Distributed services and a warehouse as an ecosystem on science and higher education

Emil Podwysocki, Łukasz Błaszczyk, Marta Niemczyk, Jarosław Protasiewicz, Marek Michajłowicz, Sylwia Rosiak, Iwona Kucharska

National Information Processing Institute, Warsaw, Poland

Keywords
warehouse; services for science; open data; web services

1. ABSTRACT

In this study, we demonstrate the concept of a new information ecosystem on science and higher education in Poland. We focus on a distributed architecture of services and a data integrating warehouse. The data warehouse is tailored to be well-suited to micro-services, the modern information system architecture. In addition, we show examples of descriptive and predictive analytics in the data warehouse, which assist policymakers in making decisions. The primary outcome of the study is a new architecture, which relies on combining the data warehouse and micro-services into one ecosystem of distributed services for science and higher education. We hope that our experiences and concepts will be beneficial to those who face challenges in redesigning of existing information systems.

2. INTRODUCTION

Data warehouse solutions have been present in higher education and science for nearly two decades (Baranovic et al., 2003; Benefelds and Niedrite, 2004). They act both as data integrators and analytical tools. Classically, a data warehouse covers several layers of data integration such as a staging area and data marts. Each layer uses data for other layers, but should concurrently remain independent as far as possible. The succeeding layers contain incremental data marts, performing more complex data transformations. Such architecture should more efficiently distribute and analyse the resources required by the science & technology sectors (Menolli and Dias; 2004, 2006). The issue of proper data cleaning likewise must not be overlooked (Mekterović et al., 2009).

Boulila et al. (2018) recognises two types of academic analytics delivered by business intelligence tools, namely (i) descriptive analytics, which presents what has happened; and (i) predictive analytics, which infers what is going to happen. Both disciplines are closely interwoven. We can enumerate several applications of decision support systems based on descriptive analytics. Menolli and Dias (2004, 2006) designed a data warehouse to support decision making in various fields of science and technology. van Dyk (2008) used a warehouse to assess teaching effectiveness based on data from a learning management system and other resources. It helped a higher education institution to provide better courses by micro-decision-making. Mekterović et al. (2009) developed a data warehouse for a higher education information system to generate scholarship grants. Ayesha et al. (2010) and Bhise et al. (2013) utilised data form a warehouse and analysed students' behaviour during courses, which helped teachers to deliver better courses in future. Miranda and Suryani (2014) proposed an analytical tool to improve institutions' performance, and help them to boost decision making processes. Mohammed and Anad (2014) proposed a human resource system for universities. It assisted those universities to employ the right candidates for specific positions.

Predictive analytics allows us to uncover visions of a possible future. Mattingly et al. (2012) showed the importance of such analytics in higher education to predict the success of students, courses, and institutions. Boulila et al. (2018) provided analytical tools to support decision-making from the perspective of students, faculty members, administrators, and decision makers. For example, they helped to answer the questions regarding dropout rates on certain courses, and why students select particular courses or have delays. Alfantookh et al. (2012) tracked various fields of education, analysed the correlations between graduates and teaches, and made predictions. The above examples prove that our demands go beyond those of Excel; we require multidimensional cubes and business intelligence tools, which not only provide users with quick reports, but can also make predictions (Moturi and Emurugat, 2015). Despite this remark, we should be aware that a warehouse, and business intelligence models require tailoring for higher education needs. Only such tailored models may work efficiently, and support strategic analysis at universities (Aljawarneh, 2016). The term of agile analytics was coined (Salaki and Ratnam, 2018). It is the adoption of an agile approach to business intelligence as an opportunity, which allows universities to analyse and improve their performance.

With the above discussion in mind, regarding the role and application of a data warehouse and analytics in science and higher education, we propose our vision of a distributed system of services, including a warehouse as well as descriptive and predictive analytics. More specifically, the aims of this study are as follows:

- To discuss the whole ecosystem of services supporting decision making in science and higher education in Poland.
- To tailor a data warehouse to be well suited to micro-services - a modern information system architecture. We define it as integration services through data.
- To demonstrate descriptive and predictive analytics in the warehouse, which helps policy makers make decisions.

The validity of the study relies on combining a data warehouse and micro-services in the ecosystem of distributed services for science and higher education. According to our best knowledge, it is the first time such architecture has been proposed.

The remainder of this paper is structured as follows: section 3 briefly discusses recent legal changes in Polish science and higher education; Section 4 depicts a distributed system of services, its architecture and our approach to designing and developing it; Section 5 demonstrates how we tailor a warehouse and business intelligence tools; Section 6 introduces our conclusions, and the subsequent sections cover references and the authors' biographies.

3. NEW LAW REQUIREMENTS FOR INFORMATION SYSTEMS ON SCIENCE AND HIGHER EDUCATION

According to Polish law, information about Polish science and higher education has been collected since 2011 in an integrated and centralised information system, POL-on. The main goals of the system are to support the division of various funds for higher education and science; to provide statistical reports for Polish Central Statistical Office; to detect fraud and plagiarism; to provide reports and analysis for policymakers in the Polish Ministry of Science and Higher Information; to evaluate the level of scientific excellence of universities and research institutes; and to share public data related to science and higher education (e.g. courses offered by universities, scientific projects, scientific publications, holders of PhD degrees or professorships, amongst many others) (Michajłowicz et al. 2018).

The system is tightly compliant with the legal requirements. Therefore, a new bill, the Constitution for Science, introduced in Poland last year, which has deeply altered the regulations for science and higher education, must result in profound and numerous changes within the system.

These changes are described below:

(i) Redesign of existing data registers, such as institutions of science and higher education, courses, students, academic teachers and researchers, projects, and patents. This redesign takes into account:
   a. changed scope of the data (e.g. academic teachers' and researchers' data must include detailed information about the discipline of scientific research);
   b. changed dependencies between registers (e.g. only universities of a specific category are allowed to conduct specific programmes of study);
   c. changed rules concerning the validation of data (e.g. a whole university can conduct only one course of study with a specific name, profile and level);
   d. integration with new sources of data (e.g. integration with Polish and European databases of patents, as well as with the database of the conservation rights of plant varieties),
   e. new categorisation of the data (e.g. a new list of scientific disciplines);
   f. new functionalities (e.g. calculating the cost-absorption of the courses);

g. new mode of data entry (e.g. students' data had to be entered by universities twice a year under the old law. Now however, they will have to be updated within two weeks of the change).

(ii) Development of new data registers such as doctoral schools; PhD degree candidates outwith schools; documents concerning scientific promotions (PhD, DSc, Professorship).

(iii) Integration with ORCID so as to provide a unique identifier to each researcher.

(iv) Development of an algorithm for the automatic assessment of the level of scientific excellence of universities and research institutes.

(v) Integration of the Uniform Antiplagiarism System with new data sources, i.e. the register of documents concerning scientific promotions.

(vi) Provision of access to new and redesigned registers for various governmental institutions dealing with higher education, as well as public access to open data.

(vii) Development of new statistics, reports and analyses.

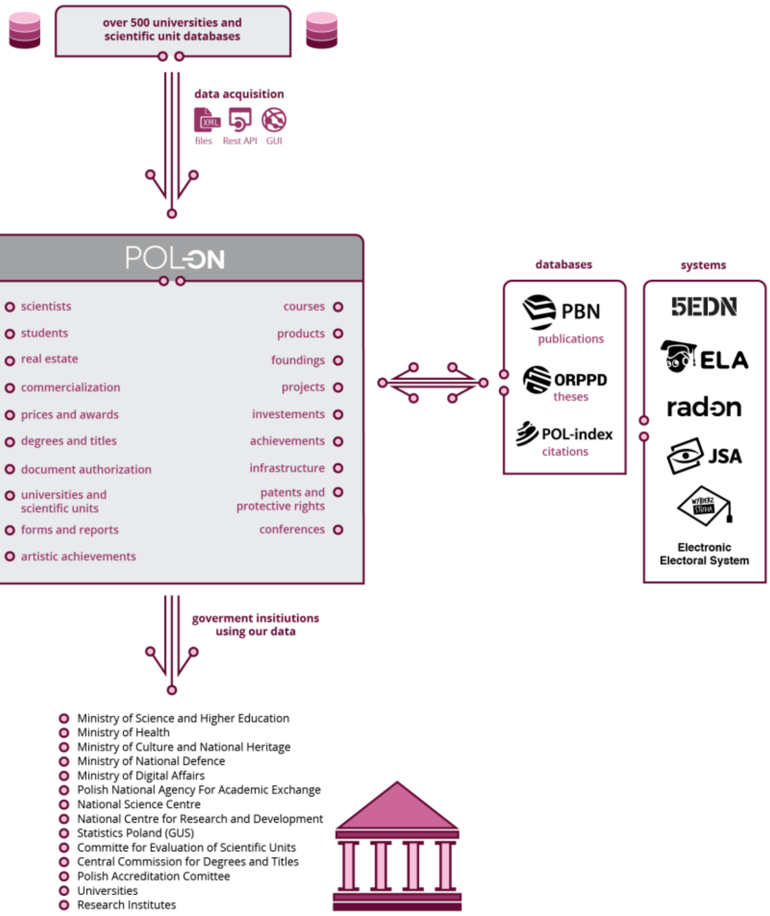

Figure 1. Architecture of the information ecosystem on science and higher education in Poland.

4. DISTRIBUTED INFORMATION ECOSYSTEM OF SERVICES

4.1. Introduction

The requirements depicted above are of crucial importance for the future architecture of the whole information ecosystem on science and higher education in Poland (Figure 1). Although a present system, POL-on, is mature, we must develop new components so as to ensure smooth data migration to the new registers. Such an approach gives the opportunity to use modern technologies and optimise business processes. As a result, we may shorten and simplify development, deployment and maintenance of the system. Amongst several improvements, we are planning to abandon the monolithic, multi-layered architecture with a central database, in favour of a distributed architecture of services. It will be composed of micro-services grouped into application, integration, and analytical layers (see Figure 2). The existing transactional systems will be replaced by light and distributed micro-services, which interacting together will provide more useful business outcomes for users. This paper presents the first experiences with the new technology stack, that has been used in the development and deployment of the application in the context of science and higher education.

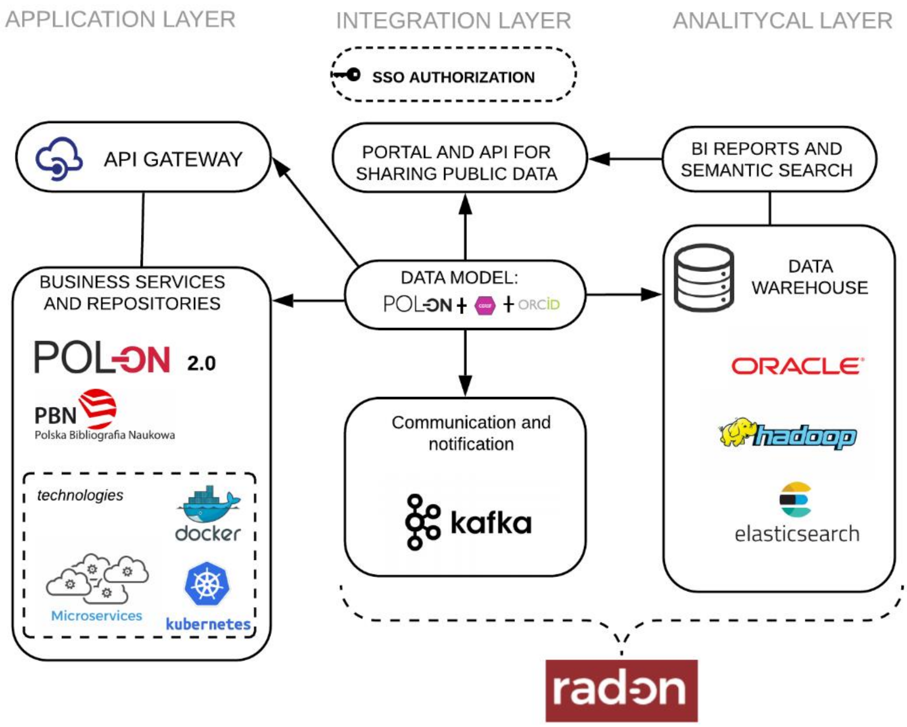

Figure 2. Vision of target architecture of the distributed system.

4.2. Motivation for change

The POL-on system is used for collecting a very wide range of data connected to science and higher education in Poland. The scope of data collection and the data model are defined by the legal framework and regulations, which tend to change frequently. Changes in the law generate a large number of modifications in the application and its data model. A lot of changes in the legal framework may create the need to perform significant rebuilding of the, but very often refer only to a few functional modules. Even if the change affects only one function, it is not possible to develop a new version of the application without integration with other modules, which lengthen the time required to reach the market. The source of many problems is the current size of the application, which makes it challenging to implement new versions of the system within a reasonable timeframe, and also causes delays in responding to users' requests. The application is divided into functional modules that are maintained and developed independently by different teams. In order to re-deploy the application in a production environment, the work of many teams has to be synchronised. This method of administering a complex software system requires a high degree of technical effort and frequent integration tests.

In the case of our monolithic architecture, scaling is performed by running more copies of the application. This solution does not solve the problem of accessing the database, because each copy of the application uses the same database. The greater the number of instances of the application, the greater the number of transactions must be handled by a single relational database. Overloading the database by calling too many instances of the application can affect system reliability. In practice, the system is not scalable due to database transactions. In addition, the one centralised database supports simple read and write operations and also various sophisticated reports used for decision support and analysis.

4.3. Paradigm shift in software design

The experience gained during the development of the previous version of the system showed that a domain model should be adaptable to evolution. The development team decided to use a new software development approach which is known as Domain Driven Design (Evans, 2004). The software development process assumes strong cooperation between cross-functional development teams with domain experts. The domain is discovered together by teams and domain experts using the Event Storming method (Brandolini, 2018). Event Storming is a workshop that makes domain exploration possible and more effective. As a result of workshops and cooperation between developers and domain experts, the development team divided the application into modules, which reflect the reality related to science and higher education. Domain experts are involved in a process of defining the boundaries of modules (bounded contexts), and identifying the real world objects (Figure 3) that should be reflected in the source code. In this way, the model of the domain is forged by all key participants.

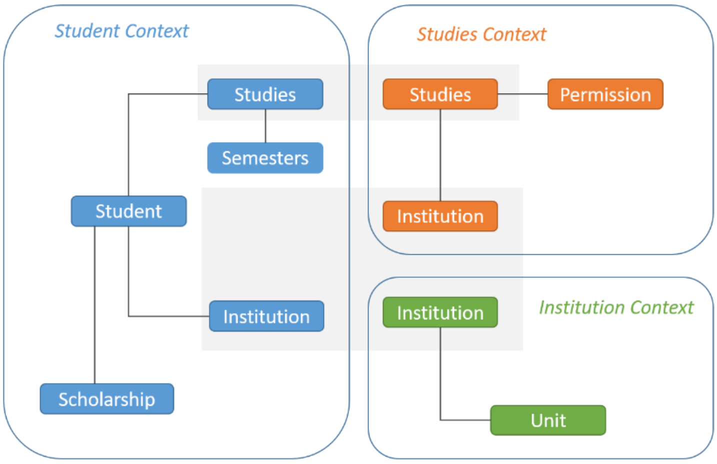

Figure 3. Bounded Contexts with unrelated concepts (student, permission, unit, scholarship) and shared concepts (studies, institution).

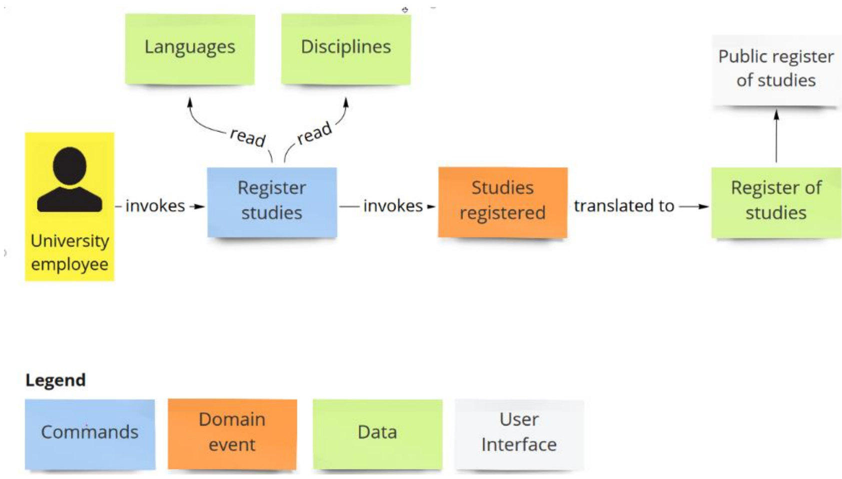

Figure 4. Simplified result of Event Storming regarding one domain event.

The domain model is used as a backbone of shared team language (ubiquitous language). Ubiquitous language is used by developers and domain experts to communicate with each other. The development teams divided the area of higher education into sub-domains using strategic design (Vernon 2013). This approach enables them to identify the core domain, supporting subdomain and generic subdomain. Significant resources are committed to developing the core domain, which is the most fundamental to business stakeholders. One of the main goals of Event Storming is to calculate how many domain events and business entities should be supported by the system (Figure 4). When the number of domain events is known, it is possible to determine how the states of the business objects change during their life cycle and which events cause these changes.

4.4. Architecture

One possible application of this approach is event sourcing, in which changes in the state of business entities will be stored as a sequence of event objects. In this way it will be possible to reconstruct the current or past state of each business entity by reading the event log (Fowler, 2005). When a service detects a data change, it will publish an event object. Each published domain event will be available and ready-to-use by all interested modules. Two main factors affect the reaction time of the system: one relational database in the infrastructure layer, and a large number of read operations. The main goal of creating a new system, besides compliance with new legal acts, is to create an efficient and user friendly application. To achieve this, we will use two different data models in each module: the first will be used to update information, and the second to read information.

To implement the domain model as working software, the development team map the bounded contexts to micro-services (Figure 5). In this approach, each bounded context is treated as a potential candidate for implementation as a micro-service. The new version of system will be a collection of independently deployable services. System development will be based on a few basic assumptions that are consistent with the philosophy of creating micro-services. It is expected that it will be a loosely coupled system, in which micro-services (modules) will have little knowledge about other components. The main advantage will be the ability to implement, test and maintain each module separately, which will reduce the time required to reach the market.

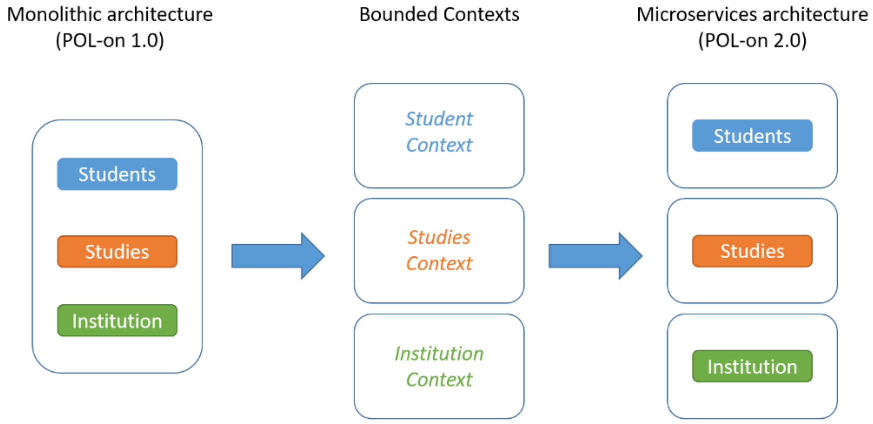

Figure 5 Migrating the monolithic system to an ecosystem of micro-services using Bounded Contexts.

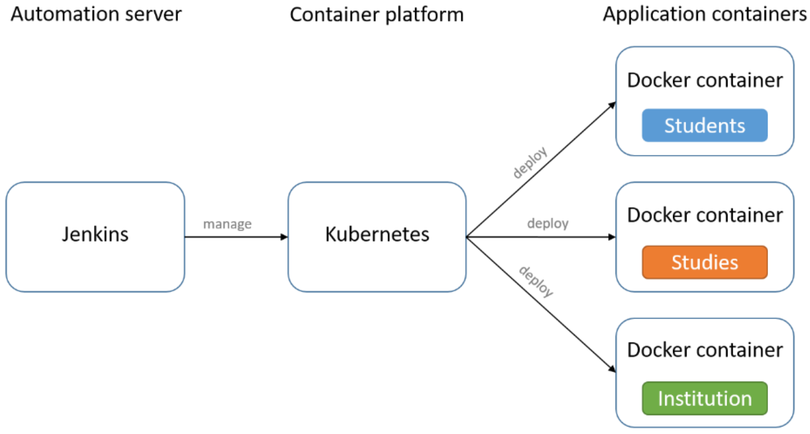

Figure 6 Target process of deploying micro-services using Jenkins, Kubernetes and Docker.

4.5. Technology background

The new version of the POL-on system will use Docker as a software tool to deploy and run modules and applications. The developer team will produce modules as containerised applications. In essence, this means that each new micro-service (module) will be created as a Docker container image (Figure 6), which will include everything needed to run an application: code, runtime, system tools, system libraries and settings ("What is a Container?", 2019). In our case, Kubernetes will be used in order to automate the deployment of containerised applications. Publishing domain events will be supported by Kafka, which will be used as a messaging tool to share domain events between modules.

4.6. Risks

New technology significantly increases the risk of inconsistencies between the system modules. These inconsistencies may be of two types:

- inconsistencies between the base module and dependent modules (e.g. if in the base module containing universities' data, the foundation date is later than the date of commencement of studies, stored in a dependent module containing student data, this situation is treated as inconsistency);
- inconsistencies in data shared by many independent modules (e.g. if the name of a person registered in the employee module is different from a name of the same person registered in the doctoral student module, this situation is treated as an inconsistency).

Due to the application of the system, the risk of both types of inconsistency should be minimised as far as possible. As to handling inconsistencies between the base module and dependent modules, the following approaches were considered:

- **Approach A.1:** neither in the base module nor in dependent modules would the system allow operations that could result in data inconsistencies (this approach is applied in the old system);
- **Approach A.2:** in the dependent modules the system would not allow the operations that would cause inconsistency with the base module. However, the data in the base module could be changed even if it would cause inconsistency in the dependent modules. In such situations inconsistent data in dependent modules would be marked as incorrect, and the system would provide tools to identify them, determine what operation caused the error, and correct them if appropriate. On the other hand, the base module would provide a mechanism for easily reversing any operation that could cause inconsistencies in the dependent modules.

As to handling inconsistencies in the data shared by many independent modules, the following approaches were considered:

- **Approach B.1:** entering the data into a detached module that would be a source of that data for the other modules;
- **Approach B.2:** entering these data independently in the modules and providing tools for reporting suspected inconsistencies and tools for unifying that data.

Approach A. 1 and B.1 eliminates many expected benefits from the new architecture (there are still maintained strong mutual dependencies between modules). Therefore, the approaches A.2 and B.2 are preferred.

5. WAREHOUSE INTEGRATING SERVICES THROUGH DATA

5.1. Introduction

The POL-on system has an independent analytics module (based on Envers[1]) but it has limitations and does not support cross-system analysis. As part of a transactional system this analytics module cannot be used in the Business Intelligence system. Using a reporting module may cause the efficiency of the main functionality of the POL-on system to decrease drastically. In order to ensure very high responsiveness of the user interfaces, we launched a new project in November 2018, The Integrated System of Services of Science – stage II (Michajłowicz et al. 2018), one of the main goals of which is the design and implementation of a data warehouse. The data warehouse will integrate the data from

---

[1] The Envers module is a core Hibernate model that works both with Hibernate and JPA. The Envers module aims to provide an easy auditing / versioning solution for entity classes.

various transactional systems without replacing it. The data will be converted into a form which allows optimal responsiveness of user interferences or web services, as well as generating various analyses and reports.

5.2. Architecture and technology background

The Polish information ecosystem on science and higher education has a highly heterogeneous data structure. Its ecosystem includes Oracle, MongoDB, PostgreSQL, Elasticsearch and some less crucial technologies. Using so many technologies causes a host of challenges in data analysis. End users expect cross-system analysis. They want to mashup data and access the most competitive insights as efficiently as possible.

Heterogeneous data sources demand a different approach during the integration process (Figure 7). In our ETL[2] process, we use well-known tools such as Oracle Data Integrator and Oracle Golden Gate, as well as home-made tools to integrate MongoDB or POL-on 2.0 events.

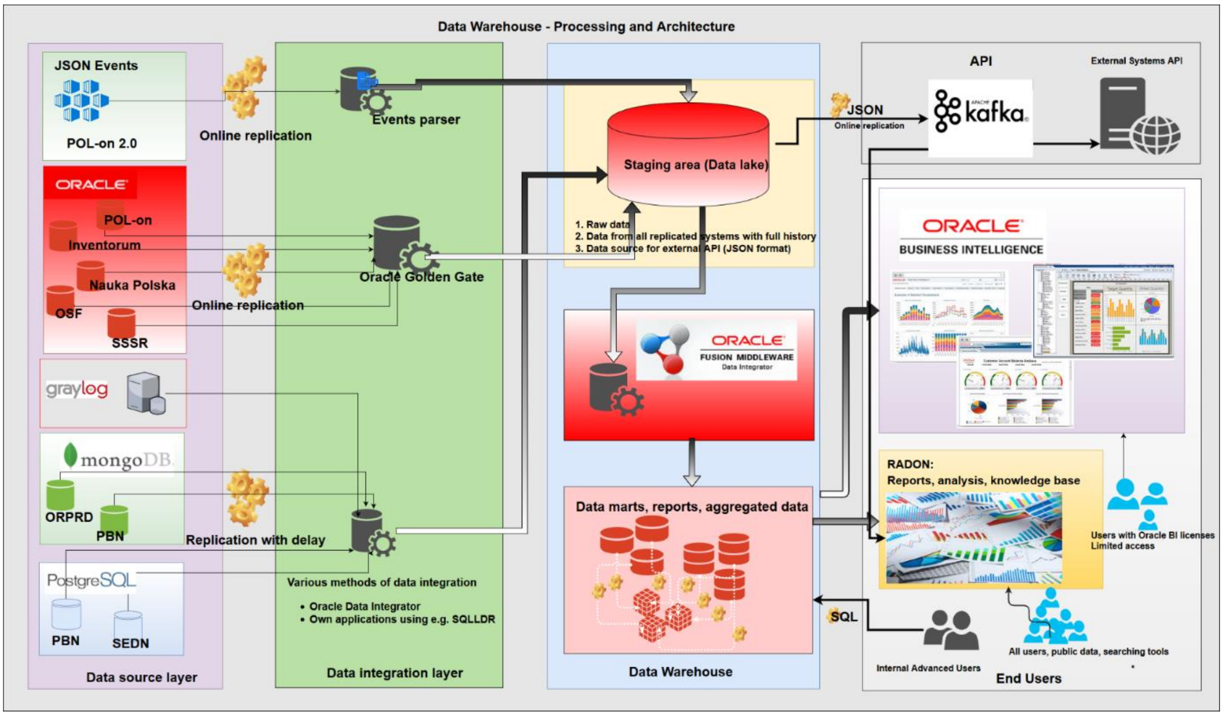

Figure 7. A simplified architecture of the data processing process.

In this paper we will focus on our own ETL tools. The new POL-on 2.0 architecture forced a complete redesign of our ETL process. Approximately 60% of the data in our database comes from the 'old' POL-on system, and includes the most important data concerning students, scientists, and institutions. In order to synchronise POL-on 2.0 with the warehouse we have designed and implemented '*Event Parser*' (Figure 8).

Every single event generated by POL-on 2.0 has to be transformed to Oracle DML[3] command. That approach provides near real time replications. Events after the transformation fill an application read models, and the next steps will be performed in conventional ODI[4] jobs. It is worth mentioning that at the beginning POL-on and POL-on 2.0 will have to work simultaneously, and some 'old' POL-on functionalities will have to be synchronised with the new ones. This creates additional work for the warehouse ETL tools.

[2] ETL - Extract, Transform and Load
[3] DML- Data Manipulation Language
[4] ODI - Oracle Data Integrator

In order to integrate MongoDB collections into our data lake, we have developed a tool which reads transaction logs from the Mongo local.oplog.rs collection and translates them to Oracle relational data format. This approach allows us to mash up datasets from two different worlds such as relational and object-based ones, in a simple manner.

In our warehouse we introduce a layer concept:

1. Data lake (staging) – the largest part of our data base which stores raw data from all integrated systems. The data is in 3NF[5] without any transformation or cleansing. It is a real copy of the data from production systems.
2. Thematic data marts – data after transformation and cleansing, divided into thematic areas. During the ETL process we denormalise the data, but avoid redundant duplication. We try to combine information from different systems to create a complete set of attributes in one row.
3. Reports - detailed reports that require additional space.

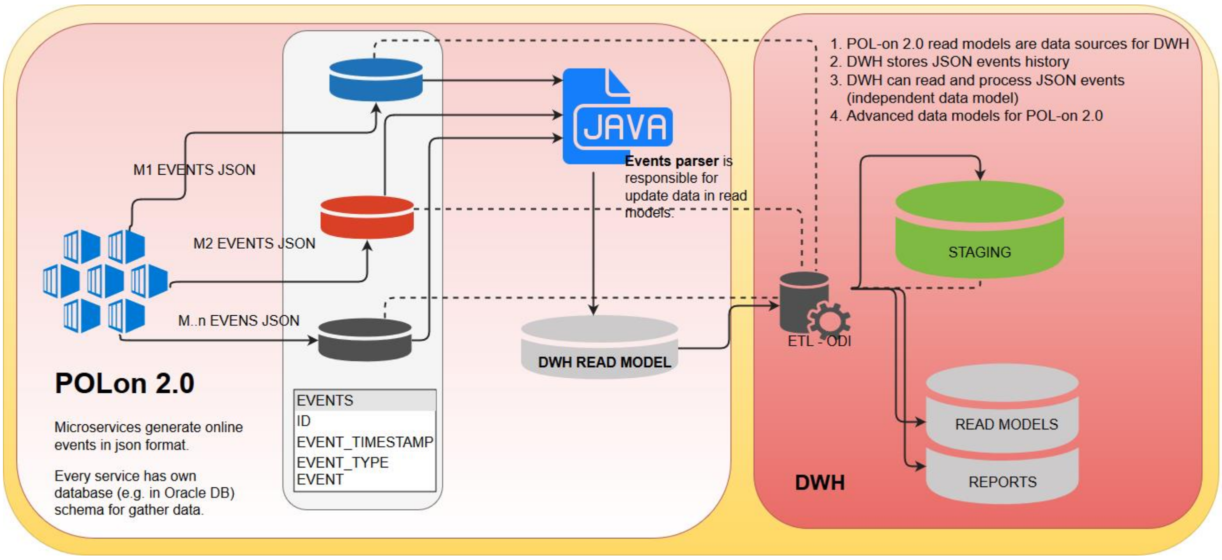

Figure 8. Events processing process.

5.3. Role of the data warehouse

One of the components of the RAD-on (Protasiewicz et al. 2019) system is a knowledge database and shared data service. These components require a special approach e.g. JSON[6] data format, due to the use of Elasticsearch, machine learning or AI[7] techniques. To fulfil these requirements we have prepared integration between the warehouse and Apache Kafka. In our case Apache Kafka is a data collector for JSON data generated directly from the warehouse. At the end of ETL process all changes in dedicated data warehouse models are transformed into a JSON structure and moved to Apache Kafka in almost real time. API REST services, an intelligent search engine and other applications can use the same data as the data warehouse or BI[8] users. In emergency situations we can recover the data warehouse using the JSON data stored in Apache Kafka.

Below are statistics (Figures 9 and 10) describing the RAD-on API, six months from launch. Currently we have five services ready to use. We share 7,744,505 documents and handle up to 8,000 API requests per month. The Uniform Antiplagiarism System is integrated with our warehouse, and reads students data from a dedicated Kafka topic. At the top of our analytic ecosystem we have the Business Intelligence tool. We decided to use Oracle Business Intelligence EE, because it is very flexible and provides many possibilities in analysing and visualising data.

[5] 3NF – Third Normal Form
[6] JSON - JavaScript Object Notation
[7] AI - Artificial Intelligence
[8] BI –Business Intelligence

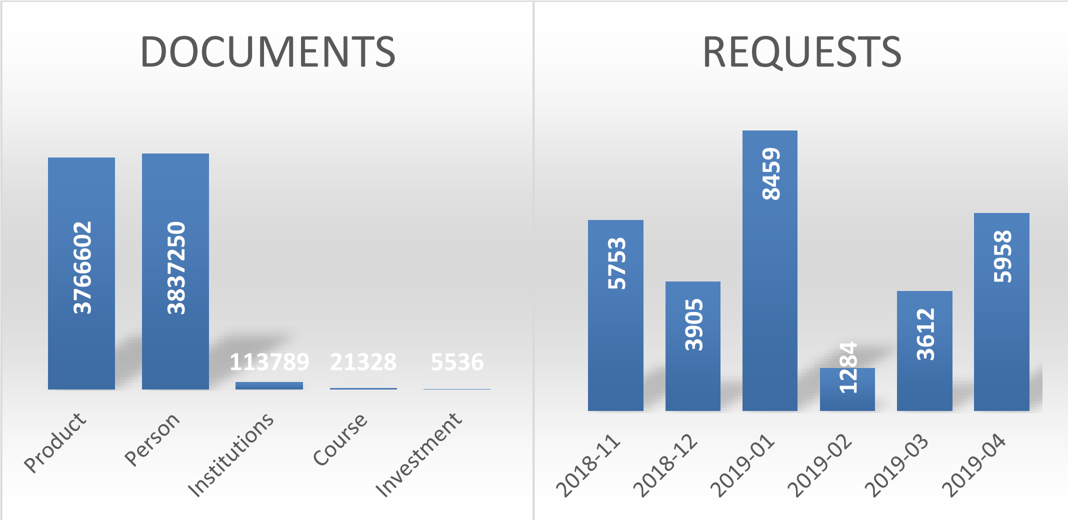

Figure 9. API utilization statistics

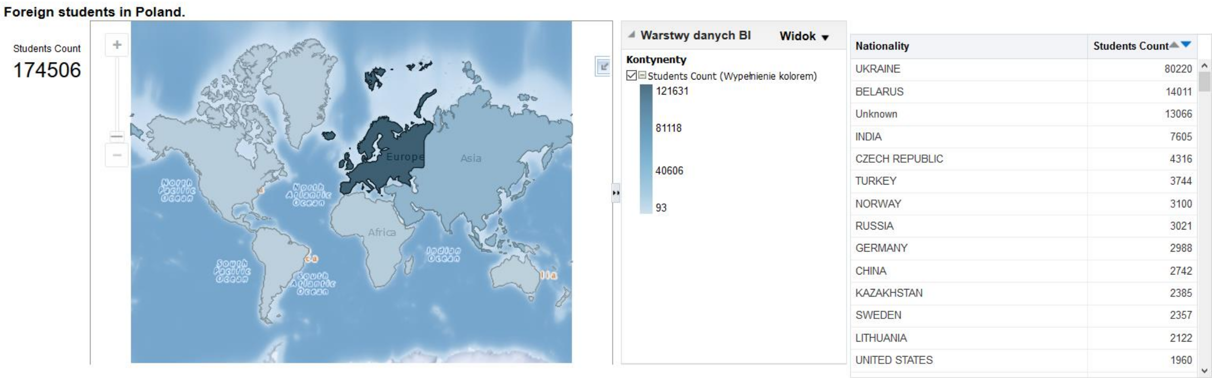

Figure 10. Oracle Business Intelligence report.

6. CONCLUSION

We have outlined an ecosystem of distributed services for science and higher education in Poland. It includes a data warehouse, which provides us with data marts for these services to make them faster. Moreover, business intelligence tools offer descriptive and predictive analytics, which helps policymakers make proper decisions.

The data warehouse filled the gap which existed previously in the Polish information ecosystem on science and higher education with analytical tools. It is crucial in the POL-on 2.0 ecosystem, because all reporting and statistics tasks will be moved to the warehouse. It will make POL-on 2.0 a very light, pure transactional system. In the context of RAD-on, the data warehouse provides ready-to-use and reliable data for intelligent applications and Open Data services.

Currently, we are developing the designed architecture. Its deployment will verify the assumptions we have made and presented in this study. We hope that in the future work we will be able to provide quantitative data demonstrating the ecosystem's performance, and share our experiences.

7. REFERENCES

Alfantookh, A. A., Ramadan, H. M., Hafez, A. E. M., Ykhlef, M. A., AlShibli, A. J., & Benchikh, M. A. (2012). A Decision-Support System for the Saudi Ministry of Higher Education. *International Journal for Computers & Their Applications*, 19(4).

Aljawarneh, I. M. (2016). Design of a data warehouse model for decision support at higher education: A case study. *Information Development*, *32*(5), 1691-1706.

Ayesha, S., Mustafa, T., Raza Sattar, A., Khan, M. I. (2010). Data mining model for higher education system. *European Journal of Scientific Research*, 43(1), 24-29.

Baranovic, M., Madunic, M., & Mekterovic, I. (2003, June). Data warehouse as a part of the higher education information system in Croatia. In *Proceedings of the 25th International Conference on Information Technology Interfaces, 2003. ITI 2003.* (pp. 121-126). IEEE.

Benefelds, J., & Niedrite, L. (2004). Comparison of Approaches in Data Warehouse Development in Financial Services and Higher Education. In *ICEIS (1)* (pp. 552-557).

Bhise, R. B., Thorat, S. S., & Supekar, A. K. (2013). Importance of data mining in higher education system. *IOSR Journal Of Humanities And Social Science (IOSR-JHSS)*, *6*(6), 18.

Boulila, W., Al-Kmali, M., Farid, M., & Mugahed, H. (2018). *A business intelligence based solution to support academic affairs: case of Taibah University*. *Wireless Networks*, 1-8.

Brandolini, A. (2018). *Introducing event storming*. Retrieved 04/17/19, from leanpub.com/introducing_eventstorming.

Evans, E. (2004). *Domain-driven design: tackling complexity in the heart of software*. Addison-Wesley.

Fowler, M. (2005, December 12). *Event Sourcing*. Retrieved 04/17/19, from www.martinfowler.com/eaaDev/EventSourcing.html.

Fowler, M. (2011, July 14). CQRS. Retrieved 04/17/19, from martinfowler.com/bliki/CQRS.html.

Mattingly, K. D., Rice, M. C., & Berge, Z. L. (2012). Learning analytics as a tool for closing the assessment loop in higher education. *Knowledge Management & E-Learning: An International Journal*, *4*(3), 236-247.

Mekterović, I., Brkić, L., & Baranović, M. (2009). Improving the ETL process and maintenance of higher education information system data warehouse. *WSEAS transactions on computers*, *8*(10), 1681-1690.

Menolli, A. L. A., & Dias, M. M. (2004). A Data Warehouse Architecture for Brazilian Science and Technology Environment. In *ICEIS (1)* (pp. 522-525).

Menolli, A. L. A., & Dias, M. M. (2006). A Data Warehouse Architecture in Layers for Science and Technology. In *SEKE* (pp. 162-165).

Michajłowicz M., Niemczyk M., Protasiewicz J., Mroczkowska K. (2018). POL-on: The Information System of Science and Higher Education in Poland, In *EUNIS 2018 Congress Book of Proceedings*.

Miranda, E., & Suryani, E. (2014). Implementation of datawarehouse, datamining and dashboard for higher education. *Journal of Theoretical & Applied Information Technology*, *64*(3).

Mohammed, M. A., & Anad, M. M. (2014, September). Data warehouse for human resource by Ministry of Higher Education and Scientific Research. In *2014 International Conference on Computer, Communications, and Control Technology (I4CT)* (pp. 176-181). IEEE.

Moturi, C. A., & Emurugat, A. (2015). Prototyping an academic data warehouse: case for a Public University in Kenya. *Current Journal of Applied Science and Technology*, 550-557.

Protasiewicz, J., Rosiak, S., Błaszczyk, Ł., Niemczyk, M., Michajłowicz, M., Kucharska, I., Podwysocki, E., (2019). RAD-on: An integrated System of Services for Science - Online Elections for the Council of Scientific Excellence in Poland

Salaki, R. J., & Ratnam, K. A. (2018, March). Agile analytics: Applying in the development of data warehouse for business intelligence system in higher education. In *World Conference on Information Systems and Technologies* (pp. 1038-1048). Springer, Cham.

Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., & Peterson, L. (2007, March). Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. In *ACM SIGOPS Operating Systems Review* (Vol. 41, No. 3, pp. 275-287). ACM.

van Dyk, L. (2008). A Data Warehouse Model for Micro-Level Decision Making in Higher Education. *Electronic Journal of e-learning*, *6*(3), 235-244.

Vernon, V. (2013). Implementing domain-driven design. Addison-Wesley.

Vernon, V. (2016). Domain-driven design distilled. Addison-Wesley Professional.

What is a Container? (2019). Retrieved from www.docker.com/resources/what-container.

8. AUTHORS' BIOGRAPHIES

Emil Podwysocki (MSc) received a Master degree in Telecommunications Systems at the Technical University of Lodz. He has 10 years of professional experience related to ETL/ELT, data warehouses and Business Intelligence. His areas of interest include Oracle technology, Big Data, Business Intelligence and data visualizations. Currently, he works as a Database and Business Intelligence Team Leader in the National Information Processing Institute.
Email: emil.podwysocki@opi.org.pl
Profile: www.linkedin.com/in/emil-podwysocki

Łukasz Błaszczyk (MSc) is a product owner and business analyst at the National Information Processing Institute. He earned his Masters degree in Forest Information Technology at the University of Applied Sciences in Eberswalde and a master's degree in Forestry at Warsaw University of Life Sciences – SGGW. He finished postgraduate studies in business analysis at SGH Warsaw School of Economics. He has 11 years of experience in business and systems analysis.
Email: lukasz.blaszczyk@opi.org.pl

Marta Niemczyk (MSc) received a Masters degree in Mathematics at the University of Warsaw. She has many years of professional experience as a business and systems analyst, endorsed by certificates in methods of IT systems modelling. Currently she works as a software designer and business analyst in the National Information Processing Institute.
E-mail: marta.niemczyk@opi.org.pl

Jarosław Protasiewicz (PhD) is an assistant professor at the National Information Processing Institute and the head of the Laboratory of Intelligent Systems. He received a Ph.D. in computer science at the Systems Research Institute of the Polish Academy of Sciences. His areas of interest include agile project management, software design and development, big data, machine learning, and bio-inspired algorithms.
Email: jaroslaw.protasiewicz@opi.org.pl

Marek Michajłowicz (MSc) is the Deputy Head of the Laboratory of Intelligent Systems. He received a Master degree in Sociology at Cardinal Stefan Wyszyński University in Warsaw and a Bachelor of Engineering (B.E.) in computer science at Warsaw School of Information Technology under the auspices of the Polish Academy of Sciences. He has several years of professional experience related to business and systems analysis. Since 2014 he has worked as the project manager of the POL-on system in the National Information Processing Institute. His areas of interest include agile project management, software design and development, big data and warehouses.
E- mail:marek.michajlowicz@opi.org.pl;
Profile: www.linkedin.com/in/marek-michajlowicz

Sylwia Rosiak (MSc) is a business analyst at the National Information Processing Institute. She received a Masters degree in Management at the University of Warsaw and completed postgraduate studies in designing information systems at the Warsaw University of Technology.
Email: sylwia.rosiak@opi.org.pl

Iwona Kucharska (MSc) received a Masters degree in Computer Science and Econometrics at the University of Computer Science and Economics in Olsztyn. She completed postgraduate studies in Business Processes Engineering and Business Intelligence at the Warsaw University of Technology. She has many years of professional experience as a software tester and as a business analyst, endorsed by certificates. Currently she works as a software designer and business analyst in the National Information Processing Institute.
E-mail: iwona.kucharska@opi.org.pl