Adaptive Software System for International Activity Level Assessment

Oleksandr V. Koval [1], Valeriy O. Kuzminykh [1], Iryna I. Husyeva [1], Xu Beibei [2] and Zhu Shiwei[2]

[1] *National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, 03056, Ukraine*
[2] *Information Institute, Qilu University of Technology (Shandong Academic of Sciences), Jinan, 250316, China*

Abstract
The paper discusses methods of building event-driven architecture systems for international activity level assessment. We propose a system structure that enables adaptive changes in the directions of information collection and processing based on the evaluation of the procedures' efficiency for obtaining information from relevant sources. We use typical scenarios to select directions for data collection and processing, which simplifies the analyst's task of selecting actions. The event-driven system architecture enables adaptive operation of the system based on the load of individual microservices. The architecture allows us to analyze the generated events and, on their basis, create new instances of containers to increase the overall efficiency of the system. Simultaneously, this approach simplifies the structure of service management and increases the efficiency of information collection and processing. The approach can be applied to develop information and analytical systems for various purposes.

Keywords
microservices, adaptation, event-driven architecture

1. Introduction

One of the most important and relevant directions for determining the development level of scientific and educational institutions, individual scientific groups, and individual scientists is the constant determination and forecasting of the efficiency indicators of their scientific activity state. Scientific activity is evaluated based on these indicators at the institutional, national, and international levels, in comparison with similar institutions. The evaluation process takes into account the current state of these indicators.

These actions take place in the form of creation and analysis of various rating systems based on defined sets of scientific activity indicators. This process enables the assessment of the current situation and prediction of influential indicators of scientific activity efficiency for the future. In addition, it becomes an important factor in determining the scientific position of the institution within the country and in the world.

To achieve successful promotion in the international scientific community and increase cooperation with foreign scientists and teachers, scientific organizations and universities, it is necessary to assess the international recognition level of the scientists being evaluated. This is especially important when investing in scientific activity both from the state and non-governmental organizations. It is important to consider the relevance, quality, and international citations of scientific papers, which must meet international academic requirements.

Presently, the concept of international activity for universities and research institutes does not have a clear definition. In most cases, both the number of criteria and indicators used for assessment and the

XXII International Scientific and Practical Conference "Information Technologies and Security (ITS-2022)", November 16, 2022, Kyiv, Ukraine
EMAIL: avkovalgm@gmail.com (O.V. Koval); vakuz0202@gmail.com (V.O. Kuzminykh); iguseva@yahoo.com (I.I. Husyeva); xubeibei1987@163.com (X. Beibei); zhusw@sdas.org (Z. Shiwei)
ORCID: 0000-0003-0991-6405 (O.V. Koval); 0000-0002-8258-0816 (V.O. Kuzminykh); 0000-0002-8762-3918 (I.I. Husyeva); 0000-0003-1430-5334 (X. Beibei); 0000-0002-6651-8449 (Z. Shiwei);

method of calculating the complex criterion depends on the goals and objectives of the analysis set by the interested party or the customer.

There are many international activity level indicators, including foreign students and teachers, international educational programs, and internships abroad. But among the most important indicators of scientific organization level, a group of scientists, or a specific individual scientist are the number of scientific publications, the level of domestic and foreign journals, references in top international bibliographic databases, and the number of citations of scientific publications placed in international bibliographic databases. At the same time, as a rule, the most important indicator is the level of international bibliographic databases.

Information about scientific publications is available in many global bibliographic sources, which are regularly replenished with new scientific papers. Information about other types of scientific activities of institutions, especially international, is difficult to access, is not consolidated in specific resources, and is not, in most cases, sufficiently completed and up-to-date. Therefore, consolidated information which describes the international activity of scientists based on the analysis of their publications can give a more or less accurate basis for assessing the comparative level of their international activity.

To address this issue, a software system is needed to collect, analyze, and consolidate data from open bibliographic systems and local databases. The system should provide conclusions about the comparative levels of international activity for scientific organizations and individual scientists.

Among the users of such software systems could be universities, and research institutes, as well as individuals, state institutions, and other organizations which invest funds into scientific research.

The information and analytics received will help scientific institutions to determine and monitor their position in the rating of international activity; state institutions - to direct the necessary attention to the level of research activity; investors - to create a complete investment picture regarding institutions and scientists, and individual scientists - to determine distribution and citation of their publications in global bibliographic sources.

The main goal of the system is to provide users with up-to-date and complete analytical information on international activity indicators from selected sources.

The main tasks of the system are:

- collecting data from sources diverse in structure and composition, mainly bibliographic and abstract scientometric databases,
- bringing to a common form and aggregating the received data according to the user's request,
- removal of redundant erroneous data and data duplicates.

The system is designed to accommodate a changing number of sources that may differ in data structure and interaction interfaces. Therefore, it requires a flexible architecture that can be modified easily and with minimal costs when new sources and functions need to be added or when existing functions need to be modified.

2. Architectural solutions

The construction of information and analytical systems is characterized by the need to implement solutions that require the integration of many functions and services. Classical monolithic solutions no longer meet the requirements of flexibility, fault tolerance, and efficiency. Theoretically, most of the principal architectural solutions were proposed decades ago [1,2]. However, with the development and dissemination of modern technological capabilities, new opportunities for developing architectural solutions have emerged.

Today, the most common architectural solutions are:

- Layered Architecture,
- Tiered Architecture,
- Service Oriented Architecture,
- Microservice Architecture,
- Event-driven architecture.

Layered Architecture (LA) is based on the division of responsibility and consists of three layers: Presentation Layer, Business Logic Layer, and Data Link Layer. The advantages of LA are: simple implementation, abstraction due to the division of responsibility, protection of layers, and increased controllability due to reduced coupling. But it does not offer scalability, or a monolithic structure, and involves passing data through each layer.

Tiered Architecture (TA) distributes responsibilities between data providers and consumers and divides software according to the 'client-server' principle. The Request-Response template is applied to communicate between the layers. The architecture offers horizontal and vertical scaling.

Service Oriented Architecture (SOA) applies several models which connect components and applications with well-defined services. The architecture has the following main components: Services; Service Bus; Service Repository; SOA Security; SOA Governance, and considers three types of main participants: service provider, service consumer, and service registry.

The basis of the service-oriented architecture is the service bus (enterprise service bus, ESB), which processes requests according to a standard protocol and data format. The service architecture distinguishes two main types of services: Atomic services – those that are not subject to decomposition and Composite services – those that combine atomic services to build complex functionality. The SOA architecture is built on several models proposed by the W3C consortium: Message Oriented Model, MOM; Service Oriented Model, SOM; Resource Oriented Model, ROM; Policy Model, PM; Management Model, MM.

Microservice architecture is a type of service-oriented architecture, which is built on separate isolated components - micro-services, and consists of the following components: Services; Service Bus; External configuration; API Gateway; Containers. The properties of microservice architecture include division into service components, focus on business needs, emphasis on the product rather than projects, decentralized management and data management, infrastructure automation, protection against failures, evolutionary design, smart endpoints, and dumb pipes. The division of microservices is based on the approach of Domain-Driven Design, DDD. Orientation on the needs of business implements the practice of team building following the microservice architecture.

Containers are created based on templates, which represent an approach formed at different levels. Two approaches are used to coordinate transactions in the microservice architecture: choreography and orchestration.

The microservice architecture of information systems comprises a set of independently deployed services, and a system designed in such a way can evolve in parts since each microservice is autonomous. In addition, such architecture allows flexible scaling of information system components, thereby ensuring optimal use of available server equipment. At the same time, each microservice is scaled independently. It became possible to allocate computing resources taking into account real user activity and demand for system functions.

Microservice architecture [3] is an approach to software development, which implies the rejection of a single, monolithic structure. This is one of the approaches to develop software applications by creating separate and independent software modules. Each of them is responsible for a certain task, and can be changed, supplemented, and expanded.

A key feature of the system is the adaptation of its parameters according to the user's request. Namely, depending on the request, one of the possible ways of extracting and processing data is formed, according to the features of the request and capabilities of the system to process this request. This is achieved due to a generalized flexible data model, event-driven, microservice architecture, and orchestration of software services.

Event-driven architecture (EDA) is a type of software architecture that takes into account events and reactions to them. An event is interpreted as an action that initiates a message, a change in the application, or an event - a significant state change. In practice, event-driven architecture is also considered as a logical development of an adaptive superstructure over micro-service architecture. In software systems with event-driven architecture, the core of the system is the database. In EDA, the focus shifts to events and the way they are handled by the system architecture. The logic of event architecture is built according to two types of topology - Broker and Mediator - which are named after programs that connect the generator and consumer of events. The main elements of the topology are: Event queues, Event mediators, Event channels, and Event processors.

The main advantages of EDA include real-time results output, lower data transmission delays, higher throughput, simple scalability, high fault tolerance, and availability.

Modern event-driven microservice architecture still lacks universal solutions for efficient performance, and complex tasks require additional, related knowledge. This is particularly relevant in the construction of information and analytical systems. Research in this direction has significant potential.

New architectural solutions based on micro-services have brought the analytical, management, and control capabilities of organizations with a complex distributed structure to a qualitatively new level, offering the possibility of building an information environment that is maximally adapted to their structure and processes.

3. Implementation of the system

An adaptive system was taken as the basis for the implementation of the system's event-driven microservice architecture with the definition of services according to functional purpose. It is necessary to dynamically expand the functionality of the system without significantly modifying the existing system elements. This allows for greater flexibility in scaling, and increases both the efficiency and throughput of the system.

This approach gives the following important benefits [4]:

- resistance to service failures;
- the possibility to develop software in different programming languages;
- simple scalability;
- independence from changes in individual services;
- data security and control;
- the possibility of partial implementation on customer's side;
- the possibility to develop parts of the system by separate distributed teams of developers.

The general functional structure of the software system (Figure 1) includes several basic elements that reflect the main functions and connections between individual elements of the structure. The system is built based on WEB access to the main user functions.

The management of the system interface distinguishes two main functional elements of the system:

- administration interfaces,
- analyst interfaces.

Administration interfaces perform the following main functions:

- granting access rights,
- system security policy implementation,
- connecting new software modules,
- testing new software modules,
- connecting new information sources,
- formation and connection of new search and data processing scenarios,
- testing and adding new scripts to the standard script repository.

Within the system framework, script management [5] implements the functions of choosing a standard procedure or forming a new scenario based on access to information sources and the system's functional capabilities for processing information and providing results in a specified form.

The results obtained from the information sources specified in the scenario are pre-processed, checked for relevance, and uniqueness, and consolidated in the database of search results.

This information, after appropriate processing, is transferred to the analyst's interface and displayed in the form specified in the data collection and processing scenario.

Further detailed processing of the received data, which is necessary for the analyst, is carried out within the functions defined by the capabilities of the analyst's interface.

Bibliographic databases such as Scopus, Core, arXiv, DBPL, Web of Science, and regional databases are used as information sources.

In addition, the data of the analysis includes data from internal information bases of organizations' scientific data, data from questionnaires on the activities of scientists, and data on participation in scientific contracts.

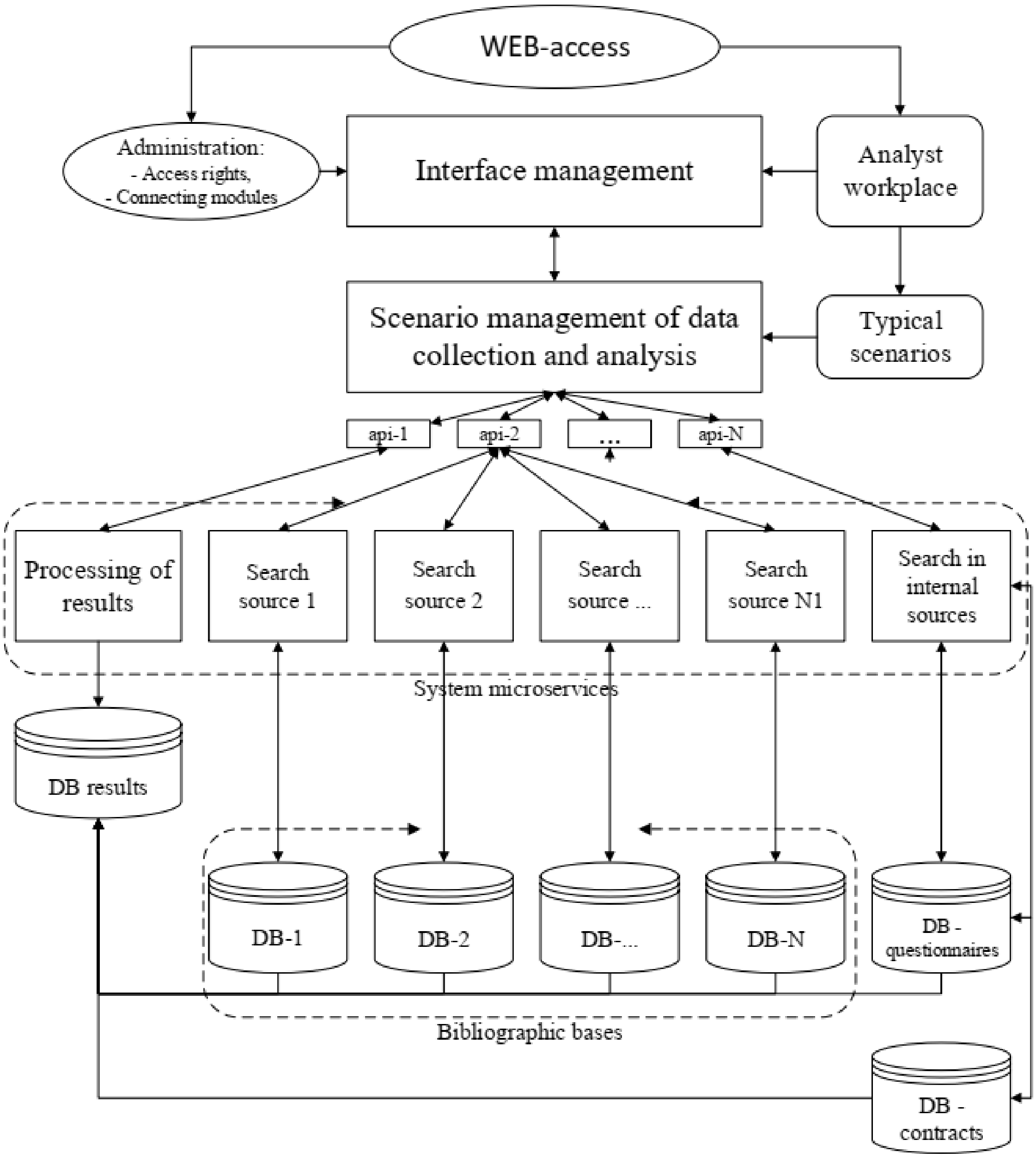

Figure 1: Functional structure of the system

The interaction of software services in the system and the organization of individual information flow is based on an adaptive event-driven architecture. This makes it possible to maintain high efficiency and throughput of the system for processing requests with a large number of users and processing complex requests, both from the point of view of big data volumes and complex types of processing and analysis of this information.

An event is a message of a fixed structure, which will be sent to a certain section in the message broker, with addresses for all services that are subscribed to update the message queue for a certain section.

With this approach to building the architecture, it is possible to obtain results in full according to the tasks set by the analyst. Despite the complexity and volume of the request, the overall load on the system, the analyst (user) will receive the correct data which was specified in the request [6].

The main advantages of the presented adaptive event-driven architecture are as follows:

- the ability to process information streams in real-time;
- possibility of parallel data processing;
- system scalability;
- speed of response to the analyst's request;
- adaptation to the quantitative assessment of the information relevance obtained during the processing of the analyst's requests.

The architecture of the adaptive event-driven system (Figure 2) is built according to the characteristics of data sources and information processing methods, which are determined by the possible needs of analysts.

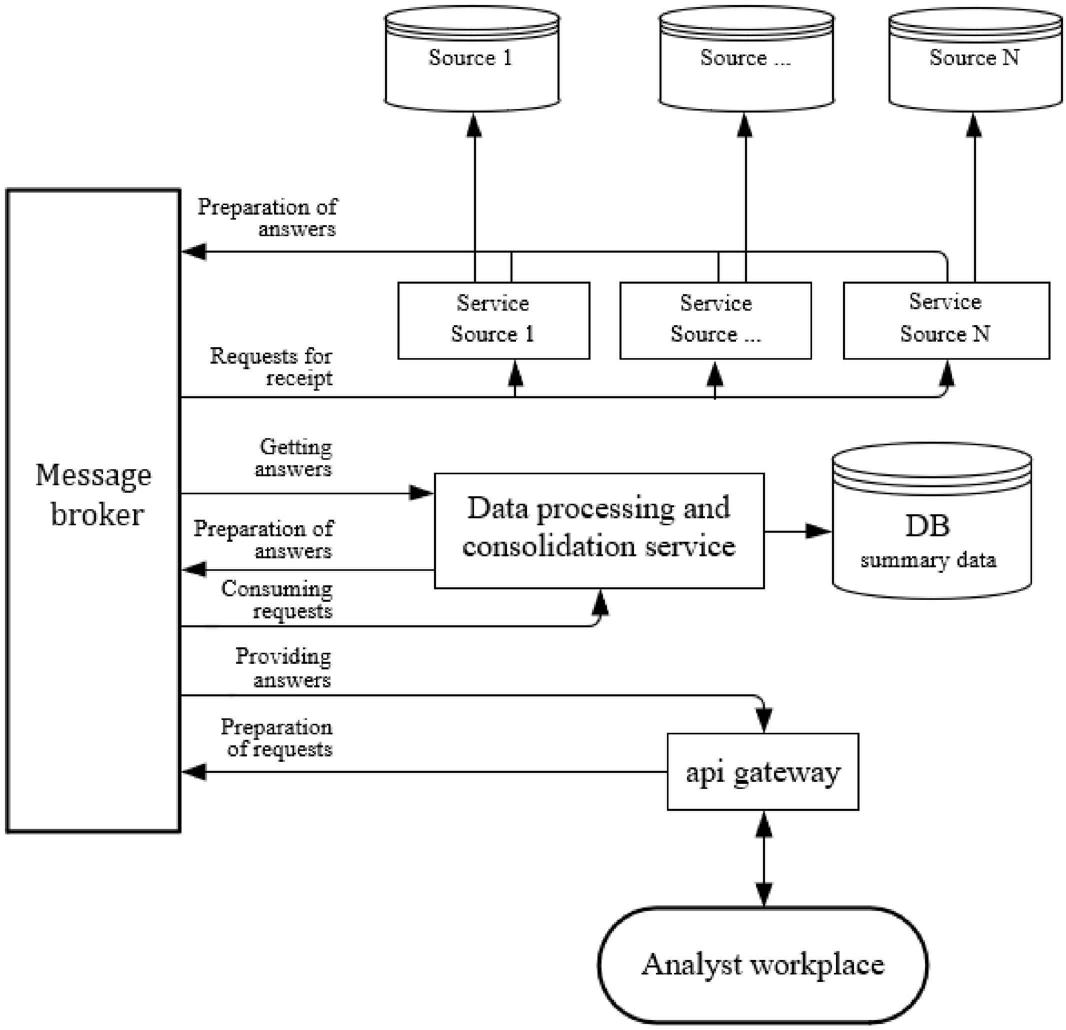

Figure 2: Architecture of an adaptive event-driven system

Features and quantitative composition of data sources do not prevent partial implementation of the system, and the structure of the system implementation allows for an increase in the capabilities of the system as needs grow. The number and features of data sources affect only the implementation of certain applications aimed at working with certain sources. The implementation of separate services for processing results can also be carried out as needed.

The adaptive features of the system when collecting information from external sources are based on the processing and analysis of information on the relevance of the results obtained from the external sources (usually bibliographic databases). Such information makes it possible to adjust the number of requests to these databases proportional to the relative stochastic evaluation [7] relevance, thus significantly improving the efficiency of collecting such information according to the request. The implementation of adaptive event-driven management of the information collecting from external sources, such as bibliographic databases, is based on the use of a linear stochastic automaton as a method that affects the formation of events that implement chains of requests to data sources [8].

The diagram of the actions sequence (Figure 3) describing the processing of the user's request for obtaining certain analytics on the publications of Figure 2 shows the general scheme of processes event management in the system.

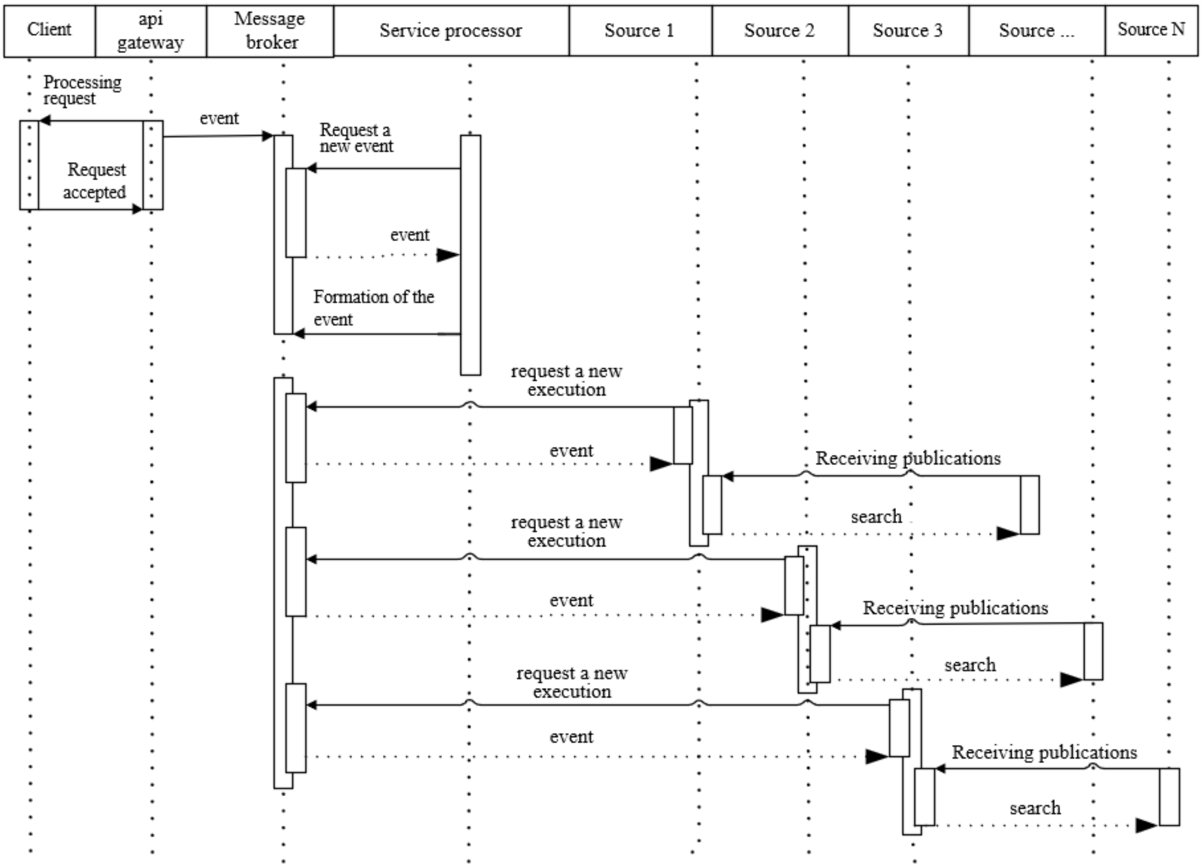

Figure 3: Diagram of the sequence of events

As shown in the diagram, the analyst forms a request to receive data for analysis, for example, publications by author group or the author, to the software interface of the system gateway via a socket channel. The gateway, having accepted the request, generates an event. After that, a message is formed, which is queued in the message broker.

In the next step, the service processor checks the existence of relevant data in the database, which stores previous query results. In this case, when the data is available, the event link is immediately formed to return the query results. In the opposite case, when there is no data, or it is partially insufficient for the completeness of the result, another event is formed and sent about the need to collect certain data according to the corresponding parameter of the request, for example, a publication in a scientific direction.

Source services (Figure 2), assigned to the collection event, extract relevant information from the databases associated with them. Their task is to collect requested data from specific external resources. Each individual provider knows how to interact with its data source - a remote resource that represents a bibliographic or scientometric base.

Data collected from this kind of resource is independent of each resource and corresponding interface. In most cases, it is data collected through an open software interface or a conditionally accessible software interface. If it is impossible to use the software interface, then the option of parsing such web resources is applied through the implementation of an additional service with parser functions.

The result of data collection can be obtained through a sequence of events in the system. Service processors (Figure 3) process the result according to the business logic defined for them. It can be data merging, additional search by parameters, search by keywords, reduplication, and others.

The result of this processing is sent back to the gateway, from where it reaches the client's side in the form of an event and in the format according to the exchange interface.

The client receives data based on requests as they are generated and received, re-sending the same requests does not require additional processing and immediately returns the result. With this approach, it is possible to ensure the independence of each component of the system. This provides opportunities for simple changes in the composition of services, scaling, and reduction of inactivity periods of the entire system.

An important feature is an ability to add new data sources and data processing methods without interfering into the operation of other system components. In general, each service can process requests differently, depending on the processing scenario [5] and the completeness of available data in a particular source, making the system adaptable to each individual user request.

Data obtained from various external sources are consolidated into a single generalized database (Figure 4). Each individual external data source has information in its own, often unique, format. At the same time, the subject area itself has a clear limited set of concepts and entities, such as publication, publisher, date of publication, place of publication, the language of publication, scientific institution, etc.

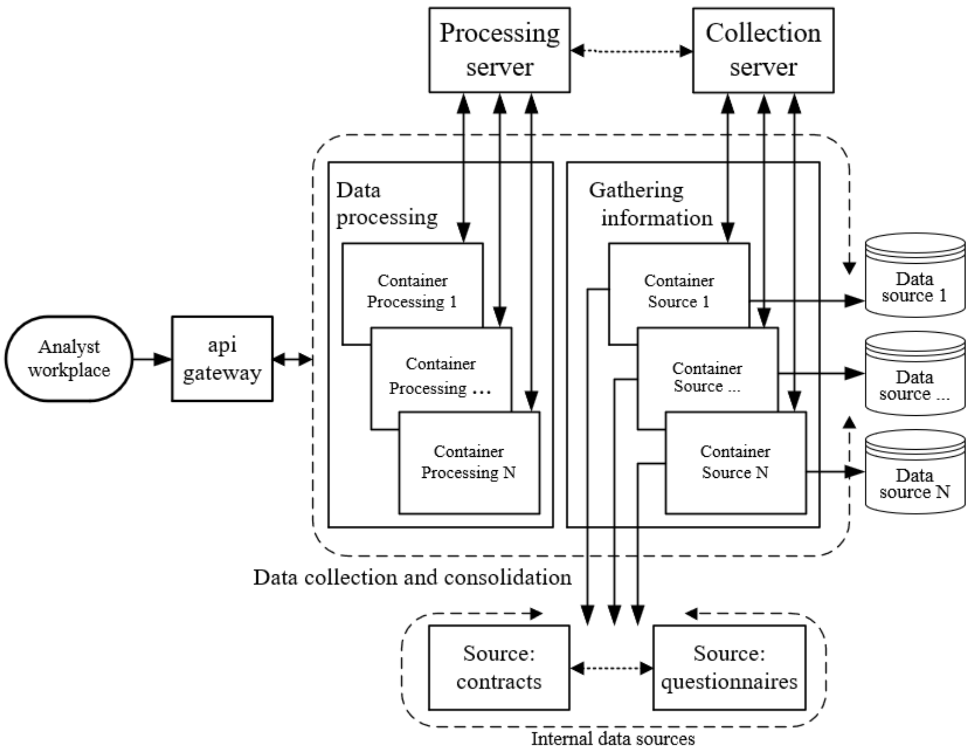

Figure 4: Data consolidation scheme

All data is transported in the system in a clearly defined format for consistent operation of all system components regardless of their functions.

The database stores an already normalized data model with established relations between them. A relational database can be used as a database. This makes it possible to reduce the amount of permanent memory required to store a large amount of processed data.

The structure of user requests has a convenient format to ensure the flexibility of requests in the form of key-value pairs within the JSON format. At the same time, with the help of keys, it is possible to set the display form, filter by fields or aggregation results, sorting, etc.

The software system is implemented using application containerization and orchestration software, as described in [9]. The use of such means for organizing the interaction of applications is based on the need for clear communication of software services, the need to make the system invariant to the means of implementing software applications, and the desire to create a secure communication structure within the system. In addition, this provides opportunities for monitoring the operation of software services and adjusting the scalability of the system.

The following advantages of using containerization and container application orchestration tools can be highlighted:

- application portability;
- system scalability;
- faster deployment of applications;
- high productivity and flexibility of application development;
- increased security of communication within the system;
- system continuity;
- ease of managing containers.

An important feature is the resource management settings for each of the services. The system adaptively responds to changes in hardware resources that are consumed by each software service in real-time. An increase or decrease in hardware load indicates that the number of requests or their complexity is changing. If at the same time there is no longer an instance of this service in the system that would decrease the load, then it is necessary to start another instance of the service to reduce the load.

In practice, the number of instances of each service is limited only by the capabilities of the system technical means. This approach implements the system's adaptive qualities to changes in the requirements for data collection and processing resources. The general diagram of the operation of software services (both own and third-party) in the container management system is shown in Figure 4. The operation of containers, both with external sources and with internal ones, is built based on relevant typical scenarios, which are an integral part of managing data collecting and processing [10].

Cloud technologies also provide opportunities for the continuous automatic expansion of computing resources. In this case, the use of self-configured system nodes will require constant support of available system resources required for a specific request.

A system built in this way can adapt and scale according to the load, expand its functionality without blocking the operation of other system components, and be built using multilingualism.

4. Conclusions

The software novelty lies in the creation of an adaptive software system that collects and processes consolidated information on the international activities of individuals and institutions. The system can flexibly expand its information sources based on user needs.

Various scientific institutions, institutes, universities, research organizations, state institutions, and other organizations that finance scientific research, as well as individual scientists, are among the users of such systems. This information and analytics enable scientific institutions to monitor their position in world rankings: state institutions can direct necessary attention to the level of research activity, investors can create a complete investment picture regarding funding relevance, and individual scientists can determine the distribution and citation of publications on global sources.

The system's main goal is to provide users with up-to-date analytical information on indicators of international activity from multiple selected sources.

The system's main tasks include collecting data from diverse sources, primarily bibliographic and scientometric databases, aggregating received data based on user requests, and removing erroneous data and duplicates.

Since the system focuses on a varying number of sources, which differ in data structure and interaction interfaces, a flexible architecture that can be modified with minimal costs is a key requirement. This architecture allows for the inclusion of new data sources, exclusion of old and no longer needed functions, and modification of still relevant existing ones.

The developed system enables the collection of information about the international activities of individual scientists or institutions from sources that are heterogeneous in terms of storage form and composition.

The composition of information sources in the system can be expanded without interfering with the operation of other system components. Each individual processor of information from the sources can provide only partial data, which is supplemented by others. The system adapts to the user's request by taking data only from sources that meet the request's requirements and can satisfy them.

The system is also characterized by load adaptability and high throughput, achieved through the use of modern approaches in organizing the software system architecture.

The system offers users a single point of entry for requests for general and analytical information on the international activities of both individuals and institutions.

5. References

[1] Ford Neal, Parsons Rebecca, Kua Patrick, Building Evolutionary Architectures: Support Constant Change, O'Reilly Media, 2017

[2] Novikov D.A., Cybernetics: Navigator. History of cybernetics, contemporary state, prospects of development, M., 2016

[3] Chris Richardson, Microservices. Development and refactoring patterns, S.-P., Peter, 2019.

[4] Adam Belnar, Building Event-Driven Microservices: Leveraging Organizational Data at Scale. USA, O'Reilly Media, 2020

[5] Koval, O.V., Kuzminykh, V.O., Husyeva, I.I., Beibei, X., Shiwei, Z. Improving the Efficiency of Typical Scenarios of Analytical Activities. CEUR Workshop Proceedings, Vol-3241, 2021, pp. 123-132. URL: https://ceur-ws.org/Vol-3241/paper12.pdf

[6] Mouett E., Using Docker / trans. with English A. V. Snastyna; science ed. A. A. Markelov, M., DMK Press, 2017

[7] Rastrigin L.A., Ripa K.K., Automated random search theory, Riga, Zinatne, 1973

[8] Valeriy O. Kuzminykh, Alexander V. Koval, Mark V. Osypenko Methods of Machine Training on the Basis of Stochastic Automatic Devices in the Tasks of Consolidation of Data from Unsealed Sources. CEUR Workshop Proceedings, Vol-2067, 2017, pp. 63-68. URL: https://ceur-ws.org/Vol-2067/paper9.pdf

[9] Pethuru Raj, Jeeva S. Chelladhurai, Vinod Singh, Learning Docker, Packt Publishing, 2015

[10] Newman S. Building Microservices, St. Petersburg, Peter, 2016