

Big Data Software Architectures: An Updated Review

Tiago Vinícius Remígio da Costa, Everton Cavalcante(✉), and Thais Batista

Federal University of Rio Grande do Norte, Natal, Brazil
everton.cavalcante@ufrn.br

Abstract. Big Data usually refers to the unprecedented growth of data and associated processes to gather, store, process, and analyze them to provide organizations and users with useful insights and information. The intrinsic complexity and characteristics of systems handling Big Data require software architectures as founded drivers for these systems to meet functional and quality requirements. In light of the relevant role of software architectures for Big Data systems, we investigate the current state of the art of Big Data software architectures. This paper presents the results of a systematic mapping study that updates existing literature reviews on this topic. We selected and analyzed 23 primary studies published in the last five years. We identified 11 architecture-related quality requirements and six architectural modules relevant to the design of software architectures for Big Data systems, besides analyzing whether existing proposals of reference architectures comply with these requirements and modules. We expect the results presented in this paper can provide a continuous update of the state of the art while highlighting essential concerns in the design of software architectures for Big Data systems.

Keywords: Software architecture · Big Data · Systematic mapping study

1 Introduction

The adoption of recent computing paradigms such as Cloud Computing, the Internet of Things (IoT), and 5G networks has been a major force to *Big Data*. This is an umbrella term usually referring to data sets whose size grows to an extent beyond the ability of traditional methods and tools to gather, store, process, and analyze the available data at a tolerable time [1,20]. Big Data is often defined by its characteristics, the so-called 5V's [2,12]: *Volume* (amount of data), *Velocity* (how quickly data are generated and transmitted), *Variety* (diversity of data types and sources), *Veracity* (quality and accuracy of data), and *Value* (what it is possible to do with data). The relevance of Big Data nowadays is noticeable: 96% of 85 interviewed companies from different domains

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
O. Gervasi et al. (Eds.): ICCSA 2022, LNCS 13375, pp. 477–493, 2022.
https://doi.org/10.1007/978-3-031-10522-7_33

manifested strongly investing in Big Data and Artificial Intelligence in 2021 [16]. Big Data systems (BDS) can indeed be found in many fields, providing valuable insights and information to organizations and users [1].

Software architectures are widely recognized as the backbone of any successful software-intensive system, thereby fundamentally contributing to achieving both business goals and quality requirements. This challenging scenario draws attention to designing Big Data software architectures. A software architecture for a BDS can aggregate different constituent elements related to functionalities such as data extraction, processing, integration, analysis, and visualization. More than this, decisions made at the software architecture level directly affect how these systems can meet quality attributes intrinsic to this context, such as scalability, availability, high performance, etc.

The relevance of BDS nowadays and the significant role of software architectures in meeting functional and quality requirements in these systems makes it worth investigating the literature to obtain a panorama of Big Data software architectures. This panorama can allow reflecting on the current state of the art and understanding the essential elements and quality requirements to consider when designing software architectures for BDS. The literature already provides secondary studies with this goal [1,20,21]. However, these existing works consider only primary studies published until 2017 and miss a time window of almost five years with recent work, despite adopting a systematic methodology.

To achieve the goal of investigating the current state of the art of the literature on Big Data software architectures while avoiding wasting time and effort with an unnecessary literature review, we grounded on a decision framework that helped us assess the need for updating existing secondary work on this topic [5,13]. After obtaining confidence that such an update would be worth it, we carried out a *systematic mapping study* (*systematic mapping*, in short) focusing on collecting, selecting, and analyzing primary studies on Big Data software architectures published since 2017. A systematic mapping is a form of secondary study that uses a systematic, rigorous procedure to obtain a comprehensive panorama of a research topic and identify research and development gaps from primary studies available in the literature [17]. This paper reports the main results of the performed systematic mapping with architecture-related requirements and modules relevant to software architectures for these systems and whether existing proposals of reference architectures to BDS encompass these elements. Our main intention with this work is that the presented results can contribute to both researchers and practitioners to design better software architectures for BDS.

The remainder of this paper is organized as follows. Section 2 discusses secondary studies related to this work. Section 3 describes the methodology to carry out the systematic mapping. Section 4 presents and discusses the results from the analysis of the relevant primary studies. Section 5 brings some conclusions.

2 Related Work

Kumar and Alencar [11] carried out a wide-scope secondary study on the interplay of Software Engineering activities in the development of BDS. After retriev-

ing results by combining manual and automated search using three electronic databases, these authors selected 170 primary studies covering the different phases of the software development life cycle. Architectural design was found in 40% of the selected studies, being acknowledged as an essential concern in building BDS. Nonetheless, this study is quite generic and did not specifically focus on Big Data software architectures.

To the best of our knowledge, Sena et al. [20] were pioneers in presenting the state of the art specifically related to Big Data software architectures. Their systematic mapping selected and analyzed 19 primary studies available in seven electronic databases, identifying eight architecture-related quality requirements and five modules regarded as essential to coping with the characteristics of BDS at the architectural level. Further work of these authors [21] specifically focused on investigating the use of well-known architectural patterns to design software architectures for BDS. From the set of studies selected in their previous work, they analyzed 11 studies that reported the use of architectural patterns for BDS. They concluded that the architectural patterns commonly used for BDS do not differ from those used in other contexts, and a layered model stood out as the most used pattern. The body of work analyzed by these two studies is now almost five years old since they considered primary studies published until mid-2017.

Avci et al. [1] carried out a systematic literature review aiming to investigate software architectures for BDS. They selected and analyzed 43 primary studies available in five electronic databases, identifying the domains in which Big Data software architectures have been applied, the application domains where they have been observed, and relevant quality attributes and architectural patterns considered. Despite having been published in 2020, these authors considered only primary studies published between 2010 and 2017, i.e., the study results are not up-to-date with the recent literature.

Rahman and Reza [19] and Montero et al. [14] recently carried out systematic mappings focusing on quality requirements relevant to BDS. The former study selected and analyzed 14 primary studies published between 2012 and 2019, retrieved from four electronic sources. The authors identified quality requirements and mapped them to the quality attributes defined in the ISO/IEC 25010:2011 International Standard [7], concluding that six of them (namely *performance efficiency*, *functional stability*, *reliability*, *security*, *usability*, and *scalability*) would be crucial to be considered in BDS design. The latter study selected and analyzed 67 primary studies published in the last decade (2010–2020), retrieved from two electronic sources and snowballing. The authors aimed to identify models proposed in the context of BDS to assess the quality of these systems. The set of analyzed studies reported 12 different quality models, and almost half of them focused on measuring the quality of data. The focus of both studies was on quality requirements, not explicitly relating them to the architectural level.

Our analysis of these systematic literature reviews and mapping studies served as driving motivations for this paper. The studies investigating the literature on Big Data software architectures covered only primary studies published until 2017, and recent work with new perspectives has been available since then.

Furthermore, recent concepts associated with BDS, such as *data lake* and *data platform*, were not considered even though they can impact architecture decisions and the overall quality of BDS. A data lake is a Big Data repository that stores raw data in their original, heterogeneous format with on-demand integration and enables users to query and explore these data [6]. A data platform is a cohesive infrastructure that provides organizations and users with several tools and utilities to develop, integrate, deploy, and manage Big Data applications and their data. Performing another literature review from scratch with no framed time window would not be efficient and worth of time and effort because the existing secondary studies adopted a rigorous, systematic methodology. Therefore, we opted to update one of these secondary studies considering the most recent literature published from 2017 to 2021.

3 Methodology

Before undertaking a systematic mapping, researchers need to ensure that the study is both needed and feasible. Kitchenham et al. [10] highlight that it is essential to consider whether the study (i) can produce outcomes that are likely to contribute to the knowledge about the addressed research topic and (ii) is distinct enough from already existing secondary studies in the same research topic, e.g., not comprehensively covering the research topic or being out of date.

This section describes the methodology adopted to carry out the systematic mapping. Section 3.1 grounds on a decision framework to assess the need for updating existing secondary work on Big Data software architectures. Section 3.2 presents the systematic mapping protocol in terms of research questions, search strategy, and selection criteria. Section 3.3 describes the process for selecting relevant primary studies.

3.1 The Need for an Updated Review

Updating a systematic mapping provides a continuous update of state of the art on a research topic and can identify how it evolves regarding the stability of existing research findings. Mendes et al. [13] point out that there have been no systematic proposals on whether and when to update systematic literature reviews and mapping studies in Software Engineering despite their increasing use in the last 15 years. These authors suggested using the Garner et al.'s decision framework [5] to guide the update (or not) of systematic literature reviews in the Software Engineering field. Figure 1 depicts the decision framework.

We followed Mendes et al. [13] on using the decision framework to check whether the existing secondary studies on Big Data software architectures would be worth updating. Table 1 summarizes the questions and answers to the framework for this work, which are provided in the following.

Currency of the Research Topic. Big Data software architectures have been evolving through the years, mainly due to the growth of data to handle (volume) and how fast they should be available (velocity) for analysis and predictions. Therefore, software architectures for these systems should have components able to address and balance the 5V's characteristics. Existing secondary

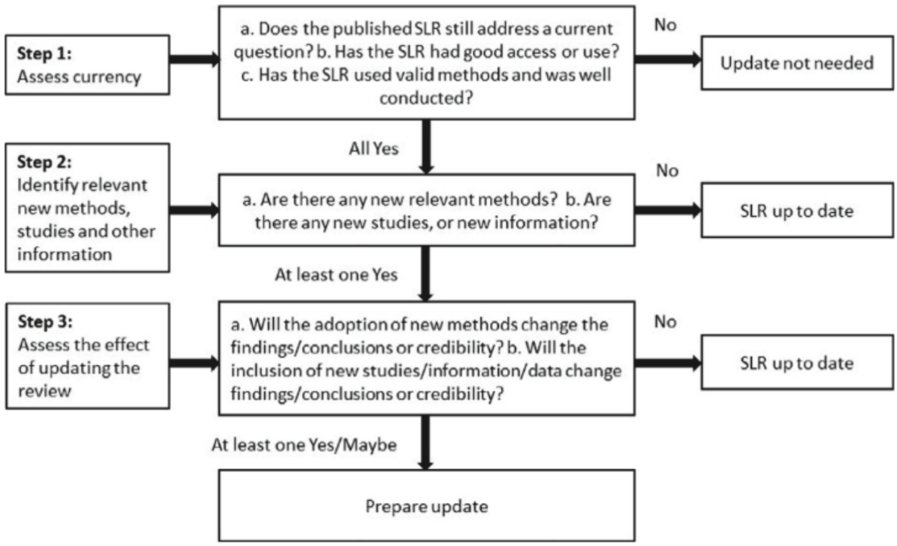

Fig. 1. The decision framework to assess the need for updating secondary studies [13].

studies on Big Data software architectures (see Sect. 2) have been all carried out by following valid systematic methods.

Relevant New Methods, Studies, and Other Information. BDS have evolved to comply with technological advances and data governance regulations. Early Big Data concerned distributed processing in enterprise data centers (e.g., using Hadoop/MapReduce), whereas there has been a recent technology shift to cloud-based solutions. Furthermore, the enforcement of regulations worldwide (e.g., GDPR in Europe, CCPA in the USA, and LGPD in Brazil) implies considering constraints posed by these legal bases in designing, developing, and deploying secure BDS. Therefore, BDS are no longer just about data ingestion, processing, storage, and visualization.

Effect of Updating the Review. The inclusion of new methods and requirements changes how software architectures or reference architectures are designed to settle new aspects. New modules and new quality requirements can impact how a Big Data software architecture is designed.

3.2 Review Protocol

Once the need for updating the existing secondary work on Big Data software architecture has been identified, we followed consolidated guidelines available in the literature [18] to carry out our systematic mapping. We first devised a protocol with research questions to answer, the adopted search strategy, the criteria to select studies, and the data extraction and synthesis methods. Next, we searched for, selected, and analyzed the primary studies according to the

Table 1. Questions and answers to the Garner et al.'s framework [5] for this work.

Questions	Answers
Does the published SLR still address a current question?	Yes
Has the SLR good access or use?	Yes
Has the SLR used valid methods and was well conducted?	Yes
Are there any new relevant methods?	No
Are there any new studies, or new information?	Yes
Will the adoption of new methods change the findings/conclusions or credibility?	Maybe
Will the inclusion of new studies/information/data change the findings/conclusions or credibility?	Maybe

SLR = systematic literature review

protocol. At last, we outlined conclusions by aggregating information extracted from the relevant primary studies upon the research questions.

Research Questions. This study aimed to investigate the recent state of the art of the literature on the design of software architectures for BDS. We grounded on the Sena et al.'s work [20] since it was the pioneering secondary study on Big Data software architectures. We considered the following research questions (RQs):

RQ1: What are the main requirements for building software architectures for BDS?
RQ2: What are the main modules/components in software architectures of BDS?
RQ3: What requirements and modules are found in existing proposals of reference architectures for BDS?

Search Strategy. We used an automated search process to retrieve primary studies with the potential to answer the posed RQs. We considered six well-known electronic publication databases with good coverage of the literature, namely IEEEXplore, ACM Digital Library, ScienceDirect, Scopus, Web of Science, and Enginering Village. Two primary keywords were initially identified from the defined RQs: *Big Data* and *software architecture*. To increase coverage of the literature search, we have built the following search string:

```
(big data OR data lake OR data platform) AND
(software architecture OR reference architecture OR reference model)
```

Selection Criteria. We used selection criteria to include studies potentially relevant to answering the posed RQs and exclude those that would not contribute to answering them. We have preserved the two inclusion criteria (ICs) and the

five exclusion criteria (ECs) from Sena et al. [20]. However, we added EC6 to exclude studies published before 2017 since our work is an updated review:
Inclusion criteria:

IC1: The primary study addresses software architectures in the context of BDS.
IC1: The primary study proposes a reference architecture or a reference model for BDS.

Exclusion criteria:

EC1: The primary study is a table of contents, tutorial description, or a summary of a conference/workshop.
EC2: The primary study is written in a language other than English.
EC3: The study is a previous version of a more complete one on the same research or of the same authors.
EC4: The full text of the primary study is not available.
EC5: The primary study is out of scope of the systematic mapping.
EC6: The primary study was published before 2017.

3.3 Selection Process

We carried out the systematic mapping in the last quarter of 2021, thus considering studies published so far. We used the Parsifal[1] Web tool to support all activities of searching, filtering, selecting, and extracting data from the retrieved studies. The automated search process considered adaptations to the base search string (see Sect. 3.2) according to the specificities of each publication electronic database and the title, abstract, and keyword fields.

Figure 2 depicts the process for selecting the relevant primary studies. The automated search procedure retrieved 1,378 results, of which 730 were removed as duplicates, i.e., studies retrieved by more than one publication database. The remaining 648 unique studies underwent a selection process in two stages. The first stage aimed to apply the selection criteria (ICs and ECs) established in the systematic mapping protocol to the title, abstract, and keywords of the retrieved results, resulting in 32 studies. Next, we fully read the study and applied the selection criteria to check if it could answer any RQ. This resulted in a final set of 23 studies relevant to this work. The selected primary studies are listed in the Appendix, identified as S1 to S23.

4 Results and Discussion

This section summarizes and discusses the results of the systematic mapping considering the RQs described in Sect. 3.2 and the data extracted and synthesized from the analyzed primary studies. Section 4.1 gives an overview of the selected primary studies. Sections,4.2 to 4.4 present the answers to each RQ.

[1] https://parsif.al/.

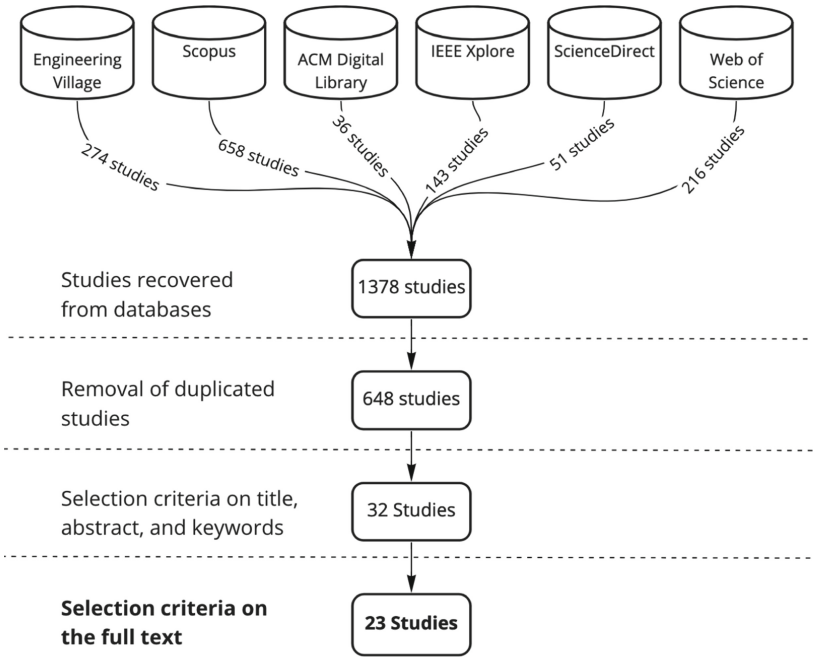

Fig. 2. Process for selecting the relevant primary studies.

4.1 Overview of the Selected Studies

Distribution of the Studies over the Years. Figure 3a shows the distribution of the publication of studies on software architectures for BDS. We observed an average of 4.6 studies (median = 4) published per year.

Publication Venues. Figure 3b illustrates the types of venues in which the selected primary studies were published. Most of the studies were published in international conference proceedings, possibly indicating a minor degree of maturity for the proposals. Only one study (S20) was published in a venue directly related to the Software Architecture field. In contrast, most studies were published in conferences and journals focusing on Big Data and its applications. The studies spread over different venues, with no one standing out among them.

Application Domains. Figure 4 shows the application domains addressed by the selected primary studies. We noticed that most of the selected studies are related to Big Data infrastructure, i.e., with no specific application domain.

4.2 Quality Requirements

RQ1 aimed to identify the main quality requirements to be considered in designing software architectures for BDS. Sena et al. [20] elicited eight requirements and associated them to the *Systems and software Quality Requirements and*

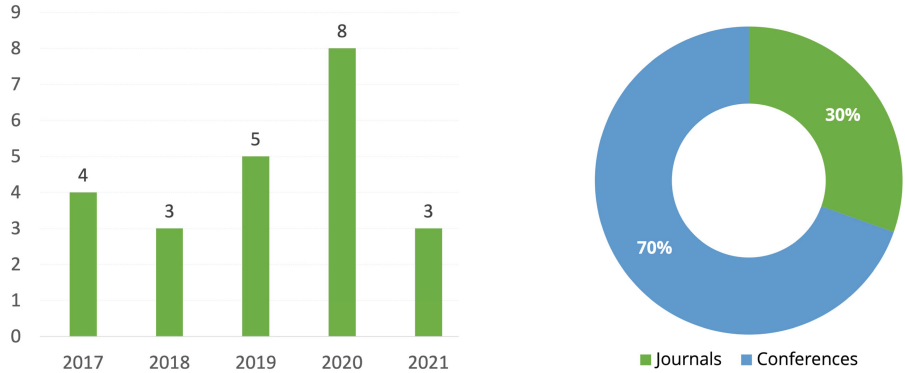

Fig. 3. Distribution of the selected primary studies per publication year (a) and type of publication venue (b).

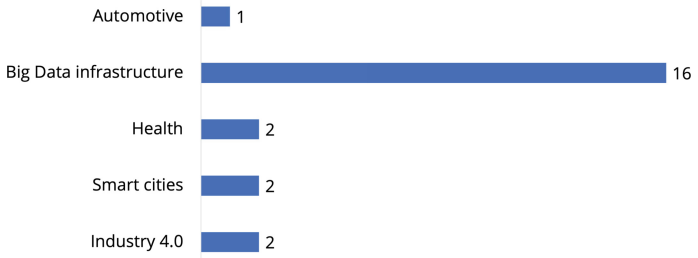

Fig. 4. Application domains addressed by the selected primary studies.

Evaluation (SQuaRE) model defined in the ISO/IEC 25010:2011 International Standard [7]. Our findings confirm the relevance of these quality requirements in the selected primary studies. These eight requirements are:

1) **Scalability:** A BDS should be able to scale storage and computational resources to deal with growing data.
2) **High performance:** A BDS should deal with different needs to process and serve data, supporting batch and streaming processing.
3) **Modularity:** A BDS software architecture should be composed of several independent, loosely coupled components.
4) **Consistency:** Data must be retrieved in the same format and compatible with previous data [22].
5) **Security:** A BDS should protect information and data so that people or other products or systems have the degree of data access appropriate to their types and levels of authorization.
6) **Real-time operation:** A BDS should be able to process data in real-time, using streaming processing to provide data when time-serving constraint is required.



Fig. 5. Architecture-related quality requirements for BDS in the selected studies.

7) **Interoperability:** A BDS should be implemented by components that can exchange information and use the information exchanged.
8) **Avaliability:** A BDS should be operational and accessible for use, usually applying fault-tolerant mechanisms, redundancy, and other strategies.

Our analysis of the selected 23 primary studies led to the identification of three additional requirements:

9) **Usability:** A BDS should be easy to use and extend as many different stakeholders can produce and consume data, and they should know how to operate the system.
10) **Suitability:** BDS should be functional according to requirements stated in architecture design, and data should comply with the predefined constraints.
11) **Maintainability:** A BDS should be easily changed to evolve, correct, and be adaptable according to data characteristics.

Figure 5 shows the 11 architecture-related quality requirements for BDS in the selected primary studies. We observed that the most frequent requirements are *scalability*, *security*, *availability*, and *maintainability*, appearing in more than 40% of the studies. Scalability is an essential requirement since volume is the most basic characteristic of Big Data. Consequently, BDS are inherently expected to scale ingestion, processing, and storage to accommodate the large volumes of data. The availability of a BDS concerns making the system operational and its data ready to use. It can be achieved through replication, horizontal scaling, distributed cluster nodes, and disaster response (e.g., studies S3 and S9). Security is also widely mentioned in terms of related characteristics such as authenticity, confidentiality, and integrity, which are required to govern data accordingly as legal regulations apply to personal data. Finally, the separation of concerns through modules and storage layers increases maintainability, which is important for managing modifications and preserving the BDS stability.

4.3 Architectural Modules

RQ2 aimed to identify modules/components common to software architectures for BDS. We mapped six architectural modules, being *Data Governance* found in addition to the five other architectural modules identified by Sena et al. [20]:

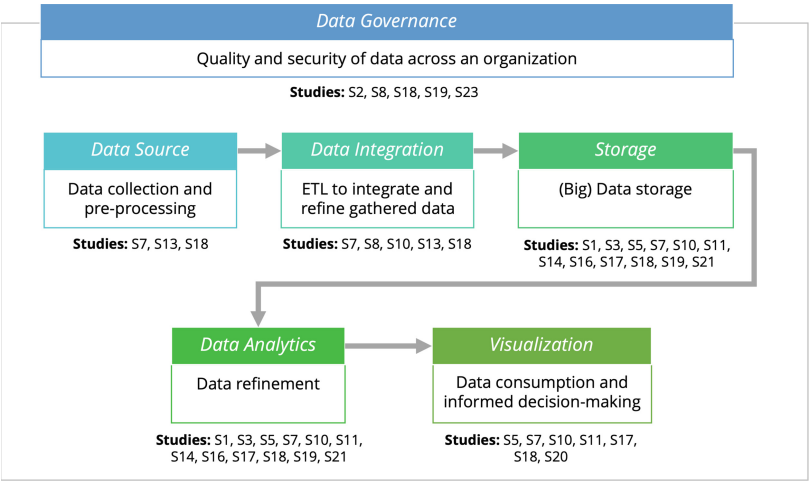

Fig. 6. Architectural modules in software architectures for BDS.

1) **Data Source:** Processes where data are captured, collected, or actively sent to a BDS. In practice, this can be done by capturing (relational) databases, streaming data to the BDS storage, or even retrieving data through API calls.
2) **Data Integration:** After ingesting sources, data must be transformed from a raw, brute format through ETL (Extract, Transform, and Load) processes to a refined state that helps data analysts.
3) **Storage:** Storage technologies can be adopted in a Big Data software architecture for different purposes.
4) **Data Analytics:** Refinement of processed data.
5) **Visualization:** Data consumption and decision-making aligned with business-related roles.
6) **Data Governance:** Processes and responsibilities to ensure the quality and security of the data used across an organization.

Identifying these architectural modules is relevant to design better software architectures for BDS, including the conception of a reference architecture for these systems. Figure 6 depicts the relationships among these modules, which can be organized into a pipeline structure encompassing the generation, processing, analysis, and consumption of data. It is also worth observing that *Data*

Governance represents a crosscutting concern that impacts the other modules, clearly defining roles and data ownership levels and policies.

Study S2 argues that storage and governance modules precede the creation of any further resources in a data-centric architecture like the ones driving BDS, where data assets play a prominent role. Moreover, data governance defines metadata and policies enabling appropriate access control, use, and administration, considering not only data but also components where data flow [8,9].

4.4 Reference Architectures

Reference architectures can be understood as abstract architectures encompassing knowledge and experiences in a given application domain, thus being able to facilitate and guide the development, standardization, interoperability, and evolution of software systems in such a domain [3,15]. Establishing reference architectures for BDS can define essential building blocks and design choices for constructing architectural solutions that can adequately meet functional and non-functional requirements. Therefore, directions provided by a reference architecture are essential elements to guide and facilitate BDS development.

RQ3 specifically investigated studies addressing reference architectures for BDS. We have found 17 studies concerning this topic and attempted to map how the previously identified architectural-related quality requirements and modules are present in those reference architectures. We noticed that the *Data Analytics* ($n = 9$), *Storage* ($n = 5$), and *Data Governance* ($n = 5$) modules were presented as the main building blocks in reference architectures for BDS. Seven studies have detailed scalability as an essential quality requirement to be encompassed by reference architectures since software architectures for BDS must be scalable to support the high volume and velocity of data. Security is also considered an essential quality requirement for reference architectures in seven studies, suggesting a relationship with the *Data Governance* module.

Our results partially confirm the ones of Sena et al. [20] regarding the essential architectural modules in reference architectures for BDS, except for the newly *Data Governance* module. However, the security requirement seems to have gained more attention in recent years when observing the previously existing results. In the previous systematic mapping, 21.03% (4/19) of the studies were found addressing this requirement in 2017, against 30.43% (7/23) now. This increase may be a consequence of the enforcement of major regulations on data, which became effective only in the last two to four years (GDPR in Europe in 2018, CCPA in the USA and LGPD in Brazil in 2020). Furthermore, data privacy and security have been a genuine concern because of recent data breaches, misuses, and other threats to companies and people.

We also noticed few studies with at least four architectural modules in the proposed reference architectures, thus meaning that these proposals do not fully meet the common relevant modules for software architectures tailoring BDS. Moreover, there are studies (e.g., S7) in which the *Data Integration* and *Storage* modules are bound as a single module, probably as a consequence of technology choice. These discrepancies regarding the architectural modules in the existing

proposals of reference architectures for BDS and how these elements are organized may point to a lack of maturity and failure in following a systematic design process, as recently observed by Garcés et al. [4].

5 Conclusion

This paper presented the results of a systematic mapping to investigate the current state of the art on how Big Data has been addressed at the architectural level. This systematic mapping updates a previous secondary study from mid-2017 [20] on software architectures for BDS, covering primary studies published in the last five years. We have systematically analyzed 23 studies available in the literature to (i) identify essential quality requirements for the design of BDS software architectures, (ii) define the main modules composing these architectures, and (iii) investigate how existing reference architectures comply with those requirements and modules.

Our analysis resulted in some findings that update the state of the art on software architecture and reference architectures for BDS. This work confirms previous findings on essential eight architecture-related quality requirements, with particular attention to scalability and availability due to the intrinsic nature of Big Data and security as a consequence of recent regulations on the use of data. Three new quality requirements were identified (usability, suitability, and maintainability) as also relevant to the design of software architectures for BDS.

We also identified six architectural modules relevant to design better software architectures for BDS. Five of these modules were identified in previous work [20], and we included *Data Governance* as a new one as referring to processes and responsibilities to ensure the quality and security of the data used across an organization. These architectural modules can also be a foundation for the conception of reference architectures for BDS, a topic we investigated in this work. Most of the primary studies concerned reference architectures for BDS, but we noticed that these proposals do not encompass all the elicited quality requirements and architectural modules in Big Data software architectures. We also had an insight into the lack of a systematic process guiding the design of existing reference architectures, an issue to be addressed in future research.

Appendix

[S1] Sfaxi, L., Ben Aissa, M.M.: Babel: A generic benchmarking platform for Big Data architectures. Big Data Research **24** (2021). http://doi.org/10.1016/j.bdr.2021.100186

[S2] López Martínez, P., Dintén, R., Drake, J.M., Zorrilla, M.: A Big Data-centric architecture metamodel for Industry 4.0. Future Generation Computer Systems **125**, 263–284 (2021). http://doi.org/10.1016/j.future.2021.06.020

[S3] Shah, S.A., Seker, D.Z., Rathore, M.M., Hameed, S., Yahia, S.B., Draheim, D.: Towards disaster resilient smart cities: Can Internet of Things and Big Data analytics be the game changers? IEEE Access **7**, 91885–91903 (2019). http://doi.org/10.1109/access.2019.2928233

[S4] Wahyudi, A., Kuk, G., Janssen, M.: A process pattern model for tackling and improving Big Data quality. Information Systems Frontiers **20**, 457–469 (2018). http://doi.org/10.1007/s10796-017-9822-7

[S5] Quinde, C., Guillermo, D., Siguenza-Guzman, L., Orellana, D., Pesántez-Cabrera, P.: A software architecture proposal for a data platform on active mobility and urban environment. In: Morales, G.R., C., E.R.F., Salgado, J.P., Pérez-Gosende, P., Cordero, M.O., Berrezueta, S. (eds.) Information and Communication Technologies, Communications in Computer and Information Science, vol. 1307, pp. 501–515. Springer Nature Switzerland AG (2020). http://doi.org/10.1007/978-3-030-62833-8_37

[S6] Moreno, J., Serrano, M.A., Fernández-Medina, E., Fernández, E.B.: Towards a security reference architecture for Big Data. In: Proceedings of the 20th International Workshop on Design, Optimization, Languages and Analytical Processing of Big Data co-located with 10th EDBT/ICDT Joint Conference, CEUR Workshop Proceedings, vol. 2062. CEUR-WS, Germany (2018)

[S7] Sang, G.M., Xu, L., de Vrieze, P.: Simplifying Big Data analytics systems with a reference architecture. In: Camarinha-Matos, L.M., Afsarmanesh, H., Fornasiero, R. (eds.) Collaboration in a Data-Rich World, IFIP Advances in Information and Communication Technology, vol. 506, pp. 242–249. IFIP, Switzerland (2017). http://doi.org/10.1007/978-3-319-65151-4_23

[S8] Awaysheh, F.M., Aladwan, M.N., Alazab, M., Alawadi, S., Cabaleiro, J.C., Pena, T.F.: Security by design for Big Data frameworks over Cloud Computing. IEEE Transactions on Engineering Management (2020). http://doi.org/10.1109/tem.2020.3045661

[S9] Garises, V., Quenum, J.: The road towards Big Data infrastructure in the health care sector: The case of Namibia. In: Proceedings of the 19th IEEE Mediterranean Electrotechnical Conference. pp. 98–103. IEEE, USA (2018). http://doi.org/10.1109/melcon.2018.8379075

[S10] Henning, S., Hasselbring, W., Möbius, A.: A scalable architecture for power consumption monitoring in industrial production environments. In: Proceedings of the 2019 IEEE International Conference on Fog Computing. pp. 124–133. IEEE, USA (2019). http://doi.org/10.1109/icfc.2019.00024

[S11] Aissa, M.M.B., Sfaxi, L., Robbana, R.: Decisional architectures from Business Intelligence to Big Data: Challenges and opportunities. In: Proceedings of the 2nd International Conference on Digital Tools & Uses Congress. ACM, USA (2020). http://doi.org/10.1145/3423603.3424049

[S12] Haroun, A., Mostefaoui, A., Dessables, F.: A Big Data architecture for automotive applications: PSA Group deployment experience. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster,

Cloud and Grid Computing. pp. 921–928. IEEE, USA (2017). http://doi.org/10.1109/ccgrid.2017.107

[S13] Joaquim, J.L.M., dos Santos Mello, R.: An analysis of confidentiality issues in data lakes. In: Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services. pp. 168–177. ACM, USA (2020). http://doi.org/10.1145/3428757.3429109

[S14] Cuzzocrea, A., Loia, V., Tommasetti, A.: Big-Data-driven innovation for enterprises: Innovative Big Value Paradigms for next-generation digital ecosystems. In: Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics. ACM, USA (2017). http://doi.org/10.1145/3102254.3102271

[S15] Zhu, Y., Liu, H.: Research on data-driven models of teaching quality evaluation. In: Proceedings of the 2020 International Conference on Computer Communication and Information Systems. pp. 10–14. ACM, USA (2020). http://doi.org/10.1145/3418994.3419001

[S16] Kaufmann, M.: Big Data Management Canvas: A reference model for value creation from data. Big Data and Cognitive Computing **3**(1) (2019). http://doi.org/10.3390/bdcc3010019

[S17] Iglesias, C.A., Favenza, A., Álvaro Carrera: A Big Data reference architecture for emergency management. Information **11**(12) (2020). http://doi.org/10.3390/info11120569

[S18] Arass, M.E., Ouazzani-Touhami, K., Souissi, N.: Data life cycle: Towards a reference architecture. International Journal of Advanced Trends in Computer Science and Engineering **9**(4), 5645–5653 (2020). http://doi.org/10.30534/ijatcse/2020/215942020

[S19] Xiaofeng, L., Jing, L.: Research on Big Data reference architecture model. In: Proceedings of International Conference on Artificial Intelligence and Big Data. pp. 205–209. IEEE, USA (2020) http://doi.org/10.1109/icaibd49809.2020.9137451

[S20] Cerezo, F., Cuesta, C.E., Moreno-Herranz, J.C., Vela, B.: Deconstructing the Lambda Architecture: An experience report. In: 2019 IEEE International Conference on Software Architecture Companion. pp. 196–201. IEEE, USA (2019). http://doi.org/10.1109/icsa-c.2019.00042

[S21] Garises, V., Quenum, J.: An evaluation of Big Data architectures. In: Proceedings of the 8th International Conference on Data Science, Technology and Applications. pp. 152–159. SciTePress, Portugal (2019). http://doi.org/10.5220/0007840801520159

[S22] Salavati, H., Gandomani, T.J., Sadeghi, R.: A robust software architecture based on distributed systems in Big Data health care. In: Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics. pp. 1701–1705. IEEE, USA (2017). http://doi.org/10.1109/icacci.2017.8126088

[S23] Giebler, C., Gröger, C., Hoos, E., Schwarz, H., Mitschang, B.: A zone reference model for enterprise-grade data lake management. In: Proceedings of the IEEE 24th International Enterprise Distributed Object Comput-

ing Conference. pp. 57–66. IEEE, USA (2020). http://doi.org/10.1109/edoc49727.2020.00017

References

1. Avci, C., Tekinerdogan, B., Athanasiadis, I.N.: Software architectures for Big Data: a systematic literature review. Big Data Anal. **5** (2020). https://doi.org/10.1186/s41044-020-00045-1
2. Beyer, M., Laney, D.: The importance of 'Big Data': A definition. Technical report, Gartner, USA, June 2012
3. Cloutier, R., Muller, G., Verma, D., Nilchiani, R., Hole, E., Bone, M.: The concept of reference architectures. Syst. Eng. **13**(1), 14–27 (2010). https://doi.org/10.1002/sys.20129
4. Garcés, L., et al.: Three decades of software reference architectures: a systematic mapping study. J. Syst. Softw. **179** (2021). https://doi.org/10.1016/j.jss.2021.111004
5. Garner, P., et al.: When and how to update systematic reviews: Consensus and checklist. BMJ **354** (2016). https://doi.org/10.1136/bmj.i3507
6. Hai, R., Geisler, S., Quix, C.: Constance: an intelligent data lake system. In: Proceedings of the 2016 International Conference on Management of Data, pp. 2097–2100. ACM, USA (2016). https://doi.org/10.1145/2882903.2899389
7. ISO/IEC 25010: Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. ISO, Switzerland (2011)
8. Janssen, M., Brous, P., Estevez, E., Barbosa, L.S., Janowski, T.: Data governance: Organizing data for trustworthy Artificial Intelligence. Govern. Inf. Q. **37**(3) (2020). https://doi.org/10.1016/j.giq.2020.101493
9. Kim, H.Y., Cho, J.S.: Data Governance Framework for Big Data implementation with a case of Korea. In: Proceedings of the 2017 IEEE International Congress on Big Data (2017). https://doi.org/10.1109/bigdatacongress.2017.56
10. Kitchenham, B.A., Budgen, D., Brereton, P.: Evidence-Based Software Engineering and systematic reviews. Chapman and Hall/CRC Press, USA (2016)
11. Kumar, V.D., Alencar, P.: Software Engineering for Big Data projects: domains, methodologies and gaps. In: Proceedings of the 2016 IEEE International Conference on Big Data, pp. 2886–2895. IEEE, USA (2016). https://doi.org/10.1109/bigdata.2016.7840938
12. Laney, D.: 3D data management: Controlling Data Volume, Velocity, and Variety. Technical report, META Group, USA, February 2001
13. Mendes, E., Wohlin, C., Felizardo, K., Kalinowski, M.: When to update systematic literature reviews in Software Engineering. J. Syst. Softw. **167** (2020). https://doi.org/10.1016/j.jss.2020.110607
14. Montero, O., Crespo, Y., Piatini, M.: Big data quality models: a systematic mapping study. In: Paiva, A.C.R., Cavalli, A.R., Ventura Martins, P., Pérez-Castillo, R. (eds.) QUATIC 2021. CCIS, vol. 1439, pp. 416–430. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-85347-1_30
15. Nakagawa, E.Y., Oquendo, F., Maldonado, J.C.: Reference architectures. In: Oussalah, M.C. (ed.) Software Architecture 1, pp. 55–82. ISTE/John Wiley & Sons Inc., United Kingdom (2014). https://doi.org/10.1002/9781118930960.ch2

16. NewVantage: Big Datra and AI Executive Survey 2021: Executive summary of findings. techreport, NewVantage Partners LLC, USA, January 2021
17. Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in Software Engineering. In: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, pp. 68–77. British Computer Society, United Kingdom (2008). https://doi.org/10.14236/ewic/ease2008.8
18. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in Software Engineering: an update. Inf. Softw. Technol. **64**, 1–18 (2015). https://doi.org/10.1016/j.infsof.2015.03.007
19. Rahman, M.S., Reza, H.: Systematic mapping study of non-functional requirements in Big Data system. In: Proceedings of the 2020 IEEE International Conference on Electro Information Technology, pp. 025–031. IEEE, USA (2020). https://doi.org/10.1109/eit48999.2020.9208288
20. Sena, B., Allian, A.P., Nakagawa, E.Y.: Characterizing Big Data software architectures: a systematic mapping study. In: Proceedings of the 11th Brazilian Symposium on Software Components, Architectures, and Reuse. ACM, USA (2017). https://doi.org/10.1145/3132498.3132510
21. Sena, B., Garcés, L., Allian, A.P., Nakagawa, E.Y.: Investigating the applicability of architectural patterns in Big Data systems. In: Proceedings of the 25th Conference on Pattern Languages of Programs. ACM, USA (2018)
22. Wahyudi, A., Kuk, G., Janssen, M.: A process pattern model for tackling and improving big data quality. Inf. Syst. Front. **20**(3), 457–469 (2018). https://doi.org/10.1007/s10796-017-9822-7