DataBench

Evidence Based Big Data Benchmarking to Improve Business Performance

D3.1 DataBench Architecture

Abstract

This document provides an overview of the DataBench Toolbox Architecture and main functional elements. The DataBench Toolbox aims to be an umbrella framework for big data benchmarking based on existing efforts in the community. It will include features to reuse existing big data benchmarks into a common framework, and will help users to search, select, download, execute and get a set of technical and business indicators out of the benchmarks' results.

The Toolbox is therefore one of the main building blocks of the project and the main interaction point with the users of benchmarking tools. This document provides the architectural foundations and main elements of the tooling support to be used by big data benchmarking practitioners. In this sense, the document gives an overview of the different elements of DataBench ecosystem to contextualize the significance of the Toolbox, as well as details about the different components of the Toolbox identified so far, and hints about their potential implementation.

This document is the first deliverable related to the DataBench Toolbox. Updates to the architecture will be provided as integral part of the different releases of the Toolbox expected in the DataBench WP3 lifecycle.

European Commission

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780966

Deliverable D3.1	DataBench Architecture
Work package	WP3
Task	3.1
Due date	30/06/2018
Submission date	07/07/2018
Deliverable lead	ATOS
Version	1.0
Authors	ATOS (Tomás Pariente, Iván Martínez and Ricardo Ruiz) GUF (Todor Ivanov) SINTEF (Arne Berre) POLIMI (Chiara Francalanci)
Reviewers	Richard Stevens (IDC)

Keywords

Benchmarking, big data, big data technologies, architecture, business performance, performance metrics, toolbox, use cases

Disclaimer

This document reflects the authors view only. The European Commission is not responsible for any use that may be made of the information this document contains.

Copyright Notice

Table of Contents

Table of Figures

Executive Summary

This document presents an overview of the DataBench ecosystem with special emphasis on the initial specification of the DataBench Toolbox architecture. To this extent, the document puts the DataBench Toolbox in the context of the main functional elements to be delivered by the different Work Packages of the project. In order to do that, the document gives an overview of the different elements of the ecosystem, namely the identification of potential benchmarks to be integrated into the Toolbox and alignment to the BDVA Reference Model, the business reports about the elements of industrial significance for benchmarking big data solutions, the identification of business KPIs and the project Handbook, along with the supporting Artificial Intelligence methods to be delivered to recommend and generalise aspects on the selection of benchmarks or derivation of KPIs.

Besides the description of the context, the core of the document is the definition of the architecture of the DataBench Toolbox. The DataBench Toolbox aims to be an umbrella framework for big data benchmarking based on existing efforts in the community. It will include features to reuse existing big data benchmarks into a common framework, and will help users to search, select, download, execute and get a set of homogenized results. The DataBench Toolbox will be an integral part of the DataBench Framework, which ultimately will deliver recommendations and business insights out of big data benchmarks. Therefore, a potential architecture along with the presentation of Unified Modelling Language (UML) use cases to represent the processes associated to the architecture are depicted and explained. Ideas, phases and potential technologies to implement the Toolbox are also explained.

1. Introduction

This document presents the DataBench Toolbox initial architecture. The DataBench Toolbox will include formal mechanisms to ease the process of reusing existing or new big data benchmarks into a common framework that will help stakeholders to find, select, download, execute and get a set of homogenized metrics. The DataBench Toolbox will be an integral part of the DataBench framework, which ultimately will deliver recommendations and business KPIs out of big data benchmarks.

The present document therefore starts by putting the DataBench Toolbox in the context with the rest of the DataBench framework to later on dive into the details of the envisaged architecture. It is important to notice that the Toolbox will be based on existing efforts on big data benchmarking, rather than proposing new benchmarks. The DataBench Toolbox therefore aims to be an umbrella framework for big data benchmarking. The idea behind it is to provide ways to declare new benchmarks into the Toolbox and provide a set of automatisms and recommendations to allow the usage of these tools to become part of the ecosystem. Due to the different nature and technical scope of the existing tools, the degree of automation may vary from one tool to another. The baseline will be the possibility of downloading the selected benchmarking tools from the Toolbox web user interface and provide adapters to actually get the results of the benchmarking into the DataBench Toolbox in order to get homogenized metrics. Other tools may be subject to tighter integration and even automation of the deployment in the benchmarking system.

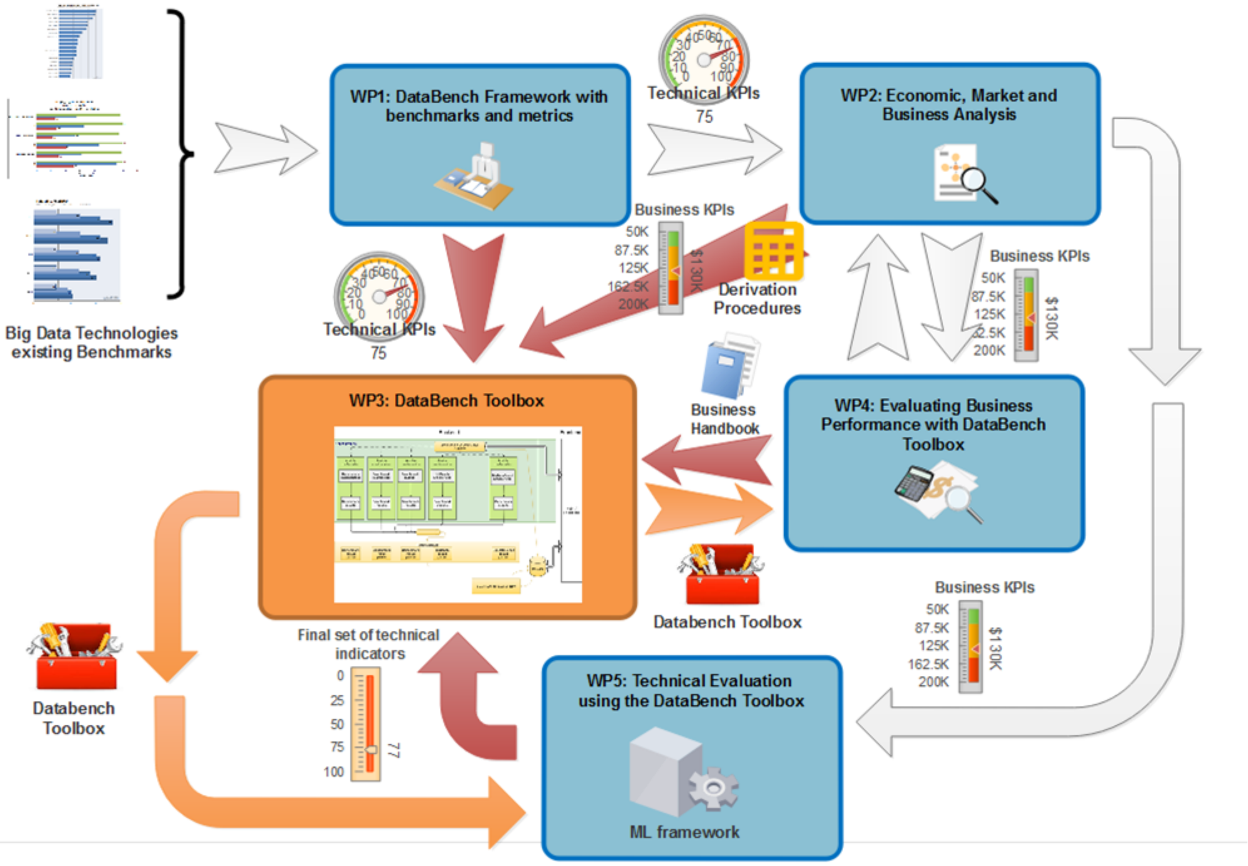
The document is structured as follows:

- Section 1 provides the introduction to the objectives of the deliverable.
- Section 2 describes the DataBench Toolbox in the context of the rest of the DataBench ecosystem, providing a functional view of the different components.
- Section 3 dives into the processes envisaged in the usage of the DataBench ecosystem, with special detail in the ones related to the DataBench Toolbox.
- Section 4 is the core section of the document, where the first ideas about the architecture of the DataBench Toolbox are explained.
- Finally, Section 0 provides the conclusions of the document and outlines the future work on the DataBench Toolbox.

2. DataBench in a nutshell

2.1 DataBench elements

The DataBench project is composed of several Work Packages, each of which is in charge of delivering a set of results of different nature (software, reports, handbooks, etc.). From the technical perspective, the DataBench Toolbox, to be delivered in WP3, is the main software element that will be provided to the users, but it is not in isolation as depicted in Figure 1.



Big Data Technologies existing Benchmarks
WP1: DataBench Framework with benchmarks and metrics
Technical KPIs
75
WP2: Economic, Market and Business Analysis
Business KPIs
50K
87.5K
125K
162.5K
200K
$130K
Derivation Procedures
Technical KPIs
75
Business KPIs
WP3: DataBench Toolbox
Business Handbook
WP4: Evaluating Business Performance with DataBench Toolbox
Databench Toolbox
Business KPIs
Final set of technical indicators
0
25
50
75
100
77
WP5: Technical Evaluation using the DataBench Toolbox
ML framework
Databench Toolbox

Figure 1 - Functional view of the DataBench ecosystem

The different elements of the DataBench ecosystem listed in Figure 1 can be divided into a set of major elements of the Framework:

- The DataBench Toolbox: Developed in the scope of WP3 and the main subject of this document, the DataBench Toolbox is the core technical component of the DataBench Framework. It will be the entry point for users that would like to perform big data benchmarking and will ultimately deliver recommendations and business insights. It will include features to reuse existing big data benchmarks, and will help users to search, select, download, execute and get a set of homogenized results. The Toolbox takes as input most of the work done in the rest of the project's Work Packages.
- The benchmarks to be integrated into the Toolbox: Located on the left-hand side of Figure 1, external big data benchmarks are the input to the whole DataBench Framework, and in particular to the DataBench Toolbox. These benchmarks will be made available to the users from the Toolbox user interface, along with the possibility to get the results of the benchmarking execution runs and homogenize

them into a common model in order to get homogenized set of metrics. The degree of integration of the Toolbox with the existing benchmarks will vary depending on the integration issues that may arise.

- The process to derive business KPIs: Deriving business KPIs from the results of running big data benchmarks is not a straightforward process. It will depend on the business context and, therefore, it is not completely clear to what extent it could be automated. However, different approaches to derive business insights are currently under study and will be reported in future deliverables. This process is a collaborative work among the different Work Packages. WP1 is investigating the different existing benchmarks in order to devise a coherent framework and providing input on available technical KPIs and mappings to existing big data reference models and to the data value chain in order to cluster them. This work will be applied in WP4 to the desk analysis and to the in-depth case studies, which will result into an analysis of the relations between technical and business KPIs, use cases and a Handbook to guide users to select and interpret the results. WP2, on the other hand, will provide guidance in terms of market value.
- The AI framework: WP5 will provide a Machine Learning framework that will serve to enable recommendations for the benchmarking community based on past experiences. It will be integrated with the Toolbox.

These elements of the Framework are described in more detail in the following sections.

2.2 The DataBench Toolbox

The DataBench Toolbox is the core technical platform to be provided by DataBench to users of the big data benchmarking community. The goal of the Toolbox is to provide tooling support to big data benchmarking users to on the one hand, search, select, deploy and run existing big data benchmarks, and on the other hand, get the results of the execution, homogenize the technical metrics, and finally help to derive business insights and KPIs. It is therefore a tool based on existing efforts on big data benchmarking, taking advantage of years of research and practice accumulated in the community. Therefore, the Toolbox will be based on existing efforts on big data benchmarking, rather than proposing new benchmarks.

The Toolbox will offer ways to reuse new benchmarks and provide a set of automatisms and recommendations to allow the usage of these tools to become part of the ecosystem. These benchmarks will be made available to the users from the Toolbox user interface, along with the possibility to get the results of the benchmarking execution runs and homogenize them into a common model in order to get homogenized set of metrics. The degree of integration of the Toolbox with the existing benchmarks will vary depending on the integration issues that may arise.

From the usage perspective of benchmarking tools, DataBench has detected two very different scenarios. On the one hand, there are organizations that do not have any specific privacy or security constraints to run big data benchmarks, and therefore the setting of the benchmark may happen in public infrastructure. On the other hand, there are other organizations and industries that will only benchmark their big data environment in a seclude infrastructure, certainly not public and without any Internet connection open. This means that the Toolbox should cater for both scenarios and be able to enable the setting of the benchmarks completely offline if the user decides so.

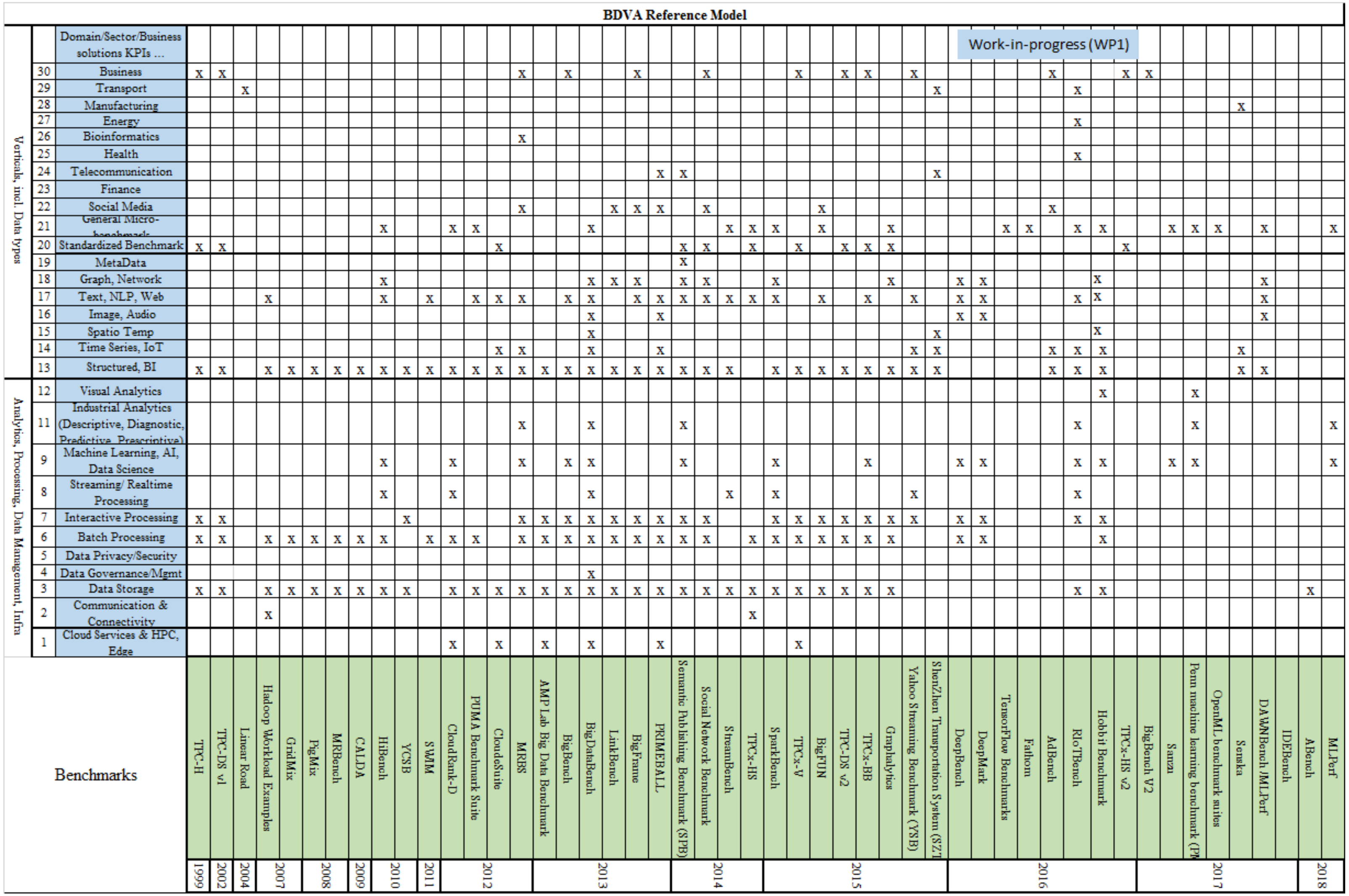
Therefore, the Toolbox will offer two different deployment strategies or Toolbox flavours. The first one, the sandbox deployment, will provide a way to create the necessary deployment recipes from the Toolbox server-side installation. This will allow users, for instance, to define where (i.e. IP and port numbers) the benchmarks selected should be deployed. This is very well suited for deployments in public clouds or public infrastructure, where users do not impose very tight security constraints to run the benchmarks. On the other hand, the Toolbox will offer a second deployment option, or in-house deployment. This is more suited for users that would like to run the benchmarks in their own infrastructure or where the system to benchmark resides. It usually involves higher security and privacy requirements. In this case, as many other benchmarking tools do, a version of the DataBench Toolbox will be downloaded after the selection of the benchmark to execute that will help the user to setup the benchmark and later, give back the results of the executions.

This feedback loop is one of the key aspects of the Toolbox, as the idea is that the results from the benchmark runs will be reported back to the Toolbox at the DataBench server-side to homogenize the technical metrics and derive a set of business insights or KPIs. This process is explained in section 2.4 but it will be supported by the Toolbox and should be enabled for both deployment strategies.

The processes associated to the Toolbox are explained in detail in section 3, while the architecture of the Toolbox itself and the current potential implementation ideas are explained in section 4.

2.3 Benchmarks integration

An essential part of the DataBench project is to identify and classify all existing Big Data benchmarks and their communities. This work is in progress as part of WP1 and will be reported in full in the scope of that Work Package. In this section, the idea is to highlight the current status of the analysis of the major benchmarking efforts that DataBench is doing, with the hope to identify the candidate benchmarks to be integrated into the Toolbox. Figure 2 is a current snapshot of a matrix positioning the Big Data benchmarks according to different criteria defined in the BDVA Reference Model (see section **Error! Reference source not found.** for more details) [BDV SRIA]. On the left (in blue) are listed the different industry application domains, data types and technology areas. At the bottom (in green), all the different Big Data benchmarks are listed in release time order.

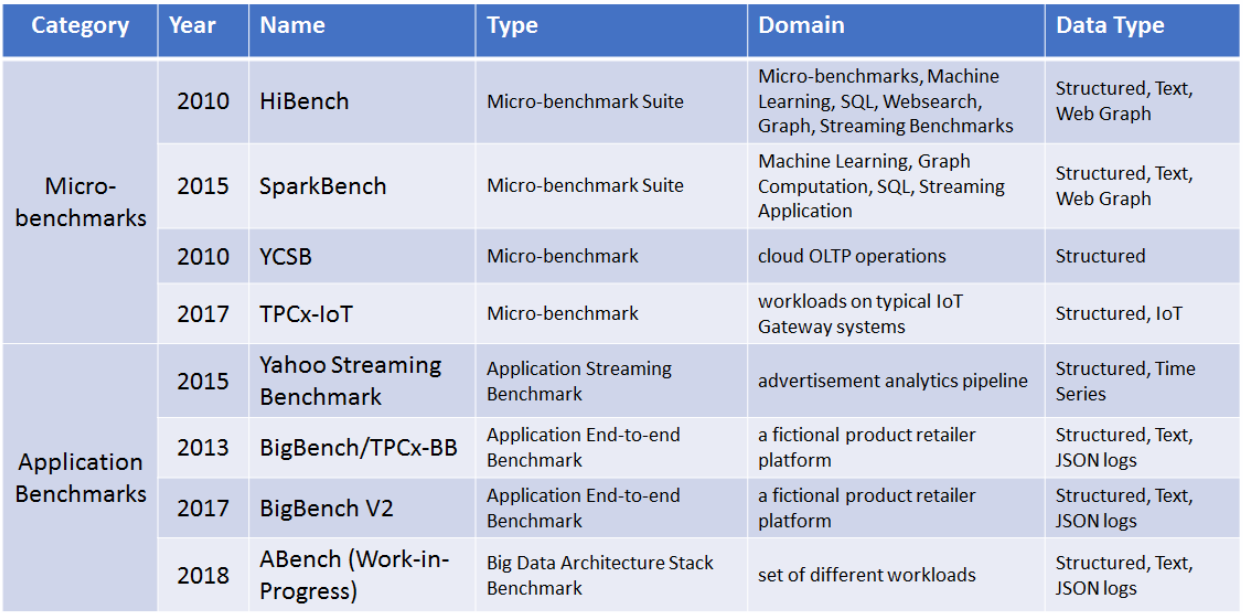


BDVA Reference Model

		Benchmarks	TPC-H	TPC-DS v1	Linear Road	Hadoop Workload Examples	GridMix	PigMix	MRBench	CALDA	HiBench	YCSB	SWIM	CloudRank-D	PUMA Benchmark Suite	CloudeSuite	MRBS	AMP Lab Big Data Benchmark	BigBench	BigDataBench	LinkBench	BigFrame	PRIMEBALL	Semantic Publishing Benchmark (SPB)	Social Network Benchmark	StreamBench	TPCx-HS	SparkBench	TPCx-V	BigFUN	TPC-DS v2	TPCx-BB	Graphalytics	Yahoo Streaming Benchmark (YSB)	ShenZhen Transportation System (SZT	DeepBench	DeepMark	TensorFlow Benchmarks	Fathom	AdBench	RIoTBench	Hobbit Benchmark	TPCx-HS v2	BigBench V2	Sanzu	Penn machine learning benchmark (PM	OpenML benchmark suites	Senska	DAWNBench /MLPerf	IDEBench	ABench	MLPerf
Verticals, incl. Data types		Domain/Sector/Business solutions KPIs ...																																		Work-in-progress (WP1)																
	30	Business	x	x													x		x			x			x				x		x	x		x						x			x	x								
	29	Transport			x																														x						x											
	28	Manufacturing																																														x				
	27	Energy																																								x											
	26	Bioinformatics															x																																			
	25	Health																																								x											
	24	Telecommunication																					x	x											x																	
	23	Finance																																																		
	22	Social Media															x				x	x	x		x					x										x												
	21	General Micro-benchmark									x			x	x					x						x	x	x		x			x					x	x		x	x			x	x	x		x			x
	20	Standardized Benchmark	x	x												x								x	x		x		x		x	x	x										x									
	19	MetaData																						x																												
	18	Graph, Network									x									x	x	x		x	x			x					x			x	x					x							x			
	17	Text, NLP, Web				x					x		x		x	x	x		x	x		x	x	x	x	x	x	x		x		x		x		x	x				x	x							x			
	16	Image, Audio																		x			x													x	x												x			
	15	Spatio Temp																		x															x							x										
	14	Time Series, IoT														x	x			x			x											x	x					x	x	x						x				
	13	Structured, BI	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x	x					x	x	x						x	x			
Analytics, Processing, Data Management, Infra	12	Visual Analytics																																								x				x						
	11	Industrial Analytics (Descriptive, Diagnostic, Predictive, Prescriptive)															x			x				x																	x					x						x
	9	Machine Learning, AI, Data Science									x			x			x		x	x				x				x				x				x	x				x	x			x	x						x
	8	Streaming/ Realtime Processing									x			x						x						x		x						x							x											
	7	Interactive Processing	x	x								x					x	x	x	x	x	x	x	x	x			x	x	x	x	x	x	x		x	x				x	x										
	6	Batch Processing	x	x		x	x	x	x	x	x		x	x	x		x	x	x	x	x	x	x	x	x		x	x	x	x	x	x	x			x	x					x										
	5	Data Privacy/Security																																																		
	4	Data Governance/Mgmt																		x																																
	3	Data Storage	x	x		x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x								x	x									x	
	2	Communication & Connectivity				x																					x																									
	1	Cloud Services & HPC, Edge												x		x		x		x			x						x																							
			1999	2002	2004	2007		2008		2009	2010		2011	2012				2013						2014				2015								2016								2017							2018	

Figure 2. Matrix with Big Data Benchmarks

The **six** selected benchmarks are divided into two categories: micro-benchmarks and application benchmarks. The micro-benchmarks stress test very specific functionalities of the Big Data technologies typically utilising synthetic data. The application benchmarks cover a wider and more complex set of functionalities involving multiple Big Data technologies and typically utilising real scenario data. From a practitioner perspective, the two categories of benchmarks are quite different. The micro-benchmarks are easy to set up and use as they typically involve one technology, whereas the application benchmarks require more time to set up and involve more technologies in the execution phase.



Category	Year	Name	Type	Domain	Data Type
Micro-benchmarks	2010	HiBench	Micro-benchmark Suite	Micro-benchmarks, Machine Learning, SQL, Websearch, Graph, Streaming Benchmarks	Structured, Text, Web Graph
	2015	SparkBench	Micro-benchmark Suite	Machine Learning, Graph Computation, SQL, Streaming Application	Structured, Text, Web Graph
	2010	YCSB	Micro-benchmark	cloud OLTP operations	Structured
	2017	TPCx-IoT	Micro-benchmark	workloads on typical IoT Gateway systems	Structured, IoT
Application Benchmarks	2015	Yahoo Streaming Benchmark	Application Streaming Benchmark	advertisement analytics pipeline	Structured, Time Series
	2013	BigBench/TPCx-BB	Application End-to-end Benchmark	a fictional product retailer platform	Structured, Text, JSON logs
	2017	BigBench V2	Application End-to-end Benchmark	a fictional product retailer platform	Structured, Text, JSON logs
	2018	ABench (Work-in-Progress)	Big Data Architecture Stack Benchmark	set of different workloads	Structured, Text, JSON logs

Figure 3. Detailed Classification of the Associated Benchmarks

Figure 3 summarises the associated micro and application benchmarks that we are evaluating. They cover multiple domains and data types as listed in the last two table columns. HiBench [Huang et al., 2010], [HiBench Suite, 2018] is developed by Intel and is one of the very first Big Data benchmarks. SparkBench [Agrawal et al., 2015], [SparkBench, 2018] is developed by IBM and focuses on improving the Spark framework performance. YCSB (Yahoo! Cloud Serving Benchmark) [Cooper et al., 2010], [YCSB, 2018] is developed by Yahoo and targets cloud OLTP workloads typical for NoSQL systems. TPCx-IoT [TPCx-IoT, 2018] is the latest standardised benchmark by the TPC (Transaction Processing Performance Council) inspired by the YCSB benchmark but focusing on IoT Gateway systems. Yahoo Streaming Benchmark (YSB) [Chintapalli et al., 2016], [YSB, 2018] is a simple advertisement analytics pipeline application benchmark developed by Yahoo. BigBench [Ghazal et al., 2013], [BigBench, 2018] is an end-to-end, technology agnostic, application-level, analytics benchmark standardised as TPCx-BB by TPC. BigBench V2 [Ghazal et al, 2017] addresses some of the limitation of the BigBench (TPCx-BB) benchmark. ABench [Ivanov and Singhal, 2018] is a Big Data Architecture Stack benchmark based on BigBench and targeting to cover multiple Big Data application scenarios.

It is important to outline that implementations of all selected benchmarks are available for download and use, which will allow us to integrate and offer them as part of the DataBench Toolbox.

2.4 Business KPI derivation process

One of the major outcomes of DataBench is related to the possibility of deriving business KPIs from the results of technical benchmarks. However, this is not a clear process and there are very few attempts in the benchmarking community in this regard, with the exception, perhaps, of TPC[1], which gives the cost of infrastructure and energy consumption as part of their results. There is no general model to map business KPIs to technical metrics, and there is no consensus if a general model could be achieved without taking into account a vast amount of context knowledge. Therefore, the main issue related with the metrics/KPIs management is related to the mapping of technical and business KPIs. This business metrics derivation process is under consideration at the time of writing this document. Deriving business KPIs from technical metrics is something that could be very specific for particular scenarios, industries and even subject to many external variables.

As an example, a recommendation system for the retail industry has a potential business impact on cumulative margin. From a technical standpoint, it is data intensive and requires benchmarks assessing query throughput and response time. Technical costs can be significant and benchmark-driven technical choices can play an important role in increasing cumulative margin. In this case, the link between technical benchmarks and business KPIs is direct, as throughput can be translated into a cost-per-recommendation and, hence, into margin-per-recommendation. The business KPI derivation process of the DataBench Toolbox can provide a framework allowing companies to input information on their standard per-transaction margin and to calculate business KPIs with alternative technical choices.

As another example, in predictive manufacturing, streaming data from sensors monitoring a production plant are analysed to improve product quality in several ways, such as identifying anomalies that could generate defects and, thus, reducing scrap and rework at the same time. In this case, quality checks have to be performed in real time, consistently with the schedule of the production process. Technical benchmarking tools can help the correct design of the architecture managing streaming data, for example by comparing different technical alternatives, depending on the real-time requirements of a company's specific production plant. In this case, the right technical choice enables business returns, however, the link between technical costs and returns is not direct but mediated by the complexity of production processes and by the specific features of a plant. In addition to this, a reduction in scrap and rework can be difficult to translate into economic business returns and, as a matter of fact, companies may decide on the technical investment without measuring business returns. In this case, the business KPI derivation process of the DataBench Toolbox can provide a framework to understand the architectural alternatives and focus benchmarking on the most important features of the architecture, while the estimate of business returns may be more effectively aided by providing methodological guidelines (maybe with an online hypertextual/interactive/multimedia access to WP4 Handbook).

Therefore, this link between technical benchmarks and business KPIs strongly depends on the use case and, therefore, will be managed with a bottom-up approach. The initial idea is to enable derivation procedures based on very specific scenarios, technical metrics and

[1] http://www.tpc.org/

their relation to a few general business KPIs. If possible, the system will learn from past and new experiences to try to generalise the derivation process using a machine leaning approach. However, in many cases, the actual interpretation from the business perspective could not be implemented in a simple algorithm or even derived from past learning experiences. In those cases, the interpretation will be left to the business user after the careful Visualization of technical results along with other material, such as business reports from WP2. The research work that will be conducted in WP1 will provide a reference framework for understanding the relationship between business KPIs and technical benchmarks. The in-depth case study research conducted in WP4 will discuss different blueprints in detail, showing how the relationship between business KPIs and technical benchmarks is obtained in real cases and providing suggestions on how this knowledge can be embedded in the business KPI derivation process of the DataBench Toolbox to help both technical and business users.

In the initial phases of the DataBench project, WP1 will focus on the design of an integrated DataBench Framework, which, based on a continuous exchange of ideas and results, will be applied in WP4 to the desk analysis and to the in-depth case studies. This research work will result into a systematic analysis of the relationship between technical benchmarks and business KPIs, providing a classification of use cases as well as blueprints of how users can be guided in the selection of the most appropriate benchmarks and in relating benchmarking results to business returns. Figure 4 shows a preliminary set of blueprints tying business KPIs with technical benchmarks, used by the DataBench team as a basis for discussion.

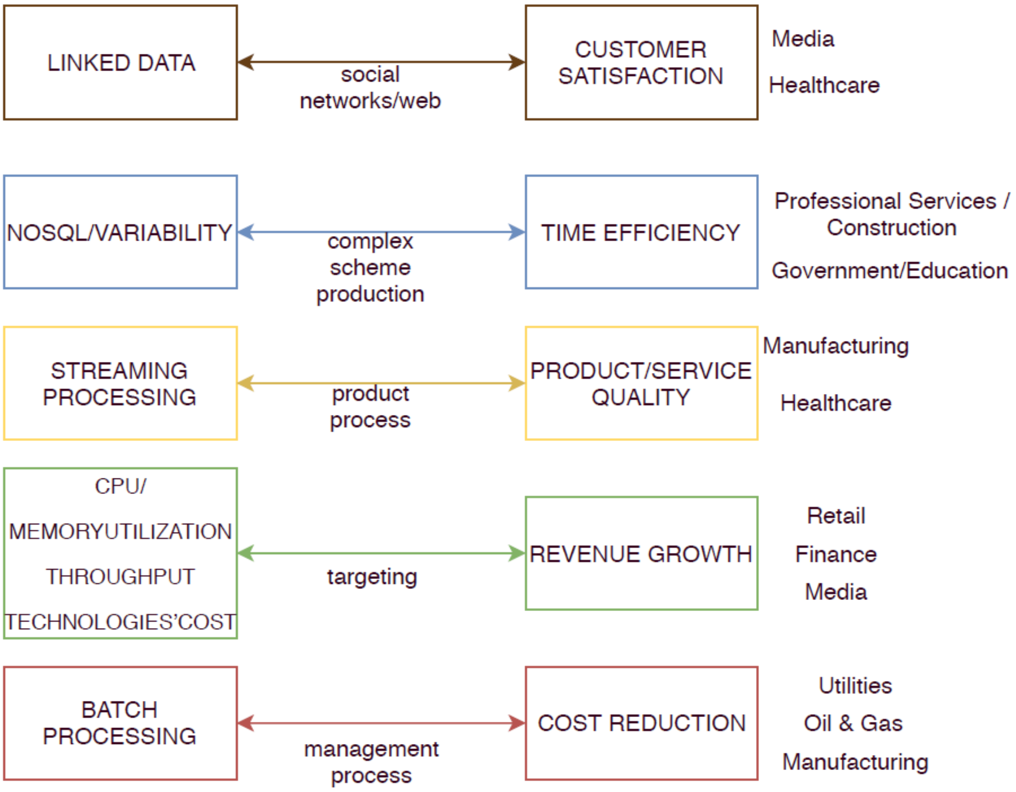


Figure 4. Preliminary set of blueprints tying business KPIs with technical benchmarks

As a matter of example, the following business KPIs give a hint of the difficulty to map technical results to business impact in a straightforward way: i) Cost reduction (it is very related to the platform itself, so therefore difficult to generalise), ii) time efficiency (it could be easy to measure the gain in time for some processes, but not so clear how this relates to the business objectives except in a few selected scenarios), iii) product/service quality (among others, scalability or fault tolerance are technical metrics that many benchmarks usually provide, but it is not clear how to relate them to quality, and they are measured in different ways depending of the benchmark); or iv) revenue growth (which is not really possible or simple to relate to a given system).

In summary, the derivation of business KPIs will involve a bottom-up, scenario, sector-based approach, and a top-down, generic, Machine Learning-based approach. We hope that the combination of these derivation process, where possible, with work on homogenisation, clustering and Visualization of technical KPIs, plus the usage of other non-technical results of the project (such as market material and the Handbook provided by WP1, WP2 and WP4), will give clear business insights to the DataBench users.

2.5 The Machine Learning framework

The purpose of the Machine Learning (ML) framework is to learn relationships between technical decisions and business outcomes for Big Data projects. This framework will recommend optimal, resource-efficient, objective-maximising technical infrastructure to achieve the big data-related goals of the organization.

Briefly, previously benchmarked technical projects are described by a variety of technical and business indicators ("features"), allowing statistical analysis and modelling of the most successful approaches to solving particular business-oriented big data problems. Initially, models will be trained and deployed based on a wide variety of openly available data repositories (the "boot-strapping" phase). As use-cases and technical survey results become available, metrics will be updated and iteratively incorporated into the ML modelling infrastructure, allowing the automated algorithms to provide more and more relevant recommendations on a wide variety of technical big data problem domains.

To get an initial set of annotated data, the project will investigate and mine different repositories such as the one provided by the Aloja[2] benchmarking framework, among others.

That is, the Machine Learning framework receives annotated data and technical infrastructure decisions, associated with business KPIs. Users may then query the system according to the type of project/goal, and their business indicators of interest. The system provides output recommending an optimal strategy for approaching the problem.

In general, there is an expectation that the Machine Learning framework will receive feedback on its recommendations from end users, in the form of technical indicator evaluation metrics.

As explained before, another potential application of the Machine Learning framework is related to the generalisation of a model to derive business KPIs. The problem here is the lack of data to train the model. We hope that in the course of the project, as initial

[2] https://aloja.bsc.es/

experiments provide examples of how business metrics can be derived from technical ones, this growing corpus will allow such an endeavour.

3. DataBench processes

There are different ways of representing the functionality of the software without being finished, one of these is the Unified Modelling Language (UML), which is the software modelling system most known and used today. It is composed of various graphic elements that combine to form diagrams. It is also supported by the Object Management Group (OMG) that is dedicated to the care and establishment of standards of object-oriented technologies.

Within UML there are different types of diagrams that allow representing the different perspectives of a system, which is also known as a model that is a simplified representation of reality. The Use Cases are diagrams that allow representing what the system will do but not how it works.

This section aims to provide the UML Use Case diagrams, their components and a brief explanation of the expected processes to be carried out within the entire DataBench toolset (Toolbox, AI, etc.). It is therefore a high-level step-by-step description of what DataBench could do for the target users.

3.1 DataBench actors

To allow a correct representation of the model, the first step consists of the identification of the users of the systems, known as actors in UML. The actors involved in the DataBench processes are the following:

- **DataBench Admin,** who adds existing or new benchmarks to the Toolbox and manages the server-side system.
- **Technical Users**, who can be developers, administrators or testers that would like to perform a technical benchmark of a given system/app. These are the actual users of the final system.
- **Business Users**. Like in the previous case, these are end users that would like to benchmark their system or app. However, in this case they do not perform the actual benchmark, but they are interested in understanding the underlying business implications. These users will rather get into the web tool to search for past executions, getting recommendations, business indicators, etc.
- **Benchmark Providers.** Typically, developers or providers of big data benchmarks (i.e. people behind HiBench). They will be able to add or update benchmarks to the Toolbox.

3.2 DataBench UML Use Cases

Considering that a “use case” is a description of the actions of a system from the user point of view, this subsection is in charge of listing the different use cases identified in the context of the DataBench project, taking as a reference what is specified in the DoA.

The following subsections describe the DataBench use case diagrams that aim to model the functionality of the DataBench Toolbox using actors and use case diagrams. The use cases will become services or functions that will be provided by DataBench to the target users.

3.2.1 DataBench Accessing use case

The DataBench accessing use case diagram shown in Figure 5 is in charge of managing the access of users to the DataBench platform.

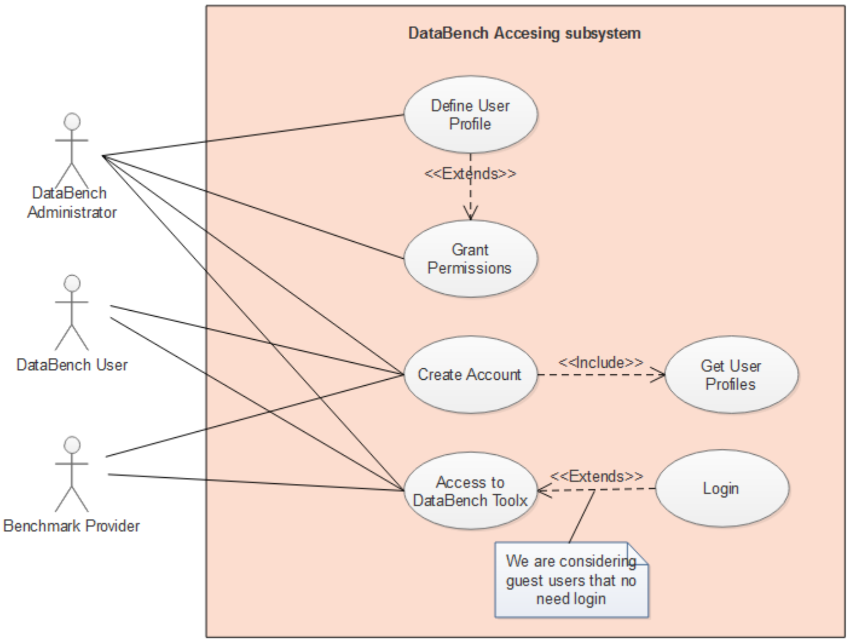


Figure 5. Accessing use case diagram

The processes or functionalities identified are the following:

- Define User Profile: The process allows the creation of a user profile following the typology of the actors of the system: Administrator, Technical User, Business User or Benchmark Provider.
- Grant Permissions: The process will allow the DataBench Administrator to grant a set of permissions to a certain user profile.
- Create Account: The process establishes the mechanism for creating an account and associate it to any kind of user profile.
- Access to DataBench Toolbox: This functionality will support the access to the system. Access to some parts of the system will be private and will include a "Login" process. For other public parts of the system, it is expected that login will not be necessary, therefore allowing guest users.
- Get User Profiles: The process aims to get the different types of user profiles available in the system.

3.2.2 DataBench KPIs management use case

The DataBench use case diagram related to technical and business KPIs management is shown in Figure 6.

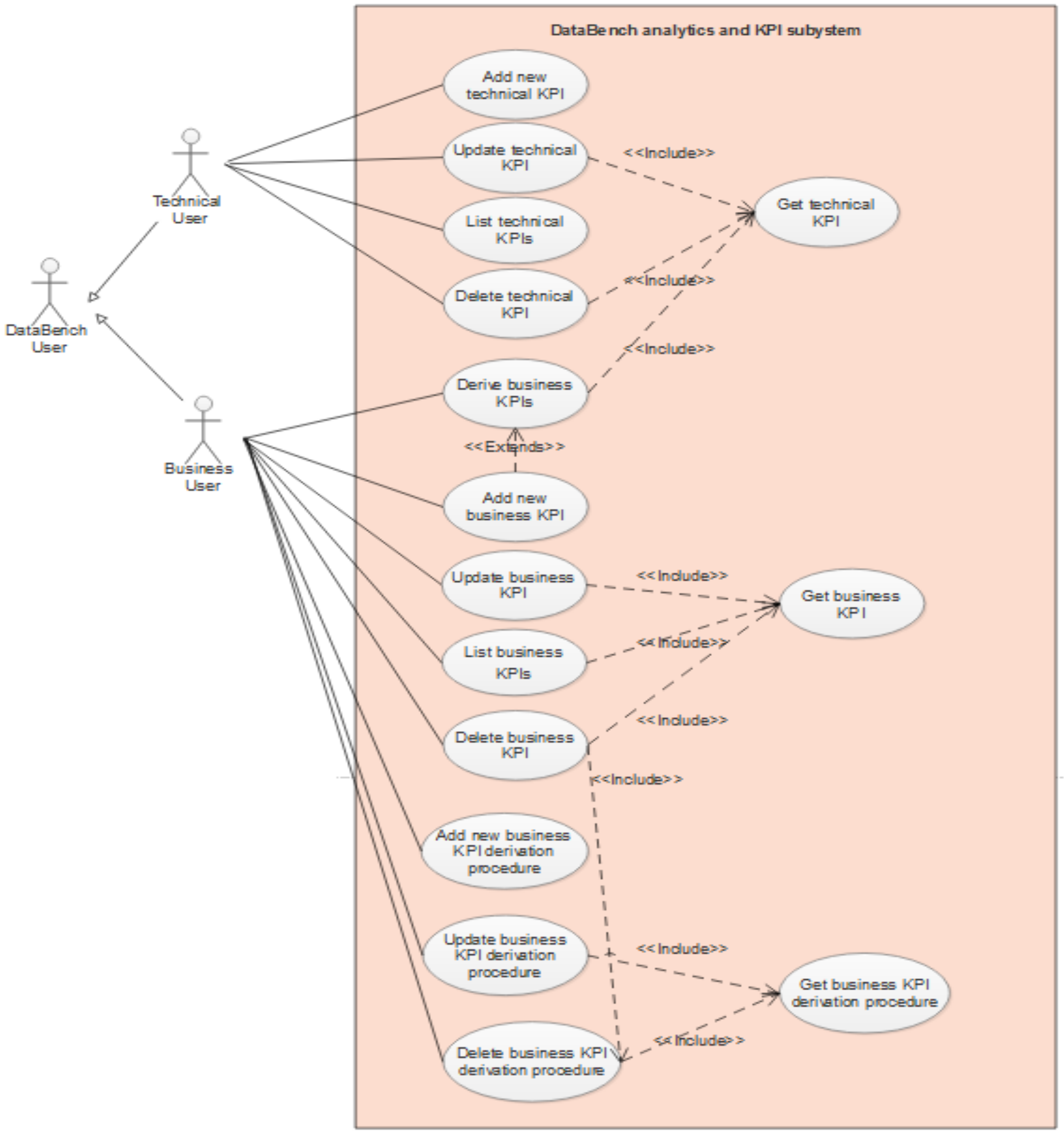


Figure 6. Technical and business KIPs management use case diagram

The processes or functionalities identified could be classified in three main groups from a business point of view:

- Technical KPIs management: This group of functionalities involves the processes of setting and managing the technical KPIs used by the existing benchmark in a common DataBench KPI framework. It will provide functionalities for i) adding a new technical KPI, ii) update an existing technical KPI, iii) listing technical KPIs associated to a concrete benchmark, iv) get an individual technical KPIs and finally v) delete a technical KPI. The Technical User is the actor that performs these functionalities. In addition, a harmonisation of the technical results is carried out as part of setup and runtime use case (see subsection 3.2.3) to resolve the mappings between the technical KPIs from a given benchmark to the similar KPIs in DataBench.
- Business KPIs management: This group of functionalities involves the processes of i) derivation of a new business KPI using the associated derivation procedure, ii) adding a new business KPI, iii) update an existing business KPIs, iv) listing business KPIs associated to a concrete benchmark, v) get an individual business KPI and,

finally, v) delete a business KPI. The Business User is the actor involved in these scenarios.

- KPIs derivation procedures management: This group of processes is covering the i) definition of a new derivation procedure, ii) the update of an existing derivation procedure and finally iii) the deletion of a derivation procedure. As in the case of business KPIs management, the Business User is the actor involved in these scenarios.

3.2.3 DataBench User intentions and Set up/Runtime use cases

Figure 7 shows a combined use case diagram that reflects the relationship between the User intentions and Set up/runtime use cases.

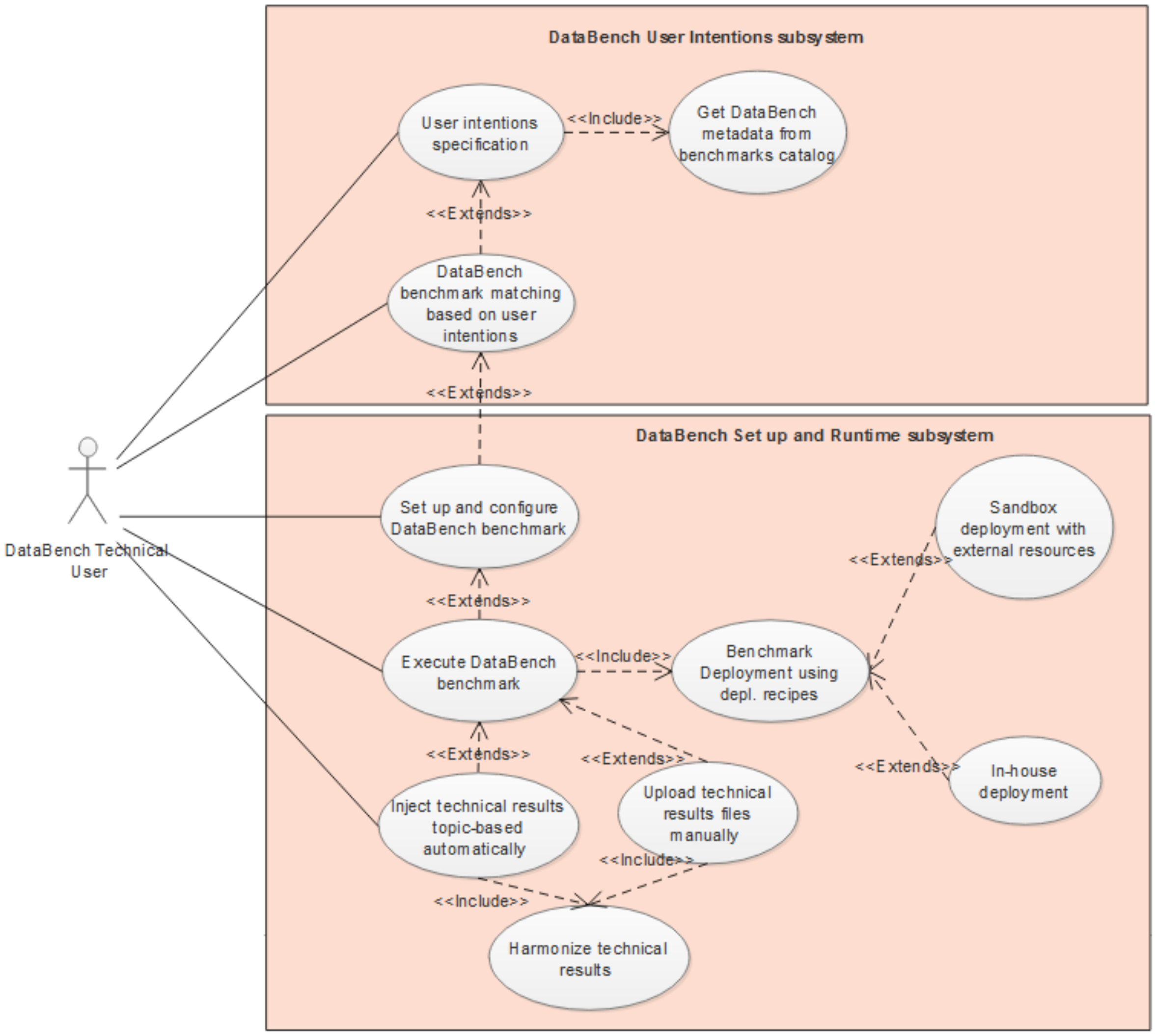

DataBench User Intentions subsystem
User intentions specification
<<Include>>
Get DataBench metadata from benchmarks catalog
<<Extends>>
DataBench benchmark matching based on user intentions
<<Extends>>
DataBench Set up and Runtime subsystem
DataBench Technical User
Set up and configure DataBench benchmark
Sandbox deployment with external resources
<<Extends>>
<<Extends>>
Execute DataBench benchmark
<<Include>>
Benchmark Deployment using depl. recipes
<<Extends>>
<<Extends>>
<<Extends>>
In-house deployment
Inject technical results topic-based automatically
Upload technical results files manually
<<Include>>
<<Include>>
Harmonize technical results

Figure 7. User intentions and Set up/Runtime use cases

By User intentions, this process means the identification and categorization of the most suitable benchmark that fits with the searching criteria specified by the Technical User. These criteria refer to the Big Data horizontal area, vertical domain and data type and will be extended with a set of goal metrics related to the purpose of the benchmark.

On the one hand, the User intentions use case aims at defining the processes related to help the end users of the DataBench Toolbox to search and select the right benchmarks that suit their needs. The use case includes the processes for i) user intentions' specification and ii) DataBench benchmark matching based on the user intentions. To specify the user intentions, the specification process could be extended performing some queries over the benchmark catalogue in order to get the associated metadata to support the formalization of the intentions. This process may also take advantage of the work done in WP5 using Artificial Intelligence (AI) techniques to deliver recommendations based on specific features.

On the other hand, the Set up/Runtime use case encompasses the processes of i) setting up and configure a DataBench associated benchmark, ii) executing a concrete benchmark, iii) the process of injection of results of the executed benchmark either automatically or manually by means of updating a set of result files and finally iv) the harmonization of the technical results in a DataBench format.

The execution process could be extended using two different deployment processes:

1. Sandbox deployment: This approach will help users to deploy the DataBench benchmark and their applications into a testing environment outside their own organization. Following this approach, a Technical User should communicate the necessary info to the DataBench platform to let the Toolbox know the details of the infrastructure where the deployment will be done (i.e. servers, credentials, etc.) This is useful, for instance, for users that would like to benchmark something in cloud resources (Amazon, Azure, Google, etc.). The user should declare to the Toolbox the hosts and infrastructure where the deployment should be done at the server side.
2. In-house deployment: This process will provide the means to download the Toolbox as an executable client-side framework to the infrastructure of the user (i.e. their production or testing in-house infrastructure, a private cloud, etc.). This can be done for instance by preparing an Ansible3 playbook, using a client user interface (i.e. a dashboard, wizard, or scripts) to set it up in the local resources, execute the playbook and allow running several benchmark rounds.

In both cases, the Toolbox will provide the means to get the results of the runs of the benchmarks back to the Toolbox. The user may decide which results to upload to DataBench, but the idea is to enable and recommend the automatic collection of results as preferred option. This is important to be able to access more advanced features, such as the publication of results into the DataBench platform, and to access more information about potential business KPIs.

3.2.4 DataBench Visualization and Reporting use case

The DataBench visualization and reporting use case diagram shown in Figure 8 is in charge of covering the visualization of the benchmark execution results and derived business metrics.

[3] https://www.ansible.com/

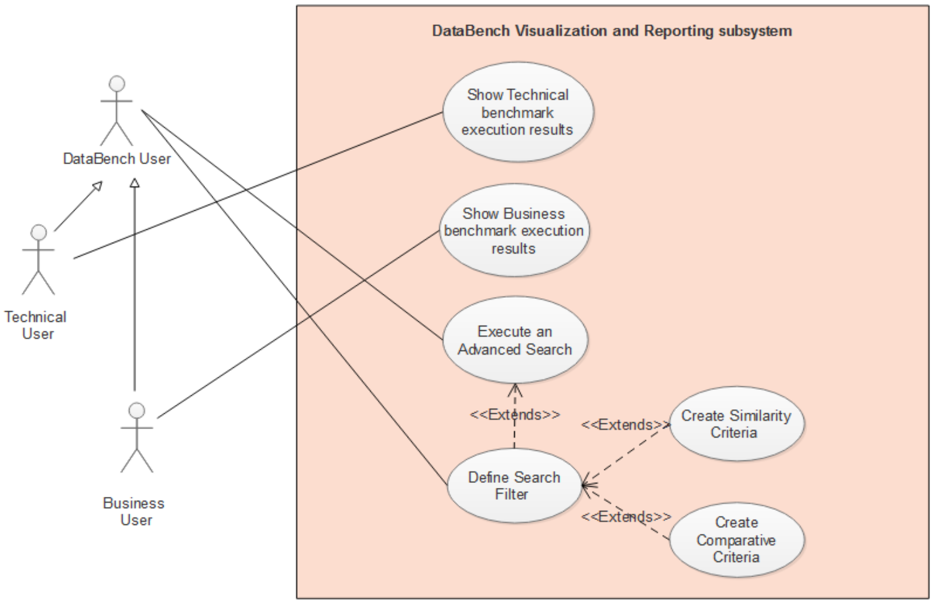

DataBench Visualization and Reporting subsystem
DataBench User
Technical User
Business User
Show Technical benchmark execution results
Show Business benchmark execution results
Execute an Advanced Search
<<Extends>>
Define Search Filter
<<Extends>>
Create Similarity Criteria
<<Extends>>
Create Comparative Criteria

Figure 8. Visualization and Reporting use case diagram

This use case relies on the communication of the results of the execution of the benchmarks explained in the previous use case. The Technical User will be able to visualize the technical results while the Business User will be able to access to potential derived business metrics. In both processes the Technical User and the Business User could look for other public results and define filters based on similarity or comparative criteria in the process of querying the results.

3.2.5 DataBench Benchmark Management use case

This use case provides an overview of the different processes related to the management of the benchmarks and metrics handled in DataBench. As such, it offers processes for adding, updating or deleting benchmarks to the Toolbox along with the necessary metadata to describe or deploy them. It also enables the population of the catalogues of technical and business KPIs and the potential definition of derivation procedures. In summary, this use case is fundamental to provide extensibility to the Toolbox. The use case diagram related to Benchmarks management is shown in Figure 9.

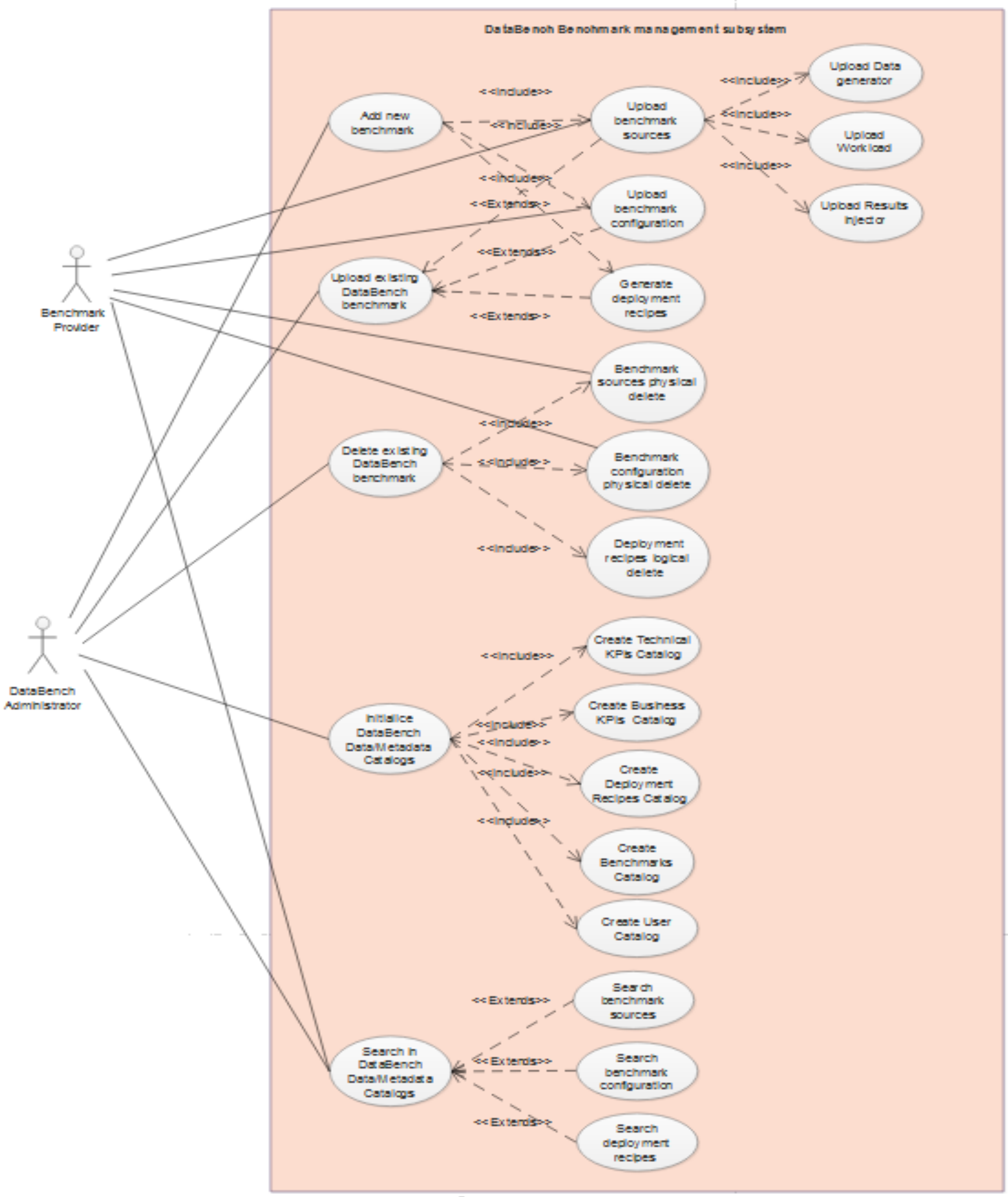

DataBench Benchmark management subsystem
Benchmark Provider
DataBench Administrator
Add new benchmark
<<include>>
<<include>>
Upload benchmark sources
<<include>>
<<include>>
<<include>>
Upload Data generator
Upload Workload
Upload Results Injector
<<include>>
<<Extends>>
Upload benchmark configuration
<<Extends>>
Upload existing DataBench benchmark
Generate deployment recipes
<<Extends>>
Benchmark sources physical delete
<<include>>
Delete existing DataBench benchmark
<<include>>
Benchmark configuration physical delete
<<include>>
Deployment recipes logical delete
<<include>>
Create Technical KPIs Catalog
<<include>>
Create Business KPIs Catalog
Initialice DataBench Data/Metadata Catalogs
<<include>>
<<include>>
Create Deployment Recipes Catalog
<<include>>
Create Benchmarks Catalog
Create User Catalog
<<Extends>>
Search benchmark sources
Search in DataBench Data/Metadata Catalogs
<<Extends>>
Search benchmark configuration
<<Extends>>
Search deployment recipes

Figure 9. DataBench Benchmark management use case diagram

Next are listed the following processes or functionalities identified:

- Add new Benchmark: This process is in charge of uploading into the DataBench platform all the necessary resources for a subsequent execution of the benchmark. These resources include:
  - Benchmark sources: These sources include i) Data Generator, ii) Workloads and iii) Results Injector.
  - Benchmark configuration
  - Deployment Recipes

- Update a Benchmark: The process allows updating the sources, configuration or/and deployment recipes associated to a concrete benchmark.
- Delete a Benchmark: This functionality gives to the DataBench Admin the possibility to delete an existing benchmark and the associated resources from the DataBench platform.
- Initialize Data/metadata catalogues: The process enables the DataBench Admin to initialize the different stores or catalogues defined into the DataBench platform as: Technical KPIs catalogue, Business KPIs catalogue, Deployment Recipes catalogue, Benchmark catalogue and Users catalogue.
- Search in Data/metadata catalogues: This functionality lets the Benchmark Provider and DataBench Admin perform queries to get sources, configuration or deployment recipes of a registered benchmark.

3.3 DataBench components

The diagrams in UML are classified according to their structure or behaviour, the latter refer to the way in which instructions are executed or how activities are given within the system. The structure diagrams define how the software is structured. An example of this is the component diagram that defines the different parts that make up the system to work.

This section aims to provide the DataBench component diagram showing the different bits and pieces, and subsystems to be explained afterwards in the following sub-sections. The global DataBench component diagram is shown in Figure 10.

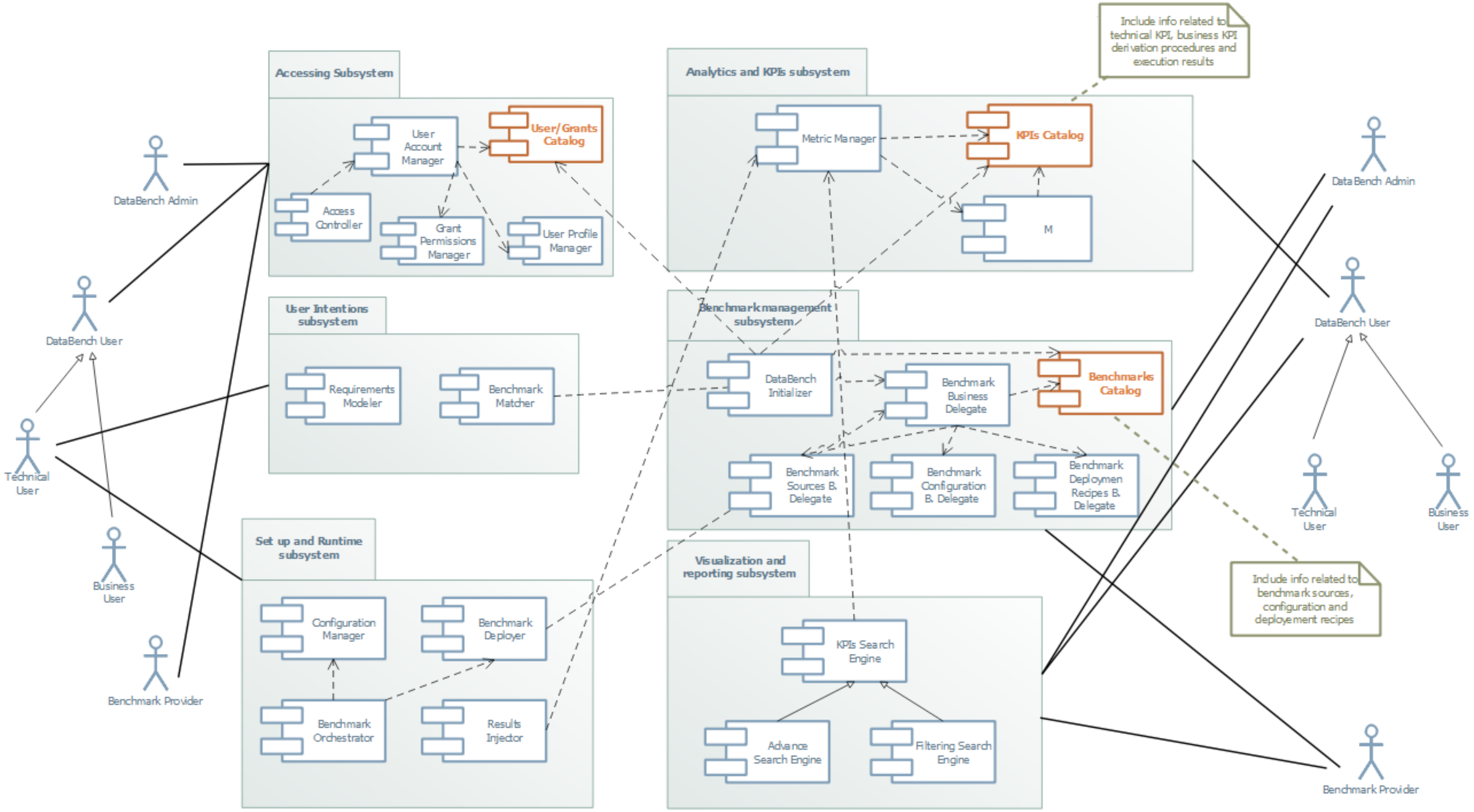


Figure 10. DataBench components diagram

3.3.1 Accessing Subsystem

The accessing subsystem shown in Figure 11 will be in charge of the authentication, access authorization and auditing. To support these functionalities the accessing subsystem is composed of five software components:

- User Account Manager: This component will allow the DataBench Admin, DataBench User or DataBench Provider to carry out one of the following actions: i) create an account, ii) update information associated with an account, iii) get all the information associated to a concrete account or finally iv) delete an account.
- Access Controller: This asset is in charge of allowing or denying the access to the Toolbox to any of the actors defined in section 3.1. The controller will communicate with the user account manager to obtain the information associated to the user profile regarding grants.
- Grant Permissions Manager: This component will let the DataBench Admin granting a set of permissions to a user profile. Initially, we have in mind three types of permissions over the actions and assets that will be defined into the Toolbox: read, write and execute.
- User Profile Manager: This asset will allow the DataBench Admin to perform the CRUD (Create, Read, Update, and Delete) operations for managing the user profiles in the Toolbox.
- User/Grants Catalogue: This asset is an internal storage in which the data related to user accounts, user profiles and grants associated with the profiles will be stored.

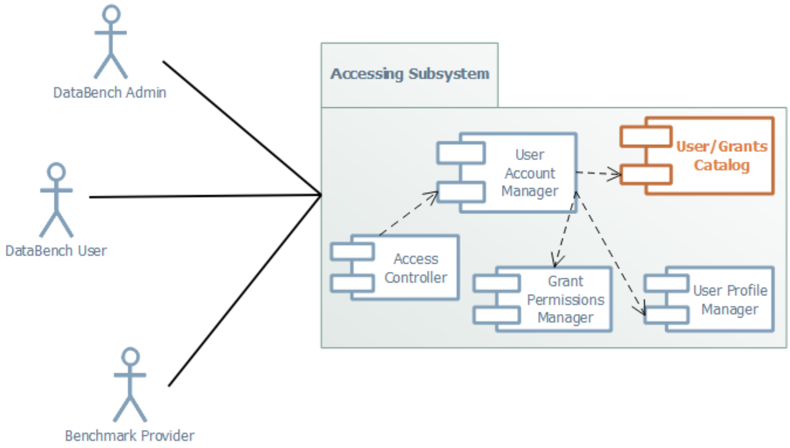


Figure 11. Accessing subsystem

The access control approach defined for DataBench covers access approval, implementing the system the decision to grant or reject an access request from an already authenticated actor, based on what the actor is authorized to access. The authorized actions will be represented by means of the user profiles and the grants associated to them.

Finally, in DataBench the authentication and access control will be combined in a single operation and the access will be approved based on successful authentication or based on an anonymous access in the case of guest users.

3.3.2 User Intentions Subsystem

The User Intentions subsystem shown in Figure 12 will be in charge of enabling the users to specify their requirements related to the usage of a big data benchmark. To support these functionalities the User Intentions subsystem is composed of two software components:

- Requirements Modeler: This component will allow technical users to define a set of initial requirements to select the most appropriate benchmark to suit their needs. Initially, the intentions of the user will be aligned to the different layers of the BDVA Reference Model specified in the BDVA Strategic Research & Innovation Agenda (SRIA)[4]. This means that the intentions should specify the concrete Big Data horizontal area, vertical domain and data type. This initial set will be extended and aligned with the results of WP5 in terms of recommendation engine based on Machine Learning, and with a set of goal metrics related to the purpose of the benchmark.
- Benchmark Matcher: It will be in charge of covering the matching process between the user requirements and the benchmark registered in the Benchmarks catalogue (see section 3.3.5). The Matcher will provide a set of benchmarks that fits with the User Intentions specified.

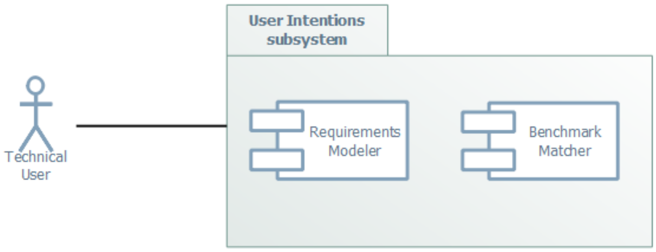


Figure 12. User Intentions subsystem

The selection of the benchmark will be done by the Technical User once the candidate benchmarks have been listed by means of the user interface that will be provided in the Toolbox.

Finally, a decision tree or other Machine Learning approaches will be explored as technical implementation for the Benchmark Mapper. For instance, in the case of decision trees, the variables proposed for modelling the user requirements coming from the BDVA Reference Model could be used as decision nodes.

3.3.3 Setup and Runtime Subsystem

The Setup and Runtime subsystem shown in Figure 13 will be in charge of supporting the overall workflow associated to the deployment and execution of one of the benchmark registered in the Toolbox. The workflow will include three main steps, namely i) setup and configuration; ii) deployment and execution; and finally, iii) injection of the results into the

[4] http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf

Toolbox. To support these functionalities, the setup and runtime subsystem is composed of four software components:

- Configuration Manager: This component will allow a Technical User to upload a set of configuration parameters associated to the benchmark that aims to execute.
- Benchmark Orchestrator: It is the orchestrator engine in charge of deploying and executing the benchmark taking the configuration parameters defined by the Technical User. The execution of the benchmark could be explicitly performed automatically or, by contrary, fired manually from the Technical User from the benchmark environment deployed.
- Benchmark Deployer: This asset is in charge of the physical provision of the execution environment request by the Benchmark executor as part of a benchmark execution.
- Results Injector: This component is in charge of injecting the results of the execution of the benchmark runs into the Toolbox, either fired by the user manually or automatically, as last step of the benchmark execution workflow. Each benchmark in DataBench should have their own instance of this module tailored to get the results from the specific format of the benchmark into DataBench.

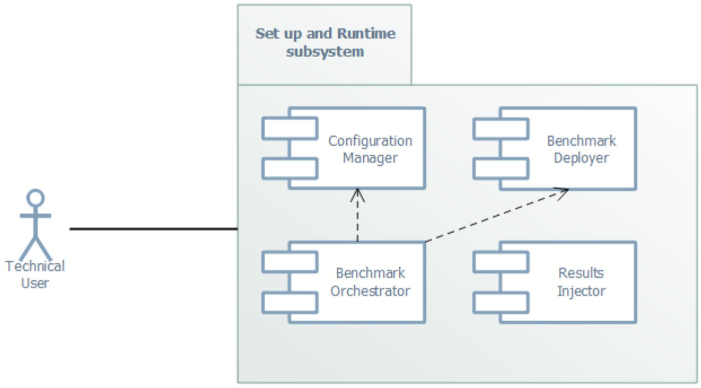


Figure 13. Set up and runtime subsystem

It is important to mention that DataBench does not provide infrastructure for executing the benchmarks, but rather a centralized platform from where the Toolbox can be accessed via web user interface enabling two potential scenarios to cover the deployment and execution steps after the selection of a given benchmark:

1. Sandbox deployment - Execution in public resources. This scenario is useful for users and organizations that would like to benchmark a big data solution in public resources or infrastructure (i.e. a public cloud) with no big security or privacy constraints. In this case, the Technical User should upload some details about the execution environment to the DataBench Toolbox through the web user interface to let it know where the deployment will take place (i.e. the hosts, credentials, etc.).
2. In-house secure deployment and execution: In many cases, users and organizations would like to run the benchmark in their own infrastructure or in any secure infrastructure. Therefore, giving the details of their internal secure environment to the DataBench Toolbox in our main central deployment is not desirable or even

feasible for users. In this case, as it is common in the majority of the benchmarking frameworks, the DataBench Toolbox will offer the possibility to download a version of the Toolbox with the necessary functionality to deploy and run the benchmarks in the user infrastructure, and to eventually return the results of the benchmarking runs to DataBench central platform. For instance, the Toolbox could be downloaded as an Ansible playbook, providing a client-side user interface to set the benchmark up in the local resources and run several benchmark rounds. The Toolbox will offer the means to get the results of the executing back to the server-side counterpart of the Toolbox, ether manually or automatically (preferred option).

Each benchmarking system provides different ways to retrieve results from the benchmarking runs, ranging from logs or other type of files to databases, plus even some visualization tools. DataBench is not aiming at replacing them, but rather to get the data into the DataBench Toolbox, harmonize it and offer added value services for comparison, KPI abstraction or business insights. In order to do that, the results from the execution rounds should be gathered. The initial step for each benchmark present in the DataBench Toolbox is therefore the creation of a connector to give their results back. During the project life-time, a set of connectors and transformation rules for some popular benchmarks will be provided, along with the guidelines to extend these connectors for new benchmarks in an easy way.

As mentioned above, it is important to get the results, so we will foster the adoption of an automatic injection mechanism where possible. This might be implemented, for instance, by defining an Apache Kafka[5] topic where the connector will publish either automatically or manually the results. For each benchmark a Kafka Producer will be provided to get the results in the appropriate format to the Kafka Topic from where the DataBench Toolbox will be listening. Other approaches for manual injection will be also checked.

3.3.4 Analytics and KPI Management Subsystem

The Analytics and KPIs management subsystem shown in Figure 14 will be in charge of supporting the management of technical and business metrics and the associated derivation procedures. To support these functionalities the KPIs management subsystem is composed of three software components:

- Metrics Manager: This component will allow Technical Users to perform the CRUD operations for managing the technical and business metrics in the Toolbox.
- Metrics Spawning: This component will take the technical metrics, analyse them and, where possible, spawn relevant business metrics from them.
- KPIs Catalogue: This component will provide the storage means to all the features related to technical and business metrics as well as the associated derivation procedures.

[5] https://kafka.apache.org/

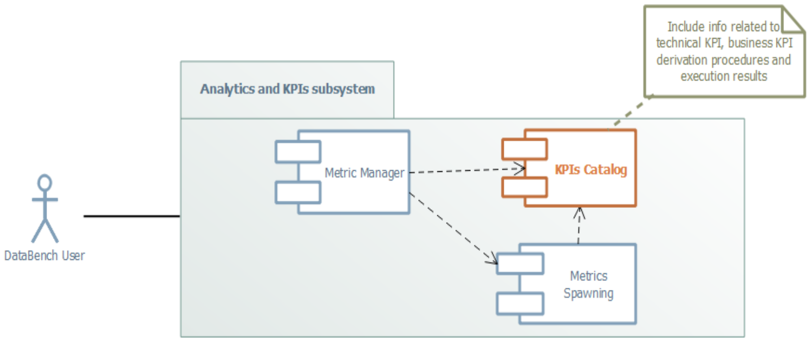

Include info related to technical KPI, business KPI derivation procedures and execution results
Analytics and KPIs subsystem
Metric Manager
KPIs Catalog
DataBench User
Metrics Spawning

Figure 14. KPIs management subsystem

As explained in section 2.4 the derivation of business KPIs from the results of technical benchmarks is not a straightforward process. It will be based mainly on a scenario-based bottom-up approach, but with the aim to generalize the procedures by making use of Machine Learning, visualization and other techniques, plus the usage of other non-technical results of the project (such as market material and the Handbook provided by WP2 and WP4).

3.3.5 Benchmark Management Subsystem

The Benchmark management subsystem shown in Figure 15 will be in charge of supporting the data and metadata supply related to a non-registered benchmark into the Toolbox.

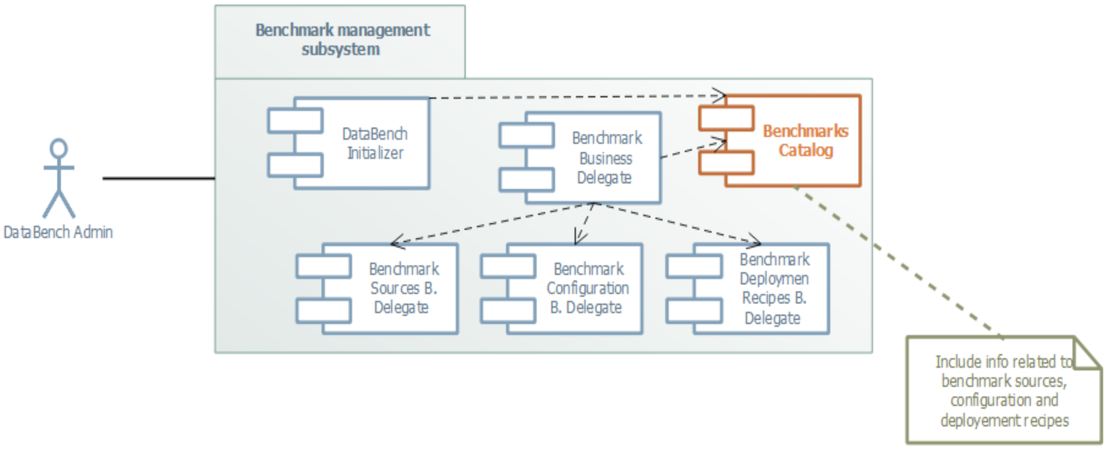

Benchmark management subsystem
DataBench Initializer
Benchmark Business Delegate
Benchmarks Catalog
DataBench Admin
Benchmark Sources B. Delegate
Benchmark Configuration B. Delegate
Benchmark Deploymen Recipes B. Delegate
Include info related to benchmark sources, configuration and deployement recipes

Figure 15. Benchmark management subsystem

To support these functionalities, the Benchmark management subsystem is composed of six software components:

- DataBench Initializer: This component will allow the DataBench Admin to perform the initialization of the main storages defined in the Toolbox (User catalogue, Metrics/KPIs catalogue and Benchmarks catalogue).

- Benchmark Business Delegate: This component will be in charge of delegating the business actions of the benchmark sources, configuration and deployment recipes management to the appropriate business object that implements the functional logic.
- Benchmark Sources Business Delegate: This component is responsible for the benchmark sources management that will be composed of i) data generator, ii) workloads and, finally, iii) a result injector.
- Benchmark Configuration Business Delegate: This component is responsible for the benchmark configuration management.
- Benchmark Deployment Recipes Business Delegate: This component is responsible for the benchmark deployment recipes. As a tentative approach, the recipes could be an Ansible playbook.
- Benchmark Catalogue: This component will be supporting the storage of all the data and metadata related to the registered benchmark. This information will be related to the benchmark sources, configuration and deployment recipes.

3.3.6 Visualization and Reporting Subsystem

The Visualization and Reporting subsystem shown in Figure 16 will be in charge of supporting the searching over the technical and derived business metrics related to a benchmark execution.

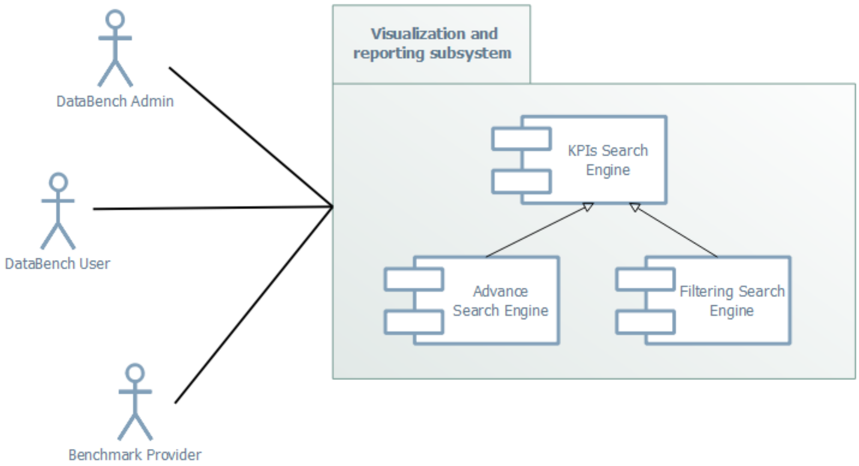


Figure 16. Visualization and Reporting subsystem

To support these functionalities, the Visualization subsystem is composed of three software components:

- KPIs Search Engine: This component will allow searching on the results storage.
- Advance Search Engine: This component will be extending the search functionality of the search engine providing mechanisms to perform advance queries.
- Filtering Search Engine: In the same way as the Advance Search, this component will be extending the search engine with filtering search.

4. DataBench Toolbox architecture

In the following sections we aim to provide a high-level overview of the proposed architecture of the DataBench Toolbox.

First, in a technology agnostic manner we will present a functional description of the main modules that will form the Toolbox as well as their interconnection pointing out the main functionality of each one.

After that, we will relate the BDVA reference model with the described Toolbox Architecture and identify the existing gaps.

Finally, we will present a potential implementation of the functional architecture in a more technical way, describing in detail each of the modules with their possible implementation and the technologies behind them.

4.1 Functional overview of the DataBench Toolbox

One of the objectives of the DataBench project is to provide an easy-to-use big data benchmarking Toolbox that will give users the opportunity to gather metrics of their systems.

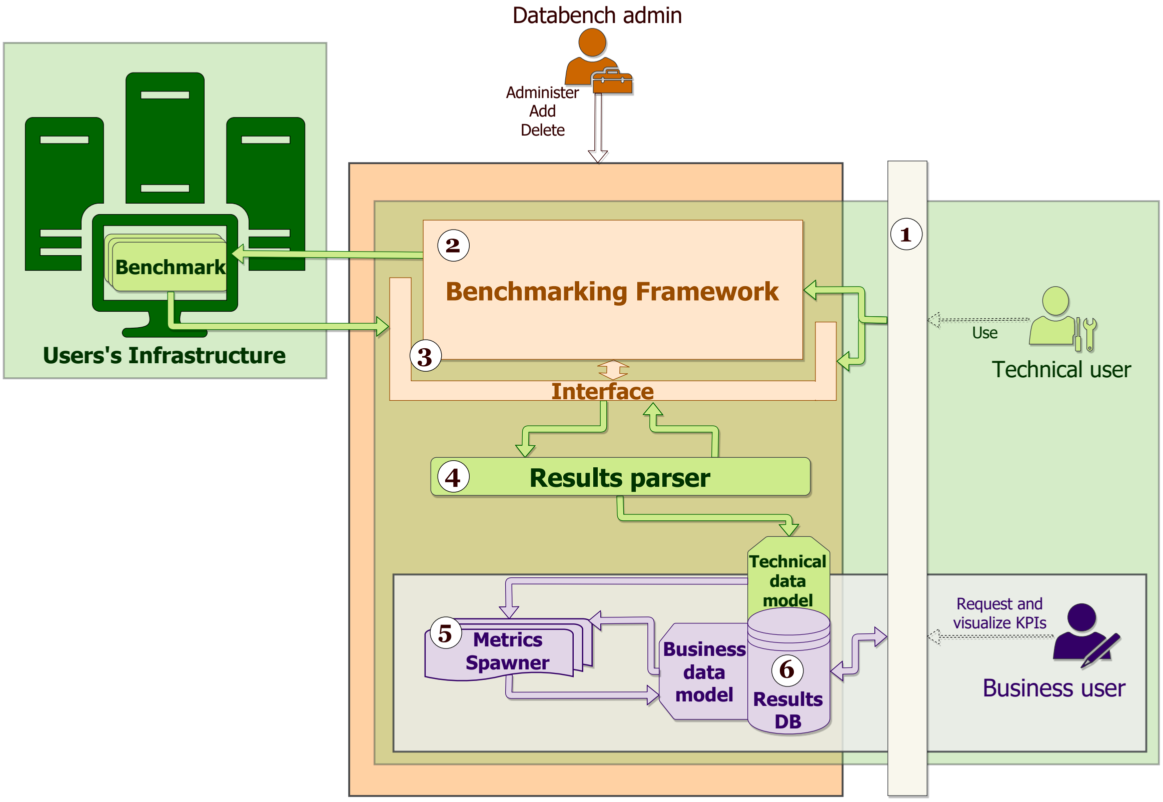


Figure 17. Functional overview of the framework architecture

To achieve this, we propose a modular framework based on templates which will be complemented with a web interface from where the user can decide and choose the metrics needed and which will also act as a dashboard where the results of the executions will be gathered and shown to the user, as can be seen in Figure 17.

Our proposed Architecture is composed of the following interconnected modules:

- Web interface;
- Benchmark framework;
- Results interface;
- Results parser;
- Metrics spawner;
- Results DB.

We will dive in the details of each module in the following sections.

4.1.1 Web interface

As explained before, the main point of access to the benchmark tool will be the web interface. This web interface, as shown in point 1 of Figure 17, will be connected to the backend of the Toolbox. This web interface module will provide different users of the tool with the functionality to choose which benchmarks they want to run. It will also be in charge of providing a layer of configuration that the users can fill in to preconfigure the templates and the benchmarks to be run later on.

As explained in section 3.2.3, the web interface will provide two different scenarios:

- First, it will be in charge of providing a layer of configuration that the users can fill in to preconfigure the templates and the benchmarks, so they can be run from the server on the configured instances.
- Second, it will act as an entry point from where the user can download the Toolbox for running it in-house, being the user the one in charge of managing all the installation and configuration of the Toolbox.

Moreover, the web module will be used to show in a dashboard the results of the executions and the derived metrics and business insights as an added value for those organizations that finalize the feedback loop by giving back to the Toolbox the results of their benchmark runs.

4.1.2 Benchmarking framework

This web interface will be connected to the backend of the Toolbox, called Benchmarking Framework in point 2 of the Figure 17, where the templates and recipes to run the benchmarks will be stored. The idea of this module is to act as a repository of recipes that will be later used to generate a template that will be provided to the Technical User for them to configure it according to the system where they want to run the benchmarks on. This module will be the main point of interaction for the administrator with the Benchmarking Framework, since he will be in charge of handling the integration, addition and deletion of the new, updated or modified benchmarks.

With this module, we can also have the functionality to run some of the required benchmarks in a cloud provider, given the credentials and the correct configuration. This would allow users without a physical cluster to run the different benchmarks on cloud and to decide the configuration of their infrastructure before provisioning it.

Moreover, as stated in the previous sections, the functionality of the framework will also be available for download, letting the user run it at the client side. Leaving the installation and the configuration of the framework to the Technical User.

4.1.3 Results interface

The results of the runs will later on be transferred back to the Benchmarking Framework through the results interface, point 3 in Figure 17, on the server side to be parsed into the defined technical data model. The project will provide this interface for the results to be transferred to the framework either automatically by the benchmark run or manually by the user.

This interface should be the interconnection point for any benchmark result added to the tool. In this way, we will have a single point of connection for the result parsers to read the data from simplifying the whole architecture of the Framework.

4.1.4 Results parser

Since each benchmark returns the results in a different way, we need to define a module capable of converting these heterogeneous results into a standardized data model. This will allow the Framework to have a consistent set of data among as many benchmarks as possible to make the calculation of the business metrics as easy as possible on the next steps.

The result parser module is going to be closely related to the results interface and will work in collaboration with it. It will be open enough to parse any data that can be gathered through the results interface.

Moreover, these result parser modules need to be easily integrated into the platform to allow any benchmark provider to generate its own parser. For this task we will generate and document the interfaces to communicate this module with the results interface and the results DB in an open way, so that we do not constrain the providers to any specific technology.

4.1.5 Metrics Spawning

In order to derive the requested KPIs from the technical results of the benchmarks, we will need another dedicated module in the Architecture.

This module will be mainly connected to the results DB module, so it can parse the corresponding results from the technical data model and calculate the defined KPIs and at the end, write them back to the results DB.

As it was hinted in section 3.2.2 the process of deriving business metrics is still under discussion at the time of writing this document. In our current view, business metrics depend not only of technical metrics gathered in the Results Interface module, but of the context. The project will therefore provide a bottom-up approach by studying specific business metrics for clear business scenarios. We foresee that some specific metrics could be calculated, but most of them will require the assessment of the business user. Therefore, the results or the Metrics Spawning component will be in many cases a combination of different methods, ranging from specific algorithms, to machine learning approaches (learned from previous or new cases), technical metrics homogenization, comparison and visualization techniques, as well as the help of the documentation around big data market trends and the DataBench Handbook to be delivered in the scope of other Work Packages.

In the same way as the result parsers, since new data can be added to the Toolbox, it needs to be open to be extended with new KPIs and calculations.

4.1.6 Results DB

The framework will need to provide a place where the Result Parsers can store the data into and also have a place from where the web interface can read the results to show them in the dashboard. It will be the results DB module.

This module needs to be complemented with at least a technical data model defined and an interface, so that the rest of the modules can make use of it. The technical data model needs to be general and open enough to fit all the results of the benchmarks into it, as well as allow an easy extension to house any new benchmark added to the platform.

4.2 Relation to the BDVA Reference Model

Based on our evaluation and first-hand experience with some of the benchmarks, we selected **six** initial benchmarks for extensive testing and integration in the DataBench Toolbox. Figure 18 depicts them as part of the different BDVA Reference Model layers that they cover. Clearly, one of the key requirements is to achieve a wide coverage of the Big Data technology layers in the BDVA architecture.

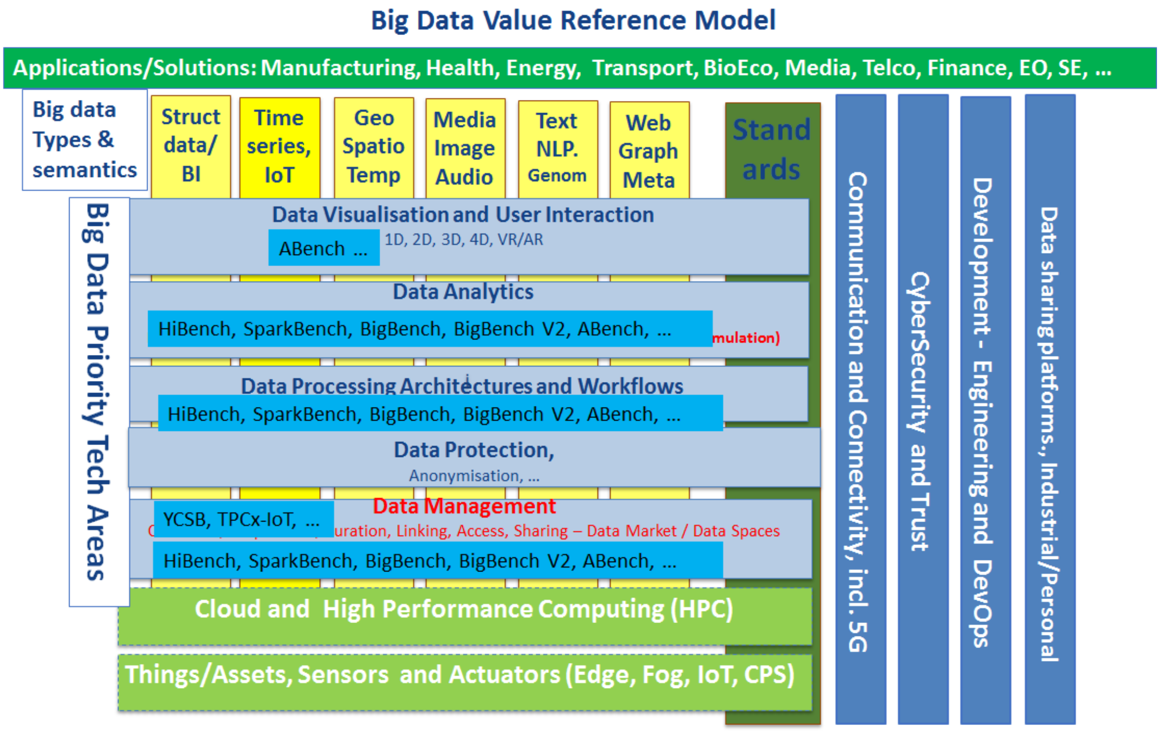

Big Data Value Reference Model
Applications/Solutions: Manufacturing, Health, Energy, Transport, BioEco, Media, Telco, Finance, EO, SE, ...
Big data Types & semantics
Struct data/ BI
Time series, IoT
Geo Spatio Temp
Media Image Audio
Text NLP. Genom
Web Graph Meta
Standards
Big Data Priority Tech Areas
Data Visualisation and User Interaction
ABench ...
1D, 2D, 3D, 4D, VR/AR
Data Analytics
HiBench, SparkBench, BigBench, BigBench V2, ABench, ...
mulation)
Data Processing Architectures and Workflows
HiBench, SparkBench, BigBench, BigBench V2, ABench, ...
Data Protection,
Anonymisation, ...
Data Management
YCSB, TPCx-IoT, ...
uration, Linking, Access, Sharing – Data Market / Data Spaces
HiBench, SparkBench, BigBench, BigBench V2, ABench, ...
Cloud and High Performance Computing (HPC)
Things/Assets, Sensors and Actuators (Edge, Fog, IoT, CPS)
Communication and Connectivity, incl. 5G
CyberSecurity and Trust
Development - Engineering and DevOps
Data sharing platforms., Industrial/Personal

Figure 18. Big Data Value Reference Model with the Associated Benchmarks

The BDV Reference Model has been developed by the BDVA, taking into account input from technical experts and stakeholders along the whole Big Data Value chain as well as interactions with other related PPPs. An explicit aim of the BDV Reference Model in the SRIA 4.0 document is to also include logical relationships to other areas of a digital platform such as Cloud, High Performance Computing (HPC), IoT, Networks/5G, CyberSecurity etc.

The BDV Reference Model may serve as common reference framework to locate Big Data technologies on the overall IT stack. It addresses the main concerns and aspects to be considered for Big Data Value systems.

It is an aim of DataBench to include support for appropriate benchmarks to cover all of the core areas of the BDV Reference Model, and to link to benchmarks for the related technical areas like Cloud, HPC, IoT etc.

The BDV Reference Model is structured into horizontal and vertical concerns.

- **Horizontal concerns** cover specific aspects along the data processing chain, starting with data collection and ingestion, reaching up to data visualization. It should be noted, that the horizontal concerns do not imply a layered architecture. As an example, data visualization may be applied directly to collected data (data management aspect) without the need for data processing and analytics. Further data analytics might take place in the IoT area – i.e. Edge Analytics. This shows logical areas – but they might execute in different physical layers.
- **Vertical concerns** address cross-cutting issues, which may affect all the horizontal concerns. In addition, verticals may also involve non-technical aspects (e.g., standardization as technical concerns, but also non-technical ones).

Given the purpose of the BDV Reference Model to act as a reference framework to locate Big Data technologies, it is purposefully chosen to be as simple and easy to understand as possible. It thus does not have the ambition to serve as a full technical reference architecture. However, the BDV Reference Model is compatible with such reference architectures, most notably the emerging ISO JTC1 WG9 Big Data Reference Architecture – now being further developed in ISO JTC1 SC42 Artificial Intelligence. The evolution of these will be followed up on by DataBench.

The technical areas as identified in the BDV Reference Model are:

Horizontal concerns:

- **Big Data Applications**: Solutions supporting big data within various domains will often consider the creation of domain specific usages and possible extensions to the various horizontal and vertical areas. This is often related to the usage of various combinations of the identified big data types described in the vertical concerns.
- **Data Visualization and User Interaction**: Advanced visualization approaches for improved user experience.
- **Data Analytics**: Data analytics to improve data understanding, deep learning, and meaningfulness of data.
- **Data Processing Architectures**: Optimized and scalable architectures for analytics of both data-at-rest and data-in-motion with low latency delivering real-time analytics.
- **Data Protection**: Privacy and anonymisation mechanisms to facilitate data protection. It also has links to trust mechanisms like Blockchain technologies, smart contracts and various forms for encryption. This area is also associated with the area of CyberSecurity, Risk and Trust.
- **Data Management**: Principles and techniques for data management including both data life cycle management and usage of data lakes and data spaces, as well as underlying data storage services.
- **Cloud and High-Performance Computing (HPC):** Effective big data processing and data management might imply effective usage of Cloud and High-Performance

Computing infrastructures. This area is separately elaborated further in collaboration with the Cloud and High-Performance Computing (ETP4HPC) communities.

- **IoT, CPS, Edge and Fog Computing**: A main source of big data is sensor data from an IoT context and actuator interaction in Cyber Physical Systems. In order to meet real-time needs, it will often be necessary to handle big data aspects at the edge of the system.

Vertical concerns:

- **Big Data Types and semantics**: The following six big data types have been identified – based on the fact that they often lead to the use of different techniques and mechanisms in the horizontal concerns, which should be considered, for instance, for data analytics and data storage: *1) Structured data; 2) Times series data; 3) Geospatial data, 4) Media, Image, Video and Audio data; 5) Text data, including Natural Language Processing data and Genomics representations; 6) Graph data, Network/Web data and Meta data*. In addition, it is important to support both the syntactical and semantic aspects of data for all big data types.
- **Standards**: Standardisation of big data technology areas to facilitate data integration, sharing and interoperability.
- **Communication and Connectivity**: Effective communication and connectivity mechanisms are necessary for providing support for big data. This area is separately elaborated further with various communication communities, such as the 5G community.
- **Cybersecurity**: Big Data often needs support to maintain security and trust beyond privacy and anonymisation. The aspect of trust frequently has links to trust mechanisms such as Blockchain technologies, smart contracts and various forms of encryption. The CyberSecurity area is separately elaborated further with the CyberSecurity PPP community.
- **Engineering and DevOps**: for building Big Data Value systems. This area is also elaborated further with the NESSI (Networked European Software and Service Initiative) Software and Service community.
- **Data Platforms**: This is a new area is being discussed as a grouping of data management and data processing, explicitly for Marketplaces, IDP/PDP, Ecosystems for Data Sharing and Innovation support. Data Platforms for Data Sharing include in particular Industrial Data Platforms (IDPs) and Personal Data Platforms (PDPs), but also include other data sharing platforms like Research Data Platforms (RDPs) and Urban/City Data Platforms (UDPs). These platforms include efficient usage of a number of the horizontal and vertical big data areas, most notably the areas for data management, data processing, data protection and cybersecurity.
- **AI platforms**: In the context of the relationship between AI and Big Data there is an evolving refinement of the BDV Reference Model – showing how AI platforms as a new area typically include support for Machine Learning, Analytics, visualization, processing etc. in the upper technology areas supported by data platforms – for all of the various big data types.

With the experience from the initially selected benchmarks there will be a continuous analysis of the inclusion of further appropriate benchmarks to ensure the coverage of the different areas.

4.3 Vision and potential implementation

A more detailed implementation of the overview specified in section 2.2 is explained in the following sections.

4.3.1 Phase 1: Alpha version

Figure 19 shows a potential implementation of the DataBench Toolbox.

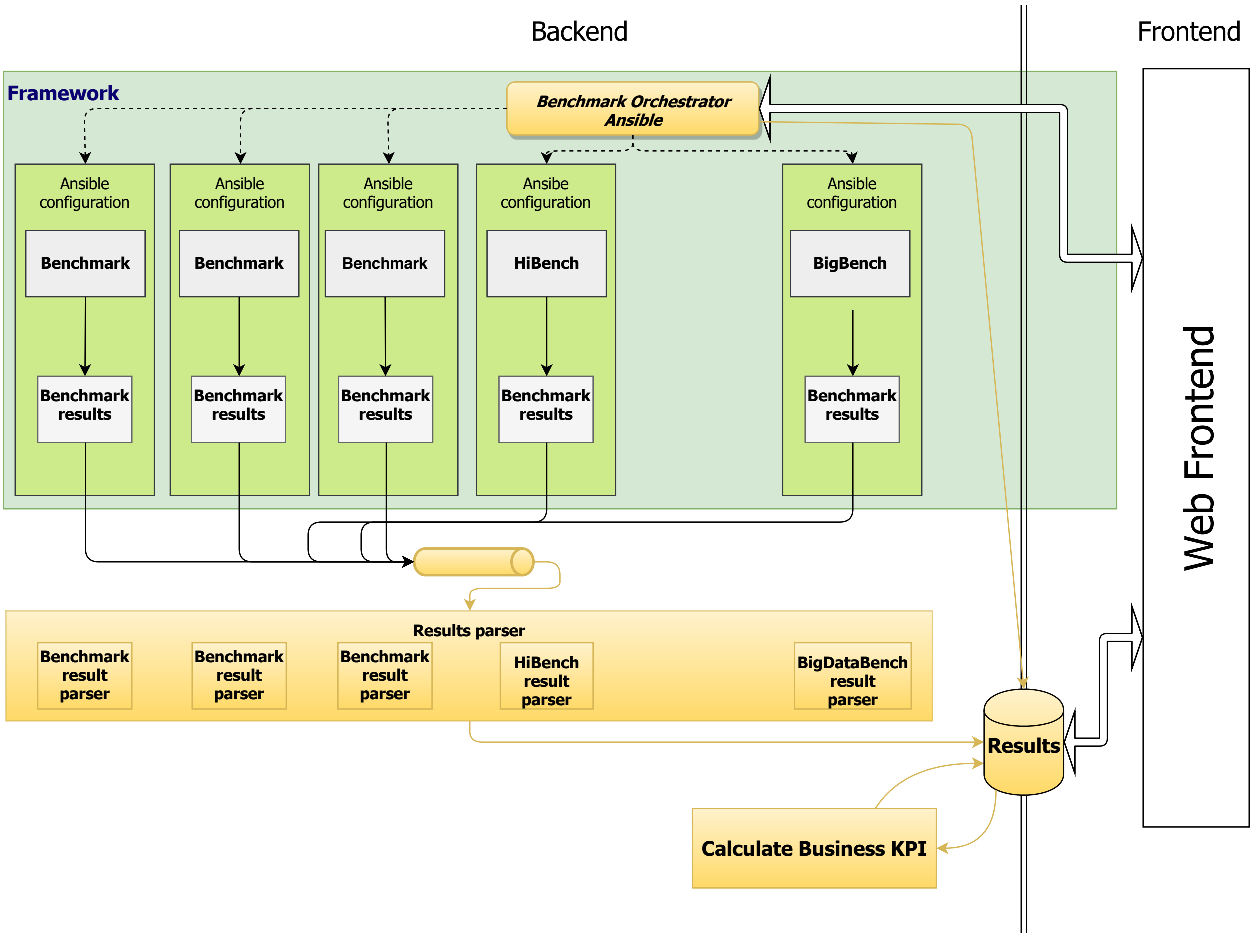


Figure 19. DataBench Toolbox potential implementation

This potential implementation is based on *Ansible* [6] for the backend. *Ansible* is an orchestration, configuration and deployment tool, based on templates called playbooks that simplify the process of deploying and configuring applications in different hosts.

The idea is to generate an *Ansible* playbook for downloading and configuring each of the benchmarks that will be integrated in the Framework, allowing them to be run by simply filling in the needed configuration of the cluster and running the playbook. To allow these playbooks to be stored we will have a playbook/configuration repository that can be a git repository.

[6] https://www.ansible.com/

With this playbook repository in place, the project will have a web frontend where the user can choose the desired benchmarks to run. This information will be used to provide the requested playbooks to the user, so they can configure them according to their requirements and run them in their environment.

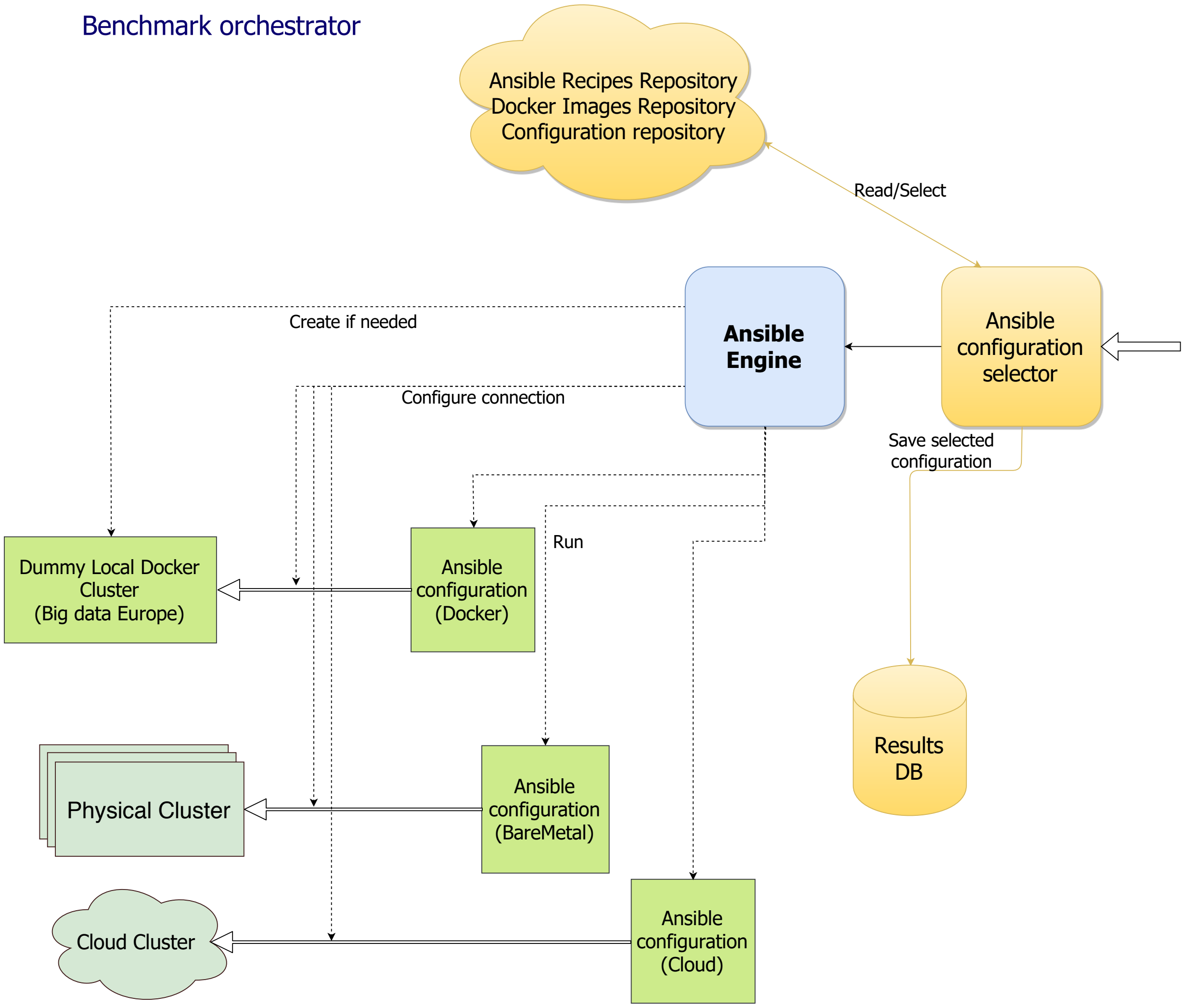

Benchmark orchestrator
Ansible Recipes Repository
Docker Images Repository
Configuration repository
Read/Select
Create if needed
Ansible Engine
Ansible configuration selector
Configure connection
Save selected configuration
Run
Dummy Local Docker Cluster (Big data Europe)
Ansible configuration (Docker)
Results DB
Physical Cluster
Ansible configuration (BareMetal)
Cloud Cluster
Ansible configuration (Cloud)

Figure 20 . Benchmark orchestrator detail

As can be seen in Figure 20, the Benchmark Orchestrator is in charge of the interaction with the repository and the frontend is the Benchmark Orchestrator. In detail, this module contains a customized coded module that will act as an interconnection hub between the repository, the frontend, the *Ansible* engine that will run the playbooks and the results DB. The main tasks of the *Ansible* configuration selector module will be to, given the set of benchmarks that the user has decided to run:

1- Get the requested playbooks from the repository
2- Provide the obtained playbooks to the user to fill in the needed configuration to run them on their infrastructure
3- Store the selected configuration in the results DB to be, later on, used when showing the results of the run.
4- Provide the configured playbook to the *Ansible* engine and run them.

The *Ansible* engine will then, given the playbook correctly configured, execute the selected benchmark in the configured infrastructure. To be able to do so, the only requirement is to have *Ansible* correctly installed in the machine that will act as master and configure it correctly so that it is able to connect to the required hosts with enough privileges.

Each benchmark generates results in a different format and measure different parts of the system. In order to be able to use them in a homogeneous way, the project needs to be able to gather and process that results in a centralized manner.

We propose *Apache Kafka*[7] as the tool for the interconnection pipeline between the benchmarks, the result parsers and the results DB, as can be seen in Figure 21.

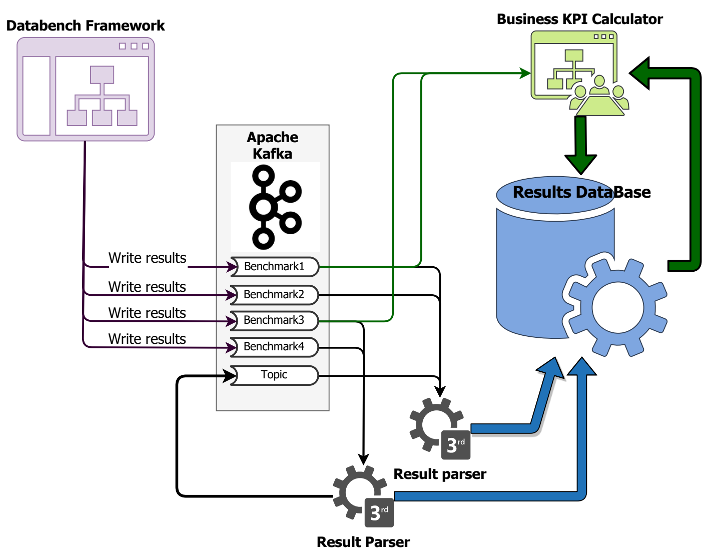


Figure 21.- Results processing

Apache Kafka is a distributed publish-subscribe messaging tool designed to be highly scalable, to have little overhead, to be really fast and with persistence. It is based on topics and replicas where any application can read/write to any given topic without impacting the rest of the applications using *Kafka*.

Since each benchmark generates its results in a different format, we need some modules in charge of parsing those results into a standardized technical data model that could be then read and interpreted to generate the needed KPIs.

The only requirement for these parsers, as can be interpreted from the diagram in Figure 21, is that they should be able to use *Kafka* as a source and/or target and relational databases as target. Since Apache *Kafka* is widely used in the Big Data community, there are lots of frameworks and libraries to use it with almost any programming language or tool so the chosen technology to implement the result parsers is completely open.

As far as the results database is concerned, the volume expectation is not large enough to explore highly scalable technologies and the proposed technical data model is quite relational. We need to relate the benchmark information, the information of the run and the results obtained, which looks like a perfect data model for any relational database. Moreover, we aim to connect directly the dashboard/ web frontend with the results database, so we need a database technology that allows connectivity from a web frontend and provides all the capabilities to be open to the public as well as to be used internally to store the results.

[7] https://kafka.apache.org/

4.3.2 Phase 2: Final version

The previous section details an initial implementation of the DataBench Framework where most of the manual work is relegated to the Technical User that owns the system to be benchmarked. One of the main goals of the project is to provide an easy-to-use platform that needs as little interaction as possible from the users. To help achieving this objective we propose an evolution of the initial idea where the main driver of the platform is the business KPIs that need to be obtained.

We propose to extend the previous phase by reusing most of the already created *Ansible* playbooks but, provided a web frontend, make the platform intelligent enough to decide which recipes to combine and populate automatically some of the configuration values.

To be able to achieve this objective, the Framework must know the relationship between business KPIs and the benchmarks, so that it is able to derive from the selected KPIs which benchmarks need to be run to calculate them.

In this case, the proposal includes a web frontend where the business users can choose the business metrics that they expect, and, with that information, the platform will automatically combine all the needed *Ansible* recipes in a single one to be presented to the Technical Users, so they can just fill in the missing variables and run it in their infrastructure.

In this second phase of the development, the main actor will be the Business User who knows what KPIs are needed, and the only task for the technical User would be to fill the system configuration and run the recipe. The Platform will be intelligent enough to combine all the needed recipes so that the benchmarks and their results can be processed to present to the Business User the requested metrics in the frontend. As explained in section 2.4, this business KPI derivation process is still under consideration and difficult to generalize. Therefore, in this second phase the final decision on the implementation of the derivation process will be made most probably involving not only algorithmic work to calculate business KPIs, but also visualization and Machine Learning techniques in combination with other material provided by DataBench (i.e. the Handbook) to help the Business User to take informed business decisions.

5. Conclusions

The DataBench Toolbox will be the central point of contact of the users of the DataBench Framework, and will allow the selection, deployment, execution of big data benchmarks, as well as the harmonization of the technical metrics and the derivation of business KPIs, where possible.

The aim of this document was therefore to present the initial architecture of the DataBench Toolbox and associated technical components. In order to do so, we presented the different elements that form the DataBench ecosystem, putting the DataBench Toolbox in the context of the main functional elements to be delivered by the different Work Packages of the project. An overview of the main elements of the ecosystem, namely the DataBench Toolbox, the associated benchmarks, the KPI derivation module and the AI framework has been presented, along with references to other supporting non-technical elements such as the DataBench Handbook.

After depicting the context of the project, the document dives in the architecture of the Toolbox. The DataBench Toolbox aims to be an umbrella framework for big data benchmarking reusing existing efforts well-settled in the community and therefore with the focus on not reinventing the wheel, but rather reuse the best-breed benchmarks. Some of the expected features to reuse existing big data benchmarks and derive business insights have been presented by means of UML use case diagrams, as well as a potential architecture for the Toolbox. However, these diagrams should not be seen as written into stone. The current view may change a bit in the future, but the main use cases will be respected and enhanced if needed. Ideas, phases and potential technologies to implement the Toolbox have been also explained and reported.

This initial architecture is the input to the first implementation of the DataBench Toolbox. An alpha version of the tool will be delivered in M18.

6. References

Ahmad Ghazal, Tilmann Rabl, Minqing Hu, Francois Raab, Meikel Poess, Alain Crolotte, Hans-Arno Jacobsen: BigBench: towards an industry standard benchmark for big data analytics. SIGMOD Conference 2013: 1197-1208

Ahmad Ghazal, Todor Ivanov, Pekka Kostamaa, Alain Crolotte, Ryan Voong, Mohammed Al-Kateb, Waleed Ghazal, Roberto V. Zicari: BigBench V2: The New and Improved BigBench. ICDE 2017: 1225-1236

BigBench, https://github.com/intel-hadoop/Big-Data-Benchmark-for-Big-Bench, 2018

BDV SRIA, Big Data Value association, European Big Data Value - Strategic Research and Innovation Agenda, vers. 4.0, Oct. 2017, http://www.bdva.eu/sria

Brian F. Cooper, Adam Silberstein, Erwin Tam, Raghu Ramakrishnan, Russell Sears: Benchmarking cloud serving systems with YCSB. Proceedings of the 1st ACM Symposium on Cloud Computing, SoCC 2010, Indianapolis, Indiana, USA, June 10-11, 2010

Dakshi Agrawal, Ali Raza Butt, Kshitij Doshi, Josep-Lluís Larrabee, Min Li, Frederick R. Reiss, Francois Raab, Berni Schiefer, Toyotaro Suzumura, Yinglong Xia: SparkBench - A Spark Performance Testing Suite. TPCTC 2015: 26-44

HiBench Suite, https://github.com/intel-hadoop/HiBench, 2018

Huang, S., Huang, J., Dai, J., Xie, T., Huang, B.: The hibench benchmark suite: Characterization of the mapreduce-based data analysis. In: Workshops Proceedings of the 26th International Conference on Data Engineering, ICDE 2010, March 1-6, 2010, Long Beach, California, USA. pp. 41–51 (2010)

Sanket Chintapalli, Derek Dagit, Bobby Evans, Reza Farivar, Thomas Graves, Mark Holderbaugh, Zhuo Liu, Kyle Nusbaum, Kishorkumar Patil, Boyang Peng, Paul Poulosky: Benchmarking Streaming Computation Engines: Storm, Flink and Spark Streaming. IPDPS Workshops 2016: 1789-1792

SparkBench, https://github.com/CODAIT/spark-bench, 2018

Todor Ivanov, Rekha Singhal: ABench: Big Data Architecture Stack Benchmark. Companion of the 2018 ACM/SPEC International Conference on Performance Engineering, ICPE 2018, Berlin, Germany, April 09-13, 2018

TPCx-IoT, http://www.tpc.org/tpc_documents_current_versions/pdf/tpcx-iot_v1.0.3.pdf, 2018

YCSB, https://github.com/brianfrankcooper/YCSB, 2018

YSB, https://github.com/yahoo/streaming-benchmarks, 2018