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 Semantic aware big data reference architecture