DOI: 10.2507/31st.daaam.proceedings.070

BIG DATA-DRIVEN DIGITAL ECOSYSTEM FRAMEWORK FOR RAILWAY REPORTING

Alexander Suleykin[a] & Peter Panfilov[b]

[a] *Russian Academy of Sciences' V.A.Trapeznikov Institute of Control Sciences, Profsoyuznaya St. 65, Moscow 117342, Russian Federation*
[b]*National Research University – Higher School of Economics, Myasnitskaya St. 20, Moscow 101000, Russian Federation*

This Publication has to be referred as: Suleykin, A[lexander] & Panfilov, P[eter] (2020). On Big Data-Driven Digital Ecosystem Framework for Railway Reporting, Proceedings of the 31st DAAAM International Symposium, pp.0499-0509, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-29-7, ISSN 1726-9679, Vienna, Austria
DOI: 10.2507/31st.daaam.proceedings.070

Abstract

In our application research, we propose a Digital Ecosystem approach to overcome data integration, orchestration and quality challenges in railway reporting system using Big Data technologies. We are building a Digital Ecosystem Framework consisting of different Agents, where each Agent is an essential part of the Railway Reporting Management System. In this work, we address different problems in building digital ecosystem including integration problems, orchestration problems and data quality problems. We present a proprietary solution called the Digital Ecosystem Reporting Framework (DERF) for building robust, reliable, fault-tolerant, scalable and high-loaded data pipelines of the Railway Reporting Management System based on Big Data technologies. DERF integrates different Digital Agents such as main ETL-pipeline Agents, technical data quality Agents, business data quality Agents, BI-services integration Agents and high-level data orchestration Agent. A test implementation of DERF has been performed for Railway Reporting Management System using KPI reporting data of the real Railway company.

Keywords: Big Data; Data Processing Pipeline; Railway Reporting Management System; Digital Ecosystem.

1. Introduction

Today, to remain competitive in the global market public corporations and companies from different sectors of the economy have to leverage their competencies in ICT taking advantages of new opportunities that digital transformation brings for new business models and demands for innovations in their respective ecosystems. The most substantial, complex and problematic sectors in this respect are industrial production, power generation and transportation. Among factors determining the competitiveness of industrial manufacturers are the flexibility and efficiency of their operations through the use of advanced digital technologies that ensure the rapid and effective collection and use of data, communication and interaction between different connected production systems and intelligent applications to solve production problems. These issues are in focus of such national and international initiatives as Smart Manufacturing System (SMS) or Industrie 4.0 (in the USA and Germany, respectively) [1].

An apparent trend in the global economy is transformation of sectoral and cross-sectoral digital platforms into digital ecosystems that allow for creating new business models, promoting innovations and increasing competitiveness. By digital ecosystem we mean a distributed socio-technical system with adaptability, self-organization, and sustainability capabilities, operating in a competitive environment and cooperation between various actors of this system (automated systems and economic actors) for knowledge exchange in the evolutionary system development. The digital ecosystem operates on the basis of computer network infrastructure using multi-agent technologies [2].

Currently, there are certain discrepancies in the definition of the digital ecosystem. It is argued [3] that the digital ecosystem is formed through the integration of IT networks, social and knowledge sharing. The concepts of e-learning ecosystem and digital ecosystem are identified. Central Energy System (CES) provides access to knowledge, global value chains, specific services, the adaptation of new technologies, the adoption of new business models. The economy is no longer viewed as a fully managed system for which a functioning plan is drawn up: individual active elements determine its functioning, depending on the current situation, and this is already ecosystem [4].

Authors in [5] refer to the digital ecosystem as a set of digital twins and infrastructure for data transmission, storage and processing, as well as system users, including social, economic, political, psychological and other factors affecting the implementation of interactions. In the digital ecosystem, "partners and competitors interact as a single team, combining resources, knowledge to work together on projects in the mode of mutual completeness of information and creation (co-creation), without ceasing to compete within other processes" [6]. In our work, we present Digital Ecosystem with Agents, which solve data integration issues, orchestration and data quality issues. All Digital Agents are connected with one main Control Agent managing the complete data pipeline.

The complexity of big data processing and analysis is extremely increasing due to data volume growth, data variety, velocity, different data formats of data transmission, integration problems and other data complexities. Building a robust, reliable and fault-tolerant data processing and storage framework that is capable of handling big data loads with data in various formats and high volumes from different data sources and systems represent a great challenge to application developers. Many researchers consider the workflow modeling as a viable approach to the design and implementation of distributed application systems for processing large volumes of data (Big Data) [7]. However, in this work we apply Ecosystem-based approach for dataflow modeling and tested it on the basis of Railway Reporting System implementation. In our work, we will concentrate on solving different data integration, orchestration and quality problems in railway sector of a wider transportation system, applying new approach to Railway Digital Reporting problem.

The current research expands previous work on ETL-pipeline data processing with Open-Source Big Data technologies [8] and railway KPIs data processing architecture [9], and main concepts of Digital Ecosystems application in Supply Chain Management in [10]. In this paper, we focus on Open-Source Big Data technologies for building Digital Ecosystem Reporting Framework (DERF) to solve data integration issues, orchestration and different data quality issues for Railway Reporting. DERF consists of a set of Digital Agents – services for solving ETL-tasks, business and technical data quality issues, interaction with Business Intelligence (BI) subsystem and solving integration issues. Based on DERF, data processing pipeline has been performed for Railway Reporting Management System.

The rest of this paper organized as followed: first, we review the main data processing challenges for Railway Reporting processes and main problematic issues. Second, we provide architectural overview of proposed solution and its main Agents, creating the methodology for Digital Ecosystem Reporting (DER). Third, we design architecture of the Digital Ecosystem Reporting Framework (DERF) for Railway Reporting, and describe main Agents and data processing pipeline in our practical implementation of the framework. Then, we provide description of the railway KPI data for reporting along with experimental setup parameters (software and hardware), which were used in our test implementation. After that, Digital Ecosystem-based pipeline is modeled and implemented based on described data, methodology, software and hardware parameters. Finally, we conclude our paper with experiment results analysis and future works.

2. Main data processing challenges for Railway Reporting

2.1. Data integration and orchestration challenges

In fact, reporting is one of the most comprehensive business processes in any organization. It suggests multitude of data integration efforts while using many different data sources, formats, protocols, and certain methodologies and tools. Reporting requires that data is ready and available by specific time, which means that many human efforts should be organized and coordinated to realize such pipeline in due term. Moreover, many ETL-steps execute in strict sequence and follow other processes or are followed by other processes. The issue of a data integration methodology is of high relevance here because we need to define an effective data pipeline with proper steps, integrate many diverse data sources and conform to the Reporting SLAs. All these challenges of data integration and ETL-processes orchestration are met with DERF with the help of central Agent, i.e. an Orchestration Agent, and ETL Agents, which launch particular ETL jobs.

2.2. *Business data quality issues*

Business data quality issues are of high importance as well as integration for Reporting. Data from different source systems can vary from methodological perspective, forming data inconsistency. Moreover, data from different sources can be incomplete or incorrect. In order to overcome data inconsistence, incompetence and incorrectness special Data

Quality jobs need to be launched right after main ETL processes are finished in order to have all necessary statistics about data quality in particular period of time, just before main reports are required to be prepared. Board of Directors wants to know exactly all the problems in reports and in data – which source system generated the problem, why it happened, where is the responsible department etc. Thus, all these business needs have been implemented in DERF, where just after main ETL-processes launched Data Quality Agents for checking data quality issues in new loaded data.

2.3. Technical data quality issues

Technical data quality issues are important in order to check and monitor the status of all ETL Agents. After each ETL data transformation is finished it is highly important to make sure that data is in a correct place and all transformations have been finished successfully. In order to check that all data have been fully transferred to another layer and no data loss occurred, it is needed to check amount of data records at each data representation layer. In DERF in a separate stream from main ETL pipeline it is launched special Technical Data Quality Agents, which count data records in each data representation layer in Storage Agents.

2.4. Business Intelligence and DWH interaction

Business Intelligence system's goal is to visualize reporting data. Data Lake or Data Warehouse represent main sources of data for the BI system, and usually data loading in BI systems starts upon successful completion of all data transformations in Data Lake. This event-driven approach proved to be effective because in real world situation preparing data by exact time is not always possible. The amount of data collected from the source systems for reporting varies from day to day, and on different days, data may be ready for BI processing in different time. Thus, an interaction between the BI service Agent and Storage Agent need to be configured in a way, so that the BI system can start loading data only when special signal from Storage Agent through API is generated. In our DERF implementation, a special BI service Agent is designed that launches BI jobs when all data marts are prepared.

3. Architectural Overview of Digital Reporting Ecosystem

Digital Reporting system is highly important system in any organization. In order to build a robust, reliable, fault-tolerant, scalable and high-loaded data pipelines for Digital Reporting that follows strict SLAs it is proposed to use the following architecture of Digital Reporting that conforms to Digital Ecosystem concept (Fig. 1):

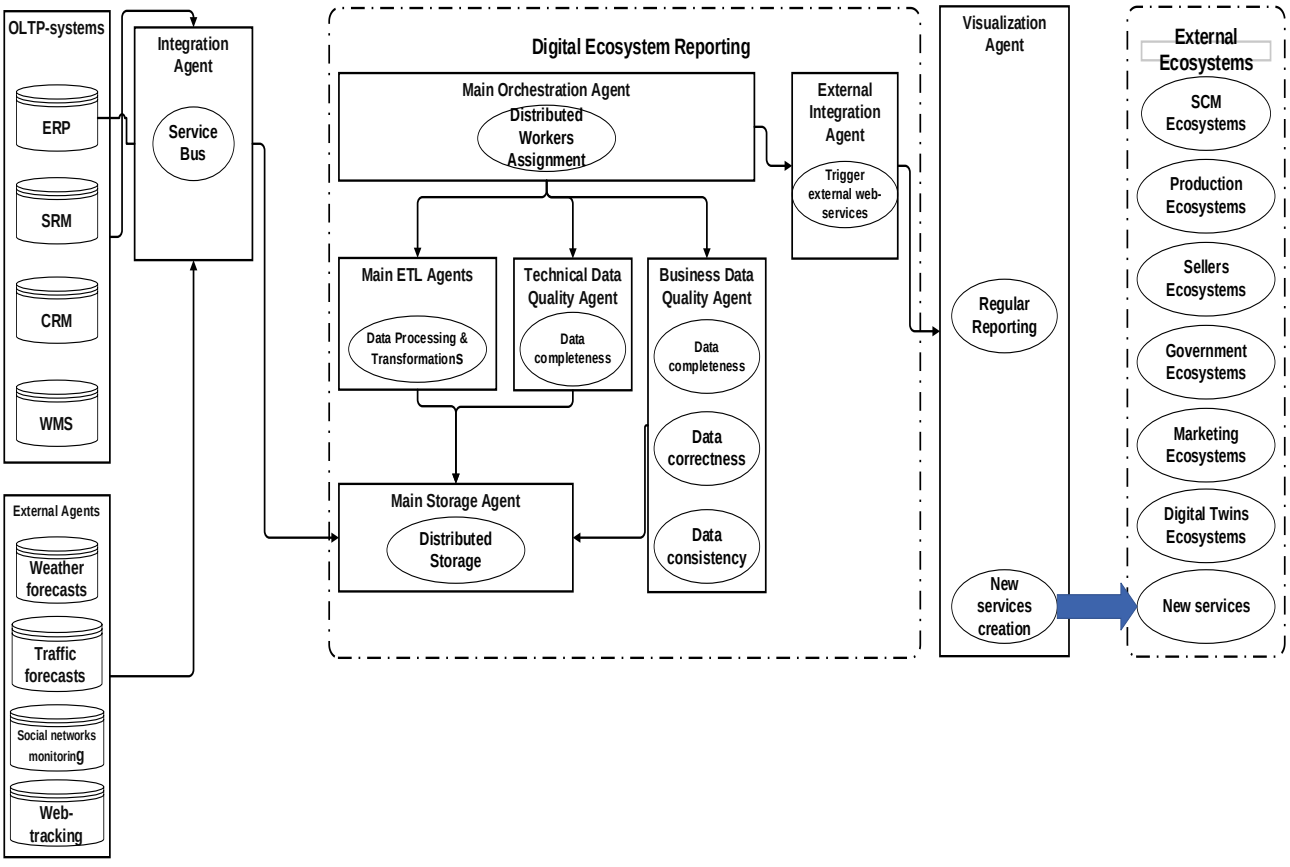

Fig. 1. The architectural overview of Digital Ecosystem Reporting (DER)

DER includes:

- Main Orchestration Agent – Agent responsible for orchestration all other Agents – ETL, Business and Technical Data Quality, BI service Agent and Storage Agent. It is central part of the system and all other Integrations and new Agents design will be orchestrated in this Agent as well. It have a flexible distributed workers assignment algorithm, and only available, free and alive worker nodes can be assigned for task. This solution provides robust, fault-tolerant, reliable and scalable design for ecosystem.
- External Integration Agent – Agent that communicates with BI Agent. The role of this Agent is to trigger BI Agent, where BI data load is supposed to immediately start. In addition, the role of this Agent is also interaction with any External Ecosystems, which can be integrated in DRE in the future.
- Main ETL Agents – Agents for ETL data processing and transformations through all data pipeline until visualization. ETL can be presented in any kind of job, but preferably they need to be distributed and fault-tolerant.
- Technical Data Quality Agent – Agent that provides data quality checks after each ETL process is finished. It is highly important to make sure that all ETL processes are finished successfully and no data loss has occurred. The main goal of this Agent is to check data completeness of each storage layer.
- Business Data Quality Agent – Agent, which is carrying out all business data quality checks on all storage layers. The goal of this Agent is to check data completeness, correctness and consistency. Data Quality issues are highly important for reporting and all data quality issues associated with source systems should be detected immediately after main ETL processing is finished, and special data quality reports prepared for management as well as main data reports itself.
- Main Storage Agent – Agent that stores all data in a distributed and reliable manner. All data is stored in this Agent and all processing results are put in this Agent as well.

External Ecosystems are required for data integration with all source systems, data visualization for reporting and interaction with other Ecosystems. Based on visualization regular reports, management will make data-driven decisions, which will inevitably lead to new services and new Ecosystem creation.

4. Digital Ecosystem Reporting Framework for Railway Sector

Based on the DER concept described above, we have proposed a DER framework for Railway Sector that includes specific ETL-processed for Railway reporting. Fig. 2 depicts the architecture of the proposed Railway DERF.

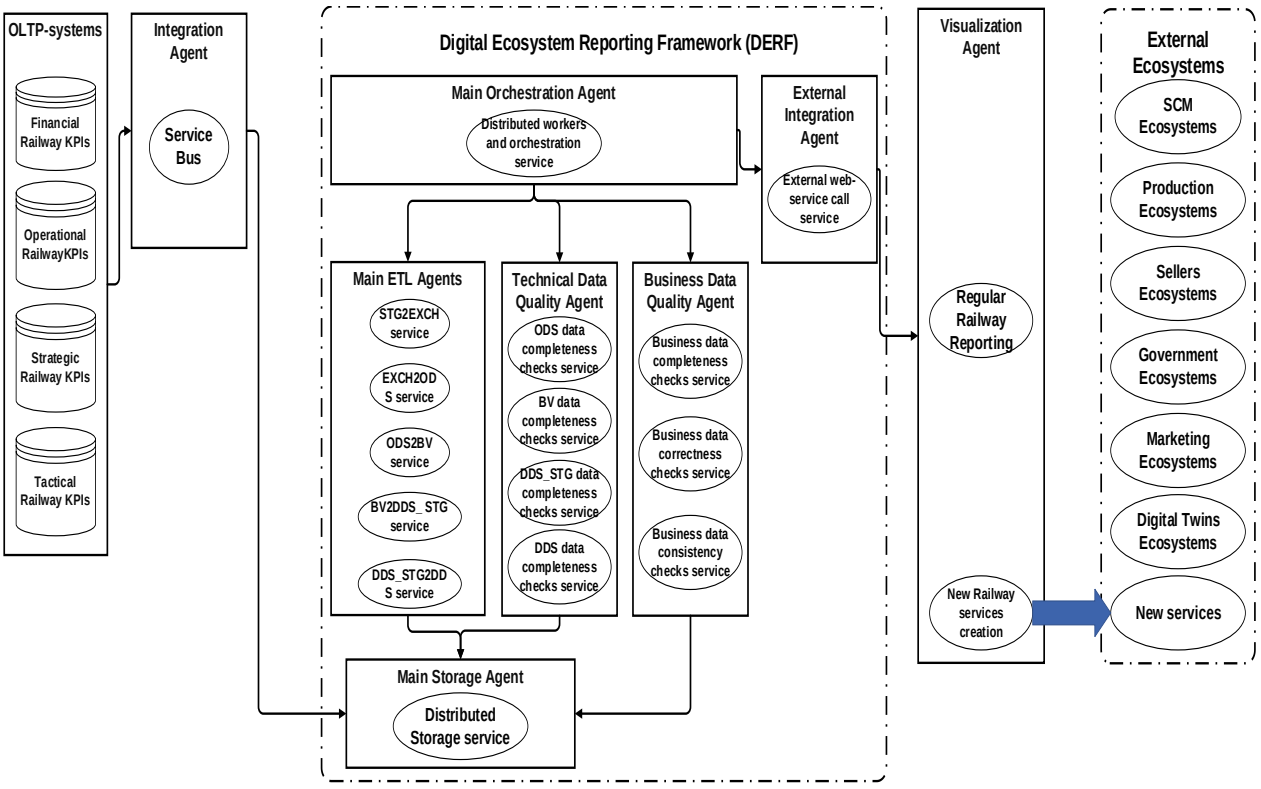

Fig. 2. The DERF architecture for Railway Reporing

4.1. Storage Agent and main ETL Agents dataflow description

In our research, we built a powerful multi-staged data processing pipeline that combine two major super stages or layers – Data Storage Layer and Serving Layer (fig. 2). Here Storage Layer and Serving Layer have their own Layers (sublayers), which are used for methodological correctness of data load. The data processing pipeline of the whole data movement is strict and should go through the following sublayers/stages inside Serving and Storage Layers as it is depicted in fig. 3. The details of the Storage Agent implementation such as open source technology used, functionality and data transformations are presented in Table 1.

Name	Abbreviation	Location	Definition and functions	Transformations
Staging Buffer Area	STG/BUF	HDFS	The area of temporary data accumulation in the format corresponding to the source without any transformations. Streaming data comes from sources.	No
Staging Exchange Area	STG/EXCH	HDFS	The intermediate region for forming the next ETL processing packet. All accumulated data are moved from the buffer to form a data processing packet. It is assigned a unique BATCH_ID.	BATCH_ID
Staging Archive Zone	STG/ARCH	HDFS	Storage of the complete archive of incoming messages without transformation of the storage format. Incoming messages are archived after successful processing.	Archiving and enlarging storage files.
Operational Data Store	ODS/FULL	HDFS	The area in which the source data scheme is stored, but they are reduced to a single binary form of storage. It contains the entire history of changes and deletions.	Convert to binary storage format. Conversion from object to relational storage.
Batch View	ODS/BW	HDFS	It contains only an actual slice of the state of objects without a change history and deleted records.	Calculation of the actual data slice.
Detail Data Store Staging	DDS_STG	PostgreSQL	Batch layer. A separate instance is created for each source system. One-to-one data is transferred from HDP and stored only between downloads. Both full data load and only line changes (deltas) can come.	
Detail Data Store	DDS	PostgreSQL	The layer of the current data slice presented in a relational form. Creating a single data model (without unification). Re-keying (generation of internal storage IDs).	Conversion from object to relational storage. Data normalization (if needed). Storing a current data slice.
Detail Data Store View	DDS_v	PostgreSQL	The layer for data access to dds	All data access to dds should be organized using views on top of dds layer istelf
Data Mart	DM	PostgreSQL	Groups showcases by a specific attribute, most often the subject area. Contains unified detailed data. It contains calculated indicators for use in reporting. Calculations of indicators used in several reports is submitted to this layer.	Data unification. Denormalization of data. Data Aggregation. Calculation of derived indicators used in several places.
Data Mart Reporting	DM_rep	PostgreSQL	The final reporting layer. From it, data are used only for display in BI tools. It is forbidden to build some reports on the basis of others. Only with the transfer of the information used in the DM layer. Calculation of indicators specific to specific reporting. It can be both logical and physical.	Calculation of derived indicators specific to a particular report.

Table 1. Storage Agent description of data processing pipeline

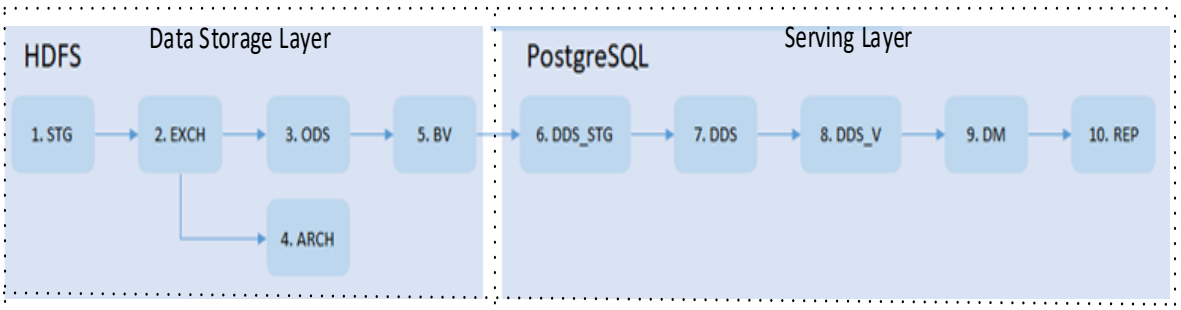

Fig. 3. Main ETL data processing pipeline for Railway DERF

4.2. Open-source technologies for DERF implementation

The DERF is built using open-source Big Data technologies, which satisfy following requirements:

- Horizontally scalable Agents for data storage – a layer for data loads to the archival and distributed storage
- Agent with distributed data processing engine capable of processing large volumes of data
- Fault tolerant Agent – the solution should work without incurring data loss even in case of single machine failure
- All Agents should be orchestrated and combined in one main Agent that can run other Agents
- Agent with relational database for data access to different BI tools
- Be Open Source.

Thus, we have defined the following Open-Source Big Data technologies for DERF implementation (table 2):

№	Component	Definition and role in DERF
1	Spark	Distributed in-memory framework for high-load data processing [11]. Spark has been used to create all main ETL Agents, which processed data and move it through all Agents
2	HDFS	Distributed fault tolerant file system optimized for storage for processing large volumes of data [12]. A main Agent for distributed data storage of Railway data for Reporting in test implementation
3	PostgreSQL	Relational database for universal data access from BI systems [13]. In test experiment, we used PostgreSQL as a main Agent for Serving layer and access to BI Agent
4	AirFlow	Orchestrator and Scheduler [14]. In our experiment, Airflow was main Agent for other Agents orchestration and scheduling
5	Python	Programming language [15]. Data quality Agents are built using python programming language.
6	Docker	Most popular containerization technology [16]. In DERF, docker was used for data quality solutions containerization for easy deployments

Table 2. Selected technologies for DERF modeling and implementation

Note: the benchmarking of selected technologies was not a part of our research. Technologies selection were based on market research, experience and their effectiveness in our previous researches.

4.3. Digital Ecosystem characteristics of the DERF

The DERF implementation conforms to Digital Ecosystem concept. From a technological perspective, the Digital Ecosystems are considered to be robust, self-organizing and scalable architectures that can automatically solve complex, dynamic problems. The multidisciplinary nature of Digital Ecosystems requires appropriate distributed computing architectures unlike traditional distributed architectures, which are developed to deal with software architectures within a single discipline, e.g., service oriented architecture or SOA. A DERF possesses all the characteristics of the Digital Ecosystem as a dynamic, distributed, complex system of systems with properties of self-organization, scalability, sustainability and dynamism. In the following paragraphs, we introduce the main characteristics of the DERF namely self-organization, scalability, stability and reliability, distribution, and interconnection with external ecosystems.

The achievement of self-organization within Digital Ecosystems assumes the necessity of superior capacity of reproduction of the Digital Agents with a minimum level of intervention from the Human Agents. The DERF is able to function without any human intervention. All Agents are deployed from the Docker [16] containers, forming their own independent digital environment, network and storage. All containers are configured in the way to be interconnected with each other. If in some reason container fails, it will automatically try to restart and make Digital Agent up again. In future, it is planned to increase the self-organization capabilities of DERF implementing Kubernetes cluster [17] on top of docker containers, which will allow better container management – load balancing, monitoring, control and auto-failover.

The characteristic of scalability of DERF is represented by the scalability of each of the ETL Agent, Storage Agent and Orchestration Agent. Used services are horizontally scalable by their nature that is Apache Spark [11], HDFS [12] and Airflow worker nodes [14]. The amount of processed data is proportionally increasing along with adding new Airflow and Spark workers in a cluster. Also, possible data storage is increasing linearly with adding new nodes to HDFS cluster.

The characteristic of stability and reliability of the system is explained and supported by the fault-tolerant properties of the central Agent of the system that is the Orchestration Agent. In case of failure of some nodes in the system, others will continue to work, and only working nodes will be assigned for new tasks by Orchestration Agent without a data loss. Storage Agents and ETL Agents are also implemented as fault-tolerant agents that prevents from data loss and provides task execution in case of single node failure. This property make DERF a stable and reliable Digital Ecosystem.

Distribution property of DERF is explained by the distributed implementation of ETL Agents and Storage Agents. Thus, all ETL data processing is carried out in distributed manner with Apache Spark allowing solution be capable with huge amount of data. Data is processed on different compute nodes according to the demands of computational resources for particular task. In addition, data is stored in distributed way with HDFS Agent.

The characteristic of external ecosystem interconnection is the ability of a DERF to be connected to other external ecosystems and integrated in even more complex configurations. Interconnections can be made using Restful APIs integrated in Airflow Agent [14]. Any ETL Agent can be triggered with external API, and otherwise, any ETL Agent can trigger external Ecosystems. This cooperation of Ecosystems potentially can be a strong driver for creation new services and Ecosystems. Thus, it is expected, that DERF will be integrated in company infrastructure, influencing changes of business processes in organization driven by Big Data Digital Ecosystem.

Additionally, the characteristic of dynamism can be considered in context of the DERF. Dynamic characteristic of the digital ecosystem can be explained as the profiles of digital entities in the system that constantly change over time. The entity profile provide description of the entity specification, such as attributes, relations, interaction states with other entities, etc.

5. Experiments and results

5.1. Railway data description

The proposed Digital Ecosystem has been tested using Reporting KPIs data from one railway company. The data is represented by star schema that consists of one fact table (the main table with events – KPIs) referencing so called dimension tables (dictionaries in this case). The data is normalized to the 3-rd normal form (3NF).

Nine entities have been used in our experiments for testing data processing workflow implementation. Those entities were datamart, cargo-type, data_type, date-type, metric-type org, unit, val_type and var. Data volumes of these entities are presented in table 3, which give us the total volume of data amounting 26,4 Gb. The description of entities and data types are presented in table 3.

Entity	Number of records
datamart	30 mln
cargo_type	46
data_type	2
date_type	5
metric_type	15
org	84
unit	120
val_type	5
var	418

Table 3. Data volumes of test entities

Entity	Attribute	Data type	Description
DATA_TYPE	ID	INTEGER	Dictionary – type of data for KPI. Can be approved or planned
	NAME	CHAR	
DATE_TYPE	ID	INTEGER	Dictionary – type of date period for selected KPI. Can be day, week, month, year
	NAME	CHAR	
METRIC_TYPE	ID	INTEGER	Dictionary – version of the KPI
	NAME	CHAR	
ORG	BK	INTEGER	Dictionary – branch of main company for selected KPI
	ID	INTEGER	
	ROOT_ID	INTEGER	
	CHILD_ID	INTEGER	
	VNAME	CHAR	
	NAME	CHAR	
UNIT	ID	INTEGER	Dictionary – unit of measurement
	NAME	CHAR	
VAL_TYPE	ID	INTEGER	Dictionary – type of cumulative data representation. Can be cumulative values by years, months, days, weeks
	NAME	CHAR	
VAR	VAR	INTEGER	Dictionary – the variable represents KPI name itself and definition
	NAME	CHAR	
DATAMART	DATE	DATE	Fact table, represent all KPIs of railway company
	MUNIT	CHAR	
	DATA_TYPE	INTEGER	
	VAR	INTEGER	
	VALUE	DOUBLE	
	VAL_TYPE	INTEGER	
	METRIC_TYPE	INTEGER	
	DATE_TYPE	INTEGER	
	SOURCE	CHAR	
	ORG	INTEGER	
	CARGO_TYPE	INTEGER	
	REASON_TEXT	CHAR	
	MOD_TIME	TIMESTAMP	
	USER_LOGIN	CHAR	
CARGO_TYPE	CARGO_TYPE	INTEGER	Dictionary – type of cargo for Cargo KPIs
	NAME	CHAR	

Table 4. Railway KPIs data description and types

5.2. Experimental setup parameters

Tests of the proposed DERF were carried out in the simulation testbed. Platform servers are located on virtual machines based on the same physical server. The testbed was deployed using Microsoft Hyper-V virtualization based on a dedicated servers, and test runs were conducted using virtual machines. The configuration of a server is shown in Table 5, and the configuration of test virtual machines is shown in Table 6 respectively:

Server model	CPU	RAM	Disks
Huawei FusionServer 1288H V5	12-core (24 logical HT) Intel Xeon Gold 6126 2.6GHz	512 Гб DDR4, 32GB, 2666MT/s, 2Rank (2G*4bit), ECC	9 TB 300GB, SAS 12Gb/s, 10K RPM Huawei Avago3508 RAID SR450C-M 2G

Table 5. Host Configuration - Virtualization Server

VM host	CPU (vCores)	RAM Gb	HDD Tb	Services and their role
pg-source_system.bd	8	16	1	Postgre SQL server – data base for Serving Layer
bd-hdp-edge01	8	32	1	Airflow server – ETL orchestration
source_systemn-hdp-01	4	32	1	• PostgreSQL metadata server for Ambari, Hive, Ranger • Ambari Server; Ranger Admin; HDFS JournalNode • Ranger KMS; Ranger Usersync • Zookepeer Server; Solr Server; Ambari Metrics Server • Ambari Metrics Grafana
source_systemn-hdp-02	4	32	1	• HBase Standby Master; HDFS Standby Namenode • Spark2 History Server • YARN Active ResourceManager; YARN Registry DNS; YARN Timeline Service V2.0 Reader • Zookeeper Server; ZKFailoverController
source_systemn-hdp-03	4	16	1	• HDFS Active NameNode; HDFS JournalNode • MapReduce2 History Server • Hive Metastore; Hive Server • YARN Standby ResourceManage • YARN Timeline Service V1.5 • ZooKeeper Server; ZKFailoverController
source_systemn-hdp-04	4	16	1	• HBase Active Master • HDFS JournalNode • Zookeper Server • Spark2 Thrift Server
source_systemn-hdp-05	4	16	1	• HDFS Datanode • HBase Regionserver • YARN Nodemanager
source_systemn-hdp-06	4	16	1	• HDFS Datanode • HBase Regionserver • YARN Nodemanager
source_systemn-hdp-07	4	16	1	• HDFS Datanode • HBase Regionserver • YARN Nodemanager
source_systemn-hdp-08	4	16	1	• HDFS Datanode • HBase Regionserver • YARN Nodemanager
source_systemn-hdp-09	4	16	1	• HDFS Datanode • HBase Regionserver • YARN Nodemanager
source_systemn-hdp-10	4	16	1	• HDFS Datanode • HBase Regionserver • YARN Nodemanager
source_systemn-hdp-11	4	16	1	• Kafka broker; kafka-hdfs-connector • NFSGateway: /mnt/hdfs • Hive client; Spark2 client

Table 6. Testbed configuration

5.3. Test results

As a result of modelling, the Directed Acyclic Graph (DAG) is created that shows DERF Agents in action. Thus, after successful execution of DAG in Airflow, the DAG itself changes to green colour indicating that all DAG steps have been finished properly (fig. 4).

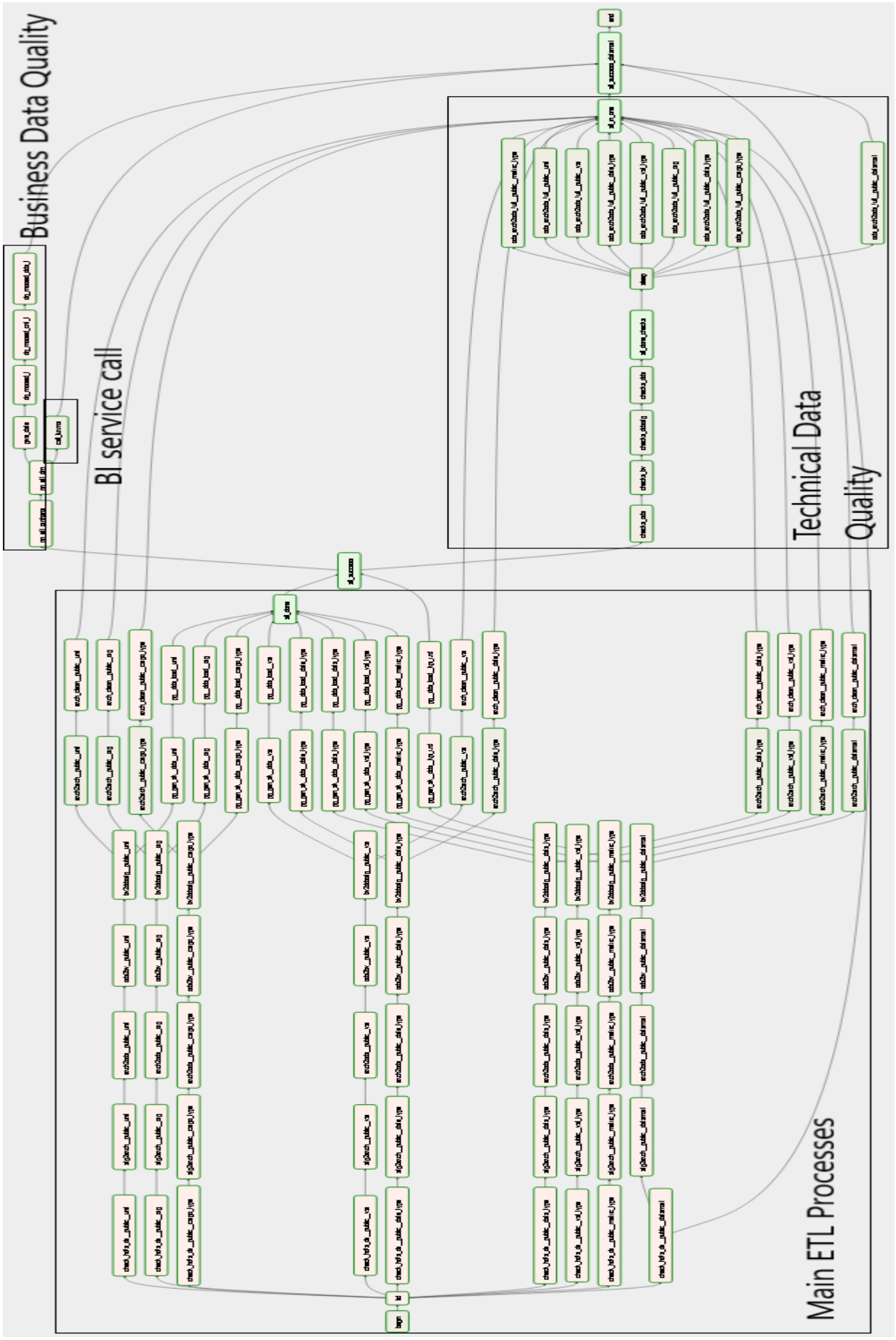

Fig. 4. Resulted DAG execution for Railway DERF.

6. Conclusion

In our paper, a Digital Ecosystem Reporting Framework (DERF) for overcoming data integration, orchestration and quality challenges using Big Data technologies for railway reporting management system has been created. We have regarded different data processing challenges for Railway Reporting sector such as Integration, orchestration, BI subsystem interaction and data quality issues. DERF framework has been built with a main concept of Digital Ecosystems and Open-Source Big Data technologies. We propose to use DERF for building robust, reliable, fault-tolerant, scalable and high-loaded data pipelines for Reporting Management System.

Proposed Ecosystem has been tested on the basis of railway data for Reporting System, and successful implementation of ETL-Agents, Integration, Data Quality and BI interaction have been performed. Test DERF implementation has shown the versatility of the proposed architectural framework and its applicability for other domains and use cases.

7. Acknowledgments

The reported study is conducted within the framework of the HSE University Project Group Competition 2020-2022 and was partially supported by RFBR grants № 20-07-00958 and № 20-31-70001.

8. References

[1] N. Bakhtadze, B. Pavlov, V. Pyatetsky, Suleykin, A. Digital Energy Ecosystems. IFAC-PapersOnLine, Volume 52, Issue 13, Berlin: Elsevier, 2019, pp. 30–35, DOI: 10.1016/j.ifacol.2019.11.088.

[2] Prince Kwame Senyo, Kecheng Liu, John Effah. Understanding Behaviour Patterns of Multi-agents in Digital Business Ecosystems: An Organisational Semiotics Inspired Framework. In book: Advances in Human Factors, Business Management and SocietyPublisher: Springer, Cham, 2018. DOI: 10.1007/978-3-319-94709-9_21.

[3] Nachira F., Dini P., Nicolai A.A. Network of Digital Business Ecosystems for Europe: Roots, Processes and Perspectives. Digital Business Ecosystems. Bruxelles: European Commission, 2007.

[4] Chang E., West M. Digital Ecosystems: A Next Generation of the Collaborative Environment. iiWAS, 2006, pp. 3–24.

[5] Dong H., Hussain F.K., Chang E. An Integrative view of the concept of Digital Ecosystem. Proceedings of the Third International Conference on Networking and Services. Washington, DC, USA: IEEE Computer Society, 2007, pp. 42—44.

[6] Baker K.S., Bowker G.C. Information ecology: open system environment for data, memories, and knowing. J. Intell. Inf. Syst., 2007, vol. 29, no. 1, pp. 127—144.

[7] Zhang, B.; Yu, L.; Feng, Y.; Liu, L.; Zhao, S. (2018). Application of Workflow Technology for Big Data Analysis Service. *Appl. Sci.* 2018, *8*, 591.

[8] Suleykin, A., Panfilov, P. (2019). Implementing Big Data Processing Workflows using Open Source Technologies, Proceedings of the 30th DAAAM International Symposium, pp.0394-0404, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-22-8, ISSN 1726-9679, Vienna, Austria DOI: 10.2507/30th.daaam.proceedings.054.

[9] A. Suleykin, P. Panfilov and N. Bakhtadze, "Industrial track: Architecting railway KPIs data processing with Big Data technologies," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 2047-2056, doi: 10.1109/BigData47090.2019.9006196.

[10] Suleykin, A., Bakhtadze N. Architecture Models of Digital Ecosystems in Supply Chain Management. Information Technologies and Computing Systems. 2019, No. 4, pp. 21-33. DOI 10.14357/20718632190403

[11] https://spark.apache.org/. Accessed on: 01-06.2020.

[12] https://hadoop.apache.org/. Accessed on: 01-06.2020.

[13] https://www.postgresql.org/. Accessed on: 01-06.2020.

[14] https://airflow.apache.org/. Accessed on: 01-06.2020.

[15] https://www.python.org/. Accessed on: 01-06.2020.

[16] https://www.docker.com/. Accessed on: 01-06.2020.

[17] https://www.docker.com/products/kubernetes. Accessed on: 02-06.2020.