]. Available: https://eagle.apache.org/ Last access: 14/10/2020
[66] Zeng, Xuezhi, et al. “SLA Management for Big Data Analytical Applications in Clouds: A Taxonomy Study.“ ACM Computing Surveys (CSUR) 53.3 (2020): 1-40.
[67] Johnson, Pontus, Robert Lagerstrm, and Mathias Ekstedt. “A meta language for threat modeling and attack simulations.“ Proceedings of the 13th International Conference on Availability, Reliability and Security. 2018.
[68] Aladwan, Mohammad N., et al. “TrustE-VC: Trustworthy Evaluation Framework for Industrial Connected Vehicles in the Cloud.“ IEEE Transactions on Industrial Informatics 16.9 (2020): 6203-6213.
[69] McDole, Andrew, Mahmoud Abdelsalam, Maanak Gupta, and Sudip Mittal. “Analyzing CNN Based Behavioural Malware Detection Techniques on Cloud IaaS.“ arXiv preprint arXiv:2002.06383 (2020).

Feras M. Awaysheh holds a PhD. in Big Data and Cloud Computing from the University of Santiago de Compostela, Spain. He obtained a BSc. Software Engineering degree from Al Balqa ’Applied University in 2008 and MSc. Degree from New York Institute of Technology (NYIT) With Honor in 2010, majoring in Information, Computer, and Network Security. Currently, he is an Assistant Professor in Data Systems at the University of Tartu, Estonia. Before that, Dr. Awaysheh worked as a researcher and Postdoc with the Centro Singular de Investigacin en Tecnoloxas Intelixentes (CiTIUS), USC Spain. Also, he held appointments in several research visits at Edinburgh University, UK, and the University of Charles Darwin, Australia. His main research interest includes large-scale distributed systems and Big Data analytics in general. Besides, developing and running software reliably in production for resource allocation (on-premises and cloud-based clusters), and middlewares for load-balancing and security solutions in HPC, Cloud, IoT, and Big Data deployment architectures.

Mohammad N. Aladwan obtained his BSc Software Engineering degree from Al Balqa ’Applied University in 2005 and a MSc Degree in Computer Networks and Security at Central Queensland University (Australia) in 2010. Most recently, he holds a Ph.D. from the University of Santiago de Compostela, CiTIUS research center. His main research interests include security in cloud environments and middleware for Cloud and Big Data.

Mamoun Alazab is an Associate Professor at the College of Engineering, IT and Environment at Charles Darwin University, Australia. He received his PhD degree in Computer Science from the Federation University of Australia, School of Science, Information Technology and Engineering. He is a cyber security researcher and practitioner with industry and academic experience. Alazab’s research is multidisciplinary that focuses on cyber security and digital forensics of computer systems with a focus on cybercrime detection and prevention. He has more than 200 research papers in many international journals and conferences. He is a Senior Member of the IEEE. He is the founding chair of the IEEE Northern Territory (NT) Subsection.

Sadi Alawadi holds a Ph.D. in Computer science/AI with honor in the Research Center in Intelligent Technologies (CiTIUS) of the University of Santiago de Compostela, Spain, in 2018. And a Master degree in Softcomputing and intelligence system, 2012, from Granada University - Spain. During the last two years he worked as a postdoc for the IOTAP Research Center at Malm University, Sweden. Currently, he is working at the Department of Information Technology, Division of Scientific Computing, Uppsala University. His main research lines include IOT systems, IoT middleware, End-User Development in the IOT, Machine learning, Deep learning, federated learning, transfer learning, Smart cities, and their related systems, Bigdata, Real-time analysis, Dimensionality reduction, and data visualization Context Awareness, and Blockchain.

Tomás F. Pena got his Ph.D. in Physics in 1994 at the University of Santiago de Compostela (USC) Spain and become an associate professor in the Department of Electronics and Computer Science, USC. Since 2010, he is a member of the Research Center in Intelligent Technologies (CiTIUS) at USC. His main research lines include Big Data and high-performance computing in general. Besides, the architecture of parallel systems, the development of parallel algorithms for clusters and supercomputers, the optimization of the performance in irregular codes and with sparse matrices, the prediction, and improvement of the performance of parallel applications in general, the development of applications and middleware for Grid and Cloud, and the use of Big Data technologies for scientific and NLP applications. Dr. Pena is an IEEE senior member and an Associate Editor of IEEE Transactions on Computers and IEEE Access journal.

José C. Cabaleiro got his Ph.D. in Physics in the University of Santiago de Compostela (Spain). From 1994 he is an associate professor in the area of Computer Architecture in the Department of Electronics and Computer Science in the University of Santiago de Compostela. From 2010, he is a member of the Research Center in Intelligent Technologies (CiTIUS) of this University. His main lines of interest in high performance computing include the development of efficient scientific applications, the architecture of parallel systems, parallel algorithms for irregular problems and with sparse matrices, prediction, and improvement of the performance of parallel applications, optimization of the memory hierarchy in irregular problems, and development of applications and middleware for Cloud and Big Data.