Project Title	High-performance data-centric stack for big data applications and operations
Project Acronym	BigDataStack
Grant Agreement No	779747
Instrument	Research and Innovation action
Call	Information and Communication Technologies Call (H2020-ICT-2016-2017)
Start Date of Project	01/01/2018
Duration of Project	36 months
Project Website	http://bigdatastack.eu/

D1.2 – Innovation Potential: Initial Plan and Activities

Work Package	WP1 – Project Coordination
Lead Author (Org)	Amaryllis Raouzaiou (ATC)
Contributing Author(s) (Org)	Amaryllis Raouzaiou (ATC), Stathis Plitsos (DANAOS), Sophia Karagiorgou (UBI), Stavroula Meimetea (UPRC), Yosef Moatti (IBM), Pavlos Kranas (LXS), Richard Mccreadie (GLA), Orlando Avila-García (ATOS), Jean-Didier Totow (UPRC), Ismael Cuadrado-Cordero (ATOS), Bernat Quesada (ATOS), Christos Doulkeridis (UPRC), Peter Jason Gould (UPRC), Mauricio Fadel Argerich (NEC), Bin Cheng (NEC), Dimitris Poulopoulos (UPRC), Paula Ta-Shma (IBM), Marta Patiño (UPM), Nikos Drosos (SILO), Luis Tomas Bolivar (RHT), Maurizio Megliola (GFT)
Due Date	30.11.2018
Date	30.11.2018
Version	1.0

Dissemination Level

X	PU: Public (*on-line platform)
	PP: Restricted to other programme participants (including the Commission)
	RE: Restricted to a group specified by the consortium (including the Commission)
	CO: Confidential, only for members of the consortium (including the Commission)

The work described in this document has been conducted within the project BigDataStack. This project has received funding from the European Union's Horizon 2020 (H2020) research and innovation programme under the Grant Agreement no 779747. This document does not represent the opinion of the European Union, and the European Union is not responsible for any use that might be made of such content.

Versioning and contribution history

Version	Date	Author	Notes
0.1	03.10.2018	Amaryllis Raouzaiou (ATC)	Outline
0.2	26.10.2018	Amaryllis Raouzaiou (ATC)	Section 2. Adding contributions from DANAOS (section 7.4.1) and UBI (section 7.3.2)
0.3	06.11.2018	Amaryllis Raouzaiou (ATC)	Sections 3 and 4. Adding contributions from UPRC (section 7.3.3).
0.4	12.11.2018	Amaryllis Raouzaiou (ATC)	Section 5. Adding contributions from IBM and LXS (sections 7.2.1, 7.2.4)
0.5	14.11.2018	Amaryllis Raouzaiou (ATC)	Section 6. Adding contributions from SILO (section 7.1.6)
0.6	20.11.2018	Amaryllis Raouzaiou (ATC)	Adding contributions from ATOS (sections 7.1.4, 7.1.5, 7.4.2), GLA (section 7.1.3).
0.7	23.11.2018	Amaryllis Raouzaiou (ATC)	Sections 1 and 8. Adding contribution from UPRC (sections 7.2.7, 7.3.1, 7.2.6), NEC (section 7.1.2), IBM (section 7.2.5), UPM (section 7.2.8)
0.8	26.11.2018	Amaryllis Raouzaiou (ATC)	Adding contribution from IBM (sections 7.2.2, 7.2.3)
0.9	27.11.2018	Amaryllis Raouzaiou (ATC)	Adding contribution from RHT (section 7.1.1), GFT (section 7.4.3)
1.0	30.11.2018	Amaryllis Raouzaiou (ATC)	Final version following internal review

Table of Contents

List of tables

List of figures

1 Executive Summary

The current document reflects the first deliverable of Innovation Management, which identifies and specifies the approach of BigDataStack consortium for the Innovation. Moreover, it presents the priorities we set up based on Big Data Value Association (BDVA) priorities, the Innovation Strategy Plan and the IPR management approach. The first results of innovation monitoring throughout the different project activities are also presented. A section is devoted to each main building block of the architecture (i.e. data-driven infrastructure management, data-as-a-service, as well as dimensioning, modelling, and interaction services), presenting the innovation-monitored components, the related priorities, their expected innovation and the progress towards it. Another section describes the expected innovation of BigDataStack use cases and their approach.

In the course of the project, the BigDataStack components and use cases will continue being monitored to ensure that the innovation potential of the project will be fully exploited. A follow-up deliverable (i.e. D1.3) summarizing BigDataStack activities relevant to Innovation Potential will be delivered at M34.

2 Introduction

2.1 Purpose and Scope

BigDataStack deploys best practice Innovation Management methods to reduce the gap between project results and targeted market, ensuring an earlier adoption of project outcomes. BigDataStack Innovation Management is based on procedures inspired on the European standard for innovation management, CEN/TS 16555-1 [1], an innovation management system supported by interfaces and interactions between scientific and technological knowledge, knowledge about the organization and the market, as well as society as a whole.

The adopted innovation management procedures reinforce and guarantee that: (i) The expected results are aligned to market needs, including needs of big data community, and data providers' communities; (ii) The results are disseminated and reach their target groups; (iii) There is a clear business exploitation strategy for the BigDataStack solutions.

BigDataStack Innovation Strategy is based on the priorities identified by Big Data Value Association (BDVA) [2], examining the relevant potential not only of BigDataStack framework, but also of its individual components and the proposed use cases.

Innovation Management is ensuring that all partners are well prepared for the challenges of market-oriented exploitation [3], in order to convert research results into innovative success stories, keeping the technological achievements always connected and relevant to market requirements and trends.

The Innovation Management activity is coordinated by the Innovation Manager (IM) of BigDataStack who is supported by the rest of the members of the Project Management Board.

2.2 Relation to other deliverables

The current deliverable, the first BigDataStack deliverable concerning Innovation Potential (the second one – *D1.3- Innovation Potential: Summary Plan and Activities*- is scheduled for M34) is related to several other BigDataStack deliverables in a direct or indirect way.

D2.1 (State of the art and Requirements analysis - I) contains a first description of how the potential innovation of BigDataStack outcomes can help to better position the proposed solutions in the market. *D2.4 (Conceptual model and Reference architecture - I)* provided information for BigDataStack concept, while there is a close relation with all the deliverables of *WP7 (Communication, Exploitation, Standardisation & Roadmapping)* concerning Exploitation and Dissemination of project results.

2.3 Structure of the document

The document is structured as follows:

- *Section 3* presents the BigDataStack approach for Innovation Management, along with the relevant activities.
- *Section 4* provides description of the Innovation Potential of BigDataStack, based on the priorities defined by BDVA.

- *Section 5* introduces the project's Innovation Strategy Plan.
- *Section 6* provides information concerning the monitoring of IPR management and licensing.
- *Section 7* presents the innovation within BigDataStack per main building block of the overall solution and per use case.
- Finally, *Section 8* contains the conclusions of the current deliverable.

3 Innovation Management: the BigDataStack approach

Innovation Management monitors the progress of R&D activities in the project to ensure relevance and alignment to market and technology advances and emerging trends.

BigDataStack is the product of not one, but 14 organisations with complementary technical expertise as well as business and market involvement, providing specific solutions targeting at business agility and competitiveness, increasing the success rate of project innovation. The industry contributes to successful market-oriented exploitation by ensuring that research outcomes are in line with market expectations, while academic partners receive valuable feedback from the research community. The IM should ensure that all partners know the innovation that is relevant to their type of organization and the related risks.

BigDataStack Innovation Management approach follows mainly the guidelines described below:

- protect innovation before the dissemination of project results – an exception should be made regarding the scientific publications, since they constitute the most reliable way for acquiring feedback from the scientific community;
- special dissemination strategy will be followed for specific industries and experts outside the consortium;
- identify and acquire feedback from different entities and communities (e.g., advisory board, related projects, big data communities etc.) to better align the project results with users' expectations;
- specify "innovation-related activities" like monitoring, risk management etc.;
- take care of the external licenses that are necessary for the project;
- update and adapt the innovation plan as market trends evolve, in order to be sure that exploitation is more accurate and aligned with relevant tendencies.

3.1 Innovation Management Activities

Rather than a chain of sub-sequent steps, Innovation Management is a continuous process consisting of closely interrelated activities. During the project execution, specific innovation activities take place in order to increase BigDataStack Innovation Potential:

- Continuously support the strong collaboration between research institutes / universities and companies of the consortium, to establish a relation between research and market trends;
- Obtain good knowledge of the market and the state-of-the-art;
- Involve end-users, mainly through the proposed use-cases;
- Engage the involved technical partners in an open source approach;

- Protect and manage Intellectual Property Rights;
- Demonstrate through use cases. The project will connect with big data solution providers, data providers and innovators throughout the value chain and with emerging technical developments, in order to create insights.

During the last year of the project, new activities should be defined, as for example:

- Make a report on the main innovations and achievements of the project;
- Evaluate the maturity of the project technical results;
- Provide input to the exploitation plan of the project;
- Evaluate use cases results, ensuring that innovations and project outcomes are valid and thus applicable to different cases and scenarios.

4 Innovation potential relevant to BigDataStack

BigDataStack Innovation plan uses as a basis the priorities defined by the Strategic Research and Innovation Agenda developed by BDVA [2]. The different components of the project use as a roadmap the defined priorities, ensuring the innovation of BigDataStack.

4.1 Data Management

Large amounts of data are being made available every second. While the collected data are increasing, the tools of data management don't evolve at the same way. Data and data analytics methods need to be combined, in order to facilitate their use by the end-user.
BigDataStack targets at the "Data Management" priority with the Data-as-a-Service offering of the project. The proposed underlying LeanXcale database addresses the need to handling unstructured and semi-structured data in an optimum way, semantic interoperability is realized through an innovative domain-specific to domain-agnostic loop, while data quality assessment and improvement as well as data cleaning, all offered by the project, aims at the provision of data quality guarantees.

4.2 Data Processing Architectures

At the beginning, Big Data were only derived from data kept in storage - data-at-rest – using existing architectures and technologies. The appearance of data streams from sensors or social networks-data-in-motion- modified the processing methods, while the processing of the combination of data-in-motion and data-at-rest is one of the most important challenges of Big Data area.

BigDataStack enables real-time analytics for data in motion and at rest, while the overall infrastructure management system is based on a data-driven decentralized architecture being scalable and adaptable to enabling processing of very large amounts of data in an efficient and performant way.

4.3 Data Analytics

In the era of Big Data the understanding of the data is really important. Numbers, text, multimedia content and other formats can be part of huge datasets and new data analytics services are becoming necessary.

The data analytics tools of BigDataStack includes pattern discovery, machine learning and reasoning and deep learning techniques, while emphasis is put on parallel, scalable and incremental analytics.

4.4 Data Visualisation and User Interaction

Data visualization is really important for the exploration and exploitation of Big Data. The results' presentation should facilitate the quick and correct interpretation of the Big Data and the relevance of information.

The Visualization Environment of BigDataStack will be adaptable to data sources and datasets, addressing this priority.

4.5 Data Protection

Data protection is really important for Big Data. With more than 90% of today's data having been produced in the last two years, a huge amount of person-specific and sensitive information from different data sources is being collected.

BigDataStack will put in place the necessary regulations for data management and analysis to ensure privacy, setting up a dedicated to BigDataStack testbed environment fully compliant with ISO/IEC 27000 standard, encrypting also data-in-motion. Emphasis will be given on the EU General Data Protection Regulation (GDPR) [4].

4.6 Big Data Standardisation

Standards are necessary in the area of Big Data in order to ensure compatibility and interoperability, facilitating the product presentation to the market.

BigDataStack structures and procedures ensure that the project uses and is in line with the relevant global standards. In addition to that, BigDataStack develops the European Open Source Initiative coordinated by Red Hat to facilitate the exploitation of open source artefacts produced by different entities. Through the European Open Source Initiative, that will be open to the whole EU research community, the outcomes of BigDataStack will be promoted, while the discussions realized in the framework of this open source community are expected to lead to novel developments and boost innovations.

4.7 Engineering and DevOps for Big Data

New engineering methodologies are necessary for building next generation Big Data systems. Engineering and DevOps tool chains for Big Data Value systems need to look at and systematically integrate a diverse set of aspects for: (1) system/software engineering; (2) development and operations; and (3) quality assurance [2].

BigDataStack provides novel infrastructure management in which all decisions including resource allocation, admission control, scaling, orchestration, runtime migration and adaptations, are data-driven. This data-driven approach analyses and incorporates the interdependencies between computing, storage and networking resources and drive resource management decisions. Adaptations of all BigDataStack components will be also realized in order to meet the evolving needs of data operations and applications during runtime.

5 Innovation Strategy plan

The main goal of BigDataStack Initial Innovation Plan is to ensure the innovation of the project follows the changing market trends and needs. During the project duration we continuously monitor the state-of-the art and the market breakthroughs and we compare the project innovation with the BDVA priorities (see *Section 4*). This is achieved as follows:

- The leaders of the technical WPs (i.e. WP3, WP4 and WP5) that represent the main building blocks (i.e. data-driven infrastructure management, data-as-a-service, dimensioning, modelling, and interaction services), in cooperation with the contributing partners, define the innovation related to each block. The innovation could concern the whole block (as a bundle), a part of the block or a specific component.
- The responsible of the innovative component is describing the component, comparing it to existing solutions and describes the expected innovation.
- The Innovation Manager, in collaboration with main building block leaders, updates, if necessary, periodically the innovation approach, based on the above-mentioned reports, the state-of-the-art and the market needs.
- Similar approach is followed for use cases. Every use case responsible describes the special features of the use case and the expected innovation.
- The Innovation Manager in collaboration with the use case responsible updates, if necessary, the use case approach in relation to innovation.
- Both Innovation Manager and Exploitation Coordinator monitor every dissemination activity to ensure that the published information doesn't harm the innovation potential of the project results.
- If an innovation-related risk emerges, the Innovation Manager in collaboration with the corresponding WP leader and the Project Management Board will redefine the approach of the project for this specific issue.

6 Monitoring IPR management and licensing

The issues related to Intellectual Property Rights (IPRs) have been carefully defined in the Consortium Agreement (CA) signed at the beginning of the project. The CA defines all the guidelines that regulate the access rights to the different IPRs, both for the contributed background, and for the foreground or results. In order to ensure a smooth execution of the BigDataStack, the project partners agree to grant each other royalty-free Access Rights to their Background and Results for the execution of the project. Results are owned by the project partner carrying out the work leading to such Results. If any Results is created jointly by at least two project partners and it is not possible to distinguish between the contribution of each of the project partners, such work are jointly owned by the contributing project partners. The same applies if an invention is made having two or more contributing parties contributing to it, and it is not possible to separate the individual contributions. Such joint inventions and all related patent applications and patents will be jointly owned by the contributing parties. Some of the project partners are using Open Source code in their deliverables or contributing their deliverables to the Open Source communities; Access Rights for Implementation and Exploitation of Results and Background that are Open Source Software are granted under the Open Source License, and not under the terms of the CA.

The licensing of BigDataStack will be driven by the central aim of the consortium to provide benefit to the European research and Open Source communities.

7 Overview of Innovation in BigDataStack

BigDataStack offers a complete solution for data operations and data-intensive applications addressing the complete data lifecycle through the envisioned infrastructure management system, the Data as a Service offering, the Application Dimensioning Workbench, the Process Modelling Framework, the Data Toolkit and the visualization environment. The innovation potential of BigDataStack is multifold:

- Delivering offerings of high-performance: The BigDataStack distributed predictive analytics and real-time CEP, as well as the infrastructure management system that includes several control loops, ensure end-to-end performance optimization of all offerings (allocation of resources and services, deployment patterns, dynamic orchestration decisions, traffic engineering and network management, adaptable re-distribution of storage, analytics tasks etc.).
- Increasing adaptability and robustness of data-oriented environments: The final BigDataStack environment will be fully adapted and scalable during runtime following monitoring of all elements (i.e. application components, data services and infrastructure resources) and adaptations to deployment patterns, resources and services allocation, network management and storage distribution decisions.
- Promoting data-driven industrial innovation and competitiveness: The BigDataStack delivers a complete Data as a Service offering realizing the concept of data-driven innovation [5]. The latter along with the Process Modelling Framework allow commercial companies to exploit the derived knowledge from data towards the development of improved processes, services and products.
- Boosting the data-driven economy through new business opportunities: Decision makers will directly “exploit” Data-as-a-Service through the Data Toolkit and Visualization Environment. The decision makers can experiment with different preferences (through the toolkit) and obtain business-related outcomes as things happen (i.e. incremental analytics providing partial results in advance). Furthermore, for the use cases specific business parameters have already been identified to improve existing businesses and open new opportunities.
- Reducing operational costs: The envisioned Process Modelling Framework and the process analytics / mining algorithms along with the Data as a Service offering will provide meaningful data insights to business operations and processes and thus drive decisions for business agility, while the infrastructure management system will enable resource effectiveness through innovative data-driven admission control technologies.

- Facilitating wide use of technologies from different stakeholders and guaranteeing quality of information: The proposed Application Dimensioning Workbench, Data Toolkit and Process Modelling Framework will allow different stakeholders (e.g. programmers, big data practitioners, data scientists, decision makers, etc) to exploit the added-value of big data.

The BigDataStack elements we monitor in relation to the innovation potential of the project, are detailed in the following sections.

Please note that additional technical details for the listed components / innovations are provided in the technical deliverables of the project (D3.1, D4.1 and D5.1).

7.1 Innovation within Data-driven Infrastructure Management

Table 1 presents the monitored components of the Data-driven Infrastructure Management.

Component	**Related priority** *(see Section 4)*	**Category** (*Technology / Research / Business*)	**Possible risk**
Cluster Management	- Engineering and DevOps for Big Data - Data Processing Architectures	- Technology	- As this component is based on operators, that are an emergent technology, there may be extra unexpected work to be done.
Dynamic Orchestrator	- Data Processing Architectures	- Technology	- Online learning is too slow to learn what deployment changes should be performed. - The Dynamic Orchestrator logic is not stable, i.e. does not correct requirements or SLOs violations in a small enough number of steps.
Ranking & Deployment	- Data Processing Architectures	- Technology	- Insufficient training data leads to poor deployment configurations to be selected. - Metrics captured are insufficient to identify effective system configurations.
Triple Monitoring Engine	- Data Processing Architectures	- Technology	- Lack of metrics, data congestion.

			- Publish channel can be closed.
QoS Evaluator	- Data Processing Architectures	- Technology	- Too strict agreements may impair the adaptability of the architecture
Information-driven Networking mechanisms	- Data Processing Architectures - Engineering and DevOps for Big Data	- Technology	No risk

Table 1 – the monitored components of Data-driven Infrastructure Management

7.1.1 Cluster Management

Existing solution

Nowadays applications containerization and container orchestration engines are becoming the most standard way to build and deploy applications due to the current DevOps and Cloud Native application trends. Among the orchestrators, the de-facto standard is Kubernetes. Consequently, in BigDataStack we decided to target containers as our development/deployment model and OpenShift as our cluster management engine. OpenShift is based on Kubernetes for the containers orchestration, but it includes extra functionalities targeting the DevOps model, such as the possibility of automatically building containers images from git repositories or handling applications upgrades, among many others.

However, in order to increase the impact and usability of BigDataStack components we need to ensure it can be used either on-premise (physical servers or private clouds) as well as on public clouds. One limitation regarding this is the lack of optimized support for installing OpenShift (and BigDataStack on top) on top of private clouds (in this case OpenStack-based as it is the de-facto standard). There is a need for better integration to boost the performance, especially regarding the network data plane.

In addition, regardless of being containerized, current applications are sometimes hard to manage over its life cycle, including actions such as application scaling or upgrades. There are already plenty of automatization tools but there is still a gap regarding life cycle management, getting better insights about the application status and self-healing/self-managed applications.

BigDataStack approach

BigDataStack proposes to tackle the above mentioned limitations by:

- Improving OpenShift integration support when running on top of OpenStack based clusters. For on-premise clouds further optimizations can be made to, for instance, avoid double encapsulation problems (through Kuryr integration). This support will also be based on operators (see delivareble D3.1 for more information) and will permit even managing the OpenShift cluster scaling options from within the OpenShift cluster, i.e., creating extra OpenStack VMs (and configuring them) to extend the OpenShift cluster when there is a need for more resources.

- Making extensive used of operators to better handle big data stack lifecycle. For instance, we will target to have a Spark operator that will enable not only installing an Spark cluster on top of OpenShift, but scaling it out/down at any point in time with simple actions, and by using the OpenShift API, thus enabling an easy integration for the other components of the big data stack.

Results so far

Initial work has targeted the OpenShift Installer integration with OpenStack as well as some initial steps on the Kuryr integration. This work is being done together with upstream development, thus focus on operators. On the plus side this will make it maintainable after the project ends, but it also imposes some extra work as we need to accommodate the upstream needs that may not always be directed related to the BigDataStack work.

In addition, as part of BigDataStack we are extending kuryr to better support OpenShift functionality on top of OpenStack. Among others, we have submitted upstream initial patch sets to add support for Network Policies as they are needed for data/traffic isolation between different components/users.

Expected Innovation

BigDataStack aims at bringing automated management of the complete stack, from the OpenShift cluster management up to the operators running on top. It will enable not only the installation and automatization support, but also life-cycle management, insights and self-healing/self-manage capabilities for the BigDataStack components. In addition, it expects to facilitate integration with other components as operators APIs are exposed through the OpenShift API, ensuring a defined and stable API over time.

7.1.2 Dynamic Orchestrator

Existing solution

Even though most modern streaming processors such as Apache Spark, Apache Storm or Hadoop offer dynamic scaling for dealing with workload changes during runtime, the user still needs to properly understand how to manage a non-trivial number of parameters in order for its application to adapt correctly. However, this is a very complicated task because it is difficult to anticipate the workloads the application will encounter and how each parameter affects the others. This often leads to poor application performance and/or resources under-utilization. In addition, applications typically have opposing objectives, e.g. latency and accuracy, which need to be balanced to deliver the best service possible.
To address this problem, at NEC we have already proposed a methodology to orchestrate applications to their current execution environment. By dynamically orchestrating the application deployment, applications can comply with their non-functional requirements or Service Level Objectives (SLOs) in different execution environments while processing different workloads. The orchestrating logic currently used is based on simple heuristics which have provided good performance in some cases, but might not perform as expected with other applications because they depend on a linear trade-off between the application objectives.

BigDataStack approach

In BigDataStack, we aim to innovate by developing a Dynamic Orchestrator which uses Reinforcement Learning to adapt applications to their variable and demanding workload. The Dynamic Orchestrator will help applications to comply with their requirements or SLOs during runtime and improving the use of system resources.
Reinforcement Learning has been chosen because of its capability of adapting the orchestrating logic to different applications, objectives and workloads thanks to its online learning process. This means that the Dynamic Orchestrator does not need to know in advance about the workload of the application, instead, it will learn about it in an on-going basis and update its knowledge over time making it more flexible than any other rule-based alternative. In addition, Reinforcement Learning can perform with little computational effort, ensuring a quick response to changes.

Results so far

The dynamic orchestration problem has been analysed and framed as a Reinforcement Learning problem, by defining the states-action configurations as well as the reward function to be used. The states are determined by the current metrics of the system, while the actions are referred to the type of change needed in the deployment to improve the current performance. The reward instead, gives a positive feedback if the applications is complying with all its requirements or SLOs, and a negative value proportional to the requirements violation otherwise.
In addition, an early prototype of the Reinforcement Learning based logic for the Dynamic Orchestrator has been developed and tested in a simple use case of a video processing application by means of simulation. In these tests, Reinforcement Learning has proven to be able to adapt the application behaviour to different workloads.

Expected Innovation

The application of Reinforcement Learning to orchestrate applications and data services deployment using multiple objectives and requirements is a novel approach which, in our knowledge, has not been addressed by any other party yet. The definition of the Reinforcement Learning approach for this task, by defining its elements (states, actions and reward) and their relationship to the applications' multiple requirements or SLOs, is a relevant contribution to the fields of self-adaptive systems and data science.
In the BigDataStack, the Dynamic Orchestrator will improve the behaviour of applications and their response to workload changes, as well as simplifying the Business Analysts and Developers work that will not have to tune their applications for scaling.

7.1.3 Ranking & Deployment

Existing solution

Currently, there is a growing need for easy means for in-expert (business) users to deploy complex data analytic applications on the cloud. However, this is challenging, as it involves identifying suitable hardware for those applications, as well as performing subsequent deployment and maintenance on the cloud. Popular solutions employed by companies to this issue tend to follow one or two strategies, depending on the availability of internal expertise within the company. First, if the application/software services were developed in-house, the developers or other IT experts within the company will handle the deployment of the application. Otherwise, companies may make use of external 'cloud consultants' (such as https://cloudspectator.com/) to handle the benchmarking and optimization of their workloads for an additional fee.

BigDataStack approach

BigDataStack aims to innovate in this space by providing the benchmarking, optimization and deployment of cloud deployments in a fully automatic manner, eliminating the need for experts to manually test and monitor user applications. Ranking and Deployment forms a critical link in achieving this, as it provides: 1) the means to automatically integrate benchmarking information about an application with the user's requirements and preferences, enabling the selection of effective deployment configurations that minimise resource waste and cost to the user; and 2) the ability for fully automatic deployment of the user's application based on the selected configuration. BigDataStack will use state-of-the-art supervised machine learning techniques to learn how to effectively configure user applications based on similar deployments in the past.

Results so far

Research and development of the Ranking and Deployment components of BigDataStack are still in their early stages. The first implementation of the Ranking component has been developed, tested and shown to be viable, which provides automatic identification of effective application configurations based on the output of the Application Dimensioning Workbench. However, this is currently a rule-based (heuristic) baseline approach, which will be replaced with superior machine learned models in future releases. The Deployment component is queued for development. In addition, to facilitate the testing of the Ranking and Deployment functionality, a test system was created. This test system was developed to provide a way for BigDataStack developers to analyse the inner workings of application deployment in general, and the Ranking functionality in particular.

Expected Innovation

BigDataStack aims to innovate in two main areas. First, from a high-level perspective, BigDataStack will provide automatic identification and selection of effective deployment configurations for user applications, reducing the need for expert-involvement in the process. Second, by using supervised learning to produce a data-driven approach, we expect to that reductions in application deployment costs for user applications can be achieved against current (manual) solutions, as lower cost deployments that still meet the users' needs can be identified in less time.

7.1.4 Triple Monitoring Engine

Existing solution

The Triple Monitoring Engine via its pub/sub interface allows consumers to subscribe to metrics then receive information related to those metrics in a streaming mode. The metrics' collector used by the triple monitoring engine is Prometheus. The latter has an alerting mechanism which can send information (metrics and some other data related, such us: time, value, etc.) after having reached some threshold. This native method is used to implement a pub/sub interface using Prometheus.

Exporters and the push gateway are the two methods by which Prometheus collects metrics. The Prometheus' library provides classes to build an exporter. The classic way embeds a http endpoint by which metrics are exposed. Our local requirement (requirement of the triple monitoring) imposes us to build an exporter for metrics concerning the engine coming from different separated components. The technic used now a day is a separated exporter for different component.

BigDataStack approach

To be able to collect metrics from different components of the same engine, we used different approach saving resource (processing power and memory). Our solution consists of building a REST-API that receives metrics from different components of the triple monitoring engine then exposing them to Prometheus. The receiving rate and the Prometheus scrape time should be harmonized so that to do not have a buffer. If those two parameters are not synchronized, we could lose some metrics.
BigDataStack has many candidates for consuming metrics using the pub/sub interface, we built a manager that is connected to a queuing system (RabbitMQ [6]). The manager registers subscriptions then queries Prometheus to get metrics. In order to allow the manager to listen permanently for new subscriptions and publish anytime we used two different channels on two different connexion objects running in separated threads. For technology limitation reason, the API (RabbitMQ client) interrupts (close) the channel used for publishing after some inactivity time.

Results so far

We are testing the channel status before each operation (publish). In case of closed state, the connection is reset for avoiding bug. Concerning the loss of metrics, we synchronized the arriving rate and the Prometheus scrape interval. Thus, currently the overall solution allows for monitoring of different elements in a holistic way.

Expected Innovation

We aim to build a smart buffer system to be able to do not relay only to synchronism. Concerning the publish channel, we plan to send a "heartbeat" signal to the channel before the RabbitMQ API closes it.

7.1.5 QoS Evaluator

Existing solution

Currently, data application and service providers offload their Big Data Analytics workloads to the underlying infrastructures (e.g. cloud environments). There is an increasing pressure on them to provide QoS (Quality of Service) to their customers. Therefore, they need the underlying infrastructure to not only provide QoS but to also satisfy certain service-level objectives. These objectives take the form of high level agreements mostly focused on the uptime/availability of cloud (compute, storage and networking) resources (e.g., 90%, 99%, 99.99%). This is usually not enough to ensure more elaborated SLAs at the data application and service level, such as throughput or response time. This situation forces providers to lease an overprovisioned infrastructure, to ensure the expected performance of their applications and services.

QoS management is becoming a hot topic in the context of Big Data Analytics applications and services as we enter complex scenarios which require high levels of flexibility regarding the velocity, veracity, variety and volume of data to be processed.

BigDataStack approach

BigDataStack proposes to monitor, evaluate and ensure a QoS of performance attributes at different layers of the architecture: applications, data services (e.g., storage, processing) and infrastructure (e.g., networking, computing and block storage). The Triple Monitoring collects metrics from these three layers while the QoS Evaluator checks them individually or in combination to detect quality violations / depredations. This enables specification of objectives at application, data service and infrastructure levels, separately or in combination.

Results so far

The QoS Evaluator component has been deployed as a micro-service in a single Docker container and run on the BigDataStack testbed (specifically, in the same docker-enabled instance). The QoS Evaluator has been successfully integrated with the Prometheus-based monitoring system to query performance metrics and to provide QoS violation metrics in return (to be recorded and published as together with the original application, data service and infrastructure level metrics).

Expected Innovation

BigDataStack aims at bringing the automated evaluation of QoS for Big Data Analytics. The QoS Evaluator is expected to provide users with a finer-grained control on the performance of their data applications and services. By facilitating simultaneous evaluation of three different architecture levels (application services, data services and infrastructure), it will enable targeting of objectives at the three levels, separately or in combination. For example, if a data engineer is interested on achieving a certain objective (e.g., data flow per second), while the application engineer is focused on optimizing the use of resources from the application, the compromise between these two stakeholders with two different concerns can now be represented in the same agreement. This means giving rise to complex multi-objective scenarios.

Service level objectives might also be specified as a combination of metrics and setpoints, covering different levels of the architecture in the objective; in contrast to more traditional approaches where a single metric and setpoint is allowed. This enables scenarios where certain o objectives are tougher (or more relaxed) when certain aspects of the architecture are in a certain state.

Another expected innovation is the role both the QoS evaluation and the overall objectives play in the data-driven infrastructure management. While they are usually used to detect violations and compute the corresponding penalties, in BigDataStack infrastructure self-management, they are responsible for notifying any departure from the setpoint (objective) to the dynamic orchestrator, triggering runtime adaptation actions to get back to expected QoS.

7.1.6 Information-driven Networking mechanisms

BigDataStack approach

The Information-driven Networking mechanisms comprise a new component which will be implemented in the context of BigDataStack project. The Information-driven Networking mechanisms provide a set of network engineering methods combined with software defined networking technologies over containers and virtual machines for the enforcement of targeted policies according to the data (real-time, near real-time and offline) and application needs. It supports a set of mechanisms operating at services layer to understand the virtual hosts, URLs and other HTTP headers and at network layer to understand the workloads in storage services, DNS and a plethora of other services that do not use HTTP.

Results so far

Although the task associated with the Information-driven Networking functionalities starts at M13, a thorough review of Software Defined Networking (SDN) mechanisms has been made.

Expected Innovation

The expected innovation refers to a novel framework which enforces network policies according to the requirements tailored to the data or application needs and the information traveling in the BigDataStack cloud environment. The policies are enforced in real-time (within ms), as workloads are created / moved / destroyed, labels are applied / updated / deleted, and therefore policies are created / updated / deleted accordingly.

7.2 Innovation within Data-as-a-Service

Table 2 presents the monitored components of the Data as a Service main block of BigDAtaAStack.

Component	**Related priority** (*see Section 4)*	**Category** *(Technology / Research / Business)*	**Possible risk**
Federation of relational data store with object store	- Data Management	- Technology	- Increased response time, need for increased consumption of memory resources
Data Skipping	- Data Management	- Research	- Data owner does not want to create the plugins that permits to handle its User Defined Functions
Data Layout Manager	- Data Management	- Research	- Data flow is not strong enough. In order to get enough data to lay it out efficiently we would need to retain data (before uploading it to Object Storage) for too long (given the

			need to give enable querying fresh data at Object Store)
Elastic Distributed Storage	- Data Management	- Technology	- Downtime or performance overhead during scalability action
Cache layer for object store	- Data Management	- Technology	- An additional SSD caching layer is needed
Predictive and Process Analytics	- Data Analytics	- Research	No risk
Domain Agnostic Cleaning	- Data Processing Architectures	- Research	- Performance cost
Distributed CEP processing in WAN	- Data Management	- Technology	- Increased response time

Table 2 – the monitored components of Data-as-a-Service

7.2.1 Federation of relational data store with object store

Existing solution

Enterprises today often have to use different database systems to fulfil different purposes: relational operational databases for handling operational load, data warehouses to submit analytical queries, key value stores to hold historical data and perform big data analysis. Combining data coming from different data sources is not an easy task, while moving data from one source to the other (i.e. operational data from a relational database to a data warehouse) is cost-demanding and requires ETLs and offline batch processing, which are often performed at night. Existing solutions today for polyglot applications often introduces the concept of data lakes or implements a federation on top of different sources in order to provide an easy way for common access. However, all these solutions are referring to different data sources, while for federation they are using technologies like Spark, which can be very resource consuming and cannot exploit the specific capabilities of each different data store.

BigDataStack approach

The seamless analytical framework will federate data coming from two different types of datastores: the relational operational LeanXcale datastore and object stores (e.g., IBM object store (COS) or open source minio), which will share the same dataset. As a result, this single component will be used within the different datastores, in order to exploit the unique characteristics of each one and this transparently from the user, without having to compromise some requirements for the benefit of others. LeanXcale database will be used for operational workload, storing data that is being continuously ingested, and when the data can be considered as outdated and historical, thus no longer needing to participate in operational transactions, they will be safely removed from LXS and moved to the object store for heavy analytical processing. LeanXcale Query Engine will be used as the federator on top of the two datastores: Offering a common JDBC interface it will provide a common way to access data and will push down operations accordingly: write operations will be pushed down to the database, as they will require transactional semantics. Read operations will be pushed down both to the LeanXcale database and to the IBM object store by the federator, which will

retrieve the intermediate results and merge them on the fly before sending back the accumulated response. The LeanXcale query engine offers intra-query parallelism in order to exploit the distributed nature of its storage and the ability of the latter to accept pushdowns of aggregated operations, execute them locally and return back the intermediate results corresponding to that data node. The same concept will be applied here. The submitted query will be executed in a distributed manner, and the results will be returned to the caller.
Moreover, as already mentioned, the relational datastore and the object store will share the same dataset: Data will be periodically moved from one datastore to the other. The ability of LeanXcale database to periodically export a dump of all write operations, while ensuring data consistency allows for the seamless framework to use a queue between the two datastores where LXS will push these dumps in a predefined format, and IBM will pull this information and finally ingest the historical data to the object store. When data are received from the latter, only then the relational datastore can drop them, so that data loss can be prevented. In this scenario, it might appear the case where data co-exist in both stores and have to be taken into account when the federator produces the final results.

Results so far

At the moment, the initial design of the seamless analytical framework has been performed, analysing the potential bottlenecks, problems due to the distributed nature of the system, race conditions when moving a data set across different data stores that they do not share the knowledge on when data are stored or accessed, and grammar issues when having a common language to be executed by different systems. Therefore, the design is based on well-known standards (i.e. JDBC) and tools that can support this at either side. Moreover, an investigation of the capabilities offered by the JDBC connectivity on top of Spark is taking place in order to investigate which operations can be directly pushdown to Spark by the federator in order to improve performance by executing operations locally and reduce the amount memory consumptions on the federator level, in cases it needs to merge intermediate results that have not yet been analysed/filtered out. Finally, an initial design of the data pipeline between the two datastores has been conducted.

Expected Innovation

The main innovation of the seamless analytical framework is that using the same dataset, it allows to be used via two different datastores, each one of those providing different capabilities. Since analytical operations compete with operational ones in transactional datastores, they cannot support both types of loads at the same time. Due to this, they move data that can be considered historical to data warehouses, using costly demanding ETLs during night, splitting the dataset into two different ones that are being examined separately. However, the seamless analytical framework, transparently, allows treating the same dataset as a whole, by live transmitting data from the operational one to the object store, ensuring the consistency of the data, on the runtime, without any interference with the system administrator, and without the need for costly and demanding operations. On the same time, the access is also being done transparently, where the federator, using the commonly used JDBC standard forwards requests to each datastore accordingly, parallelize the operation and return the results. From the application level of view, the seamless framework can be considered as a black box database that combines the features and special characteristics of two different ones, without compromising the expected performance that each of the two stores could have been achieved on their own.

7.2.2 Data Skipping

Existing solution

Data Skipping is a well-known technique which consists at creating data skipping indexes on datasets residing in the Object Storage, and subsequently exploiting these indexes to skip over objects irrelevant to a given SQL query. One of the innovations of BigDataStack is to do significantly extend Data Skipping by:

1. Allowing users to define new data skipping indexes, without changing the core data skipping library
2. Supporting data skipping for arbitrary query predicates including UDFs and AND/OR/NOT.

BigDataStack approach

The Data Skipping module supports arbitrary query predicates, including UDFs (user defined functions) and AND/OR/NOT. This is particularly challenging, because UDFs can be arbitrary functions about which the Spark optimizer knows very little. This work will be applied to geospatial UDFs, which are important for the ship management use case.

Results so far

The Data Skipping code now supports a truly pluggable architecture, where new data skipping index types can be added without changing the core data skipping library. For example, we added a new geospatial data skipping index type using this capability, which works well in conjunction with geospatial UDFs. External users can also exploit this capability.

Expected Innovation

These data skipping techniques are novel in the industry, in addition envisioned synergy with data layout technology may lead to further advances in big data analytics.

7.2.3 Data Layout Manager

Existing solution

The purpose of the Data Layout Manager component is to organize/partition the rows of a dataset as objects in the Object Storage in such a way that improves analytics performance, by reducing the number of bytes sent from the Object Storage to an analytics framework, such as the Apache Spark (the main targeted KPI). In the Data Layout module implementation, the user specifies explicit commands to control the data layout, by specifying the columns to use for layout and their relative priorities. The innovation that BigDataStack will bring aims to include collecting query history and data statistics for specific use cases, and based on this, to automatically recommend a data layout and associated choice of data skipping indexes for that use case.

BigDataStack approach

Until now data layout was performed on datasets already existing in Object Storage: this is “Offline Data Layout”. BigDataStack plans to innovate and permit the data layout to be performed dynamically while data are ingested into Object Storage: this is “Online Data Layout”.

There are two main flows in which Data Skipping and Data Layout participate. In the ingestion flow, Online/Offline Data Layout is used to organize the dataset rows into objects. Moreover, the Data Skipping module is used to create data skipping indexes which can be used by subsequent analytics. The parameters such as which columns to use for layout and skipping are currently provided by the user. In future, there will be a query logging component which records the query history, as well as a data logging component which records dataset properties and their change over time as new data are ingested. This information can then be used to recommend the above parameters.

Results so far

The Data Layout Manager has undergone a complete refactoring. It has also been updated to handle string data types as well as numeric data types. String data types behave differently from numeric data types since in most cases they are queried using '=' or 'IN' predicates and not using inequality predicates. Therefore, our k-d-tree approach which splits the data according to the median value may not be ideal in this case. The k-d-tree approach was extended to support arbitrary cut nodes with unlimited number of children which enables each cut node to implement its own logic to decide how to split the data. We are experimenting with various alternatives of cut nodes, some of which take into account the number of distinct values in a column, and the number of appearances of each value. A key challenge is to do this efficiently. We also support sampling the input dataset when running the Data Layout Manager, which is important to achieve good performance.

Expected Innovation

The above-mentioned dynamic data layout techniques are novel in the industry, in addition envisioned synergy with data skipping technology may lead to further advances in big data analytics.

7.2.4 Elastic Distributed Storage

Existing solution

Even if the ability for a storage system to dynamically adapt to diverse workloads by scaling horizontally the resources needed has been already implemented, scaling in/out a database introduces additional challenges: fragment a dataset to smaller portions and move these portions to different nodes for load balancing. Typically, NoSQL database systems can scale in/out sufficiently and move their regions across the available nodes, but they compromise the consistency of the data, that they never promised to offer. On the contrary, fragmenting tables of a relational data store and move the corresponding regions across the nodes will introduce significant concerns regarding data consistency, under OLTP workloads. Due to this, transactional datastores either never support elasticity, or when they do support, they suffer from long periods of downtime or significant decrease of their performance, and as a result, they cannot be considered as elastic, as they cannot scale efficiently during the runtime.

BigDataStack approach

In the scope of the Adaptable Distributed Storage of the BigDataStack, a novel mechanism will be implemented that will allow the storage system of LeanXcale relational datastore to be provide elastic scalable capabilities, thus being adapted to diverse workloads on the run-time. This component will allow LeanXcale to partition its datasets to smaller portions (fragments) of data by splitting (or later merging them) and move those partitions effectively among the available data nodes, in order to achieve the balance of the load, both in terms of incoming workload, and in terms of the stored data load. In order to achieve this balance, this component will rely on the monitoring information that is being generated by the storage subsystem, and which provides useful insights regarding the resource consumption of each data region inside a node. Given that, a novel algorithm will be implemented, that will solve the non-linear resource allocation problem, taking into account the multi-dimensional aspects of the problem to be solved: CPU, memory, storage, network consumptions. The proposed configurations of the algorithm and the ability of LeanXcale database to distribute its load on the runtime with no downtime or decrease of its performance, will allow to dynamically adapt and reconfigure the data regions to deal with increased workloads. Finally, based on that, scaling in/out the data nodes of the database will transparently trigger the reconfiguration process, making this component truly elastic.

Results so far

At this phase, all necessary tools to enable elasticity have been implemented: splitting/joining data to different regions and moving these regions to different nodes. Experimentation has been conducted, validating the initial hypothesis that a dynamic reconfiguration of a dataset would not significantly affect the overall performance, while on the same time, the consistency of the data under operational workloads can be ensured, without having to stop all, or part of the system, which would lead to a downtime. Moreover, a first version of the resource allocation algorithm has been implemented and it is being validating while integrating with the component. Finally, the tools to allow a dynamic deployment of part of LXS components has been already provided, which is a necessary step and definitely a crucial requirement towards elasticity.

Expected Innovation

The innovation lies on the fact that this component is not a simple write-once, read-only storage system, where data are being ingesting once and it is being used only for analytical processing, where scaling the available resources can be achieved by simply adding/removing additional nodes. Even in these systems, elasticity is not an easy process, as adding new nodes might require an internal redistribution of the dataset which in turns might trigger the re-creation of the indexing in order to support big data analytics. However, when it comes to relational transactional datastores, things are even more complicated: Splitting and moving data regions of a datatable, that is being referenced by others under a foreign key relationship might not be possible if concurrent update transactions frequently occur on that table, otherwise, data consistency cannot be ensured. Due to this, database systems either do not allow this reconfiguration of the tables (and as a fact, they cannot scale) or they block all transactions that affect these tables, until the whole process of scaling and moving data is finished (which leads to downtime or significant decrease of the performance, thus cannot be considered elastic). The Adaptable Distributed Storage of BigDataStack however, relies on the LeanXcale storage subsystem that allows the dynamic reconfiguration during runtime, without downtimes or increased overheads, thus being truly elastic.

7.2.5 Cache layer for object store

Existing solution

In order to perform analytics efficiently on object storage, a client-side caching/acceleration layer is needed. This is critical for a hybrid cloud scenario, where some of the customer data are on premise (potentially LXS and Spark) and some is in the cloud (potentially IBM COS). In such a scenario, when performing analytics, data needs to move from COS to Spark across the WAN, therefore minimizing the amount of data movement when part of the data is retrieved multiple times is of utmost importance.
A similar scenario involves multi-cloud, where a dataset may be distributed among more than one cloud, also requiring data transfer across the WAN for the purposes of analytics.
The caching capability complements data skipping and data layout techniques to further reduce the KPI measuring the number of bytes sent from object storage to Spark.

BigDataStack approach

The COS Acceleration Layer is a client-side gateway to IBM COS, based on flash storage. Applications access this layer in the same way they would access COS itself, via an S3 API. When an application retrieves data, the layer is responsible for reading data from a remote instance of COS and caching the data locally in flash storage, while also returning the data to the application. The layer evicts data from the cache according to some cache replacement algorithm. The following figure depicts the proposed architecture of this feature.

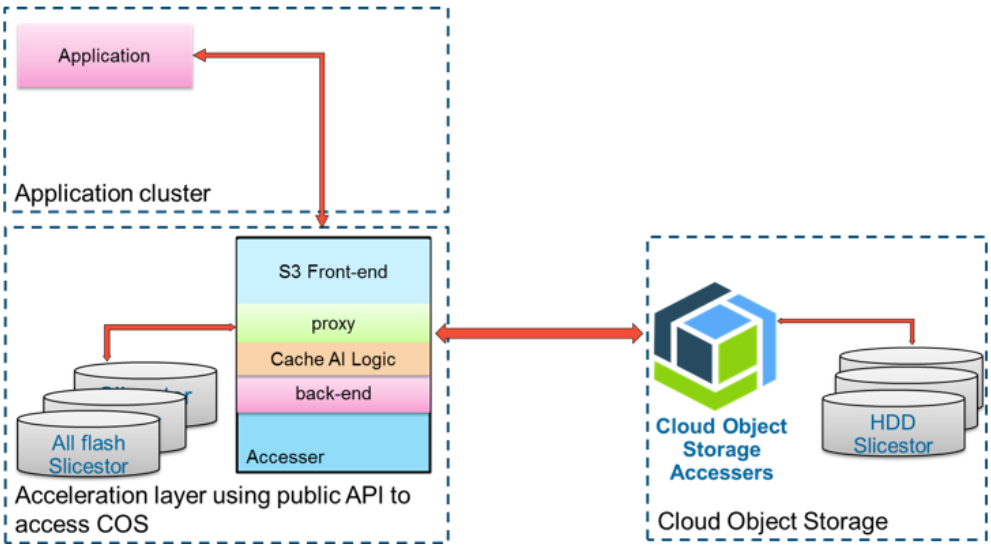

Figure 1 – Proposed architecture of the Cache layer for object store

The API to access this layer is at the object storage level, therefore this layer is general purpose and all applications which require object storage can be supported. However, in the case of BigDataStack, the data stored in Object Storage will typically be rectangular data, and the workload will typically be SQL queries over that data. For this scenario, more information is known about the workload which can potentially provide hints to the caching layer to prefetch data which is about to be read, retain in cache data which is likely to be read again in the near future, or discard data from cache which is unlikely to be read in the near future.

This feature complements the data skipping and data layout technologies described in a separate innovation block of Data-as-a-Service. Moreover, we expect high synergy between these topics.

Results so far

This work is intended for Y2 and Y3.

Expected Innovation

This cache layer is novel in the industry, in addition envisioned synergy with data skipping technology may lead to further advances in big data analytics.

7.2.6 Predictive and Process Analytics

Existing solution

In Predictive and Process Analytics the goal is, given an event log, to recommend to the user the next step in a process model, for example, a less costly (option defined by the user) route in a process flow, to perform process analysis, given an event log and a process, answering the question how well the process model describes the data etc. There are many tools at this moment that offer, individually, these services. The ProM framework [7] is such a framework, through its many plug-ins a user can perform many analytics on event driven data.

BigDataStack approach

In the scope of the BigDataStack project, the Predictive and Process Analytics component aims at proposing a global solution, to many of the needs recognized, for process discovery and enhancement.

Results so far

Until now, many algorithms have been tested-researched for use in this component and the strengths of each have been documented. At this point, a collection method is being put into place to collect and catalogue results of the executions of a few of the algorithms researched. In a later phase these metrics collected will be used to derive important results concerning the relationship between steps in a process model. This information will be used to propose a process flow with logical and practical solutions, for the user.

Expected Innovation

The innovation of this component lies at the architectural level. We propose a multipurpose, multifunctional solution for the transformation, execution and final recommendation to the user. Using multiple process algorithms, focusing on each one’s strengths, on the same dataset we strive to create a homogeneous solution.

7.2.7 Domain Agnostic Cleaning

Existing solution

The data analysis pipeline passes through several distinct phases, with each of them posing different challenges. In contrast to the much more researched modelling phase, the cleaning step is often seen as a sore point, even though without it the modelling phase is of little use. For example, consider the case of vessels maintenance data, where a central processing unit collects information from many sensors. If the data are too noisy, or even flawed because of a damaged source, the predictions might be associated with a high level of uncertainty or even reduced to garbage. In such cases, domain experts either manually check the incoming signal from every sensor or systemically clean the arriving data using predefined criteria.

BigDataStack approach

A new, data cleaning mechanism will be developed to enable domain-agnostic cleaning and harmonisation of data. The latter is proposed as an approach that results to a mechanism being "generalised" and applicable to different application domains and as a result to different datasets. To take this one step further, by monitoring the frequency of the errors that arrive from a specific source, we can identify possible deterioration in its performance.

Results so far

Our view on the problem takes advantage of recent developments in artificial neural networks (ANN) and deep learning (DL), to extract latent features that correlate different fields, and identify possible defects in their measurements. Furthermore, by harnessing the power of Machine Learning and Deep Learning we can automate the process of Data Harmonization, an approach to data quality that improves the condition and utilization of the data. Data Harmonization is about creating a single source of truth. It works by absorbing diverse data from various sources, brushing away any inconsistencies, purifying it and presenting it as a whole. Artificial Intelligence and Machine Learning can simplify and automate this operation, thereby speeding up the process of data modelling.
On the first version of the data cleaning module we focus on a schema inference mechanism. A simple flow consists of reading dataset files (e.g., csv files) from object store, infer the schema and store it back to the database. On serving time, the system retrieves that schema, compares it to a new dataset and prints out an anomaly report, which stores again to the database. Every file that is produced during this process should be available to the user, through explicitly developed APIs, or through the datastore.
With this mechanism we are able to compare dataset schema's, detect anomalies, such as missing values, erroneous data types, etc., and also detect skew between training datasets and drift between training and serving datasets.

Expected Innovation

The main innovation in this component has to do with the keyword "domain-agnostic". There should be no prior knowledge about the dataset and the mechanism should not work like an expert system, but more like a probabilistic system.
More concretely, in our approach we forward the data through a deep ANN, to construct a distributed representation of each concept, e.g. temperature, blood pressure or oxygen saturation level, which is a much more comprehensible notion for the machine. This process helps us gain knowledge about the connections between fields and diagnose irregularities in the input data.
To take this one step further, by monitoring the frequency of the errors that arrive from a specific source, we can identify possible deterioration in its performance. For instance, in our medical example, if the sensor that is responsible for taking the temperature of a patient gives flawed measurements regularly, we can assume that this sensor may be broken, and explicitly run a set of diagnostics, to calibrate or replace it.

7.2.8 Distributed CEP processing in WAN

Existing solution

Data streaming engines and Complex Event Processing (CEP) have become pervasive a data management technology during the last years. There is an increasing number of events that are continuously produced and must be analysed on the fly before they are stored or even without being stored. IoT records, gathered by what we call the “ship management on board application” (Real-time Ship Management use-case), monitoring of infrastructures (e.g. cars in highways), network messages, stock prices, etc are examples of continuous streams of data that are analysed as they are produced in order to find patterns or check conditions. Many data are produced at remote locations (monitoring patients at home, sensors geo-distributed) and must be transferred to a central location for being processed. Several open source and commercial solutions are available that either are centralized (running on a single node) or distributed (running on several nodes on cluster) however, these systems are not designed to run on a geo-distributed system. So, they process the data far away from where the data are produced therefore, analysing the data with a large delay.

BigDataStack approach

The CEP will target the deployment of queries close to the location where the data are produced in order to improve the latency of queries and avoiding expensive data transfers. The CEP will take into account the location of the data and analyse the queries in order to produce a deployment of the queries as close as possible to the data source. This information will be permanently monitored in order to change reconfigure dynamically the deployment if the conditions change (rate data are produced, bandwidth, resources available…) a more efficient configuration can be deployed.

Results so far

At the time being, the requirement from the use cases have been gathered. This information has been used for the design of the CEP. These requirements indicate that part of the computation must be done close to the data source (for instance on board in the vessels). While some data must be sent to the data centre for further analysis and correlation with stored data. The CEP can also be used to prepare the data for being analysed as soon as it is produced so that, the analytics are done over recent data and incrementally, not over data from previous days. A first prototype is available and some of the use case queries have been programmed.

Expected Innovation

The main innovation of the CEP is the design of a CEP system able to process data in a geo-distributed environment. The query compiler of the CEP will take into account the location where the data comes from, network latency, bandwidth and available resources in order to produce an efficient query plan. The CEP will also monitor the performance of the system and the resources in order to dynamically adapt the deployment to changing conditions.

7.3 Innovation within Dimensioning, Modelling & Interaction

Table 3 presents the monitored components of the Dimensioning, Modelling and Interaction Services block of BigDataStack.

Component	Related priority *(see Section 4)*	Category *(Technology / Research / Business)*	Possible risk
Process Mapping	- Data Analytics	- Research	- Sub-optimal mapping of process step to a ML algorithm
Data Toolkit	- Data Processing Architectures - Data Analytics - Engineering and DevOps for Big Data	- Technology - Research	No risk
Application Benchmarking and Dimensioning facility (including load injection and potentially DoE aspects)	- Engineering and DevOps for Big Data	- Technology	- Costs and time needed for the benchmarking process

Table 3 – the monitored components of Dimensioning, Modelling and Interaction Services

7.3.1 Process Mapping

Existing solution

In the context of BigDataStack, Process Mapping refers to the correspondence between a process step and the underlying algorithm/implementation. Typically, Process Mapping is necessary in order to translate a Process Model or a Workflow into executable software that fulfils the functionality specified in the model. For certain tasks, this mapping is straightforward to obtain. However, in machine learning (ML) tasks, the correspondence between a process step and the underlying ML algorithm may be obscure, since this depends on various issues, including the statistical properties of the input data set, the requirements of the ML algorithm, potential constraints on performance, and so on. Current practice shows that the task of selecting an appropriate ML algorithm for a specific task is typically done empirically, based on the knowledge of ML experts. However, this is still largely performed empirically, and furthermore ML tasks are increasingly needed by non-expert users, outside the domain of computer science (e.g., researchers in physical sciences, health, etc.). To bridge this gap, one solution is to employ algorithm selection techniques, where the aim is to build a system that is able to suggest the most suitable algorithm for a given task from a pool of available algorithms. Although this issue has been explored in the past, there is still a lack of a principled methodology and corresponding tools for automatic algorithm selection for ML tasks over Big Data.

BigDataStack approach

In BigDataStack, Process Mapping is going to explore the automatic selection of an algorithm for the execution of a specific part of a process, focusing mainly on processes corresponding to ML tasks. Hence, the key functionality targeted by the Process Mapping component is stated as follows: Given a machine learning task, a data set, and a set of available ML algorithms that can handle the given task, select (or recommend) the subset of ML algorithms with best performance. Essentially, the problem can be cast as a search problem, where the search space consists of the available ML algorithms, and the objective is to identify the best performing algorithms. In BigDataStack, the available ML algorithms that are going to be used can be both: (a) algorithms available in a library for machine learning over Big Data (e.g., Spark's MLib), and (b) custom implementations of ML algorithms that provide specific functionality and can be classified under specific ML tasks (e.g., a new prediction algorithm).

Results so far

At the time of this writing, the Process Mapping component has been designed and is under development. One dimension of particular interest is related to the meta-features that can be extracted from data sets and will enable the subsequent comparison between any two given data sets. The in-depth study of which subset of meta-features is appropriate for the correct identification of similarities between data sets, in terms of obtaining similar performance by ML algorithms, is the main focus of the current work.

Expected Innovation

The Process Mapping component is going to perform algorithm selection for Machine Learning tasks, following a meta-learning approach and exploiting meta-knowledge extracted and maintained from previous executions of ML algorithms on other data sets. It is expected that the Process Mapping component will be additionally provided as a stand-alone tool that can facilitate the non-expert user to perform ML tasks, without exhaustive exploration of all potential algorithms, thus saving significant time and effort. Even though the topic of algorithm selection for ML tasks has been investigated in the past, there is still a lack of easy-to-use tools that have been rigorously tested and ultimately provide solutions, even for specific categories of ML tasks (e.g., classification, clustering, etc.). Consequently, the expected innovation is in the form of a tool that includes novel techniques for ML algorithm selection, with a particular focus on Big Data.

7.3.2 Data Toolkit

BigDataStack approach

The Data Toolkit is a new component which will be implemented in the context of BigDataStack project. The Data Toolkit is responsible to create a common serialization format and executable Big Data workflows in the form of valid analytics pipelines. It supports Spark and Spark MLlib but it may be also extended to support other machine learning frameworks (indicatively TensorFlow, scikit-learn, etc.).

Results so far

As the task associated with the Data Toolkit functionalities starts at M13, we have just finished a detailed literature review to address the challenges arising from synchronizing Big Data analytics pipelines, signalling and orchestrating end-to-end Big Data applications in a seamless manner. Therefore, no results are currently available as the Data Toolkit is in its initial development phase.

Expected innovation

The expected innovation w.r.t. the Data Toolkit is to build a robust workflow engine for Big Data analytics with the ambition to be agnostic, extensible as the end users can inject their own analytics and decoupled from specific processing frameworks. The potential innovation outcome lies in the field of business analysts' and data scientists' empowerment to mix and match machine learning technologies in a unified runtime environment.

7.3.3 Application Benchmarking and Dimensioning facility

Existing solution

Existing solutions focus on benchmarking of Cloud services but in the context of elementary workloads that are typically not changing in the course of time. This is on one hand needed for comparability of the results, however they present limits in the sense of examined workloads and service setups, limited e.g. in the type of resources in which the application is run.

BigDataStack approach

With the BigDataStack benchmarking and dimensioning workbench, it is expected that the process of stress testing will be significantly faster and easier to manage, enabling the adaptation per case and the examination of aspects that are more suitable for big data issues, as well as incorporation of specific deployment and configuration options of the examined data services. Furthermore, the incorporation of a number of potential deployment candidates may enable the existence of more options for the end user in order to trade between QoS and cost.

Results so far

At the moment, design of the solution has been performed in order to cover for the generic configuration and execution setup, aiming at abstracting from the specifics of each case, by investigating aspects of generalization on one hand but also adaptation in data service type. Therefore, the design aims at having a generic infrastructure that is able to be parameterized per examined case with the minimal needed human intervention. Furthermore, investigation of the parameters of the main Big Data Stack services has been performed in order to indicate the search space in which workload investigation should be performed.

Expected Innovation

The innovation in this case lies on the fact that different service setups and configuration parameters of specific Big Data technologies solutions will be investigated in order to better depict the effect of these features to various workload aspects. Furthermore, this type of testing can be extended to private cloud deployments and have an a priori understanding of the infrastructure sizing that is needed in order to achieve specific QoS levels. This may in turn lead to optimized offerings and adapted deployments to individualized applications of the Big Data technologies included in the project.

7.4 Innovation in BigDataStack Use Cases

7.4.1 Real-time ship management

Use Case Purpose

Suppose that a vessel has to complete its route within a given period of time. When a part of the main engine fails unexpectedly, the ship risks staying off-hire. This can be very damaging to a shipping company. On one hand chartering revenues decrease. On the other hand, replacing a spare part immediately increases cost. Thus, identification of potential failure allows timely ordering, or even replacement of spare parts before failure.
The main engine, posing the highest risk, consists of various spare parts depending on many parameters. Thus, it is difficult to accurately predict failures. If false alarms occur, the operating costs increase, as ordering of unnecessary parts is not optimal.
The multivariate nature of a supply department makes the selection of the port, where the spare part will be delivered, challenging. The price depends on the port, the time frame of order, and the personnel replacing the part.
This use case focuses on:

- Monitoring the main engine of a vessel;
- Identification of malfunction patterns and notification of the technical and the supply department;
- Automatic ordering of the appropriate spare part to be delivered at a port on route (upon confirmation);
- Minimization of overall maintenance cost;
- Avoidance of off-hire seasons due to machinery failures and unexpected but compulsory maintenance.

BigDataStack provides an adequate architecture for big data management, thus enhancing the use of analytics and methods for scheduling of orders, preventive maintenance, visualization of the current state and final results. By incorporating these aspects through the DANAOS platform, BigDataStack allows to shipping companies to cherish their data and use them in difficult decision making processes, such as the preventive maintenance of the machinery, the supply management of a fleet and its implications on the dynamic routing of a vessel.

Expected Innovation

The expected innovation of the proposed use case is trifold. In terms of monitoring, we propose an improved approach that utilizes CEP to alert the user when an anomaly occurs on the main engine of a vessel, while also informs the user on the data quality acquired by sensors on-board. Additionally, the analytics performed over these data inform the user about unexpected malfunctions. In terms of operations optimization, this minimizes machinery failures that cause the ship to go off-hire, reduces operating costs by optimizing the requisition process of new spare parts, thus better organizing the supply department. Finally, in terms of big data management the BigDataStack platform itself is a product that offers to a shipping company to deploy big data applications and cherish their assets-data.

The above could be depicted on a set of KPIs. Indicatively,

- Increased data variety in analytics;
- Increased cross-stream events processed in real-time by CEP;

- Increased accuracy of predictions by considering additional datasets;
- Increased efficiency of processes due to modelling and mining;
- Reduced maintenance and idle times due to predictive maintenance;
- Reduced fuel consumption due to dynamic routing;
- Reduced CO_2 emissions due to dynamic routing;
- Increased service availability due to overall maintenance process optimization.

Current Progress

So far, Danaos has performed the following actions:

- Had a F2F with the head of the technical department to define our needs with respect to the malfunctions to be inspected;
- Focused on a single type of damages that happens gradually and so far is detectable only by on-sight inspection;
- Organized 2 rounds of F2F interviews with 2 fleet managers to gain detailed information about this type of damages;
- Collected maintenance data;
- Gained access to data for 10 vessels;
- Submitted use case description;
- Submitted data description;
- Submitted a data sample for 1 vessel;
- Meeting with UPRC, UBI and ATC to define the Business Analyst and Data Analyst views and walk these views through the architecture as a concrete example;
- Currently supporting CEP and analytics teams through the problem and data.

The dataset that DANAOS exported is a 5-years dataset for 10 vessels based on:

- operational data, i.e., telegrams (38K records);
- general purpose sensor data (21 different sensors, 29M records);
- main engine sensor data (100 different sensors, 29M records);
- reports for a specific damage that the analytics algorithms will tackle (61 records, i.e. 61 incidents).

The total size of this dataset is 122GB, excluding meteorological and oceanographic data, which will be joined in the framework of the project.

7.4.2 Connected Consumer

Use Case Purpose

Globalization along with the development of new technologies has allowed consumers to have instant access to information and products that were unreachable in the past. In this scenario, new global companies have arisen and competitors in the retail industry have changed from local to global competitors. Keeping customer loyalty has also become more difficult since the products offer has become global. Thus, being able to give personalized and accurate recommendations to customers becomes a must for every retailer if they want to keep on track.

In this context, being able to recommend to the customers those products they need (i.e. to anticipate their purchase) can be very interesting for retailers. And this is exactly what our use case is focused on. We are collaborating with one of our retail customers in building a recommender system for the grocery market. Concretely, the recommender system is expected to generate the following items:

- Recommendations of products to customers based on the historic of their own purchases and the product characteristics (predictive shopping list);
- Recommendations of products to customers based on what other customers bought (cross-selling);
- Promotions that best suits to customers based on the products that are expected to be purchased by the customer in the next act of purchase

On the one hand, our use-case is expecting to provide the history of 4 years of purchases (around 4 TBytes of information). This information will allow us to test the capabilities of BigDataStack with real big data.
On the other hand, BigDataStack will provide us a suitable infrastructure to test our system. In this way, BigDataStack has components which are able to make decisions based on data – e.g. it has been designed to adapt its infrastructure based on both current data and its own historical data. Furthermore, BigDataStack offers a set of analytics and tools that will allow the business expert and the data scientist to easily carry out their work by defining their business processes and injecting their algorithms in a friendly and natural way.

Expected Innovation

The recommender will use tailored machine learning algorithms to recommend customers what they need in their next purchase. On the one hand, analytics of the use case will take into account the historic of sales and the behaviour towards the recommendations made to them in the past. On the other hand, they will also take into account the items that other customers bought. Furthermore, the recommender is also planned to recommend those promotions that best suit customers. This decision will be taken based on what the customer is expected to buy. This is a plus from the classic ecommerce recommenders which are mainly providing cross-selling recommendations based on what other users bought together (e.g. chip potatoes goes together with beer). The effectiveness of the recommender could be measured with the following KPI's:

- Increase of customer loyalty;
- Increase of sales;
- Increase of marketing campaigns ROI;
- Increase of customer satisfaction;
- Increase of the accuracy of the predictions.

Current Progress

So far ATOS WL has performed the following actions:

- 3 conferences have been held with our partner in the project Eroski to define the requirements of the use case. A requirements analysis document that describes the business requirements of the recommender has been generated;
- Submitted use case description and requirements analysis;
- Submitted data description;
- Collect real data from one week of purchases to start working in the components;
- Submitted data from one week of purchases;

- Focused on creating a first prototype to recommend products on what other customers bought.

Currently, we are supporting CEP for the pre-processing of the data and working in the analytics.

7.4.3 Smart Insurance

Use Case Purpose

GFT use case focuses on the development of an IT solution for Insurance companies, developing software and systems addressing the needs of such institutions. A data-centric paradigm will be followed, addressing the provision of services according to the customer “tailored” requirements.
The main goal is allowing insurance companies to better develop the customer management, by providing personalized services to the customer, as well as new corporate services for the customers’ profitability.
The key aspects in this case related to data analytics are:

- **Customer segmentation**: customers’ segmentation according to their financial sophistication, age, location, etc. Thus, all the customers are classified into groups by spotting coincidences in their attitude, preferences, behavior, or personal information. This grouping allows developing attitude and solutions especially relevant for the particular customers. As a result, target **cross-selling (**recommendations of products to customers based on what other customers bought**)** and **upselling** (recommendations of more advanced products to customers based on what they have bought) policies may be developed and personal services may be tailored for each particular segment (such as lower priced premiums)
- **Lifetime value prediction**: Customers lifetime value (CLV) is a complex phenomenon representing the value of a customer to a company in the form of the difference between the revenues gained and the expenses made projected into the entire future relationship with a customer. Prediction of the CLV is typically assessed via customer behavior data in order to predict the customer’s profitability for the insurer. Thus, the behavior-based models are widely applied to forecast the customer **retention**. This allows forecasting the likelihood of the customers’ **churn** (identify which customers are likely to cancel contracts in the near future).

We are collaborating with HDI Assicurazioni, which is part of a large German insurance group, of international standing, the Talanx Group of Hannover[1] (born HDI group). Talanx is the third insurance company in Germany and operates in150 countries.
An initial set of data has been provided by HDI, which will allow us to test the components of BigDataStack in relation with big data, by exploiting a set of analytics and tools for the scenario processes.

Expected Innovation

The use case will exploit the BigDataStack optimized seamless analytics framework, enabling prediction of customers behaviour and intelligent decision making regarding personalized and tailored offerings. This will allow to increase HDI customers’ development, through ad-hoc

[1] http://www.talanx.com/

cross-selling and upselling intelligent strategies, as well as have a better control of customers' profitability and churn detection. A number of KPI's will be monitored, such as:

- Increase of sales
- Increase of customer loyalty
- Increase of customer satisfaction

Current Progress

In this initial phase, GFT has performed the following actions:

- Many meetings held with insurance company HDI (our partner in the project use case) to define the requirements of the use case. A requirements analysis has been generated;
- Submitted use case description and requirements analysis;
- Submitted data description;
- Started to design and develop a first prototype to provide tailored offerings on what other customers bought.

7.5 Scientific/Technical and Market Alignment

During the first eleven months, the project didn't attend many conferences or events, since we didn't have important outcomes to present. BigDataStack was represented by DANAOS in **Posidonia 2018.** DANAOS made a video presentation, had various face to face conversations with people over the objectives of the project and collected valuable feedback about the ship management use case.
Posidonia 2018 set a high benchmark for visitor numbers. Over 23,000 visitors crowded the stands and attended the many events in the four halls and conference rooms at the state-of-the-art Metropolitan Expo. The energy and excitement of the popular Posidonia Games kicked off five days of great business for the 2,009 exhibiting companies from over 98 countries and territories.
Over nearly 50 years the international shipping community has gathered at Posidonia, the home of Greek shipping, welcomed in 2018 by the owners of a fleet of some 4,400 ships, including an orderbook of $22bn. Greek owners are also investing billions of dollars to ensure their fleet continues to lead the way in clean technologies and efficient operations. A multi-billion market for Posidonia's exhibitors.

8 Conclusions

This document presented the current progress of the BigDataStack project with regards to Innovation Potential. The initial plan and specific activities of Innovation Management, along with the innovation within the project are also presented.

The priorities defined by BDVA constitute the basis of our strategy. The preliminary overview of existing solutions for the different components affirms the importance of BigDataStack for both scientific community and market. BigDataStack provides a holistic, innovative solution for big data applications and operations that no product in the market is currently able to offer.

References

[1] Innovation Management - Part 1: Innovation Management System. Available Online: https://standards.cen.eu/dyn/www/f?p=204:110:0::::FSP_PROJECT,FSP_ORG_ID:35932,671850&cs=13A816A57184977C465944D2F2E2C5645

[2] Big Data Value Association, "Strategic Research and Innovation Agenda", http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf

[3] European Commission, "How to convert research into commercial success stories? Analysis of EU-funded research projects in the field of industrial technologies," 2013. https://ec.europa.eu/research/industrial_technologies/pdf/how-to-convert-research-into-commercial-story_en.pdf

[4] EU General Data Protection Regulation, http://www.eugdpr.org/

[5] SIIA White Paper, "Data-Driven Innovation – A Guide for Policymakers: Understanding and Enabling the Economic and Social Value of Data", 2013.

[6] Available Online: https://www.rabbitmq.com/

[7] http://www.promtools.org/