

The state of big data reference architectures: a systematic literature review

Abstract

This template helps you to create a properly formatted L^AT_EX manuscript.

Keywords: `elsarticle.cls`, L^AT_EX, Elsevier, template

2010 MSC: 00-01, 99-00 s

1. Introduction

The rapid development of software technologies, the proliferation of digital devices and networking infrastructure of today, have by and large, augmented user's capability to generate data [1]. In the age of information, users are
5 unceasing generators of structured, semi-structured, and unstructured data that if collected and crunched correctly, may reveal game-changing patterns [2].

The unprecedented proliferation of data have emerged a new ecosystem of technologies; one of these ecosystems is big data (BD)[3]. BD is a term emerged to describe large amount of data that comes in various forms from different
10 channels. Within the years, BD has attained a lot of attention from academia and industry, and many strive to benefit from this new material. Howbeit, adopting BD requires the absorption of great deal of complexity and many traditional systems cannot cope with characteristics of this domain.

A recent survey published by Databricks in partnership with MIT Technol-
15 ogy Review Insights, stated that only 13% of companies excel at delivering on their data strategy [4]. In the same vein, Vintage Partners highlighted that only 24% of companies have successfully adopted BD [5]. Sigma computing report presented that 1 in 4 business experts have given up on getting insights they needed because the data processing took too long [6]. Moreover, Gartner

20 approximated that only 20% of companies have successfully adopted BD.

Some of the most highlighted challenges of BD is 'lack of business context', 'organizational challenges', 'BD architecture', 'data engineering', 'rapid technology change', and 'lack of talent' [7]. Whereas similar issues may exist in other domains, it is exacerbated when it comes to BD systems. This is due the
25 inherent complexity of BD engineering, the need for real-time processing, the scalability requirement of these systems, and the sensitivities around data.

Today, majority of BD systems are designed underlying ad-hoc and complicated architectural solutions [8], that do not seem to adhere to similar patterns. This will challenge software architects to design a suitable solution for any given
30 context, creates a foundation for an immature architectural decision, and does not promote the growth and development of BD systems as a whole.

Therefore, since the approach of ad-hoc design to BD systems is undesirable and leaves many engineers in the dark, there is a need for more software engineering research for BD systems. To this end, this study presents a systematic
35 literature review (SLR) on BD (BD) reference architectures (RAs).

2. Why reference architectures?

Conceptualization of the system as an RA, helps with understanding of the system's key components, behavior, composition and evolution of it, which in turn affect quality attributes such as maintainability, scalability and performance [9]. Therefore RAs can be a good standardization artefact and a communication medium that not only results in concrete architectures for BD systems, but also provide stakeholders with unified elements and symbols to discuss and progress BD projects.
40

This approach to system development is not new to practitioners of complex system. In software product line (SPL) development, RAs are utilized as generic artifacts that are instantiated and configured for a particular domain of systems [10]. In software engineering, IT giants like IBM have referred to RAs as the 'best of best practices' to address complex and unique system design challenges
45

[9]. In other international standardization, RAs have been repeatedly used to
50 standardize an emerging domain, a good example of this is BS ISO/IEC 18384-1
RA for service oriented architectures [11].

3. State of the art

Despite the undeniable benefits of RAs, and their potential to solve some of
the complex issues of BD systems, we think that this area is underdeveloped and
55 needs more attention from both academia and practice. This insight is derived
from our preliminary systematic review in academia, and a search for available
big data RAs ([2]).

To the best of our knowledge, one of the most comprehensive BD RA pub-
lished, is the National Institute of Standards and Technology (NIST) BD RA.
60 This RA is published by Big Data Public Working Group (NBD-PWG) with
large set of contributors from academia, industry, non-profit organizations,
agents, and government representatives. This was announced as an initiative
from White house in March 2012, and the the RA was published under the title
'NIST Big Data Interoperability Framework: Volume 6, Reference Architecture'
65 in October 2019.

Given the substantial investment on BD RAs, one might infer the value of
these artifacts, and this can in turn highlights the necessity for more research
in this domain. Another factor that worths mentioning is how vaguely the
phrase 'reference architecture' is defined and institutionalized. For instance,
70 the difference between a 'concrete architecture' and an RA is hardly discussed,
and different domains seem to have defined the artifact slightly differently. For
instance, Cloutier et al ([9]) defined RAs as 'Reference Architectures capture the
essence of existing architectures, and the vision of future needs and evolution
to provide guidance to assist in developing new system architectures'. This
75 definition is derived from the system engineering domain and by the means of
collaborative forum from Steven's institute of technology.

In another effort, Muller et al ([12]) defines RA as 'artifacts that captures

the essence of architecture of a collection of systems. This definition is driven from the product line engineering domain'. Moreover, the difference between
80 RAs and concrete architectures is rarely discussed. Another definition by Bass et al ([13]) stated that 'A reference architecture is a reference model mapped onto software elements (that cooperatively implement the functionality defined in the reference model) and the data flows between them'.

Angelov et al ([14]) defined RAs proposed that 'A reference architecture is a
85 generic architecture for a class of information systems that is used as a foundation for the design of concrete architectures from this class'. Although different authors may have defined RAs with different syntax, the essence remains the same: to reuse the software engineering knowledge for a class of systems, particularly in relation to architecture.

90 Given the failure rate of BD projects, we posit RAs as potential solution to facilitate system development and BD architecture, and aim to explore this area through a systematic literature review. Up to date, there's only one SLR that explored this area ([2]), which is outdated, suffers from methodological clarity, and is published as a conference paper, which implies lack of detail.

95 Based on this, the objective of this review is to find and collate the BD RAs available from the body of evidence, highlight their architectural commonality and point out the limitations. This study can be considered a useful primer for practitioners or academics who are interested in partaking in a BD project.

The research questions are formulated as the following;

- 100
1. What are current BD RAs available in academia and industry?
 2. What are major architectural components of these BD RAs?
 3. What are the limitations of current BD RAs?

4. Review Methodology:

This research follows the guidelines of PRISMA ([15]). In addition, we
105 adopted PRISMA-S ([16]) to improve our search strategy and lastly we have used Barbara et al's guidelines for evidence based software engineering and

systematic reviews [17]. Although PRISMA is a comprehensive guidelines on conducting a systematic literature review, it is derived from the healthcare community and sometimes makes assumptions that may not be relevant to software engineering and information system researchers. Barbara et al [17] have translated many of these assumptions to the domain of software engineering and have included many guidelines for lone researchers and projects with small number of researchers.

We have therefore utilized PRISMA as the underpinning of our research design, with complementary studies to reduce bias, improve transparency and systematicity. SLR has been chosen because it is a qualitative research methodology that is aimed at driving knowledge and understanding about the subject matter and the elements around it. Besides, SLR provides a transparent and reproducible procedure that elicits patterns, relationships, trends, and delineates the overall picture of the subject [18].

The main objective of this study is to assess the current state of BD RAs, identify their major architectural components, point out fundamental concepts and discuss their limitations. This objective is achieved in four phases. In first phase, research questions are stated, literature are identified and pooled, and exclusion and inclusion criteria are defined. In second phase, literatures are assessed for their quality based on inclusion/exclusion criteria and relevance to research questions. Thirdly, selected pool of literature is coded based on research questions. Lastly, findings are synthesized by the means of thematic synthesis, and themes realized are depicted.

This study builds on the SLR conducted by Ataei et al [2] and aims to improve it by covering the years 2020 to 2022. Unlike Ataei's work, this paper aims to employ thematic synthesis, and provide a more detailed view of BD RAs and their properties.

4.1. Identification

The first phase of the SLR began, by adoption of PRISMA-S ([16]) to develop a robust multi-database search strategy. This extension of PRISMA provided

us with a framework of 12 items to increase transparency, systematicity, and reduce bias.

For the purposes of this study, following electronic databases were search:
140 ScienceDirect, IEEE Explore, SpringerLink, AISEL, JSTOR and ACM library. To pursue to goal of finding all literature available on the topic, and to avoid overlooking valuable research, abstract and citation databases and search engines such as Google Scholar, and Research Gate was used.

We also searched the grey literature on the topic, using the search string
145 "big data" AND "reference architecture*" on Google (in June 2022). The first 40 results were selected for screening. This was done in 'incognito mode' to avoid any personal customization of the google search pages.

Reference lists of included studies were manually screened to identify additional studies. This is to achieve the critical component of 'completeness' as
150 suggested by Kitchenham et al [17].

The platform search capabilities varied, but our search strategy remained uniform for most parts. For instance, if a platform did not support wildcards (like asterisk), we just searched twice for the singular and plural version of the word. The only exception that made the selection process longer was Springer-
155 Link, because it did not support bulk download of references in BibTex format. The reproducible search for the chosen databases is as follows:

- ("Document Title":big data) AND ("Document Title":reference architecture) OR ("Document Title":big data architecture)

The reason we included architecture is due to the fact that these terms may
160 have been used interchangeably, and an architecture that is at the abstraction level of an RA, might have been called just an architecture. Therefore it was critical for us to firmly define these terms and then categorize studies based on these definitions.

The search was limited to 2020 to 2022, as the work of Ataei et al [2] covered
165 the years 2010-2020, and only to researches that has been published in English. Despite the previous effort to collect evidence on the topic, we still included the

years 2010 to 2020 to make sure no research is left out or overlooked. The first limit was imposed because more contemporary researches are focused on the facilitation of big data system development in light of failure rates and there's
170 now SLR that has covered those, and the second limit was imposed to increase specificity.

It is worth mentioning that what we refer to by *limit* here should, not be confused with *filters* or inclusion criteria. To achieve these limits, we have utilized databases features. All databases supported the selection of year range,
175 and the language limit was automatically applied by doing an advanced search with the aforementioned keywords.

Our approach to systematic collection of evidence was to search databases using the keywords aforementioned and then bulk download the BibTex files. Majority of the databases supported bulk downloading of BibTex files except
180 for SpringerLink, Google Scholar, and Research Gate. For SpringerLink we downloaded the studies in CSV format and then converted them to a BibTex using a custom script. For Google Scholar and ResearchGate, unfortunately, we had to take the manual path of creating a bib file for the studies.

Once all the bib files have been created, we merged them into one large bib
185 file and imported it to a software called JabRef ([19]) for deduplication. 172 studies are pooled initially, out of which 6 duplicates have been identified. We removed the SLR that this study is based on, and also another paper that we could not find the citation for. In the other hand, we found 5 white papers and 4 website blogs and added them to the selection pool. At the end of this phase,
190 173 studies have been pooled.

4.2. Screening and Eligibility

Stage 1 of screening started with assessing the title, abstract, and keywords of the pooled studies. For grey literatures simply the title. This was achieved based on our inclusion and exclusion criteria;

- 195 • Primary and secondary studies (including grey literature) between Jan 1st

2010 and June 1st 2022 on the topics of BD RAs, BD models, and BD architectural components were included.

- Research that Indicates the current state of RAs in the field of BD and demonstrates possible outcomes
- 200 • Studies that are scholarly publications, book, book chapter, thesis, dissertation, or conference proceedings
- Grey literature such as white paper that includes extensive information on BD RAs

And the studies with the following topics were excluded:

- 205 • Informal literature surveys without any clearly defined research questions or research process
- Duplicate reports of the same study (a conference and journal version of the same paper)
- Short papers (less than 5 pages)

210 Disagreement among researchers were resolved using Krippendorff's alpha ([20]). Our aim was not to get involved in a very complicated statistics model, so we've done most of the computations using SPSS, specifically with premade SAS macros. Our

$$\alpha \tag{1}$$

value was within the acceptable range, and any disagreement was solved by
215 inviting a third person or a moderator. When

$$\alpha \tag{2}$$

fell to a very low value, we stopped the process, and tried to clarify fundamental concepts.

After excluding papers based on inclusion and exclusion criteria, and as suggested by Kitchenham et al [17], we assessed studies based on their quality.

220 Considering the rather heterogeneous nature of software engineering and information systems (IS) papers, and difficulty of defining quality in studies with varying nature, we first analyzed a few well-established checklists such as Critical Appraisal Skills Programme (CASP [21]), and JBI's critical appraisal tool ([22]).

225 Whereas these checklists could potentially account for the requirements of this study, we opted for something that is more specific to software engineering and IS. We realised for example that, Runeson et al ([23]) provided a checklist designated to help researchers reading and undertaking software engineering case studies. In the same vein, Dyba et al ([24]) proposed a quality criteria based
230 on CASP checklist for qualitative studies in software engineering systematic reviews.

However, the challenge is that our study includes a large number of different study types that needs to go through a single checklist.

Most literature selected were within the years 2016-2021 as they provided
235 with most recent and relevant information. Howbeit, some studies dating back to 2010 have been included to provide fundamental knowledge regarding big data systems, reference architectures and architecture evaluation.

After the initial search, it becomes apparent that AISel and Elsevier are good sources for good quality big data literature, whereas MIS Quarterly provided with the highest quality of Information Systems (IS) research.
240

5. Improvements

1. The current writing style looks like a summary description, lacks new insight on the topic. The overall contribution needs to be enhanced.
2. The inclusion criteria and exclusion criteria are ambiguous and question-
245 able. Is it possible for readers to reproduce this study according to these criteria? Did the author perform the reliability and validity tests? The author needs to provide more detail about the review methodology.

- 250 3. In general, each larger-scale system requires a more understanding of architectural components, owing largely to the complex nature of system architects. However, I cannot find a case that the authors demonstrate the uniqueness of BD systems, and the actual development challenges in BD systems.
- 255 4. The findings yielded by investigating the research questions of this SLR should constitute many discussion points around the research and practice of BD systems. However, the manuscript is completely missing a discussion section. One should expect that the results of SLR can inform the current knowledge and provide several research directions for future research.
- 260 5. Last, one of the core challenges with the paper is to situate it within an ongoing scholarly conversation. The authors currently reference a fairly diverse set of papers, but remain at a fairly abstract level when it comes to elaborating how your work builds upon and expands existing work. In turn, this makes it difficult to appreciate theoretical implications of your work.

265 References

- [1] B. Bashari Rad, N. Akbarzadeh, P. Ataei, Y. Khakbiz, Security and privacy challenges in big data era, *International Journal of Control Theory and Applications* 9 (43) (2016) 437–448.
- [2] P. Ataei, A. T. Litchfield, Big data reference architectures, a systematic literature review (2020).
- 270 [3] B. B. Rada, P. Ataeib, Y. Khakbizc, N. Akbarzadehd, The hype of emerging technologies: Big data as a service (2017).
- [4] Databricks.
URL <https://databricks.com/>

- 275 [5] N. Partners, Big data and ai executive survey 2021 (2021).
 URL https://www.supplychain247.com/paper/bi_data_and_ai_executive_survey_2021/pragmadik
- [6] S. Computing, Bridging the gap between data and business teams (2020).
 URL <https://www.sigmacomputing.com/resources/data-language-barrier/>
 280
- [7] B. B. Rad, P. Ataei, The big data ecosystem and its environs, International Journal of Computer Science and Network Security (IJCSNS) 17 (3) (2017) 38.
- [8] I. Gorton, J. Klein, Distribution, data, deployment, STC 2015 (2015) 78.
- 285 [9] R. Cloutier, G. Muller, D. Verma, R. Nilchiani, E. Hole, M. Bone, The concept of reference architectures, Systems Engineering 13 (1) (2010) 14–27.
- [10] M. Derras, L. Deruelle, J.-M. Douin, N. Levy, F. Losavio, Y. Pollet, V. Reiner, Reference architecture design: A practical approach, in: IC-SOFT, pp. 633–640.
 290
- [11] I. Iso, Information technology — reference architecture for service oriented architecture (soa ra) — part 1: Terminology and concepts for soa, International Organization for Standardization (2016) 51.
 URL <https://www.iso.org/standard/63104.html>
- 295 [12] G. Muller, A reference architecture primer, Eindhoven Univ. of Techn., Eindhoven, White paper (2008).
- [13] L. Bass, I. Weber, L. Zhu, DevOps: A software architect’s perspective, Addison-Wesley Professional, 2015.
- [14] S. Angelov, P. Grefen, D. Greefhorst, A classification of software reference architectures: Analyzing their success and effectiveness, in: 2009
 300

Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, IEEE, 2009, pp. 141–150.

- [15] M. J. Page, D. Moher, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, S. E. Brennan, et al.,
305 Prisma 2020 explanation and elaboration: updated guidance and exemplars for reporting systematic reviews, *Bmj* 372 (2021).
- [16] M. L. Rethlefsen, S. Kirtley, S. Waffenschmidt, A. P. Ayala, D. Moher, M. J. Page, J. B. Koffel, Prisma-s: an extension to the prisma statement for reporting literature searches in systematic reviews, *Systematic reviews*
310 10 (1) (2021) 1–19.
- [17] B. A. Kitchenham, D. Budgen, P. Brereton, Evidence-based software engineering and systematic reviews, Vol. 4, CRC press, 2015.
- [18] M. Borrego, M. J. Foster, J. E. Froyd, Systematic literature reviews in engineering education and other developing interdisciplinary fields, *Journal of Engineering Education* 103 (1) (2014) 45–76.
315
- [19] [link].
URL <https://www.jabref.org/>
- [20] K. Krippendorff, Computing krippendorff’s alpha-reliability (2011).
- [21] [link].
320 URL <https://casp-uk.net/casp-tools-checklists/>
- [22] [link].
URL <https://jbi.global/critical-appraisal-tools>
- [23] P. Runeson, C. Andersson, T. Thelin, A. Andrews, T. Berling, What do we know about defect detection methods?[software testing], *IEEE software*
325 23 (3) (2006) 82–90.

- [24] T. Dybå, T. Dingsøy, Empirical studies of agile software development: A systematic review, *Information and software technology* 50 (9-10) (2008) 833–859.