



# DataBench

**Evidence Based Big Data Benchmarking to Improve Business Performance**

## ***D3.4 Release Version of DataBench Toolbox including visualization and search components***

### **Abstract**

This is a supporting document of the release version of the DataBench Toolbox. The document provides an overview of all features evolved from the Alpha and Beta version of the Toolbox including new technical features not existing in the previous versions.

This document is the fourth deliverable related to the DataBench Toolbox after D3.1, D3.2 and D3.3. It provides an update of the development presented in the previous deliverables in the light of the advances done in the final release version of the Toolbox.



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

Deliverable D3.4	Release Version of DataBench Toolbox including visualization and search components
<b>Work Package</b>	WP3
<b>Task</b>	3.4
<b>Due date</b>	31/06/2020
<b>Submission date</b>	20/07/2020
<b>Deliverable lead</b>	GUF
<b>Version</b>	1.0
<b>Authors</b>	GUF (Todor Ivanov and Mehran Eftekhari) ATOS (Tomás Pariente, Iván Martínez and Ricardo Ruiz)
<b>Reviewers</b>	Arne Berre (SINTEF) Chiara Francalanci (POLIMI)

## Keywords

Benchmarking, Big Data, Big Data technologies, architecture, business performance, performance metrics, Toolbox, use cases, Nuggets

## 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

Copyright belongs to the authors of this document. Use of any materials from this document should be referenced and is at the user's own risk.

## Table of Contents

Executive Summary.....	5
1. Introduction.....	6
2. DataBench Toolbox Final version essentials.....	8
2.1 Toolbox in a nutshell.....	8
2.2 Toolbox architecture update.....	9
2.3 Toolbox subsystems coverage.....	11
2.4 Toolbox back-end.....	13
2.5 Toolbox front-end .....	13
2.6 Benchmarks integrated .....	14
2.7 Support for Knowledge Nuggets .....	14
2.8 Features included to describe benchmarks and Knowledge Nuggets .....	15
2.9 Quality Metrics.....	15
3. DataBench Toolbox back-end.....	17
3.1 Support for adding and configuring benchmarks.....	17
3.2 Support for deployment of benchmarks .....	20
3.3 Support for retrieving results.....	21
3.4 Metrics Spawner .....	22
3.5 Support for adding Knowledge Nuggets.....	22
3.6 Quality Metrics.....	23
4. DataBench Toolbox front-end.....	34
4.1 DataBench Toolbox main page.....	34
4.2 User journey for user type .....	35
4.3 Knowledge Nuggets Management lifecycle.....	37
4.4 Search functionality.....	40
4.5 Quality Metrics.....	42
5. Conclusions and future work.....	44
6. References.....	45

## Table of Figures

Figure 1 - Functional view of the DataBench ecosystem and Work Packages .....	8
Figure 2. Updated functional overview of the framework architecture .....	10
Figure 3 - View of all Quality Metrics.....	16

Figure 4 - AWX new template GUI .....	20
Figure 5 – Example of a configuration file to define the execution of a benchmark from the Toolbox web GUI.....	21
Figure 6 - Play Framework Request/Response Architecture .....	23
Figure 7 – Doughnut Charts.....	27
Figure 8 - Showing the rate of successful tasks per users .....	27
Figure 9 - Showing the first part of the usability metric.....	29
Figure 10 - Showing the second part of the usability metric.....	31
Figure 11 - Number of last Users .....	33
Figure 12 - Main page of the DataBench Toolbox.....	34
Figure 13 - Technical user journey .....	36
Figure 14 - Business user journey .....	37
Figure 15 - Page to create a new Knowledge Nugget.....	38
Figure 16 - Knowledge Nugget Catalogue.....	38
Figure 17 - Example of display of a Knowledge Nugget.....	39
Figure 18 - Example of update of a Knowledge Nugget .....	39
Figure 19 - Tag-based and use-case-based search functionality .....	40
Figure 20. Search by use case .....	40
Figure 21. Search by BDV Reference Model.....	41
Figure 22. Free text search.....	41
Figure 23 - Quality metrics Dashboard .....	42
Figure 24 - Example of Quality Metric display .....	42

## List of Tables

Table 1 – Processes and functions covered in the Release version.....	13
Table 2 – Quality Metrics Visibility for the different Users .....	16
Table 3 - Evaluation process of usability metric .....	28

## Executive Summary

This document is the final release of the DataBench Toolbox. It is providing an overview of the implementation of the final Toolbox release covering all of the expected features as of June 2020.

Instead of explaining the complete Toolbox, the document reports on the updates and new developments after the Alpha and Beta versions released with 6 months of interval each (reported in the DataBench deliverables D3.2 and D3.3) before outlining the status of the implementation at the time of submission (June 2020). For a complete overview of the functionalities covered it is advisable to read before deliverable D3.2 and D3.3 referenced from this document.

The document is a “demonstrator” deliverable, meaning that the DataBench Toolbox is accompanied with software and the Toolbox Dashboard accessing to the back-end functionality and integrated benchmarks. The URL of the Release version is available to the public.

This document provides an overview of the functionalities covered by the Toolbox. The Toolbox is offered as a web-based tool that enables access to catalogues of resources, namely the Benchmarking Tools and the Knowledge Nuggets catalogues. The Benchmarking Tools catalogue represents an attempt to list the most relevant technical Big Data benchmarking tools as presented in deliverables D1.2, D1.3 and D1.4. On the other hand, the Knowledge Nuggets catalogue builds on top of the knowledge gathered in the project, especially to offer to the community different pieces of information around benchmarking that pieced together could offer a better understanding of the business value of Big Data benchmarking. Both catalogues are indexed and annotated with selected tags to enable a rich search experience to different types of users of the Toolbox. Besides, the web tool offers advice to the different users on how to navigate and find the most suitable resources around benchmarking and access to statistics of the usage of the tool (visibility in some cases only to administrators).

Although this is the final release of the Toolbox, it is expected that the content available in this version will evolve in the future. The tool has been designed with extensibility points to enlarge and annotate the catalogues of benchmarking tools and Knowledge Nuggets.

## 1. Introduction

This deliverable is the final release of the DataBench Toolbox. As a demo deliverable, it provides software accessible via a web dashboard by the public. The document reports on the final implementation details of the Toolbox and provides cumulative updates over the Alpha and Beta versions reported in the DataBench deliverables D3.1[1], D3.2[2] and D3.3[3]. The document points to these deliverables, especially to D3.3 for further references.

The DataBench Toolbox, as its name implies, can be considered as a “box of tools”. This means that it is conceived as an entry point to access to resources (tools and knowledge) about Big Data benchmarking, rather than being a pure benchmarking system. Therefore, the focus is not providing just access to a specific benchmarking solution, but to a plethora of tools that will enable organizations and individuals to understand better what is available for benchmarking Big Data, as well as its context. The Toolbox is comprised of the following elements:

- A web-based front-end enabling access to the main functionalities of the Toolbox.
- A Big Data Benchmarking Tool Catalogue. This catalogue lists the most relevant technical Big Data benchmarking tools researched and presented in WP1 deliverables D1.1[4], D1.2[5], D1.3[6] and D1.4[7]. The initial population of benchmarks directly inherit its content from deliverable D1.2, and the set of annotations from D1.3 and D1.4. The catalogue is extensible to new benchmarks in the future.
- A catalogue of Knowledge Nuggets (KN). KN are pieces of information around benchmarking emerging from the research carried out within the project, but open to be enlarged with extra knowledge in the future. This catalogue builds on top of the information collected, especially to offer to the community a better understanding of the business value of Big Data benchmarking.
- Searching features. Both catalogues are indexed and annotated with selected tags to enable a rich search experience to different types of users of the Toolbox. The Toolbox provides three types of search: 1) a search box where users may type parts or any of the existing tags or words of the title or the resources; 2) an advanced search enabling the selection of tags to navigate to the resources; and 3) a search interface using the Big Data Value Reference Model explained in the BDV SRIA[8] to navigate to specific technical benchmarks that provide coverage for the horizontal or vertical layers of the model.
- Tag-based navigation between resources. From the web interface users can navigate between resources using the existing tags.
- Advice to the different users on how to navigate and find the most suitable resources. The front-page of the Toolbox has dedicated links to each type of user with examples and suggestions on how to use the tool more efficiently.
- Statistics of the usage of the tool (visibility only to administrators, although selected statistics are open to anyone).

Although this is the final release of the Toolbox, it is planned that the content available in this version will evolve in the future. The tool has been designed with extensibility points to enlarge and annotate the catalogues of benchmarking tools and Knowledge Nuggets.

The document follows a similar approach as Deliverable D3.2 and D3.3:

- Section 1 provides the introduction to the objectives of the deliverable.
- Section 2 is a summary of the essential features of the DataBench Toolbox Release version.
- Section 3 dives into the back-end processes updated in this Release version.
- Section 4 provides screenshots of the fronted in the Release version of the Toolbox.
- Finally, Section 5 provides the conclusions of the document as DataBench Toolbox.

## 2. DataBench Toolbox Final version essentials

### 2.1 Toolbox in a nutshell

To understand better the context of the DataBench Toolbox, delivered in the scope of WP3, it is important to understand its central role in the DataBench project as catalyser of the results of other Work Packages as depicted in Figure 1.

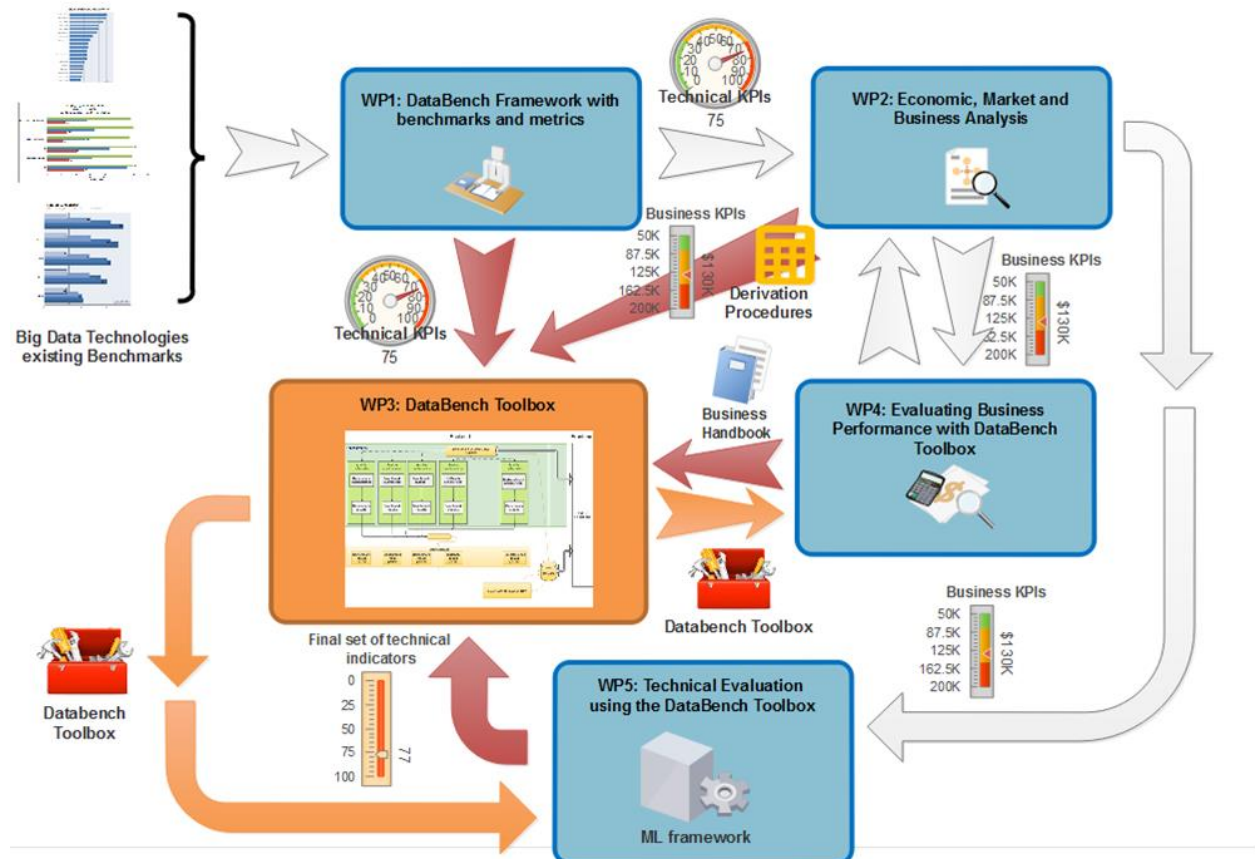


Figure 1 - Functional view of the DataBench ecosystem and Work Packages

The figure shows the relations among the different Work Packages and displays clearly the relation among them and with the Toolbox. The Toolbox is designed to give voice to the results coming from the project in the following way:

- WP1: DataBench Framework for Technical Benchmarks. This Work Package provided the framework of reference for technical Big Data benchmarking. As such it provided both a catalogue of existing initiatives and tools on Big Data benchmarking and a framework to classify them according to several aspects, such as the relation with the BDV Reference Model, annotation criteria, etc. The DataBench Toolbox provides access to the WP1 catalogue of benchmarks (deliverables D1.2, D1.3 and D1.4), as well as providing an entry point to update and annotate it according to the tags defined in D1.1. The Toolbox also provides a way of integrating the deployment and execution of a benchmarking tool into the Toolbox (feature tested with some selected benchmarks).
- WP2: Business benchmarks. WP2 researched business aspects related to Big Data benchmarking. This research led to a set of business indicators and field studies to



measure their impact in different industries and use cases. WP2 also provided means for organizations to self-assess their performance.

- **WP4: Business performance.** WP4 complements WP2 by analyzing the use of Big Data technologies in specific industries and use cases to evaluate their business performance and to create a set of architectural blueprints. WP4 will also generate the project Handbook that will be accessible from the Toolbox.
- **WP5: Validation.** WP5 provided a model for knowledge representation that was followed partially in WP3. WP5 also provided the means to log and evaluate the usage of the Toolbox giving access to statistics that can be visualized via the Toolbox (most of them available to administrators only). Future work within WP5 is related to certification of AI systems, which will be potentially integrated as another tool in the Toolbox.

The DataBench Toolbox sits therefore at the core of DataBench results providing access to different types of users to the resources made available by the project and by external initiatives. It is therefore a one-stop shop for Big Data benchmarking giving access to existing efforts in the community along with studies performed by the project to provide a framework of reference and knowledge about benchmarking.

A summary of the **main actors** (users) of the Toolbox, already identified in D3.1, is the following:

- **DataBench Administrators:** Admins will be in charge of overseeing the system. Admins have functionalities to finalize the editorial workflow of approving new content (benchmarks or KN) populated to the Toolbox, approve new users after they register to the Toolbox and check the statistics of usage and downtime of the Toolbox.
- **Technical users:** Technical users are users who are interested in searching and eventually deploy and execute benchmarks to evaluate technical indicators of Big Data systems, components or applications. They should have a certain degree of IT expertise (developers, testers, system administrators). Their main interest is therefore in finding the right benchmarks, information and knowledge to ease their benchmarking tasks.
- **Business users.** Business users are typically interested in business benchmarks rather than performing a technical benchmarking task. They are usually interested in the underlying business implications of selecting a Big Data system and how this choice influence their business indicators. Therefore, they typically would like to find similar cases in their industry, high level architectural blueprints of reference for their domain or use case. They would navigate through the KN catalogue to find the right answers. etc.
- **Benchmark Provider.** A Benchmark Provider is someone that would like to add or manage their own technical benchmark in the benchmark catalogue of the Toolbox. They are 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, and ultimately to declare and provide the automation mechanisms to enable the deployment and execution from the Toolbox.

## 2.2 Toolbox architecture update

The architecture of the DataBench Toolbox is described in detail in deliverables D3.1 and D3.2. and remains mostly unchanged in through the Beta and Release version. The main changes made to the original architecture are the support for Knowledge Nuggets and the addition of the quality metrics framework needed for the metrics dashboard defined in

Deliverable D5.3 [9]. Both updates are explained in more detail in section 3.5 and 3.6 respectively.

This section provides a summary of the architectural elements described in D3.1 including the new elements as shown in Figure 2.

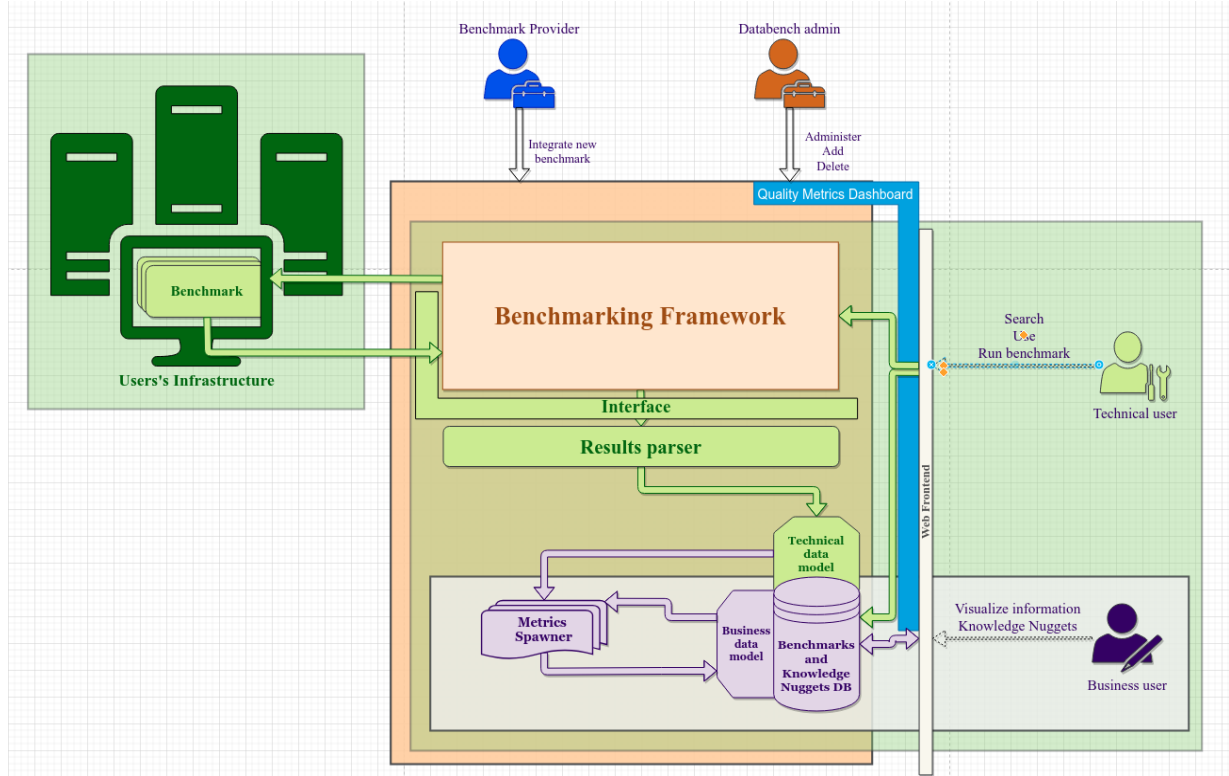


Figure 2. Updated functional overview of the framework architecture

The functional overview shown in Figure 2 presents the main components of the DataBench Toolbox:

- **Web interface:** The main access point to the DataBench Toolbox, currently accessible via the URL <http://databench.ijs.si/>. Please refer to section 4 for a complete overview of the front-end.
- **Benchmarking framework:** This module allows the storage of the templates and recipes to run integrated benchmarks. Benchmark Providers are able to interact with this module to add or manage technical benchmarks to the Toolbox. This is what we called integrated benchmarks (examples of integrated benchmarks are YSCB[15] or HiBench[16]). Once available, Technical users might configure, deploy an execute the integrated benchmarks in their selected infrastructure (i.e. in-house or in the cloud). With this module we can also have functionality to run some of the required benchmarks. The rest of the benchmarks are accessible and configurable using their own set of instructions, accessible from the content existing in the Toolbox.
- **User's Infrastructure:** Infrastructure outside of the scope of DataBench where the Technical users would like to run the benchmarks. As explained in the case of the

Benchmarking Framework, it could be in-house or cloud infrastructure for the integrated benchmarks.

- **Results Interface:** This module provides the means to transfer the results of the runs of the integrated benchmarks (also available for any other benchmark not integrated in the Toolbox) using specific Kafka topics.
- **Results Parser:** The results gathered via the Results Interface are transformed in this component to parse them and allowing their visualization. One new parser have to be defined and implemented for each of the benchmarks integrated.
- **Metrics spawner:** Initially, the idea of the metrics spawner was to derive business KPIs. However, this objective is only partially achieved. The Spawner provides now the metrics of the execution of benchmarks and access to related Knowledge Nuggets to help Technical and Business users with samples and knowledge about benchmarking activities, results, architectural blueprints, etc.
- **Benchmarks and Knowledge Nuggets DB:** DataBench Toolbox main database, where the catalogues of benchmarking tools, Knowledge Nuggets, tags, results from executions and quality metrics are stored.
- **Quality Metrics Dashboard:** Web interface accessible from the Toolbox where the quality metrics of the usage of the Toolbox are managed and presented.

From the implementation point of view, the decisions taken with respect to the frameworks for developing the Toolbox in deliverables D3.1 and D3.2 remain unchanged. Basically, the Toolbox is developed using frameworks and tools such as Bootstrap[17], Play![18], AWX[19] and Kafka[13]. More details about the technical choices can be found in D3.2 section 2.

### 2.3 Toolbox subsystems coverage

Deliverable D3.1 provided a detailed list of the processes and subsystems that take part in the lifecycle of the DataBench Toolbox as a set of UML diagrams. Deliverables D3.2 and D3.3 provided an update of the processes just highlighting the updates and functionalities done for the Alpha and Beta versions of the Toolbox. In this document we follow the same approach by describing the updates and improvements done to the Toolbox from the Release version grouped in the same way as described in D3.1, D3.2 and D3.3. For more details in the processes, please refer to D3.1. section 3.

As it was highlighted in the report of the Alpha and Beta versions of the Toolbox, the Graphical User Interface (GUI) was not planned until the final version, but a preliminary version was delivered as early as on the Beta version. The Release version, however, delivers a more advanced GUI, giving access not only to technical benchmarks, as in the previous versions, but to knowledge encapsulated in the form of Knowledge Nuggets, as well as to other tools developed in the scope of DataBench.

Table 1 shows the main elements covered in the release version and remarks about potential future enhancements.

Process	Covered in Release version	Remarks
<b>Accessing</b>	Improved user management. Users can register to the tool.	Final management of access control for guest users.  The front-end for sign-in, log-in and administrative management.  Improve and complete the definition for Technical users, Business users and Benchmark Providers, as well as the user grants by the administrator.
<b>User Intentions</b>	Guided search by tags. Guided search by use case. Guided search by the BDV Reference Model. Implementation of sample user journeys for Technical users, Business users and Benchmark Providers.	Improved visual search and navigation. Added a FAQ section.  Added specific pages for user journeys for the different types of users (Technical users, Business users and Benchmark Providers).
<b>Setup and Runtime</b>	Improved the module that monitors Kafka and stores the results into the database to handle any new benchmark that will be added.  Injection of results of the execution of benchmarks using Kafka to handle any future benchmark with little to no changes in the code.	Configuration of 4 benchmarks done, opened to Benchmark Providers to add new benchmarks for deployment and execution in the future.  The module is implemented only for the benchmarking tools integrated so far. There is no homogenization of the results, as they proved to be of very different nature.
<b>Analytics and metrics management</b>	Some simple analytics are done over the results to link use cases and benchmarks.  Results are parsed into metrics and can be shown to the user for better understanding of their results.	Analysis of the usage of the Toolbox has been added as a new module.
<b>Visualization and Reporting</b>	New look and feel of the front-page and the whole aspect of the Toolbox web site.  Inventory of benchmarks. Metadata-based Google-like search box.	Many enhancements done in the look & feel of the web site, search and navigation functionalities.  Added new visualization and search of Knowledge Nuggets as well as the dashboard metrics added,

Process	Covered in Release version	Remarks
	<p>Filtered advanced search by an initial set of metadata (tags).</p> <p>Navigation using tags to different benchmarking tools and Knowledge Nuggets.</p> <p>Metrics dashboard added.</p>	<p>The visualization of technical metrics remains like the Beta version.</p> <p>Future work: In the coming months the focus will be on providing more content, check the tags of the existing content and therefore providing a better navigation experience. But no changes in the implementation are expected.</p>
<b>Benchmark Management</b>	No new functionality in the Release version.	No changes or just small improvements.
<b>Knowledge Nuggets management (NEW)</b>	<p>Added the entire module to manage Knowledge Nuggets,</p> <p>The Release version Navigation using tags to different benchmarking tools and Knowledge Nuggets.</p>	<p>New module.</p> <p>Added tag navigation between Nuggets and benchmarking tools.</p>
<b>Quality metrics dashboard (NEW)</b>	Added the entire module to manage quality metrics of the usage of the Toolbox	<p>New module.</p> <p>Adding statistics of usage of the Toolbox.</p>

Table 1 – Processes and functions covered in the Release version

## 2.4 Toolbox back-end

This release version of the Toolbox has suffered some changes in the back-end mainly related to the inclusion of the Knowledge Nuggets and the integration of the quality metrics dashboard.

To allow the inclusion of the Knowledge Nuggets in the Toolbox, the back-end needed changes in almost every part. From the data model, to allow the storage of the Nuggets and their attachments as it will be explained in Section 3.5, and the integration of the Nuggets with the existing tags and features used in the Toolbox, to the back-end login and methods to allow create, read, update and delete operations over them. This logic will also be further explained in the document.

Moreover, in previous Deliverable D3.3 the logging mechanism was explained and a glimpse of how it was going to be implemented in the quality metrics dashboard was shown. In this release, the quality metrics dashboard has been completely integrated with the rest of the back-end as it will be explained in Sections 2.9 and 3.6.

## 2.5 Toolbox front-end

In this Release version of the Toolbox there have been many changes regarding the front-end. Starting with the main page, which has been redesigned to have a look & feel closer to the DataBench webpage. Moreover, extra functionalities like user journeys or shortcuts have been added to it.

On top of that, with the addition of some functionalities in the back-end, some extra effort was made to create the graphical user interface (GUI) for them and to extend the GUI changes to the whole platform so it looks similar overall.

More detailed information as well as some screenshots with the new changes are described in Section 4.

### 2.6 Benchmarks integrated

As a result of the work performed by partners in the project (especially the work done in WP1 in the scope of the revision of the deliverable D1.2 [5]), there has been an extension of the catalogue of benchmarks categorized and complemented with extra information. These benchmarks have been added to the Toolbox to be searched and its metadata has been properly filled with the information given in that list.

In the DataBench Toolbox an integrated benchmark is one described by a Benchmark Provider and enabled by an Administrator user with the necessary configuration elements for automatic deployment and execution from the Toolbox. The functionality and workflow to perform such an integration has been clarified and explained in the Release version and documented in section 3.1. As stated in the previous sections, the Release version of the Toolbox focused mainly on the improvements in its web user interface, search capabilities, inclusion of the quality metrics dashboard and the addition of the concept of Knowledge Nuggets. Therefore, no extra benchmarks have been integrated in the Toolbox, only some minor fixes and updates to already integrated benchmarks have occurred in the last 6 months. However, there is an ongoing work of implementing some additional standardized benchmarks such as TPCx-BB (BigBench) that will be integrated in the coming months.

### 2.7 Support for Knowledge Nuggets

One of the main additions to the Toolbox from the Release version is the inclusion of Knowledge Nuggets. These Knowledge Nuggets are small pieces of information extracted mostly from, but not limited to, the work performed by the project or information compiled from the benchmarking community. A Knowledge Nugget, as used by the DataBench Toolbox, is structured as follows:

- Title: Short title of the content of the Nugget;
- Description: In-depth description of what the Nugget contains;
- URL (Optional): A link to extra information regarding the Nugget;
- Attachments (Optional): A set of extra documents related to the Nugget.

To integrate these Nuggets with the Toolbox, the database data model has been extended and these Nuggets can also be tagged with features as the benchmarks as will be further explained in the next Section.

To allow the inclusion of attachments, which is a core part of a Nugget, the back-end has been modified to allow uploading documents to it as well as dynamically linking them in the GUI, so we allow the front-end to render the images when they are attached to a Nugget.

The Toolbox has been extended to support full CRUD operations over these Knowledge Nuggets and different views have been developed to create, list, update and delete the Nuggets.



## 2.8 Features included to describe benchmarks and Knowledge Nuggets

With the addition of the Knowledge Nuggets and the user journeys, the features included in the Toolbox have been extended with a new category called “Toolbox-specific features”.

These features are mainly created to categorize the Nuggets for the user journey experience, so we now have tags for beginner, intermediate or advanced users as well as tags for technical, business, benchmark expert and research/academia.

Moreover, to better integrate some of the compiled Nuggets, we have created some ad-hoc tags that describe the Nuggets and proved to be useful for navigation, such as Blueprints, Star performers, Benchmarking Organization, Top 3 use cases, etc.

These tags are also used in other parts of the Toolbox to present the information to the user in a structured way depending on their level of experience with these technologies. Moreover, some of the tags are used for “shortcuts” in different parts of the Toolbox because they provide a set of pieces of information that might be useful for certain user groups.

The Toolbox uses the tags as a navigation paradigm to link elements (benchmarks and Nuggets). Therefore, the user can click on the tags associated to one element and jump directly to the list of other elements annotated with the same tag. This navigation aspect has been implemented for the Release version of the Toolbox.

## 2.9 Quality Metrics

The goal of Deliverable D5.3[9] was to assess the technical usability, relevance, scale and complexity of the DataBench Framework with respect to usage and utilization. The DataBench Evaluation Methodology presented in D5.3 described and defined the different factors, features and metrics applied and implemented in the evaluation process. In D5.3 Section 3.2[9] were defined a set of metrics mapped to different quality factors and characteristics. Then by applying the evaluation process described in D5.3 Section 3.3, the metrics were categorized into three different types of Dashboards listed in D5.3 Table 2. Then in D5.3 Section 4 were described the framework architecture and implementation steps, which we followed to integrate the Quality Metrics Dashboards into the Toolbox. Table 2 lists the mapping between quality metrics and the user visibility based on the different Dashboard category (Platform, User and Administrator) defined in D5.3. This indicates what type of metrics the four different Toolbox users (Business users, Technical users, Benchmark Providers and Administrators) will be able to monitor.

Quality Factors	User visibility/ Dashboard Category
Effectiveness	All users
Reliability	All users
Satisfaction	All users
Functional Suitability	All users
Efficiency	Technical users, Benchmark Providers and Administrators

<b>Compatibility, Portability</b>	Technical users, Benchmark Providers and Administrators
<b>Usability</b>	Technical users, Benchmark Providers and Administrators
<b>Maintainability</b>	Administrators
<b>Performance efficiency</b>	Administrators

Table 2 – Quality Metrics Visibility for the different Users

Figure 3 shows a screenshot enabling all metrics in the Toolbox grouped by one of the 9 Quality Factors.

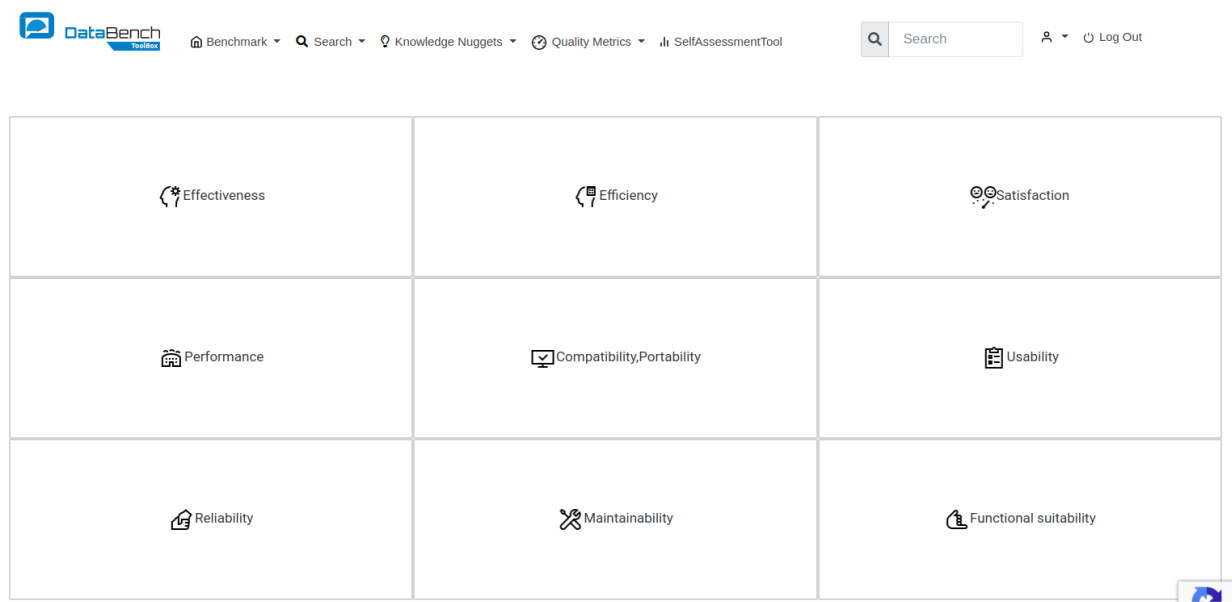


Figure 3 - View of all Quality Metrics

More information about the quality metrics is found in section 3.6.



### 3. DataBench Toolbox back-end

This section reports on the updates in the Toolbox since the Beta version reported in deliverable D3.3, which at the same time reports updates on D3.2 based on the design produced in D3.1. For more details of the back-end of the Toolbox, please refer to these previous deliverables.

#### 3.1 Support for adding and configuring benchmarks

The aim of this section is to describe, from the point of view of a Benchmark Provider user, the set of tools and reusable components that have been created in this project to help them add and potentially integrate their benchmarks with the Toolbox.

The addition of a benchmark to the Toolbox consists of two steps:

##### *Step 1 – Registering the new benchmark in the catalogue:*

The DataBench Toolbox offers the possibility to external registered users with the role of Benchmark Providers or Administrators to add new benchmarks to the Benchmarks catalogue. The user simply has to fill in a set of forms describing the benchmark and providing a set of features (tags) defined in the Toolbox DB data model and described in deliverable D3.1.

Once this description is done, an editorial flow is followed. The Benchmark Provider should contact via email (email provided in the Toolbox website) the Administrator user. The Administrator will revise the provided content, request for more info if needed, and eventually approve it. The benchmark will be then indexed and searchable in the Toolbox web interface. Other users in the system will then be able to look for the benchmark in the catalogue and follow the description and instructions from the provider to use it.

##### *Step 2 – Integration of the benchmark for deployment and execution:*

This step is optional and depends on potential technical constraints or on the willingness of the Benchmark Provider to expose the functionality of the benchmark from the Toolbox. The integration of a benchmark in the Toolbox means giving the Benchmark Provider the possibility of automation, totally or partially, of the process of configuring, deploying, running and getting the results of the execution back in the Toolbox DB. This process has already been explained in D3.1, D3.2 and D3.3. This section provides a summary and digs into some specific technical aspects of the solution.

The integration of a benchmark in the Toolbox requires two main steps: the creation of the Ansible Playbooks[12] by the Benchmark Providers and the final test and inclusion in the Toolbox AWX[19] by the Administrator. These two steps are explained in more detail below:

#### **Step 2.1 – Creation by the Benchmark Provider of the Ansible playbook to execute the benchmark**

The steps to be followed by a Benchmark Provider to design and prepare the benchmark with the necessary playbooks for the automation from the Toolbox are:

##### 1. Preparation of the hosts where the benchmark is going to run

The integrated benchmarks in DataBench are designed to be executed on external infrastructure, being in a cloud provider or in specific in-house hosts. In some cases, organizations are not prone to run the benchmarks in a public environment, but rather in a

close one isolated from the Internet for security, privacy or even commercial/IP protection reasons. To overcome this constraint, the Toolbox is designed to support automation using Ansible Playbooks split in 2 parts: A part executed locally, and the actual run of the benchmark executed in the target host. For clarification purposes, the local host is the machine where Ansible is executed and the target host is the machine where the system to be benchmarked is located (they can be the same machine).

If the benchmark is going to run on the local host:

- Ensure that there is a folder where the results of the run can be stored before being sent to the Toolbox.

If the benchmark is going to run on a target host different than the local host:

- Load variables files.
- Install required software (git, pip...) necessary to run the benchmark.
- Get the benchmark code (from Github, scp...).
- Ensure that the path to store the benchmark results exists.
- Configure the benchmark.
- Text that the benchmark can be executed and the results are copied back to the local host.

Most of these steps can be skipped using the different variables in the variables' configuration file. Examples of variables' configuration files are provided in the repository to help the Benchmark Provider on this.

## 2. Naming conventions

The Benchmark Provider must follow certain naming conventions to ensure that the benchmark is configured properly:

- The benchmark playbooks name:

*Benchmark\_<benchName>.yml*

- Specific variables files:

*variables\_<benchName>.yml*

- Kafka topics for the results:

*<benchName>\_results*

- Key of the message when sending to kafka:

*<AWX\_job\_run\_id>/<user\_id>/<filename>*

*eg.: 0/testid\_todo/20181107T104028\_workloada\_arangoDB\_LoadResults.txt*

## 3. Reusable playbooks

There are some playbooks created to help developing new playbooks to integrate benchmarks into the Toolbox. These can be found in the GIT repository of the project. Benchmark Providers may access to these templates and use them to prepare the playbooks for their own benchmarks.

As a matter of example, one of the templates in the repository is the playbook in charge of sending the results of the benchmark executions to the Kafka informing the Toolbox of the

topics from where the Toolbox should subscribe to get the results from the runs among other things:

*ansiblePlaybooks/Results\_SendToKafka.yml*

This playbook requires as input the following arguments:

- **kafka\_broker**: URL and port to the Kafka broker
- **topicName**: Name of the topic to send the results to. Must end with `_results`
- **benchName**: Name of the benchmark that creates the results
- **local\_resultsPath**: Path where the results are stored

To use this playbook in a new playbook the Benchmark Provider can just add the following block:

Code:

```
- block:
  - include: Results_SendToKafka.yml kafka_broker=<kafka_broker_ip_and_port>
    topicName=<kafka_topic_name> benchName=<benchmark_name>
    local_resultsPath=../results/YCSB/results
  when: autoSendResults
```

#### 4. Notifying the DataBench Toolbox administrators

After the completion of the Ansible playbook, the Benchmark Provider should open a pull request to the project git or contact via email with the email provided in the Toolbox for the Administrator user to start the process of integrating the playbook in the Toolbox.

##### Step 2.2 – Integration of the Ansible playbook in the Toolbox by the administrators

Once the Benchmark Provider has notified the administrators of the creation of a new playbook, the admins will finalize the process of integrating it in the Toolbox. The steps to be followed by the Administrator are:

##### 1. Revise the code and upload it to the git repository

To do so, they should review the code provided and check that it follows the requirements described before in order to accept the change and move it to the git repository of the Toolbox.

##### 2. Create the template for the new benchmark in AWS

Once the code is in the git repository, the administrator will log into AWX and create what the tool calls a “template”. The user interface used for this is shown in Figure 4. To create a template the administrator should provide a name for it following the pattern: *“DataBench\_Benchmark\_<benchmarkName>”* and provide a description for the template that can be the same as the one given to the benchmark in the benchmark catalogue.

The screenshot displays the 'NEVA PLANTILLA DE TRABAJO' (NEVA WORK TEMPLATE) interface. At the top, there are tabs: 'DETALLES' (selected), 'PERMISOS', 'NOTIFICACIÓN', 'TAREAS COMPLETADAS', 'PROGRAMACIONES', and 'AÑADIR ENCUESTA'. The main form is organized into several sections:

- \* NOMBRE**: A text input field.
- \* INVENTARIO**: A search input field with a magnifying glass icon and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- \* CREDENCIAL**: A search input field with a magnifying glass icon and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- \* NIVEL DE DETALLE**: A dropdown menu currently set to '0 (Normal)' and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- ETIQUETAS**: A text input field.
- TIMEOUT**: A numeric input field set to '0'.
- DESCRIPCIÓN**: A text input field.
- \* PROYECTO**: A search input field with a magnifying glass icon.
- FORKS**: A numeric input field set to '0'.
- ETIQUETAS DE TRABAJO**: A search input field with a magnifying glass icon and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- GRUPOS DE INSTANCIAS**: A search input field with a magnifying glass icon.
- MOstrar CAMBIOS**: A toggle switch currently set to 'OFF' and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- \* TIPO DE TRABAJO**: A dropdown menu set to 'Ejecutar' and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- \* PLAYBOOK**: A dropdown menu set to 'Escoja un playbook'.
- LÍMITE**: A text input field and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- OMITIR ETIQUETAS**: A text input field and a checkbox labeled 'PREGUNTAR AL EJECUTAR'.
- JOB SLICING**: A dropdown menu set to '1'.
- OPCIONES**: A list of checkboxes:
  - ☐ ACTIVAR LA ELEVACIÓN DE PRIVILEGIOS
  - ☐ PERMITIR LA EJECUCIÓN UTILIZANDO CALLBACKS
  - ☐ ACTIVAR LOS TRABAJOS CONCURRENTES
  - ☐ USAR CACHE DE EVENTOS
- VARIABLES ADICIONALES**: A section with tabs for 'YAML' and 'JSON', and a checkbox labeled 'PREGUNTAR AL EJECUTAR'. Below the tabs is a text area with a line number '1' and a blue cursor.

At the bottom right, there are two buttons: 'CANCELAR' and 'GUARDAR'.

Figure 4 - AWX new template GUI

### 3. Select the project and the playbook

The next step is to select the project and the playbook from the ones in the selectable list, they are loaded from the playbooks stored in git. Moreover, the needed variables need to be written in the “*extra variables*” field as if it was a variables file. This information will be later shown to the user when trying to run a benchmark and are all the variables that need to be configured in order to run the benchmark from the Ansible playbook correctly.

### 4. Allow the configuration of the template at runtime from the web

The last step is to mark the tick “ask when run” to the fields: *inventory*, *credential* and *extra vars*. This way the template can be configured from the front-end via the AWX API allowing it to be configured by the user.

### 5. Finalize the integration

Once these steps are completed and the template is saved, the administrator can then go to the benchmark update page to select from the selectable the newly created template and apply it to that benchmark, now the benchmark is integrated and ready to be run from the web front-end.

## 3.2 Support for deployment of benchmarks

This functionality was already explained in Deliverable D3.3 and did not change in the Release version of the Toolbox. For clarity sake, this section summarizes what was explained in D3.3 and provides an example on how Technical users should use the Toolbox to deploy and execute integrated benchmarks.

As explained in section 3.1 the Toolbox relies on Ansible, AWX and the use of Ansible Playbooks to deploy and run the benchmarks on the selected infrastructure. To allow this functionality to work from the Toolbox, a business logic was implemented to handle the requests to the AWX API.

The execution of an integrated benchmark from the Toolbox is only allowed for registered users with the role of Technical user or Administrators. Therefore, this functionality is not available for users before logging in to the Toolbox web interface.

In order to make it easier for a user to run a benchmark, the variables needed by the benchmark to be run can be configured from the Toolbox GUI and sent to AWPX via the API before its running. The Benchmark Provider should have prepared the template for this configuration in advance, as explained in section 3.1. An example of the configuration file to be filled in by the Technical user is shown in Figure 5.

Reference:

<https://github.com/brianfrankcooper/YCSB>

**Configuration Page**

Use Case:  
 Add custom...

Extra vars:

```

1- # YCSB benchmark specific
2  ## Define the number of recors for the execution
3  YCSB_recordcount: 1000
4  YCSB_workloads: ["workloads"]
5  ## Cleanup results after execution?
6  YCSB_remote_cleanup: false
7  YCSB_hostname: 127.0.0.1
8  # ArangoDB specific
9  ## Run the benchmark for this DB: true or false
10 YCSB_arango: true
11 YCSB_arango_port: 8529
12
13- # Orientdb specific
14 ## Run the benchmark for this DB: true or false
15 YCSB_orientdb: false
16 YCSB_orientdb_user: root
17 YCSB_orientdb_password: rootpwd
18 YCSB_orientdb_url: remote:130.206.113.246/ycsb
19
20- # MongoDB specific
21 ## Run the benchmark for this DB: true or false
22 YCSB_mongo: false
23 YCSB_mongo_port: 27017
24
25- # Redis specific
26 ## Run the benchmark for this DB: true or false
27 YCSB_redis: false
28 YCSB_redis_port: 6379
29
30- # Couchbase specific
  
```

Select Inventory:

Create New

Host IP (Eg: 127.0.0.1)

Select Credentials:

Launch Job

**Figure 5 – Example of a configuration file to define the execution of a benchmark from the Toolbox web GUI**

The example shows the configuration file of the YCSB [15] benchmark that compares different metrics of several NoSQL databases. Technical users will enter to this form by selecting the YCSB benchmark from the catalogue of benchmarks of the Toolbox. The user has to modify the parameters in the configuration file to determine elements such as the targeted hosts, ports and path to store the benchmark, the path in that host where the results of the execution would be stored, the databases that would like to compare, credentials, etc. Once this is configured, the user can execute the playbook directly from the Toolbox web interface and the system will take care of deploying and running the benchmark.

### 3.3 Support for retrieving results

This functionality is further detailed in deliverables D3.2 and D3.3.

This module allows uploading the results of the execution of benchmarks in the user's infrastructure (i.e. in-house or cloud) to the user space of the Toolbox DB. The Toolbox has

implemented a Kafka infrastructure to enable benchmarking tools to subscribe and send the results of the benchmarking runs. This can be done programmatically, or via a specific GUI of the Toolbox that allows uploading files to the Kafka topic, thus decoupling the gathering of results from the rest of the Toolbox functionality and ensuring a better stability and scalability of the solution.

This functionality has been available since the Beta version.

### 3.4 Metrics Spawner

No major changes have been done to this module since the Beta version, only minor fixes found while testing the integrated benchmarks and the calculation of their metrics. It was explained in depth in deliverables D3.2 and D3.3.

### 3.5 Support for adding Knowledge Nuggets

All the functionality regarding the Knowledge Nuggets was newly developed for this final version of the Toolbox.

In previous versions, the Toolbox was mainly storing information about the technical benchmarks. However, knowledge is key to bridge the gap between pure technical benchmarking and business insights. To that extent this knowledge has been encapsulated under the concept of Knowledge Nuggets containing examples, best practices, similar cases, architectural blueprints, benchmarking needs per industry and use cases, etc.

A modification of the Toolbox DB data model was needed to integrate the Knowledge Nuggets. A new table to store the information regarding the Nuggets was created. Nuggets can be tagged using the same set of features used by the technical benchmarks in the Toolbox as tags. Thanks to this, the Toolbox implements a tag navigation paradigm that enables the possibility of showing Nuggets related to the technical benchmarks. Having this relation with the common features, the search functionalities can show not only the benchmarks, but the related Nuggets as well.

It was mandatory to have a way of creating, updating and deleting these Nuggets, so the Toolbox back-end was extended to allow access to all these functions from the front-end. Moreover, since the Nuggets are usually graphical pieces of information, there was the need to create an upload mechanism to store the attachments in the Toolbox to allow downloading or rendering them in the Nugget. This is done by the back-end storing references to the uploaded attachments that are linked to the download button when needed.

Also, since many times these Nuggets are complemented with images, both the back-end and the front-end were modified to store and render the images. At the back-end, a new logic was added to create dynamic links to the uploaded images as attachments. Since these attachments are created with the use of the Toolbox there was the need of this logic, otherwise the images cannot be rendered by the front-end due to security issues related to the Play! Framework used in the back-end. So, in the Release version, when rendering the information of a single Nugget, if there is an image attached (either .jpg or .png) the front-end is able to show that attachment as an image after the description instead of only showing the download link.

### 3.6 Quality Metrics

In this section we describe the overall flow of the Play! framework and how the Dashboard Quality Metrics defined in D5.3[9] have been implemented. Furthermore, the implementation of two quality metrics Effectiveness and Usability are described in detail.

The Play framework is completely stateless and only request/response oriented. All HTTP requests follow the same path:

1. An HTTP request is received by the framework.
2. The router component attempts to find the most specific route that this request can accept. The corresponding action method is then called.
3. The application code is running.
4. When a complex view needs to be generated, a *template file* is rendered.
5. The result of the action method (HTTP Response Code, Content) is then written as an HTTP response.

The following diagram on Figure 6 summarizes the HTTP request path.

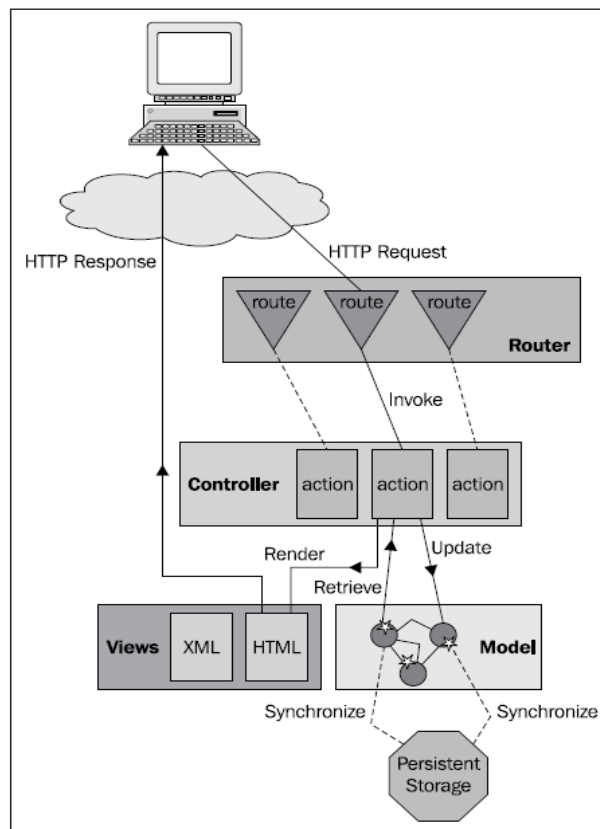


Figure 6 - Play Framework Request/Response Architecture<sup>1</sup>

<sup>1</sup> <https://www.playframework.com/documentation/1.2/main#mvc>



### 3.6.1 Effectiveness

**Effectiveness** is the accuracy and completeness with which users achieve certain goals. It is important to track the successful completion of tasks, such as benchmark executions, started by DataBench users. It is defined in D5.3[9] as:

“Percentage of successful benchmark runs”

Number of successful benchmark runs

Total number of benchmark runs

After selecting the effectiveness link from the main page of the metric dashboard, controller action “metricResults” created and executed. The implementation is defined as listed in Code Snippet 1.

```
public Result metricResult(. . .) throws FileNotFoundException {
    . . .
    switch(selectedmetric) {
        case "Effectiveness" :
            arr.add(rateOfSuccessTasks().toString());
            arr.add(String.valueOf((int) Math.round(benchRunsCount())));
            arr.add(String.format( "%.2f", (
                rateOfSuccessTasks()/benchRunsCount()) * 100));
            doughnutChart = rateOfSuccessTaskPerUser();
            barChart = rateOfSuccessTaskPerUser();
            break;
        case "Efficiency" : . . .
        . . .
        default :
        . . .
        break;
    }
    return ok(views.html.metrics.metricResults.render( id1 ,
        doughnutChart , arr ,barChart . . .));
}
```

Code Snippet 1 - Part of "metricResult" method to identify selected metric parameters

"metricResult" method controls by passing parameters "selectedmetric" as the keyword for switch statement to find out which metric was chosen. At the end, one of nine blocks of code defined in switch is selected and executed. The "Ok" constructor used in Code Snippet 1 renders the view "views.html.metrics.allmetrics" with its parameters. The methods that calculate the values of the parameters for Effectiveness are implemented as in Code Snippet 2.

"rateOfSuccessTasks" returns the total successful benchmark executions of all users. Successful are the executions that have a value equal to "%READ\_ R%" for the column "name" in the table "benchmark\_results".

```
public Integer rateOfSuccessTasks() {
    return Result_Metric_DB.find.query()
        .select("id, name , value,benchmark_results.benchmark_run.user_id")
        .fetch("benchmark_results.benchmark_run")
}
```



```

        .where().like("name" , "%READ_ R%").findCount();
    }

```

Code Snippet 2 - "rateOfSuccessTasks" Method

"benchRunsCount" shows the total number of benchmarks executions. After that, "rateOfSuccessTasks/benchRunsCount" simply calculates average of all successful executions.

"rateOfSuccessTaskPerUser" is calculated number of total or successful runs per user. This method generates a JSON string for both bar and doughnut charts. The implementation looks like in Code Snippet 3.

```

public String rateOfSuccessTaskPerUser() {
    List<User_DB> userList= User_DB.find().query().findList();
    String benchSubCountPerUsertring = "{ \"dataSets\" : [ \" \" ;
    boolean coma = true ;
    for( User_DB item: userList)
    {
        Integer rateOfSuccessTask = Result_Metric_DB.find().query()
        .select("id, name , value , benchmark_results
        .benchmark_run.user_id ").fetch("benchmark_results.benchmark_run")
        .where().like("name" , "%READ_ R%")
        .eq("benchmark_results.benchmark_run.user_id",
        item.getId()).findCount();
        Integer rateOfTask = Benchmark_run_DB.find().query().where()
        .eq("user_id", item.getId()).findCount();
        if(rateOfSuccessTask > 0){
            if (coma == false) { benchSubCountPerUsertring += ","; }
            coma = false;
            benchSubCountPerUsertring += "{ \"key\": \"" + item.getUserName()
            + "\", \"value\": " + rateOfSuccessTask + " , \"value1\": " +
            rateOfTask + " }";
        }
    }
    benchSubCountPerUsertring += "]}";
    return benchSubCountPerUsertring;
}

```

Code Snippet 3 - "rateOfSuccessTaskPerUser" Method

The values are represented by doughnut and bar charts. Doughnuts are divided into segments. The arc of each segment shows the proportional value of each piece of data. A bar chart, on the other hand, provides a way to display data values that are represented as a vertical bar. It is sometimes used to display trend data and compare multiple records sides by side.

"mychart.js" file contains all JavaScript methods used in views and the project should be imported at the beginning. This JavaScript file generates chart objects in our project. Below in Code Snippet 4 is the code implementing the doughnut and bar charts.

```

Bar Chart:
if( document.getElementById("metricPagest").innerText == "Effectiveness" ){

```

```

d1.label = 'Successful tasks' ;
d2.label = 'Total tasks' ;

var ctxCompareUser =
    document.getElementById('compareBarcanvas').getContext('2d');

var objBarChartSt =
    JSON.parse(document.getElementById("barChartSt").innerText);

for (var index = 0; index < objBarChartSt.dataSets.length ; ++index) {
    dataPerUser.labels.push( objBarChartSt.dataSets[index].key);
    d1.data.push(objBarChartSt.dataSets[index].value);
    d2.data.push(objBarChartSt.dataSets[index].value1 );
}

var templateCompareUser = {   type: 'line' ,    data: dataPerUser ,
                             options: optionTemplateLine };

var lineChartcompareBarcanvas = new Chart(ctxCompareUser, templateCompareUser);
}

```

#### Doughnut Chart:

```

if(document.getElementById("metricPagest").innerText == "Effectiveness" ) {
    var objDoughnutChartSt = JSON.parse(document.getElementById("doughnutChartSt").innerText);
    var ctxSuccessfulTasksPerUser =
        document.getElementById('successfulTasksPerUser').getContext('2d');

    insertData();

    var templateSuccessfulTasksPerUser = {   type: 'doughnut' ,    data: data1 ,
                                             options: optionTemplateRightlegend };

    templateSuccessfulTasksPerUser.options.plugins.doughnutlabel.labels =
        [ { text: count.toString() , font: {size: '60'} } ];

    var chartSuccessfulTasksPerUser =
        new Chart(ctxSuccessfulTasksPerUser, templateSuccessfulTasksPerUser);
}

```

#### Code Snippet 4 - Creating the Charts for effectiveness

Here the JSON string are interpreted by *JSON.parse* and the data are added dynamically in the corresponding chart objects. Then Figure 7 and Figure 8 show the resulting diagrams created from the converted data.

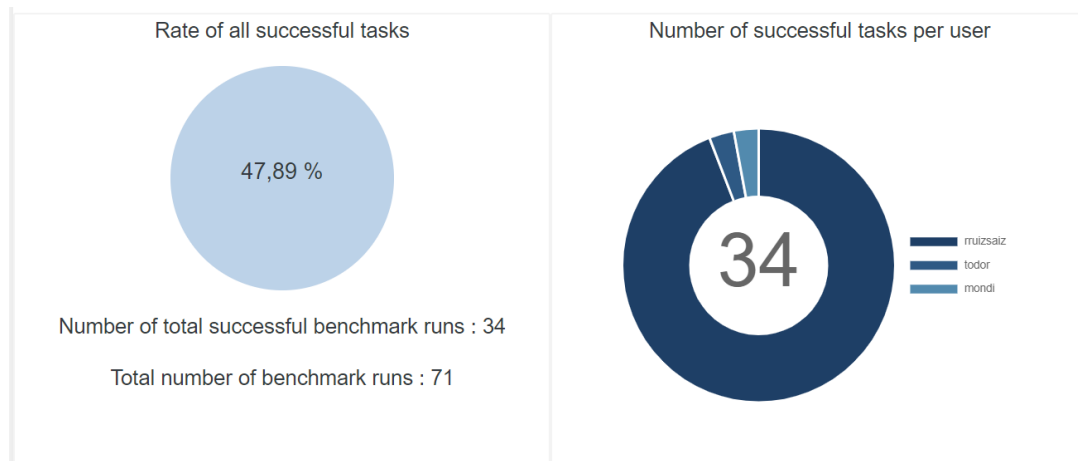


Figure 7 - Doughnut Charts

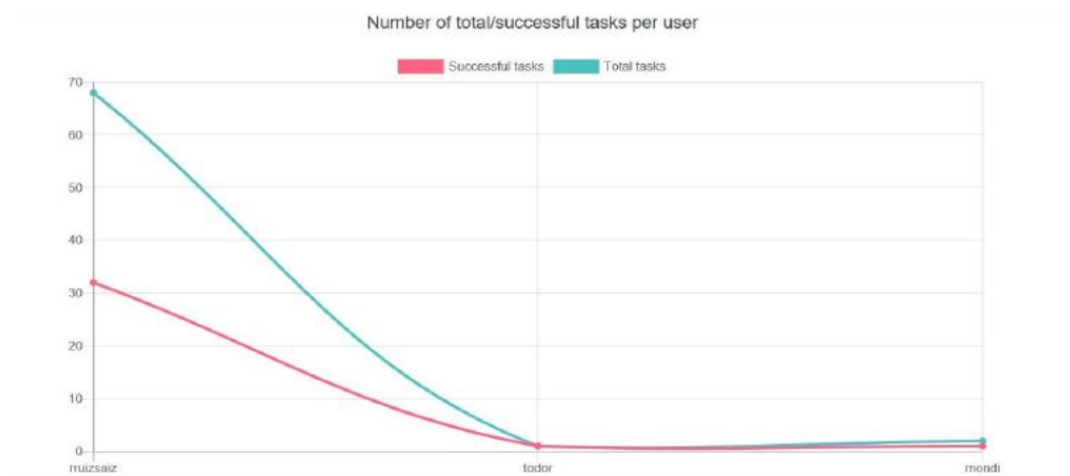


Figure 8 - Showing the rate of successful tasks per users

### 3.6.2 Usability

**Usability** shows the degree to which a product or system can be used by specific users to achieve specific goals with effectiveness, efficiency and satisfaction in each context of usage. Table 3 briefly describes the assignment of factors to the phases of the assessment process and the metrics used to measure this quality factors as defined in D5.3 [9].

Quality factors	Level 1 - Requirements	Assessment	Level 2 - Specify rating	Stage 4 - Implementation and completion of the evaluation
<b>Usability</b>	How many benchmarks does a user run on average?		Average of executed per benchmarks user	User Administrator Metrics Dashboard and
<b>Usability</b>	How many benchmark searches have been performed? What are the most interesting benchmarks?		Number of benchmark searches	User Administrator Metrics Dashboard and

<b>Usability</b>	How many benchmark results have been submitted to the DataBench Toolbox?	Number of benchmarks submitted	Platform, User and Administrator Metrics Dashboard
<b>Usability</b>	How many use case scenarios are implemented?	Number of use case scenarios	Platform and Administrator Metrics Dashboard

Table 3 - Evaluation process of usability metric

Using the block code shown in Code Snippet 5 from “*Controller.metricResult*”, the methods of this part of Action are explained below.

```

case "Usability"

    arr.add(String.format( "%.2f", benchRunsCount() / usersCount() ));
    arr.add(useCaseCounts().toString());
    doughnutChart = benchSubCountPerUser();
    doughnutChart6 = benchSubCountPerBench();
    barChart = searchCount();
    barChart2 = activeUsers();

    break;

```

Code Snippet 5 - Part of “metricResult” method to identify selected metric parameters

“*benchRunsCount*” method calculates the total number of benchmarks runs and “*usersCount*” method returns the number of all users by a request from the table “*User\_DB*”. Then the average of the executed benchmarks per user is calculated. “*useCaseCounts*” method generates the number of implemented use case scenarios from the table “*User\_DB*”.

“*benchSubCountPerUser*” and “*benchSubCountPerBench*” methods return the number of submitted benchmark results additionally separated by user and benchmark in a JSON string. As seen in Code Snippet 6, the values of the benchmark results per user are generated in the form of *HashMap* variables like “*User or Benchmark: Value*”.

```

public String benchSubCountPerUser() {

    List<Benchmark_result_DB> benchmark_result =

        Benchmark_result_DB.find().query().fetch("benchmark_run").findList();

    Map<String , Integer> benchmark_result_bench = new HashMap<>();

    for( Benchmark_result_DB item: benchmark_result)

    {

        benchmark_result_bench.put( (""+item.benchmark_run.getUser_id()+"").toLowerCase()

        , enchmark_result_bench.getDefault((""+item.benchmark_run.getUser_id()+"")

```

```

        .toLowerCase(), 0) + 1);
    }

    String benchSubCountPerUserstring = "{ \"dataSets\" : [ \" ";
    boolean coma = true ;
    for(String ii : benchmark_result_bench.keySet()){
        if (coma == false) {    benchSubCountPerUserstring += \",\";    }
        coma = false;
        benchSubCountPerUserstring +=
            \"{ \"key\": \"\" + User_DB.findById(Integer.parseInt(ii)).getUserName()
            + \"\" , \"value\": \" + benchmark_result_bench.get(ii) + \"\" }\";
        benchSubCountPerUserstring += \",\";
    }
    return benchSubCountPerUserstring;
}

```

Code Snippet 6 - "benchSubCountPerUser" Method

The doughnut chart, depicted in Figure 9, is created from the generated data. On the left side of the page the values are displayed by user, while on the right side the values are displayed by benchmark.

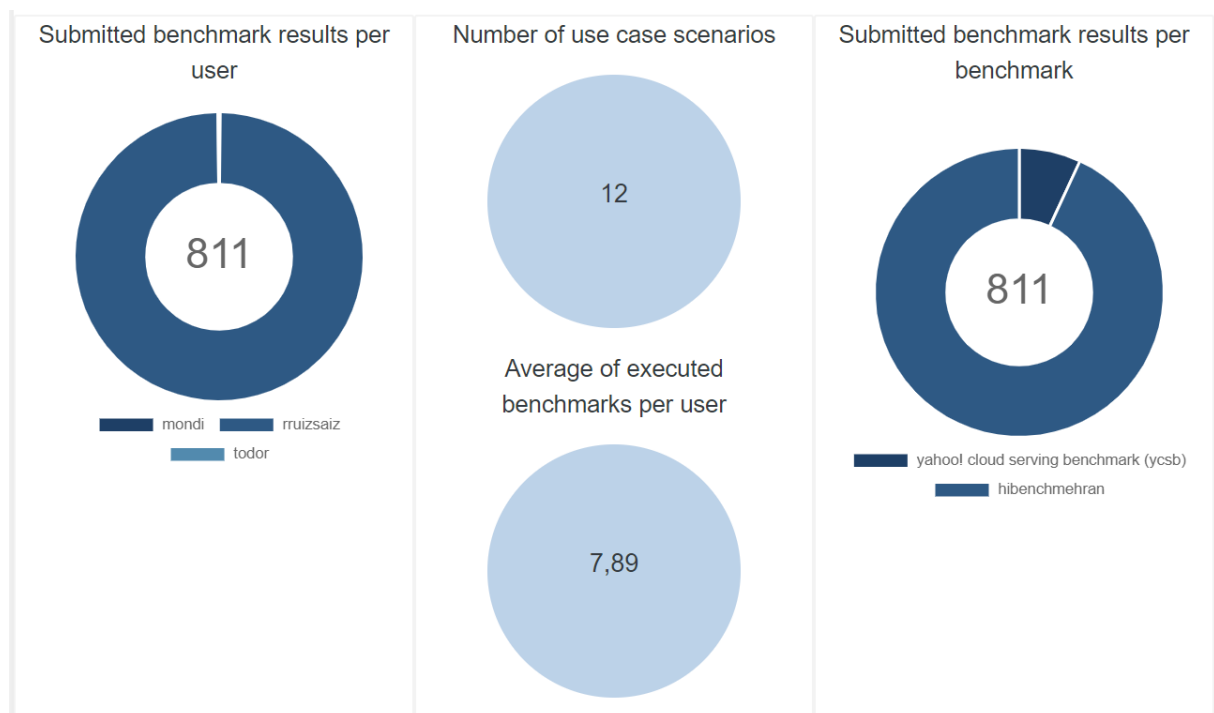


Figure 9 - Showing the first part of the usability metric

"searchCount" also calculates the other sub property of the user usability. This generates the number of searches by benchmark (additionally separated by user and benchmark). Here, the guided searches of the users are read from "application.log" file. The lines that contain "statusSuccess = [Status: SUCCESS]" and "actionSearch = [Action: Guided Search]" substrings

as part of the lines are filtered. Now the count rows store corresponds to each user and in this way is calculated the number of searches for benchmarks by each user. The values are then created again in a JSON string and sent to views as shown in Code Snippet 7.

```
public String searchCount() throws FileNotFoundException {
    File file = new File("C:\\application.log");
    Scanner sc = new Scanner(file);
    sc.useDelimiter("\\n");
    String templine ;
    String statusSuccess ="[Status: SUCCESS]" , actionSearch = "[Action: Guided Search ]";
    Map<String, Integer> searchCountMap = new HashMap<>();
    while(sc.hasNext()){
        templine = sc.next();
        if((templine.contains(actionSearch)) & (templine.contains(statusSuccess))){
            List<User_DB> use_dbb = User_DB.find.all();
            for(User_DB user : use_dbb)
                if(templine.contains("User: " + user.getUserName().toLowerCase()) ||
                    templine.contains("User: " + user.getUserName().toUpperCase())){
                    searchCountMap.put( (user.getUserName().toLowerCase() ,
                        searchCountMap.getOrDefault(user.getUserName().toLowerCase(), 0) + 1);
                }
        }
    }
    String searchesLog = "{ \"dataSets\" : [ " ;
    boolean coma = true ;
    for(String ii : searchCountMap.keySet()){
        if (coma == false) { searchesLog += ","; }
        coma = false;
        searchesLog += "{ \"key\":\"\" + ii + "\", \"value\": "
            + searchCountMap.get(ii) + " }";
    }
    searchesLog += "]}";
    sc.close();
    return searchesLog;
}
```

**Code Snippet 7 - " rateOfSuccessTaskPerUser " Method**

The final version of the second part of the usability metric can be seen in Figure 10.

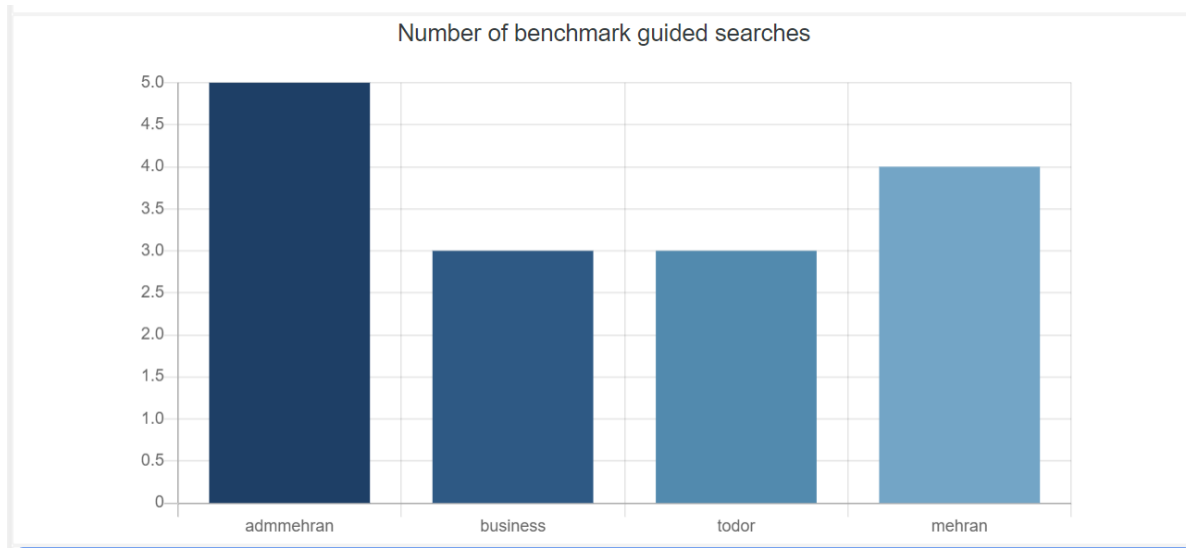


Figure 10 - Showing the second part of the usability metric

The third part of the usability metric "*activeUsers*" method generates the number of users who have performed at least one task in the last month. Users are differentiated between administrators, Technical users, Business users, and Benchmark Providers.

Here the guided search of the users is read from "application.log" file. The subsets "statusSuccess = [Status: SUCCESS]" and "actionSearch = [Action: Guided Search]" and "[Action: Login]" are part of the target query and the number of guided searches is simply calculated by user. The values are then created again in a JSON string and sent to views as shown in Code Snippet 8.

```
public String activeUsers() throws FileNotFoundException {
    . . .
    // Initialize array with random values for 4 groups of users
    . . .
    File file = new File("application.log");
    Scanner sc = new Scanner(file);
    sc.useDelimiter("\n");
    String actionLogin = "[Action: Login]", statusSuccess = "[Status: SUCCESS]",
        actionSearch = "[Action: Guided Search ]";
    Map<String, Integer> searchCountMap = new HashMap<>();
    while(sc.hasNext()){
        templine = sc.next();
        if((templine.contains(actionSearch)) & (templine.contains(statusSuccess))){
            List<User_DB> use_dbb = User_DB.find.all();
            for(User_DB user : use_dbb)
                if(templine.contains("User: " + user.getUserName().toLowerCase()) ||
                    templine.contains("User: " + user.getUserName().toUpperCase())){
```

```

        searchCountMap.put( (user.getUserName()).toLowerCase() ,
        searchCountMap.getDefault(user.getUserName().toLowerCase(), 0) + 1);
    }
}

found = ((templine.contains(actionLogin)) & (templine.contains(statusSuccess)));
if(found){
    lines= lines + templine;
    sixDaysBehind = LocalDate.parse(templine.substring(0,10));
    for(int j =1 ;j < 8; j++){
        if (-Period.between(LocalDate.now().minusDays(5*j), sixDaysBehind).getDays() < 5) {
            if(templine.contains("[is_admin: true]")) { count[j][0] =
                toString().valueOf(Integer.parseInt(count[j][0]) + 1);}
            if(templine.contains("[is_business: true]")){
                count[j][1]=toString().valueOf(Integer.parseInt(count[j][1]) + 1);}
            if(templine.contains("[is_provider: true]")){
                count[j][2]=toString().valueOf(Integer.parseInt(count[j][2]) + 1);}
            if(templine.contains("[is_technical: true]")){
                count[j][3]=toString().valueOf(Integer.parseInt(count[j][3]) + 1); }
            break;
        }
    }
    i=i+1;
    tx= tx + "\n" + templine ;
    found= false;
}
}

boolean coma = true ;
String usersLog = "{ \"dataSets\" : [ \" ";
// generate the JSON String to presentation charts
. . .
usersLog = usersLog + "]}";
return usersLog;
}

```

Code Snippet 8 - *activeUsers*

The chart on Figure 11 shows the results of the methods.



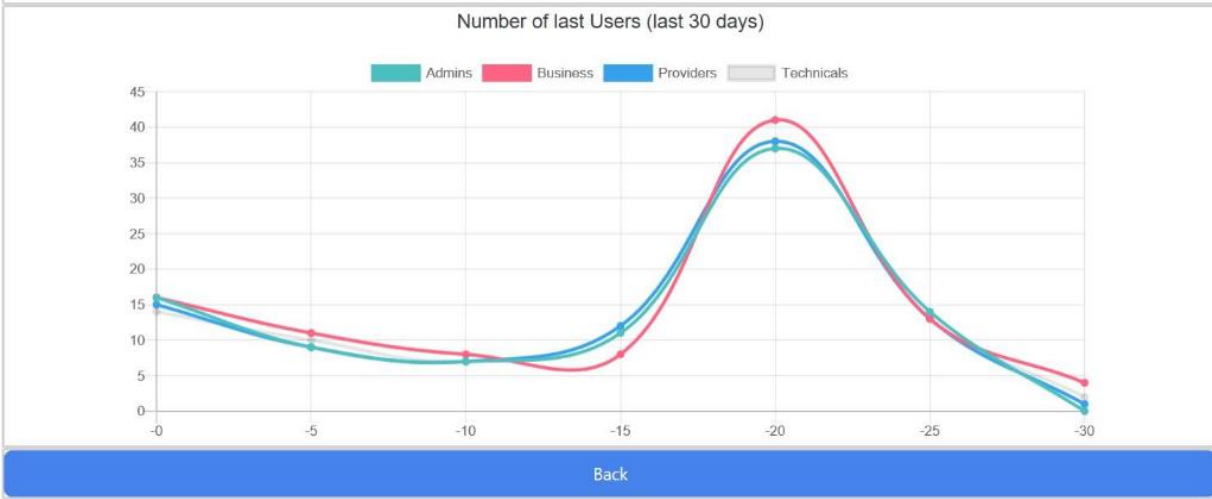


Figure 11 - Number of last Users

## 4. DataBench Toolbox front-end

This final release was mainly focused on improving the user interface of the Toolbox hence most of the changes were done in this area.

In this section we present the evolution of the GUI developed for the Toolbox. This document presents the final release, and therefore takes some information already reported in previous WP3 deliverables (D3.1, D3.2 and D3.3) for clarity sake and reference to those deliverables when needed.

### 4.1 DataBench Toolbox main page

The front-end of the Release version dashboard can be accessed through the following URL:

<http://databench.ijs.si>

**Front Page:** The first page that can be seen when using the Toolbox is the front page. This page has been reworked completely to adopt a look & feel closer to the DataBench project website.

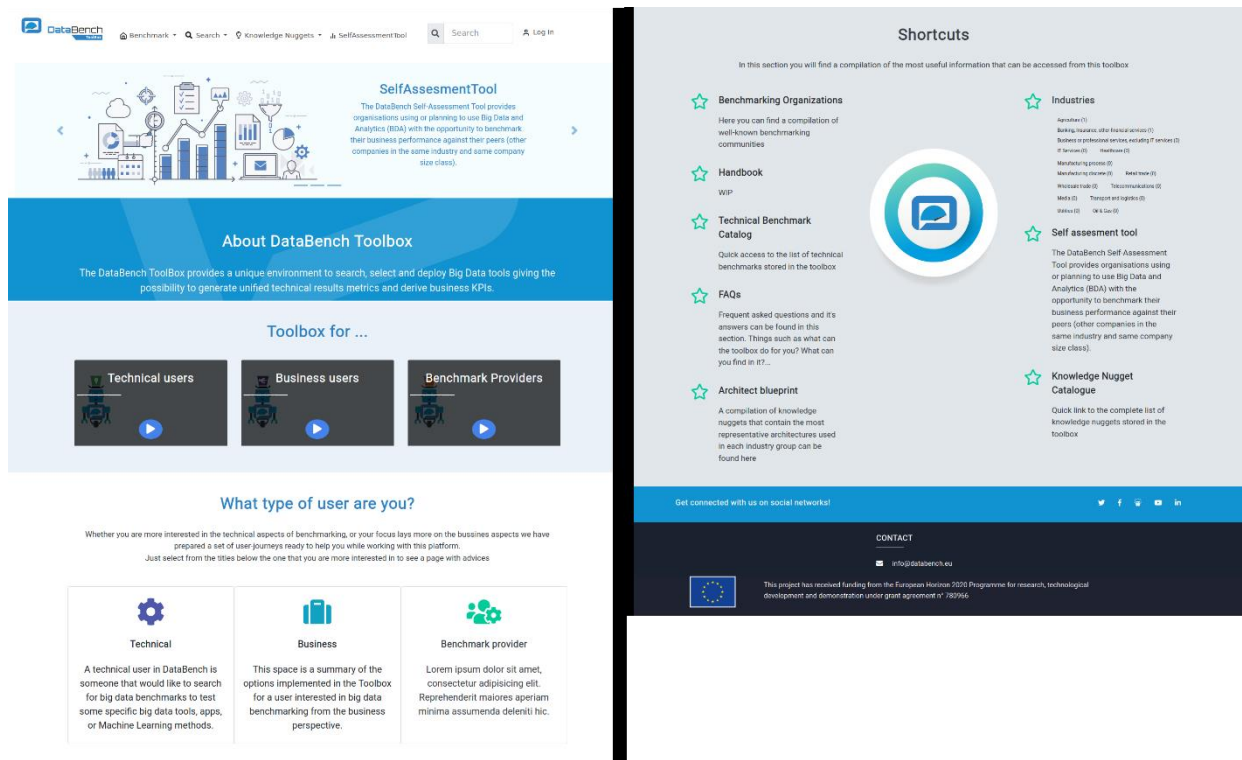


Figure 12 - Main page of the DataBench Toolbox

One of the first things that has been reorganized is the navigation menu. It is now split in 3 groups of dropdowns and two extra links to different parts. Here we will explain the whole navigation bar, but it changes depending on the access level of the user that has logged in.

The menu (navbar) is structured in the following groups of options (It is role-sensitive, which means that users with different role may have more or less options.):

- **Benchmarks:**

The menu option “Benchmark” contains options related to browsing or managing technical benchmarks. All users will have access to the “Benchmark catalogue” option, but only users with the Benchmarking Provider or Administrator roles will be able to access the options that allow to “Register new” benchmarks or “Manage benchmarks” of the benchmark catalogue (update and delete benchmarks). Technical users will be able to access the results of the runs of the integrated benchmarks they executed previously from the “Results” option. Besides browsing their own results, in this last case the user has the possibility to “Share results”, in other words to share metrics of the execution of the benchmarks in the Toolbox with other users. The results stored in the Toolbox are shown formatted as text following the structure of the particular benchmark execution. For some of the benchmarks access to the graphic displays of those results has been developed.

- **Search:**

The menu option “Search” gives access to all users to two main options. The first one is the search using the “BDV Reference Model”. This BDV Reference model can be found in the BDV SRIA [8] and it is at the core of the classification of benchmarks implemented in the Toolbox. The BDV Reference Model is rendered, and clickable in the different boxes, either horizontal or vertical, making it easy to navigate to the Benchmarks or Nuggets related to each box.

Moreover, also under the search option in the menu, users can find the “Guided search” option. This search has also been reworked to make it more useful adding a feature- or tag-based search that replaces the search functionality that was in place in the Beta version, which only covered some of the features. This functionality is further explained in detail in Section 4.4.

- **Knowledge Nuggets:**

The menu gives access to the “Knowledge Nuggets” options, containing similar options as the one for benchmarks. As in the previous case, all users are free to browse the “Nuggets Catalogue”, where the complete list of Nuggets can be seen, but only Administrator users could access to the “Create new Knowledge Nugget” or “Manage Knowledge Nugget” options. This will be explained in detail in Section 4.3.

- **Quality metrics and other options:**

Apart from this menu options, the Toolbox shows links in the menu to other existing tools and options, such as the self-assessment tool, the Project Handbook, or the link to the metrics dashboard explained in detail in Section 4.5.

Besides the changes made in the menu since the Beta version of the Toolbox, the menu includes now other options in the main page further described below such as videos explaining the functionality, links to pages explaining the different user journeys included, Frequent Asked Questions (FAQ), and a set of shortcuts to different important sections of the Toolbox as can be seen in Figure 12.

## 4.2 User journey for user type

In this section we will explain one of the main new features included in the final Release of the Toolbox: The user journeys.

As it was explained in section 2.1, the Toolbox is aimed to three different types of users: Technical, Business and Benchmark Providers. For each one of them, the Toolbox offers a separate web page aiming to help them while getting familiar with the Toolbox and guiding them to interesting resources considering their role.

Moreover, since the expertise of a user inside these 3 groups might be different, we have created a set of user journeys split by the experience of the user. The first group that can be found in the main page is the Technical user journey.

As can be seen in Figure 13, the user is presented with a set of links to different parts of the Toolbox depending if they consider themselves more at a beginner level or a more advanced one. At the beginner level the user is presented with useful links to start working with the Toolbox and answer the most accessible questions. The Toolbox provides links to the FAQs as well as showing the catalogue of benchmarks and Nuggets. Moreover, a set of Nuggets has been identified as an entry level information and is accessible from this section.

In the advanced section, the Toolbox contains links to a set of Nuggets identified as useful for this kind of user. It also explains the way of configuring and running one of the integrated benchmarks from the Toolbox as well as the process to share your results with the Toolbox.

Technical user journey page:

## Welcome to the technical user journey

A technical user in DataBench is someone that would like to search for big data benchmarks to test some specific big data tools, apps, or Machine Learning methods. There are a number of existing benchmarks out there, but unless you are an expert on benchmarking (and even so), it is hard to decide which is the best benchmark, or even if there are benchmarks that suit your needs.

You can browse and search in the benchmark catalogue from the Toolbox without registering, but if you want to have full access to DataBench resources you should register to the Toolbox. It is easy and painless.

User journey

Beginner

As a beginner, you have a set of Knowledge Nuggets (knowledge pieces) to understand what big data benchmarking is and what it can do for you. We recommend you click on the FAQs link in the main page or type "beginner" in the search box and enjoy!

Searching for existing technical benchmarks: Users have the possibility of searching in several ways:

- Search box in the top right corner of the Toolbox. This box allows you to introduce any of the metadata fields.
- Browse the entire catalogue. A user can search for benchmarks guided by the BDVA Reference model just by clicking on any of the boxes.
- Guided search by selecting some of the most used metadata fields

Once located, you could select one technical benchmark and navigate to their own page to look for more content. Benchmarks are marked with metadata indicating their main features. You may be interested in looking at different architectural blueprints. Type "blueprint" in the search box to find them. You can also have a look to the Knowledge Nuggets Catalogue to understand the context

User journey

Advanced

As an advanced technical user, you are supposed to have experience setting and execution big data benchmarks. We recommend you type "advanced" or "intermediate" in the search box and enjoy!

Configuring and deploying for execution some selected benchmarks:

- Some of the benchmarks listed in the catalogue are also integrated in the Toolbox for configuration and deployment.
- These are marked in the list of results from any search or in the catalogue as executable.
- So, the Toolbox allows registered users to configure, deploy and run those benchmarks in your own infrastructure without burdening you with a complex configuration.
- The Toolbox prompts you with a few configuration questions, such as the HOSTS where you would like to deploy the benchmark and few more technical details.
- Nothing an expert technical user cannot easily deal with.
- Et voilà! The benchmark is deployed and the execution enabled or done!
- You will have the results of the run in the proprietary format of the selected benchmark!

Getting the results of the benchmarks into the DataBench platform: You can upload the results of the run to DataBench. This will allow you to have an historical record of the execution that only you can access. The visualization of the results is done via the Results option of the Toolbox.

But there is more. Eventually, the tool will provide further technical and business insights based on previous knowledge, best practices from existing projects, etc.

Figure 13 - Technical user journey

In a similar way as is presented to the Technical user, a business-oriented user can find a user journey defined for them. In it, the user can find a similar set of links in the beginner section such as the FAQs or the different catalogues. For the advance user, we have compiled a set of Nuggets more business-oriented such as quantitative or qualitative KPIs. The page can be seen below in Figure 14

Business user journey page:

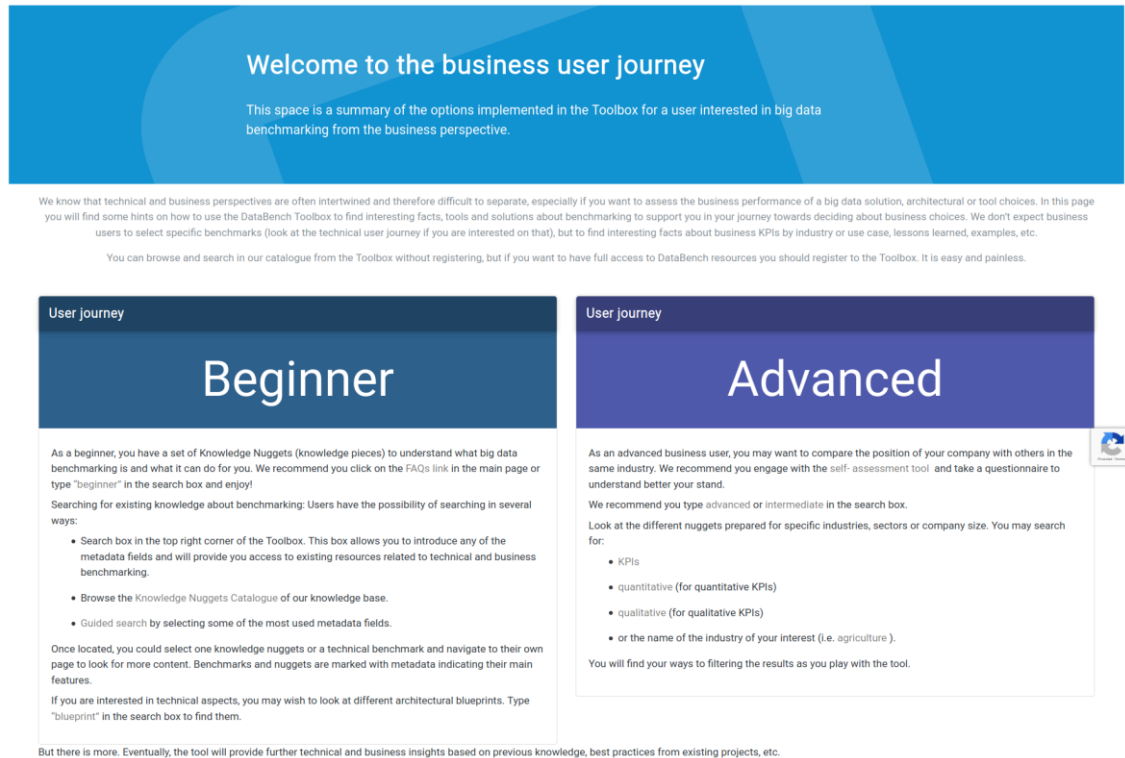


Figure 14 - Business user journey

The current content of these user journeys will be adapted and updated after getting feedback of its usefulness for the users.

### 4.3 Knowledge Nuggets Management lifecycle

Knowledge Nuggets are a new concept introduced in this last version of the Toolbox, as explained in section 2.7. They are small pieces of information, mostly gathered and created during the project, that complement the information about benchmarks and are useful for the users of DataBench to bridge the gap between technical and business perspectives.

These Nuggets share a common view with the benchmarks information and the GUI is presented in a similar way.

The Toolbox provides full CRUD operations over the Nuggets and some screenshots of the generated GUI are presented here:

GUI for creating a new Knowledge Nugget:

CREATE NEW NUGGET

Title:

Title

Description:

Description

Url:

Url

Nugget attachments:

Browse

Nugget tags

Nothing selected

Create new

Figure 15 - Page to create a new Knowledge Nugget

Nuggets catalogue:

The Nuggets catalogue is presented as a simple list like the catalogue of benchmarks but, in the case of the Nuggets, they are presented in 4 columns to emphasize that they are small pieces of information.

List of Knowledge Nuggets

[<- Back](#)

8 KPIs to Measure Big Data Business Impacts

DataBench selected the metrics shown in the figure to measure the business impacts of Big Data. The business KPI definitions are based on business and marketing literature, but have been simplified and operationalized to allow measurement through business surveys and case studies. T...

Quantitative Benchmarks by Industry

This table shows the benchmark improvements of profit, revenues and cost reduction achieved by industry (median value). The financial, business/IT services, telecom/media industries and utilities have the highest benchmark improvements. Manufacturing is aligned with the overall sam...

Quantitative Benchmarks for the Agriculture Industry

Agriculture is one of the industries the industry with the lowest benchmarks in terms of profit, revenues, and cost reductions. The potential impact of BDA in the agricultural sector is impeded to some extent by the inevitable inflexibility of core production processes and the historical trend of s...

Quantitative Benchmarks for the Financial Services Industry

Strong differences are seen in the quantitative KPIs used as benchmarks. Profit and revenues increase presents larger values than cost reduction (i.e. profit increase is almost double than cost reduction). This is not entirely surprising, as the financial services industry has always been a leader ...

Quantitative Benchmarks for the Healthcare Industry

Quantitative Benchmarks for the Manufacturing Industry

Quantitative Benchmarks for the Business/IT Services Industry

[Quantitative Benchmarks for the Retail and Wholesale Industry](#)

Figure 16 - Knowledge Nugget Catalogue

Single Nugget information:

One of the biggest differences between the GUI of a benchmark and the GUI of a Nugget is the inclusion of attachments in the Nuggets. Most of these pieces of information are better understood with the help of an image and the Nuggets take care of this allowing the inclusion of attachments in the concept of the Nugget.

The Toolbox detects these attached images and, if they can be rendered to the GUI (images such as .jpg or .png) they are shown below the description of the Nugget as can be seen in the Nugget example below in Figure 17.

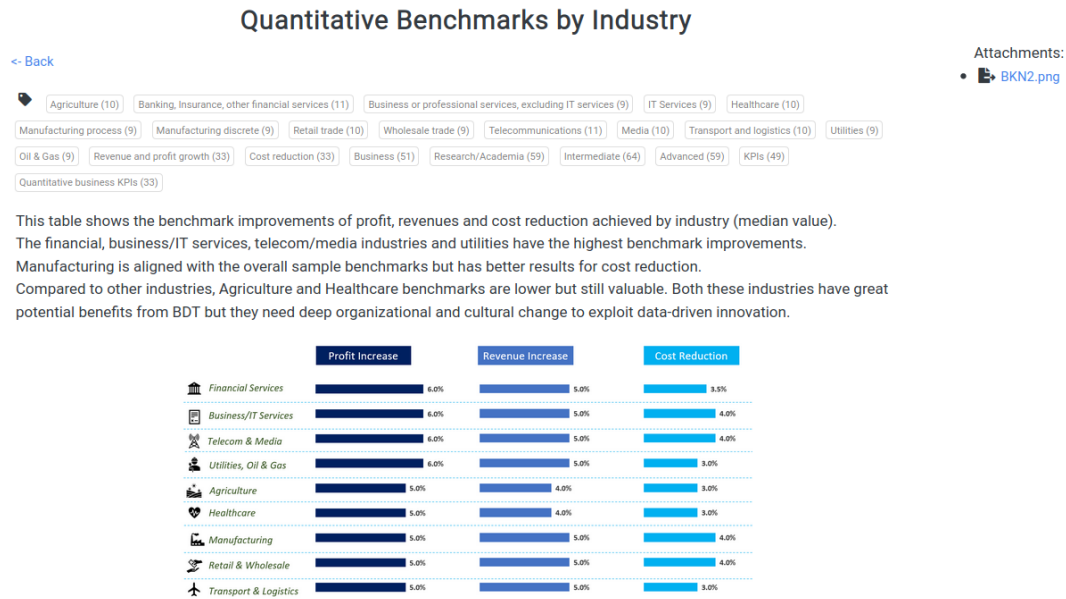


Figure 17 - Example of display of a Knowledge Nugget

## Update:

The GUI for updating a Nugget was created to mimic the one used to create a new Nugget. The only difference is done in the back-end to load the already stored information of that Nugget to be presented to the user updating it and the inclusion of a selector to enable the Nugget after reviewing that the information is correct and follows the needs of DataBench.

### UPDATE NUGGET

Attachments:  
• BKN1.png

Title:

8 KPIs to Measure Big Data Business Impacts

Description:

DataBench selected the metrics shown in the figure to measure the business impacts of Big Data.  
 <br>
 The business KPI definitions are based on business and marketing literature, but have been simplified and operationalized to allow measurement through business

Url:

https://www.databench.eu/wp-content/uploads/2020/01/databench-d.2.4\_1.0-pdf.pdf

Nugget attachments:

Browse

Nugget tags

Business, Research/Academia, Beginner, Intermediate, Advanced, Quantitative business benchmark, Qualitative business benchmark, Qualitative business KPIs, Quantitative business KPIs

Enabled:

True

Update

Figure 18 - Example of update of a Knowledge Nugget



## 4.4 Search functionality

The search functionality is another part that has been reworked completely, mainly due to the addition of the Knowledge Nuggets.

Both the Benchmarks and the Nuggets have been tagged with features to offer extra information and help in the navigation. Taking advantage of these tags, the Toolbox has implemented a feature-based search, that can be seen in Figure 19, making easier for the users to search for information related to their fields of interest.

These features are categorized for a simpler navigation.

The screenshot displays the 'Search' section of the DataBench Toolbox. At the top, a blue banner reads 'Search Search in the platform' and 'Below you'll find the tags' categories that the toolbox presents for searching'. Below this, there are two main search methods:

- SEARCH BY TAG:** This section allows users to 'Navigate through the tag categories:'. It features a list of categories with expandable arrows:
  - Business Features
  - Big Data Application Features
  - Platform and Architecture Features
  - Benchmark-specific Features (expanded):
    - Benchmark data type: Synthetic data (16), Real data (1), Hybrid (9)
    - Benchmark type: Micro-Benchmark (12), Application benchmark (11), Benchmark suite (15)
    - Benchmarking aspect: Fault tolerance (15), Privacy (1), Security (2), Governance (1), Data Quality (1), Veracity (1), Variability (3), Data Management (9)
    - Benchmarking performance metrics: Execution time (26), Throughput (21), Cost (10), Energy consumption (4), Accuracy (4), Precision (2), Availability (3), Durability (3)
    - Execution environment: SandBox/VM (5), In-house/On-Premise (20), Cloud (20)
    - Input data format: JSON (4), XML (2), CSV (4), Proprietary (17), Fixed-sized records (3), Sensor data (5)
    - Output data format: Execution log (13), Timeseries report (3), Pre-defined report (11)
  - Toolbox-specific Features
- SEARCH BY USE CASE:** This section prompts the user to 'Select a use case'. A dropdown menu currently shows 'Nothing selected'. A blue 'Search' button is located below the dropdown.

Figure 19 - Tag-based and use-case-based search functionality

Since we now have the possibility to specify the use case for the benchmarks, a new filter based in this parameter has been added to the search functionality. A new section was created in the guided search to expose that functionality as can be seen in Figure 20.

The screenshot shows a section titled 'SEARCH BENCHMARKS BY USE CASE'. It includes a sub-header 'Select use case' and a dropdown menu with the text 'Nothing selected'. A blue 'Search' button is positioned below the dropdown.

Figure 20. Search by use case



Another possibility is using the search by the structure of the BDV Reference Model to mouse over and click on the desired layer of the model. Benchmarks and Nuggets tagged with those layers will be retrieved, as shown in Figure 21.

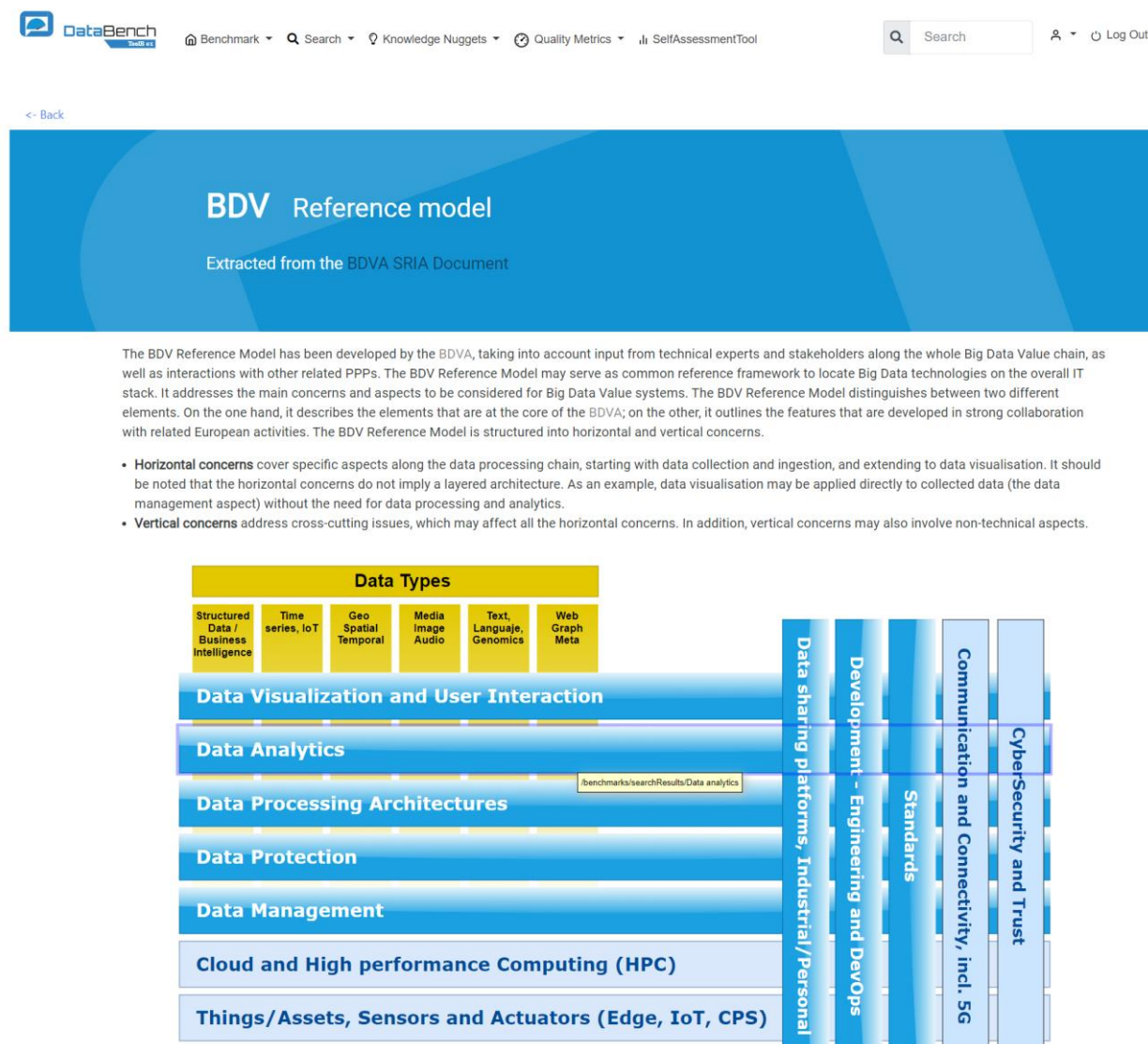


Figure 21. Search by BDV Reference Model

Besides these options located under the “Search” menu option, the front page of the Toolbox offers a free text search box as shown in Figure 22. Users might type any text. The system will match the text with any existing tag or with free text in the name and description of the benchmarks and Nuggets. For partial search matching, users can introduce the character “%” before or after the text (e.g. “spar%” will match elements containing “SPARQL” or “SPARK”), or the “\_” to replace a single character (e.g. “Had\_op”).



Figure 22. Free text search

4.5 Quality Metrics

This dashboard presents the metrics already described in Deliverable 5.3 [9]. It is mainly created for administrators to see the performance of the Toolbox and the usage that the users are making of it. Here we present the GUI created for it. The first page that is seen when entering the dashboard is a compilation of categories as can be seen in Figure 23.

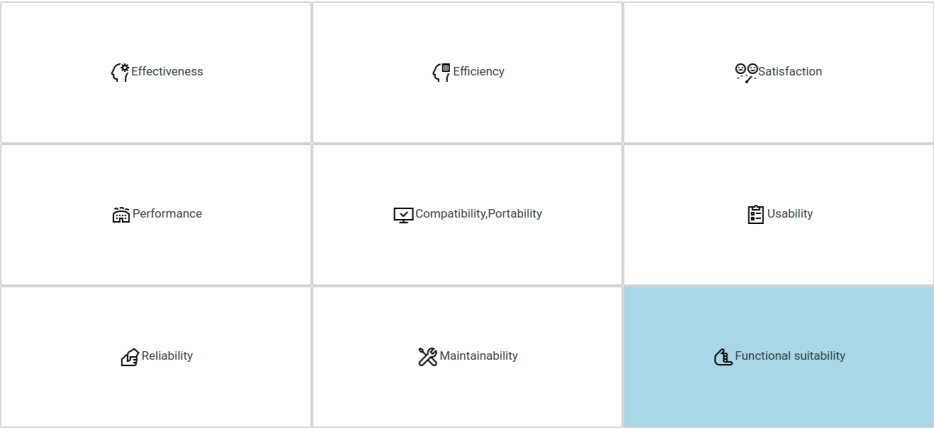


Figure 23 - Quality metrics Dashboard

Each one of these buttons open a new page with different metrics represented mostly as graphs, as can be seen below in Figure 24. These graphs are calculated dynamically mainly from the logs of the Toolbox and gathering aggregated information from the back-end Database. Example of one of the metrics: display

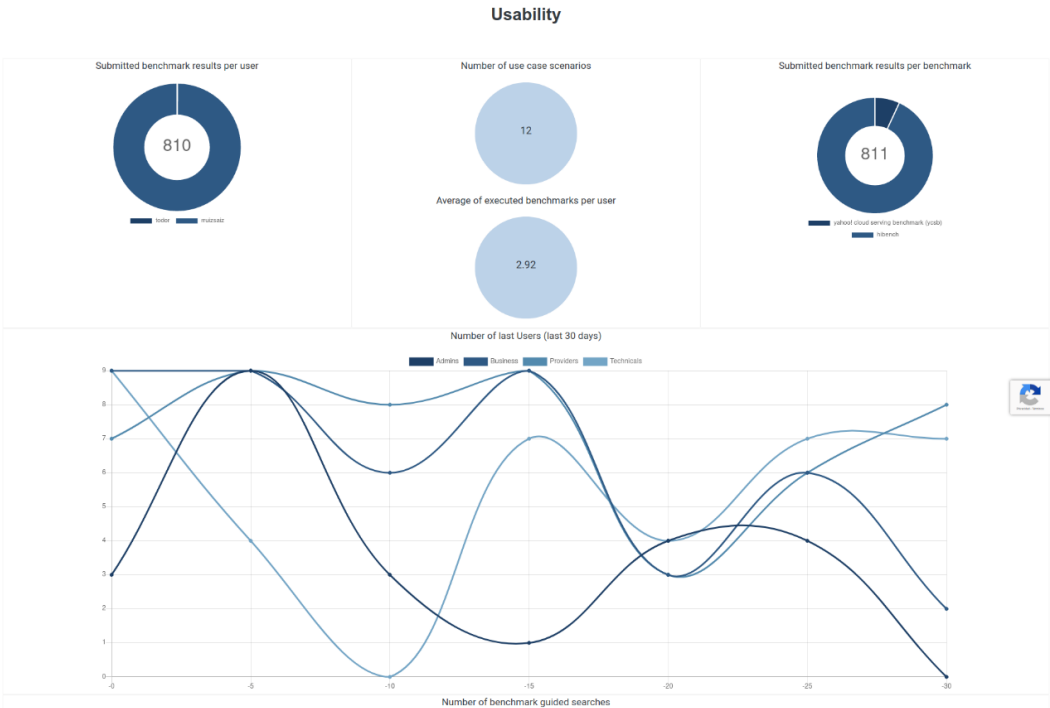


Figure 24 - Example of Quality Metric display

In this section we have presented the updates of the DataBench Toolbox dashboard front-end as released for the Final version by July 2020.

## 5. Conclusions and future work

This document provides a summary and an update of the DataBench Toolbox as it has been delivered for its release version. It is an incremental version based on the same structure and architecture presented for the Alpha and Beta versions presented in the deliverables D3.1, D3.2 and D3.3.

Therefore, the document summarizes the main functionalities presented in previous versions and focuses mainly on the new functions covered in the Release version highlighting the main changes with respect to the previous release.

The current URL accessing to the user interface of the Release version of the Toolbox is public on the web, but we encourage users to create their own user profiles and contact us in order to test the platform and provide feedback. Several early adopters from sister EU-funded projects have been contacted so far, and we also encourage other projects and organizations to test the Toolbox functionality and use some of the integrated benchmarks. We encourage interested users to get in touch via the DataBench contact web page (<https://www.databench.eu/contact/>).

This document is the last one of WP3. The Toolbox has been designed with extensibility points to enlarge and annotate the catalogues of benchmarking tools and Knowledge Nuggets. Therefore, the work on improving and populate content to the Toolbox (i.e. more Knowledge Nuggets or benchmarks), and any significant improvement on its implementation will be reported in upcoming WP5 deliverables.

## 6. References

- [1]. D3.1. DataBench Deliverable D3.1. DataBench Architecture.
- [2]. D3.2. DataBench Toolbox - Alpha including support for reusing of existing benchmarks
- [3]. D3.3. DataBench Toolbox - Beta including end-to-end-scenario tool
- [4]. D1.1. DataBench Deliverable D1.1. Industry Requirement with benchmark metrics and KPIs.
- [5]. D1.2. DataBench Deliverable D1.2. DataBench Framework – with Vertical Big Data Type benchmarks
- [6]. D1.3 DataBench Deliverable D1.3. Horizontal Benchmarks – Analytics and Processing
- [7]. D1.4 DataBench Deliverable D1.4. Horizontal Benchmarks – Data Management
- [8]. Big Data Value Strategic Research and Innovation Agenda (BDV SRIA).  
[http://bdva.eu/sites/default/files/BDVA\\_SRIA\\_v4\\_Ed1.1.pdf](http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf)
- [9]. D5.3 Assessment of Technical Usability, Relevance, Scale and Complexity
- [10]. D2.4 DataBench Deliverable D2.4. Benchmarks of European and Industrial Significance
- [11]. D5.1 DataBench Deliverable D5.1. Initial Evaluation of DataBench Metrics
- [12]. Ansible - <https://www.ansible.com/>
- [13]. Apache Kafka - <https://kafka.apache.org/>
- [14]. CLASS EU project - <https://class-project.eu/>
- [15]. YCSB, <https://github.com/brianfrankcooper/YCSB>, 2018
- [16]. HiBench Suite - <https://github.com/intel-hadoop/HiBench>, 2018
- [17]. Bootstrap documentation - <https://getbootstrap.com/>
- [18]. Play ! - <https://www.playframework.com/>
- [19]. AWX project – <https://www.ansible.com/products/awx-project>