IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 9, 2019, accepted December 21, 2019, date of publication December 31, 2019, date of current version January 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963283

Seaport Data Space for Improving Logistic Maritime Operations

DAVID SARABIA-JÁCOME[1], CARLOS E. PALAU[1], (Senior Member, IEEE), MANUEL ESTEVE[1], AND FERNANDO BORONAT[2], (Senior Member, IEEE)
[1]Communication Department, Universitat Politècnica de València, 46022 Valencia, Spain
[2]Communication Department, Universitat Politècnica de València at Campus Gandia, 46730 Gandia, Spain

Corresponding author: David Sarabia-Jácome (dasaja@teleco.upv.es)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme through the PIXEL Port Project under Grant 769355, and in part by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), Ecuador.

ABSTRACT The maritime industry expects several improvements to efficiently manage the operation processes by introducing Industry 4.0 enabling technologies. Seaports are the most critical point in the maritime logistics chain because of its multimodal and complex nature. Consequently, coordinated communication among any seaport stakeholders is vital to improving their operations. Currently, Electronic Data Interchange (EDI) and Port Community Systems (PCS), as primary enablers of digital seaports, have demonstrated their limitations to interchange information on time, accurately, efficiently, and securely, causing high operation costs, low resource management, and low performance. For these reasons, this contribution presents the Seaport Data Space (SDS) based on the Industrial Data Space (IDS) reference architecture model to enable a secure data sharing space and promote an intelligent transport multimodal terminal. Each seaport stakeholders implements the IDS connector to take part in the SDS and share their data. On top of SDS, a Big Data architecture is integrated to manage the massive data shared in the SDS and extract useful information to improve the decision-making. The architecture has been evaluated by enabling a port authority and a container terminal to share its data with a shipping company. As a result, several Key Performance Indicators (KPIs) have been developed by using the Big Data architecture functionalities. The KPIs have been shown in a dashboard to allow easy interpretability of results for planning vessel operations. The SDS environment may improve the communication between stakeholders by reducing the transaction costs, enhancing the quality of information, and exhibiting effectiveness.

INDEX TERMS Analytics, big data, industry 4.0, industrial data spaces, Internet of Things, maritime, seaport, intelligent transport.

I. INTRODUCTION

The rapid growth of new technologies is leading the industry to the fourth industrial revolution, named Industry 4.0 [1]. This concept refers to the digitalization and optimization of industrial processes through the use of emerging technology enablers such as the Internet of Things (IoT), Cloud Computing, Big Data, or Artificial Intelligence [2], [3]. Although the concept of Industry 4.0 has been present for some years, only about 48% of manufacturing companies declared that they are ready to face technological changes supported by such building blocks [4]. The Industry 4.0 technologies adoption gap is caused by the existing barriers encountered during the enabling of industrial environments 4.0 [4].

The maritime industry is one of the transportation and logistics industries with the most significant economic impact on world trade. Maritime seaports support about 80% of the world trade [5]. Each year the traffic that seaports support increases by 1.4% [6]. Seaports must be able to adapt to this constant growth in an efficient manner, minimizing unproductive operations. Industry 4.0 enablers can transform the seaports into smart seaports capable of optimizing their processes to support the expected growth of traffic in the coming years.

Seaports are complex intermodal terminals where several stakeholders are involved. The synergy between them is of

The associate editor coordinating the review of this manuscript and approving it for publication was Dalin Zhang.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

vital importance for the efficient management of resources and optimization of stakeholders' processes. The coordinated interaction between stakeholders might bring several benefits, such as reliability, timeliness, safety, lower transaction costs, and lower operational costs [7]. However, enabling such coordinated interaction is challenging because each of the participants involved in the distribution chain uses heterogeneous systems. Heterogeneous data sharing is one of the challenges with greater difficulty than the industrial environments 4.0 have to face [3]. Currently, the stakeholders use systems based on Electronic Data Interchange (EDI) to exchange information under the same data format, but this approach has shown drawbacks such as incorrect, double, and out of time information exchange [7]. Another problem associated with enabling coordinated communication is privacy and security. The stakeholders are reluctant to share their data to improve the maritime processes since the data is one of the most critical assets [8]. Thus, seaports require a secure-by-design environment, overcoming the limitations of current information exchange systems to become an intermodal intelligent transport terminal.

The Industrial Data Space (IDS) initiative emerges as a reference architecture model to solve the problems of heterogeneous data sharing, considering the data sovereignty, privacy, and traceability [9]. This model developed by the Industrial Data Space Association (IDSA) is in the process of being standardized by the German Institute for Standardization (DIN). The main objective of IDS is to enable a trusted virtual data space to support the secure exchange and linkage of data in business ecosystems. IDS architecture has been used in several industrial cases successfully. IDS architecture is a novelty as it has not been implemented in the maritime industry yet.

This work presents the Seaport Data Space (SDS) based on the IDS architecture to solve the problem of data interoperability and associated interoperation among stakeholders in a seaport to lead to the promotion of the smart seaport concept. SDS enables a secure virtual environment for sharing data in a Seaport environment. Additionally, this work presents a Big Data architecture to provide scalability and reliability to support the massive data shared in the SDS. The SDS was evaluated using three stakeholders: (i) a port authority; (ii) a container terminal operator, and (iii) a shipping company. Each stakeholder implemented an IDS connector based on the Fiware IoT platform [10] to interconnect to each other in the SDS. The port authority shared data related to the vessel position in real-time, while the container terminal operator historical load/unload berth operations. The shipping company implemented the Big Data architecture to manage the massive shared data and exploit it to extract useful information. The Big Data architecture used the flow-based system Apache NiFi [11] for pulling data from the IDS connector and pushing them to the Big Data modules. The Big Data modules were implemented using Big Data open-source frameworks and systems such as Apache Spark [12], Apache Kafka [13], and among others. The Big Data architecture facilitated the development of relevant Key Performance Indicators (KPIs) about vessels' fuel consumption, time at berth and anchorage, and about container terminal occupancy. These KPIs might be useful to improve operational planning of a shipping company fleet, and consequently, seaport operations.

In summary, the main contributions and novelties of the proposed work are:

- A SDS where seaport stakeholders can share and track data to overcome the information exchanging issues with ownership, interoperability, privacy, and security guarantees.
- A Big Data Architecture which is integrated with the IDS architecture to handle the massive data shared and extract useful information to improve making decisions.
- Several KPIs that are extracted from the massive data shared in SDS to improve planning operations.

The remainder of the paper is structured as follows. Section 2 reviews the current literature concerning this field of research. Section 3 presents the SDS architecture and the Big Data architecture overview, as well as implementation and the integration process details. Section 4 presents the Big Data analytics results and KPIs by using the Big Data Architecture in the SDS scenario. Finally, Section 5 presents conclusions and future work.

II. RELATED WORKS AND MOTIVATION

The fundamental pillars of smart seaports are the automation of operations and seaport equipment, and the interconnection of the participants involved in the seaport logistics chain [14], [15]. Cyber-physical systems (CPSs) are being used to enable the automation of seaport equipment. These systems are able to connect physical devices with the virtual world. Currently, the Reference Architectural Model Industry (RAMI) 4.0 and the Industrial Internet Reference Architecture (IIRA) leads the implementation of the CPSs [16]. Meanwhile, IoT allows the interconnection of any seaport equipment to the Internet. IoT as a smart seaport enabler is being used within important European seaports such as the Seaport of Valencia, Hamburg, Rotterdam, among others, demonstrating its effectiveness [17]–[20].

Nowadays, the maritime environment employs EDI-based systems for the information exchange between subsystems in charge of container tracking, rail management, and inland navigation, and between partners in the supply chain [21], [22]. These systems allow vertical cooperation between stakeholders. Moreover, Port Community Systems (PCS) have been created to reduce the complexity of the information interchange between the stakeholders in the seaport operations [23]. The PCS are systems that centralize the vessels' information and the goods they transport so that the stakeholders can better control and coordinate the movements of goods [24]. Also, the Port Collaboration for Decision Making (PortCDM) platform proposed by the Sea Traffic Management (STM) aims to serve as an integral point of transport information systems to encourage cooperation among them and allows the intelligent management of maritime traffic [25]. The PortCDM provides information about the cargo arrival and delay, and loading-unloading process in the operations terminal to facilitate making decisions. The current information exchange systems based on EDI and PCS are not sufficient to solve problems in cooperative communication. Mainly, these systems do not exchange information on time, accurately, and efficiently [7]. The information exchange is vital to improve the quality of transport.

Unlike the information interchange model, data sharing involves both vertical and horizontal collaboration between companies. A data market might be created to encourage collaboration among competitors to achieve a common goal using the data-sharing approach. The IDS architecture proposes a secure environment to ease the data sharing between companies involved in the production and distribution of a product [9]. The IDS reference architecture was developed to meet the industrial needs of trust, security, data sovereignty, data ecosystem, standardized interoperability, value-added applications, and a data market. The IDS reference architecture is composed of five layers: business, functional, information, process, and system layer. The business layer defines the specific roles to enable data exchange, Fig.1. The functional layer describes the characteristics of trust, security, data ecosystem, interoperability, value-added applications, and data market. The process layer specifies the interactions between the components of the architecture. These components are grouped into sub-processes that are responsible for accessing the data space, exchanging data, publishing, and using applications. The information layer specifies the information model to facilitate compatibility and interoperability. The system layer describes the specific roles of the business layer to cover the functional requirements. This layer defines the connector, the broker, the identity provider, and the application store [26].

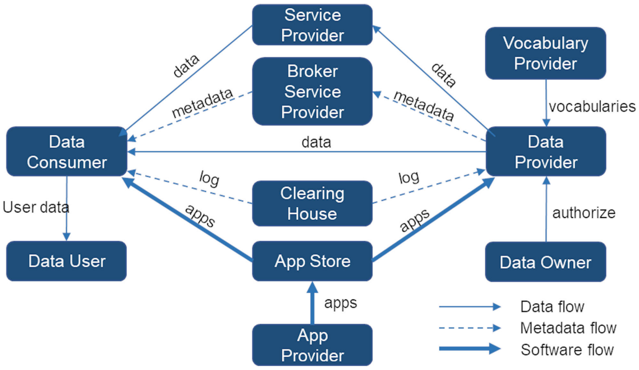


FIGURE 1. Roles interaction in the IDS Architecture.

Since the IDS architecture does not define the interfaces to be used nor provide details for implementation, the interaction between the academia and industry provides relevant information through implementation cases. For example, some research has provided relevant information about components implementation [27], security implementation [28], and the ontology-based information model [29]. On the other hand, the industry has implemented the IDS architecture successfully in logistics cases: to optimize the loading and unloading times of trucks [30] and to predict railway-tracks maintenance [31]. In the case of the maritime industrial sector, the SINTEF Ocean institute analyzed the use of the IDS architecture to support the Maritime Data Space (MDS) [32]. They stated that the obstacles to enable the MDS are the connection of vessels with the IDS, the shipping system complexity, and the international nature of shipping. Even though IDSA encourages the industry to use the IDS architecture in industrial environments, the implementation of the IDS architecture model in the Seaport case has not been implemented yet.

Recently, the Boost 4.0 project was released to design and implement Big Data middleware for IDS support. The main project motivation is to fill the gap between IDS architecture and Big Data management [33]. The project is planning to publish its results by the end of 2021. Also, few Big Data architectures were proposed in the current literature for the maritime industry [34], [35]. The primary approach used by these architectures was to employ the Lambda processing architecture, which has proven to be efficient in meeting the requirements of scalability, efficiency, and high availability [36]. However, these architectures did not consider IoT requirements for Big Data management or the use of the Big Data life cycle model for their designs. The International Telecommunication Union (ITU) has released a bunch of recommendations (Y. 2066 [37] and Y4114 [38]) to be considered in the design of the Big Data architecture for IoT. Also, the Big Data life-cycle model (BDLM) proposed by Demchenko *et al.* [39] provides essential advantages to the data re-usability at any life cycle stage and the massive reduction of the data at an initial stage. Big Data architecture is fundamental to extract relevant information from shared data to improve seaport operations.

There are several models, considered as state of the art, used to estimate some vessel operations process. For example, these models are intended to estimate fuel consumption and pollution generated by vessels [5], [40], [41]. However, the problem appears when the models are applied to large datasets without the support of adequate processing infrastructure. The models need to be adapted so that they can be efficiently exploited by the resources used by the Big Data architecture [42]. The lack of know-how to implement these algorithms in a Big Data architecture is limiting the efficient exploitation of the shared data to improve the operation in the seaport.

Unlike related works, the main motivation of this work is facilitating the coordinated communication between stakeholders in a multimodal seaport terminal thought the IDS reference architecture. Also, this work fills the gap between IDS reference architecture and Big Data management by providing a Big Data architecture implementation details and know-how. Finally, this work proposes vessel operations algorithms based on Big Data techniques to improve seaport operations.

III. VALENCIA SEAPORT DATA SPACE

This section presents the design and implementation of the SDS, the Big Data architecture implementation details, and the Big Data architecture and SDS integration. The proposal is applied to the Valencia-Spain Seaport.

A. REQUIREMENTS

Valencia seaport is considered one of the most important ports on the Mediterranean coast. This seaport supports more than 4.7 million Twenty-foot Equivalent Units (TEUs) per year [43]. Recently, traffic has shown a growth of 1.77 million TEUs, which has affected the seaport operations efficiency [15]. Valencia seaport requires strategies that would allow it to optimize seaport operations and exploit its resources efficiently.

The transformation of Valencia seaport into a smart seaport requires solving the problems that appear in the data sharing process between stakeholders to improve decision making.

The stakeholders involved in the maritime port's operations are the port authority, terminal operators, shipping companies, truck companies, railway operators, seaport equipment maintenance companies, and cold container maintenance companies. This work studies the integration of the port authority, a container terminal operator, and a shipping company to demonstrate the feasibility of the SDS. Fig. 2 shows an overview of the Valencia Seaport case.

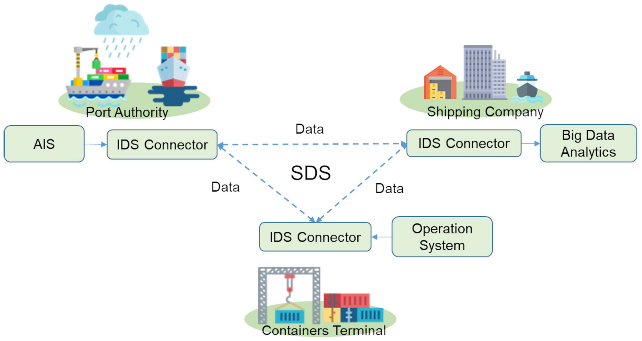


FIGURE 2. Valencia SDS case overview.

The container terminal operator is responsible for loading and unloading containers from vessels, trains, and trucks. In this case, the container terminal operator has a system based on Structured Query Language (SQL) to record the operations of loading and unloading performances. The shared data are structured using the JavaScript Object Notation (JSON) data format.

The port authority controls the arrivals and departures of vessels, trains, and trucks to and from the seaport. In this case, the port authority keeps track of the vessels that are near the port through the Automatic Identification System (AIS). The AIS provides information about the vessel's navigation state.

B. SDS ARCHITECTURE OVERVIEW

The main objective of this case is to design a secure Big Data sharing environment among seaport stakeholders based on the IDS reference architecture. The SDS architecture overview presents the details about the SDS architecture components, systems adapters, data models, and the sharing process. Fig. 3 shows the high-level SDS architecture.

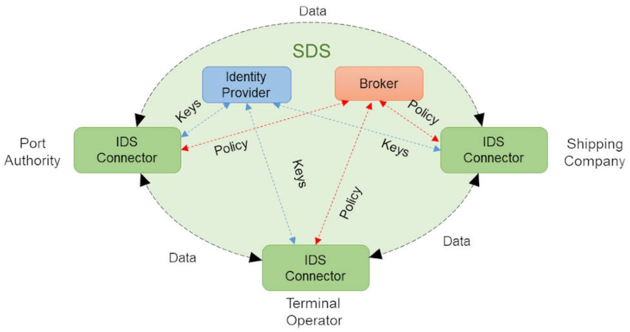


FIGURE 3. SDS architecture components.

1) SDS ARCHITECTURE COMPONENTS

The SDS architecture is composed of IDS connectors, an identity provider, and an IDS broker.

The IDS connectors share data, ensure data sovereignty, and keep the interoperability between systems [9]. This connector uses a publish/subscribe mechanism to share data, a proxy Policy Enforcement Point (PEP) to ensure data sovereignty, and an information model to keep the same data model and format. The IDS connector functionalities are implemented using the Fiware IoT platform Generic Enablers (GEs) Orion Context Broker, and Wilma. Each stakeholder implements an IDS connector in their technological infrastructure to connect to the SDS and share data. Fig.4 shows the structure of the IDS connector for this case. The GE Orion Context Broker provides a publish/subscribe mechanism to receive entities context updates. To do so, the Orion Context Broker uses the NGSI9 and NGSI10 interfaces to send information about the context data

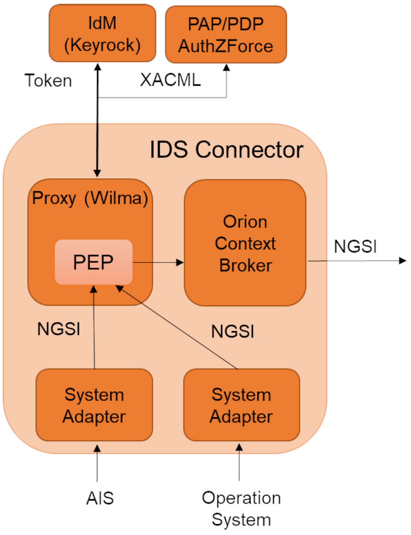


FIGURE 4. IDS connector.

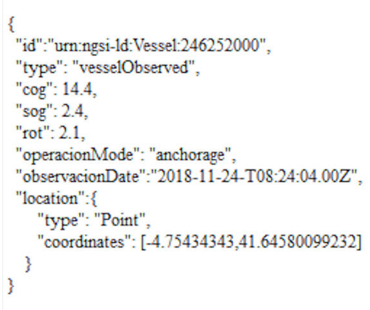
and send context data. On the other hand, the GE Wilma provides PEP-Proxy functionalities to keep control of the data.

The identity provider keeps the information about the IDS connectors of the SDS and validates the connectors' identity [9]. These functionalities are implemented using the GE Identity Manager-Keyrock (IdM). The IdM uses the Oauth2 protocol to allow connectors authentication. Also, the IdM keeps a record of its service through logs.

The IDS broker keeps the information about the data sources, data models, and usage policies [9]. These functionalities are implemented using the GE Policy Decision Point/Policy Administration Point (PDP/PAP) AuthZForce GE. AuthZForce uses the eXtensible Access Control Markup Language (XACML) to allow the definition of fine-grained policies.

2) DATA MODELS

Data models abstract the stakeholders' systems to organize the data into the Fiware IoT platform. Vessels are modeled through the entity vesselObserved in the case of the AIS system. This entity represents a vessel with its characteristics. These characteristics are the maritime mobile service identity (MMSI), position (latitude and longitude), course over ground (COG), speed over ground (SOG), rate of turn (ROT), observation date, and operation mode. The identity created in the Fiware based IDS connector is updated with every AIS vessel message. Fig. 5 shows a vesselObserved JSON example.



```
{
 "id":"urn:ngsi-ld:Vessel:246252000",
 "type": "vesselObserved",
 "cog": 14.4,
 "sog": 2.4,
 "rot": 2.1,
 "operacionMode": "anchorage",
 "observacionDate":"2018-11-24-T08:24:04.00Z",
 "location":{
    "type": "Point",
    "coordinates": [-4.75434343,41.64580099232]
  }
}
```

FIGURE 5. JSON schema of *vesselObserved* entity.

Meanwhile, the seaport berths are modeled through the entity berth in the case of the container terminal operating system. this entity represents a container terminal berth. These characteristics are the initial and final operational dates.

3) SYSTEMS ADAPTERS

The systems adapters interconnected the stakeholders' Application Programming Interfaces (APIs). This connection is developed using the Node-RED platform [44]. This platform facilitates the development of data flows through its web user interface. The flows use nodes that are capable of making data transformations.

The AIS system flow queries the AIS HTTP server, converts the AIS message format (NMEA standard) to JSON format, assemblies the JSON based on the data model entity, adds the Fiware JSON headers and sends the data to the IDS connector. Fig. 6 shows the AIS system Node-RED flow.

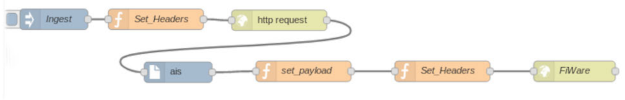


FIGURE 6. AIS system adapter node-RED flow.

4) SHARING PROCESS

The process of data sharing between Fiware-based IDS connectors exploits the federation functionality of the Orion Context Broker. The federation push mode allows the sending of context notifications between two Orion Context Brokers. After enabling the Orion Context Broker federation, the sharing process requires a subscription notification that contains the entities' id and the uniform resource locator (URL) of the other IDS connector that is going to receive the entities' data.

C. BIG DATA ARCHITECTURE OVERVIEW

The architecture is based on the Lambda processing architecture [36] and the BDLM [39]. Also, the Big Data architecture design considers the ITU recommendations Y. 2066 [37] and Y4114 [38]. The Big Data architecture is composed of several modules that can be adapted depending on the needs of each SDS member. The different Big Data architecture modules are implemented using open source platforms for Big Data management. In the case studied, the Big Data architecture is implemented in the shipping company technological infrastructure to exploit the data, Fig.7. Next, the functionalities, technologies, and platforms used for the implementation are described:

- **Integration module:** is in charge of facilitating the connection between the IDS connector and the Big Data architecture. This module exploits the pull/push mechanisms employed by the IDS connector to collect the data. This module is implemented using the flow automation system, Apache NiFi [11]. The selection of Apache NiFi responds to its ability to design flows visually through its user interface and as a highly scalable, configurable, and secure tool. Apache NiFi provides several processors capable of performing specific operations over the data flow. The primary operations to be carried out in this module are the connection with the IDS connector and the conversion of the data format JSON to the Parquet format. The Parquet format is a high-performance format [45]. The historical data repository receives the data resulting from the converting format tasks. In the case of real-time data, they are sent to the data processing platform in real-time through Apache Kafka [13].

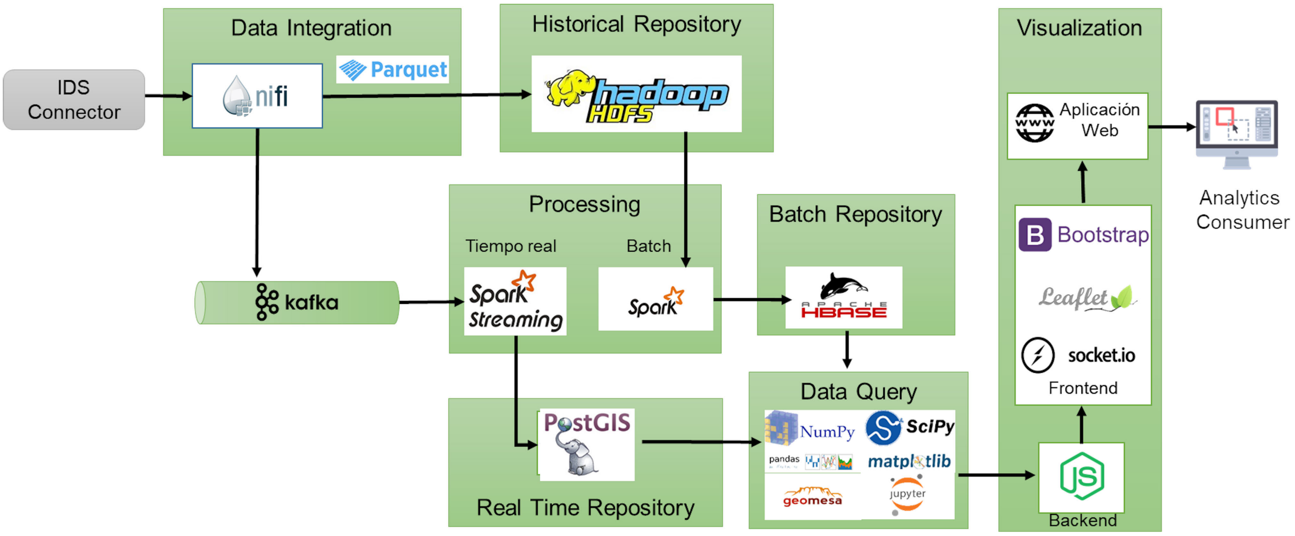


FIGURE 7. Big data architecture implementation.

- **Historical Database:** stores all the data that come from the IDS connector. This module is implemented using the Hadoop Distributed File System (HDFS). HDFS is widely used in the current literature in Big Data ecosystems because it provides high availability, reliability, and scalability [46]. The HDFS cluster consists of a Name Node and several Data Nodes. The Data Nodes are intended for data storage in 64 megabytes data blocks, while the Name Node manages the location of the files and their replicas in the Data Nodes.
- **Processing module:** provides batch and real-time data processing. The data processing modules are implemented using the Apache Spark framework [12]. The selection of Apache Spark responds to its ability to execute jobs both in batch mode and in real-time through its APIs. Apache Spark temporarily stores the results of its operations in memory, so it has shown better performance than Hadoop, which stores its operations on disk [47]. The Spark Streaming API processes real-time data. This API allows the execution of applications almost in real-time. The applications are focused on performing operations over data in a time sliding window.
- **Batch Repository:** stores the batch processing results. This repository is implemented using the NoSQL Apache HBase database system [48]. This database system provides scalability and high availability.
- **Real-time Repository:** is in charge of storing the real-time processing results. This module is implemented using the Postgres relational database system and its PostGIS extension [49]. PostGIS is an efficient system to store geospatial data. The selection of PostGIS responds to the data reduction in the initial processing stages, which reduces the scaling problems overtime.
- **Query Manager module:** manages queries that are used to generate descriptive analytics. Mainly, the module exploits the platforms' API and functions to generate data queries. This module is implemented using the tools provided by the GeoMesa framework and the SparkSQL API [50]. GeoMesa allows the analysis of geospatial data through a set of tools that are integrated with processing frameworks such as Apache Spark and with a database system like HBase. GeoMesa provides a Spatio-temporal indexation to store data of point, line, polygon type in HBase. While SparkSQL allows structuring data in DataFrames for analysis using a language similar to SQL. Also, SparkSQL presents functionalities to perform a descriptive analysis of data (descriptive statistics) and to perform a data cleaning, aggregations, and filtering. These tools are used to extract useful information from the data.
- **Data Visualization module:** displays the results of Big Data Analytics to users. This module implements a graphical user interface (GUI) for the deployment of KPIs and diagrams. The GUI goal is to help operators infer the information extracted from the data. The GUI is a web application implemented using Bootstrap, NodeJS, Socketio, ChartJS, and Leaflet. The web application uses a backend and frontend structure to present the information to the user efficiently.

D. BIG DATA AND IDS ARCHITECTURE INTEGRATION

The integration of Big Data architecture and IDS architecture is developed by implementing a dataflow in Apache NiFi [11]. This dataflow is in charge of the data extraction from the IDS connector, the data transformation, and loading data to Big Data platforms. The dataflow is composed of 4 processors: ListenHTTP, PutParquet, PublishKafka, and LogAttribute. Fig. 8 shows the dataflow and the processors used for its implementation.

The ListenHTTP processor is in charge of receiving the Context Notification from the IDS connector. The ListenHTTP processor implements an HTTP server.

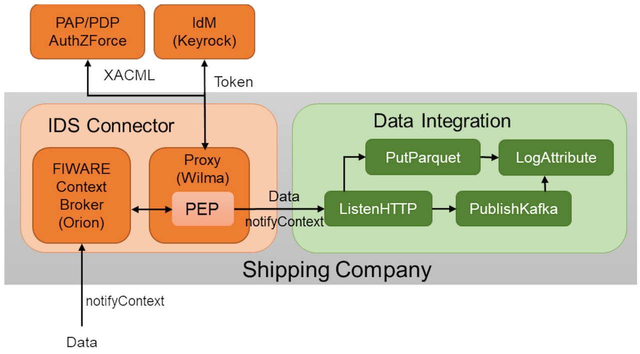


FIGURE 8. Apache NiFi DATAFLOW for integration.

This server listens for POST requests on a specific port. The port configured is port 8080 in this case. The IDS connector sends POST requests with context notifications in JSON format to the HTTP server. The ListenHTTP processor redirects these notifications to the PutParquet and PublishKafka processors.

The PutParquet processor is in charge of receiving the Context Notification from the ListenHTTP processor, converting to Parquet format, and storing it into the HDFS cluster. The PutParquet processor requires the data schema in Avro format to translate successfully to Parquet. The Avro format is a JSON format that describes the data types and protocols used in the definition of the data model. Also, the PutParquet processor requires the information of the HDFS cluster to store the converted data. The processor needs access to the configuration files of the HDFS cluster (coresite.xml and hdfs-site.xml) to know what the Node Name and Data Nodes servers are, and the configuration of the replication blocks. Another essential configuration parameter is the file tree path, where the data are loaded. Finally, the PutParquet processor loads the converted data into the HDFS file tree path.

The PublishKafka processor is in charge of receiving the Context Notification from the ListenHTTP processor and publishing it into the Kafka broker. The PublishKafka processor requires the Kafka broker URL and the topic as configuration parameters. The Apache Spark cluster receives the data by subscribing to the same topic to processing on-the-fly.

Finally, the LogAttribute processor completes the dataflow. This processor allows registering the status of each transaction of both the PutParquet processor and PublishKafka. This process facilitates the easy identification of errors that may occur during the dataflow operation.

IV. RESULTS

The results show the SDS feasibility and the use of Big Data architecture to extract useful information in planning the shipping company operations. For this, two datasets were used in the experimental evaluation. Table 1 describes the datasets used in this experiment. The data shared between the IDS connectors of the SDS are exploited to extract relevant KPIs for the planning of a shipping company operations. The KPIs are about the vessel's average occupation time in the containers loading and unloading process in the seaport container terminal, the container terminal occupancy, the waiting time of the shipping company vessels in the seaport anchorage zone and the vessel's fuel consumption estimation during its waiting time in the seaport anchorage zone. Several applications were developed in Apache Spark to load, process, and analyze the shared data for generating these KPIs.

TABLE 1. Seaport datasets details.

Dataset	Size	Period
AIS	10 GB	2016/01/01 - 2016/03/31
Terminal Operation	520 MB	2014/01/01 - 2019/01/31

A. VESSELS AVERAGE TIME OCCUPANCY

The vessels' occupation time in the container terminal is calculated using the container terminal operations dataset. The container terminal IDS connector shares the data about berths load/unload processes to the shipping company IDS connector. Subsequently, the HDFS repository stores the data shared in the IDS environment by following the data flow defined in the Big Data and IDS architecture integration subsection.

The loaded data from HDFS is structured in a DataFrame using a SparkSQL function. The features related to the berth' id, unloading process start timestamp, and loading process finish timestamp are selected from the first DataFrame to calculate the occupation time. Vessels occupancy on berths is performed by the difference between the unloading and loading timestamps. Next, the application calculates the average time at berth throughout an aggregation function by evaluating the new DataFrame in a week as frequency.

The vessels' maximum and minimum time occupancy in berth provides more information to make decisions. Since the dataset is a time-series data, the time-series decomposition into season and trend components is necessary to assess whether the maximum and minimum values vary over time. For this, the DataFrame obtained in the previous phase is decomposed using the Python Statsmodels library. Fig. 9 shows the decomposition into components of the DataFrame. The figure shows an incremental trend in time and a repeated season pattern every two months. As a result, vessels' average time is not the same in short periods (season), and it varies over time (trend). In the same way, the maximum and minimum values also vary over time, so the maximum and minimum calculation is made using the maximum and minimum average values per week. The average, maximum, and minimum occupation time values are used as KPIs to estimate the time that the shipping company fleet is going to be berthing in the seaport.

B. CONTAINER TERMINAL OCCUPANCY WEEKLY

The container terminal occupancy gives information about how many vessels the seaport terminal can support during

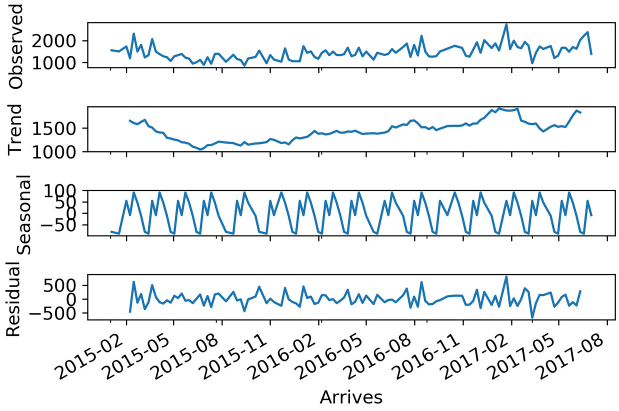


FIGURE 9. Terminal containers time-series decomposition.

the week. Similar to previous KPI, this is calculated using the dataset of the container terminal operations shared from the container terminal IDS to the shipping company IDS.

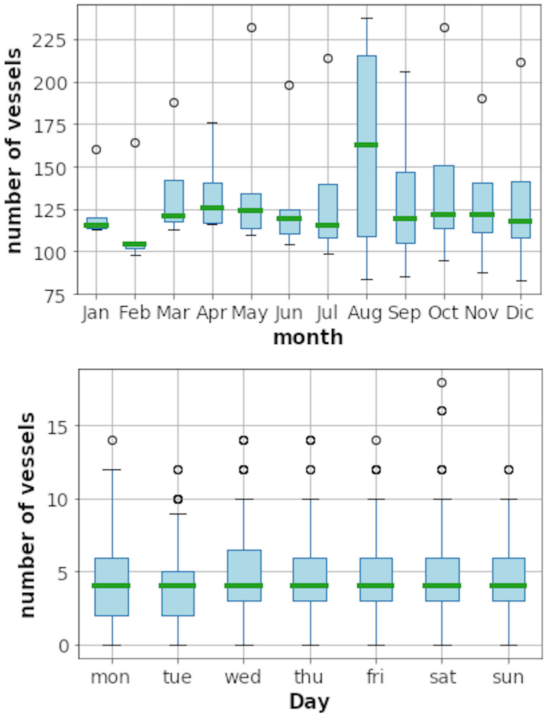
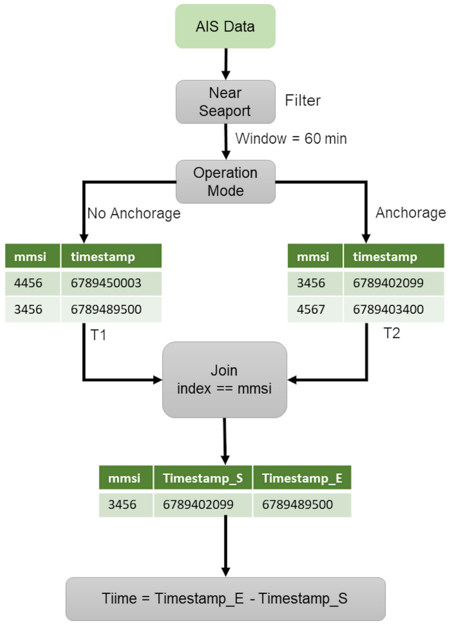


FIGURE 10. Box-whisker container terminal occupancy by day and month.

Unlike the previous KPI, the features related to the vessel' MMSI and the starting timestamp are selected from the first DataFrame to calculate berth's occupancy. In this case, the application aggregates the data by days, weeks, and months, and applies the count function on them. In this way, the application extracts information about the number of vessels per day, week, and month that are berthing in the container terminal. The result is summarized in a box-whisker diagram to represent the days and months most busy. Fig. 10 shows that Saturday and August are the day of the week and the month of the year most busy.



mmsi	timestamp
4456	6789450003
3456	6789489500

mmsi	timestamp
3456	6789402099
4567	6789403400

mmsi	Timestamp_S	Timestamp_E
3456	6789402099	6789489500

FIGURE 11. The algorithm used for calculating vessels berthing time.

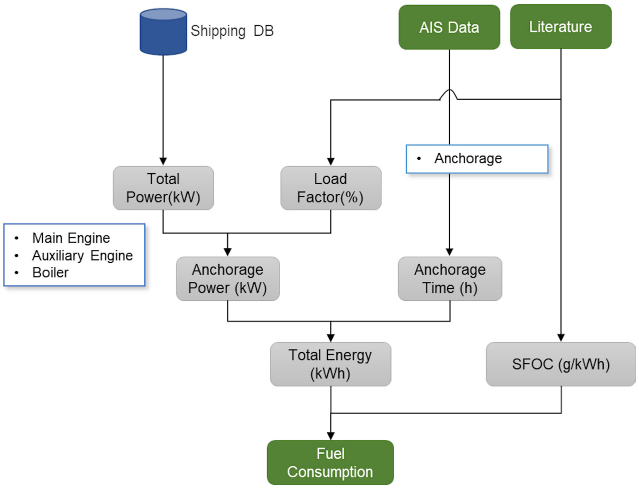


FIGURE 12. The algorithm used for estimating vessel fuel consumption.

The information obtained allows the shipping company to plan its operations in the days and months with less working load so that its vessels stay the least amount of time possible in the seaport.

C. AVERAGE TIME WAITING FOR A FREE TERMINAL

The vessels' anchorage time reduces the shipping company efficiency and produces a higher operational cost. The application estimates this time by using the AIS dataset. The port authority IDS connector shares AIS data (only messages from the company's fleet) to the shipping company IDS connector. Next, the shared data are stored in the HDFS repository and sent to the Apache Spark platform following the data flow defined in the Big Data and IDS architecture integration subsection. Fig. 11 shows the application data flow developed using Apache Spark Streaming and SparkSQL.

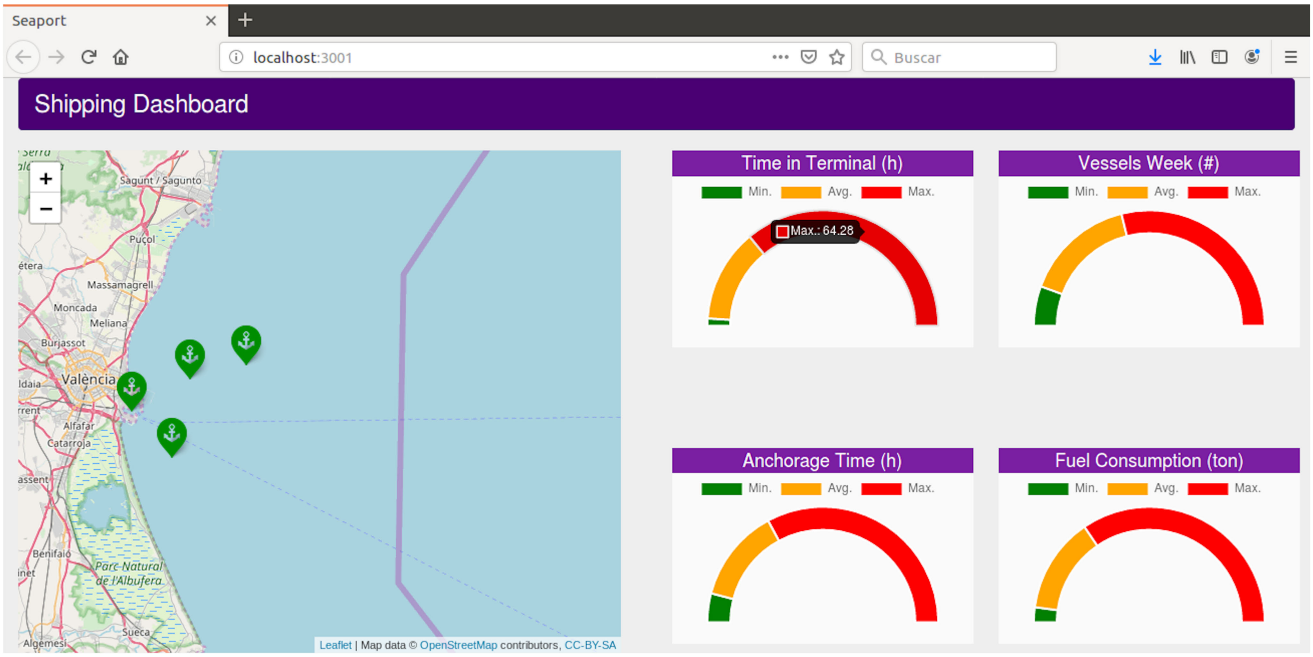


FIGURE 13. Shipping company Web GUI dashboard.

In this case, the application executes the SparkSQL functionalities on the real-time data through the Apache Spark Structures Streaming engine. This engine takes advantage of micro-batch processing to execute highly scalable, fast, and fault-tolerant queries. The AIS messages are filtered in real-time to select only the AIS messages whose vessels' positions are near the seaport. The application applies a filter based on a polygon of 4 geospatial points (latitude, longitude). Next, the filtered data get onto a 60-minutes-sliding window. This data is structured in a DataFrame by using the Structured Streaming engine. Then, the application splits the DataFrame into two DataFrames based on vessels' operation mode (anchorage and not anchorage). The new DataFrame contains the vessels' MMSI identification and timestamp. Next, the application searches the vessels that have changed the operation mode from the anchorage state. The application employs the MMSI identifier in the searching process. The join DataFrame function allows making a comparison of the DataFrames and selects the rows where the MMSI is the same in both DataFrames. Next, the application creates a new DataFrame with the selected rows. This new DataFrame contains the MMSI and the start and finish timestamp of the anchorage mode. The application updates the anchorage DataFrame by deleting the row with the MMSI founded in the searching process. Finally, the application calculates the time at anchor by mean of the difference between the timestamps (finish-start) and loads the results into a table in the PostGIS database. The web application queries the table and shows the average, the minimum, and maximum time at anchor per vessel as KPI. In this way, the shipping company can order to their vessels to reduce the speed before arriving at the seaport to save fuel.

D. VESSELS AVERAGE FUEL CONSUMPTION IN THE WAITING PERIOD

The fuel consumption estimation provides information about the tons of fuel occupied by the vessels when they are waiting for a berth. As the previous KPI, this is calculated using the AIS message dataset shared from the port authority IDS to the shipping company IDS. Fig. 12 shows the algorithm used based on the study presented in [40] and [5].

The fuel consumption estimation depends directly on the power used during the period evaluated. Vessels' power information is not available on the AIS messages, so it is estimated based on the motor's load factor during anchorage operating mode. The literature estimates that the load factor is 5% for the main engine (ME) and 50% for the auxiliary engine (AE) [41]. The total power is the result of the sum ME and AE powers during the anchorage operation mode. These results are stored on a table inside the HBase database. Next, the application multiplies the total power by the waiting time at the anchorage zone (previously calculated in the subsection C) by the base-specific consumption fuel and by a factor of 1.1, according to [40]. The base-specific fuel consumption has been estimated at 195g/kWh for vessels built since 2001 and at 205 g/kWh for vessels built between 1984 and 2000 [5]. Finally, the application stores the results in a table in the Postgres database. The web application queries the table and shows an average, minimum, and maximum fuel consumption during the vessels' waiting time at the seaport anchorage zone.

The web GUI dashboard groups the KPIs developed for a straightforward user interpretation. Fig. 13 shows the GUI with the developed KPIs. Also, the GUI has a map to show the vessel's position near the Valencia seaport.

V. CONCLUSION

In this paper, we have proposed the use of the IDS reference architecture to overcome the current limitations on seaport systems data sharing and facilitate the cooperative communication interoperability and interaction between stakeholders. Data sovereignty is the main advantage of using IDS architecture in industry 4.0, and specifically in transport and logistics. Since the IDS architecture does not consider the Big Data management, it has been proposed the integration of a Big Data architecture in our SDS environment proposal. The integration module facilitates the connection to the IDS connector to extract, clean, and load data into the Big Data platforms. The integration module functionalities were implemented using Apache NiFi. Apache NiFi proved to be useful on the integration due to its high capacity for designing data flows. The rest of the Big Data architecture modules provide features to store, process, and analyze the data shared in IDS. These functionalities were implemented using current Big Data open-source platforms and frameworks such as Hadoop, HDFS, Apache Spark, Apache Kafka, and HBase. The chosen platforms guaranteed efficient management of the data shared in the IDS architecture and provided a scalable and high available environment to develop and execute applications for massive data processing.

The feasibility of our proposal was validated by using several datasets related to vessel positions and terminal operations. The use of the Big Data architecture in SDS allowed the extraction of valuable information for the operations planning of a shipping company. The information was transformed into KPIs for a better interpretation of the data analysis results. The KPIs were compiled on a dashboard to improve the decision-making. Also, we adapted some state of art vessel operation models to be used in the Big Data Architecture.

The SDS improves the coordination between stakeholders by lower transaction costs. Also, the SDS allows the re-use of information by multiple parties and improving the quality of information. Finally, the data shared through SDS in time enhanced the vessel transit time and saved cost in the seaport operations. Although this paper has evaluated the proposed architecture in the maritime application domain, it is extensible and flexible to any industrial sector.

Moreover, the proposed Big Data architecture covers the IoT requirements proposed by the ITU-T so that it can be extended to application domains and cases involving industrial IoT devices. As future work, there will be further testing of the Big Data architecture in other application domains and cases to demonstrate its extensibility and adaptability. Also, more stakeholders and their dataset will be added to the SDS.

REFERENCES

[1] A. V. Bogoviz, "Industry 4.0 as a new vector of growth and development of knowledge economy," in *Industry 4.0: Industrial Revolution of the 21st Century*. Cham, Switzerland: Springer, Jul. 2018, pp. 85–91, doi: 10.1007%2F978-3-319-94310-7_8.

[2] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, doi: 10.1109/hicss.2016.488.

[3] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *J. Ind. Inf. Integr.*, vol. 6, pp. 1–10, Jun. 2017, doi: 10.1016/j.jii.2017.04.005.

[4] C. Baur and D. Wee, "Manufacturing's next act," *McKinsey Quart.*, Jun. 2015. Accessed: Nov. 12, 2018. [Online]. Available: https://www.timereaction.com/papers/manufacturing_next_act.pdf

[5] O. Buhaug, J. J. Corbett, O. Endresen, V. Eyring, J. Faber, S. Hanayama, D. S. Lee, D. Lee, H. Lindstad, and A. Markowska, "Second imo ghg study 2009," Int. Maritime Org. (IMO) London, U.K., Tech. Rep. 1, 2009, vol. 20.

[6] *Publications—Eurostats: Maritime Transport of Goods Quarterly Data*. Accessed: Nov. 18, 2018. [Online]. Available: http://ec.europa.eu/eurostat/statistics-explained/index.php

[7] A. Gharehgozli, H. De Vries, and S. Decrauw, "The role of standardisation in European intermodal transportation," *Maritime Bus. Rev.*, vol. 4, no. 2, pp. 151–168, Jun. 2019, doi: 10.1108/mabr-09-2018-0038.

[8] T. K. Sung, "Industry 4.0: A Korea perspective," *Technol. Forecasting Social Change*, vol. 132, pp. 40–45, Jul. 2018.

[9] *International Data Spaces Association*. Accessed: Nov. 7, 2018. [Online]. Available: https://www.internationaldataspaces.org/

[10] FI-WARE Consortium. *FIWARE: The Open Source Platform for Our Smart Digital Future*. Accessed: Sep. 21, 2018. [Online]. Available: https://www.fiware.org/

[11] *Apache NiFi*. Accessed: Jan. 10, 2019. [Online]. Available: https://nifi.apache.org/

[12] *Apache Spark—Lightning-Fast Cluster Computing*. Accessed: Jul. 12, 2019. [Online]. Available: http://spark.apache.org/

[13] *Apache Kafka. A Distributed Streaming Platform*. Accessed: Jul. 24, 2019. [Online]. Available: https://kafka.apache.org/

[14] C. Liu, H. Jula, K. Vukadinovic, and P. Ioannou, "Comparing different technologies for containers movement in marine container terminals," in *Proc. IEEE Intell. Transp. Syst. (ITSC)*, Oct. 2000, pp. 488–493, doi: 10.1109/ITSC.2000.881118.

[15] D. Sarabia-Jacome, I. Lacalle, C. E. Palau, and M. Esteve, "Enabling industrial data space architecture for seaport scenario," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Apr. 2019, pp. 101–106.

[16] N. Jesse, "Internet of Things and big data—The disruption of the value chain and the rise of new software ecosystems," *IFAC-PapersOnLine*, vol. 49, no. 29, pp. 275–282, 2016, doi: 10.1016/j.ifacol.2016.11.079.

[17] I. Schirmer, P. Drews, S. Saxe, U. Baldauf, and J. Tesse, "Extending enterprise architectures for adopting the Internet of Things–lessons learned from the smartPORT projects in Hamburg," in *Business Information Systems*. Springer, 2016, pp. 169–180, doi: 10.1007/978-3-319-39426-8_14.

[18] P. Fernández, J. Santana, S. Ortega, A. Trujillo, J. Suárez, C. Domínguez, J. Santana, and A. Sánchez, "SmartPort: A platform for sensor data monitoring in a seaport based on FIWARE," *Sensors*, vol. 16, no. 3, p. 417, Mar. 2016, doi: 10.3390/s16030417.

[19] A. Belfkih, C. Duvallet, and B. Sadeg, "The Internet of Things for smart ports: Application to the port of Le Havre," in *Proc. Int. Conf. Intell. Platform Smart Port (IPaSPort)*, 2017.

[20] A. Belsa, D. Sarabia-Jacome, C. E. Palau, and M. Esteve, "Flow–based programming interoperability solution for IoT platform applications," in *Proc. IEEE Int. Conf. Cloud Eng. (ICE)*, Apr. 2018.

[21] L. Heilig and S. Voß, "Information systems in seaports: A categorization and overview," *Inf. Technol. Manage.*, vol. 18, no. 3, pp. 179–201, Sep. 2017, doi: 10.1007/s10799-016-0269-1.

[22] J. Mu nuzuri, L. Onieva, P. Cortés, and J. Guadix, "Using IoT data and applications to improve port-based intermodal supply chains," *Comput. Ind. Eng.*, to be published. doi: 10.1016/j.cie.2019.01.042.

[23] V. Carlan, C. Sys, and T. Vanelslander, "How port community systems can contribute to port competitiveness: Developing a cost–benefit framework," *Res. Transp. Bus. Manage.*, vol. 19, pp. 51–64, Jun. 2016.

[24] M. Baron and H. Mathieu, "PCS interoperability in Europe: A market for PCS operators?" *Int. J. Logistics Manage.*, vol. 24, no. 1, pp. 117–129, May 2013.

[25] M. Lind, T. Andersen, M. Bergmann, R. T. Watson, S. Haraldson, M. Karlsson, M. Michaelides, J. Gimenez, R. Ward, and N. Bjørn-Andersen, "The maturity level framework for PortCDM," Sea Traffic Manage., Norrköping, Sweden, Tech. Rep. 13, 2018.

[26] B. Otto, M. ten Hompel, and S. Wrobel, "International data spaces," in *Digital Transformation*. Springer, 2019, pp. 109–128.

[27] Á. Alonso, A. Pozo, J. Cantera, F. De La Vega, and J. Hierro, ''Industrial data space architecture implementation using FIWARE,'' *Sensors*, vol. 18, no. 7, p. 2226, Jul. 2018.

[28] G. S. Brost, M. Huber, M. Weiß, M. Protsenko, J. Schütte, and S. Wessel, ''An ecosystem and iot device architecture for building trust in the industrial data space,'' in *Proc. 4th ACM Workshop Cyber-Phys. Syst. Secur. (CPSS)*, 2018, pp. 39–50.

[29] J. Pullmann, N. Petersen, C. Mader, S. Lohmann, and Z. Kemeny, ''Ontology-based information modelling in the industrial data space,'' in *Proc. 22nd IEEE Int. Conf. Emerging Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.

[30] *International Data Spaces Association*. Accessed: Jun. 18, 2018. [Online]. Available: https://www.internationaldataspaces.org/thyssenkrupp-implements-first-use-case-for-industrial-data-space

[31] *International Data Spaces Association*. Accessed: Jun. 18, 2018. [Online]. Available: https://www.internationaldataspaces.org/success-stories/#advaneo/

[32] O. J. Rødseth and A. J. Berre, ''From digital twin to maritime data space: Transparent ownership and use of ship information,'' presented at the 13th Int. Symp. Integr. Ship's Inf. Syst. Mar. Traffic Eng. Conf. (ISIS–MTE), Berlin, Germany, Sep. 2018.

[33] *Boost40*. Accessed: Jun. 16, 2019. [Online]. Available: https://boost40.eu/

[34] G. Vouros, C. Doulkeridis, G. Santipantakis, A. Vlachou, N. Pelekis, H. Georgiou, Y. Theodoridis, K. Patroumpas, E. Alevizos, and A. Artikis, ''Big data analytics for time critical maritime and aerial mobility forecasting,'' *Adv. Database Technol.-EDBT*, vol. 2018, pp. 612–623, 2018.

[35] H. Wang, O. L. Osen, G. Li, W. Li, H.-N. Dai, and W. Zeng, ''Big data and industrial Internet of Things for the maritime industry in Northwestern Norway,'' in *Proc. IEEE Region Conf. (TENCON)*, Nov. 2015, pp. 1–5.

[36] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Real-time Data Systems*. Shelter Island, NY, USA: Manning, 2015. [Online]. Available: http://nathanmarz.com/about/

[37] *Next-Generation Networks—Common Requirements of the Internet of Things*, document ITU-T-REC-Y.2066, 2014. [Online]. Available: https://www.itu.int/rec/T-REC-Y.2066-201406-I

[38] *Internet of Things and Smart Cities—Specific Requirements and Capabilities of the Internet of Things for Big Data*, document ITU-T-REC-Y.4114, 2017. [Online]. Available: https://www.itu.int/rec/T-REC-Y.4114/en

[39] Y. Demchenko, C. De Laat, and P. Membrey, ''Defining architecture components of the big data ecosystem,'' in *Proc. Int. Conf. Collaboration Technol. Syst. (CTS)*, May 2014, pp. 104–112.

[40] L. Schrooten, I. De Vlieger, L. I. Panis, K. Styns, and R. Torfs, ''Inventory and forecasting of maritime emissions in the Belgian sea territory, an activity-based emission model,'' *Atmos. Environ.*, vol. 42, no. 4, pp. 667–676, Feb. 2008.

[41] P. De Meyer, F. Maes, and A. Volckaert, ''Emissions from international shipping in the Belgian part of the North Sea and the Belgian seaports,'' *Atmos. Environ.*, vol. 42, no. 1, pp. 196–206, Jan. 2008, doi: 10.1016/j.atmosenv.2007.06.059.

[42] X. Sun, Z. Tian, R. Malekian, and Z. Li, ''Estimation of vessel emissions inventory in qingdao port based on big data analysis,'' *Symmetry*, vol. 10, no. 10, p. 452, Oct. 2018.

[43] *Valencia Port Report 2016*. Accessed: Dec. 14, 2018. [Online]. Available: https://www.valenciaport.com/wp-content/uploads/Boletin-EstadisticoDici%embre-2016.pdf

[44] *Node-Red Low-Code Programming for Event-Driven Applications*. Accessed: Jul. 21, 2019. [Online]. Available: https://nodered.org/

[45] J. Kestelyn. (2016). *Benchmarking Apache Parquet: The Allstate Experience*. [Online]. Available: https://blog.cloudera.com/blog/2016/04/benchmarking-apache-parquet-the-%allstate-experience/

[46] *HDFS Architecture Guide*. Accessed: Mar. 4, 2019. [Online]. Available: https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html

[47] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, ''Spark: Cluster computing with working sets,'' *HotCloud*, vol. 10, 2010.

[48] *Apache Hbase*. Accessed: Mar. 16, 2019. [Online]. Available: https://hbase.apache.org/

[49] *Postgis Spatial Database*. Accessed: Mar. 21, 2019. [Online]. Available: https://postgis.net/

[50] *Geomesa*. Accessed: Mar. 22, 2019. [Online]. Available: https://www.geomesa.org/

DAVID SARABIA-JÁCOME received the M.Sc. degree in communications technologies, systems, and networks from the Universitat Politècnica de València, Spain, in 2016, where he is currently pursuing the Ph.D. degree with the Escuela Técnica Superior de Ingenieros de Telecomunicación. His research activities and interests include the Internet of Things, big data, cloud computing, fog computing, and virtualization.

CARLOS E. PALAU (Senior Member, IEEE) received the M.Sc. and Ph.D. (Dr.Ing.) degrees in telecommunication engineering from the Universitat Politècnica de València, in 1993 and 1997, respectively. He is currently a Full Professor with the Escuela Técnica Superior de Ingenieros de Telecomunicación, Universitat Politècnica de València. He has over 20 years of experience in ICT research areas in the field of networking. He has collaborated extensively in the research and development of multimedia streaming, security, networking, and wireless communications for government agencies, and defense. He was a Main Researcher with the European Commission of EU-FP6, EU-FP7, and EU-H2020 Programs. He has authored or coauthored over 120 research articles. He is a TPC Member of several IEEE, ACM, and IFIP conferences.

MANUEL ESTEVE received the M.Sc. degree in computer engineering and the Ph.D. (Dr.Ing.) degree in telecommunication engineering from the Universitat Politècnica de València (UPVLC), in 1989 and 1994, respectively. He is currently a Full Professor with the Escuela Técnica Superior de Ingenieros de Telecomunicación, UPVLC, where he is also the Leader of the Distributed Real-Time Systems Research Group. He has over 25 years of experience in ICT research areas networking. He is managing several research and development projects at regional, national, and international levels. He has collaborated extensively in the research and development of projects for government agencies and defense, and EU-FP6, EU-FP7, and EU-H2020 Programs as the Chairman of the agreement between Spanish MoD and UPVLC. He has authored or coauthored over 100 research articles.

FERNANDO BORONAT (Senior Member, IEEE) was born in Gandia, Spain. He received the M.E. and Ph.D. degrees in telecommunication engineering from the Universitat Politècnica de València (UPV) at Campus Gandia, Gandia, in 1994 and 2004, respectively. After working for several Spanish telecommunication companies, he moved back to UPV, in 1996, where he is currently an Associate Professor with the Communications Department. He is also the Head of the Immersive Interactive Media Research and Development Group, UPV. He has authored two books, several book chapters, an IETF RFC, and more than 100 research articles. He is involved in several IPCs of national and international journals and conferences. His main topics of interest include communication networks, multimedia systems, multimedia protocols, and media synchronization. He is a member of the ACM. He is an Editor of *Media Sync: Handbook on Multimedia Synchronization* (Springer, 2018).