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 categorized our requirements based on the major characteristics of big data, that is value, variety, velocity, veracity, and volume ([**?**]), plus . These requirements are as followings:

Volume	1) System needs to support asynchronous, streaming, and batch processing to collect data from centralized, distributed, and cloud data sources, and sensors, instrument and other IOT devices 2) System needs to be able to process large heterogenous data with varying schemas 3) System needs to provide a scalable storage for massive data sets
Velocity	1) System needs to support slow, bursty, and high-throughput data transmission between data sources and computing clusters 2) System needs to stream data to data consumers in a timely manner 3) System needs to able to ingest multiple, continuous, time varying data streams 4) System shall support fast search from streaming and processed data with high accuracy and relevancy 5) System should be able to process data in real-time or near real-time manner
Variety	1) System needs to support data in various formats ranging from structured to semi-structured and unstructured graph, web, text, document, timed, spatial, multimedia, simulation, instrumental, and geo-spatial data. 2) System needs to support aggregation, standardization, and normalization of data from disparate sources
Value	1) System needs to able to handle compute-intensive analytical processing and machine learning techniques 2) System needs to support two types of analytical processing: batch and streaming. 3) System needs to support different output file formats for different for reporting and visualizations. 4) System needs to support streaming results to the consumers 5) System should support descriptive analytics 6) System shall support predictive analytics
Security & Privacy	1) System needs to protect and retain privacy and security of sensitive data. 2) System needs to have access control, and multi-level, policy-driven authentication on protected data and processing nodes.
Veracity	1) System needs to support data quality curation including classification, pre-processing, format, reduction, and transformation. 2) System needs to support data provenance including data life cycle management and long-term preservation. 3) System needs to support data validation in two ways: automatic and human annotated. 4) System should be able to handle data loss or corruption.
Variability	1) System shall support adaptations mechanisms for schema evolution. 2) System can provide mechanisms to automatically include new data sources