

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/

D6.1 – Use case description and implementation

Work Package	WP6 – Use case description and implementation – M18
Lead Author (Org)	Maurizio Megliola (GFT)
Contributing Author(s) (Org)	Stathis Plitsos, Konstantina Rousia (DANAOS), Bernat Quesada Navidad (ATOS Worldline)
Due Date	01.07.2019
Date	01.07.2019 (Re-submission: 04.10.2019)
Version	2.1

Dissemination Level

<input checked="" type="checkbox"/>	PU: Public (*on-line platform)
<input type="checkbox"/>	PP: Restricted to other programme participants (including the Commission)
<input type="checkbox"/>	RE: Restricted to a group specified by the consortium (including the Commission)
<input type="checkbox"/>	CO: Confidential, only for members of the consortium (including the Commission)

Versioning and contribution history

Version	Date	Author	Notes
0.1	12.10.2018	Maurizio Megliola (GFT)	ToC
0.2	06.05.2019	Maurizio Megliola (GFT)	Updated title of GFT use case
0.3	06.05.2019	Maurizio Megliola (GFT)	Updated Table of Contents
0.4	10.06.2019	Bernat Quesada Navidad (ATOS Worldline)	Chapter 4 added
0.5	14.06.2019	Konstantina Rousia (DANAOS)	Chapter 3 added
0.6	25.06.2019	Maurizio Megliola (GFT)	Executive Summary, Introduction added
0.7	26.06.2019	Maurizio Megliola (GFT)	Chapter 5 and Conclusion added
0.8	29.06.2019	Bernat Quesada Navidad (ATOS Worldline)	Chapter 4 updated
0.9	30.06.2019	Stathis Plitsos (DANAOS)	Chapter 3 updated
1.0	01.07.2019	Maurizio Megliola (GFT)	Final version
1.1	24.09.2019	Bernat Quesada Navidad (ATOS Worldline), Maurizio Megliola (GFT)	Chapters 4 and 5 updated respectively to address GDPR-related issues
2.0	30.09.2019	Dimosthenis Kyriazis	Candidate final version
2.1	04.10.2019	Maurizio Megliola (GFT)	Updated picture page 39

Disclaimer

This document contains information that is proprietary to the BigDataStack Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to a third party, in whole or parts, except with the prior consent of the BigDataStack Consortium.

Table of Contents

List of tables.....	4
List of figures	4
1. Executive Summary.....	5
2. Introduction.....	6
2.1. Relation to other deliverables.....	6
2.2. Document structure.....	6
3. Real-time Ship Management (RSM)	7
3.1. Overview and goal	7
3.2. Pilot description.....	9
3.3. Datasets	11
3.4. Use Case Scenarios	16
3.4.1. Activity 1: Acquire data	16
3.4.2. Activity 2: Select attributes.....	16
3.4.3. Activity 3: Monitor the selected attributes.....	16
3.4.4. Activity 5: Use an analytics algorithm or deploy a new one.....	17
3.4.5. Activity 6: Produce alerts based on real-time monitoring information	17
3.5. Next steps.....	18
3.5.1. Activity 7: Order spare parts when necessary	18
3.5.2. Activity 8: Re-route the vessel accordingly.....	18
4. Connected Consumer (CC)	19
4.1. Overview and goal	19
4.2. Pilot description.....	21
4.3. Datasets	22
4.4. Use Case Scenario	30
4.4.1. Activity 1: Definition of the analytics for the recommender	30
4.4.2. Activity 2: Deployment of the application services.....	31
4.4.3. Activity 3: Re-deployment of the application services	32
4.4.4. Activity 4: Visualize recommendations.....	33
4.4.5. Activity 5: Provide recommendations	33
4.4.6. Activity 6: Collect feedback.....	34
4.5. Next steps.....	34
5. Smart Insurance (SI)	36
5.1. Overview and goal	36
5.2. Pilot description.....	37
5.3. Datasets	39
5.4. Use Case Scenarios	46
5.4.1. Activity 1: Data acquisition.....	46
5.4.2. Activity 2: Analytics definition.....	46
5.4.3. Activity 3: Deployment of the application services.....	46
5.4.4. Activity 4: Display recommendations	47
5.4.5. Activity 5: Provide recommendations	47
5.5. Next steps.....	48

6. Conclusions.....	49
---------------------	----

List of tables

Table 1 – Real-time Ship Management Actors	7
Table 2 – Connected Consumer Actors	19
Table 3 – Smart Insurance Actors	36

List of figures

Figure 1 – RSM Pilot's architectural overview	9
Figure 2 – Web Services access layer	21
Figure 3 – Analytic flow	22
Figure 4 – CC Database structure.....	23
Figure 5 – Architecture Schema of the SI use case.....	39

1. Executive Summary

BigDataStack delivers a complete high-performant stack of technologies addressing the needs of data operations and applications. The BigDataStack project was conceived as a data centric platform, integrating approaches for Data as a Service. Approaches for data cleaning, data layout and efficient storage, combined with seamless data analytics will be realised holistically across multiple data stores and locations. BigDataStack holistic solution incorporates approaches for data-focused application analysis and dimensioning, and process modelling towards increased performance, agility and efficiency. A toolkit allowing the specification of analytics tasks in a declarative way, their integration in the data path, as well as an adaptive visualization environment, realize BigDataStack's vision of openness and extensibility. This deliverable includes the description of the three BigDataStack use cases together with their implementation on top of the above mentioned architecture components. The use cases are the following:

- Real-time Shipping Management (RSM)
- Connected Consumer (CC)
- Smart Insurance (SI).

The RSM use case utilizes the BigDataStack architecture for big data management (emphasis on the data as a service key offering), its 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 a difficult decision-making process, such as the supply management of a fleet.

The CC use case utilizes the BigDataStack environment to implement and provide a recommender system for the grocery market. All of the data that are used for training the analytic algorithms of the use case is corporate data provided by EROSKI, one of the top food retailers companies in Spain.

The SI use case will use BigDataStack to implement smart recommendation systems for the insurance market. The datasets that will be used within the process of the use case is corporate data provided by an insurance company, based in Italy, selected from the GFT customers' portfolio.

Thus, the current deliverable presents the description for the three BigDataStack use cases, the context, the goal, the datasets and the main scenarios implemented or next to be implemented. It should be noted that v2.0 of this deliverable has been released to include relevant GDPR-related information (updates in Sections 3.3, 4.3 and 5.3). An updated version of this report is scheduled for M34.

2. Introduction

2.1. Relation to other deliverables

The current deliverable, the first BigDataStack deliverable concerning **Use Cases** (D6.2 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)* and *D2.2 (State of the art and Requirements analysis - II)* identify and specify the technical requirements for BigDataStack both through use case (UC) providers and technology providers, while *D2.4 (Conceptual model and Reference architecture - I)* and *D2.5 (Conceptual model and Reference architecture - II)* provide information about the key functionalities of the overall architecture, the interactions between the main building blocks and their components, along with a first version of the internals of these components regarding research approaches to be realised during the course of the project. In addition, the technical deliverables from WP3, WP4, WP5 (respectively, D3.1 WP3 Scientific Report and Prototype description - Y1, D4.1 WP4 Scientific Report and Prototype description - Y1, D5.1 WP 5 Scientific Report and Prototype Description - Y1) are the deliverables which present the current technical status (dealing with **Data-driven Infrastructure Management**, **Data as a service** and **Dimensioning, Modelling and Interaction Services** respectively) of BigDataStack project.

2.2. Document structure

The document is structured as follows:

- Section 2 provides an introduction to the deliverable and a description of the document structure.
- Section 3 introduces and describes the Real-time Ship Management use case by DANAOS, providing a description and information about goals, datasets, scenarios and next steps.
- Section 4 introduces and describes the Connected consumer use case by ATOS WorldLine, providing a description and information about goals, datasets, scenarios and next steps.
- Section 5 introduces and describes the Smart Insurance use case by GFT, providing a description and information about goals, datasets, scenarios and next steps.
- Finally, in Section 6, conclusions are reported.

3. Real-time Ship Management (RSM)

This section provides a description of the Real-time Ship Management use case scenario from DANAOS.

3.1. Overview and goal

The Real-time Ship Management (RSM) use case exploits the BigDataStack environment with an emphasis on the data as a service offering for big data management, its analytics and methods for scheduling of orders, preventive maintenance, visualization of the current state and final results.

DANAOS platform is an integrated web-based service that combines data from sensors onboard, operational data from a shipping company's database and weather data in order to assist the technical and operations department of a shipping company to monitor its vessels. Its practical use is to provide an overview of a vessel's performance and alert the users when policy-based rules are violated. By incorporating these aspects through the DANAOS platform, BigDataStack allows shipping companies to utilize their data and use them in a difficult decision-making process, such as the monitoring of the vessel, the identification of potential failures and the supply management of a fleet.

What is the scenario's goal?

The scenario's goal is to:

- Monitor the main engine of a vessel.
- Identify malfunction patterns and notify accordingly the supply department.
- Automatically order the appropriate spare part to be delivered at a port on route (upon confirmation).
- Minimize the overall maintenance cost.
- Avoid off-hire seasons due to machinery failures and unexpected but compulsory maintenance.

What is the business objective?

The main business objectives are:

- Advanced monitoring of key components in the engine room and at office.
- Better organisation of the supply department.
- Minimization of machinery failures that cause the ship to go off-hire.
- Reduction of operating costs, by optimising the requisition process of new spare parts.

Who are the actors in the use case?

Table 1 – Real-time Ship Management Actors

Id	Name	Description
ROL-01	Data Owner	BigDataStack offers a unified Gateway to obtain both streaming and stored data from data owners and record them in its underlying storage infrastructure that supports SQL and NoSQL data stores.
ROL-02	Data Scientist	BigDataStack offers the Data Toolkit to enable data scientists both to easily ingest their analytics tasks and to

		specify their preferences and constraints to be exploited during the dimensioning phase regarding the data services that will be used (for example preferences for the data cleaning service).
ROL-03	Business Analysts	BigDataStack offers the Process Modelling Framework allowing business users to define their functionality -based business processes and optimize them based on the outcomes of process analytics that will be triggered by BigDataStack. Mapping to specific process analytics tasks will be performed in an automated way.
ROL-04	Application Engineers and Application Service Owners	BigDataStack offers the Application Dimensioning Workbench to enable application owners and engineers to experiment with their applications and dimension it in terms of its data needs and data-related properties.
ROL-05	Technical Department	BigDataStack offers the Data and Services Monitoring component to monitor the produced streams of data. Results of the analytics algorithm for preventive maintenance are alerts that the engineer of the Technical department wishes to further investigate through the respective data.
ROL-06	Operations Department	BigDataStack offers the Deployment Engine Component through which custom services can be deployed. In this case the deployed services allow to an employee of the Operations department to adjust accordingly the vessel's route, if a spare part is delivered to a specific port.
ROL-07	Supply Department	BigDataStack offers the Deployment Engine Component through which custom services can be deployed. In this case, an employee of the Supply department wishes to order a specific spare part, when a malfunction occurs.

What are the key activities these actors are involved in?

The RSM use case consists of the following main activities:

1. **Acquire data.** The Data Owner wishes to set a data source from which BigDataStack obtains data to store them in its infrastructure
2. **Select attributes.** The Data scientist wishes to select a set of attributes depending on custom criteria by utilizing the Data Toolkit of BigDataStack.
3. **Monitor the selected attributes.** The Data Scientist wishes to monitor the data through the Visualization Environment of BigDataStack.
4. **Deploy the application services.** The application engineers and application service owners wish to deploy their application services on BigDataStack.
5. **Use an analytics algorithm or inject a new one.** The Data Scientist wishes to use an algorithm from a list of existing algorithms or inject a new one into BigDataStack.
6. **Produce alerts depending on monitoring.** An engineer of the Technical department wishes to receive predictive maintenance alerts.

7. **Order spare parts when necessary.** An employee of the Supply Department wishes to receive a notification to order new spare parts.
8. **Re-route the vessel accordingly.** An employee from the Operations department wishes to know and confirm the adjusted route once a spare part order is placed from the Supply department.

3.2. Pilot description

A vessel has to complete its route within a specific time-frame. When a part of the main engine fails unexpectedly, the ship risks staying off-hire. This can be very damaging to a shipping company, as chartering revenues decrease, while replacing a spare part immediately increases cost. Thus, identification of potential failure allows timely ordering, or even replacement of spare parts before failure. Furthermore, we wish to identify malfunctions on sensors that cause data-loss. Last, we wish to be alerted when policies of the shipping company are violated.

Regarding the main engine, in this project we focus particularly on a specific malfunction of a component of the main engine, i.e., the cross-head bearings. This is a damage that is not sudden, it evolves gradually and can only be detected with an on-site inspection. To the best of our knowledge, there is no correlation of the main engine data that pinpoint this malfunction. Its main cause is bad lubrication due to slow steaming, a policy imposed by charterers to ship-owning companies in order to reduce fuel consumption.

BigDataStack can assist with its architecture and the provided functionalities. In order to integrate DANAOS platform into BigDataStack and cherish its flexibility and functionality, all components of the architecture should be utilized. However, there is a set of components which are of major importance. The following figure describes these components and their interconnection.

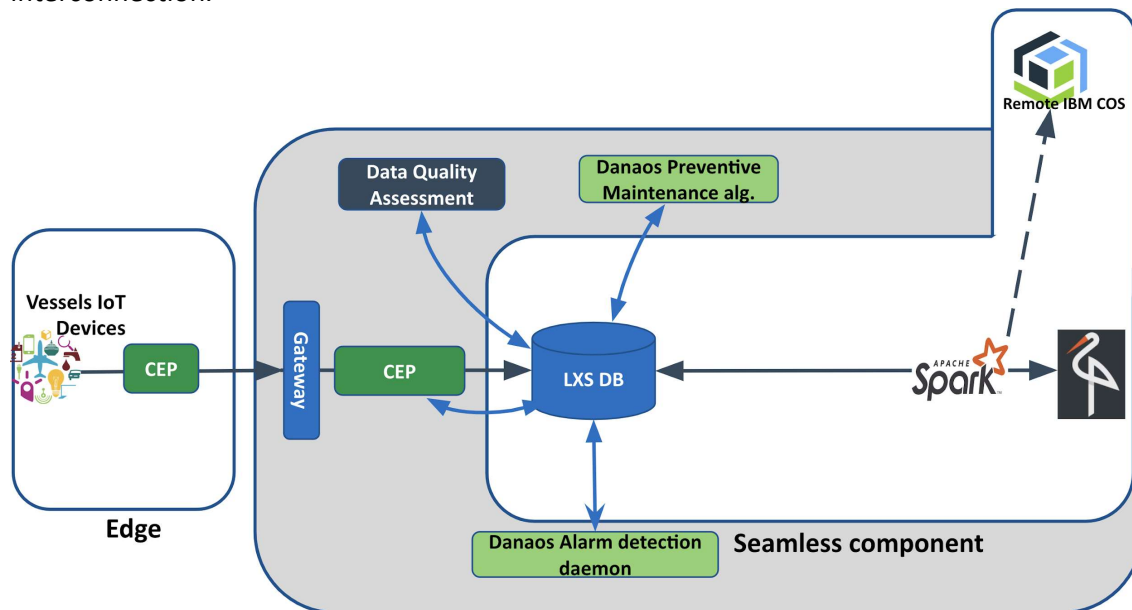


Figure 1 – RSM Pilot's architectural overview

The Edge of this architecture comprises of the IoT devices onboard along with the CEP component. At this point, CEP is used to monitor sensors and identify malfunctions on sensors that cause data-loss. That is, the identification of consecutive null values from a source or erratic values. The latter is identified by complex domain-specific rules. Once such a pattern is identified an alert is produced.

Past the edge, raw data and alerts flow from the Gateway again to another CEP component. This is used to monitor data and produce alerts whenever the policies of the shipping company are violated. The rules of these policies are stored in LXS DB. In this case, we identify the violation of the Charter Party Agreement, i.e., a deal between the shipping company and the charterer which guarantees a set of thresholds for the consumption of the vessel under specific weather conditions. Once data are processed from the CEP component, if necessary, an alert is produced and data are stored into LXS DB.

The seamless component comprises of the LXS DB, data-skipping and the IBM object storage. We use LXS DB to store fresh data and IBM object storage for historical data. The seamless component allows querying over both data-stores without worrying about data location.

On top of this, the DANAOS preventive maintenance algorithm, uses the seamless component to fetch data and identify malfunction patterns on the provided series of data. If a malfunction pattern is identified an alert is produced and stored into LXS DB. Note here, that this algorithm uses filtered/cleaned data, not raw data. The cleansing is performed by the Data Quality Assessment tool.

Finally, the DANAOS platform, uses a daemon that queries LXS DB for alerts and visualizes the produced alerts, whether it's a CEP or a preventive maintenance alert, to the end user.

Status

The current status includes the following major completed tasks:

- The use case services have all been deployed on BigDataStack infrastructure and the corporate datasets have been stored in the BigDataStack storage engine.
- Integration of DANAOS platform services with the BigDataStack storage engine and more specifically with the LeanXscale database to store and retrieve the corresponding datasets.
- Containerization of DANAOS platform services using Docker and Docker-compose.
- Deployment of containerized DANAOS platform services in BigDataStack infrastructure.
- Implementation and testing of a new predictive maintenance algorithm that efficiently detects a specific malfunction of the main engine of a vessel.
- Integration of the aforementioned algorithm in BigDataStack infrastructure so that it can be deployed and managed by the BigDataStack infrastructure management offering.
- Integration of CEP and analytics algorithm with DANAOS platform services. The CEP use is twofold: first we use it to identify data-loss onboard and secondly to produce alerts whenever a policy of the shipping company is violated.

Inclusion of IBM Object Storage data-skipping technology in the Seamless component that enhances query speed, as demonstrated in a joint presentation of IBM Israel and Danaos Shipping Co. Ltd. in the THINK conference.

Challenges

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.

3.3. Datasets

DANAOS Shipping Co. Ltd. possesses a complete dataset generated from sensors installed onboard along with operational data and historical data of malfunctions.

In more detail the used datasets in this project include the aforementioned categories for 10 vessels, i.e.,

- Operational data, i.e., telegrams (38K records) produced every 12 hours or upon arrival/departure of a vessel. Rarely updated, e.g., when an error is identified.
- General purpose sensor data (21 different sensors, 29M records, per minute basis).
- Main engine sensor data (100 different sensors, 29Mrecords, per-minute basis).
- Damage reports (25 different cross-head bearing damage incidents on the provided vessels).

Given the data schemas described below, we want to state that the DANAOS datasets do not have any GDPR-related aspect.

TELEGRAMS table structure (14 attributes)

id: Telegram id,

vessel_code: The id of the vessel,

telegram_date: Telegram timestamp (UTC),

type: Telegram type: D:Departure, A:Arrival, N:Noon-telegram,

total_teus: Total Twenty-foot Equivalent Unit (TEU) (# of containers)

total_feus: Total Forty-foot Equivalent Unit (FEU) (# of containers)

cons_ifo_static_counter: sensor-based measurement TEUs

cons_ifo_static1_counter: sensor-based measurement of FEUs,

draft_aft: Vessel draft at stern (m),

draft_fore: Vessel draft at fore (m),

sea_temperature: Sea temperature (°C),

port_name: Current port name,

next_port: The name of the next port,

eta_next_port: ETA to the next port

VESSEL_DATA table structure (23 attributes)

vessel_code: Vessel id,
datetime: Timestamp of the measurement (UTC),
power: Consumed power (kW),
apparent_wind_speed: Wind-speed (kn),
speed_overground: GPS speed (kn),
stw_long double precision: Speed through water – longitudinal (kn),
stw_trans double precision: Speed through water – transverse (kn),
rpm: rotations per minute of the main shaft,
apparent_wind_angle: Wind angle (0-359.99 degrees),
total_teus: Total Twenty-foot Equivalent Unit (TEU) (# of containers),
total_feus: Total Fourty-foot Equivalent Unit (FEU) (# of containers),
cons_ifo_static_counter: Low-sulfur fuel oil consumption (metric tones),
cons_ifo_static1_counter: High-sulfur fuel oil consumption (metric tones),
port_mid_draft: Vessel draft at port-side (left-side looking to the fore) (m),
stbd_mid_draft: Vessel draft at starboard-side (right-side looking to the fore) (m),
draft_aft: Vessel draft at stern (m),
draft_fore: Vessel draft at fore (m),
stw: Speed through water – calculated by stw_trans and stw_lon (kn),
equivalent_teus: Total number of containers,
mid_draft: Vessel draft at mid-line (m),
trim: The trim of the vessel, calculated by draft_aft and draft_fore,
latitude: The latitude of the vessel's position,
longitude: The longitude of the vessel's position,

MAIN_ENGINE_DATA table structure (102 attributes)

vessel_code: The id of the vessel,
datetime: Timestamp of measurement in UTC,
airCoolerCWInLETPress: Air Cooler Cooling Water Inlet Pressure (Pa)
scavAirFireDetTempNo1: Cyllinder #1 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo2: Cyllinder #2 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo3: Cyllinder #3 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo4: Cyllinder #4 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo5: Cyllinder #5 Scavenge Air Fire Detection Temperature (°C),

scavAirFireDetTempNo6: Cylinder #6 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo7: Cylinder #7 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo8: Cylinder #8 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo9: Cylinder #9 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo10: Cylinder #10 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo11: Cylinder #11 Scavenge Air Fire Detection Temperature (°C),
scavAirFireDetTempNo12: Cylinder #12 Scavenge Air Fire Detection Temperature (°C),
coolerCWinTemp: Air Cooler Cooling Water Inlet Temperature (°C)
cfWInPress: Cooling Fresh Water Inlet Pressure (Pa),
controlAirPress: Control Air Pressure (Pa),
cylLoTemp: Cylinder Lube Oil Temperature (°C)
exhVVSprngAirInPress: Exhaust Valve Spring Air Inlet Pressure (Pa)
foFlow: Fuel Oil Flowrate (lt),
foInPress: Fuel Oil Inlet Pressure (Pa),
foInTemp: Fuel Oil Inlet Temperature (°C),
hfoViscosityHighLow: Heavey Fuel Oil Viscosity High Low (mm²/s)
hpsBearingTemp: HPS Bearing Temperature (°C),
jcfWInTempLow: Jacket Cooling Fresh Water Inlet Temperature Low (°C)
cylExhGasOutTempNo1: Cylinder #1 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo2: Cylinder #2 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo3: Cylinder #3 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo4: Cylinder #4 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo5: Cylinder #5 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo6: Cylinder #6 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo7: Cylinder #7 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo8: Cylinder #8 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo9: Cylinder #9 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo10: Cylinder #10 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo11: Cylinder #11 Exhaust Gas Out Temperature (°C),
cylExhGasOutTempNo12: Cylinder #12 Exhaust Gas Out Temperature (°C),
cylJCFWOutTempNo1: Cylinder #1 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo2: Cylinder #2 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo3: Cylinder #3 Jacket Cooling Fresh Water Outlet Temperature (°C),

cylJCFWOutTempNo4: Cylinder #4 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo5: Cylinder #5 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo6: Cylinder #6 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo7: Cylinder #7 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo8: Cylinder #8 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo9: Cylinder #9 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo10: Cylinder #10 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo11: Cylinder #11 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylJCFWOutTempNo12: Cylinder #12 Jacket Cooling Fresh Water Outlet Temperature (°C),
cylPistonCOOutTempNo1: Cylinder #1 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo2: Cylinder #2 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo3: Cylinder #3 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo4: Cylinder #4 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo5: Cylinder #5 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo6: Cylinder #6 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo7: Cylinder #7 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo8: Cylinder #8 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo9: Cylinder #9 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo10: Cylinder #10 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo11: Cylinder #11 Piston Cooling Outlet Temperature (°C),
cylPistonCOOutTempNo12: Cylinder #12 Piston Cooling Outlet Temperature (°C),
tcExhGasInTempNo1: Turbo-Charger #1 Exhaust Gas Inlet Temperature (°C)
tcExhGasInTempNo2: Turbo-Charger #2 Exhaust Gas Inlet Temperature (°C),
tcExhGasInTempNo3: Turbo-Charger #3 Exhaust Gas Inlet Temperature (°C),
tcExhGasInTempNo4: Turbo-Charger #4 Exhaust Gas Inlet Temperature (°C),
tcExhGasOutTempNo1: Turbo-Charger #1 Exhaust Gas Outlet Temperature (°C),
tcExhGasOutTempNo2: Turbo-Charger #2 Exhaust Gas Outlet Temperature (°C),
tcExhGasOutTempNo3: : Turbo-Charger #3 Exhaust Gas Outlet Temperature (°C)
tcExhGasOutTempNo4: Turbo-Charger #4 Exhaust Gas Outlet Temperature (°C)
tcLOInLETPressNo1: Turbo-Charger #1 Lube Oil Inlet Pressure (Pa),
tcLOInLETPressNo2: Turbo-Charger #2 Lube Oil Inlet Pressure (Pa),
tcLOInLETPressNo3: Turbo-Charger #3 Lube Oil Inlet Pressure (Pa),
tcLOInLETPressNo4: Turbo-Charger #4 Lube Oil Inlet Pressure (Pa),

tcLOOutLETTempNo1: Turbo-Charger #1 Lube Oil Outlet Pressure (Pa),
tcLOOutLETTempNo2: Turbo-Charger #2 Lube Oil Outlet Pressure (Pa),
tcLOOutLETTempNo3: Turbo-Charger #3 Lube Oil Outlet Pressure (Pa),
tcLOOutLETTempNo4: Turbo-Charger #4 Lube Oil Outlet Pressure (Pa),
tcRPMNo1: Turbo-Charger #1 RPMs,
tcRPMNo2: Turbo-Charger #2 RPMs,
tcRPMNo3: Turbo-Charger #3 RPMs,
tcRPMNo4: Turbo-Charger #4 RPMs,
orderRPMBridgeLeverer: Order RPM (Bridge Lever)
rpm: Rotations per minute of the main shaft
scavAirInLetPress: Scavenge Air Inlet Pressure (Pa),
scavAirReceiverTemp: Scavenge Air Receiver Temperature (°C),
startAirPress: Starting Air Pressure (Pa),
thrustPadTemp: Thrust Pad Temperature (°C),
mainLOInLetPress: Main Lube Oil Inlet Pressure (Pa),
mainLOInTemp: Main Lube Oil Inlet Temperature (°C)
foTemperature: Fuel Oil Temperature (°C)
foTotVolume: Fuel Oil Total Volume (lt)
power: Consumed power (kW),
scavengeAirPressure: Scavenge Air Pressure (Pa)
torque: Torque of the main shaft (N/m),
coolingWOutLETTempNo1: Turbo-Charger #1 Air Cooler Cooling Water Outlet Temperature (°C),
coolingWOutLETTempNo2: Turbo-Charger #2 Air Cooler Cooling Water Outlet Temperature (°C),
coolingWOutLETTempNo3: Turbo-Charger #3 Air Cooler Cooling Water Outlet Temperature (°C),
coolingWOutLETTempNo4: Turbo-Charger #4 Air Cooler Cooling Water Outlet Temperature (°C),
foVolConsumption: Fuel Oil Consumption (lt/min)

VESSEL_DAMAGES table structure (5 attributes)

vessel_code: The id of the vessel,

defect_type: Type of damage (Main Bearing, Crosshead Bearing, Crankpin Bearing)

defect_details: Short description of damage

date_of_damage: Date of damage

cause_of_damage: Short description for cause of damage

3.4. Use Case Scenarios

3.4.1. Activity 1: Acquire data

The first activity is to define the data sources and acquire the required data. The Data Owner wishes to define the data sources and access protocols. The process is described in the following table.

Step	Description
RSM-A1-01	The Data Owner picks the data source type from a drop down list, e.g., database, FTP server etc. If the selected type is database then the RDBMS should be defined, again from a drop-down list, e.g., postgres, SQL-server etc.
RSM-A1-02	The Data Owner enters in plain text format the required connection attributes, i.e., IP address, username, password, etc.
RSM-A1-03	The system validates the connection.
RSM-A1-04	The Data Owner defines what particular resource is to be accessed. For example, if the selected datasource type is database, selects from a drop down list the preferred tales. If the source is an FTP server, the exact path should be defined.
RSM-A1-05	The system informs the user that the requested data sources are successfully defined.

3.4.2. Activity 2: Select attributes

Another activity is to select a set of attributes from the defined data sources. The process is described in the following table.

Step	Description
RSM-A2-01	The Data Analyst picks from the selected data-sources the contained attributes. For example, if the source is a table in a database, a list of columns is displayed where the Data Analyst picks the preferred columns.
RSM-A2-02	The system informs the user that the requested attributes are successfully defined.

3.4.3. Activity 3: Monitor the selected attributes

The Data Toolkit of BigDataStack provides the ability to monitor the selected attributes. The Data Analyst wishes to activate the monitoring component for the selected set, or sub-set of defined attributes.

Step	Description
RSM-A3-01	The Data Analyst picks from the defined attributes which are to be monitored.
RSM-A3-02	The system informs the user that the requested attributes are successfully defined.
RSM-A3-03	The Data Analyst sets the CEP rules for each variable or combination of variables.
RSM-A3-04	The Data Analyst confirms the rules for CEP monitoring.
RSM-A3-05	The system prompts the user to set the output source where results from CEP monitoring will be put.
RSM-A3-06	The system informs the user that the CEP monitoring is successfully configured.

3.4.4. Activity 5: Use an analytics algorithm or deploy a new one

BigDataStack offers a set of well-established analytics algorithms from which a Data Analyst can pick and deploy. Alternatively, the Data Analyst has the ability to inject a custom algorithm that suits the analytical task accordingly.

Step	Description
RSM-A5-01	The Data Analyst picks from a drop-down list the preferred algorithm.
RSM-A5-02.1.1	If the selected algorithm is an existing one, the system prompts the user to define the preferred attributes.
RSM-A5-02.1.2	The Data Analyst picks the preferred attributes.
RSM-A5-02.1.3	The system prompts the user to define the source where the output of the algorithm will be put.
RSM-A5-02.1.4	The Data Analyst sets the output source, e.g., database and table name etc..
RSM-A5-02.2.1	If the selected algorithm is a custom one, the system prompts the user to define the algorithm. That is, programming language, source code etc.
RSM-A5-02.2.2	The Data Analyst sets the requested information.
RSM-A5-02.2.3	The system prompts the user to define the preferred attributes.
RSM-A5-02.2.4	The Data Analyst picks the preferred attributes.
RSM-A5-02.2.5	The system prompts the user to define the source where the output of the algorithm will be put.
RSM-A5-02.2.6	The Data Analyst sets the output source, e.g., database and table name etc..
RSM-A5-03	The system informs the user that the analytics algorithm definition is successfully completed.

3.4.5. Activity 6: Produce alerts based on real-time monitoring information

Once the monitoring framework of BigDataStack is configured and the application services are successfully deployed, an engineer of the Technical department of the shipping company wishes to be informed whenever an alert from CEP monitoring is produced.

Step	Description
RSM-A6-01	The system stores an alert in the defined output source of the CEP.
RSM-A6-02	The alert is accessible from the application services, thus the application prompts the user that an alert is produced.

3.5. Next steps

The next steps include the implementation and deployment of the following use-cases.

3.5.1. *Activity 7: Order spare parts when necessary*

This activity includes the required steps to place an order for a spare part. After an alert is produced for an upcoming malfunction, an employee of the Supply department wishes to place the order for the respective spare part.

Step	Description
RSM-A7-01	The Supply department picks from a list of preferred suppliers the one that will deliver the spare part.
RSM-A7-02	The supplier through Danaos-One platform sends the respective offer with costs and time-frame of delivery.
RSM-A7-03	If the offer is accepted, the system produces an alert for the Operations department to re-route, if necessary, the vessel.

3.5.2. *Activity 8: Re-route the vessel accordingly*

This activity includes the required steps to re-route the vessel if an order for a spare part is placed.

Step	Description
RSM-A8-01	The Operations department is notified that a spare part has been ordered and will be delivered at a specific port.
RSM-A8-02	The Operations department, places the current location of the vessel and the delivery port in the system.
RSM-A8-03	The system calculates the optimal or close-to optimal route.
RSM-A8-04	The Operations is notified about the new route.

4. Connected Consumer (CC)

This section provides a description of the Connected Consumer use case scenario from ATOS Worldline.

4.1. Overview and goal

The Connected Consumer use case utilizes the BigDataStack environment to implement and offer a recommender system for the grocery market. All of the data that are used for training the analytic algorithms of the use case are corporate data provided by EROSKI, one of the top food retailers companies in Spain.

What is the scenario's goal?

The final goal of the scenario is to take advantage of the capabilities of BigDataStack to produce recommendations to the customers of the on-line applications of EROSKI.

What is the business objective?

The business objective of this scenario is to create a collaborative-filtering recommender system that produces recommendations of new products to the customers of EROSKI. Thus helping EROSKI to improve the user experience of some of their current applications (e-commerce site, loyalty app, ...) and at the same time to improve their customers' loyalty.

Who are the actors in the use case?

The actors involved in the first scenario of the CC scenario have been extracted from D2.1. Concretely, these are summarized in the following table:

Table 2 – Connected Consumer Actors

Id	Name	Description
ROL-01	Data Owner	BigDataStack offers a unified Gateway to obtain both streaming and stored data from data owners and store them in its underlying storage infrastructure that supports SQL and NoSQL data stores. In the CC the Data Owner is EROSKI.
ROL-02	Data Scientist	BigDataStack offers the Data Toolkit to enable data scientists both to easily ingest their analytics tasks by utilizing a declarative paradigm, and to specify their preferences and constraints to be exploited during the dimensioning phase regarding the data services that will be used (for example preferences for the data cleaning service)
ROL-03	Business Analysts	BigDataStack offers the Process Modelling Framework allowing business users to define their functionality-based business processes (through declaratively-defined models) and optimize them based on the outcomes of process analytics that will be triggered by BigDataStack.

ROL-04	Application Engineers and Application Service Owners	BigDataStack offers the Application Dimensioning Workbench to enable application owners and engineers to experiment with their applications and dimension it in terms of its data needs and data-related properties
ROL-CC-1	Grocery Consumers	BigDataStack offers a performant environment enabling the achievement of the goals set by the use case in terms of resource provisioning and execution of analytics based on given service level objectives. Based on that, the customers of EROSKI, are end-users of any of the applications of EROSKI (e.g. someone who is browsing in the e-commerce site of EROSKI) and will obtain the recommendation results following the successful execution of the analytics pipelines on BigDataStack infrastructure.

What are the key activities these actors are involved in?

The CC use case can be broken down into the following main activities:

1. **Definition of the analytics for the recommender.** This activity aims at defining the business processes and the main analytical tasks that the recommender needs. The main actors in this activity are the Business Analyst and the Data Scientist.
2. **Deployment of the application services.** This activity aims at testing the capability to deploy on top of BigDataStack the application services of the recommender system. This activity applies to several actors: the business analyst, the data scientist and the application engineer. The former sets the time constraint for the whole process of recommendations. The data scientist calculates the service level objectives (SLOs) needed for each of the application services that compose the application. The application engineer tests several configurations before proceeding to do the final deployment.
3. **Re-Deployment of the application services.** This activity demonstrates the capability of BigDataStack to make run-time adaptations in the configuration of the application services due to changes in the incoming data. The activity applies to the application engineer and the data owner.
4. **Visualize recommendations.** The Data scientist wants to know which recommendations the system will propose for a given user. Main actors in this activity are the data scientist and the business analyst.
5. **Provide recommendations.** The External Applications need to provide recommendations to their users. E.g. a new App oriented to strengthen the loyalty of the EROSKI end-users wants to have the feature of suggesting new products to its users. Main actor in this activity is an end user of the client applications of the recommender.

6. **Collect feedback.** The Data scientist wants to have further information about the success or failure of the recommendations provided by the recommender. For this reason, the system is prepared for collecting feed-back from the external systems about the usage of the recommendations. Main actors in this activity are the data-owner and the data scientist. This activity aims at testing the capabilities of BigDataStack to ingest data through the Gateway.

4.2. Pilot description

The pilot that has been implemented relies on the LeanXcale data store of BigDataStack. Before running the pilot, 2 years of history have been loaded into LeanXcale.

The pilot that has been implemented contains 2 different sets of application and data services:

- i. web services access layer
- ii. estimate recommendations

The first set of application services, Web Services access layer, is the entry-door for the external applications to the recommender system.

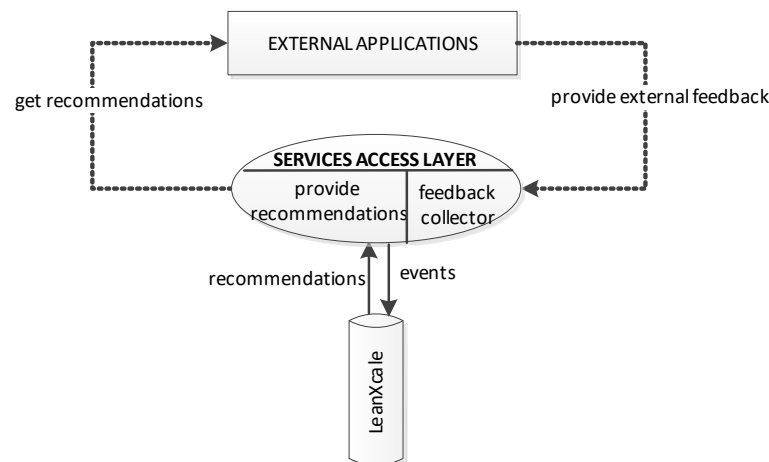


Figure 2 – Web Services access layer

It consists of two different services:

- *Feedback collector service*, external applications provide information about the interactions between a user and a product by means of this service.
- *Provide recommendations service*, external applications retrieve the recommendations calculated by the system with this service. Consumers of these services will be the client applications that need to show recommended products to its users. Multi-channel is provided by the system.

The second set of application services, named 'calculate recommendations', aims at running the analytic flow to calculate the products that will be recommended to the user.

3 different application services have been implemented in the analytic flow:

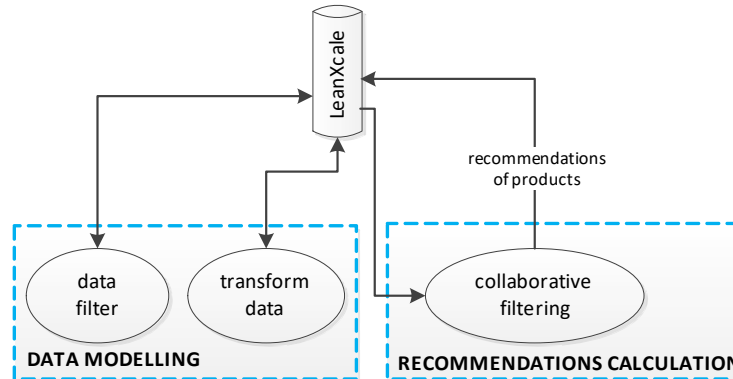


Figure 3 – Analytic flow

Concretely,

- Data filter. In charge of selecting the necessary data to run the model.
- Transform data. In charge of calculating the input matrix needed by the model.
- Collaborative filtering. Execution of the algorithm.

Many retailers nowadays are offering recommendations of products and promotions to their customers. However, these recommendations are sometimes calculated weeks in advance to a certain promotional campaign. To make matters worse, many times the calculation of these recommendations is done using outdated customer segments. In this context, the challenge is to be able to speed up the process of calculation of recommendations so that users can be offered personalized suggestions of products. In order to fulfil this objective, the system implemented aims at providing the capability to make in-store consumer-tailored predictions to its users. As soon as an interaction between a user and a product arrives to the system, the analytic flow needed to calculate the recommendations for a certain user is triggered. In this way, the system is offering fresh recommendations to its users.

4.3. Datasets

As stated before in the document, all of the data that have been used for training the analytic algorithm of the use case are corporate data provided by EROSKI, one of the top food retailers companies in Spain.

The dataset contains information about EROSKI clients. However, GDPR aspects have been taken into account before sharing the data with the consortium. Concretely:

- The only data that could be used to uniquely identify a person related to the field "ID_CLIENTE".
- ID_CLIENTE is an internal identifier of the database of EROSKI that is not known by

the customers. I.e. only a person with access to the database of EROSKI could identify the customer from ID_CLIENTE.

- ID_CLIENTE has been encrypted by EROSKI with an SHA-1 algorithm. Encryption has been done before providing the dataset to BigDataStack consortium. A SHA-1 (168 bits) algorithm has been used for encryption of ID_CLIENTE.
- For each ID_CLIENTE, SHA-1 has been applied to “string_1”+ID_CLIENTE+”string_2”. String_1 and string_2 are alphanumeric that contain capital and non-capital letters, numbers and special characters. These 2 values are only known by EROSKI.

This section describes the data entities that have been used in the implementation of the recommender system.

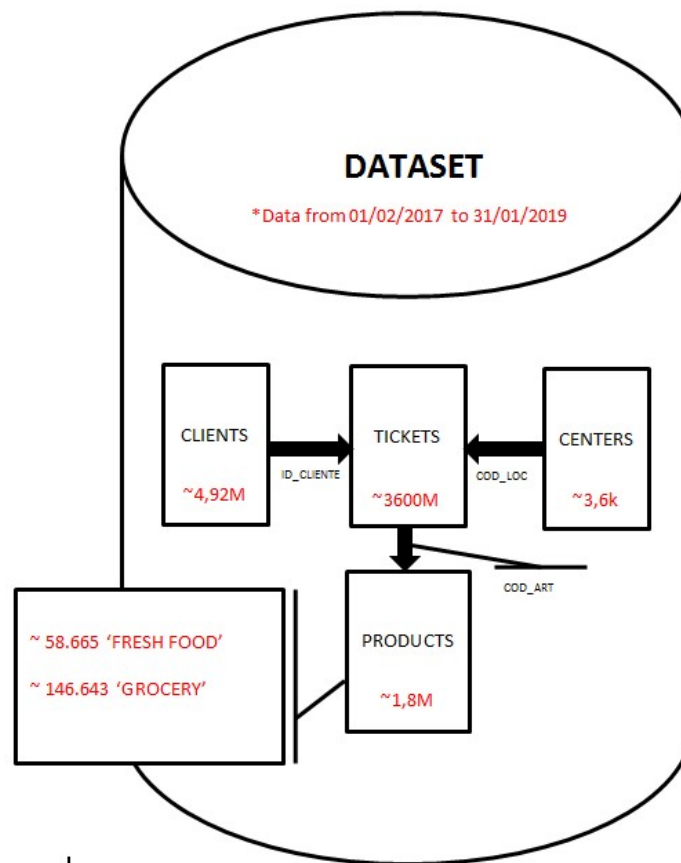


Figure 4 – CC Database structure

The attributes for each entity have been included in this section.

CLIENTS table structure (21 attributes)

ID_CLIENTE: Client id,

TIPO_CLIENTE_ORO: Type of gold client

FLG_CLIENTE_APP: Flag if the client is an app client or not,

FLG_CLIENTE_WEB: Flag if the client is a web client or not,

FLG_CLIENTE_NUTRICIONAL: Customer shows interest in healthy products

FRANJA_GASTO_ORO_INICIAL: Initial Range of expenditure

POSIBLE_VALOR_ORO: Percentage indicating the discount given to the customer for being a gold customer

CLIENTE_1000_ORO: Flag indicating whether the client is 1000 Oro or not

FRANJA_GASTO_ORO_ACTUAL: Current Range of expenditure

TIPO_MADUREZ: Type of maturity of the client

DESC_SEG_C_CLIENTE: Description of the type of maturity of the client

DESC_SEG_G_FIDELIDAD: Segmentation of the customer according to his loyalty

DESC_INTERES_AHORRO: Segmentation of the customer according to his interest in promotions

DESC_INTERES_FRESCOS: Segmentation of the customer according to his interest in fresh food

DESC_INTERES_LOCAL: Segmentation of the customer according to his interest in local food

DESC_INTERES_SALUD: Segmentation of the customer according to his interest in healthy food

DESC_INTERES_SALUD_DETALLE: additional detail on which type of healthy food the customer is interested in

DESC_MISION_COMPRA: description of the purchase mission of the customer

DESC_SEG_SEC: segment description

DESC_SEG_SOCIODEMO: Socio-demographic segment of the client.

COD_LOC: preferred store

TICKETS (36 attributes)

ID_CLIENTE: Client id,

COD_LOC: Store's localization id,

DIA: Day,

COD_CAJA: Till id,

NUM_TICKET: Ticket number (id),

NUM_LINEA: Line number (id),

COD_TIPO_MOVIM: Movement type,

HORA_EMISION: Timestamp of tickets emission,

COD_TIPOMARCA_HIST: Type of brand of the product

COD_F_PAGO_DET -> M_FORMA_PAGO: Type of payment procedure,

UNID_VENTA_TARIFA: Total amount of items sold in tariff's type,
UNID_VENTA_OFERTA: Total amount of items sold in offer's type,
UNID_VENTA_COMPETE: Total amount of items sold in competence's type,
UNID_VENTA_LIQUID: Total amount of items sold in liquidation's type,
UNID_VENTA_CAMPANA: Total amount of items sold in campaign's type,
IMP_VENTA_TARIFA: Total economic amount of the items sold by tariff's type,
IMP_VENTA_OFERTA: Total economic amount of the items sold by offer's type,
IMP_VENTA_COMPETE: Total economic amount of the items sold by competence's type,
IMP_VENTA_LIQUID: Total economic amount of the items sold by liquidation's type,
IMP_VENTA_CAMPANA: Total economic amount of the items sold by campaign's type,
IMP_DTO_CONSUMER: Discount amount applied for using VISA Eroski,
IMP_DTO_TRAVEL: Discount amount applied for using loyalty card Travel Club,
IMP_DTO_COUPON: Discount amount applied for the usage of coupons,
IMP_DTO_CUOTA: Discount amount applied for being member of EROSKI Club,
IMP_DTO_ONSITE: Discount amount applied after redemption of loyalty Travel points,
IMP_DTO_OTROS: Other discounts,
IMP_DTO_VALE: Amount of discounts coming from the redemption of a supplier coupon,
IMP_CONSUMO_RAP: Special discount applied in the shop,
COD_ART: Article's id,
FLG_TECLA: information about whether the product has been sold by a direct key or not
ANO_OFERTA: year of the offers applied to the order
COD_OFERTA: offer code
COD_TIPO_CENTRO: type of shop (primary/secondary)
FLG_SCANNER: has the product been scanned during the purchase (Y/N)
IMP_PVP_TARIFA: amount of the order if all of the items had been charged to the customer with catalogue prices

CENTERS structure (55 attributes)

COD_LOC: Store's localization id,
COD_PROVIN: Province id,
DESC_LOC: Center's description,
DESC_PROVIN: Province's name,
FLG_PLATAF : Indicator of distribution platform,
FEC_MODIF: Date of last modification,

COD_ZONA: Zone id,
DESC_ZONA: Zone description,
COD_REGION: Region id,
DESC_REGION: Region description,
COD_AREA: Area id,
DESC_AREA: Area's description,
COD_ENSENA: Type of center id,
DESC_ENSENA: Type of center description (Eroski City, Eroski Center...),
COD_NEGOCIO: Store's id,
DESC_NEGOCIO: Store's type,
COD_SOCIEDAD: Type of company,
DESC_SOCIEDAD: Company's description,
COD_GAMA_OBLIG: Code of mandatory catalogue,
COD_FINANZIA: financing code,
DESC_DIRECCION: address,
DESC_POBLACION: location,
FLAG_CUOTA: quota flag,
FEC_INI_LOC: opening date,
FEC_FIN_LOC: closing date,
NUM_CAJAS: number of boxes,
NUM_M2: squared meters of the store,
NUM_M_LINEA: linear meters,
COD_LOC_AME: store code in AME system,
COD_TP_LOC: type of location,
DESC_TP_LOC: description of the type of location,
COD_LOC_PADRE: father location code,
COD_MUNICIPIO: location code,
COD_TP_POTENCIAL: type of potential code,
FEC_ULT_APERTURA : last opening date,
COD_POSTAL: zip code,
COD_AGR_IMP: grouping code,
FLG_CECO_MODELO_COSTES: cost model flag,
LATITUD: latitude,

LONGITUD: longitude,
COD_ISLA: ISLA code,
FLG_LEAN: lean flag,
FLG_TRANSFORMADO: transformed flag,
FLG_PUESTA_PUNTO_PLUS: tuning flag,
COD_NIVEL_ESTR_LOC: code of local structure of sales of the center,
COD_N1: code of the level 1 of the structure of sales of the center,
DES_N1: description of the level 1 of the structure of sales of the center,
COD_N2: code of the level 2 of the structure of sales of the center,
DES_N2: description of the level 2 of the structure of sales of the center,
COD_N3: code of the level 3 of the structure of sales of the center,
DES_N3: description of the level 3 of the structure of sales of the center,
COD_N4: code of the level 4 of the structure of sales of the center,
DES_N4: description of the level 4 of the structure of sales of the center,
COD_N5: code of the level 5 of the structure of sales of the center,
DES_N5: description of the level 5 of the structure of sales of the center,

PRODUCTS structure (79 attributes)

COD_ART: product id,
DESC_ART: product description,
FLG_TECLA: exists a direct key to sell the product or not,
COD_TIPOMARCA: type of brand code,
DESC_TIPOMARCA: description of the type of brand code,
COD_N1_PPAL: Area's id,
DESC_N1: Area's description,
COD_N2_PPAL: Section's id,
DESC_N2: Section's description,
COD_N3_PPAL: Category's id,
DESC_N3: Category's description,
COD_N4_PPAL: Subcategory's id,
DESC_N4: Subcategory's description,
COD_N5_PPAL: Segment's id,
DESC_N5: Segment's description,

FEC_INI_ART: Article start time,
FEC_FIN_ART: Article finishes time,
COD_FORMATO: Format id (KG, Gr, Unities...),
COD_MARCA: Brand's id,
COD_EAN: EAN code,
COD_TALLA: Size code,
DESC_TALLA: Size code description,
COD_COLOR: Colour code,
DESC_COLOR: Colour code description,
COD_PACK : Number of items per pack,
COD_BLOQUEO: has the product blocked for the sales?,
COD_ENS_EROSKI: commercial codification in the Hypermarket,
COD_ENS_CONSUM: commercial codification in the SUPERmarket,
COD_TIPO_FORMATO: unit of measurement (related to COD_FORMATO),
COD_ART_PRIM: father product code,
COD_TIPO_MARCA2: code of EROSKI Brand (only for products belonging to a EROSKI brand)),
DESC_TIPO_MARCA2: description of EROSKI Brand (only for products belonging to a EROSKI brand)),
FEC_ULT_BLOQ: date on which the product was blocked for the sales,
COD_PORCI_CONS: product has info for the consumer related to the number of portions,
DESC_PORCI_CONS: indicator about whether the product has a description for the portions,
CC_CAPRABO: Comercial code of CAPRABO,
COD_CATEGORI_HIP: Category code hypermarket,
DESC_CATEGORI_HIP: Description of the Hypermarket Category,
COD_CATEGORI_SUP: Category code supermarket,
DESC_CATEGORI_SUP: Description of the supermarket Category,
COD_SENSIBI_HIP: SENSIBI code hypermarket,
DESC_SENSIBI HIP: Description of the SENSIBIcode of the hypermarket,
COD_SENSIBI SUP: Category code supermarket,
DESC_SENSIBI SUP: Description of the SENSIBIcode of the supermarket,
FLG_COMPRA: indicator about whether the product is for purchasing,
FLG_VENTA: indicator about whether the product is for sales,
COD_FAMILIA: family of the product,

DESC_FAMILIA: description of the family of the product,
COD_AMBITO_EROSKI: Scope code of the product in the hypermarkets,
DESC_AMBITO_EROSKI: Description of the scope of the product in the hypermarkets,
COD_AMBITO_CONSUM: Scope code of the product in the supermarkets,
DESC_AMBITO_CONSUM: Description of the scope of the product in the supermarkets,
COD_CODMARCA: brand code (related to COD_MARCA)
FLG_MMPP: Does the product belong to a EROSKI brand?,
COD_POSICION_MARCA: Maker brand / EROSKI Brand code,
DESC_POSICION_MARCA: Description of the code of maker Brand / EROSKI Brand code,
FLG_SALUD_BIENESTAR: health indicator,
FLG_INNOVACION: innovation indicator,
FLG_GAMA_TURISTICA: tourism product,
FLG_PODER_ADQUISITIVO: indicator about product for customer with a high purchasing power,
FLG_BLOQ_DEFINITIVO: Product definitely blocked,
COD_SUBMARCA: sub-brand code,
DESC_SUBMARCA: sub-brand description
FLG_GAMA_LOCAL: local product,
FLG_GAMA_REGIONAL: regional product,
FLG_PESO_SGA: flag product by weight,
FLG_LIQUIDABLE: flag payable,
FLG_EXDEPRECIACION: depreciation flag,
COD_TP_ART: product type,
DESC TIPO_ARTICULO: description of the product type,
CANTIDAD: number of ítems per lot,
FEC_LANZAM: launch date,
PORC_IVA: VAT rate,
COD_PROVR_GEN: code of generic supplier,
COD_PROVR_TRABAJO: code of work supplier,
NOMBRE: name of the work supplier,
PESO: weight (in grams),
PESO_NETO: net weight (in grams),
VOLUMEN: volume (in cm3)

4.4. Use Case Scenario

4.4.1. Activity 1: Definition of the analytics for the recommender

The first activity is related to the definition of the analytic flow of the recommender system in BigDataStack.

In this activity the main actors are the business analyst who defines the high-level analytic tasks needed for the recommender and the data scientist who has to map the high-level tasks defined by the business expert to concrete analytic tasks. The following characters, who represent different end-users, will take part in the activity:

- **Fernando** is the business analyst who is in charge of declaring the business process at a high level. He has also defined some business constraints for the system.
- **Martí** is a data scientist who has been tuning the different parameters for the analytic algorithm and testing the different application services.

Step	Description
CC-A1-01	<p>From the Process Modelling Framework, Fernando builds the graph that represents the application. Concretely, Fernando adds the nodes feedback collector, data filter, transform data and collaborative filtering and links them together.</p> <p>Fernando is also in charge of defining the global business goals of the system. One of them is Completion time < 1 minute (i.e. when the recommender receives feedback of an interaction <user, product>, the refreshment of the recommendations for this user should take less than a minute)</p> <p>Once the graph is ready and saved the work made by Fernando in the Process Modelling Framework is available for Martí in the Data Toolkit.</p>
CC-A1-02	<p>From the Data Toolkit, Martí opens the graph deployed by Fernando. He concretizes the analytic tasks defined by Fernando in the business graph. Martí makes the following actions in the Data Toolkit:</p> <ul style="list-style-type: none"> • Links the high-level analytic tasks defined by Fernando to concrete executables.

	<ul style="list-style-type: none"> Sets the SLOs for each of the executables. Concretely, the needed throughput for each of the application services of the analytic flow. To make his calculations, Martí takes into account the global business goal defined by Fernando in the previous step. Sets values for parameters of recommendation algorithm such as implicitPrefs, maxIter (maximum number of iterations to be run).
--	--

4.4.2. Activity 2: Deployment of the application services

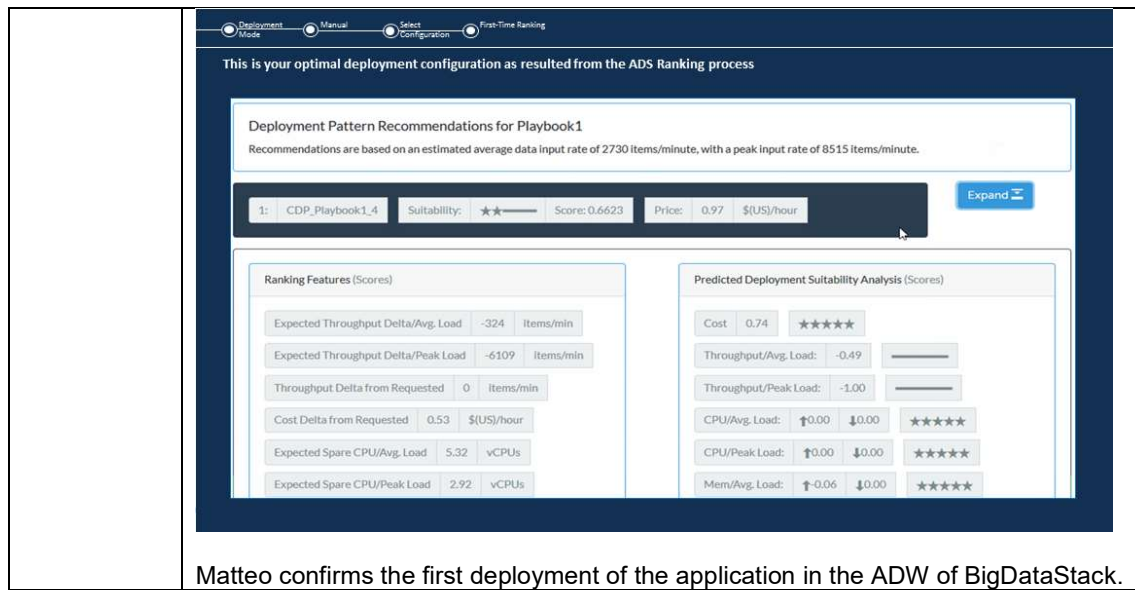
This activity is related to the deployment of the application specific services of the recommender system.

The user will have two possibilities to deploy his application in the Application Dimensioning Workbench (ADW): assisted and manual. This activity is focused on the manual deployment. In this type of deployment, the user selects the QoS and the resources that are expected to be needed for each of the application services. Based on these input parameters, BigDataStack suggests to the user the deployment configuration that suits best to the application.

In this activity, the main actors are the application engineer, the data scientist (who has to define the application service level objectives (SLOs) for each of the application services of the recommender) and the business analyst. We will work with the following characters that represent the different end-users:

- Fernando** is the business analyst who is in charge of declaring the business process at a high level. He has also defined some business constraints for the system.
- Martí** is a data scientist who has been tuning the different parameters for the analytic algorithm and testing the different application services.
- Matteo** is the application engineer who has been developing the different application components of the recommender system

Step	Description
CC-A2-01	Fernando sets the business goals of the system. One of them says 'once the recommender receives feedback of an interaction <user, product> the refreshment of the recommendations for this user shouldn't take more than one minute.
CC-A2-02	Martí, needs to deploy the application services of the recommender system he has been experimenting with. Martí calculates the SLOs for the application services of the recommender so that the whole process takes less than one minute for the calculation of the recommendations of a user. Then provides the numbers to Matteo with the help of the data toolkit.
CC-A2-03	Matteo makes an estimation of the resources (CPUs, Memory and number of replicas) needed for each application service and then registers the SLO's and the estimation in ADW. BigDataStack provides Matteo the a priori best deployment pattern.



Matteo confirms the first deployment of the application in the ADW of BigDataStack.

4.4.3. Activity 3: Re-deployment of the application services

Once the Application Services have been deployed in BigDataStack, changes in the incoming data may happen and the fulfilment of the SLO's could be in danger. This activity aims at demonstrating the capability of BigDataStack to identify that the SLO's defined for the application services are not being fulfilled as well as to adapt the underlying needed resources so that the SLO's are fulfilled again.

In this activity, the main actors are the application engineer who has to confirm or not the redeployment of the application services and the data owner who has to accept the increment of economic costs of the application due to the new resources needed.

- **Matteo** is the application engineer who has been developing the different application components of the recommender system. He was the one making the first deployment of the application services of the recommender system with the Application Dimensioning Workbench.
- **Rosanna** is the data owner. She is also the owner of the budget for the recommender system.

Step	Description
CC-A3-01	Matteo receives a warning from ADW that alerts him that there has been detected a quality of service failure and recommending him to alter the application deployment. In ADW, Matteo visualizes the different new alternatives that have been calculated by BigDataStack. He can see that all of them are more expensive than the current one.
CC-A3-02	Matteo transmits to Rosanna the different possibilities in terms of costs and performance. Having analysed the different possibilities, Rosanna accepts one of the new configurations without altering the SLO's.

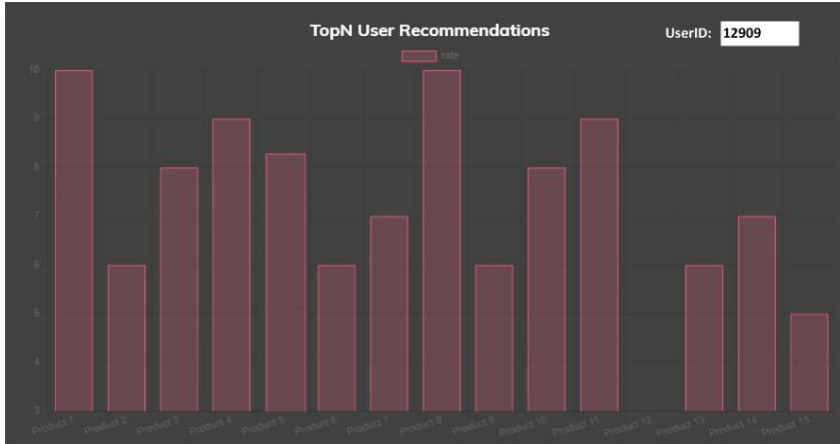
CC-A3-03	Matteo selects the configuration accepted by Rosanna and confirms the re-deployment of the application in the ADW of BigDataStack.
----------	--

4.4.4. Activity 4: Visualize recommendations

This activity is related to the retrieval and visualization of the recommendations calculated by the system. It demonstrates the capability of the adaptable visualizations component of BigDataStack to display the outcomes produced by the recommender system.

In this activity, the main actor is the data scientist, who wants to visualize the recommendations calculated by the system for a given customer. We will work with the following character that represents the end-user:

- **Martí** is a data scientist who is testing and analyzing the results of the application services layer offered by the system.

Step	Description
CC-A4-1	<p>Martí, is experimenting with the recommender system and needs to display the outcomes calculated for the system for a given user. He connects to the adaptable visualization component and enters the user whose recommendations wants to analyse.</p> <p>BigDataStack provides him with information about the recommendations calculated for the user – the top products calculated for the user as well as the ranking given for each of these products. Find below a prototype of the screen he will be displaying:</p> 

4.4.5. Activity 5: Provide recommendations

This activity relates to the capability of the recommender system to provide the recommendations calculated by the system to external systems. The data storage used for the recommender system has been the LeanXcale SQL data store provided by BigDataStack. It demonstrates the capability of BigDataStack to retrieve data from one of its data stores.

In this activity, the main actor is an end user of the ecommerce site of EROSKI, who is browsing in the e-commerce site of EROSKI. We will work with the following character that represents the EROSKI end-user,

- **Manuel** is a shopper who is making a purchase in the e-commerce site of EROSKI.

Step	Description
CC-A5-01	Manuel is navigating in the e-commerce site of EROSKI in order to make his weekly purchase. He accesses the section of the store named beers. The e-commerce site requests to the recommender system the recommendations to provide for the user Manuel.
CC-A5-02	The recommender retrieves the data from the BigDataStack data store to provide the e-commerce the recommendations calculated.
CC-A5-03	The e-commerce site selects those products that are available in the store on which Manuel is currently buying: Then displays them to Manuel somewhere in the screen of the section beers.

4.4.6. Activity 6: Collect feedback

This activity relates to the capability of the recommender system to collect feedback from the external systems that are using the recommendations calculated by the system. It demonstrates the capability of BigDataStack to store valuable data in one of its data stores, concretely, in LeanXcale.

In this activity, the main actor is an end user of the ecommerce site of EROSKI, who is browsing in the e-commerce site of EROSKI. We will work with the following character that represents the EROSKI end-user:

- **Manuel** is a shopper who is making a purchase in the e-commerce site of EROSKI.

Step	Description
CC-A6-01	Manuel is navigating in the e-commerce site of EROSKI in order to make his weekly purchase. He is located in the section of the store named beers. The e-commerce has recommended to him to buy 'Feta cheese' because the collaborative algorithm has calculated to be likely he would like this product.
CC-A6-02	Manuel has heard of Feta cheese before but never tasted it. He clicks on the product in order to see the detailed product cart.
CC-A6-03	The e-commerce site sends an event to the recommender system warning it that Manuel may be interested in the item.
CC-A6-04	The recommender system stores the event 'Product visualized' in the LeanXcale data store and launches the calculation of the recommendations for Manuel.

4.5. Next steps

In the second half of the project both business and integration objectives are expected to be fulfilled. Concretely, these are summarized in the following table:

Type of objective	Description
Business	Introduce a clustering algorithm for Segmentation of customers in the current scenario that will help to produce more accurate recommendations. The current algorithm is calculating the recommendations based on the shop of the purchases.
Business	Create a model for prediction of the next purchase of a given customer. The goal is to predict what the customers will need in their next purchase based on both the corporate data (history of purchases and product characteristics) and open data, i.e. predictive shopping list.
Integration	Use CEP and Gateway for injecting corporate data in real-time such as order items, products,
Integration	Full integration with the end-user tools of BigDataStack such as Process Modelling and Data Toolkit, since we want to be able to inject new analytical models.

5. Smart Insurance (SI)

This section provides a description of the Smart Insurance (SI) use case scenario from GFT.

5.1. Overview and goal

The BigDataStack financial use case has been moved from the Banking context (as described in the DoA and in the initial project deliverables, as D2.1) to the Insurance sector due to the delay in data collection regarding the banking context. The Smart Insurance (SI) use case will use BigDataStack to implement smart recommendation systems for the insurance market. The datasets that will be used within the process of the use case is corporate data provided by an insurance company, based in Italy, selected from the GFT customers' portfolio.

What is the scenario's goal?

The overall goal of the BigDataStack insurance demonstrator is to exploit the BigDataStack platform big data technologies in order to access and analyse information coming from diverse and heterogeneous data sources including the in-house data (e.g. customers location, customers portfolio), in order to provide recommendations for Insurance Companies allowing them a better customers' management (e.g. strategies for cross-selling and up-selling).

What is the business objective?

The business objective of this scenario is to allow insurance companies to improve the management of their customers' portfolio, by providing smart services that provide personalized products to their clients, through collaborative-filtering algorithms and cross-selling/up-selling strategies. Also, the aim is to help the company to improve their customers' propensity (i.e. their will to continue using the insurance's services) and on the other hand, predict the customers' churn rate.

Summarizing, the expected benefits of the BigDataStack adoption by Insurance companies, is to improve the customer's satisfaction through personalised offering and support. This is made possible by rapid processing and analysis of huge volumes of cross domain data.

Who are the actors in the use case?

The actors involved in the first scenario of the SI pilot have been extracted from D2.1, as reported in the following table.

Table 3 – Smart Insurance Actors

Id	Name	Description
ROL-01	Data Owner	BigDataStack offers a unified Gateway to obtain both streaming and stored data from data owners and store them in its underlying storage infrastructure that supports SQL and NoSQL data stores. In the SI the Data Owner is the Insurance company.
ROL-02	Data Scientist	BigDataStack offers the Data Toolkit to enable data scientists both to easily ingest their analytics tasks by utilizing a declarative paradigm, and to specify their preferences and constraints to be exploited during the

		dimensioning phase regarding the data services that will be used (for example preferences for the data cleaning service)
ROL-03	Business Analysts	BigDataStack offers the Process Modelling Framework allowing business users to define their functionality-based business processes (through declaratively-defined models) and optimize them based on the outcomes of process analytics that will be triggered by BigDataStack.
ROL-04	Application Engineers and Application Service Owners	BigDataStack offers the Application Dimensioning Workbench to enable application owners and engineers to experiment with their applications and dimension it in terms of its data needs and data-related properties
ROL-SI-1	Insurance company	Agents of the Insurance company. For example, they will receive suggestions about which products offer to which clients, through a web application developed within the project.

What are the key activities these actors are involved in?

The SI use case can be broken down into the following main activities:

1. Data acquisition. This activity aims at set up a data source from which BigDataStack obtains data to store in its infrastructure. At this stage, personal information is removed from the datasets from the data owners. Thus, the main actor in this step is the Data Owner.
2. Analytics definition. This activity aims at defining the business processes and the main analytical tasks that the recommender system needs. The main actors in this activity are the Business Analyst and the Data Scientist.
3. Deployment of the application services. This activity aims at testing the capability to deploy on top of BigDataStack the application services of the recommender system. This activity mainly applies to the application engineers.
4. Display recommendations. The Data scientist wants to know which recommendations the system will propose for a given user. Main actor in this activity is the data scientist.
5. Provide recommendations. The External Applications need to provide recommendations to their users. For example, a new web application for Insurance agents, devoted to strengthen the loyalty of the insurance company end-users suggesting new products and offers. Main actor in this activity is an Insurance Agent through the client applications of the recommender system.

5.2. Pilot description

Data analytics in the insurance industry is transforming the way insurance businesses operate. Data and analysis have always been the basis of the insurance industry, although in the last years the evolution of ICT solutions in data analysis reflected, as reported by Accenture¹, in

¹ See: https://www.accenture.com/_acnmedia/PDF-79/Accenture-Technology-Vision-Insurance-2018.pdf

an increase of the investments. In fact, the 80% of insurers currently invest moderately or significantly in new digital technologies and the 61% are expecting to increase their investment soon. On the other hand, the Chartered Institute of Loss Adjusted² stated that 82% of industry professionals believe organizations which do not utilize big data will likely become uncompetitive. The technological support impacted on the insurance sector as well as in the other fields, and insurers are leveraging data to attract, retain and service clients and producers, develop new products, assess and mitigate risks, set rates, process claims and manage financial performance.

The BigDataStack platform is the perfect candidate as backbone for the full management of the data collected by insurance companies: BigDataStack has the right tools to make the data, from different sources and of different formats, meaningful, and to manage them in the whole stack of related applications. By combining data with analytics, insurers can generate insights that help transform the business, create closer relationships with customers, gain competitive advantage or perhaps generate entirely new business models.

The SI pilot comprises two main scenarios:

1. Customer segmentation: customers' segmentation according to their history, age, location, etc. Thus, all the customers are classified into groups by spotting coincidences in their attitude, preferences, behaviour, 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 similar customers bought) and upselling (recommendations of more advanced products to customers based on what they have already bought) strategies may be developed and personal services may be tailored for each particular segment. This scenario is distributed in three steps:
 - - Data collection.
 - - Optimizing the product configuration, and recommendations.
 - - Show recommended products.
2. Lifetime value prediction scenario: customers' lifetime value represents 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 (propensity). Prediction of the CLV is typically assessed via customer behaviour data in order to predict the customer's profitability for the insurer. Thus, the behaviour-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), as well as the customer propensity likelihood. From a machine learning perspective, propensity/churn can be formulated as a binary classification problem.

The pilot that has been defined relies on the LeanXcale data store of BigDataStack. Figure 5 describe the high-level foreseen architecture of the pilot.

[Accessed on June 18, 2019].

² See: <https://www.the-digital-insurer.com/wp-content/uploads/2015/03/478-3-ASNY-Presentation-Predictive-Analytics-Final.pdf> [Accessed on June 18, 2019].

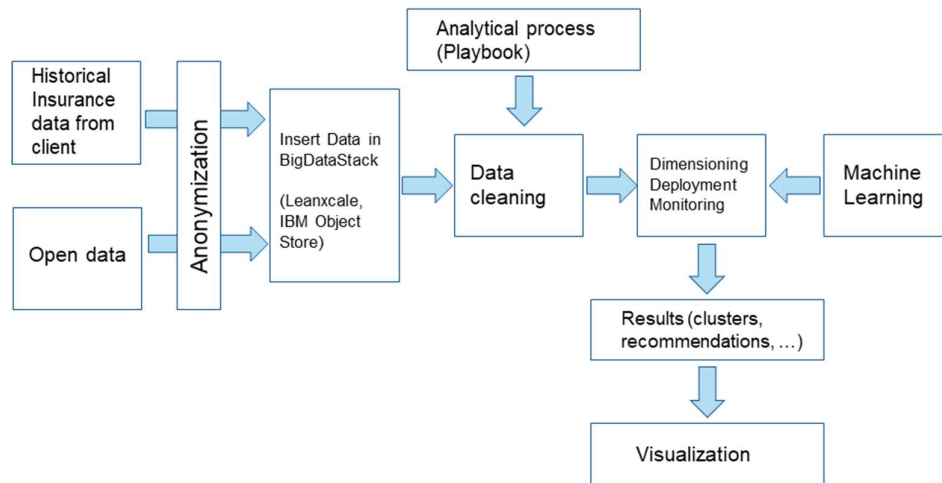


Figure 5 – Architecture Schema of the SI use case

In addition, a set of application services provide the communication between the external applications to the recommender systems.

5.3. Datasets

The datasets provided by the Insurance Company (customer of GFT) are described in the following in terms of tables and records structure and description.

Following the GDPR directive, all sensitive information of the datasets have been anonymized. For the encryption, we used a cryptographic hash function, the MD5 algorithm. It is a unidirectional function different from coding and encryption because it is irreversible. The spread of this encryption algorithm is still widespread (just think that the most frequent integrity check on file is based on MD5). This function takes as input an arbitrary length string and outputs another 128 bit output. The process happens very quickly and the output (also known as "MD5 Checksum" or "MD5 Hash") returned is such that it is highly unlikely to obtain the same hash value in output with two different input strings.

We have modeled the length of the encrypted string, based on the length of the field to be encrypted. For example, for the tax code the encrypted string is 16 characters, while for the license plate it is 8 characters. This eliminates the possibility of tracing back to the initial value. We have performed several decrypting tests present on numerous online sites and no one has been able to decrypt the string entered.

Furthermore, we have carried out a univocal check of all the encrypted keys, so that the possibility of two different string yielding identical encrypted strings is excluded.

In the following, the datasets tables and records are described. The fields highlighted in blue have been anonymized as explained above.

ana

id_univoco_anagrafica	string	Flow unique identifier: REGISTRY
id_univoco_master	string	
codice_fiscale	string	Subject unique identifier
tipo_anagrafica	string	Registry type (P = person, N = company)
cognome	string	Surname / company name
nome	string	Name
Sesso	string	Gender (M=male, F=female, N=company)
pubblica_amministrazione	string	Public Administration (YES/NO)

ana_ptf

codice_fiscale	string	Subject unique identifier
idpolizza	string	Policy unique identifier
ruolo	string	Subject role
cognome	string	Surname / company name
nome	string	Name

ana_sin

id_univoco_anagrafica	string	Flow unique identifier: REGISTRY
id_univoco_master	string	
codice_fiscale	string	Subject unique identifier
idsinistro	string	Claim unique identifier
ruolo	string	Subject role
cognome	string	Surname / company name
nome	string	Name

ana_vei

codice_fiscale	string	Subject unique identifier
targa	string	License plate
cognome	string	Surname / company name
nome	string	Name

anaage

codice_fiscale	string	Subject unique identifier
agenzia	string	Agency ID
descrizione	string	Description

anaaia

codice_fiscale	string	Subject unique identifier
codice_anomalia	string	Anomaly identifier

anabds

codice_fiscale	string	Subject unique identifier
bds	bigint	
p1	bigint	
p2	bigint	
p3	bigint	
p4	bigint	
p5	bigint	
p6	bigint	

anacci

codice_fiscale	string	Subject unique identifier
tipo_assicurazione	string	Insurance type

ente_comunicante	string	Communicating entity
data_infortunio	string	Accident date
luogo_infortunio	string	Accident place
lesione_1	string	Injury nr 1
lesione_2	string	Injury nr 2
lesione_3	string	Injury nr 3
lesioni_ulteriori	string	Other Injuries
percentuale_inabilita	double	Disability percentage
data_decesso	string	Date of death

anacnt

codice_fiscale	string	Subject unique identifier
tipo_contatto	string	Contact type
contatto	string	Contact

anacontatori

codice_fiscale	string	Subject unique identifier
portafoglio	bigint	Total insurance policies number
portafoglio_auto	bigint	Auto insurance policies number
portafoglio_re	bigint	Elementary branches insurance policies number
portafoglio_vita	bigint	Life insurance policies number
portafoglio_cauzioni	bigint	Deposits policies number
sinistri_aperti	bigint	Open claims number
veicoli_attivi	bigint	Insured vehicles number

anafid

codice_fiscale	string	Subject unique identifier
tipo_soggetto	string	Subject type

anaind

codice_fiscale	string	Subject unique identifier
comune	string	Subject main address, city
provincia	string	Subject main address, province
nazione	string	Subject main address, country
flag_principale	string	

analnkcnt

tipo_contatto	string	Contact type
contatto	string	Contact
codice_fiscale_a	string	Subject unique identifier a
codice_fiscale_b	string	Subject unique identifier b

crvdlnk

partita_iva	string	VAT number
codice_fiscale	string	Subject unique identifier
denominazione	string	Subject / company name
cognome	string	Surname / company name
nome	string	Name

crvdsem

codice_fiscale	string	Subject unique identifier
semaforo	string	Traffic light

ptf

idpolizza	string	Policy unique identifier
agenzia	string	Agency ID
descrizione_agenzia	string	Agency description

provincia_agenzia	string	Province of the agency
ramo	string	Policy branch
tipo_polizza	string	Policy type (Individual / Collective)
stato_polizza	string	Policy state (Active/ Canceled / Suspended)
stato_coass	string	No coinsurance / Our delegation / Delegation
codice_prodotto	string	Product Code-Product Description
prodotto	string	Product
data_effetto	string	Policy effective date
data_scadenza	string	Policy effective deadline
premio	double	Policy premium

sin

idsinistro	string	Claim unique identifier
idpolizza	string	Policy unique identifier
data_sinistro	string	Claim occurrence date (Format: YYYY-MM-DD)
ora_sinistro	string	Claim occurrence time (Format: HH: MM)
tipo_sinistro	string	Accident type (RCA / ARD / RE)
tipo_danno	string	Damage reported type (1 = THINGS / 2 = PEOPLE / 3 = MIXED)
tipo_gestione	string	Claim management type
flag_autorita_presenti	string	Authority flag present (S - Yes, N - No)
stato_sinistro	string	Accident status
data_definizione_sinistro	string	Claim closing date (Format: YYYY-MM-DD)
numero_veicoli	bigint	Vehicles involved number
comune	string	Claim occurrence address, city
provincia	string	Claim occurrence address, province
pagato	double	Paid
riservato	double	Reserved
data_denuncia	string	Claim complaint date (YYYY-MM-DD)

sinantifrode

idsinistro	string	Claim unique identifier
semaforo	string	Traffic light
verifica	string	Verification
note_verifica	string	Verification notes
approfondimento	string	Deepening
note_approfondimento	string	Deepening notes
antifrode	string	Anti fraud

sinantifrodectl

idsinistro	string	Claim unique identifier
controllo	string	Check

vei

targa	string	License plate
marca	string	Vehicle brand
modello	string	vehicle model
tipo_veicolo	string	Vehicle type
tipo_targa	string	License plate type
data_immatricolazione	string	Matriculation date

vei_ptf

targa	string	Vehicle identifier
idpolizza	string	Policy unique identifier

vei_sin

targa	string	Vehicle identifier
idsinistro	string	Claim unique identifier

5.4. Use Case Scenarios

5.4.1. Activity 1: Data acquisition

The first activity is to define the data source in order to obtain the required datasets. The process is described in the following table.

Step	Description
SI-A1-01	The Data Owner picks the data source type from a drop down list, e.g., database, FTP server etc. If the selected type is database then the RDBMS should be defined, again from a drop-down list, e.g., postgres, SQL-server etc.
SI-A1-02	The Data Owner enters in plain text format the required connection attributes, i.e., IP address, username, password, etc.
SI-A1-03	The system validates the connection.
SI-A1-04	The Data Owner defines what particular resource is to be accessed. For example, if the selected datasource type is database, selects from a drop down list the preferred tables. If the source is an FTP server, the exact path should be defined.
SI-A1-05	The system informs the user that the requested data sources are successfully defined.

5.4.2. Activity 2: Analytics definition

This activity concerns the definition of the analytic flow of the recommender system in BigDataStack. The main actors are the business analyst (BA) who defines the high-level analytic tasks needed for the recommender and the data scientist (DS) who will map the high-level tasks to concrete analytic tasks.

Step	Description
SI-A2-01	From the Process Modelling Framework, the BA sets the global business goals of the system and builds the graph that represents the application. Only analytic tasks are added in the graph. Once the graph is ready and the business goals are saved, the work made by the BA in the Process Modelling Framework is available for the DA in the Data Toolkit.
SI-A2-02	From the Data Toolkit, the DS opens the graph and concretizes the analytic tasks, by linking the high-level analytic tasks defined by the BA to concrete executables. Then she/he sets the SLOs for each of the executables and finally sets values for parameters of recommendation algorithm.

5.4.3. Activity 3: Deployment of the application services

This activity is related to the deployment of the application specific services of the recommender system. The user will deploy his application in the Application Dimensioning Workbench (ADW), by selecting the QoS and the resources that are expected to be needed for each of the application services. Based on these input parameters, BigDataStack suggests to the user the deployment configuration that suits best to the application.

The main actors are the application engineer (AE), the data scientist (DS) and the business analyst (BA).

Step	Description
SI-A3-01	The BA sets the business goals of the system.
SI-A3-02	The DS needs to deploy the application services of the recommender system he has been experimenting with. The DS calculates the SLOs for the application services and then provides the numbers to the AE with the help of the data toolkit.
SI-A3-03	The AE estimates the resources (CPUs, Memory and number of replicas) needed for each application service and then registers the SLO's and the estimation in ADW. BigDataStack provides the AE the a priori best deployment pattern. The AE confirms the first deployment of the application in the ADW of BigDataStack.

5.4.4. Activity 4: Display recommendations

This activity is related to the retrieval and displaying of the recommendations calculated by the system. It demonstrates the capability of the adaptable visualizations component of BigDataStack to display the outcomes produced by the recommender system. The main actor is the data scientist (DS).

Step	Description
SI-A4-1	The DS is experimenting with the recommender system and needs to display the outcomes calculated for the system for one or more users. She/he connects to the adaptable visualization component and enters the user whose recommendations wants to analyse. BigDataStack provides her/him with information about the recommendations calculated for the users.

5.4.5. Activity 5: Provide recommendations

This activity concerns the capability of the recommender system to provide the recommendations calculated by the system to external systems. The data storage for the recommender system is the LeanXcale SQL data store provided by BigDataStack. The main actor is a user of the insurance company (IC), who accesses the specific web application foreseen for the implementation of the use case, which supports her/him for the definition of tailored offerings to customers.

Step	Description
SI-A5-01	IC user accesses the specific web application in order to visualize the report for each of her/his associated clients. The external web application requests from the recommender system to provide the recommendations for the clients associated to the IC user.
SI-A5-02	The recommender retrieves the data from the BigDataStack data store to provide the external web application the calculated recommendations.
SI-A5-03	The external web application selects the tailored offerings and then displays them to the IC user in order to allow her/him to define the customer strategies.

5.5. Next steps

The use case activities will continue with the full implementation of the scenarios as well as the integration with the BigDataStack platform. In addition to corporate datasets (Client registry, Portfolio, Vehicles, Claims, etc., adequately anonymised), open data will be taken into consideration, including hydrogeological risk data, earthquake risk data, crime data, in order to improve the algorithms and allow a better customers' segmentation strategies definition.

6. Conclusions

This document presents the three use cases of the BigDataStack project: Real-time Shipping Management, Connected Consumer and Smart Insurance. All use cases have been deployed on BigDataStack infrastructure and the corresponding datasets have been ported to the BigDataStack storage engine (LeanXcale database and Object store). The applications and their data / analytics services are using BigDataStack offerings to achieve the corresponding identified business and technical goals for each use case.

Moreover, for each of the three pilots, this deliverable has provided information on their description and scenarios. In terms of next steps, the use case activities will continue with the full implementation of the scenarios as well as the integration with the BigDataStack platform. For example, RSM activity will include the required steps to re-route the vessel if an order for a spare part is placed, while in the CC scenario, a clustering algorithm for segmentation of customers will be introduced, as well as the full integration with the end-user tools of BigDataStack such as Process Modelling and Data Toolkit. The SI pilot will complete the scenario' implementation and allow the customer's value calculation.

Where required, additional datasets will be included in the analysis of the use cases (for example, open data in the SI pilot). Also, use cases evaluation will be fully addressed, as well as the BigDataStack Usability & Performance assessment, based on the available overall integrated prototype of the BigDataStack environment.