Conference Paper Title*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

I. Requirements Specification

Precursor to theorizing about the potential of microservices patterns for big data systems, we need to define what we mean by big data systems and what are the requirements of these systems. System and software requirements come in different flavour and can range from a sketch on a napkin to formal (mathematical) specifications. Therefore, we first need to identify what kind of requirements is the most suitable for the purposes of this study. To answer this question, we first explored the body of evidence to understand the current classification of software requirements.

There's been various attempts to defining and classifying software and systems requirements. For instance, Sommerville ([**?**]) classified requirements into three levels of abstraction that are namely 1) user requirements, 2) system requirements and 3) design specifications. The author then mapped these requirements against user acceptance testing, integration testing and unit testing. While this could satisfy the requirements of this study, we opted for a a more general framework provided by Laplante ([**?**]). In Laplante's approach, requirements are categorized into three categories of 1) functional requirements, 2) non-functional requirements, and 3) domain requirements.

Our objective is to define the high-level requirements of big data systems, thus we do not seek to explore 'non-functional' requirements. Non-functional requirements are emerged from the particularities of an environment, such as a banking sector and do not correlate to our study. Therefore, the type of requirements we are looking for is functional and domain requirements.

After clarifying the type of requirements, we then explored the body of evidence to realize the general requirements of big data systems. Indeed, the most discussed characteristics of big data systems are the popular 5Vs which are velocity, veracity, volume, Variety and Value ([**?**], [**?**], [**?**], [**?**], [**?**], [**?**]). Many researchers such as Nadal et al. ([**?**]) have underpinned their artifact development on these characteristics and requirements that emerge from them.

In an extensive effort, NIST Big Data Public Working Group embarked on a large scale study to extract requirements from variety of application domains such as Healthcare and Life Sciences, Commercial, Energy, Government, and Defense. The result of this study was the formation of general requirements under seven categories. In another effort by Volk et al. ([**?**]),9 use cases for big data projects are identified by collecting theories and use cases from the literature and categorizing them using a hierarchical clustering algorithm. Bashari et al. ([**?**]) focused on the security and privacy requirements of big data systems, Yu et al. presented the modern components of big data systems [**?**], Eridaputra et al. ([**?**]) created a generic model for big data requirements using goal oriented approaches, and Al-jaroodi et al. ([**?**]) investigated general requirements to support big data software development.

We've also studied the reference architectures developed for big data systems to understand general requirements. In one study, Ataei et al. ([**?**]) assessed the body of evidence and presented with a comprehensive list of big data reference architectures. This study helped us realized the spectrum of big data reference architectures, how they are designed and the general set of requirements.

By analyzing these studies and by evaluating the design and requirement engineering required for big data reference architectures, we created a set of high-level requirements based on big data characteristics. We have then looked for

Identify applicable funding agency here. If none, delete this.

a rigorous approach to present these requirements. There are numerous approaches used for requirement representation including informal, semiformal and formal methods. For the purposes of this study, we opted for an informal method because it's a well established method in the industry and academia ([**?**]).

Our approach follows the guidelines explained in ISO/IEC/IEEE standard 29148 for representing functional requirements. Our requirement repesentation is organized in system modes, that is we explain the major components of the system and then describe the requirements. This approach is inspired by the requirement specification expressed for NASA WIRE (wide-field infrared explorer) system explained in [**?**]. We also taken inspiration from Software Engineering Body of Knowledge Version

Taking all into consideration, we deduce the following general requirements for big data systems:

1 NIST requirements

DATA SOURCE REQUIREMENTS (DSR) DSR-1: Needs to support reliable real-time, asynchronous, streaming, and batch processing to collect data from centralized, distributed, and cloud data sources, sensors, or instruments. DSR-2: Needs to support slow, bursty, and high-throughput data transmission between data sources and computing clusters. DSR-3: Needs to support diversified data content ranging from structured and unstructured text, document, graph, web, geospatial, compressed, timed, spatial, multimedia, simulation, and instrumental data.

TRANSFORMATION PROVIDER REQUIREMENTS (TPR) TPR-1: Needs to support diversified compute-intensive, statistical and graph analytic processing, and machine learning techniques. TPR-2: Needs to support batch and real-time analytic processing. TPR-3: Needs to support processing large diversified data content and modeling. TPR-4: Needs to support processing data in motion (streaming, fetching new content, tracking, etc.).

CAPABILITY PROVIDER REQUIREMENTS (CPR) CPR-1: Needs to support legacy and advanced software packages (software). CPR-2: Needs to support legacy and advanced computing platforms (platform). CPR-3: Needs to support legacy and advanced distributed computing clusters, co-processors, input output (I/O) processing (infrastructure). CPR-4: Needs to support elastic data transmission (networking). CPR-5: Needs to support legacy, large, and advanced distributed data storage (storage). CPR-6: Needs to support legacy and advanced executable programming: applications, tools, utilities, and libraries (software).

DATA CONSUMER REQUIREMENTS (DCR) DCR-1: Needs to support fast searches from processed data with high relevancy, accuracy, and recall. DCR-2: Needs to support diversified output file formats for visualization, rendering, and reporting. DCR-3: Needs to support visual layout for results presentation. DCR-4: Needs to support rich user interface for access using browser, visualization tools. DCR-5: Needs to support high-resolution, multi-dimension layer of data visualization. DCR-6: Needs to support streaming results to clients.

SECURITY AND PRIVACY REQUIREMENTS (SPR) SPR-1: Needs to protect and preserve security and privacy of sensitive data. SPR-2: Needs to support sandbox, access control, and multi-level, policy-driven authentication on protected data.

LIFE CYCLE MANAGEMENT REQUIREMENTS (LMR) LMR-1: Needs to support data quality curation including pre-processing, data clustering, classification, reduction, and format transformation. LMR-2: Needs to support dynamic updates on data, user profiles, and links. LMR-3: Needs to support data life cycle and long-term preservation policy, including data provenance. LMR-4: Needs to support data validation. LMR-5: Needs to support human annotation for data validation. LMR-6: Needs to support prevention of data loss or corruption. LMR-7: Needs to support multi-site archives. LMR-8: Needs to support persistent identifier and data traceability. LMR-9: Needs to support standardizing, aggregating, and normalizing data from disparate sources.

OTHER REQUIREMENTS (OR) OR-1: Needs to support rich user interface from mobile platforms to access processed results. OR-2: Needs to support performance monitoring on analytic processing from mobile platforms. OR-3: Needs to support rich visual content search and rendering from mobile platforms. OR-4: Needs to support mobile device data acquisition. OR-5: Needs to support security across mobile devices.

2 Fundamentals of data engineering

generation ingestion transformation serving storage security data management data governance - discoverability - metadata - data accountability data modeling and design data lineage data integration and operability data lifecycle management data quality - accuracy - completeness - timeliness ethics and privacy dataops orchestration software engineering

3 Semantic aware big data reference architecture

1. Volume R1.1 The BDA shall provide scalable storage of massive data sets. R1.2 The BDA shall be capable of supporting descriptive analytics. R1.3 The BDA shall be capable of supporting predictive and prescriptive analytics. 2. Velocity R2.1 The BDA shall be capable of ingesting multiple, continuous, rapid, time varying data streams. R2.2 The BDA shall be capable of processing data in a (near) real-time manner. 3. Variety R3.1 The BDA shall support ingestion of raw data (structured, semi-structured and unstructured). R3.2 The BDA shall support storage of raw data (structured, semi-structured and unstructured). R3.3 The BDA shall provide mechanisms to handle machine-readable schemas for all present data. 4. Variability R4.1 The BDA shall provide adaptation mechanisms to schema evolution. R4.2 The BDA shall provide adaptation mechanisms to data evolution. R4.3 The BDA shall provide mechanisms for automatic inclusion of new data sources. 5. Veracity R5.1 The BDA shall provide mechanisms for data provenance. R5.2 The BDA shall provide mechanisms to measure data quality. R5.3 The BDA shall

provide mechanisms for tracing data liveliness. R5.4 The BDA shall provide mechanisms for managing data cleaning.

4 Towards a big data reference architecture Maier

1.Volume - The system shall store data up to a volume of ¡p1: specify data volume¿ in the following formats ¡p2: specify required data formats¿. - The system shall be scalable, in the sense that the processed data volume per time unit can be improved by adding hardware resources while making use of the additional resources in a linear manner with a factor of at least ¡p1: define scaling factor¿. - The system shall respond to a query that involves ¡p1: define amount of data¿ within a response time of ¡p2: define response time¿, while the system runs on ¡p3: define base hardware configuration¿. 2. Velocity - The system shall handle data while it is flowing in with a rate of up to ¡p1: specify inflow rate¿. - The system shall conduct analysis tasks on streaming data, create insights as specified in VEL1.1.1 and react to them as specified in VEL1.1.2 while data is flowing in with a rate as specified in VEL1. - The system shall conduct analysis tasks on streaming data and create insights as specified in ¡p1: specify functional requirements that describe the actual data analysis steps¿ while data is flowing in - The system shall create insights from streaming data as specified in VEL1.1.1 within a time frame of ¡p1: specify time acceptable for generating insights¿ while data is flowing in with a rate as specified in VEL1. - The system shall pre-compute the following rules and models ¡p1: specify models to pre-compute¿ from the persistent data available in the system and apply these models to process streaming-in data. - The system shall communicate the insights from VEL1.1.1 as specified in ¡p1: specify functional requirements that describe to which systems or recipients and in which format insights should be communicated¿. - The system shall communicate the insights from VEL1.1.1 as specified in VEL1.1.2 within a time frame of ¡p1: specify time acceptable for communicating insights¿ while data is flowing in with a rate as specified in VEL1. - The system shall process and acquire data as specified in ¡p1: specify functional requirements that describe how and which parts of the streaming data should be acquired and pre-processed¿ and store it. - The system shall acquire, pre-process and store streaming data as specified in VEL1.2 with an acquisition rate of ¡p1: specify acquisition rate¿ while data is flowing in with a rate as specified in VEL1. 3. Variety: - The system shall extract data in the following formats ¡p1: specify required data formats¿ from the following sources ¡p2: specify required data sources¿ - The system shall filter out data extracted from sources based on the following rules ¡p1: specify filter rules¿ - The system shall handle multistructured data in the following formats ¡p1: specify required data formats¿ and from the following source ¡p2: specify required data sources¿, where handling means to store them, manage them, extract the necessary information and apply analysis tasks as specified in the other requirements. - The system shall extract (additional) machine-readable information from the following formats ¡p1: specify data formats¿ and from the following source ¡p2: specify required data sources¿ and therefore impose structure onto this data. - The system shall host and manage several extractors using different information extraction techniques and algorithms and apply a configurable subset of them to data from different sources. - The system shall store the information extracted by different extractors according to requirement VAR2.1.1 in a structured form in one integrated data model and related to the source data. - The system shall integrate heterogeneous data from the following sources ¡p1: specify data sources¿ and with the following formats ¡p2: specify data formats¿ into an unified view - The system shall maintain a global schema and a semantic mapping of the schemas of the different data sources specified in VAR3 onto that global schema. The global schema should be ¡p1: virtual / persisted¿. - The system shall be able maintain rules to resolve and match similar entities, that is to identify tuples and field values that refer to the same entity and collapse them. This should be done ¡p1: virtually / persistent¿ - The system shall enable users and analysis tasks to directly access the original source data without any pre-processing or integration. 4. Variety - The system shall gather and store metadata to describe: • the data source structure, schema and recording method • the data structures used within the system • analysis techniques and processing steps within the system • for data items from which source they are from and which processing steps have already been conducted (data provenance) • operational information about the analysis processes - The system shall store metadata using the following data structures ¡p1: describe format / schema of the structures¿ and directly related to the data or structures it describes. - The system shall additionally extract the following metadata in the following formats ¡p1: specify metadata formats, e.g. microformats¿ when extracting data from: • Data source • Structure of the data source • Recording method of the data in the data source - The system shall collect metadata during the processing of data to track the provenance for data sets within the system, that is their source and which processing steps where conducted to them - The system shall allow administrators and users to inspect the available metadata for each data source, data item, data structure etc. and to adjust metadata they consider to be incorrect 5. Veracity - The system shall handle uncertain data, that is give meaningful results in the face of uncertain data and put those results into context. - The system shall improve data quality by cleaning data from the following sources ¡p1: specify data sources¿ and in the following formats ¡p2: specify data formats¿. - The system shall use the following techniques ¡p1: specify techniques¿ to fill empty fields with estimated values under the following conditions ¡p2: specify conditions¿. - The system shall use the following conditions ¡p1: specify conditions¿ to identify untrustworthy or incorrect data and apply the following techniques ¡p2: techniques¿ to resolve the untrustworthiness. - The system shall track the trustworthiness of data on the level of ¡p1: specify level to track trustworthiness¿ and calculate a trustworthiness metric in the following way ¡p2: specify method to calculate trustworthiness¿. 6. Value - The system shall conduct analysis tasks as specified in sub-requirements VAL1.1, VAL1.2 and VAL1.3 over data from

the following sources ¡p1: specify sources¿. - The system shall conduct the following deep analytics tasks ¡p1: specify functional requirements that describe the necessary deep analytics tasks¿ as batch-jobs over the following data sources ¡p2: specify data sources¿ - The system shall provide users with pre-calculated standard reports as specified in ¡p1: specify functional requirements to describe standard reports¿ with traditional OLAP-like navigation functionality using data from the following sources ¡p2: specify data sources¿. - The system shall enable users to interactively work with the available data and to query and analyse it in an ad-hoc manner. - The system shall provide users an overview of and enable them to navigate through all available data sources, data available in the system and already computed results for analysis tasks and single processing steps all together with the related metadata. - The system shall provide users with an end-point to formulate and process ad-hoc queries and processing tasks over data from the following sources ¡p1: specify data sources¿ using the following methods ¡p2: specify methods to formulate queries and processing tasks, e.g. query languages¿. - The system shall make it easy for users to explore available data, analysis results as well as results of intermediate steps and should support users in understanding and interpreting those results. - The system shall visualize the following data and analysis results ¡p1: specify data to be visualized¿ as specified in ¡p2: specify functional requirements that describe the necessary visualization tasks¿. - The system shall support programmer productivity by allowing programmers to focus on the application logic while abstracting low-level implementation and infrastructure details away. - The system shall provide an end-point that allows users to formulate queries and analysis tasks in the following declarative query language(s) ¡p1: specify declarative query languages¿ and process those queries over the following data sources ¡p2: specify data sources¿. - The system shall incorporate the following analytical packages ¡p1: specify analytical packages¿ to be used by users to analyse data from the following sources ¡p2: specify data sources¿. - As already mentioned in requirement VAL1.3, power users work interactively with data, making hypotheses, testing these by experimenting with the data, drawing some conclusions and refine their hypotheses. They also need to experiment with new analysis methods, data mining and machine learning techniques, parametrisation of those etc. This kind of experimentation needs to be supported by the system, e.g. by making it easy to extract samples out of data and to work with it in a sandbox. - The system shall manage the lifecycle of the stored data, that is track data records and documents from their creation and, determine based on the following rules ¡p1: specify functional requirements that describe data lifecycle management rules¿ if data is still to retain or became stale and to archive or delete data - The system shall compress data where possible using the following compression technique ¡p1: specify compression technique¿. - The system shall archive or delete data based on the rules specified in requirement VAL4 using the following archiving method ¡p1: specify archiving method and archiving system used¿. - The system shall ensure privacy of data, that is data should not be able to be accessed by people not authorized for it. The system shall therefore conduct the following measures ¡p1: specify functional sub-requirements that describe measures to ensure privacy¿.

REFERENCES

[1] I. Sommerville, *Software Engineering, 9/E*. Pearson Education India, 2011.

[2] P. A. Laplante, *Requirements engineering for software and systems*. Auerbach Publications, 2017.

[3] Y. Demchenko, C. De Laat, and P. Membrey, "Defining architecture components of the big data ecosystem," in *2014 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, 2014, Conference Proceedings, pp. 104–112.

[4] J. Bughin, "Big data, big bang?" *Journal of Big Data*, vol. 3, no. 1, p. 2, 2016.

[5] M. Bahrami and M. Singhal, *The role of cloud computing architecture in big data*. Springer, 2015, pp. 275–295.

[6] B. B. Rad and P. Ataei, "The big data ecosystem and its environs," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 17, no. 3, p. 38, 2017.

[7] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable real-time data systems*. New York; Manning Publications Co., 2015.

[8] H.-M. Chen, R. Kazman, and S. Haziyev, "Agile big data analytics development: An architecture-centric approach," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2016, Conference Proceedings, pp. 5378–5387.

[9] S. Nadal, V. Herrero, O. Romero, A. Abelló, X. Franch, S. Vansummeren, and D. Valerio, "A software reference architecture for semantic-aware big data systems," *Information and software technology*, vol. 90, pp. 75–92, 2017.

[10] M. Volk, D. Staegemann, I. Trifonova, S. Bosse, and K. Turowski, "Identifying similarities of big data projects–a use case driven approach," *IEEE Access*, vol. 8, pp. 186 599–186 619, 2020.

[11] B. Bashari Rad, N. Akbarzadeh, P. Ataei, and Y. Khakbiz, "Security and privacy challenges in big data era," *International Journal of Control Theory and Applications*, vol. 9, no. 43, pp. 437–448, 2016.

[12] J.-H. Yu and Z.-M. Zhou, "Components and development in big data system: A survey," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 51–72, 2019.

[13] H. Eridaputra, B. Hendradjaya, and W. D. Sunindyo, "Modeling the requirements for big data application using goal oriented approach," in *2014 international conference on data and software engineering (ICODSE)*. IEEE, 2014, pp. 1–6.

[14] J. Al-Jaroodi and N. Mohamed, "Characteristics and requirements of big data analytics applications," in *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*. IEEE, 2016, pp. 426–432.

[15] P. Ataei and A. T. Litchfield, "Big data reference architectures, a systematic literature review," 2020.

[16] M. Kassab, C. Neill, and P. Laplante, "State of practice in requirements engineering: contemporary data," *Innovations in Systems and Software Engineering*, vol. 10, no. 4, pp. 235–241, 2014.