Registry services and systems – conceptual and architectural aspects

B. SZAFRAŃSKI

boleslaw.szafranski@wat.edu.pl

Military University of Technology, Faculty of Cybernetics
Institute of Computer and Information Systems
Kaliskiego 2, 00-908 Warszawa, Poland

After discussing the function of public registers in the information infrastructure of the State, the paper focuses on the architectural aspects of ensuring the possibility of standardised reference to the services of public registry systems, despite the technological and functional differences between them. Taking this fact into account, the concept was presented to include, in the architecture, the platform for access registry services of an intermediary layer consisting of registry connectors, registry broker and universal registry services, whose (layer) main task is to "mask" the above differences by adapting the services of autonomous registries to the standard required by the main bus of this platform.

Keywords: public register systems, architecture, information system of the State.

DOI: 10.5604/01.3001.0015.8604

1. Introduction

The capacity of public administrations to handle public tasks electronically is an important factor in its assessment and a prerequisite for progress in building the information society [2]. Without a coherent and modern information infrastructure of public administration it will not be possible to have interoperable, secure and reliable interoperability of information systems of public entities in mutual relations and in relations with individuals and entities not belonging to public administration, especially entrepreneurs. In order to rationally direct the development of the State's information system, it is necessary to constantly conduct anticipatory research and conceptual works. Recognizing that public registers are essential for efficient, automated handling of public tasks [3], after discussing the functions of registers in the state's information infrastructure, attention was focused on the architectural aspects of interoperability, referencing and functioning of register systems[1]. It is precisely the need to address this kind of issues in study programmes and research works that is often raised by decision-makers of public and economic entities, in which the implementation of information processes is supported by an extensive IT technical and system infrastructure. They are also concerned about the qualifications of graduates of computer science departments at Polish universities, which are increasingly focusing on the application-based use of "small" mobile devices, abstracting from the knowledge necessary to design and operate large-scale IT infrastructures. The aim of this paper is to encourage students and PhD students, using the example of registry systems, to take up such issues for this class of information systems in their diploma and doctoral theses.

2. Functions of registers

The literature of the subject matter [1] most often distinguishes the following functions of public registers: information, identification, verification, control, classification, integration, standardisation and constituting. The services provided by information systems supporting the operation of public tasks, both in terms of their type and mode of operation, must have the properties to ensure that registers fulfil the roles that result from the previously identified functions of registers, at least to the same extent

[1] Registry system – software used to create and process the content of public registers for the purpose of handling public tasks.

as in the case of traditional handling of public tasks / services. After analysing the aforementioned functions of registers, the number of functions (roles) may be limited without harmful simplification to the following list: information, identification, verification (including control), integration (including standardisation and classification), constituting. For the purpose of approximating the issues, examples of types of registry services are given below for each of the following functions:

- information function: searching for and making available the values of the object(s) information features from one or more registers and downloading and making available statistics calculated on the basis of the content of the registers,
- identification function: provision of object identifier or list of objects' identifiers and verification of the existence of object or identifier registers,
- verification function: verification of information compliance, verification of data structure definitions, verification of compliance with the classification standard,
- integration function: making registry standards available, conversion of data types,
- function constituting: entry of object data, update of object data, confirmation of the presence of objects in the register.

3. Standardization of the public registers

The Act on the Computerization of Activities of Entities Performing Public Tasks (in short: the Act on Computerization) [3] refers to public registers through the provisions on the National Interoperability Framework, including in particular by setting minimum requirements for public registers. The minimum requirements have been laid down primarily because of the need to ensure the consistency of the operation of the electronic systems used to handle public tasks. Achieving the above-mentioned cohesion (interoperability) feature of the operation of electronic systems depends directly on registers and registry systems, as virtually all public tasks supported by information technology must refer to them.

The natural consequence of the existence of minimum requirements is the need to reflect them in the conceptual and design assumptions of registry systems. The degree and pace of this reflection must result from the rate of expansion of the set of public tasks to be carried out electronically and the scope and form of possible changes of minimum requirements. A close link between the minimum requirements and the concepts of registry systems should:

- to a greater extent prevent the creation of new identifiers used to identify objects described in public registers. Objects should be identified by means of identifiers taken from registers already existing and indicated in the minimum requirements,
- agree on the structure and formats of information items in public registers and in other information resources in the scope of information items listed in the minimum requirements,
- ensure that minimum requirements for statutory public registers are the main driver of the solutions used in registry systems.

Considering interoperability more broadly for practical reasons, there is also a need to use a less formal instrument, i.e. the so-called recommendations for interoperability developed within the public administration, which are the result of reconciliation procedures based on good design practices often recognized in the world[2]. Permanent and positively verified recommendations for interoperability may in the future act as minimum requirements or be included in the National Interoperability Framework.

4. Architectural aspects of integration of IT systems

Public administration IT systems cooperating in the processes of handling public tasks are dispersed and differ very often, for example due to the year of their production, in technical and technological terms. In such a diverse environment, the importance of choosing the appropriate architecture of the State information system is growing. In order to better illustrate the problem, three significantly different topologies of the interoperability of IT systems, i.e. full network topology, star topology and bus topology, will be briefly discussed. At the same time, they reflect the evolution from the simplest and oldest solution to a solution representing the current, most commonly used approach. In order to enhance transparency and to preserve

[2] An important role in this respect is played by "the Governmental and Self-Government Cooperation Line" initiative set up as a platform for the development of arrangements and guidelines in the field of computerisation of public tasks and services [2].

the synthetic coverage of the topic, each approach is presented in the form of a short, illustrated by a drawing, description of the essence of topology and a list of major disadvantages and advantages.

4.1. Full network topology

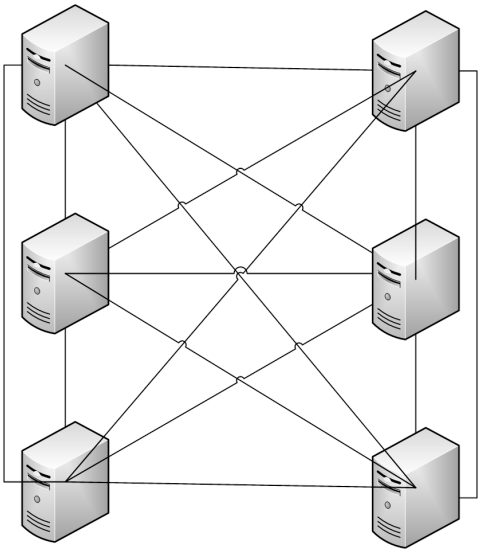

Fig. 1. The full network topology[3]

This is the oldest and simplest solution[4], which responds to the need to make services and/or data available bilaterally between IT systems intervening in the process of handling public tasks. In this architecture, the services (applications) of the systems communicate directly with each other, without the need for any intermediation. The "web" of bilateral connections seen in Figure 1 is formed. Each such connection entails the need to implement, on each side, tailor-made adaptors.

Disadvantages (examples):

- complex network of connections in the case of multiple communicating subsystems – each pair of systems must have dedicated interfaces for this cooperation ("web of connections"),
- lack of (most often) a common communication protocol for all connections, services (applications), they have at the same time to implement and maintain multiple interfaces, which increases costs and reduces the reliability of maintaining connections in the systems of cooperating entities,
- each change to the communication protocol necessitates modification of the above--mentioned interfaces (adaptors),
- the joint implementation of actions by two systems requires synchronous activity and availability at the same time.

Advantages (examples):

- high speed of communication due to lack of intermediary mechanisms,
- communication may consist in direct interaction between applications, i.e. in the call of specific services/functions of the bilaterally cooperating system,
- no costs to manufacture intermediating solutions.

4.2. The Star Topology

From the point of view of the evolution of information technology, this is a response to deficiencies in the management with architecture of the full network type. A central application (*hub*) is created to which all other subsystems are connected. The integration of systems takes place by means of adaptors for each subsystem and therefore no common message format is required.

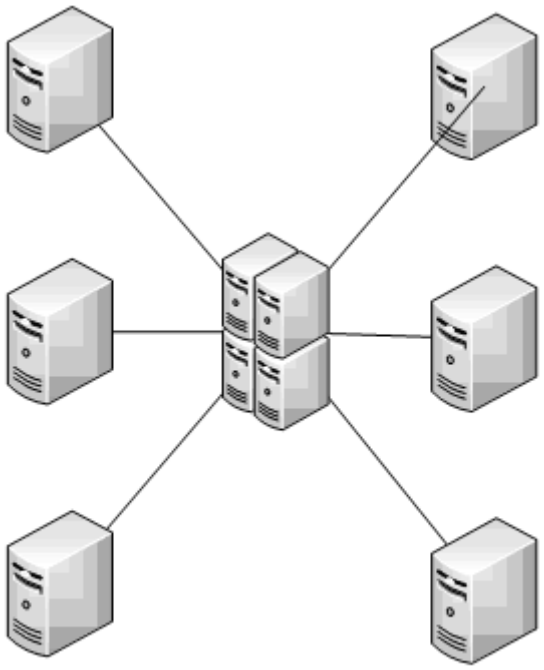

Fig. 2. The star topology

The central application provides the following functions:

[3] The drawings in the article were prepared by the author.

[4] This form of cooperation is also referred to as an action in line with the bilateral framework for interoperability.

- transformation of messages between formats used by subsystems,
- control over exchanged messages, direct them to relevant addressees (*routing*),
- asynchronous communication (queuing and buffering messages), which allowed for the implementation of connections in the absence of continuous availability of subsystems,
- central control, configuration and monitoring of the operation of the entire solution.

The central application, by obscuring differences between subsystems, may force these subsystems to use a specific protocol in communication with the intermediate platform. It can also make advanced tools available for creating complex interactions between subsystems, i.e. constructing and launching business processes using services from different subsystems.

Disadvantages (examples):
- the need to build and maintain an additional central application (intermediary platform),
- the need to agree on communication protocols (formats) between subsystems and the central application,
- the communication of subsystems is slowed down due to the need to interact with the intermediate application, which may become a "bottleneck" that reduces the efficiency of the entire solution.

Advantages (examples):
- the possibility of constructing new functions based on the use of resources of different subsystems in the implementation of complex processes,
- the obscuration of technological and conceptual differences of the cooperating subsystems,
- ensuring central management and supervision of communication between subsystems.

4.3. The Bus Topology

The bus topology is another step in the evolution of the architecture of IT systems. In this solution, the bus "enforces" cooperation of applications in accordance with one agreed message standard (format). This is an architectural option similar to the previous one (the star topology) in the sense that the bus can also be viewed as a central component, which is an intermediary in ensuring overall interoperability.

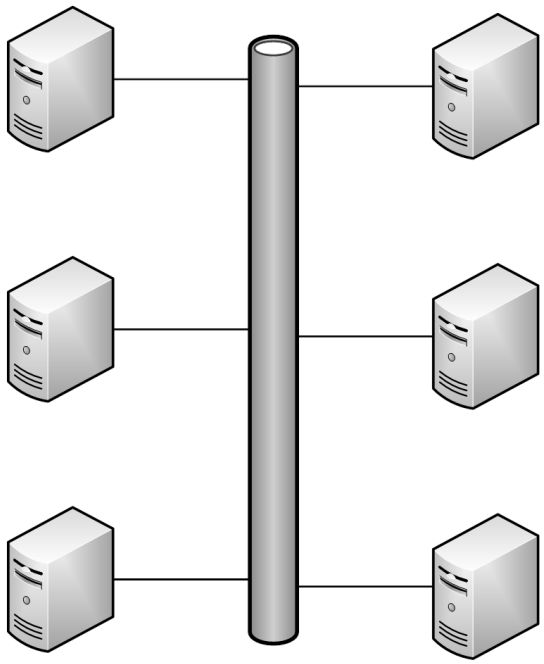

Fig. 3. The Bus Topology

The following distinction between the star topology and the bus topology is often accepted: all services available in the central system in the star architecture (e.g. messaging broker, process engine) are also available in the bus, but as services connected to it and not as directly implemented functions. This fact makes such architecture much more scalable and configurable. Its services can be implemented on many physical servers, thereby achieving overload resistance.

Disadvantages (examples):
- the need to build and maintain a bus,
- more complex and less transparent control of process implementation.

Advantages (examples):
- good scalability and configuration ability,
- resistance to overloads,
- ensuring central management and supervision of communication between subsystems due to the maintenance and availability of the meta-information layer.

5. Conceptual assumptions of the layer of universal services of access to registers – variants of interconnection of public registers

Public registers are resources held by many independent entities. Currently, their cooperation is still very limited. A single registry system often repeats part of the functionality already existing (implemented) in other systems (e.g. maintains overly subsets of data outside its domain resources instead of retrieving them from the reference registers when needed). At this point, it is worth noting that the architecture of the State's information infrastructure (both based on the star topology and the bus topology) should facilitate or even force a gradual process of standardising registers, enabling or facilitating their interoperable, automatic interaction. According to the previous justification, it was assumed that the bus topology makes it possible to connect the registry systems by means of the agreed "standard" for message exchange. Taking this into account, the following two options for cooperation between existing (or planned to be implemented) registry systems have been proposed:

- **a direct variant** in which registry systems adapt to interoperability requirements, bearing the full burden of compliance,
- **an intermediate layer variant** which acts as a platform for obscuring technological differences between registry systems.

5.1. Direct variant

In this variant, individual registry systems must be adapted to direct cooperation with the bus, accepting the imposed communication protocol. This means that the registry system concerned must be modified so that the functions to be made available on the platform can be activated by means of its own services. These services must be implemented in accordance with the standard and technology imposed by bus ownership. Each technical change (change of format, encryption, message header, etc.) will require modification in the registry system. On the organizational basis, a larger share of the responsibility for the operation of the connection lies with the registry manager, which, with such a large number of registers, will result in a very dispersed responsibility for the operation of such a solution. There may also be burdensome and labour-intensive modifications of services, e.g. a change in the scope of data on either side may result in unavoidable modifications of the registry system and the bus. In addition, in the case of small (low-budget) registry systems, some of the following problems may arise:

- lack of adequate, sufficiently reliable and prepared hardware infrastructure for continuous operation,
- lack of support for multi-access and transaction abilities necessary to ensure data consistency in conditions of simultaneous processing,
- lack of extensive access control mechanisms,
- lack of support for communication in the required format (e.g. in XML format for business-level messages) and difficulties in implementing message conversion due to lack of standard libraries for the technology in which the registry system was developed.

5.2. Variant with intermediate layer

This architectural variant distinguishes the software layer dedicated to cooperation with registry systems. It is composed of an intermediary system (messaging broker) and registry connectors. This allows the registry systems to waive their compliance with the requirement of "having" a uniform, agreed interface of cooperation with the registry services provisioning platform. Given that standard ("standardised") registry services should be available at the main bus level, this is the intermediate layer, which takes over an adjustment role (adapting to the required standard), which obscures inter-registry differences. It comprises, shown in Figure 4, the registry connectors, the registry broker and the universal (standard) registry services layer.

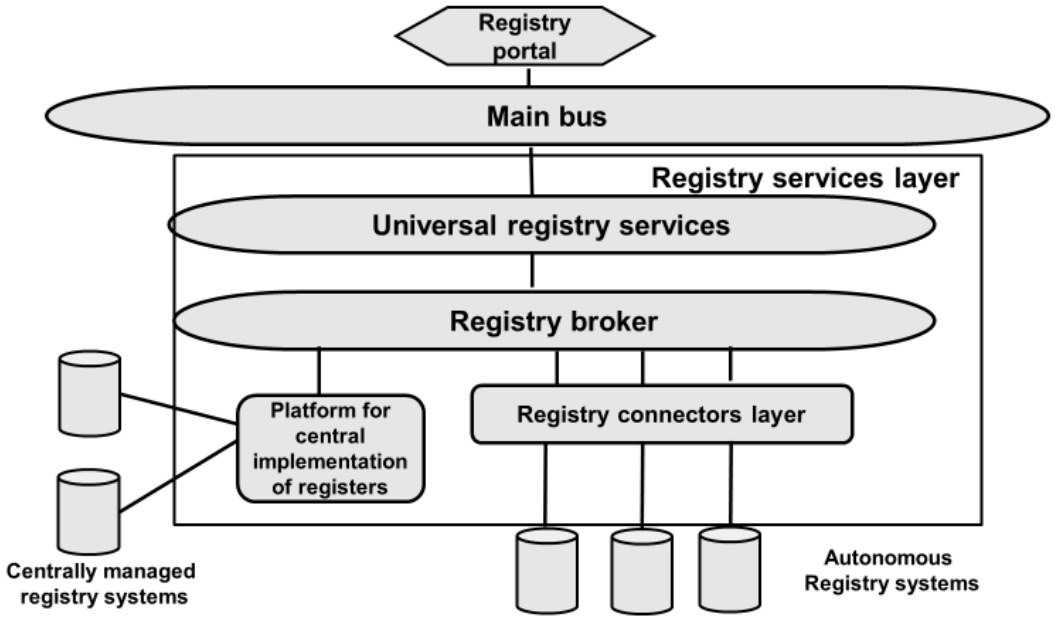

Fig. 4. The registry services layer location in the registry management system

5.2.1. Registry connectors

Registry connectors are components of the intermediate layer and are responsible for communicating the registry system with the central intermediary. When communicating with registry systems, the connectors will fulfil, together with the registry broker, a role that adapts communication and specific, differing services of registry systems to the standard required by the main bus. They may distinguish between connectors specific to individual registry systems or connectors common to a group of registry systems identical in terms of performance technology and functionality of services.

5.2.2. The Registry Broker

The registry broker acts as an intermediary in bilateral communication between autonomous registry systems and the main bus, creating an environment, in which registry connectors are firstly developed and then subsequently operated. The functions performed by the registry broker include, inter alia:

- cooperation with registry connectors,
- processing of bus messages in order to interact with registry connectors,
- handling security mechanisms, registering calls, etc.,
- provision of metadata relating to registry services to the bus.

Fulfilling the above functions will require components to be built in at the intermediary system level, among others:

- components supporting the synchronous cooperation of registry services (if required),
- components managing (configuring) parameters of communication with connectors, including, inter alia, connecting new connectors or modifying existing ones,
- components ensuring two-way translation of messages between the main bus and the registry systems,
- components creating and making available the content of the metadata repository describing registry services, in particular in the context of communication parameters and configuration of individual connectors,
- components adjusting the security features (e.g. authorizations, authentication) used by the connectors to the security features required by the main bus.

5.2.3. Universal registry services

From the point of view of the users of the platform for the standard provision of registry services (both persons and other IT platforms/systems), it is important to ensure that it is possible to perform on any registry from any domain, at least the services[5] that implement operations:

[5] A set of operations is called CRUD, from English words: Create, Read, Update, Delete.

- creating information in the register,
- deletion of information from the register,
- updating information in the register,
- searching and making available registry data on the basis of.

The creation of a common standard message format for these elementary services will enable:

- simplification of defining complex processes using access to registers from the main bus level,
- usage of advanced *workflow* design tools, in which multiple registry systems interact.

6. Conclusion

The paper presents a proposal for architecture suitable for the standard provision of registry services, despite the presence of technologically and functionally diverse registry systems in the information infrastructure. When discussing this proposal, the focus was on the components whose introduction into the architecture enables such standard access to registry services. As per this approach:

- the architecture distinguishes components responsible for communication with registry systems, which such components include the layer of universal registry services, the registry broker layer and the registry connector layer;
- it should be assumed that registry connectors should be implemented in a way that limits as far as possible the scope of modification of existing autonomous registry systems;
- the set of registry services available on such a designed platform should contain basic universal registry services with the functionality necessary for modelling (defining) complex processes that take into account the interaction of primary services provided by autonomous registry systems.

The architectural options discussed do not solve, for example, problems related to registers which, for various reasons (e.g. technological, financial, data quality, compliance with the requirement of continuous availability, ...) will not be included in the set of autonomous registry systems during the relevant time horizon. For this reason, the proposed architecture takes into account the possibility of using the central implementation function of the registers. In that case, instead of carrying out costly work on its own registry system, the registry keeper may use the tools available on the proposed platform to establish, manage and make available registry services dedicated to his registry.

Finally, it should be stressed that, by focusing on registry issues, the paper deliberately omits many other problems, which are important for the functioning of this type of platform, such as ensuring security or asynchronous message handling.

7. Bibliography

[1] Oleński J., *Infrastruktura informacyjna państwa w globalnej gospodarce*, Uniwersytet Warszawski Wydział Nauk Ekonomicznych, Warszawa 2006. [Eng. Oleński J., *Infrastructures of information state in the global economy*, University of Warsaw and Faculty of Economic Sciences, Warsaw 2006].

[2] Szafrański B., Weydmann R., *Infrastruktura informacyjna nowoczesnego państwa*, Wojskowa Akademia Techniczna, Warszawa 2019. [Szafrański B., Weydmann R., *Infrastructures of the modern State*, Military University of Technology, Warsaw 2019].

[3] Ustawa z dnia 17 lutego 2005 r. o informatyzacji działalności podmiotów realizujących zadania publiczne (Dz.U.2005.64.565 z późn. zm.) [Eng. Act of February 17, 2005 on the computerization of the activities of entities performing public tasks].

[4] Ustawa z dnia 6 września 2001 r. o dostępie do informacji publicznej (Dz.U. 2001 Nr 112, poz. 1198).

[5] Ustawa z dnia 24 września 2010 r. o ewidencji ludności (tekst jedn. Dz.U. z 2015 r., poz. 388 z późn. zm.).

Usługi i systemy rejestru – aspekty koncepcyjne i architektoniczne

B. SZAFRAŃSKI

Po omówieniu funkcji rejestrów publicznych w infrastrukturze informacyjnej państwa w artykule skupiono się na architektonicznych aspektach zapewnienia możliwości ujednoliconego odniesienia do usług systemów rejestrów publicznych, pomimo różnic technologicznych i funkcjonalnych między nimi. Mając na uwadze ten fakt, przedstawiono koncepcję włączenia do architektury platformy dostępu do usług rejestrowych warstwy pośredniczącej składającej się z konektorów rejestrów, brokera rejestrów i uniwersalnych usług rejestrowych, których głównym zadaniem (warstwy) jest „maskowanie" powyższych różnic poprzez dostosowanie usług rejestrów autonomicznych do standardu wymaganego przez magistralę główną tej platformy.

Słowa kluczowe: systemy rejestrów publicznych, architektura, system informacyjny państwa.